

UNIVERSITY OF CALIFORNIA

Los Angeles

**Designing Autonomic Wireless Multi-Hop Networks
for Delay-Sensitive Applications**

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Electrical Engineering

by

Hsien-Po Shiang

2009

© Copyright
by Hsien-Po Shiang
2009

The dissertation of Hsien-Po Shiang is approved.

Mario Gerla

Jason Speyer

Kung Yao

Mihaela van der Schaar, Committee Chair

University of California, Los Angeles

2009

To my parents

TABLE OF CONTENTS

1. Introduction	1
I. Dissertation Goal	1
II. Challenges in Dynamic Multi-hop Wireless Networks	3
III. Organization of the Dissertation	5
2. Cross-layer Optimization for Multimedia Streaming in Multi-Hop Wireless Networks Based on Priority Queuing	12
I. Introduction	12
II. Multi-user Video Streaming Specification	18
A. Video priority classes	18
B. Network specification	20
C. Cross-layer joint transmission strategy vector	20
D. Problem formulation	22
III. A Distributed Packet-Based Solution Based on Priority Queuing	24
A. Required information feedback among network nodes for the distributed solution	24
B. Self-learning policy for dynamic routing	25
C. Delay-driven policy for MAC/PHY	27
D. Complexity analysis in terms of route selection	28
IV. Multi-Hop Priority Queuing Analysis for Multimedia Transmission	29
A. Assumptions for priority queuing analysis	29
B. Priority queuing analysis for an elementary structure	31
C. Generalization to the multi-hop case	33
V. Priority Queuing Analysis Considering Interference of Wireless Networks	35
A. Incidence matrix and interference matrix	35
B. Priority queuing with virtual-queue service time modification	37

VI. Convergence Discussion	40
VII. Simulation Results	41
VIII. Conclusions	47
IX. Appendix	48
3. Autonomic Decision Making for Transmitting Delay-Sensitive Applications	
Based on Markov Decision Process	49
I. Introduction	49
II. Autonomic Decision Making Problem Formulation	54
A. Delay-sensitive application characteristics	54
B. Autonomic multi-hop network setting	55
C. Actions of the autonomic wireless nodes	56
D. Problem formulation	56
III. Distributed Markov Decision Process Framework	59
A. States of the autonomic wireless nodes	60
B. Centralized Markov decision process formulation	62
C. Distributed Markov decision process formulation	63
D. Convergence of the distributed Markov decision process	65
IV. On-line Model-Based Learning for Solving the Distributed Markov decision process	66
A. Model-free reinforcement learning	68
B. Model-based reinforcement learning	69
C. Upper and lower bounds of the model-based learning approach	72
V. Simulation Results	73
A. Simulation results for different network topologies	73
B. Comparisons of the learning approaches	75
C. Heterogeneous learning	79
D. Simulation results for the upper and lower bounds	80

VI. Conclusions	81
VII. Appendix A	81
VIII. Appendix B	85
4. Adapting the Information Horizon – Risk-Aware Scheduling for Multimedia Streaming over Multi-Hop Wireless Networks	87
I. Introduction	87
II. Problem Formulation and System Description	91
A. Overlay network specification	92
B. Centralized cross-layer optimization for multi-user wireless video transmission	93
C. Proposed distributed cross-layer adaptation based on information feedback	94
III. Impact of Accurate Network Status	97
A. Information feedback frequencies and information horizon	97
B. The impact of various information horizons	99
C. Distributed cross-layer adaptation based on information feedback with larger information horizons	100
IV. Risk-Aware Scheduling for Multimedia Streaming	101
A. Risk estimation based on priority queuing analysis	102
B. Feedback-Driven Scheduling	104
V. Risk-Aware MAC Layer Retransmission Strategy	108
VI. Overhead Analysis for Information Feedback	109
VII. Simulation Results	110
VIII. Conclusions	114
5. Feedback-Driven Interactive Learning in Wireless Networks	115
I. Introduction	115
II. Network Settings and Problem Formulation	120

A.	Network settings	120
B.	Actions and strategies	121
C.	Utility function definition	122
D.	Problem formulation	123
E.	Learning efficiency	125
III.	Information Feedback for Interactive Learning	126
A.	Characterization of information feedback	126
B.	Cost-efficiency tradeoff when adjusting the information feedback	128
IV.	Interactive Learning with Private Information Feedback	130
A.	Reinforcement learning based on private information feedback	131
B.	Adaptive reinforcement learning	132
V.	Interactive Learning with Public Information Feedback	133
A.	Action learning based on public information feedback	134
B.	Adaptive action learning	136
VI.	Simulation Results	137
A.	Comparisons among different learning approaches	138
B.	Convergence of the learning approaches	141
C.	Adaptive reinforcement learning using different time scales	141
D.	Adaptive action learning from different neighboring users	143
E.	Mobility effect on the interactive learning efficiency	144
VII.	Conclusions	144
VIII.	Appendix A	146
IX.	Appendix B	147
X.	Appendix C	148
XI.	Appendix D	149
6.	Resource Management in Single-Hop Cognitive Radio Networks	150
I.	Introduction	150

II.	Modeling the Cognitive Radio Networks as Multi-Agent Interactions	154
A.	Agents in cognitive radio networks	154
B.	Models of the dynamic resource management problem	155
III.	Dynamic Resource Management for Heterogeneous Secondary Users using Priority Queuing	157
A.	Prioritization of the users	157
B.	Heterogeneous channel conditions	158
C.	Goals of the heterogeneous users	158
D.	Example of three priority classes with different utility functions	160
E.	Priority virtual queue interface	161
IV.	Priority Queuing Analysis for Delay-Sensitive Multimedia Users	163
A.	Traffic models	164
B.	Priority virtual queue analysis	166
C.	Information overhead and aggregate virtual queue effects	168
V.	Dynamic Channel Selection with Strategy Learning	170
VI.	Simulation Results	174
A.	Impact of the delay sensitive preference of the applications	175
B.	Impact of the primary users	178
C.	Comparisons with other cognitive radio resource management solutions	179
VII.	Conclusions	181
VIII.	Appendix	182
7.	Resource Management in Multi-Hop Cognitive Radio Networks	184
I.	Introduction	184
II.	Main Challenges and Related Work	186
A.	Main challenges in multi-hop cognitive radio networks	186
B.	Related work	187

III. Multi-Hop Cognitive Radio Network Settings	189
A. Network entities	189
B. Source traffic characteristics	190
C. Multi-hop cognitive radio network specification	191
D. Interference characterization	191
E. Actions of the nodes	194
IV. Resource Management Problem Formulation	195
V. Distributed Resource Management with Information Constraints	198
A. Considered medium access control	199
B. Benefit of acquiring information and information constraints	200
C. Cost of information exchange	204
VI. Distributed Resource Management Algorithms	206
A. Resource management algorithms	207
B. Adaptive fictitious play	210
C. Information exchange overhead reduction	213
VII. Simulation Results	214
A. Reward of cost of information exchange	216
B. Application layer performance with different information horizons and interference ranges	216
C. Reducing the frequency of learning	219
D. Impact of the primary users	220
E. Impact of the mobility	221
VIII. Conclusions	222
8. Conjecture-Based Channel Selection in Multi-Channel Wireless Networks	224
I. Introduction	224
II. Problem Formulation for Foresighted Channel Selection	229

A. Network model	229
B. Conventional centralized decision making	231
C. Conventional distributed decision making	232
D. Foresighted decision making	234
III. Conjecture-Based Channel Selection Game and Conjectural Equilibrium	235
IV. Distributed Channel Selection When There is Only One Foresighted User	237
A. Belief function when only one user is foresighted	237
B. Linear regression learning to model the belief function	239
C. Altruistic foresighted user	240
D. Self-interested foresighted user	242
V. Distributed Channel Selection When There Are Multiple Foresighted Users	245
A. Performance degradation when multiple users learn	245
B. Reaching system-wise Pareto optimal solution when every user builds belief using a prescribed rule	246
C. On-line coordination of the foresighted channel selection	248
VI. Simulation Results	251
A. Single foresighted user scenario	252
B. Multiple foresighted user scenario	255
VII. Conclusions	257
VIII. Appendix A	258
IX. Appendix B	258
X. Appendix C	259
9. Conclusions	261
Bibliography	267

LIST OF FIGURES

Fig. 1.1 The autonomic decision making framework for delay-sensitive applications.	3
Fig. 1.2 The organization of the dissertation.	5
Fig. 2.1 Illustrative example of the considered directed acyclic multi-hop networks.	20
Fig. 2.2 Integrated block diagram of the proposed distributed per-packet algorithm.	25
Fig. 2.3 Priority queuing analysis system map.	31
Fig. 2.4 The elementary structure.	31
Fig. 2.5 (a) Network settings of the elementary structure. (b) Analytical average end-to-end waiting time of the 8 video classes.	43
Fig. 2.6 (a) Network settings of the 6-hop overlay network (by cascading the elementary structure). (b) Analytical average end-to-end waiting time of the 8 video classes.	45
Fig. 2.7 (a) Primary paths of the 6-hop overlay network using self-learning policy. (b) Analytical average end-to-end waiting time of the 8 video classes.	46
Fig. 3.1 (a) Conventional distributed decision making of an agent. (b) Proposed foresighted decision making of an agent.	58
Fig. 3.2 Expected delay evaluation and the required local information.	61
Fig. 3.3 Proposed decentralized Markov decision process framework and the necessary information exchange among the agents.	64
Fig. 3.4 System diagram of the proposed model-based online learning approach at the agent m_h .	67
Fig. 3.5 (a) 6-hop network topology (b) MDP delay values of the first five priority classes.	74
Fig. 3.6 (a) 2-cluster skewed network topology (b) MDP delay values of the first five priority classes.	75
Fig. 3.7 Comparisons of the MDP delay values using different learning approaches.	76
Fig. 3.8 Comparisons of the expected end-to-end delay using different learning approaches.	77
Fig. 3.9 Source node of packets in class C_1 , C_4 disappears after $t = 60$.	78
Fig. 3.10 The upper and the lower bounds of the MDP delay values for the first priority class traffic at different hops.	80

Fig. 4.1 The directed acyclic multi-hop overlay network for an exemplary wireless infrastructure. (a) Actual network topology that has 2 source-destination pairs, 5 relay nodes. (b) Overlay network topology that has 2 source-destination pairs, 6 relay nodes (with one virtual node in the 1-hop intermediate nodes).	93
Fig. 4.2 Illustrative example of an application layer overlay network with information horizon $\vec{h} = 2$.	96
Fig. 4.3 System map for the IFDS packet scheduling.	105
Fig. 4.4 Risk estimation vs. time interval for 2 users.	107
Fig. 4.5 Simulation settings of a 6-hop overlay network with 2 video sequences.	110
Fig. 4.6 Y-PSNR vs. various information horizon cases under different network transmission efficiencies	113
Fig. 5.1 (a) Conventional distributed power control. (b) Payoff-based interactive learning with private information feedback. (c) Model-based interactive learning with public information feedback.	118
Fig. 5.2 System diagram of the dynamic joint power-spectrum resource allocation.	120
Fig. 5.3 (a) Throughput B_v vs. P_v in a selected frequency channel f_v with fixed interference. (b) Utility u_v vs. P_v in a selected frequency channel f_v with fixed interference.	122
Fig. 5.4 Interactions among users and the foresighted decision making based on information feedback.	125
Fig. 5.5 Examples of different types of information feedback $\mathcal{Z}_v^t$.	127
Fig. 5.6 System block diagram for the adaptive interactive learning for dynamic resource management.	129
Fig. 5.7 Topology settings for the simulation.	137
Fig. 5.8 Average utility vs. time slot of the proposed algorithms when $T = 700$ Kbps.	141
Fig. 5.9 Performance of user m_1 adopting adaptive reinforcement learning with private information feedback using different ω_1 .	142
Fig. 5.10 Performance of user m_1 adopting adaptive action learning with public information feedback using different $ \mathbf{V}_1^t $.	143
Fig. 5.11 Average utility over time using the adaptive interactive learning when receivers have mobility ($T = 2100$ Kbps) (a) $\nu = 0.5$, (b) $\nu = 1$, (c) $\nu = 2$ (m/time slot).	145
Fig. 6.1 An illustration of the considered network model.	155

Fig. 6.2 The architecture of the proposed dynamic resource management with priority virtual queue interface.	162
Fig. 6.3 Actions of the secondary users a_{ij} and their physical queues for each frequency channel	164
Fig. 6.4 The block diagram of the priority virtual queue interface and dynamic strategy learning of a secondary user.	172
Fig. 6.5 Analytical expected delay of the secondary users with various strategies in different frequency channels, shadow part represents a bounded delay below the delay deadline (stable region).	176
Fig. 6.6 (a) Simulation results of the DSL algorithm – strategies of the secondary users and the utility functions of less delay-sensitive applications ($\theta_i = 0.2, \sigma = 0.05, \chi_{ij} = 0$). (b) Simulation results of the DSL algorithm – strategies of the secondary users and the utility functions of delay-sensitive applications ($\theta_i = 0.8, \sigma = 0.05, \chi_{ij} = 0$).	177
Fig. 6.7 Steady state strategies of the secondary users and the utility functions vs. the normalized loading of PU_1 for delay-sensitive applications ($\theta_i = 0.8, \sigma = 0.05, \chi_{ij} = 0.02$).	178
Fig. 7.1 A simple multi-hop cognitive radio network with three nodes and two frequency channels.	193
Fig. 7.2 Transmission time line at the node n with local information $\mathcal{L}_n$.	199
Fig. 7.3 Example of the static reward of information $J_n(k, \mathcal{Z}_n(x))$, dynamic reward of information $J_n^d(k, \mathcal{Z}_n(x))$ and optimal expected delay $K_n(k, x)$ (where the information horizon $h_n(k, \nu) = 3$, average packet length $L_k = 1000$ bytes, and average transmission rate $T = 6$ Mbps over the multi-hop network).	201
Fig 7.4 (a) 2-hop information cell network without information exchange mismatch problem. (b) 1-hop information cell network with information exchange mismatch problem.	204
Fig. 7.5 System diagram of the proposed distributed resource management.	206
Fig. 7.6 Block diagram of the proposed distributed resource management at network node n .	210
Fig. 7.7 (a). Block diagram of the proposed distributed resource management algorithm using the AFP. (b). Impact of the network variation on the FP and the video performance.	211
Fig. 7.8 Wireless network settings for the simulation of two video streams.	215
Fig. 7.9 Reward J_n^d and cost J_n^c of different information horizon at different node for video V_1 .	215

Fig. 7.10 (a) Packet loss rate vs. average transmission bandwidth using different approaches ($H_n^I = 80$ meters). (b) Packet loss rate vs. average transmission bandwidth using different approaches ($H_n^I = 40$ meters).	218
Fig. 7.11 Packet loss rate vs. learning frequency b_n / c (average $T = 5.5$ Mbps, $H_n^I = 80$ meters).	219
Fig. 7.12 Packet loss rate vs. time fraction ρ of the primary users occupying frequency channel F_1 around network node $n = 7, 11, 12$ (average $T = 5.5$ Mbps, $b_n / c = 1$, $H_n^I = 80$ meters).	220
Fig. 7.13 Packet loss rate vs. mobility v of the secondary users (network relays) (average $T = 8$ Mbps, $\rho = 0$, $b_n / c = 1$, $H_n^I = 80$ meters).	222
Fig. 8.1 Considered queuing model for multi-user channel access.	230
Fig. 8.2 Block diagram of the (a) myopic channel selection and (b) foresighted channel selection.	235
Fig. 8.3 An illustrative example of the solutions in the utility domain for a 2-user case (v_i is the foresighted user).	244
Fig. 8.4 Flowchart of the on-line foresighted channel selection procedure.	251
Fig. 8.5(a)(d) The action of the foresighted user v_1 over time, while participating in the channel selection game [(a) in network setting 1, (d) in network setting 2]. (b)(c)(e)(f) The actual remaining capacity C_{1j} and the estimated linear belief function $\tilde{C}_{1j}$, $j = 1, 2$ [(b)(c) in network setting 1, (e)(f) in network setting 2].	253
Fig. 8.6 Reaching the system-wise Pareto optimal solution and the Stackelberg Equilibrium.	254
Fig. 8.7 Delay of the foresighted user at different equilibrium for various numbers of myopic users in the network.	255

LIST OF TABLES

TABLE 2.1 THE CHARACTERISTIC PARAMETERS OF THE VIDEO CLASSES OF THE TWO VIDEO SEQUENCES.	41
TABLE 2.2 ANALYTICAL AND SIMULATION RESULTS FOR UNIFORM RELAY SELECTING PARAMETERS WITH DIFFERENT NETWORK EFFICIENCIES OVER THE ELEMENTARY STRUCTURE.	43
TABLE 2.3 ANALYTICAL AND SIMULATION RESULTS FOR UNIFORM RELAY SELECTING PARAMETERS WITH DIFFERENT NETWORK EFFICIENCIES OVER THE 6-HOP NETWORK.	45
TABLE 2.4 ANALYTICAL AND SIMULATION RESULTS FOR SELF-LEARNING POLICY RELAY SELECTING PARAMETERS WITH DIFFERENT NETWORK EFFICIENCIES (THE ANALYTICAL RESULTS ARE APPROXIMATED ACCORDING TO THE PRIMARY PATH SELECTED BY THE SELF-LEARNING POLICY).	46
TABLE 2.5 COMPARISON OF THE DYNAMIC SELF-LEARNING POLICY WITH THE CONVENTIONAL FIXED SINGLE- PATH AND MULTI-PATH ALGORITHMS (USING THE SAME NETWORK SETTINGS AS IN TABLE 2.4).	47
TABLE 3.1. COMPLEXITY SUMMARY OF THE MODEL-FREE REINFORCEMENT LEARNING	69
TABLE 3.2. COMPLEXITY SUMMARY OF THE MODEL-BASED REINFORCEMENT LEARNING	71
TABLE 3.3. THE CHARACTERISTIC PARAMETERS OF THE DELAY-SENSITIVE APPLICATIONS.	73
TABLE 3.4 THE RESULTS OF HETEROGENEOUS LEARNING SCENARIOS.	79
TABLE 4.1 DESCRIPTIONS FOR THE FOUR CASES OF THE SIMULATION RESULTS ($t_{SI} = 100\text{ms}$).	111
TABLE 4.2 SIMULATION RESULTS FOR IFDS SCHEDULING WITH VARIOUS INFORMATION HORIZONS AND DIFFERENT NETWORK EFFICIENCIES.	113
TABLE 5.1 COMPARISONS OF THE PROPOSED LEARNING ALGORITHMS.	137
TABLE 5.2 SIMULATION RESULTS OF THE FIVE SCHEMES WHEN $T = 700\text{ KBPS}$.	139
TABLE 5.3 SIMULATION RESULTS OF THE FIVE SCHEMES WHEN $T = 2100\text{ KBPS}$.	140
TABLE 5.5 SUMMARY OF THE USED NOTATIONS OF CHAPTER 5.	146
TABLE 6.1 SIMULATION PARAMETERS OF THE SECONDARY USERS.	175
TABLE 6.2 SIMULATION PARAMETERS OF THE PRIMARY USERS.	175
TABLE 6.3 COMPARISONS OF THE CHANNEL SELECTION ALGORITHMS FOR DELAY-SENSITIVE APPLICATIONS WITH $N = 6, M = 10$.	180

TABLE 6.4 COMPARISONS OF THE CHANNEL SELECTION ALGORITHMS FOR DELAY-SENSITIVE APPLICATIONS WITH $N = 20 + r, M = 10$, WHERE r IS THE SECONDARY USERS WITH DELAY INSENSITIVE $\theta_k = 0$ APPLICATIONS.	181
TABLE 7.1. Y-PSNR OF THE TWO VIDEO SEQUENCES USING VARIOUS APPROACHES ($H_n^I = 40$ METERS).	216
TABLE 7.2. Y-PSNR OF THE TWO VIDEO SEQUENCES USING VARIOUS APPROACHES ($H_n^I = 80$ METERS).	217
TABLE 8.1. CONSIDERED NETWORK SETTINGS.	251
TABLE 8.2. RESULTS AT DIFFERENT EQUILIBRIUMS.	254
TABLE 8.3. NUMERICAL RESULTS IN DIFFERENT SCENARIOS.	257

ACKNOWLEDGEMENTS

I would like to start by thanking my advisor Mihaela van der Schaar for her enthusiasm and support through the course of my PhD. She has always encouraged me to look beyond the details of specific problems and to see the big picture. Under her guidance, I was able to complete papers on a wide range of research topics. Her breath and creativity have been constant sources of inspiration for me throughout my stay here at UCLA.

I would also like to thank Professors Jason Speyer, Kung Yao, and Mario Gerla for their interest in my work, and their time invested to be part of my committee. Their helpful comments and advices have guided my work.

I would also like to thank my labmates Fangwen Fu, Hyunggon Park, Nick Mastronarde, Brian Foo, Yi Su, and Zhichu Lin for helping me to think through many of my research ideas, and for helping to peer review my papers before submission. I will also treasure the personal times spent with each of them, and how they have enriched my life through both meaningful and fun conversations. I would also like to thank my supervisor at Intel, Dilip Krishnaswamy, for giving me the opportunity to do very interesting research with them in their research group.

Finally, I would like to thank my family, my mom and dad, and my sister Judy, for their continued love and support for me during the course of my PhD career. I would like to dedicate my dissertation to my family.

VITA

- 2000** B.A., Electrical Engineering, National Taiwan University
Taipei, Taiwan
- 2002** M.A., Electrical Engineering, Communications,
National Taiwan University, Taipei, Taiwan
Second Lieutenant, Information Office, Army
Taiwan Ministry of National Defense, Taiwan
- 2004** Software Engineer, High Tech Computer Corp. Taiwan
- 2005** Teaching Assistant, Electrical Engineering Dept., UCLA
Joined the Multimedia Communication and System Lab.
under Prof. Mihaela van der Schaar.
- 2006** Teaching Assistant, Electrical Engineering Dept., UCLA
Summer interned at Intel Corp., Folsom, CA
- 2007** Received Award, Emerging Leaders in Multimedia,
from IBM T. J. Watson Research Center, Hawthorne, NY

PUBLICATIONS

D. Krishnaswamy, H.-P. Shiang, J. Vicente, W. S. Conner, S. Rungta, W. Chan and K. Miao, "A Cross-Layer Cross-Overlay Architecture for Proactive Adaptive Processing in Mesh Networks," in *2nd IEEE Workshop on Wireless Mesh Networks (WiMesh 2006)*, Sep 2006.

Y.-L. Li, H.-H. Chen, Y. Chen, H.-P. Shiang, Y. Lee, "Low-Complexity Receiver Design for OFDM Packet Transmission with Mobility Support," *IEEE Global Telecommunications Conference*, vol. 1, pp. 599-604, Nov. 2002.

H.-P. Shiang, D. Krishnaswamy, and M. van der Schaar, "Quality-aware Video Streaming over Wireless Mesh Networks with Optimal Dynamic Routing and Time Allocation," in *Proceedings of the 40th Asilomar Conference on Signals, Systems, and Computers*, Oct 2006.

H.-P. Shiang, J.-S. Liu, and Y.-R. Chien, "Estimate of minimum distance between convex polyhedra based on enclosed ellipsoids," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000 (IROS 2000)*, vol. 1, pp. 739 - 744, Oct 2000.

H.-P. Shiang, M. van der Schaar, "Multi-user Video Streaming over Multi-hop Wireless Networks: A Cross-layer Priority Queuing Approach," in *IEEE Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2006)*, pp. 255-258, Dec 2006.

H.-P. Shiang, M. van der Schaar, "Multi-user Video Streaming over Multi-hop Wireless Networks: A Distributed, Cross-layer Approach Based on Priority Queuing," *IEEE Journal of Selected Areas in Communications*, vol. 25, no. 4, pp. 770-785, May 2007.

H.-P. Shiang, M. van der Schaar, "Informationally Decentralized Video Streaming over Multi-hop Wireless Networks," *IEEE Transactions on Multimedia*, vol. 9, no. 6, pp. 1299-1313, Oct 2007.

H.-P. Shiang, M. van der Schaar, "Queuing-Based Dynamic Channel Selection for Heterogeneous Multimedia Applications over Cognitive Radio Networks," *IEEE Transactions on Multimedia*, vol. 10, no. 5, pp. 896-909, Aug. 2008.

H.-P. Shiang, M. van der Schaar, "Delay-Sensitive Resource Management in Multi-hop Cognitive Radio Networks" in *IEEE Dynamic Spectrum Access Networks (DySPAN 2008)*, Oct. 2008.

H.-P. Shiang, M. van der Schaar, "Dynamic Channel Selection for Multi-user Video Streaming over Cognitive Radio Networks," in *Proc. Int. Conf. On Image Processing. (ICIP 2008)* Oct. 2008.

H.-P. Shiang, M. van der Schaar, "Risk-aware scheduling for multi-user video streaming over wireless multi-hop networks," in *IS&T/SPIE Visual Communications and Image Processing (VCIP 2008)*, San Jose, Jan 2008.

H.-P. Shiang, M. van der Schaar, "Conjecture-Based Channel Selection Game for Delay-Sensitive Users in Multi-Channel Wireless Networks" in *International Conference on Game Theory for Networks (GameNets 2009)*, 2009. (Invited paper)

H.-P. Shiang, W. Tu, M. van der Schaar, "Dynamic Resource Allocation of Delay Sensitive Users Using Interactive Learning over Multi-carrier Networks," in *Proc. Int. Conf. Commun. (ICC 2008)*, May 2008.

H.-P. Shiang, M. van der Schaar, "Distributed Resource Management in Multi-hop Cognitive Radio Networks for Delay Sensitive Transmission," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 941-953, Feb 2009.

H.-P. Shiang, M. van der Schaar, "Feedback-Driven Interactive Learning in Dynamic Wireless Resource Management for Delay Sensitive Users," *IEEE Transactions on Vehicular Technology*, to appear.

H.-P. Shiang, M. van der Schaar, "Information-Constrained Resource Allocation in Multi-Camera Wireless Surveillance Networks," submitted to *IEEE Transactions on Circuits and Systems for Video Technology*.

H.-P. Shiang, M. van der Schaar, "Conjecture-Based Channel Selection for Autonomous Delay-Sensitive Users in Multi-Channel Wireless Networks," submitted to *IEEE Transactions on Networking*.

J. Wu, H.-P. Shiang, K. T. Chen, and H. W. Tsao, "Delay and Throughput Analysis of the High Speed Variable Length Self-Routing Packet Switch," *IEEE workshop on High Performance Switching and Routing (HPSR 2002)*, pp:314 - 318, May 2002.

ABSTRACT OF THE DISSERTATION

Designing Autonomic Wireless Multi-Hop Networks
for Delay-Sensitive Applications

by

Hsien-Po Shiang

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2009

Professor Mihaela van der Schaar, Chair

Emerging multi-hop wireless networks provide a low-cost and flexible infrastructure that can be simultaneously utilized by multiple users for a variety of applications, including delay-sensitive applications, such as multimedia streaming, mission-critical applications, etc. However, this wireless infrastructure is often unreliable and provides dynamically varying resources with only limited QoS support.

To improve the performance of the delay-sensitive applications and to support timely reaction to the network dynamics, the multi-hop network needs to be composed of autonomic nodes (agents), which can *adapt, make their own transmission decisions and negotiate their wireless resources* based on their available local information. Current wireless networking research has focused on coping with the environment disturbances,

such as variations (uncertainties) of the wireless channel (e.g. fading) or source (e.g. multimedia traffic) characteristics, while neglecting the coupling dynamics among nodes, due to the shared nature of the wireless spectrum. However, characterizing and learning the neighboring nodes' actions and the evolution of these actions over time is vital in order to construct an efficient and robust solution for delay-sensitive applications. Hence, we propose and analyze various interactive learning schemes for these agents to learn the network dynamics and, based on this knowledge, foresightedly adapt their cross-layer transmission decisions such that they can efficiently utilize the shared, time-varying network resources. We show that the foresighted decision making significantly improves the agents' utilities under a variety of dynamic network scenarios (e.g. multimedia streaming over WLAN, energy-efficient transmission in mobile ad hoc networks, joint route/channel selection in multi-hop cognitive radio networks) and various network topologies as compared to existing state-of-the-art solutions.

In conclusion, our research adds a *new*, “*cognitive*”, *dimension* to existing multi-hop wireless networks that enables the autonomic nodes to *dynamically forecast the expected response* to network dynamics of neighboring nodes and evaluate how specific forms of explicit and implicit signaling impact the performance of delay-sensitive applications.

Chapter 1

Introduction

Emerging multi-hop wireless networks provide a low-cost and flexible infrastructure that can be simultaneously utilized by multiple nodes for a variety of applications, including delay-sensitive applications, which form the main focus in this dissertation. These multi-hop wireless networks can be either constructed using passive nodes that follow the coordination of a central coordinator (e.g. a network planner), which directs their transmission strategies, or using autonomic nodes that can determine and adapt their own transmission strategies to maximize the network utility. Such wireless networks are referred to as autonomic wireless networks. These networks are established based on the voluntary participation of autonomic wireless nodes (also interchangeably referred to as agents in this dissertation), which interact with each other (i.e. make their own decisions) in order to maximize their own utilities. Many features make such autonomic decision making an appealing approach for driving the resource management and information exchanges for delay-sensitive applications. First, in the multi-hop wireless environment, the decisions on how to adapt the cross-layer transmission strategies at the various sources and relays need to be performed in an informationally-decentralized manner, because the tolerable delay does not allow propagating messages back and forth throughout the network to a centralized coordinator. Second, even if information were centralized, the centralized cross-layer optimizations are too complex to be solved in a timely manner. This leads to a “decomposition” of the optimization which relies on the dynamic reconfiguration of the autonomous nodes. Third, both the applications and the wireless network conditions are time-varying and hence, it is necessary that the source nodes and relay nodes self-organize to adapt to new environmental conditions.

I. DISSERTATION GOAL

This dissertation presents principles and design rules that enable autonomic nodes to proactively construct multi-hop networks for the efficient transmission of delay-sensitive applications. We study how these autonomic nodes can coordinate with the other nodes in order to self-organize themselves to transmit delay-sensitive applications over a multi-hop wireless network. Importantly, we discuss two main concepts that enable the autonomic nodes to make autonomous decisions and maximize the applications' performances:

- *Foresighted cross-layer transmission strategies.* In dynamic multi-user wireless environments, the nodes' strategies are coupled, since the transmission actions taken by the nodes impact the utility of each other. Thus, nodes need to select their optimal cross-layer strategies by anticipating the impact of their actions on both their immediate utility as well as on their long term performance. For instance, a node's aggressive transmission strategy may be rewarded in the short term by a high utility gain, but this will trigger the other nodes to respond by adapting their own transmission strategies, which will ultimately impact its long term reward. Hence, autonomic nodes need to build accurate models about the other wireless nodes' response strategies to forecast future utilities and, based on this, make foresighted decisions on which cross-layer transmission strategies they should adopt in real-time.
- *Interactive learning.* In order to build these accurate models about the other nodes, the autonomic nodes can adopt interactive learning approaches to learn the strategies of the other nodes based on local "observed information". Such information is often obtained through control message exchange mechanisms made possible by network protocols. The autonomic nodes can proactively determine what messages they would like to exchange with other nodes and, using these messages, negotiate and coordinate with other nodes the usage of available network resources. Various classes of interactive learning approaches can be adopted depending on the information

exchange mechanism, which results in different transmission overheads and complexity costs that lead to different learning efficiency. This dissertation also discusses the tradeoffs between the costs of the information exchanges, which are necessary for the distributed coordination of nodes, and the learning efficiency, by evaluating their impact on the nodes' utilities.

This dissertation, by focusing on these two core concepts, provides a systematic framework (shown in Figure 1.1) for building highly efficient multi-hop wireless networks to support delay-sensitive applications. The chapters of the dissertation present illustrative examples of how the framework can be integrated with existing protocols, standards and deployed systems.

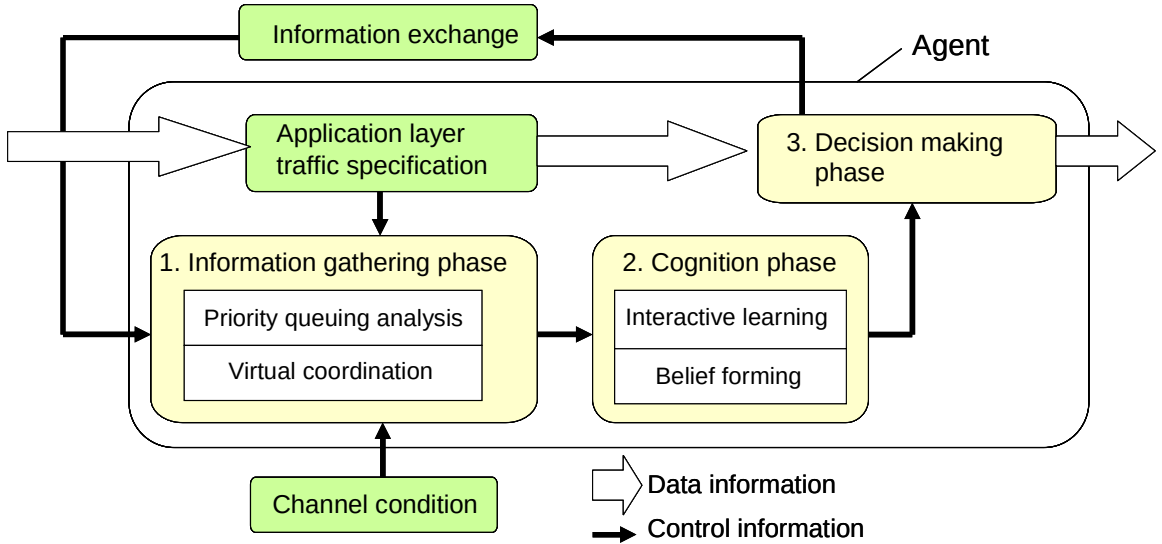


Fig. 1.1. The autonomic decision making framework for delay-sensitive applications.

II. CHALLENGES IN DYNAMIC MULTI-HOP WIRELESS NETWORKS

A first challenge comes from the **informationally-decentralized nature of multi-hop wireless networks**. Each node possesses different utilities and makes decisions based on its “observed information” (i.e. their “environment” measurements and/or exchanged control messages). In general, in a practical transmission scenario, the utility of each node

is not known by other nodes. Moreover, the nodes are not always directly aware of the transmission strategies of other nodes. Different types of observations can be made by the nodes depending on their adopted wireless protocols. Moreover, we highlight the importance of considering the cost of the induced information overheads and their impact on the nodes' utilities.

The second challenge arises due to the **delay-sensitive characteristics of the applications**. As the source characteristics are changing, the tolerable delays at the application layer and the required utility (e.g. quality or fidelity) can vary significantly. This influences the performance of the applications and, ultimately, the choice of the optimal transmission strategy adopted by the node. Moreover, the delay-sensitive characteristics of the applications also make a centralized solution impractical, since the tolerable delay does not allow propagating control information back and forth throughout the multi-hop network to a centralized decision maker. Hence, this further emphasizes the need for developing informationally-decentralized resource management solutions, where autonomic nodes coordinate their resource usage by proactively exchanging information.

Third, the wireless network is a highly **dynamic transmission environment**. The transmission channel condition is unreliable and the network topology may vary over time. To address this issue, it is important to provide distributed solutions that can timely adapt to these changes in the network.

Finally, in a multi-user setting, the utility and the decision of a node varies depending on both its experienced "environment" (e.g. application, source and channel characteristics), and the other nodes' strategies. Thus, a key challenge associated with delay-sensitive transmission in ad-hoc wireless networks is the coupling of the **wireless nodes' actions and their utility performances**, as the individual decisions of the nodes and that of their relaying peers will have a significant impact on each others' utilities. More challenges are arising when multiple delay-sensitive applications are simultaneously utilizing the same wireless networks.

To cope with these challenges, the autonomic nodes need to coordinate with each other to form a multi-hop network and optimize their cross-layer transmission strategies by taking in to account the response of the other nodes. To do so, the nodes will need to learn the other nodes responses to their strategies and correspondingly adapt their strategies in real-time. To estimate the response of the other nodes, interactive learning approaches can be deployed.

III. ORGANIZATION OF THE DISSERTATION

The subsequent chapters of the dissertation aim to address the abovementioned challenges. Figure 1.2 shows the organization of the various chapters.

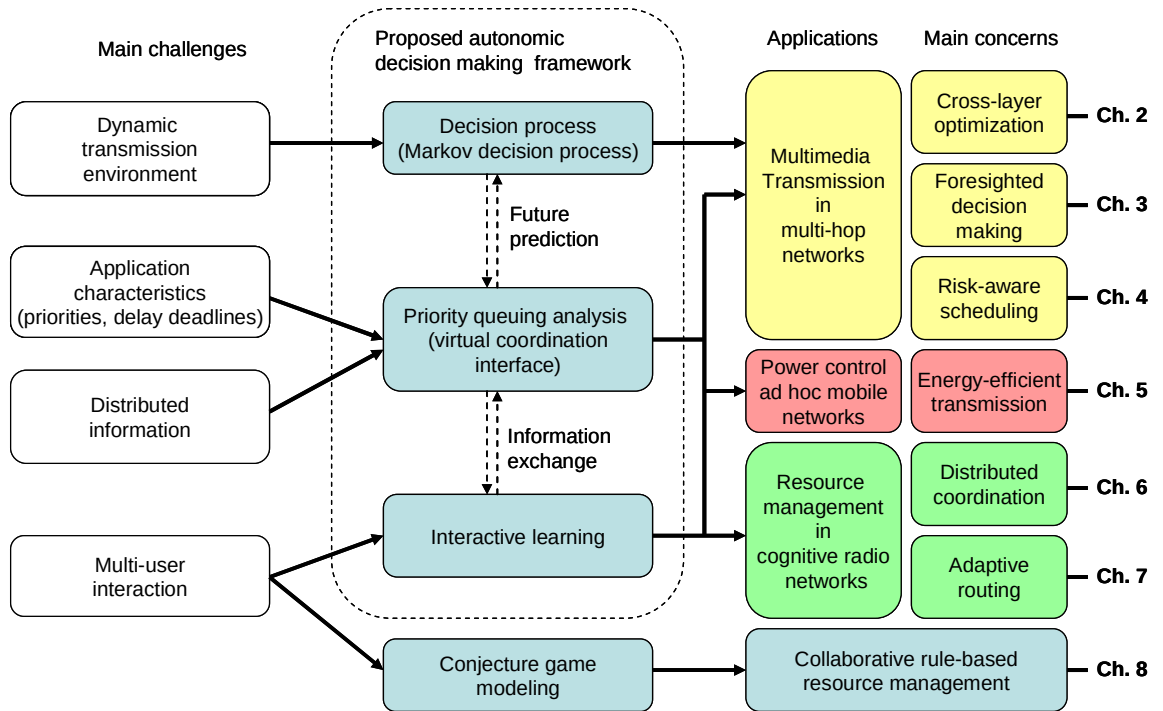


Fig. 1.2. The organization of the dissertation.

Chapter 2 discusses the *cross-layer design* of video streaming over multi-hop wireless

networks. Distributed packet-based cross-layer algorithms are presented to maximize the decoded video quality of multiple nodes engaged in simultaneous real-time streaming sessions over the same multi-hop wireless network. These algorithms explicitly consider packet-based distortion impact and delay constraints in assigning priorities to the various packets and then rely on priority queuing analysis to model the coupling impact from the other nodes and to drive the optimization of the various nodes' transmission strategies *across the protocol layers* as well as *across the multi-hop network*. Solutions enabled by the scalable coding of the video content (i.e. nodes can transmit and consume video at different quality levels) will be discussed. The cross-layer strategies we consider in this chapter include the application layer packet scheduling, the policy for choosing the routing relays in network layer, the MAC retransmission strategies, and the PHY modulation and coding schemes. The main component of the proposed solution is a low-complexity, distributed, and dynamic routing algorithm referred to as self-learning policy, which relies on prioritized queuing to select the path and time reservation for the various packets, while explicitly considering instantaneous channel conditions, queuing delays and the resulting interference. Based on the local information exchange, the cross-layer transmission strategies are optimized at each node, in a fully distributed manner.

Chapter 3 addresses the *network dynamics* in multi-hop wireless networks. The considered network dynamics include 1) time-varying traffic characteristics, 2) time-varying channel conditions, and 3) inter-node coupling. We study how wireless nodes learn the network dynamics and optimize their cross-layer transmission decisions to support delay-sensitive applications, such as surveillance, security monitoring, and mission-critical applications in military operations, etc. We consider the network delay minimization problem in a dynamic multi-hop wireless network, where multiple source nodes transmit simultaneously delay-sensitive data through relay nodes to one or multiple decision makers (destinations). Again, since there is no time to propagate control information back and forth to a central decision maker, the multi-hop network needs to be

built by autonomic nodes that can make their own transmission decisions. In such a network, the nodes can be modeled as *agents* that can make timely transmission decisions based on available local information. We formulate the autonomic decision making problem as a Markov decision process (MDP). By decomposing the centralized MDP formulation, we construct a *distributed MDP framework*, which takes into consideration the decentralized nature of the multi-hop wireless network. We prove that the distributed MDP converges to the same optimal cross-layer transmission policies of the agents as the centralized MDP. We further propose an online model-based reinforcement learning approach for agents to solve the distributed MDP at runtime, by modeling the network dynamics using priority queuing. Specifically, we allow the agents to minimize the delays of the applications by modeling the queuing delay and anticipating the network state transition probabilities. We determine the upper and the lower bounds of the delays to show the accuracy of the proposed model-based learning approach and show that they both asymptotically converge to the optimal expected delay. Moreover, we compare the proposed model-based reinforcement learning approach with the conventional model-free reinforcement learning approaches.

In **Chapter 4**, we investigate risk-aware scheduling for autonomic nodes to transmit packets of delay-sensitive applications in its queue. Various packet scheduling approaches have been proposed to address multi-user multimedia streaming over multi-hop wireless networks. However, these cross-layer transmission strategies can be efficiently optimized only if they use accurate information about the network conditions and hence, are able to timely adapt to network changes. Distributed solutions that adapt the transmission strategies based on timely information feedback need to be considered. To acquire this information feedback for cross-layer adaptation, we deploy an overlay infrastructure, which is able to relay the necessary information about the network status and incurred delays across different network “horizons” (i.e. across a different number of hops in a predetermined period of time). Based on the information feedback, we can

estimate the *risk* that packets from different priority classes will not arrive at their destination before their decoding deadline expires. In this chapter, we propose a distributed risk-aware scheduling approach that is optimized based on the local information feedback acquired from the various network horizons. We investigate the distributed cross-layer adaptation at each wireless node by considering the advantages resulting from an accurate and frequent network information feedback from larger horizons as well as the drawbacks resulting from an increased transmission overhead.

Chapter 5 studies the *interference coupling* among the delay-sensitive applications over wireless networks. We focus on a decentralized power control setting, where wireless nodes make their own transmission decisions in order to maximize their energy-efficient utilities as evaluated based on exchanged information. Specifically, two types of information exchange are discussed in this chapter, which result in two different classes of learning approaches. One is the *private information feedback* between a transmitter-receiver pair. The other is the *public information feedback* among nodes (i.e. different transmitter-receiver pairs). Due to the informationally-decentralized nature of the wireless network, a node cannot have complete information about the transmission actions of its interfering neighbors. However, the node can model implicitly or explicitly the transmission strategies (power spectrum profile) of its major interference sources based on the observed information. A node can adopt model-based learning schemes to explicitly model the other nodes' strategies if public information is available, or adopt payoff-based learning schemes to implicitly model the impact of other nodes' actions on its utility if only private information is available. Based on these models, the node creates beliefs and is able to strategically adapt its decisions to maximize its own utility. Importantly, we investigate the cost-efficiency tradeoffs resulting from the information gathered with different frequencies and from various nodes. By adjusting the information exchange, the node can adapt its interactive learning scheme to approach the utility upper bound. The energy efficiency of delay-sensitive nodes in mobile ad hoc networks can be

significantly improved by adopting the interactive learning schemes introduced in this chapter.

In **Chapter 6**, we present the dynamic channel selection in *single-hop cognitive radio networks* for transmitting delay-sensitive applications. The majority research in this area seldom considers the requirement of the application layer. In this chapter, we present the solutions especially suitable for heterogeneous multimedia applications with various rate requirements and delay deadlines. Note that in a cognitive radio networks, the wireless nodes usually possess private utility functions, application requirements, and distinct channel conditions in different frequency channels. To efficiently manage available spectrum resources in a decentralized manner, efficient information exchange coordination among nodes is necessary. The term “cognitive” in this dissertation refers to both the capability of the network nodes to achieve large spectral efficiencies by dynamically exploiting available frequency channels *as well as* their ability to learn the “environment” (the actions of interfering nodes) based on the designed information exchange. Hence, we first introduce the priority virtual queuing interface that determines the required information exchanges. With the primary nodes as the highest priority traffic, each node evaluates its expected delays based on the information. Such expected delays are important for multimedia applications due to their delay-sensitivity nature. The expected delays are evaluated using priority queuing analysis that considers the wireless environment, traffic characteristics, and build models of the other nodes’ behaviors in the same frequency channel. Next, we discuss the Dynamic Strategy Learning (DSL) algorithm that exploits the expected delay and dynamically adapts the channel selection strategies to maximize the node’s utility function.

Chapter 7 studies the dynamic resource management in *multi-hop cognitive radio networks* for transmitting delay-sensitive applications. Since the tolerable delay does not allow propagating global information back and forth throughout the multi-hop network to a centralized decision maker, the source nodes and relays need to adapt their actions

(transmission frequency channel and route selections) in a distributed manner, based on *local* network information. We propose a distributed resource management algorithm that allows network nodes to exchange information and that explicitly considers the delays and cost of exchanging the network information over the multi-hop cognitive radio networks. Note that the node competition is due to the mutual interference of neighboring nodes using the same frequency channel. Based on this, we adopt a multi-agent learning approach, adaptive fictitious play, which uses the available interference information. We also discuss the tradeoff between the cost of the required information exchange and the learning efficiency.

In **Chapter 8**, we introduce conjecture-based channel selection for delay-sensitive applications in multi-channel wireless networks. In our considered communication scenario, nodes make their channel selections in a selfish manner, by minimizing their expected delays in sending packets over the network. Since the nodes' strategies for selecting channels are coupled, it is important for a node to consider the impact of the other nodes' channel selection strategies when making their own decision. This is in contrast to the conventional multi-channel MAC protocols, which either require nodes to obey a centralized allocation determined by a network moderator or, in a distributed setting, only enable nodes to *react* to an aggregate channel measurement (e.g. contention level experienced in a certain channel) when selecting their transmission channels. Existing centralized approaches result in efficient allocations, but require intensive message exchanges among the nodes (i.e. they are not informationally efficient). Current distributed approaches do not require any message exchange, but they often result in inefficient allocations, because nodes only respond to their experienced contention in the network. As a result, these myopic distributed approaches often result in a suboptimal solution from the nodes' or the communication system's perspective. Alternatively, in this chapter we study a distributed channel selection approach, which does not require any message exchanges, and which leads to a system-wise Pareto optimal solution by

enabling nodes to predict the implications (based on their beliefs) of their channel selection on their expected future delays and thereby, foresightedly influence the resulting multi-user interaction. We model the multi-user interaction as a channel selection game and show how nodes can play an ε -consistent conjectural equilibrium by building near-accurate beliefs and competing for the remaining capacities of the channels. We study two different operation scenarios – 1) when the wireless system has only one foresighted node acting as a leader, 2) when the wireless system has multiple foresighted nodes. We analytically show that when the system has only one foresighted node, this self-interested leader can deploy a linear belief function in each channel and manipulates the equilibrium to approach the Stackelberg equilibrium. Alternatively, when the leader is altruistic, the system will converge to the system-wise Pareto optimal solution. We propose a low-complexity learning method based on linear regression for the foresighted node to learn its belief functions. When the system has multiple foresighted nodes, we show how these nodes can approach the system-wise Pareto optimal solution by collaboratively complying with prescribed rules of building beliefs. An on-line coordination procedure that enables the nodes to reach the system-wise Pareto optimal solution in a distributed manner is provided.

Finally, we conclude the dissertation in **Chapter 9**.

Chapter 2

Cross-layer Optimization for Multimedia Streaming in Multi-Hop Wireless Networks

I. INTRODUCTION

In this chapter, we focus on transmitting multiple delay-sensitive video bitstreams across the same multi-hop wireless local area network (WLAN). Such wireless infrastructures often provide dynamically varying resources with only limited support for the Quality of Service (QoS) required by real-time multimedia applications. Hence, efficient solutions for multimedia streaming must accommodate time-varying bandwidths and probabilities of error introduced by the shared nature of the wireless medium and quality of the physical connections. In the studied distributed transmission scenario, users need to proactively collaborate in sharing the available wireless resources, in order to ensure that the various multimedia applications are provided with the necessary QoS. Such collaboration is needed due to the shared nature of the wireless infrastructure, where the cross-layer transmission strategy deployed by one user impacts and is impacted by the other users.

Prior research on multi-user multimedia transmission over multi-hop wireless networks has focused on centralized, flow-based resource allocation strategies based on a pre-determined rate-requirement [WZ02][SYZ05]. These solutions are not scalable to the network size or the number of users and attempt to solve the end-to-end routing and path selection problem as a combined optimization using algorithms designed for Multi-Commodity Flow [AL94] problems. Such an optimization ensures that the end-to-end utility function (benefit) is maximized while satisfying constraints on individual link capacities. For instance, in [NMR05], a dynamic routing policy based on

queuing backpressure is proposed, which ensures that the average delay is bounded for the various users as long as the transmission rates are inside the capacity region of the network. However, the flow-based optimization does not guarantee that explicit packet-based delay constraints are met for video applications. These network layer research chapters do not consider the real-time adaptation to time-varying channel conditions, video characteristics and encoding parameters (that influence packet-based delay constraints). Importantly, they do not take into account the loss tolerance provided by video applications, which can be exploited by the wireless network to support a larger number of users. Therefore, these solutions often lead to inferior network efficiency and suboptimal resulting qualities for the video users.

Alternatively, the majority of the video-centric research does not consider the protection techniques available at lower layers of the protocol stack (MAC, PHY) and/or optimizes the video transport using purely end-to-end metrics, thereby excluding the significant gains of cross-layer design [BT05][WCZ05][DAB03]. Recent results on the practical throughput and packet loss analysis of multi-hop wireless networks have shown that the incorporation of appropriate utility functions (that take into account specific parameters of the protocol layers such as the expected retransmissions, the error rate and bandwidth of each link [DAB03], as well as expected transmission time [DPZ04]) can significantly impact the actual performance. In [AMV06], an integrated cross-layer optimization framework was proposed that considers the video quality impact. However, the solution proposed in [AMV06] considers only the single user case, where a set of paths and transmission opportunities are statically pre-allocated for each video application. This leads to a sub-optimal, non-scalable solution for the multi-user case, which ignores important problems such as inefficient routing and time allocation to avoid interference among neighboring nodes. In summary, while significant contributions have been made to enhance the separate performance of the various OSI layers, no framework exists that integrates distributed and adaptive routing and resource allocation with

cross-layer optimization for efficient multi-user multimedia streaming over multi-hop wireless networks.

In this chapter, we propose such an integrated cross-layer solution for multiple video users. Our solution relies on the users' agreement to collaborate by dynamically adapting the quality of their multimedia applications to accommodate the more important flows/packets of other users. Unlike commercial multi-user systems, where the incentive to collaborate is minimal and there are often free-riders, we investigate the proposed approach in an enterprise network setting where users exchange accurate and trustable information about their applications (e.g. packet priorities). In our setting, the importance of the packets is determined based on their contribution to the overall distortion of a particular video as well as their delay deadlines. This information is encapsulated in the header of each transmitted packet and is used by intermediate nodes to drive the cross-layer transmission strategies. Moreover, our priority queuing approach also enables path diversity gains due to the delay-optimized dynamic routing, since the packets of the same application may be transmitted over different paths between the source and destination nodes.

To increase the number of simultaneous users as well as to improve their performance given time-varying network conditions, we deploy scalable video coding schemes that enable a fine-granular adaptation to changing network conditions and a higher granularity in assigning the packet priorities. In our set-up, each user has a distinct source-destination pair. We assume a directed acyclic multi-hop overlay network [KV04] that can convey (in real-time) information about the expected delay for each priority class from a specific node to the destination. Each receiving node performs polling-based contention-free media access [IEE03] that dynamically reserves a transmission opportunity interval in a service interval (SI). The network topology and the corresponding channel condition of each link are assumed to remain unchanged within the SI. Each node maintains a queue containing video packets from various users and correspondingly determines the

transmission strategies based on the network information feedback from the neighbor nodes of the next hop. At intermediate nodes, we select the next hop based on a shortest-delay policy similar to the Bellman-Ford routing algorithm [BG87]. However, in our approach, we explicitly consider the packet deadlines and their priorities. Based on this intermediate node selection, we determine the expected delay for the packet and relay this information via the overlay network to the previous nodes.

The main contributions of this chapter are listed below.

1. Packet-based vs. flow-based/layer-based solutions

We introduce a novel video streaming approach based on priority queuing that enables us to optimize the cross-layer transmission strategies per packet. The proposed cross-layer adaptation differs from existing solutions for multimedia transmission over multi-hop networks, where the path (or limited multiple paths) is predetermined for the entire bitstream or layer [AMV06]. Moreover, the MAC retransmission and PHY link adaptation are often not considered for these flow-based/layer-based solutions [SYZ05]. Our approach is based on a multi-path routing algorithm that determines the next relay per packet. The proposed priority and delay-driven approach allows us to avoid global optimizations based on pre-determined rate requirements or path selections, which are not adaptive to network changes, the number of users or streamed video content characteristics.

2. Distributed solution based on dynamic routing vs. conventional centralized solutions

Existing research [SYZ05][WCZ05] poses the problem of multi-user resource allocation and cross-layer adaptation over ad-hoc wireless networks as a static, centralized optimization that maximizes the utility (e.g. video quality) of the various users given pre-determined channel (capacity) constraints [TG03] and video rate requirements. These solutions have several limitations. First, the video bitstreams are changing over time in terms of required rates, priorities and delays. Hence, it is difficult

to timely allocate the necessary bandwidths across the wireless network infrastructure to match these time-varying application requirements. Second, the delay constraints of the various packets are not explicitly considered in centralized solutions, as this information cannot be relayed to a central resource manager in a timely manner. Third, the complexity of the centralized approach grows exponentially with the size of the network and number of video flows. Finally, the channel characteristics of the entire network (the capacity region of the network) need to be known for this centralized, oracle-based optimization. This is not practical as channel conditions are time-varying, and having accurate information about the status of all the network links is not realistic.

Alternatively, in our solution, we optimize the cross-layer strategies (dynamic routing, MAC retransmission limit, and PHY modulation and coding scheme) per packet at the various intermediate nodes, in a distributed manner, which allows us to efficiently adapt to changes in the video bitstream, channel characteristics, and network resource. This approach is well suited for the informationally decentralized nature of the investigated multi-user video transmission problem. We also discuss the required information/parameters exchange among networks/layers for implementing such a distributed solution.

3. Priority queuing analysis with interference consideration

Our solution aims at minimizing the packet loss rate of the packets in higher priority video classes based on the proposed priority queuing analysis. The analysis is performed for network environments with and without transmission interference consideration. To cope with the interference problems that exist in multi-hop networks due to the broadcast nature of the wireless medium, we adopt a polling-based, contention-free MAC that allocates transmission opportunities at each node to the various classes/packets based on their priorities [IEE03]. To analyze the expected waiting time for the various packets in the presence of interference, we apply a novel virtual queuing method based on the "service-on-vacation" queuing model.

4. Bottleneck identification

Using our priority queuing analysis, we can estimate the expected packet loss at the transmitter side. This information can be used by the application layer to decide how many quality layers are transmitted or to adapt its encoding parameters (in the case of real-time encoding) to improve its video quality performance given the current number of users, priorities of the competing streams and network conditions, but also, importantly, to alleviate the network congestion. Note that our analysis provides this network bottleneck identification for each priority class, which is used in our solution to simplify the routing decision strategies. Furthermore, this information can be exploited to improve the network infrastructure such that it can support various multimedia application scenarios under different levels of network congestion.

The rest of this chapter is organized as follows. Section II introduces the multi-user video streaming specification (video priority classes, network specification, cross-layer parameters etc.) and subsequently gives the cross-layer optimization problem formulation and highlights the need for a distributed per-packet solution. In Section III, we present our distributed solution which involves dynamically selecting relays that minimize the end-to-end packet loss probability of the higher priority video packets of the various users. In Section IV, we present the queuing delay analysis required in the proposed solution to determine the expected delay at each node. Based on the expected delay, a relay will be dynamically selected. In this section, we do not consider the effect of interference, as is the case in wireless networks where the nodes can simultaneously transmit and receive in orthogonal channels. Subsequently, in Section V, our analysis is extended to a wireless network environment where the transmission is performed in the same channel, and thus the interference needs to be considered. In Section VI, we show that the proposed distributed routing algorithm converges to a steady-state under certain assumptions. Finally, Section VII presents our simulation results, and Section VIII concludes the chapter.

II. MULTI-USER VIDEO STREAMING SPECIFICATION

A. Video priority classes

We assume that there are V video users (with distinct source-destination pairs) sharing the same multi-hop wireless infrastructure. In [VAH06], it has been shown that partitioning a scalable embedded video flow (stream) into several priority classes (quality-layers) can improve the number of simultaneously admitted stations in a congested 802.11a/e WLAN infrastructure, as well as the overall received quality. Similarly, in this chapter, we categorize the video units (video packets, video subbands, video frames) of the video bitstream into several priority classes. We adopt an embedded 3D wavelet codec [AMB04] and construct video classes by truncating the embedded bitstream [VAH06]. We assume that the packets within each class have the same delay deadline (see e.g. [VT07][VAH06] for more detail on how the delay is computed per class). For a video sequence v , we assume there are N_v classes, and these video classes are characterized by:

- λ_v , a vector of the quality impact of the various video classes. We prioritize the video classes based on this parameter. The video classes are organized in an embedded bitstream in terms of their video quality impact, i.e. $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{N_v}$.
- $\mathbf{R}_v$, a vector of the rate requirements of the various video classes.
- $\mathbf{d}_v$, a vector of the delay deadlines of the various video classes. Due to the hierarchical temporal structure deployed in 3D wavelet video coders (see [VT07][WV06]), the lower priority packets also have a less stringent delay requirement, i.e. $d_1 \leq d_2 \leq \dots \leq d_{N_v}$. This is the reason why we can prioritize the video bitstream only in terms of the quality impact. However, if the used video coder did not exhibit this property, we need to deploy alternative prioritization techniques $\lambda_k^{video}(\lambda_k, d_k)$ that jointly consider the distortion impact and delay constraints (see the more sophisticated methods discussed in e.g. [CM06][JF07]).
- $\mathbf{L}_v$, a vector of the average packet lengths of the various video classes.

- $\mathbf{P}_v^{succ}$, a vector containing the probabilities of successfully receiving the packets in the various video classes at the destination.

We denote the video classes using f_k , which can be characterized by the elements $\lambda_k, R_k, d_k, L_k, P_k^{succ}$ in the above mentioned vectors.

At the client side, the expected received video quality for video v can be modeled using any desirable video rate-distortion model:

$$Q_v^{rec} = F_v(\boldsymbol{\lambda}_v, \mathbf{R}_v, \mathbf{d}_v, \mathbf{L}_v, \mathbf{P}_v^{succ}), \quad (1)$$

represented by the function $F_v(\cdot)$ which can be computed as in e.g. [VT07][OR98][WV06], based on the successfully received video classes.

We assume that the client implements a simple error concealment scheme, where the lower priority packets are discarded whenever the higher priority packets are lost [VT07]. This is because the quality improvement (gain) obtained from decoding the lower priority packets is very limited (in such embedded scalable video coders) whenever the higher priority packets are not received. For example, drift errors can be observed when decoding the lower priority packets without the higher priority packets [WV06]. Hence, we can write:

$$P_k^{succ} = \begin{cases} 0 & \text{if } P_{k'}^{succ} \neq 1 \text{ and } f_{k'} \prec f_k \\ (1 - P_k) = E[I(D_k \leq d_k)], & \text{otherwise} \end{cases}, \quad (2)$$

where we use the notation in [CM06] - $f_{k'} \prec f_k$ to indicate that the class f_k depends on $f_{k'}$. Specifically, if f_k and $f_{k'}$ are classes of the same video stream, $f_{k'} \prec f_k$ means $k' < k$ due to the descending priority ($\lambda_{k'} > \lambda_k$). This error concealment policy facilitates our priority queuing solution, which will be discussed in Section III. P_k represents the end-to-end packet loss probability for the packets of class f_k . D_k represents the experienced end-to-end delay for the packets of class f_k . $I(\cdot)$ is an indicator function. Note that the end-to-end probability P_k^{succ} depends on the network resource, competing users' priorities as well as the deployed cross-layer transmission strategies vector, which will be discussed in more detail in Section III.C.

B. Network specification

Let $\mathfrak{N} = [\Gamma, \mathbf{C}]$ represent the network specification, where Γ represents the given network graph, and $\mathbf{C}$ represents the interference matrix. The network graph Γ defines the network nodes (including the source nodes, destination nodes and relays) and the available transmission links in the multi-hop wireless network. The interference matrix $\mathbf{C}$ defines whether or not two different links can transmit simultaneously, and will be discussed in Section V in more detail. Besides the V source-destination pairs, we assume the network graph Γ consists of H hops with M_h intermediate nodes (relays) at each h -th hop ($0 \leq h \leq H - 1$). The number of source and destination nodes are the same, i.e. $M_0 = M_H = V$, and each node will be tagged with a distinct number m_h ($1 \leq m_h \leq M_h$) as shown in Figure 2.1. The other parameters in the figure will be defined in the following subsection.

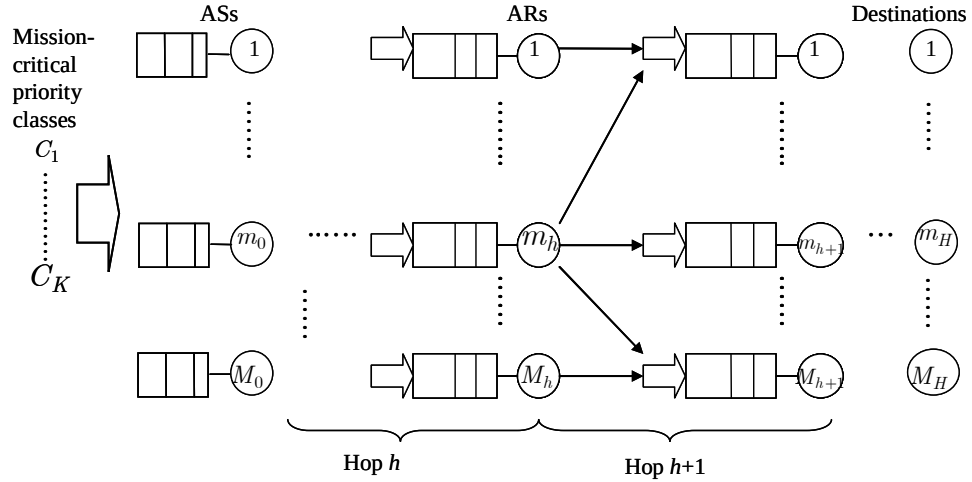


Fig. 2.1 Illustrative example of the considered directed acyclic multi-hop networks.

C. Cross-layer joint transmission strategy vector

Next, we define the transmission strategies of video units (video packets) at various layers across the network. Let us define the cross-layer joint strategies vector $\mathbf{STR} = \{STR_{h,m_h}(\vartheta) \mid \vartheta=1 \dots N^{tot}, \quad 1 \leq m_h \leq M_h, \text{ and } 0 \leq h \leq H - 1\}$ as a vector of

transmission strategies that can be deployed for packets present in the queue at the various nodes. N^{tot} is the total number of packets. $STR_{h,m_h}(\vartheta \in f_k)$ $= [\pi_{h,m_h}, \beta_{k,h+1,m_{h+1}}, \gamma_{k,m_h,m_{h+1}}^{MAX}(\vartheta), \theta_{k,m_h,m_{h+1}}(\vartheta)]$ represents the cross-layer transmission strategies for a packet ϑ at the intermediate node m_h at the h -th hop. Next, we describe the cross-layer transmission strategies.

- **Application layer**

The packet headers are extracted at the various relays, to determine the packet priority, delay deadlines and packet lengths required for our cross-layer solution. Based on this information, the packet scheduling π_{h,m_h} should transmit a packet in the highest priority class f_k (i.e. the class with the highest quality impact) that is present in the queue at the node m_h . Thus, the packets with the largest quality contribution are scheduled first for transmission. The packets for which the delay deadline has expired are discarded from the queue. In other words, the higher priority packets are transmitted to the level that the network can accommodate, while the lower priority packets are queued and will be dropped if their delays exceed the delay deadline.

- **Network layer**

We define β_{k,h,m_h} as the percentage of packets in priority class f_k (fraction of time) to select the node m_h as its relay at the h -th hop. We refer to this term as the *relay selecting parameter*. By assigning relays according to the relay selecting parameter, multiple paths can be chosen for the packets in class f_k , i.e. $0 \leq \beta_{k,h,m_h} \leq 1$. The relay selecting parameters provide a routing description across the network with multi-path capability. Whenever an intermediate node m_h is not reachable for class f_k , then $\beta_{k,h,m_h} = 0$. Since the total number of intermediate nodes in the h -th hop is M_h , we have $\sum_{m_h=1}^{M_h} \beta_{k,h,m_h} = 1$. Note that since each class f_k has a pre-determined destination (i.e. $m_H = v$), the relay selecting parameter at the last hop (β_{k,H,m_H}) is equal to ‘1’, if m_H is the destination of the class, and ‘0’, otherwise. Instead of selecting a fixed relay for all packets of class f_k , these video packets select the intermediate nodes m_{h+1} as their

relay according to the corresponding $\beta_{k,h+1,m_{h+1}}$. At the intermediate nodes in the h -th hop, β_{k,h,m_h} are the incoming relay selecting parameters, and $\beta_{k,h+1,m_{h+1}}$ are the outgoing relay selecting parameters. The proposed dynamic routing solution is based on priority queuing while considering the lower layer goodput (effective transmission rate after factoring in packet losses) of all the possible link choices. We will discuss the relay selecting mechanism in Section III.B in more detail. Note that different paths can be selected for packets in the same class.

- **MAC layer**

At the MAC layer, we assume the network deploys a protocol similar to that of IEEE 802.11a/e [IEE03], which enables packet-based retransmission and polling-based time allocation. Let $\gamma_{k,m_h,m_{h+1}}^{MAX}(\vartheta)$ represent the maximum number of retransmissions for packet ϑ of priority class f_k over the link (m_h, m_{h+1}) at the $h+1$ -th hop. The optimal retransmission limit is adapted based on the delay deadline d_k of the packet, which will be discussed in more detail in Section III.C.

- **PHY layer**

Let $\theta_{k,m_h,m_{h+1}}(\vartheta)$ denote the modulation and coding scheme used for packet ϑ of class f_k for transmission over the link (m_h, m_{h+1}) at the $h+1$ -th hop. (This is affected by the packet length). Let $T_{k,m_h,m_{h+1}}(\theta_{k,m_h,m_{h+1}})$ and $p_{k,m_h,m_{h+1}}(\theta_{k,m_h,m_{h+1}})$ represent the corresponding transmission rate and packet error rate. Recall that the goodput over the link is defined as $T_{k,m_h,m_{h+1}}^{goodput}(\theta_{k,m_h,m_{h+1}}) = T_{k,m_h,m_{h+1}}(\theta_{k,m_h,m_{h+1}}) \cdot (1 - p_{k,m_h,m_{h+1}}(\theta_{k,m_h,m_{h+1}}))$.

In Section III.B, we will discuss the various cross-layer strategies in more detail.

D. Problem formulations

- **Centralized problem formulation**

The conventional formulation of the multi-user wireless video transmission problem can be regarded as a cross-layer optimization that maximizes the overall video quality¹:

¹ A Max-Min fairness criterion can also be applied to address the fairness issue, which will affect the prioritization λ_k values accordingly.

$$\mathbf{STR}^{opt} = \arg \max_{\mathbf{STR}} \sum_{v=1}^V Q_v^{rec}(\boldsymbol{\lambda}_v, \mathbf{R}_v, \mathbf{d}_v, \mathbf{L}_v, \mathbf{P}_v^{succ}(\mathbf{STR}, \mathfrak{R})), \quad (3)$$

with the constraint that all successfully received packets must have their end-to-end delay D_k smaller than their corresponding delay deadline d_k (i.e. for every ϑ , $\vartheta \in f_k$, $D_k(\vartheta) \leq d_k$).

Due to the informationally decentralized nature of the multi-users video transmission over multi-hop networks, a centralized solution for this optimization problem is not practical. For instance, the optimal solution depends on the delay incurred by the various packets across the hops, which cannot be timely relayed to a central controller. Instead, we propose a distributed packet-based solution to optimize the quality of the various users sharing the same multi-hop wireless infrastructure.

- **Proposed distributed problem formulation**

Based on the proposed prioritized video classes and deployed error concealment strategy, a distributed cross-layer optimization can be formulated as a per-hop minimization of the end-to-end packet loss rate at the node m_h of the h -th hop:

$$\begin{aligned} STR_{h,m_h}^{opt}(\vartheta^* \in f_k) &= \arg \max_{STR} R_k \cdot P_k^{succ}(STR_{h,m_h}(\vartheta^*), \mathfrak{R}) \\ &= \arg \min_{STR} P_k(STR_{h,m_h}(\vartheta^*), \mathfrak{R}) \end{aligned}, \quad (4)$$

where we minimize P_k for the selected packet $\vartheta^* \in f_k$ in the queue of the node m_h according to the scheduling π_{h,m_h} , with the delay constraint $D_k(\vartheta^*) \leq d_k$.

Note that in a directed acyclic multi-hop network shown in Figure 2.1, the end-to-end packet loss probability P_k can be decomposed based on the hop-by-hop packet loss probability $P_{k,h}$:

$$P_k = 1 - \left(\prod_{h=0}^{H-1} (1 - P_{k,h}) \right), \quad (5)$$

where $P_{k,h}$ represents the packet loss probability incurred due to delay deadline expiration during a specific hop h , given that the packet was not lost in the previous hop. In the next section, we present our distributed cross-layer solution of equation (4) based on the dynamic routing over such multi-stage overlay structure.

III. A DISTRIBUTED PACKET-BASED SOLUTION BASED ON PRIORITY QUEUING

In this section, we present our distributed packet-based solution. We show that the packet priorities (determined by λ_k for class f_k) and their delay constraints (d_k) drive the selection of optimal transmission strategies at the different layers in a distributed manner at each hop.

A. Required information feedback among network nodes for the distributed solution

The proposed distributed approach not only simplifies the proposed cross-layer solution but also makes it adaptive to the varying network characteristics, as it does not require feedback about the entire network status. At each node, the transmission strategies for the prioritized video packets are determined based on the information feedback from the neighboring nodes. In order to implement the mentioned distributed solution for multimedia transmission based on priority queuing, the following two types of information feedback to a node m_h are provided:

- $E[Delay_{k,m_{h+1}}]$: the expected delay from nodes m_{h+1} to the destination node of the packets of class f_k (this information can be relayed by the overlay infrastructure and is required for the dynamic routing solution, which will be discussed in Section III.B).
- $SINR$: the Signal-to-Interference-Noise-Ratio (SINR) from the nodes m_{h+1} in the next hop that are able to establish a link with node m_h according to the network graph Γ . This information can easily be extracted from existing 802.11 WLAN standards [IEE03].

We provide a block diagram in Figure 2.2 that indicates the parameters/information that need to be exchanged across layers/various nodes in the proposed cross-layer transmission solution.

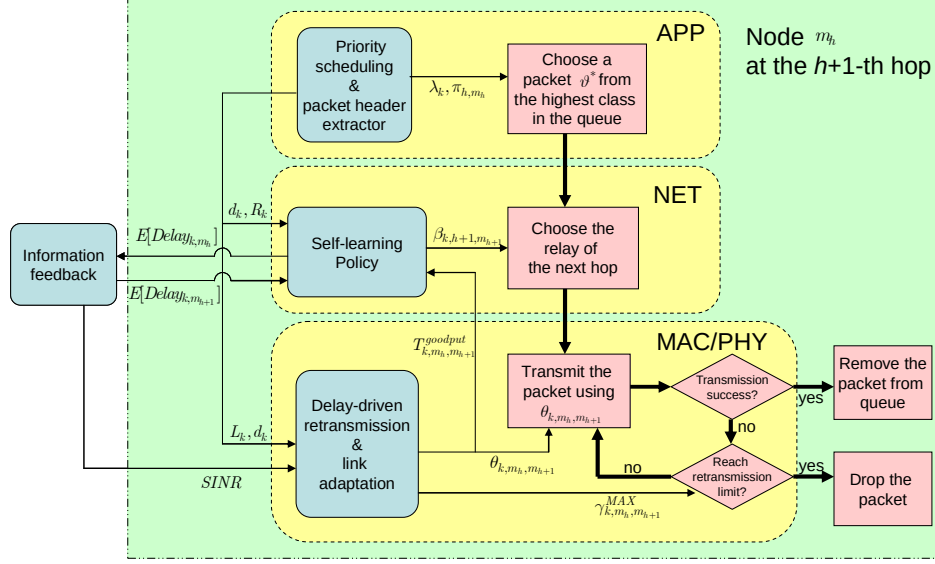


Fig. 2.2. Integrated block diagram of the proposed distributed per-packet algorithm.

B. Self-learning policy for dynamic routing

In this section, we provide our dynamic routing solution that minimizes the end-to-end packet loss probability P_k (see equation (4)). By definition $P_k = E[I(D_k > d_k)]$ and thus, minimizing P_k is equivalent to minimizing the expected end-to-end delay $E[D_k]$, given a fixed delay deadline d_k for the packets of class f_k .

To minimize the end-to-end delay over the multi-hop overlay structure shown in Figure 2.1, we propose a dynamic routing policy to determine the relay selecting parameters. Recall that each node m_h maintains and feeds back to the previous hop the expected delay from itself to the destination $E[Delay_{k,m_h}]$ for each class f_k . $E[Delay_{k,m_h}]$ becomes the cost that will be minimized at each stage, and will be updated at each node using the information feedback from the next hop. Note that $E[Delay_{k,m_h}]$ equals $E[D_k]$, if the node m_h is the source node of the class f_k packets. Note that $E[Delay_{k,m_h}]$ becomes $E[D_k]$ if m_h is the source node of the class f_k video traffic. Specifically, the expectation of delay to the destination of each class can be determined at node m_h as [Ber95]:

$$E[Delay_{k,m_h}] = \min_{\beta_{k,h+1,m_{h+1}}} \left(E[W_{k,m_h}(\beta_{k,h+1,m_{h+1}}, T_{k,m_h,m_{h+1}}^{goodput})] + \sum_{m_{h+1}=1}^{M_{h+1}} \beta_{k,h+1,m_{h+1}} E[Delay_{k,m_{h+1}}] \right), \quad (6)$$

where $E[Delay_{k,m_{h+1}}]$ is given by the information feedback obtained from the nodes of the next hop, and the relay selecting parameter $\beta_{k,h+1,m_{h+1}}$ is chosen such that $E[Delay_{k,m_h}]$ is minimized. $E[W_{k,m_h}]$ is the average queuing delay at the current relay queue, which can be obtained using the priority queuing analysis introduced in Section IV. In a congested network, equation (6) is dominated by the second term (the accumulated queuing delay in the rest of the network). Thus, we can simplify this equation as:

$$E[Delay_{k,m_h}] = E[W_{k,m_h}] + \min_{\beta_{k,h+1,m_{h+1}}} \left(\sum_{m_{h+1}=1}^{M_{h+1}} \beta_{k,h+1,m_{h+1}} E[Delay_{k,m_{h+1}}] \right) \quad (7)$$

To determine the relay selecting parameter $\beta_{k,h+1,m_{h+1}}$, we apply the following soft minimum (probabilistic) policy to enable transmission across multiple paths:

$$\beta_{k,h+1,m_{h+1}} = \frac{Coeff_k}{1 + \kappa E[Delay_{k,m_{h+1}}]^\varphi}. \quad (8)$$

$Coeff_k$ are normalized coefficients to make sure that the summation of the percentages (fraction) equals to one:

$$Coeff_k = \left[\sum_{m_{h+1} \in \mathbf{M}_{k,h+1}} \frac{1}{1 + \kappa E[Delay_{k,m_{h+1}}]^\varphi} \right]^{-1}, \quad (9)$$

where κ and φ are constants. equation (8) is inspired from the balking arrival probability in queuing theory [Kle75]. The value of κ is set depending on the arrival rate according to [Kle75]. The term φ weighs the average delay $E[Delay_{k,m_{h+1}}]$ such that the routing policy favors paths leading to significantly lower delays to the destination. $\mathbf{M}_{k,h+1}$ represents a set of nodes m_{h+1} in the $h+1$ -th hop that feedback the information $E[Delay_{k,m_{h+1}}]$. We set $\beta_{k,h+1,m_{h+1}} = 0$ for the nodes whose information feedback is not received, indicating that node m_{h+1} is not connected to node m_h using the overlay infrastructure [KV04]. We refer to this relay selecting policy as the *self-learning policy*, since the decision of $\beta_{k,h+1,m_{h+1}}$ will influence the future information feedback. The complete algorithm of the proposed self-learning policy including the information feedback is given in the Appendix of this chapter. The self-learning policy

will dynamically adapt the relay selection to minimize the delay through the network. Finally, the next relay m_{h+1} can be determined for the packet ϑ^* at the node m_h according to the percentage (time fraction) $\beta_{k,h+1,m_{h+1}}$.

This method is inspired by the Bellman-Ford shortest path (delay) routing algorithm [BG87] that minimizes the end-to-end delay across the network. Our routing algorithm reduces to the well-known Bellman-Ford algorithm when $\beta_{k,h+1,m_{h+1}} = 1$ to the node m_{h+1} that feedbacks the smallest $E[Delay_{k,m_{h+1}}]$ (which can be implemented using a large φ). Note that our algorithm is prioritized and the delay of class f_k will be influenced by equal or higher priority traffic, which will be discussed in more details in Section IV.

C. Delay-driven policy for MAC/PHY

If a node m_{h+1} is selected with probability $\beta_{k,h+1,m_{h+1}}$ for the selected packet ϑ^* at each intermediate node m_h , we can determine the corresponding transmission rate $T_{k,m_h,m_{h+1}}$ and the packet error rate $p_{k,m_h,m_{h+1}}$ for the link by selecting $\theta_{k,m_h,m_{h+1}}$ based on the *link adaptation* scheme presented in [QCS02]. To describe the channel conditions, we assume as in [Kri02] that each wireless link is a memory-less packet erasure channel. The link packet error rate for a fixed packet of length L_k bits is $p_{k,m_h,m_{h+1}}(\theta_{k,m_h,m_{h+1}}, L_k) = 1 - (1 - BER(\theta_{k,m_h,m_{h+1}}))^{L_k}$, where $BER(\theta_{k,m_h,m_{h+1}})$ is the bit error rate when the modulation scheme $\theta_{k,m_h,m_{h+1}}$ is selected. Recall that the packet error rate and the effective transmission rate (goodput) can be approximated using the sigmoid function as in [Kri02]:

$$p_{k,m_h,m_{h+1}}(\theta_{k,m_h,m_{h+1}}, L_k) = \frac{1}{1 + e^{\zeta(SINR - \delta)}}, \quad (10)$$

$$T_{k,m_h,m_{h+1}}^{goodput} = (1 - p_{k,m_h,m_{h+1}}(\theta_{k,m_h,m_{h+1}}, L_k))T_{k,m_h,m_{h+1}}(\theta_{k,m_h,m_{h+1}}) = \frac{T_{k,m_h,m_{h+1}}(\theta_{k,m_h,m_{h+1}})}{1 + e^{-\zeta(SINR - \delta)}}, \quad (11)$$

where $SINR$ is the Signal-to-Interference-Noise-Ratio, and ζ and δ are constants corresponding to the modulation and coding schemes for a given packet length. This method maximizes the goodput given the average packet length L_k of the specific class

over a selected link (m_h, m_{h+1}) based on the SINR feedback.

For a fixed $T_{k,m_h,m_{h+1}}^{goodput}$, we choose the retransmission limit $\gamma_{k,m_h,m_{h+1}}^{MAX}$ for the selected packet ϑ^* in the priority class f_k such that the delay constraint is satisfied. Specifically, let $delay_{h,m_h}^{curr}(\vartheta^*)$ represent the current measured delay incurred by the selected packet from the source to a current node m_h . The maximum retransmission limit for the packet of class f_k over the link from m_h to m_{h+1} is determined based on the delay deadline d_k (where $\lfloor \cdot \rfloor$ is the floor operation) [VAH06]:

$$\gamma_{k,m_h,m_{h+1}}^{MAX}(\vartheta^*) = \left\lfloor \frac{T_{k,m_h,m_{h+1}}^{goodput} (d_k - delay_{h,m_h}^{curr}(\vartheta^*))}{L_k} \right\rfloor - 1. \quad (12)$$

D. Complexity analysis in terms of route selection

In this section, we compare the complexity of our proposed distributed solution with a centralized approach.

- **Complexity of a conventional centralized approach (exhaustive search)**

Assume that we have a total of $K = \sum_{v=1}^V N_v$ classes across the users in an H -hop network. Let us assume the maximum number of the intermediate nodes that can be selected as a relay for a class f_k packet at the h -th hop is $\mathcal{C}_{k,h}$. The maximum number of possible end-to-end paths is then $\prod_{h=1}^H \mathcal{C}_{k,h}$. Thus, the total complexity (in terms of the number of path combinations) of a centralized exhaustive search can be up to $\prod_{k=1}^K \prod_{h=1}^H \mathcal{C}_{k,h}$. Due to the informationally decentralized nature of the wireless multi-hop network, the control overhead of the centralized approach can induce a significant amount of delay (inefficient when the number of hops is large) for doing the optimization. Hence, the distributed approach is proposed and its complexity is also investigated.

- **Complexity of the proposed distributed relay selecting algorithm**

In our distributed approach, for a packet (of class f_k) at the node m_h at the h -th hop, the complexity is $\mathcal{C}_{k,h}$ (i.e. $\mathcal{C}_{k,h}$ is the number of relays that can be selected). Thus, the complexity for the packet over the H hops is $\sum_{h=1}^H \mathcal{C}_{k,h}$. Then, the total complexity by

considering all the different classes equals $\sum_{k=1}^K \sum_{h=1}^H \mathcal{C}_{k,h}$.

Next, we investigate the delay analysis for the proposed distributed approach based on priority queuing.

IV. MULTI-HOP PRIORITY QUEUING ANALYSIS FOR MULTIMEDIA TRANSMISSION

In this section, we present the analysis of the expected queuing delay $E[W_{k,m_h}]$ (that forms $E[Delay_{k,m_h}]$) and packet loss probabilities P_{k,h,m_h} using queuing theory. Based on these values, a relay will be dynamically selected. In this section, we do not consider the effect of interference. In the next section, we extend our analysis to a network environment where the interference is considered. Before introducing the queuing model, several assumptions for the priority queuing analysis are made in Section IV.A. Then in Section IV.B, we determine the end-to-end packet loss probability P_k by considering a simple 2-hop network structure (with only one set of intermediate nodes), which we refer to as the “elementary structure”. We further extend this result by cascading the elementary structure to create a general H -hop network (with $H - 1$ sets of intermediate nodes) in Section IV.C.

A. Assumptions for priority queuing analysis

The priority queuing analysis is based on the following assumptions:

1. We assume that the arrival traffic at each intermediate node is from various video sources and is assumed to be a Poisson process. This approximation is reasonable if the number of intermediate nodes is large enough and the selection of paths is relatively balanced. We model the queues in the intermediate nodes as preemptive-repeat priority M/G/1 queues [Kle75]. For our analysis, we do not apply the non-preemptive model because when a packet with higher priority arrives at the queue, it will interrupt future transmissions (i.e. the retransmission of the same packet when this is lost or the transmission of a lower priority packet). The preempted packet will be retransmitted later.

2. We assume the transmission rate and the packet error rate for each link are fixed in a SI, as these are determined, as discussed in Section III, by selecting the appropriate modulation and coding scheme using the link adaptation mechanism. As an example, let us consider a link from node m_h to node m_{h+1} . The selected $\theta_{k,m_h,m_{h+1}}$ determines the physical transmission rate $T_{k,m_h,m_{h+1}}$ (equation (11)) and packet error rate $p_{k,m_h,m_{h+1}}$ (equation (10)) for class f_k over this link. Each packet will be retransmitted until it is either successfully received or discarded because its delay deadline d_k was exceeded. In summary, assuming the packet length of a class f_k is fixed to be L_k , with a header length L_{Header} , we can formulate the service time for a packet as a geometric distribution from these assumptions. If $X_{k,m_h,m_{h+1}}$ is the service time, then the probability of there being exactly i transmissions (including retransmissions) will be:

$$\text{Prob}\left(X_{k,m_h,m_{h+1}} = i \times \left(\frac{L_k + L_{Header}}{T_{k,m_h,m_{h+1}}} + Time_o\right)\right) = \begin{aligned} & (1 - p_{k,m_h,m_{h+1}}) p_{k,m_h,m_{h+1}}^{i-1}, \text{ for } i \leq \gamma_{k,m_h,m_{h+1}}^{MAX} + 1 \\ & 0, \text{ for } i > \gamma_{k,m_h,m_{h+1}}^{MAX} + 1 \end{aligned} \quad (13)$$

where $Time_o$ denotes the time overhead including the time of waiting for the acknowledgement, polling delay, and the expected background traffic in the contention-based period, etc [IEE03].

3. We assume that the queue waiting time dominates the overall delay (i.e. the transmission delay across the various network hops is relatively small).

Figure 2.3 illustrates the deployed priority queuing at each intermediate node. Given the application layer video priorities and class characteristics, the relay selecting parameters of the network layer, the retransmission strategy at the MAC layer, and the modulation and coding scheme at the PHY layer, we can determine the average input rate and the service time for the packets in a certain class, thereby obtaining a steady state waiting time distribution for all video priority classes. In the next subsection, we analyze the video quality problem using priority queuing analysis for our elementary structure with only one set of intermediate nodes (a 2-hop structure).

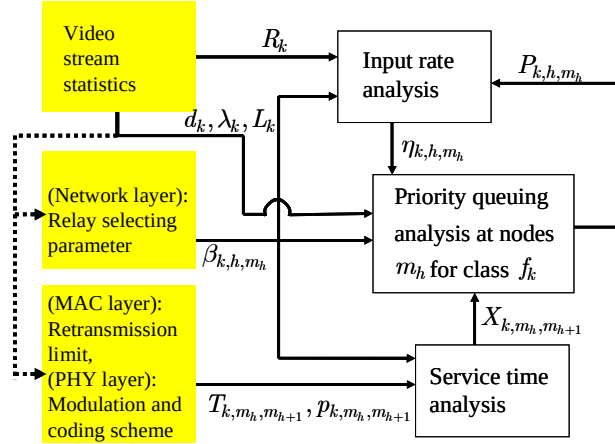


Fig. 2.3. Priority queuing analysis system map.

B. Priority queuing analysis for an elementary structure

We first analyze the priority queuing model for an elementary structure. The elementary structure is an overlay 2-hop network with V video streams and one set of M intermediate nodes (relays) between sources and destinations, as illustrated in Figure 2.4. A packet of class f_k will be routed from its source through an intermediate node m with percentage $\beta_{k,m}$ toward its destination. Each intermediate node contains a queue that schedules the waiting packets based on their header information (quality impact parameter λ_k and delay deadline d_k).

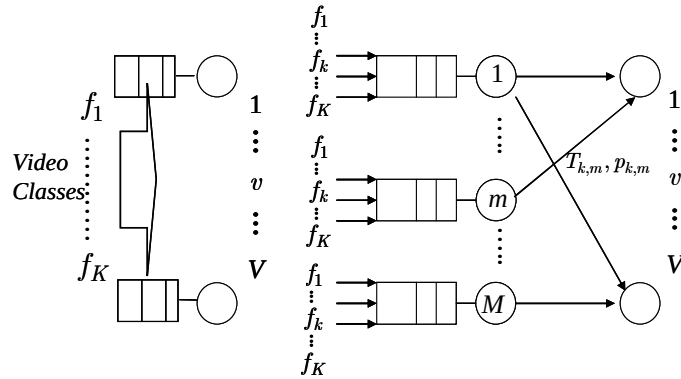


Fig. 2.4. The elementary structure.

From the geometric distribution assumption above, the first and second moment of the service time at queue m are (using the approximation $p_{k,m}^{\gamma_{k,m}^{MAX}+1} \ll 1$):

$$E[X_{k,m}] = \frac{\hat{L}_{k,m} (1 - p_{k,m}^{\gamma_{k,m}^{MAX}+1})}{T_{k,m} (1 - p_{k,m})} \approx \frac{\hat{L}_{k,m}}{T_{k,m} (1 - p_{k,m})} = \frac{\hat{L}_{k,m}}{T_{k,m}^{goodput}}, \quad (14)$$

$$E[X_{k,m}^2] \approx \frac{\hat{L}_{k,m}^2 (1 + p_{k,m})}{T_{k,m}^2 (1 - p_{k,m})^2}. \quad (15)$$

For a class f_k , which is relayed through the intermediate node m , let $\hat{L}_{k,m}$ be the effective packet “length”, which includes both the video packet length L_k and the time overhead $Time_{o,m}$ (as in equation (13)). $T_{k,m}$ and $p_{k,m}$ are the transmission rate and packet error rate for the packets of class f_k that are transmitted through the intermediate node m to the destination. Note that the modulation and coding strategy changes depending on the chosen link status, and this will consequently impact² $T_{k,m}$ and $p_{k,m}$ (see equation (10), equation (11)).

Let $\eta_{k,m}$ be the average arrival rate of the Poisson input traffic of queue m for class f_k . Given the relay selecting parameters $\beta_{k,m}$, we have:

$$\eta_{k,m} = \beta_{k,m} R_k (1 - P_{k,0}), \quad (16)$$

where $P_{k,0} = P(W_{k,0} > d_k)$ is the packet loss probability at the source queue due to packet expiration and can be calculated from the queue waiting time $W_{k,0}$ tail distribution for each class.

Let $E[W_{k,m}]$ be the average waiting time of class f_k that goes through node m . For a preemptive-priority M/G/1 queue, the priority queuing analysis gives the following result [BG87]:

$$E[W_{k,m}] = \frac{\sum_{i=1}^k \eta_{i,m} E[X_{i,m}^2]}{2 \left(1 - \sum_{i=1}^{k-1} \eta_{i,m} E[X_{i,m}] \right) \left(1 - \sum_{i=1}^k \eta_{i,m} E[X_{i,m}] \right)}. \quad (17)$$

Based on this expected average waiting time, the probability of packet loss due to the

² To simplify the notation, here as well as in the subsequent part of the chapter, we do not explicitly state the dependency of the throughput, goodput, packet error rate etc. on the optimal modulation strategy chosen for that link, but assume that this is implicitly considered.

expiration can be calculated by the tail distribution of the waiting time:

$$\text{Prob}(W_{k,m} > t) \approx \left(\sum_{i=1}^K \eta_{i,m} E[X_{i,m}] \right) \exp\left(-\frac{t \sum_{i=1}^K \eta_{i,m} E[X_{i,m}]}{E[W_{k,m}]}\right). \quad (18)$$

In equation (18), we adopt the G/G/1 tail distribution approximation based on the work of [JTK01][ACW95]. Let us now express this probability in terms of the packet delay deadline d_k . This probability of packet loss (at the intermediate node m) is denoted $P_{k,m}$ (recall that the waiting time is assumed to dominate the overall delay):

$$P_{k,m} = \text{Prob}(W_{k,m} + E[W_{k,0}] > d_k), \quad (19)$$

where $E[W_{k,0}]$ is the expected queuing delay of the packets at the source queue, which depends on the number of packets of a class in one GOP. Then, the end-to-end packet loss probability P_k for class f_k can be calculated as:

$$P_k = 1 - (1 - P_{k,0}) \left(1 - \sum_{m=1}^M \beta_{k,m} P_{k,m} \right). \quad (20)$$

We can observe from the above derivation that the resulting end-to-end packet loss probability for each class f_k is affected by the various cross-layer parameters (as shown in equation (4)): the relay selecting parameters $\beta_{k,m}$, the modulation and coding scheme $\theta_{k,m_h,m_{h+1}}$ that affects the average queue waiting time. Finally, the received video quality can be estimated by substituting equation (20) into equation (1).

C. Generalization to the multi-hop case

We now extend our analysis to a general directed acyclic multi-hop overlay network (as shown in Figure 2.1) by cascading the elementary structure. Importantly, note that the deployed structure is very general and any multi-hop network that can be modeled as a directed acyclic graph can be modified to fit into this overlay structure by introducing virtual nodes [EM93]. We introduce virtual nodes with zero service time for users that have a smaller number of hops, and fix the path for particular classes to pass through the virtual node (by setting $\beta_{k,h,m_h} = 1$). Methods to construct such overlay structures given a specific multi-hop network and a set of transmitting-receiving pairs can be found in

[WR03][Jan02].

The network is assumed to have H hops from sources to destinations. All the queues in the intermediate nodes perform a preemptive-repeat priority M/G/1 model as mentioned in the previous subsection. For the queue at node m_h , let η_{k,h,m_h} be the average arrival rate between the h -th hop and $(h+1)$ -th hop ($1 \leq h \leq H-1$), and $P_{k,h-1}$ be the packet loss due to delay expiration from the previous hop. $R_{k,h}$ is the updated arrival rate of class f_k for all the intermediate nodes between the h -th hop and $(h+1)$ -th hop, and we set $R_{k,0} = R_k$ for the source nodes. Then, the average arrival rates η_{k,h,m_h} have the following recursive relationship:

$$R_{k,h} = (1 - P_{k,h-1})R_{k,h-1}, \quad (21)$$

$$\eta_{k,h,m_h} = \beta_{k,h,m_h} R_{k,h}. \quad (22)$$

equation (21) illustrates that the video rate was reduced from hop to hop due to the packet deadline expiration. equation (22) shows that the average input rate is distributed based on the relay selecting parameters at the h -th hop.

Recall that X_{k,h,m_h} is the service time of the priority M/G/1 queue at node m_h between the h -th hop and $(h+1)$ -th hop. Given the relay selecting parameters, we can obtain the first two moments of the service time:

$$\begin{aligned} E[X_{k,h,m_h}] &\approx \sum_{m_{h+1}=1}^{M_{h+1}} \beta_{k,h+1,m_{h+1}} \frac{\hat{L}_k}{T_{k,m_h,m_{h+1}}(1 - p_{k,m_h,m_{h+1}})}, \\ E[X_{k,h,m_h}^2] &\approx \sum_{m_{h+1}=1}^{M_{h+1}} \beta_{k,h+1,m_{h+1}} \frac{\hat{L}_k^2(1 + p_{k,m_h,m_{h+1}})}{T_{k,m_h,m_{h+1}}^2(1 - p_{k,m_h,m_{h+1}})^2}. \end{aligned} \quad (23)$$

Similarly, recall W_{k,m_h} is the queue waiting time at node m_h for video class f_k . Then, the expected average value can be calculated similarly to equation (17):

$$E[W_{k,m_h}] = \frac{\sum_{i=1}^k \eta_{i,h,m_h} E[X_{i,h,m_h}^2]}{2 \left(1 - \sum_{i=1}^{k-1} \eta_{i,h,m_h} E[X_{i,h,m_h}] \right) \left(1 - \sum_{i=1}^k \eta_{i,h,m_h} E[X_{i,h,m_h}] \right)}. \quad (24)$$

Therefore, the expectation of the waiting time $E[W_{k,h}]$ over the h -th hop for packets of class f_k is:

$$E[W_{k,h}] = \sum_{m_h=1}^{M_h} \beta_{k,h,m_h} E[W_{k,m_h}]. \quad (25)$$

The probability of packet loss due to the expiration becomes:

$$\begin{aligned} P_{k,h,m_h} &= \text{Prob} \left(W_{k,m_h} > d_k - \sum_{j=0}^{h-1} E[W_{k,j}] \right) \\ &\approx \left(\sum_{i=1}^K \eta_{i,h,m_h} E[X_{i,h,m_h}] \right) \exp \left(- \frac{\left(d_k - \sum_{j=0}^{h-1} E[W_{k,j}] \right) \left(\sum_{i=1}^K \eta_{i,h,m_h} E[X_{i,h,m_h}] \right)}{E[W_{k,m_h}]} \right). \end{aligned} \quad (26)$$

Similar to equation (19), the probability of packet loss at the node m is the waiting time tail distribution when the accumulated waiting time exceeds the delay deadline. Then, the expected hop-by-hop packet loss probability of the hop h is:

$$P_{k,h} = \sum_{m_h=1}^{M_h} \beta_{k,h,m_h} P_{k,h,m_h}. \quad (27)$$

Recursively, we can write:

$$(1 - P_{k,H-1}) \cdot R_{k,H-1} = \prod_{h=0}^{H-1} (1 - P_{k,h}) \cdot R_k. \quad (28)$$

Finally, the received video quality can be estimated by substituting equation (27) into equation (5) and equation (1). Note that the model can be applied even for the 1-hop case, the average waiting time at the source $E[W_{k,0}]$, and the packet loss probability $P_{k,0} = P(W_{k,0} > d_k)$ can be obtained using the above equations.

V. PRIORITY QUEUING ANALYSIS CONSIDERING INTERFERENCE OF WIRELESS NETWORKS

In the Section IV, we determined the priority queuing analysis without considering the interference of other simultaneous transmissions. This can be considered as being the case in a network with multiple orthogonal channels for transmission. However, for regular wireless networks, the interference is severely rooted in the broadcasting nature of the medium. Hence, it is important to include the performance degradation due to the interference effect. First, we introduce two matrices to describe the interference in Section V.A. Then, in Section V.B, we present the priority queuing analysis with the virtual-queue service time modification.

A. Incidence matrix and interference matrix

In [TG03], a rate matrix was introduced to describe the state of the network at a given

time. In [WCZ05], an elementary capacity graph was used to represent the physical layer state of the various links. In [XJB04], a node-link incidence matrix was used. Here, we assume a similar incidence matrix to describe a network with n nodes and l links. This matrix is defined as $\mathbf{A} = [A_{ij}]_{n \times l}$, where i is the nodes' index, and j is the index of directional links:

$$A_{ij} = \begin{cases} 1 & , \text{ if link } j \text{ flows into node } i \\ -1 & , \text{ if link } j \text{ flows out of node } i \\ 0 & , \text{ otherwise} \end{cases} \quad (29)$$

The existence of links is determined by the SINR value, i.e. links having a SINR below a predetermined value are not considered viable [Kri02].

Additionally, we introduce here a matrix $\mathbf{C}$ to characterize the interference in the multi-hop network. Two types of interference are considered in this chapter. One type of interference is the transmission rate decrease due to the SINR degradation. The other type of interference, which is referred as the feasibility of simultaneous transmission links, is from the fact that in a regular wireless network environment, a node cannot transmit and receive data at the same time, and it cannot transmit two flows and receive two flows at the same time due to the wireless radio limitation. First, let $\mathbf{B} = [B_{jk}]_{l \times l} = \mathbf{A}^T \mathbf{A}$. If $B_{jk} > 0$, there exists transmitter-receiver interference between link j and k . If $B_{jk} < 0$, there exists transmitter-transmitter or receiver-receiver interference between link j and k . If $B_{jk} = 0$, there exists no second type of interference between link j and k . The interference matrix $\mathbf{C} = [C_{jk}]_{l \times l}$ is defined as:

$$C_{jk} = \begin{cases} 1, & \text{ if } B_{jk} = 0 \\ 0, & \text{ if } B_{jk} \neq 0 \end{cases} \quad (30)$$

Note that the interference matrix $\mathbf{C}$ is defined to observe the feasibility of simultaneous transmission links. Link j and link k could transmit simultaneously if and only if $C_{jk} = 1$.

Given the interference matrix $\mathbf{C}$, the set $\Phi = \{\Phi_z\}$ represents all the combinations of transmission links that can transmit simultaneously. A combination Φ_z must satisfy the following condition:

$$\prod_{j,k \in \Phi_z} C_{jk} = 1. \quad (31)$$

We denote link $l_h = (m_h, m_{h+1})$ to be the link connecting node m_h with node m_{h+1} . Denote the air-time fractions r_{Φ_z} as the average time portion (a probability) for the link combination Φ_z to happen in a SI [IEE03]. Note that $\sum_{\Phi} r_{\Phi_z} = 1$. In general, the decision on the routing as well as the nodes participating in the video streaming session depend largely on a number of system-related factors that transcend the video streaming problem [AMV06] (e.g. node cooperation strategies/incentives and network coordination and routing policies imposed by the utilized protocols). Hence, such information can be provided by the negotiation and arbitration of the polling-based contention-free MAC protocol statistically. We define $PR_{\Phi_z, l_h}^{(I)}$ as the probability that a particular combination of links that simultaneously transmit (i.e. Φ_z) occurs, given that the link l_h is transmitting:

$$PR_{\Phi_z, l_h}^{(I)} = \begin{cases} 0, & \text{if } l_h \notin \Phi_z \\ \frac{r_{\Phi_z}}{\sum_{l_h \in \Phi_z} r_{\Phi_z}}, & \text{if } l_h \in \Phi_z \end{cases}. \quad (32)$$

B. Priority queuing with virtual-queue service time modification

Since our model has only one server per queue at each intermediate node, only one transmission can take place at a time from the same queue. However, we still have to avoid the case that a receiver simultaneously receives more than one packet from distinct nodes. In fact, for a regular polling-based wireless network with a single channel, the packets are kept in the servers while waiting for the interfering transmission to finish the service. Hence, we assume that the servers at each intermediate node form a “virtual queue” to the same destination [ML99]. In a virtual queue, packets of different queues wait in turns at the servers to be transmitted to the same destination. The concept is similar to the “service on vacation” [BG87] in queuing theory, and the waiting time of the virtual queue can be regarded as the “vacation time”. The total sojourn time (queue waiting time plus the transmission service time) of the virtual queue now becomes the actual service time at each of the intermediate nodes. As the packet in the server is

waiting in the virtual queue, the node is able to receive packets from the previous hop. For simplicity, we assume that the receiving process can still be approximated as a regular Poisson process. In addition, the arrival process of the virtual queue is also assumed to be an M/G/1 priority queue.

Let $\tilde{\eta}_{k,m_{h+1}}$ be the average arrival rate of class k to the virtual M/G/1 queue that has node m_{h+1} as its destination. Next, we denote all random variables for the virtual queues with a tail on it.

$$\tilde{\eta}_{k,m_{h+1}} = \beta_{k,h+1,m_{h+1}} R_{k,h}, \quad (33)$$

where $R_{k,h}$ is the updated input rate after the h -th hop defined in equation (21).

Denote X_{k,l_h,Φ_z} as the service time of the priority M/G/1 queue in the node m_h , when the transmission is on the link $l_h = (m_h, m_{h+1})$ in the combination Φ_z . Both the first moment and the second moment need to be modified, since the channel is different due to the SINR degradation from simultaneous transmissions. $T_{k,m_h,m_{h+1}}$ is changed into T_{k,l_h,Φ_z} , and $p_{k,m_h,m_{h+1}}$ is changed into p_{k,l_h,Φ_z} . Let $\hat{L}_{k,l_h,\Phi_z}$ represent the new effective packet length including the time overhead $Time_o$ for MAC operations similar to equation (14). The first three moments of X_{k,l_h,Φ_z} become (assuming $p_{k,l_h,\Phi_z} \gamma_{k,l_h}^{MAX} + 1 \ll 1$):

$$\begin{aligned} E[X_{k,l_h,\Phi_z}] &\approx \frac{\hat{L}_{k,l_h,\Phi_z}}{T_{k,l_h,\Phi_z}(1 - p_{k,l_h,\Phi_z})}, \quad E[X_{k,l_h,\Phi_z}^2] \approx \frac{\hat{L}_{k,l_h,\Phi_z}^2(1 + p_{k,l_h,\Phi_z})}{T_{k,l_h,\Phi_z}^2(1 - p_{k,l_h,\Phi_z})^2}, \\ E[X_{k,l_h,\Phi_z}^3] &\approx \frac{\hat{L}_{k,l_h,\Phi_z}^3(1 + 4p_{k,l_h,\Phi_z} + p_{k,l_h,\Phi_z}^2)}{T_{k,l_h,\Phi_z}^3(1 - p_{k,l_h,\Phi_z})^3}. \end{aligned} \quad (34)$$

Let $\tilde{S}_{k,m_{h+1}}$ be the service time of the virtual queue having destination node m_{h+1} . The first moment of service time for class f_k of this virtual M/G/1 queue can be obtained as:

$$E[\tilde{S}_{k,m_{h+1}}] = \sum_{m_h=1}^{M_h} \beta_{k,h,m_h} E[\tilde{S}_{k,m_h,m_{h+1}}] = \sum_{m_h=1}^{M_h} \beta_{k,h,m_h} \sum_z PR_{\Phi_z,l_h}^{(I)} E[X_{k,l_h,\Phi_z}], \quad (35)$$

where $E[\tilde{S}_{k,m_h,m_{h+1}}]$ is the statistical average service time from intermediate node m_h to node m_{h+1} through all the possible transmission combinations Φ_z . The second and the third moment are similarly:

$$E[\tilde{S}_{k,m_{h+1}}^2] = \sum_{m_h=1}^{M_h} \beta_{k,h,m_h} \sum_z PR_{\Phi_z,l_h}^{(I)} E[X_{k,l_h,\Phi_z}^2], \quad E[\tilde{S}_{k,m_{h+1}}^3] = \sum_{m_h=1}^{M_h} \beta_{k,h,m_h} \sum_z PR_{\Phi_z,l_h}^{(I)} E[X_{k,l_h,\Phi_z}^3]. \quad (36)$$

Let random variable $\tilde{W}_{k,m_{h+1}}$ be the waiting time of the virtual queue with node m_{h+1} as its destination. From the Pollaczek-Khinchin formula, the first moment of $\tilde{W}_{k,m_{h+1}}$ for the virtual queue [BG87] is:

$$E[\tilde{W}_{k,m_{h+1}}] = \frac{\sum_{i=1}^k \tilde{\eta}_{i,m_{h+1}} E[\tilde{S}_{i,m_{h+1}}^2]}{2 \left(1 - \sum_{i=1}^k \tilde{\eta}_{i,m_{h+1}} E[\tilde{S}_{i,m_{h+1}}] \right)}. \quad (37)$$

Using the Takacs recurrence formula [Kle75], we have the second moment:

$$E[\tilde{W}_{k,m_{h+1}}^2] = 2E[\tilde{W}_{k,m_{h+1}}]^2 + \frac{\sum_{i=1}^k \tilde{\eta}_{i,m_{h+1}} E[\tilde{S}_{i,m_{h+1}}^3]}{3 \left(1 - \sum_{i=1}^k \tilde{\eta}_{i,m_{h+1}} E[\tilde{S}_{i,m_{h+1}}] \right)}. \quad (38)$$

The expected virtual queue waiting time $E[\tilde{W}_{k,m_{h+1}}]$ are the same through all the intermediate nodes m_h , since the packets eventually join the same virtual queue (to node m_{h+1}). However, the sojourn time $\tilde{D}_{k,m_h,m_{h+1}}$ of the virtual queue will be different, since the transmission time from various intermediate nodes m_h to the same m_{h+1} are different.

The first moment and the second moment of the sojourn time are:

$$E[\tilde{D}_{k,m_h,m_{h+1}}] = E[\tilde{W}_{k,m_{h+1}}] + E[\tilde{S}_{k,m_h,m_{h+1}}], \quad (39)$$

$$E[\tilde{D}_{k,m_h,m_{h+1}}^2] \approx E[\tilde{W}_{k,m_{h+1}}^2] + 2E[\tilde{W}_{k,m_{h+1}}]E[\tilde{S}_{k,m_h,m_{h+1}}] + E[\tilde{S}_{k,m_h,m_{h+1}}^2]. \quad (40)$$

Note that equation (40) is obtained by ignoring the correlation of the waiting and service time. Finally, the service time of the priority M/G/1 queue at the intermediate node m_h can be modified as (similar to equation (23)):

$$E[\tilde{X}_{k,m_h}] = \sum_{m_{h+1}=1}^{M_{h+1}} \beta_{k,h+1,m_{h+1}} E[\tilde{D}_{k,m_h,m_{h+1}}], \quad E[\tilde{X}_{k,m_h}^2] = \sum_{m_{h+1}=1}^{M_{h+1}} \beta_{k,h+1,m_{h+1}} E[\tilde{D}_{k,m_h,m_{h+1}}^2]. \quad (41)$$

Let $W_{k,m_h}^{(I)}$ be the waiting time for a packet of class f_k that goes through an intermediate node m_h when the interference effect is considered:

$$E[W_{k,m_h}^{(I)}] = \frac{\sum_{i=1}^k \eta_{i,m_h} E[\tilde{X}_{i,m_h}^2]}{2 \left(1 - \sum_{i=1}^{k-1} \eta_{i,m_h} E[\tilde{X}_{i,m_h}] \right) \left(1 - \sum_{i=1}^k \eta_{i,m_h} E[\tilde{X}_{i,m_h}] \right)}. \quad (42)$$

The expectation of the waiting time over the h -th hop for packets of class f_k is (as

equation (25)):

$$E[W_{k,h}^{(I)}] = \sum_{m_h=1}^{M_h} \beta_{k,h,m_h} E[W_{k,m_h}^{(I)}]. \quad (43)$$

The probability of packet loss of class f_k at intermediate node m_h due to the expiration now becomes:

$$\begin{aligned} P_{k,m_h}^{(I)} &= \text{Prob} \left(W_{k,m_h}^{(I)} > d_k - \sum_{j=0}^{h-1} E[W_{k,j}^{(I)}] \right) \\ &\approx \left(\sum_{i=1}^K \eta_{i,m_h} E[\tilde{X}_{i,m_h}] \right) \exp \left(- \frac{\left(d_k - \sum_{j=0}^{h-1} E[W_{k,j}^{(I)}] \right) \sum_{i=1}^K \eta_{i,m_h} E[\tilde{X}_{i,m_h}]}{E[W_{k,m_h}^{(I)}]} \right). \end{aligned} \quad (44)$$

VI. CONVERGENCE DISCUSSION

Next, we show that the self-learning routing algorithm will converge to a steady-state under certain assumptions:

Lemma: Given a set of fixed (pre-determined) outgoing relay selecting parameters $\{\beta_{k,h+1,m_{h+1}} \mid m_{h+1} = 1, \dots, M_{h+1}, k = 1, \dots, K\}$, the incoming relay selecting parameters $\{\beta_{k,h,m_h} \mid m_h = 1, \dots, M_h, k = 1, \dots, K\}$ will converge to a steady-state, under the assumption that the network condition is not changing over time, and given stationary statistics for the video sources.

Proof:

Since all the $\beta_{k,h+1,m_{h+1}}$ are fixed and the network condition is not changing, the first two moments of the service time $E[X_{k,h,m_h}]$ and $E[X_{k,h,m_h}^2]$ remain constant over time (see equation (23)). Thus, the balking arrival queues converge to a steady state (see [Kle75] for more details) having the average queue waiting times $E[W_{k,m_h}]$. In addition, the fixed $\beta_{k,h+1,m_{h+1}}$ also implies that the expected delays $E[\text{Delay}_{k,m_{h+1}}]$ from the relay m_{h+1} (in the next hop) are fixed over time for every class of traffic. Consequently, from equation (7), $E[\text{Delay}_{k,m_h}]$ will also converge to a steady state for every node m_h (at the current hop). This ensures that the incoming relay selecting parameters β_{k,h,m_h} will also have a steady-state, because they only depend on these $E[\text{Delay}_{k,m_h}]$ (see equation (8)). ■

Theorem: The self-learning policy over an H -hop directed acyclic overlay network will converge to a steady-state solution for the relay selecting parameters.

Proof:

Since the relay selecting parameters β_{k,H,m_H} at the last hop are fixed according to the pre-determined destination node of each traffic class, the relay selecting parameters $\beta_{k,H-1,m_{H-1}}$ will converge in time to a steady-state according to the above **Lemma**. Then, starting from the last hop, the relay selecting parameters of the entire multi-hop infrastructure will converge sequentially to a steady-state. ■

VII. SIMULATION RESULTS

In this section, two video sequences “Mobile”, and “Coastguard” (16 frames per GOP at a frame rate of 30 Hz) are compressed using an embedded scalable video codec [AMB04]. Each scalable bitstream is separated into 4 classes ($N_v = 4, K = 8$). The characteristic parameters of the video classes of the two video streams are given in Table 2.1 (see [VT07][VAH06] for more details on how to determine these parameters). In the simulation, the packet length L_k is up to 1000 bytes. No further fragmentation is performed at the lower layers (network or MAC layer). The application playback delay deadline is set to 0.533 seconds. We analyze the performance of our algorithms in terms of the received video quality (PSNR) of the various users. We compare our analytical results based on a steady-state analysis of the proposed distributed solution with the simulation results obtained using a multi-hop overlay network test-bed [KV04].

TABLE 2.1 THE CHARACTERISTIC PARAMETERS OF THE VIDEO CLASSES OF THE TWO VIDEO SEQUENCES.

Video Classes	Video 1 “Mobile” 1668 Kbps				Video 2 “Coastguard” 1500 Kbps			
	f_1	f_4	f_6	f_8	f_2	f_3	f_5	f_7
f_k								
λ_k (dB/Kbps)	0.0170	0.0064	0.0042	0.0031	0.0105	0.0064	0.0048	0.0042
R_k (Kbps)	556	333	334	445	500	300	300	400

In our simulation, we captured the packet-loss pattern under different channel conditions (described in the chapter by the link SINR) using our wireless streaming

test-bed [KV04]. In this way, we can assess the efficiency of our system under real wireless channel conditions and link adaptation mechanisms currently deployed in state-of-the-art 802.11a/g wireless cards with 802.11e extension. Link adaptation selects one appropriate physical-layer mode (modulation and channel coding) depending on the link condition, in order to continuously maximize the experienced goodput [KV04]. The various efficiency levels are represented by varying the available time fraction for the contention-free period in the polling-based MAC protocol, which induces the various available transmission rates for the video packets over the links. In our elementary structure, these network efficiency levels are represented by the transmission rate multiplier T_m ranging from 0.3 Mbps to 0.6 Mbps. A larger transmission rate multiplier gives a higher network efficiency.

In the analytical results, we determine the end-to-end packet loss rate based on the average measured SINR and the average T_m obtained for each link from the test-bed over the duration of the simulation experiments. Figure 2.5 shows the elementary structure with the two video streams and four intermediate nodes. The analytical expected end-to-end delays $E[D_k]$ of the packets in the eight classes are also shown for different network efficiency levels. The dashed line represents the delay deadline. Once the end-to-end delay exceeds the delay deadline, the packets in that class are dropped. Table 2.2 shows the results of the end-to-end packet loss probability for each video class using our priority queuing approach. The almost-binary results (0 or 100%) obtained by our packet loss analysis are due to the fact that in equation (44), we approximate $delay_{k,m_h}^{curr}$ (current delay, see equation (12)) using $\sum_{j=0}^{h-1} E[W_{k,j}^{(I)}]$ instead of $\sum_{j=0}^{h-1} W_{k,j}^{(I)}$, i.e. we use the expected waiting time instead of the exact waiting time, as this is only known instantaneously, at each queue, during the streaming simulation. Note though that the estimations of P_k are accurate enough for the important classes, thereby leading to an accurate video quality estimation.

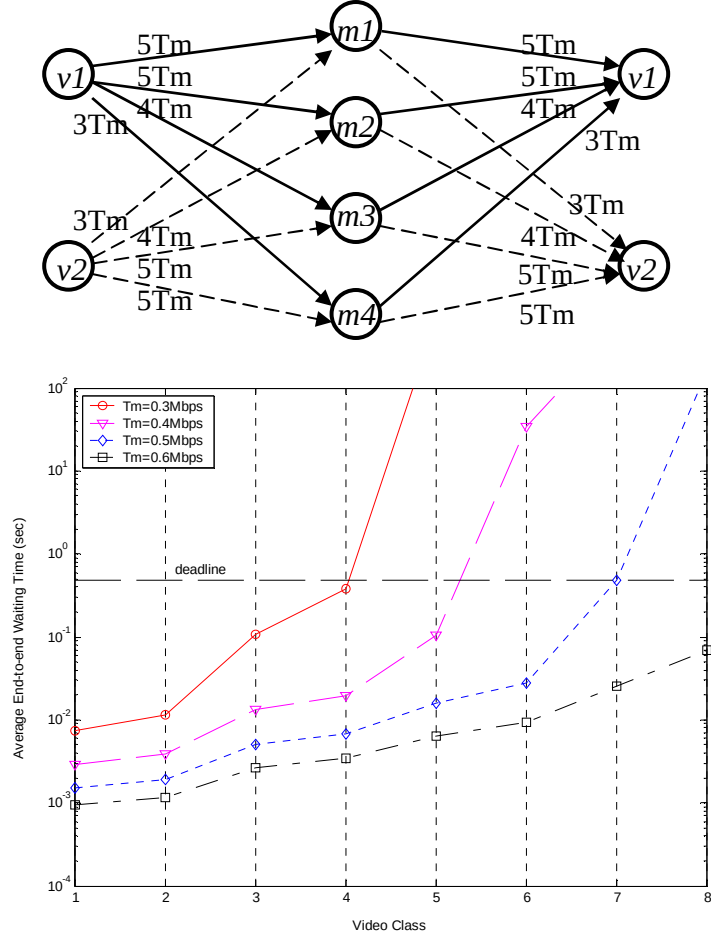


Fig. 2.5 (a) Network settings of the elementary structure.
(b) Analytical average end-to-end waiting time of the 8 video classes.

TABLE 2.2
ANALYTICAL AND SIMULATION RESULTS FOR UNIFORM RELAY SELECTING PARAMETERS WITH DIFFERENT
NETWORK EFFICIENCIES OVER THE ELEMENTARY STRUCTURE.

	Video 1 “Mobile”					Video 2 “Coastguard”				
Tm (Mbps)	Packet loss rate P_k				Y-PSNR (dB)	Packet loss rate P_k				Y-PSNR (dB)
	f_1	f_4	f_6	f_8		f_2	f_3	f_5	f_7	
Analytical result										
0.3	0	8.2%	100%	100%	30.15	0	0	100%	100%	32.49
0.4	0	0	100%	100%	30.34	0	0	0	100%	33.93
0.5	0	0	0	100%	31.74	0	0	0	15%	35.34
0.6	0	0	0	0	33.12	0	0	0	0	35.61
Simulation result										
0.29	0	39%	78%	99%	29.34	4.5%	23%	69%	98%	32.26
0.41	0	6.5%	32%	95%	31.41	0.4%	2.5%	19%	71%	34.29
0.50	0	0.5%	9.8%	77%	32.00	0	0	3.1%	30%	35.10
0.61	0	0.3%	2.1%	10%	33.05	0	0	0.8%	2.2%	35.59

In Figure 2.6, we consider a larger network (the 6-hop network) with the same network settings as in Figure 2.5. By increasing the number of hops, both the average queue waiting time and the end-to-end packet error rate increase. Comparing the results in Table 2.3 with the results in Table 2.2, the error between the analytical and simulation results decreases, since the assumption that the waiting time dominates the overall delay is more accurate in a larger network. The accuracy of the analysis could be further improved by separating the video into a larger number of classes.

The results of the proposed self-learning policy are shown in Table 2.4. Note that in Table 2.2 and Table 2.3, we use a uniform relay selection among the intermediate nodes of each hop. The resulting primary paths are marked in bold arrows in the network plot of Figure 2.7. We observe significant improvements in terms of end-to-end packet loss and video qualities using the self-learning policy. Interestingly, similarly to the Bellman-Ford algorithm, we found that this policy tries to transmit the two video streams over distinct paths in order to limit the effect of interference and congestion among the flows.

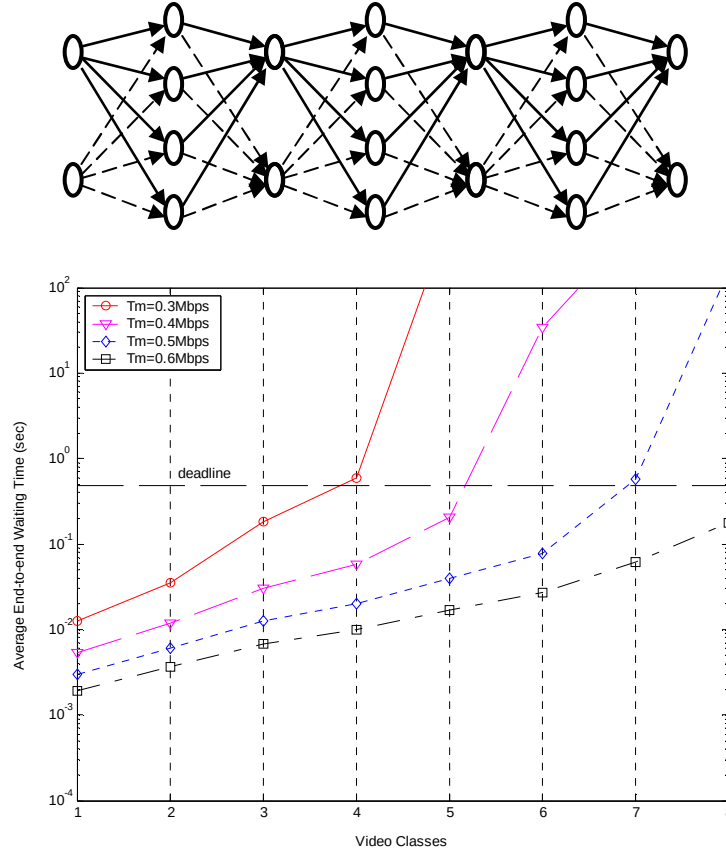


Fig. 2.6. (a) Network settings of the 6-hop overlay network (by cascading the elementary structure).
(b) Analytical average end-to-end waiting time of the 8 video classes.

TABLE 2.3
ANALYTICAL AND SIMULATION RESULTS FOR UNIFORM RELAY SELECTING PARAMETERS WITH DIFFERENT NETWORK EFFICIENCIES OVER THE 6-HOP NETWORK.

	Video 1 “Mobile”					Video 2 “Coastguard”				
Tm (Mbps)	Packet loss rate P_k				Y-PSNR (dB)	Packet loss rate P_k				Y-PSNR (dB)
	f_1	f_4	f_6	f_8		f_2	f_3	f_5	f_7	
Analytical result										
0.3	0	100%	100%	100%	28.20	0	0.3%	100%	100%	32.48
0.4	0	0	100%	100%	30.34	0	0	0.4%	100%	33.92
0.5	0	0	0.1%	100%	31.74	0	0	0	100%	33.93
0.6	0	0	0	1%	33.12	0	0	0	0.1%	35.61
Simulation result										
0.30	0	75%	97%	100%	28.39	7.7%	42%	88%	100%	31.86
0.39	0	21%	65%	99%	30.21	0.1%	11%	38%	93%	33.56
0.51	0	3.4%	12%	92%	31.35	0	1.6%	12%	64%	33.88
0.60	0	0	1.1%	39%	32.85	0	0	0.4%	10%	35.58

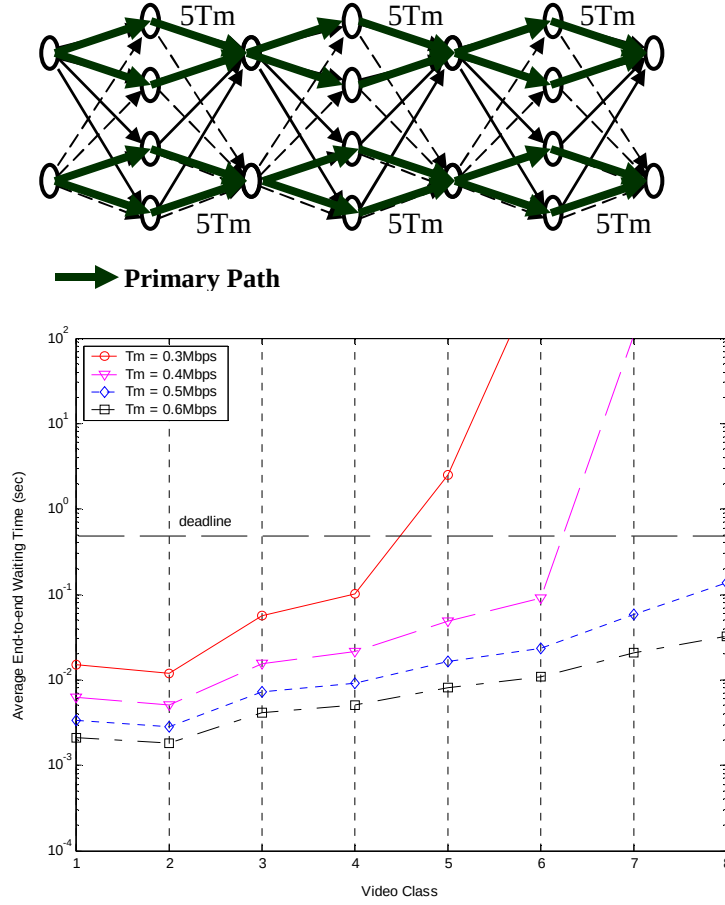


Fig. 2.7. (a) Primary paths of the 6-hop overlay network using self-learning policy. (b) Analytical average end-to-end waiting time of the 8 video classes.

TABLE 2.4 ANALYTICAL AND SIMULATION RESULTS FOR SELF-LEARNING POLICY RELAY SELECTING PARAMETERS WITH DIFFERENT NETWORK EFFICIENCIES (THE ANALYTICAL RESULTS ARE APPROXIMATED ACCORDING TO THE PRIMARY PATH SELECTED BY THE SELF-LEARNING POLICY).

	Video 1 “Mobile”					Video 2 “Coastguard”				
Tm (Mbps)	Packet loss rate P_k				Y-PSNR (dB)	Packet loss rate P_k				Y-PSNR (dB)
	f_1	f_4	f_6	f_8		f_2	f_3	f_5	f_7	
Analytical result										
0.3	0	0	100%	100%	30.34	0	0	100%	100%	32.49
0.4	0	0	0	100%	31.74	0	0	0	100%	33.93
0.5	0	0	0	0	33.12	0	0	0	0	35.61
0.6	0	0	0	0	33.12	0	0	0	0	35.61
Simulation result										
0.31	0.4%	21%	53%	83%	30.42	0	8.3%	35%	66%	33.27
0.42	0	0	7.3%	48%	32.53	0	0	0.5%	14%	35.23
0.50	0	0	0	3.3%	33.10	0	0	0	1.2%	35.61
0.60	0	0	0	0	33.10	0	0	0	0	35.61

In Table 2.5, we compare the proposed “Self-learning Policy” with a state-of-the-art routing algorithm [JM96] – “Fixed Optimal Path” and a multi-path routing algorithm [MD01] – “Fixed Multi-path”. In “Fixed Optimal Path”, we statically select the links for transmission such that the goodput is maximized (determined a single path per class). In “Fixed Multi-path”, besides the optimal path, several loop-free link-disjoint paths are also statically selected per class. As our dynamic “Self-learning policy”, the proposed priority queuing framework is also deployed for the other two algorithms using the same network settings. The simulation results show that the proposed dynamic routing approach significantly outperforms the static routing algorithms, since it provides the ability to alleviate congestion and interference.

TABLE 2.5 COMPARISON OF THE DYNAMIC SELF-LEARNING POLICY WITH THE CONVENTIONAL FIXED SINGLE-PATH AND MULTI-PATH ALGORITHMS (USING THE SAME NETWORK SETTINGS AS IN TABLE 2.4).

Method	Tm = 0.3 (Mbps) Low Network Efficiency		Tm = 0.6 (Mbps) Medium Network Efficiency	
	“Mobile” Y-PSNR (dB)	“Coastguard” Y-PSNR (dB)	“Mobile” Y-PSNR (dB)	“Coastguard” Y-PSNR (dB)
Fixed Optimal Path	24.98	30.67	31.37	34.32
Fixed Multi-path	28.39	31.86	32.85	35.58
Self-learning Policy	30.42	33.27	33.10	35.61

VIII. CONCLUSIONS

In this chapter, we present a novel distributed cross-layer streaming algorithm for the transmission of multiple videos over a multi-hop wireless network. The essential feature behind our approach is the priority queuing, based on which, the most important video packet is selected and transmitted at each intermediate node over the most reliable link, until it is successfully transmitted or its deadline is expired. Besides the application layer scheduling and MAC layer retransmission policy, the transmission strategy over the network includes selecting the optimal modulation and coding scheme. Importantly, our end-to-end cross-layer strategy also includes the selection of the appropriate relay nodes for multi-hop routing. We introduce a self-learning policy for dynamic routing that

minimizes the end-to-end packet loss for each class of the video streams. The end-to-end packet loss probabilities are estimated given the information feedback from the nodes of the next hops. The proposed distributed algorithm is fully adaptive to changes in the network, number of users, priorities of the users.

IX. APPENDIX

Algorithm 3.1 The Self-learning Algorithm

1. Initialization: Set β_{k,h,m_h} as uniform distribution at each node of each hop
 2. For each service interval
 3. For each priority class
 4. For hop $h+1$ ($0 \leq h \leq H-1$) at each node m_h
 5. Receive the $E[Delay_{k,m_{h+1}}]$ from all the nodes m_{h+1} at the end of this hop.
 6. Determine $\beta_{k,h+1,m_{h+1}}$ using the equation (3.8) and (3.9).
 7. Estimate the $E[W_{k,m_h}]$ using equation (3.42).
 8. Feedback to the nodes m_{h-1} of the previous hop h with $E[Delay_{k,m_h}]$ using equation (3.7).
 9. Send packets according to $\beta_{k,h+1,m_{h+1}}$.
-

Chapter 3

Autonomic Decision Making for Transmitting Delay-Sensitive Applications Based on Markov Decision Process

I. INTRODUCTION

Autonomic wireless networks are composed of autonomic wireless nodes (also interchangeably referred to as agents in this chapter) endowed with the capability of individually sensing the network environment, online learning the dynamic network changes based on their local information, and promptly adapting their transmission actions in an autonomous manner to optimize the utility of the applications which they are serving. The dynamic network changes include variations in network topology, wireless channel conditions, application requirements, etc.. When these network dynamics occur, the autonomic nodes can self-configure themselves and immediately react to these changes, without the need of propagating messages back and forth to a centralized coordinator. Autonomic wireless networks are especially suitable for delay-sensitive applications, since the autonomic behavior allows the wireless nodes to promptly discover local network changes and instantaneously react to these changes, such that the important data packets they are relaying will arrive at their destinations within their predetermined delay deadlines. Moreover, autonomic wireless nodes endowed with online learning capabilities can successfully model the network dynamics and foresightedly adapt their packet transmission to maximize the utility of the delay-sensitive applications.

In this chapter, we investigate how these agents in the multi-hop network optimize

their cross-layer transmission decisions to support delay-sensitive applications. There are several challenges for optimizing the performance of the delay-sensitive applications in such a context. First, optimizing the cross-layer strategies in a decentralized manner, by each node, presents its own challenges. In the multi-hop network, a node's decision impacts and is impacted by the decisions of the neighboring nodes. We refer to this coupling among the decision making performed by agents as the *spatial dependency* among the multi-hop network's nodes. To solve this coupling efficiently, we need to determine the required information exchange among the agents and computing the agents' associated utility impact due to the information exchange. Various research efforts have been devoted to optimally solving such spatial dependencies in the literature, e.g. [KMT98][SV07a][XCR08]. The second challenge occurs when considering the multi-hop network dynamics, e.g. time-variant wireless channel conditions, application requirements. However, many existing solutions that consider the spatial dependency ignore the dynamic nature of the networks. They react to experienced network dynamics in a "myopic" way by optimizing the transmission decisions based only on the information about the current network dynamics and application requirements. In the dynamic multi-hop network, however, the agents need to adopt "foresighted" adaptation by considering not only the immediate network status, but also how the network dynamics evolve over time, in order to make optimal cross-layer transmission decisions. Importantly, in addition to the spatial dependency, agents need also to consider the *temporal dependency* among their sequential decisions (performed over time), since their current decisions will also impact the information exchange in the future. By considering both the spatial and temporal dependency, an agent can evaluate the immediate and future expected delay and determines its optimal transmission action through real-time adaptation.

The delay-sensitive applications require the network to support various transmission priorities, security, robustness requirements, and stringent transmission delay deadlines

[TL08][GFX01]. In this chapter, we focus on minimizing the network delays of the delay-sensitive applications, and rely on other literatures (such as [GFX01][HN08]) for the security and reliability requirements of the delay-sensitive applications.

In the multi-hop network, the cross-layer transmission decisions, especially the route selection, cannot be determined selfishly by the autonomic nodes. In [RT02], the authors show that the performance degradation is unavoidable if the agents do not maximize the network utility (minimizing the overall network delays) in a cooperative manner. To maximize the overall performance of the applications, a Network Utility Maximization (NUM) framework has been introduced for determining the optimal transmission actions at various layers, see e.g. [KMT98][Low03][XCR08]. It has been shown that by allowing users to cooperatively exchange information, they can determine their transmission actions such that a Pareto-efficient solution can be reached in a distributed manner. However, such solutions only consider the spatial dependency among the agents, but do not address the dynamic nature (and hence, the temporal dependency) of the multi-hop network. In [NMR05][GJ07], dynamic routing policies based on queuing backpressure are proposed, which ensure that the expected delay is bounded for the delay-sensitive applications as long as the transmission rates are inside the capacity region of the network. However, computing the capacity region requires a high computational complexity [TG03] and, moreover, does not guarantee that the required delay constraints of the delay-sensitive applications are met.

Note that the network dynamics may not be known a priori in practice. Reinforcement learning solutions have been proposed for the nodes to learn the network dynamics and optimize the performance in routing [BL94][DCC05] and admission control [TB00] solutions at runtime. However, these solutions do not minimize the delays of the delay-sensitive applications. Moreover, the majority of these solutions focus on model-free reinforcement learning approaches, which are not suitable for the delay-sensitive applications due to their slow convergence rates [TO98].

In summary, there is no integrated framework that considers the spatio-temporal dependency among the agents in the multi-hop network to minimize the network delays of the delay-sensitive applications, based on application priorities, packet-based delay deadlines, and the network dynamics. In this paper, we provide a systematic framework based on which agents can optimize their cross-layer transmission actions and minimize the delays of the delay-sensitive applications, while considering the spatio-temporal dependencies among their actions. We assume that all the source and relay nodes are able to make their own cross-layer transmission decisions, which are the packet-based scheduling decisions in the application layer, the routing decisions in the network layer and the modulation and coding scheme decisions in the physical layer, based on their local information exchanges with their neighboring nodes. We propose that each agent models the queuing delay using a preemptive-repeat priority M/G/1 queuing model [SV07a] and models the network state transition over time, using maximum-likelihood state transition probabilities [BBS95]. Using these models, the agents are able to forecast the future network status and optimize their cross-layer transmission actions using a Markov Decision Process (MDP) [Put94]. Based on the MDP, the agents are able to perform foresighted decision making that considers the multi-hop network dynamics. The role of the network designer becomes in this case the careful development of policies such that the agents autonomously work towards minimizing the overall delays of the delay-sensitive applications. In this chapter, we assume that agents minimize the discounted sum of expected end-to-end delays of the delay-sensitive applications, which is referred to in this chapter as the *MDP delay value*.

In summary, the chapter makes the following contributions:

1) Distributed MDP framework that considers the spatio-temporal dependency. To account for the dynamic nature of the multi-hop network, we construct an MDP framework which minimizes the MDP delay values of the delay-sensitive applications. To address the informationally-decentralized nature of the multi-hop network, we further

decompose the MDP formulation into a distributed MDP, such that each agent in the multi-hop network can deploy its own cross-layer transmission policy based on only local information exchanges with its neighboring agents. We investigate the required local information exchange among the agents in multi-hop network and prove that the distributed MDP converges to the same optimal policy as the centralized MDP.

2) Model-based online learning approach to solve the distributed MDP. In practice, it is not known a priori to the agents how the network dynamics change over time. We propose an online model-based learning approach for the agents in multi-hop network to solve the distributed MDP in real-time. Unlike the conventional model-free reinforcement learning approaches for solving MDPs, the proposed model-based learning algorithm takes advantage of the priority queuing model to solve the distributed MDP, and provides a faster convergence rate and shorter delays for the delay-sensitive applications. The upper and lower bounds of the resulting MDP delay value are provided to verify the accuracy of the proposed model-based online learning approach at different network locations. Moreover, we compare the proposed model-based reinforcement learning approaches [BBS95][TO98] with the model-free reinforcement learning approaches [WD92][Sut88] in terms of delay performance, computational complexity, and the required information exchange overheads.

This chapter is organized as follows. In Section II, we discuss the network settings and the cross-layer transmission actions of the autonomic wireless nodes in multi-hop wireless networks, and formulate the autonomic decision making problem in the multi-hop network. In Section III, we discuss the MDP framework for solving the problem and study how to decompose the MDP into a distributed MDP to make the framework suitable for an autonomic wireless network. In Section IV, we propose a model-based online learning approach for the autonomic wireless nodes to implement the distributed MDP. Section V provides simulation results. Section VI concludes the chapter.

II. AUTONOMIC DECISION MAKING PROBLEM FORMULATION

A. Delay-sensitive application characteristics

Unlike most cross-layer design chapters that consider only a single application, we assume that there are multiple sources transmitting simultaneously delay-critical information over the multi-hop network. Let $\mathbf{V} = \{V_i\}$ represent the set of the delay-sensitive applications. Each application has a certain number of packets to be transmitted. We assume that the packets of an application V_i are prioritized into K_i priority classes. The total number of the priority classes in the network is $K = \sum_{i=1}^{|\mathbf{V}|} K_i$. Let $\{C_k, k = 1, \dots, K\}$ represent all the priority classes in the network. A priority class C_k is characterized by the following four parameters $\{\lambda_k, D_k, R_k, L_k\}$:

- λ_k represents the impact factor of the class C_k packets, which shows how critical the packets are among the delay-sensitive applications. We prioritize the packets of the delay-sensitive applications based on the impact factors. In the subsequent part of the chapter, we label the K classes (across all applications) in descending order of their priorities, i.e. $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_K$.
- D_k represents the delay deadline of the packet class. A packet of a delay-sensitive application is useful only if it is received at the destination before its delay deadline.
- R_k represents the average source rate of the packets in class C_k . Based on the source rate, the source node generates a certain number of packets per unit time, which impacts the traffic load of the multi-hop network.
- L_k represents the average packet length of the packets in class C_k . The length of a packet directly impacts the packet error rate and the transmission rate of sending a class C_k packet.

As discussed in the introduction, we assume that the multi-hop network consists of autonomic wireless nodes that make their own transmission decisions to transmit delay-sensitive packets in different priority classes. Next, we discuss the settings of the multi-hop network.

B. Autonomic multi-hop network settings

The multi-hop network is represented by a network graph $\mathcal{G}(\mathbf{V}, \mathbf{M}, \mathbf{E})$, where $\mathbf{M} = \{m_1, \dots, m_{|\mathbf{M}|}\}$ represents the set of agents and $\mathbf{E} = \{e_1, \dots, e_{|\mathbf{E}|}\}$ represents a set of edges (transmission links) that connect the various agents. There are two types of agents defined in this chapter:

- **Autonomic Source Agents (ASs).** Each AS generates a delay-sensitive application and would like to transmit the application to a predetermined destination node. The ASs packetize their applications and only determine their own cross-layer transmission actions to the next relays without specifying the cross-layer transmission actions for the relays on the entire transmission paths until the destinations.
- **Autonomic Relay Agents (ARs).** ARs relay the packets from the AS to the corresponding destination node. Unlike the ASs, the ARs do not generate their own traffic. They make their cross-layer transmission decisions and forward the packets for the ASs.

To enable us to better discuss the various networking solutions, we label the agents using the same directed acyclic graph as shown in Figure 2.1, which consists of H hops from the ASs to the destination nodes. Each agent at the h -th hop will be tagged with a distinct number m_h ($1 \leq m_h \leq M_h$). Let $\mathbf{M}_h \subseteq \mathbf{M}$ represent the set of agents at the h -th hop. The agent m_h processes a priority queue and it can only transmit the packets in the queue to a subset of ARs in $\mathbf{M}_{h+1}$. Through periodic information exchange (e.g. hello message exchange in [PB94]), we assume that each agent m_h knows the existence of its neighboring nodes (i.e. the other agents $m_h' \in \mathbf{M}_h$ in the same hop and the agents $m_{h+1} \in \mathbf{M}_{h+1}$ in the next hop), as well as the interference matrix of the current hop that defines whether or not any of two different links (m_h, m_{h+1}) can transmit simultaneously.

C. Actions of the autonomic wireless nodes

An agent's *cross-layer transmission action* varies when transmitting different priority class traffic. Denote $A_{m_h} = \{A_{k,m_h}, \forall C_k\}$ as the cross-layer transmission action of agent m_h , where $A_{k,m_h} = \{\pi_{k,m_h}, \beta_{k,m_h,m_{h+1}}, \theta_{k,m_h,m_{h+1}}, m_{h+1} \in \mathbf{M}_{h+1}\} \in \mathcal{A}_{k,m_h}$ represents the action of agent m_h when sending packets in class C_k . $\mathcal{A}_{k,m_h}$ represents the set of feasible actions for the agent m_h . In this chapter, we assume that the cross-layer transmission action includes the application layer packet scheduling π_{k,m_h} of transmitting packets in class C_k , the network layer relay selection parameter $\beta_{k,m_h,m_{h+1}}$, which determines the probability of selecting a node $m_{h+1} \in \mathbf{M}_{h+1}$ in the next hop as the next relay, and $\theta_{k,m_h,m_{h+1}}$ represents the modulation and coding scheme chosen at the physical layer for transmission over the link (m_h, m_{h+1}) . The modulation and coding scheme determines the packet error rate and the transmission rate when transmitting packets in class C_k (see Section III.A for more details). Denote $\mathbf{A} = \{A_{m_h}, \forall m_h \in \mathbf{M}\}$ as the actions of all the agents in the multi-hop network. Note that the delay $Delay_k(\mathbf{A})$ of packets in class C_k is a function of all agents' actions.

D. Problem formulation

In this subsection, we discuss how to determine the cross-layer transmission decisions for transmitting the delay-sensitive applications over the multi-hop network.

- Centralized decision making

The majority of the cross-layer design chapters assume that a central controller collects all the network information and make transmission decisions for all the agents in the multi-hop network, e.g. in [SYZ05]. The centralized optimization can be performed as a rate-constrained optimization:

$$\begin{aligned} \mathbf{A}^{opt}(\mathcal{G}) = \arg \max_{\mathbf{A}} \sum_{V_i \in \mathbf{V}} \sum_{C_k \in V_i} \lambda_k R_k (1 - P_k(\mathbf{A}, \mathcal{G})), \\ \text{s.t. } [R_k (1 - P_k(\mathbf{A}, \mathcal{G})), \forall C_k] \in CR \end{aligned} \quad (1)$$

where CR represents the capacity region of the network [TG03], and $P_k(\mathbf{A}, \mathcal{G}) = \text{Prob}(Delay_k(\mathbf{A}, \mathcal{G}) > D_k)$ represents the packet loss probability of class C_k

traffic due to the delay deadline expiration. However, the above optimization is very complicated in a multi-hop network (especially the computation of the capacity region).

Alternatively, delay-constrained optimization [CF06][SV07b] that minimize the expected delay starting from the highest priority class to the lowest priority class traffic are considered. Specifically, let $\mathbf{A}_k = [A_{k,m_h}, \forall m_h \in \mathbf{M}]^1$ represent the actions of all the agents sending class C_k traffic. The following delay constrained optimization are considered:

$$\begin{aligned} \mathbf{A}_k^{subopt}(\{\mathbf{A}_1, \dots, \mathbf{A}_{k-1}\}, \mathcal{G}) = \arg \min_{\mathbf{A}_k} E[Delay_k(\mathbf{A}_k, \{\mathbf{A}_1, \dots, \mathbf{A}_{k-1}\}, \mathcal{G})] \\ \text{s.t. } Delay_k(\mathbf{A}_k, \{\mathbf{A}_1, \dots, \mathbf{A}_{k-1}\}, \mathcal{G}) \leq D_k \end{aligned} \quad (2)$$

Based on the above equation, the actions for transmitting the priority class C_k traffic can be computed after the actions for the higher priority classes $\{\mathbf{A}_1, \dots, \mathbf{A}_{k-1}\}$ are determined and the action $\mathbf{A}_k$ will not affect any of the actions for $\{\mathbf{A}_1, \dots, \mathbf{A}_{k-1}\}$. An advantage of such delay-driven approach in equation (2) is that the optimization only needs to be done for the higher priority classes, in which the delay constraints are satisfied. The packets should be simply dropped for the lower priority classes. In addition, another advantage is that the optimization can be decomposed into a fully distributed manner (as shown in the previous chapter) that does not require the global information $\mathcal{G}$ to be gathered at a central controller.

- Distributed decision making for the agent m_h

Denote $\mathcal{L}_{m_h}$ as the local information gathered by the agent m_h . The agent m_h can minimize the expected delay for the highest priority class C_k in its queue using the following optimization:

$$\begin{aligned} A_{k,m_h}^{opt}(\mathcal{L}_{m_h}) = \arg \min_{A_{k,m_h}} E[Delay_{k,m_h}(A_{k,m_h}, \mathcal{L}_{m_h})] \\ \text{s.t. } Delay_{k,m_h}(A_{k,m_h}, \mathcal{L}_{m_h}) \leq D_k - Delay_{k,m_h}^{PASS}, \end{aligned} \quad (3)$$

¹ The action $A_{k,m_h} = \{\beta_{k,m_h,m_{h+1}}, \theta_{k,m_h,m_{h+1}}, m_{h+1} \in \mathbf{M}_{h+1}\}$ hereafter does not include the application layer scheduling, since the greedy algorithm has selected the highest priority packet to be transmitted. To simplify the notation, we use the same notation for the cross-layer transmission actions and assume that the class C_k is the highest priority class existing in the queue of the agent m_h when taking the action A_{k,m_h} .

where $Delay_{k,m_h}^{PASS}$ represents the delay that has already passed when the class C_k packet arrives at the agent m_h , which is encapsulated in the packet header. $E[Delay_{k,m_h}(A_{k,m_h}, \mathcal{L}_{m_h})]$ represents the expected delay from m_h to the destination node of the class C_k traffic. Figure 3.1 (a) illustrates how such a conventional distributed decision making works. First, the agent evaluate utility (the expected delay $E[Delay_{k,m_h}(A_{k,m_h}, \mathcal{L}_{m_h})]$ in this chapter) based on local information $\mathcal{L}_{m_h}$. Then, the agent determines the transmission action using equation (3). The required local information $\mathcal{L}_{m_h}$ for computing $E[Delay_{k,m_h}(A_{k,m_h}, \mathcal{L}_{m_h})]$ will be discussed later in Section III.B.

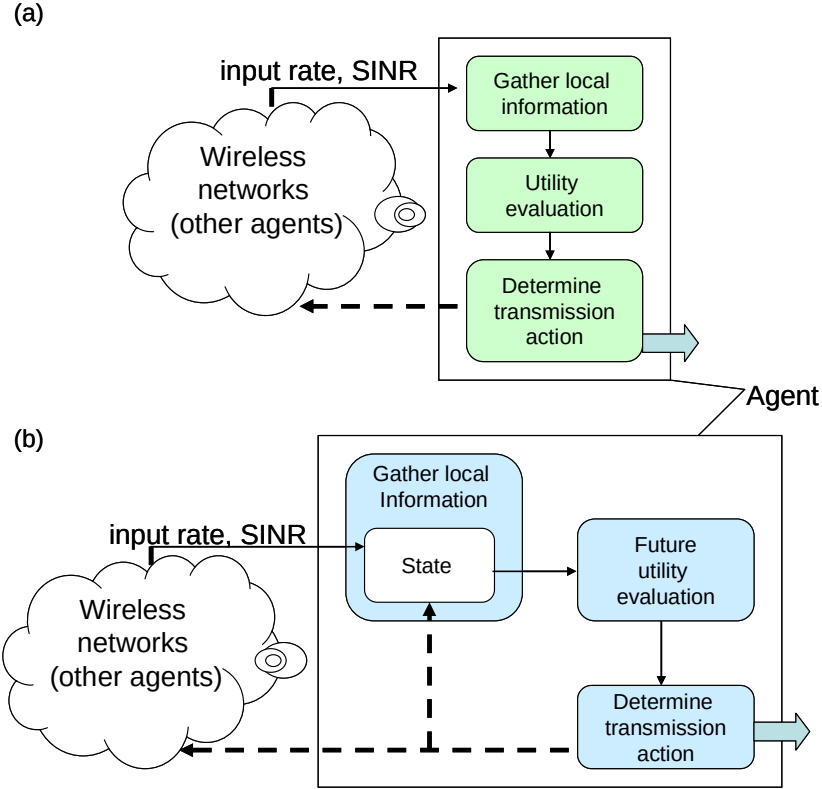


Fig. 3.1 (a) Conventional distributed decision making of an agent.
(b) Proposed foresighted decision making of an agent.

However, due to the dynamic nature of the multi-hop network, the network dynamics and hence, the local information is changing over time, it is important for the agents to

consider not only the current expected delay, but also the future expected delay as the network dynamics evolve. Figure 3.1 (b) illustrates how an agent anticipates the evolution of the network dynamics by considering the impact of its current transmission action on the future network state (which will be defined in Section III.A), and based on it, makes foresighted transmission decisions to transmit delay-sensitive applications. Next, we formulate the foresighted decision making of an agent in the multi-hop network.

- Proposed foresighted decision making for the agent m_h

Assume $E[Delay_{k,m_h}^{t_0}]$ as the expected delay of agent m_h at current service interval t_0 . Given current local information $\mathcal{L}_{m_h}^{t_0}$, the agent m_h makes foresighted decision by taking into account not only the current expected delay but also the discounted expected delays in the future service intervals, i.e.

$$\mu_{k,m_h}(\mathcal{L}_{m_h}^{t_0}) = \arg \min_{A_{k,m_h}} \left\{ \sum_{t=t_0}^{\infty} \gamma^{t-t_0} E[Delay_{k,m_h}^t(A_{k,m_h}, \mathcal{L}_{m_h}^t)] \right\}, \quad (4)$$

where $0 < \gamma < 1$ represents the discount factor to decrease the utility impact of the later transmitted packets. Equation (4) means that the agent will consider the long-term performance that values more the current utility than the future utility to determine its optimal action for transmitting the class C_k packets. We refer to the function $\mu_{k,m_h}(\mathcal{L}_{m_h})$ as the cross-layer transmission policy given the local information $\mathcal{L}_{m_h}$. In the next section, we will discuss how to compute this cross-layer transmission policy using MDP.

III. DISTRIBUTED MARKOV DECISION PROCESS FRAMEWORK

In this section, we discuss how to systematically compute the cross-layer transmission policy $\mu_{k,m_h}(\mathcal{L}_{m_h})$ for the agents in the multi-hop network. First, we define the state of the agents in Section III.A. Then, in Section III.B, we justify the Markovian property of the state transition at the agent m_h . Then, we formulate a centralized MDP in Section III.C, in which the AS makes decisions for all the relay nodes on the route of the packets in class C_k . In Section III.D, we further decompose the MDP formulation to a distributed MDP which allows all the agents to make their own decisions.

A. States of the autonomic wireless nodes

We define the network state at the agent m_h as $s_{m_h} = \{[\eta_{k,m_h}, \forall C_k], [x_{m_h, m_{h+1}}, \forall (m_h, m_{h+1})]\} \in \mathcal{X}_{m_h}$, where $x_{m_h, m_{h+1}}$ represents the channel condition, i.e. Signal-to-Interference-Noise-Ratio (SINR), over the link (m_h, m_{h+1}) and η_{k,m_h} represents the arrival rate of the class C_k packets at the agent m_h . These state values are the sufficient statistics when computing the expected queuing delay $E[W_{k,m_h}]$, when a certain action $A_{k,m_h} = \{\beta_{k,m_h, m_{h+1}}, \theta_{k,m_h, m_{h+1}}, m_{h+1} \in \mathbf{M}_{h+1}\}$ is taken. To evaluate the expected delay $E[Delay_{k,m_h}]$, the agent m_h need to first compute the expected queuing delay $E[W_{k,m_h}]$ for which the packets in class C_k will be queued.

For example, in a memory-less packet erasure channel [Kri02], given the channel condition $x_{m_h, m_{h+1}}^t$ and the modulation and coding scheme $\theta_{k,m_h, m_{h+1}}^t$ at the current service interval t , we can compute the transmission rate and the packet error rate over the link (m_h, m_{h+1}) . Let $T_{k,m_h, m_{h+1}}(\theta_{k,m_h, m_{h+1}})$ and $p_{k,m_h, m_{h+1}}(\theta_{k,m_h, m_{h+1}}, x_{m_h, m_{h+1}})$ represent the corresponding transmission rate and packet error rate. In this context, the modulation and coding scheme can be easily determined to be the one that maximize the goodput over the link, which is defined as $T_{k,m_h, m_{h+1}}^{goodput} = T_{k,m_h, m_{h+1}} \cdot (1 - p_{k,m_h, m_{h+1}})$. The packet error rate and the effective transmission rate (goodput) can be approximated using the sigmoid function [Kri02] shown in Chapter 2.3.C. Based on these, the first two moments of the service rate can be obtained. Together with the arrival rate η_{k,m_h} , the expected queuing delay $E[W_{k,m_h}]$ can be computed using an priority M/G/1 queuing model [SV07a].

Figure 3.2 illustrates how the expected delay $E[Delay_{k,m_h}]$ is evaluated based on the state s_{m_h} and the action A_{k,m_h} of the agent m_h . We assume that each agent will feed back its expected delays to all the agents in the previous hop (similar to DSDV protocols [PB94]). Hence, the agent m_h is able to select the next relay that minimize the sum of current queuing delay and the expected delay from the next hop to the destination node of class C_k , i.e.

$$\begin{aligned}
E[Delay_{k,m_h}(A_{k,m_h}, \mathcal{L}_{m_h})] &= \sum_{h'=h}^{H-1} E[W_{k,m_{h'}}(A_{k,m_{h'}}, \mathcal{L}_{m_{h'}})] \\
&= E[W_{k,m_h}(A_{k,m_h}, \mathcal{L}_{m_h})] + E[Delay_{k,m_{h+1}}(A_{k,m_{h+1}}, \mathcal{L}_{m_{h+1}})]
\end{aligned} \tag{5}$$

Importantly, the agent m_h 's transmission action will impact the information feedback $E[Delay_{k,m_{h+1}}]$, since it will select the next relay $m_{h+1} \in \mathbf{M}_{h+1}$ that feeds back different expected delay values. Moreover, the expected delay $E[Delay_{k,m_h}]$ will be fed back to the agents in the previous hop and hence impact their transmission actions. Hence, the agent m_h 's action A_{k,m_h} will affect its own future state s_{m_h} and also will influence the future expected delay as the network dynamics evolve. We denote the probability that the agent m_h has a state $s_{m_h}^{t+1}$ in service interval $t+1$ as $p(s_{m_h}^{t+1})$, which is modeled as a function of agent m_h 's current state $s_{m_h}^t$ and current action A_{k,m_h}^t , i.e.

$$p(s_{m_h}^{t+1}) \cong \hat{\mathcal{F}}_{s_{m_h}^{t+1}}(s_{m_h}^t, A_{k,m_h}^t). \tag{6}$$

Note that the real $p(s_{m_h}^{t+1})$ can be very complicated in a real network, since it is impacted by the decisions of all the agents in the previous hop as well as the interference among the agents in the current hop. Note that in our solution, the agents do not need to know the exact form of $p(s_{m_h}^{t+1})$. Online learning approaches will be discussed in Section IV for the agents to learn the state transition function in equation (6). Next, we formulate the cross-layer optimization of the agent m_h as an MDP for *each* class.

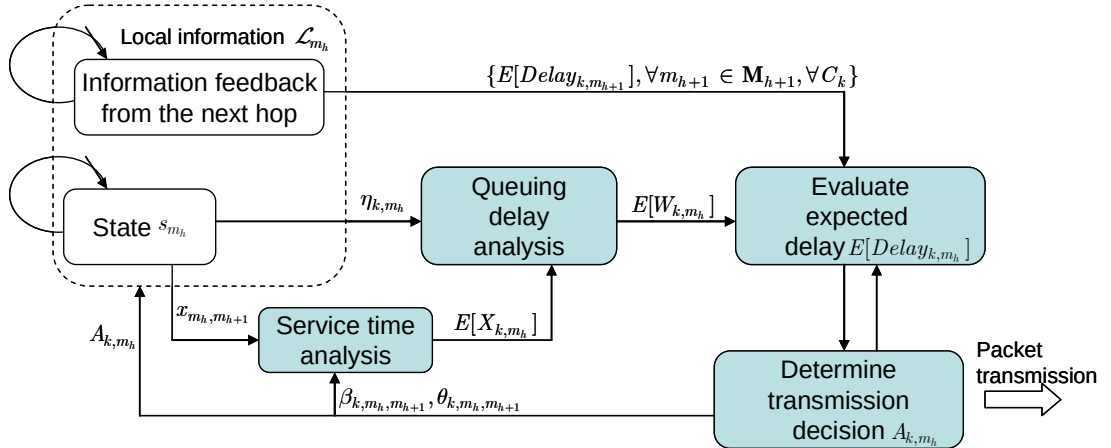


Fig. 3.2. Expected delay evaluation and the required local information.

B. Centralized Markov decision process formulation

The MDP framework of the multi-hop network is defined by a tuple $\langle \mathcal{X}, \mathcal{A}, \mathcal{T}, \mathcal{U}, \gamma \rangle$ for class C_k , which is listed as follows:

- **States:** $\mathbf{s} = [s_0, \dots, s_{H-1}] \in \mathcal{X}_0 \times \dots \times \mathcal{X}_{H-1} = \mathcal{X}$ represents the state of the network, where $s_h = [s_{m_h}, \forall m_h \in \mathbf{M}_h] \in \mathcal{X}_h$ represents the states of the agents $m_h \in \mathbf{M}_h$ at the h -th hop.
- **Actions:** $\mathbf{A} = [A_0, \dots, A_{H-1}] \in \mathcal{A}_0 \times \dots \times \mathcal{A}_{H-1} = \mathcal{A}$ represents the cross-layer transmission actions adopted across the network, where $A_h = \{A_{k,m_h}, \forall m_h \in \mathbf{M}_h\}$ represents the actions of the agents $m_h \in \mathbf{M}_h$ at the h -th hop.
- **State transition probabilities:** $T_{\mathbf{s}\mathbf{s}'}(\mathbf{A}) \in \mathcal{T} : \mathcal{X} \times \mathcal{X} \times \mathcal{A} \rightarrow [0,1]$ represents the stationary state transition probabilities from state $\mathbf{s}$ to state $\mathbf{s}'$ when the action $\mathbf{A}$ is taken. The state transition probabilities characterize what the next state $\mathbf{s}'$ is, given the current state $\mathbf{s}$ and the cross-layer transmission action $\mathbf{A}$ across the network. To simplify the centralized MDP, we approximate the state transition probabilities as

$$T_{\mathbf{s}\mathbf{s}'}(\mathbf{A}) \cong \prod_{m_h \in \mathbf{M}} \hat{\mathcal{F}}_{s_{m_h}}(s_{m_h}, A_{k,m_h}).$$

- **Cost:** The expected end-to-end delay $E[\text{Delay}_k(\mathbf{s}, \mathbf{A})] \in \mathcal{U}$ represents the cost function. As mentioned in Section III.A, we rely on a priority-based queuing model to compute the cost function (see Section IV for more detailed discussion about the priority queuing model). Note that the expected delay $E[\text{Delay}_k(\mathbf{s}, \mathbf{A})]$ of a higher priority class C_k will not be influenced by the other lower priority classes. However, if the class C_k is one of the lower priority classes, the actions and states associated with the higher priority classes are required to obtain the expected delay $E[\text{Delay}_k(\mathbf{s}, \mathbf{A})]$. The Bellman equation [Ber95] of the MDP can be formulated as:

$$\begin{aligned} V(\mathbf{s}) &= \min_{\mathbf{A} \in \mathcal{A}} \left\{ \sum_{t=1}^{\infty} \gamma^{t-1} E[\text{Delay}_k^t(\mathbf{s}, \mathbf{A})] \right\} \\ &= \min_{\mathbf{A} \in \mathcal{A}} \left\{ E[\text{Delay}_k(\mathbf{s}, \mathbf{A})] + \gamma \sum_{\mathbf{s}'} T_{\mathbf{s}\mathbf{s}'}(\mathbf{A}) V(\mathbf{s}') \right\}, \end{aligned} \quad (7)$$

where $V(\mathbf{s})$ is referred to the MDP delay value at a state $\mathbf{s}$. Recall that γ is the same discount factor in equation (4). We denote the Q-value [WD92] of taking a cross-layer

transmission action $\mathbf{A}$ at the state $\mathbf{s}$ as $Q(\mathbf{s}, \mathbf{A}) = E[\text{Delay}_k(\mathbf{s}, \mathbf{A})] + \gamma \sum_{\mathbf{s}'} T_{\mathbf{ss}'}(\mathbf{A}) V(\mathbf{s}')$. We define the centralized stationary policy as:

$$\mu_k^c(\mathbf{s}) = \arg \min_{\mathbf{A} \in \mathcal{A}} Q(\mathbf{s}, \mathbf{A}). \quad (8)$$

The Bellman equation in equation (7) can be solved using the value iteration or policy iteration in [Ber95], if the $E[\text{Delay}_k(\mathbf{s}, \mathbf{A})]$ and $T_{\mathbf{ss}'}(\mathbf{A})$ are accurately obtained. However, it is difficult to obtain these two values in a centralized manner. This is because the central controller cannot gather the global information in real-time to evaluate $E[\text{Delay}_k(\mathbf{s}, \mathbf{A})]$ due to the resulting delays and overheads in the multi-hop network when propagating information back and forth throughout the network. To overcome this challenge, we decompose the MDP in Section III.C, which allow all the agents to make their own cross-layer transmission decisions based on local information to overcome this problem. Moreover, in the distributed multi-hop network, the state transition probabilities $T_{\mathbf{ss}'}(\mathbf{A})$ may not be known a priori. To address this challenge, we will discuss an online learning approach to allow each agent to learn the state transition probabilities in the multi-hop network in Section IV.

C. Distributed Markov decision process formulation

Denote $F_h^{b,t}(m_h) = E[\text{Delay}_{k,m_h}^t]$ as the feedback information from node m_h to the agents in the previous hop and let $F_h^{b,t} = [F_h^{b,t}(m_h), m_h \in \mathbf{M}_h]$ represents the feedback information in the h -th hop. Denote $F_h^{f,t}(m_h) = \{\text{Delay}_{k,m_h}^{PASS}, \eta_{k,m_h}\}$ as the feedforward information from node m_h to the selected AR in the next hop and let $F_h^{f,t} = [F_h^{f,t}(m_h), m_h \in \mathbf{M}_h]$ represents the feedforward information in the h -th hop. Given the feedforward information $F_{h-1}^{f,t}$, the agent m_h computes the average delay $\text{Delay}_{k,h-1}^{PASS}$ of passing through the previous hops as

$$\text{Delay}_{k,h-1}^{PASS} = \sum_{m_{h-1}=1}^{M_{h-1}} \frac{\eta_{k,m_{h-1}}}{R_k} \text{Delay}_{k,m_{h-1}}^{PASS}. \quad (9)$$

If $\text{Delay}_{k,h-1}^{PASS}$ exceeds the delay deadline D_k , the packet in class C_k should be dropped and no MDP is needed for class C_k traffic at the agent m_h . Figure 3.3 shows the considered system diagram for the distributed MDP that allows the agents to exchange

information with the nodes in the neighboring hops. The agents in the same hop take transmission decisions simultaneously².

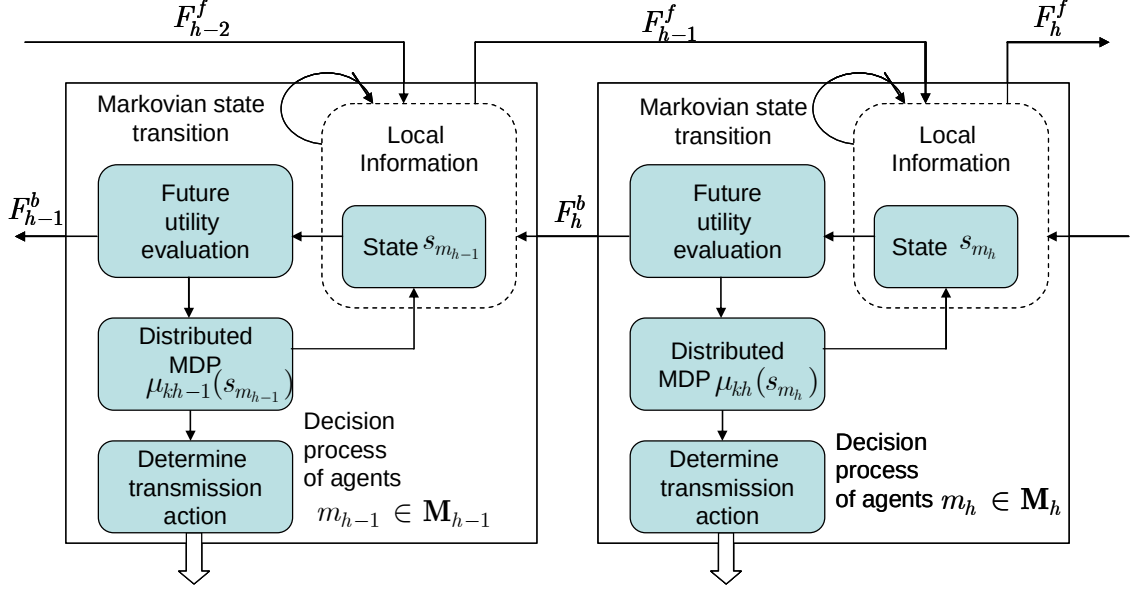


Fig. 3.3 Proposed decentralized Markov decision process framework and the necessary information exchange among the agents.

Distributed MDP at m_h to m_{h+1} , $h = 1, \dots, H - 1$

Step 1. Gather local information. From the information feedforward $F_{h-1}^{f,t}$ from the previous hop, the agent m_h computes $Delay_{k,h-1}^{PASS}$ and determine whether the distributed MDP should be performed for class C_k traffic. Then, gathers the local information $\mathcal{L}_{m_h}^t = \{s_{m_h}, \{E[Delay_{k,m_{h+1}}], \forall m_{h+1} \in \mathbf{M}_{h+1}\}\}$

Step 2. Evaluate queuing delay and state transition probabilities. Based on state s_{m_h} and action A_{m_h} the agent m_h evaluates $E[W_{k,m_h}^t]$. The state transition probabilities is computed using $T_{s_{m_h}s_{m_h}}(A_{m_h}) = \hat{\mathcal{F}}_{s_{m_h}}^{t+1}(s_{m_h}, A_{k,m_h}^t)$ in equation (6).

Step 3. Update the transmission policy. The agent m_h updates the MDP delay value:

² The transmission actions of the agents in the same hop interfere with each others and such interference is considered in the state through SINR $x_{m_h, m_{h+1}}^t$.

$$V_{m_h}^{t+1}(s_{m_h}, F_{h+1}^{b,t}) = \min_{A_{m_h} \in \mathcal{A}_h} \left\{ E[W_{k,m_h}^t(s_{m_h}, A_{m_h})] + F_{h+1}^{b,t}(A_{m_h}) + \gamma \sum_{s_{m_h}} T_{s_{m_h} s_{m_h}}(A_{m_h}) V_{m_h}^t(s_{m_h}, F_{h+1}^{b,t-1}) \right\}. \quad (10)$$

We will prove that the above iteration converges to a steady-state in the multi-hop network in the next subsection.

Denote $Q_{m_h}^t(s_{m_h}, A_{m_h}, F_{h+1}^b) = E[W_{k,m_h}^t(s_{m_h}, A_{m_h})] + F_{h+1}^{b,t}(A_{m_h}) + \gamma \sum_{s_{m_h}} T_{s_{m_h} s_{m_h}}(A_{m_h}) V_{m_h}^t(s_{m_h}, F_{h+1}^{b,t-1})$ as the Q-value at the agent m_h when a cross-layer transmission action A_h is taken in state s_h . The stationary policy of the agent m_h is $\mu_{kh}^{d,t}(\mathcal{L}_{m_h}^t) = \arg \min_{A_{m_h} \in \mathcal{A}_h} Q_{m_h}^t(s_{m_h}, A_{m_h}, F_{h+1}^b)$.

Step 4. Update the information exchange. After the policy $\mu_{kh}^{d,t}(\mathcal{L}_{m_h}^t)$ is determined, the next relay m_{h+1} is selected and m_h can then update the feedback information $F_h^{b,t+1}(m_h)$:

$$F_h^{b,t+1}(m_h) = F_{h+1}^{b,t}(m_{h+1}) + E[W_{k,m_h}^t(s_{m_h}^t, \mu_{kh}^{d,t}(\mathcal{L}_{m_h}^t))], \quad (11)$$

Based on the feedback information, the relays $m_{h-1} \in \mathbf{M}_{h-1}$ in the previous hop are able to update their updating equations as in equation (10). The wireless node m_h also needs to update its feedforward information $F_h^{f,t+1}(m_h)$:

$$F_h^{f,t+1}(m_h) = \text{Delay}_{k,h-1}^{PASS} + E[W_{k,m_h}^t(s_{m_h}^t, \mu_{kh}^{d,t}(\mathcal{L}_{m_h}^t))]. \quad (12)$$

Based on the feedforward information, the next relay m_{h+1} are able to update the $\text{Delay}_{k,h}^{PASS}$ for the class C_k .

D. Convergence of the distributed Markov decision process

In this section, we discuss the convergence of the proposed distributed MDP in the multi-hop network. We denote the remaining delay deadline as $D_{k,m_h}^{rem} = D_k - \text{Delay}_{k,h-1}^{PASS}$ for the packets in class C_k at agent m_h .

Lemma: For an agent m_h , the updating equation in equation (10) converges to the finite $V_{k,m_h}^*(s_{m_h}, F_{h+1}^b) = \lim_{t \rightarrow \infty} V_{k,m_h}^t(s_{m_h}, F_{h+1}^{b,t-1})$, if a) the class C_k is not dropped, i.e. $E[W_{k,m_h}^t(s_{m_h}, A_{m_h})] \leq D_{k,m_h}^{rem}$, and b) the feedback information $F_{h+1}^{b,t}$ depends on the current state $s_{m_h}, \forall m_h \in \mathbf{M}_h$.

Proof: See Appendix A.

If the priority of the class C_k packets is high enough, the expected queuing delay is

small compared to the remaining delay deadline and then the packets will not be dropped. The condition b) is always satisfied at the last hop of the multi-hop network, since the agent m_{H-1} has no information feedback, and it can only select the destination node of the class C_k in its action. Given the network states, the nodes in the last hop have convergence. When the MDP delay values of the agents in the last hop converges, based on Observation 1, the feedback information $F_{H-1}^{b,t}$ at the $(h-2)$ -th hop will only depends on the states of the agents $m_{H-2} \in \mathbf{M}_{H-2}$.

Theorem: The distributed MDP solution $[\mu_{kh}^{d,t}(s_{m_h}, F_{h+1}^{b,t}), h = 0, \dots, H-1]$ converges to the centralized MDP solution $\mu_k^c(s)$ if and only if $E[W_{k,m_h}^t(s_{m_h}, A_{m_h})] \leq D_{k,m_h}^{rem}, \forall m_h \in \mathbf{M}$.

Proof: See Appendix A.

In the multi-hop network, each agent determines its own cross-layer transmission policy based on the distributed MDP. The Theorem shows that if the class C_k packets are not dropped in the multi-hop network, the policy derived using the proposed distributed MDP solution converges to the optimal policy of the centralized MDP.

Note that in order to solve the Bellman equations, the agents need to know the state transition probabilities $T_{s_{m_h}s_{m_h}}(A_{m_h})$ in the updating equation (10). However, the state transition probabilities may not be known to the agents a priori. Next, we discuss the online learning approaches for solving the distributed MDP.

IV. ONLINE MODEL-BASED LEARNING FOR SOLVING THE DISTRIBUTED MDP

In this section, we discuss the online learning approaches for solving the distributed MDP introduced in the previous section, in real-time, at transmission time. We propose a novel model-based reinforcement learning approach that is suitable for the agents to transmit delay-sensitive applications over the multi-hop network. We compare the proposed model-based reinforcement learning approach with another two types of learning approaches – model-free reinforcement learning approaches and model-based multi-agent learning approaches.

Unlike the conventional model-free reinforcement learning approaches, which evaluate

the optimal cross-layer transmission policy $[\mu_{kh}^d, h = 0, \dots, H-1]$ without an explicit cost model and state transition model, the proposed model-based reinforcement learning approach adopts the priority queuing model $E[W_{k,m_h}(s_{m_h}, A_{m_h})]$ for the cost and estimates the state transition probabilities $T_{s_{m_h}s_{m_h}'}(A_{m_h})$ to solve the distributed MDP. Although the computational complexity is high, we show that the proposed model-based learning methods converge faster than the model-free learning approaches, since it takes less time for the autonomic node to explore different states and correctly evaluate the Q values. Such results are also discussed in [TO98] for more general learning settings. Compared to the model-based multi-agent learning approaches, which directly model the behaviors of the neighboring agents, the proposed model-based reinforcement learning approach requires significantly smaller information exchange overheads, since an agent only needs to model its own cost and state transition. Figure 3.4 provides a system block diagram of the proposed online learning approach at the agent m_h .

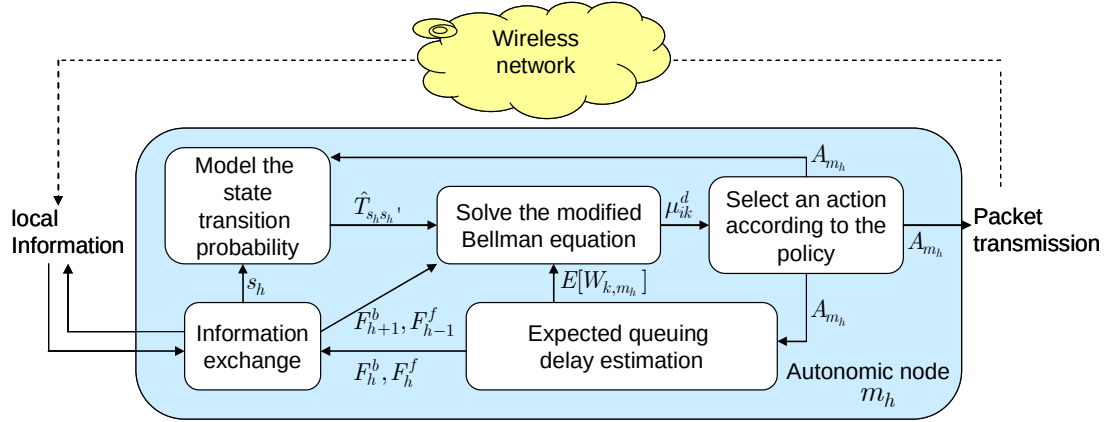


Fig. 3.4. System diagram of the proposed model-based online learning approach at the agent m_h .

A. Model-free reinforcement learning

The model-free learning methods, e.g. Q-learning [WD92] and TD-learning [Sut88], can be applied at an agent m_h to learn the next Q values $[Q_{m_h}^{t+1}(s_{m_h}, A_{m_h}), \forall s_{m_h} \in \mathcal{S}_{m_h}]$ without characterizing the state transition probabilities $T_{s_h s_h'}(A_h)$. Taking Q-learning as an example, given the feedback value $F_{h+1}^{b,t}$, the autonomic node m_h updates the Q-value using the following updating equation:

$$Q_{m_h}^{t+1}(s_{m_h}, A_{m_h}) = (1 - \rho_t)Q_{m_h}^t(s_{m_h}, A_{m_h}) + \rho_t \left\{ Cost_{k, m_h}^t + F_{h+1}^{b,t}(A_{m_h}) + \gamma \min_{A_{m_h}} Q_{m_h}^t(s_{m_h}^{t+1}, A_{m_h}) \right\}, \quad (13)$$

where $0 < \rho_t < 1$ represents the learning rate, and $\sum_t \rho_t = \infty$ and $\sum_t (\rho_t)^2 < \infty$ are ensured for the convergence of the Q-value [WD92]. The $Cost_{k, m_h}^t$ represents the local experienced cost (current queuing delay of sending packets in class C_k) and $s_{m_h}^{t+1}$ represents the next state after the agent m_h takes the cross-layer transmission action $A_{m_h}^t$. For exploration purposes, instead of following the optimal stationary policy $\mu_{kh}^{d,t}(s_{m_h}) = \arg \min_{A_{m_h} \in \mathcal{A}_h} Q_{m_h}^t(s_{m_h}, A_{m_h})$, the next action is selected according to a soft-min policy. Assume that $\pi_{kh}^t(s_{m_h}, A_{m_h})$ denotes the probability for agent m_h to take the action A_h given the state s_h . The soft-min policy $\mu_{kh}^{d,t}(s_{m_h}) = [\pi_{kh}^t(s_{m_h}, A_{m_h}), \forall A_{m_h} \in \mathcal{A}_h]$ is defined using the Boltzmann distribution [BBS95][TO98][WD92]:

$$\pi_{kh}^t(s_{m_h}, A_{m_h}) = \frac{\exp(-Q_{m_h}^t(s_{m_h}, A_{m_h}) / \tau_t)}{\sum_{\forall A_h' \in \mathcal{A}_h} \exp(-Q_{m_h}^t(s_{m_h}, A_{m_h}') / \tau_t)}, \quad (14)$$

where τ is the temperature parameter. A small τ provides a greater probability difference in selecting different actions. If $\tau \rightarrow 0$, the approach reduces back to $\mu_{kh}^{d,t}(s_h) = \arg \min_{A_h \in \mathcal{A}_h} Q_h^t(s_h, A_h)$. On the other hand, a larger τ allows the agents to explore various actions with higher probabilities. We provide detailed steps of the model-free reinforcement learning in Algorithm 3.1 in Appendix B. Table 3.1 summarizes the required local information, memory complexity, and computational complexity of the model-free reinforcement learning approaches.

TABLE 3.1. COMPLEXITY SUMMARY OF THE MODEL-FREE REINFORCEMENT LEARNING

Required local information	$\mathcal{L}_{m_h}^t = \{s_{m_h}^t, \{Cost_{k,m_h}^t, \forall C_k\}, F_{h-1}^{f,t}, F_{h+1}^{b,t}\}$		
Memory complexity	Transmission policy	State transition model	Q-value
	$ \mathcal{X}_{m_h} \times \mathcal{A}_{m_h} \times K$	Not required	$ \mathcal{X}_{m_h} \times \mathcal{A}_{m_h} \times K$
Computational complexity	$O(\mathcal{X}_{m_h} \mathcal{A}_{m_h} K)$		

In each service interval, the model-free reinforcement learning approaches need to update the Q-values of $\forall s_{m_h} \in \mathcal{X}_{m_h}, \forall C_k$, and for each state, $\min_{A_{m_h}} Q_{m_h}^t(s_{m_h}^{t+1}, A_{m_h})$ over $\forall A_{m_h} \in \mathcal{A}_{m_h}$ is calculated. Hence, the computational complexity is $O(|\mathcal{X}_{m_h}| |\mathcal{A}_{m_h}| K)$. Note that the dynamics in the multi-hop network may change before the updated policy converges when using a model-free learning approach. Hence, we consider alternative model-based reinforcement learning in the next subsection, which is more suitable for the agents in the multi-hop network due to a faster convergence rate.

B. Model-based reinforcement learning

In this section, we propose our model-based learning approach that enables the agent m_h to directly model the expected queuing delay $E[W_{k,m_h}(s_{m_h}, A_{m_h})]$ and estimate the state transition probabilities $\hat{T}_{s_{m_h}s_{m_h}'}(A_{m_h})$ to solve the Bellman equation through value iteration [Ber95]. Our approach is similar to the Adaptive-RTDP in [BBS95], where maximum-likelihood state transition probabilities are adopted. Specifically, let $\hat{T}_{s_{m_h}s_{m_h}'}^t(A_{m_h})$ denote the estimated state transition probability at the service interval t . The Q-value $Q_{m_h}^{t+1}(s_{m_h}, A_{m_h}^t)$ is updated at the agent m_h as:

$$Q_{m_h}^{t+1}(s_{m_h}, A_{m_h}^t) = (1 - \rho_t)Q_{m_h}^t(s_{m_h}, A_{m_h}^t) + \rho_t \left\{ E[W_{k,m_h}^t(s_{m_h}, A_{m_h}^t)] + F_{h+1}^{b,t}(A_{m_h}^t) + \gamma \min_{A_{m_h} \in \mathcal{A}_{m_h}} \sum_{s_{m_h}'} \hat{T}_{s_{m_h}s_{m_h}'}^t(A_{m_h}) Q_{m_h}^t(s_{m_h}', A_{m_h}) \right\}. \quad (15)$$

$s_{m_h}^{t+1}$ represents the next state after the node m_h takes the cross-layer transmission action $A_{m_h}^t$. We provide detailed steps of the model-based reinforcement learning in Algorithm 3.2 in Appendix B.

The main differences between the model-based online learning approach and model-free learning approaches are the following:

1) We model the expected queuing delay $E[W_{k,m_h}^t(s_{m_h}, A_{m_h}^t)]$ with an action $A_{m_h}^t$ realized from the policy $\mu_{k,h}^{d,t}$ using the preemptive-repeat priority M/G/1 queuing model as in the previous chapter:

$$E[W_{k,m_h}(s_{m_h}, A_{m_h})] = \begin{cases} \frac{\sum_{i=1}^k \eta_{i,m_h} E[X_{i,m_h}^2]}{2 \left(1 - \sum_{i=1}^{k-1} \eta_{i,m_h} E[X_{i,m_h}] \right) \left(1 - \sum_{i=1}^k \eta_{i,m_h} E[X_{i,m_h}] \right)} + E[X_{k,m_h}], & \text{if } E[W_{k,m_h}] \leq D_{k,m_h}^{rem} \\ \infty, & \text{otherwise} \end{cases} \quad (16)$$

From equation (16), we know that if the queuing time exceeds the remaining delay deadline D_{k,m_h}^{rem} , the expected queuing time $E[W_{k,m_h}]$ becomes infinite, since the packets will be useless (no utility gain) and they will be dropped at the agent m_h .

2) Instead of using the Q-value of the next state $s_{m_h}^{t+1}$ at each service interval, the maximum-likelihood state transition probabilities are updated and used. In Algorithm 3.2, $n_{s_{m_h}s_{m_h}}^t(A_{m_h})$ represents the observed number of times before service interval t that the action A_{m_h} is taken when the state was in s_{m_h} and made a transition to s_{m_h}' and $n_{s_{m_h}}^t(A_{m_h}) = \sum_{s_{m_h}' \in \mathcal{X}_{m_h}} n_{s_{m_h}s_{m_h}'}^t(A_{m_h})$ represents the observed number of times before service interval t that the action A_{m_h} is taken when the state was in s_{m_h} . We apply the maximum-likelihood state-transition probabilities [BBS95] in Algorithm 3.2 to update the state transition probabilities $\hat{T}_{s_{m_h}s_{m_h}}^t(A_{m_h})$.

3) Unlike regular value iteration, instead of updating the value $Q_{m_h}^{t+1}(s_{m_h}, A_{m_h}^t)$ for $\forall s_{m_h} \in \mathcal{X}_{m_h}$, we only update the value for states in a particular set $\mathcal{B}_{m_h}$. This procedure is similar to the asynchronous DP in [Ber98]. According to Theorem in Section III, in order to converge to a stationary policy, the following condition must hold

$$\frac{L_k}{T_{k,m_h,m_{h+1}}^{goodput}} \leq D_{k,m_h}^{rem} \Rightarrow \frac{L_k}{D_{k,m_h}^{rem}} \leq \frac{T_{k,m_h,m_{h+1}}}{1 + e^{-\zeta(x_{m_h,m_{h+1}} - \delta)}} \quad \text{for any possible SINR } x_{m_h,m_{h+1}} \text{ at the agent } m_h.$$

Hence, the set $\mathcal{B}_{m_h}$ is defined as

$$\mathcal{B}_{m_h} = \{s_{m_h} : x_{m_h, m_{h+1}} \geq \delta - \ln \left(\frac{T_{k, m_h, m_{h+1}} D_{k, m_h}^{rem}}{L_k} - 1 \right)^{\frac{1}{\xi}} \}, \quad (17)$$

which depends on the physical layer parameters δ and ξ of the agent m_h . We only update the Q-values of the states $s_{m_h} \in \mathcal{B}_{m_h}$ in Algorithm 3.2. The rest of the states $s_{m_h} \notin \mathcal{B}_{m_h}$ have insufficient SINR values to keep the transmission time within the remaining delay deadline D_{k, m_h}^{rem} . Based on equation (16), these states will have an infinite queuing delay and hence, they should never be selected as the next state. Table 3.2 summarizes the required local information, memory complexity, and computational complexity of the proposed model-based reinforcement learning approach.

TABLE 3.2. COMPLEXITY SUMMARY OF THE MODEL-BASED REINFORCEMENT LEARNING

Required local information	$\mathcal{L}_{m_h}^* = \{s_{m_h}^t, F_{h-1}^{f,t}, F_{h+1}^{b,t}\}$		
Memory complexity	Transmission policy	State transition	Q-value
	$ \mathcal{B}_{m_h} \times \mathcal{A}_{m_h} \times K$	$ \mathcal{B}_{m_h} \times \mathcal{B}_{m_h} \times \mathcal{A}_{m_h} \times K$	$ \mathcal{B}_{m_h} \times \mathcal{A}_{m_h} \times K$
Computational complexity	$O(K \mathcal{B}_{m_h} ^2 \mathcal{A}_{m_h})$		

The proposed model-based reinforcement learning approach has higher computational complexity than model-free reinforcement learning approaches. For the proposed model-based reinforcement learning approach, the Q-values of $\forall s_{m_h} \in \mathcal{B}_{m_h}, \forall C_k$ need to be updated in each service interval, and for each state over $\forall A_{m_h} \in \mathcal{A}_{m_h}$, the last term

$\min_{A_{m_h} \in \mathcal{A}_{m_h}} \sum_{s_{m_h}} \hat{T}_{s_{m_h} s_{m_h}}^t (A_{m_h}) Q_{m_h}^t(s_{m_h}^{t+1}, A_{m_h})$ in equation (15) is calculated.

Hence, the computational complexity is $O(K |\mathcal{B}_{m_h}|^2 |\mathcal{A}_{m_h}|)$. Although the computational complexity is significantly larger, the convergence rate of the proposed model-based reinforcement learning approach is much faster than the model-free reinforcement learning approaches. In Section V.B, we verify the convergence through extensive simulation results.

C. Upper and lower bounds of the model-based learning approach

Since the maximum-likelihood state-transition probabilities $\hat{T}_{s_{m_h}s_{m_h}}^t(A_{m_h})$ are used in the proposed model-based learning approach, there is no guarantee that the resulting MDP delay value can converge to the optimal value $V_{k,m_h}^*(s_{m_h}, F_{h+1}^b)$ in equation (7). In this subsection, we investigate the accuracy of the proposed model-based learning in terms of the resulting MDP delay value. Let $\bar{V}_{k,m_h}^t(s_{m_h}, F_{h+1}^{b,t-1})$ and $\underline{V}_{k,m_h}^t(s_{m_h}, F_{h+1}^{b,t-1})$ denote the upper and the lower bounds of the value, respectively, using $\hat{T}_{s_{m_h}s_{m_h}}^t(A_{m_h})$ in the proposed model-based learning approach in service interval t . We define ε as the $(1 - \delta)$ -confidence interval of the real MDP delay value (using the unknown $T_{s_{m_h}s_{m_h}}(A_{m_h})$ in Section III) in service interval t , i.e.

$$\text{Prob}(\bar{V}_{k,m_h}^t(s_{m_h}, F_{h+1}^{b,t-1}) - V_{k,m_h}^t(s_{m_h}, F_{h+1}^{b,t-1}) \geq \varepsilon) \leq 1 - \delta \quad (0 < \delta < 1).$$

Proposition: There exists a $(1 - \delta)$ -confidence interval ε , such that an agent m_h can update the upper bound of value $\bar{V}_{k,m_h}^t(s_{m_h}, F_{h+1}^{b,t-1})$ using

$$\bar{V}_{k,m_h}^{t+1}(s_{m_h}, F_{h+1}^{b,t}) = \min_{A_{m_h} \in \mathcal{A}_{m_h}} \left\{ E[W_{k,m_h}^t(s_{m_h}, A_{m_h})] + F_{h+1}^{b,t}(A_{m_h}) + \gamma \sum_{s_{m_h}} \hat{T}_{s_{m_h}s_{m_h}}^t(A_{m_h}) \bar{V}_{k,m_h}^t(s_{m_h}, F_{h+1}^{b,t-1}) + \varepsilon \right\}, \quad (18)$$

and update the lower bound $\underline{V}_{k,m_h}^t(s_{m_h}, F_{h+1}^{b,t-1})$ using

$$\underline{V}_{k,m_h}^{t+1}(s_{m_h}, F_{h+1}^{b,t}) = \min_{A_{m_h} \in \mathcal{A}_{m_h}} \left\{ E[W_{k,m_h}^t(s_{m_h}, A_{m_h})] + F_{h+1}^{b,t}(A_{m_h}) + \gamma \sum_{s_{m_h}} \hat{T}_{s_{m_h}s_{m_h}}^t(A_{m_h}) \underline{V}_{k,m_h}^t(s_{m_h}, F_{h+1}^{b,t-1}) - \varepsilon \right\}, \quad (19)$$

and the following two conditions are satisfied:

- 1) $n_{s_{m_h}}^t(A_{m_h}) = \left\lceil \frac{1}{2} \ln \left(\frac{|\mathcal{A}_{m_h}| |\mathcal{B}_{m_h}|}{\delta} \right) \left(\frac{V_{\max}}{\varepsilon} \right)^2 \right\rceil$, where $V_{\max} = \frac{\max_k D_{k,m_h}^{\text{rem}}}{1 - \gamma}$ represents the largest MDP delay value, $\forall A_{m_h} \in \mathcal{A}_{m_h}$.
- 2) $\underline{V}_{k,m_h}^*(s_{m_h}, F_{h+1}^b) \leq V_{k,m_h}^*(s_{m_h}, F_{h+1}^b) \leq \bar{V}_{k,m_h}^*(s_{m_h}, F_{h+1}^b)$ with probability at least $1 - 2\delta$.

Proof: See Appendix A.

This proposition shows that the estimated values $V_{k,m_h}^{t+1}(s_{m_h}, F_{h+1}^{b,t})$ become more accurate as $n_{s_{m_h}}^t(A_{m_h})$ becomes larger than $\left\lceil \frac{1}{2} \ln \left(\frac{|\mathcal{A}_{m_h}| |\mathcal{B}_{m_h}|}{\delta} \right) \left(\frac{V_{\max}}{\varepsilon} \right)^2 \right\rceil$. Moreover, the closer the agent m_h is to the destination node, the remaining path becomes shorter and provides a

smaller $V_{\max}$ and leads to a smaller requirement on $n_{s_{mh}}^t(A_{m_h})$. Hence, using the same proposed model-based learning approach to accumulate $n_{s_{mh}}^t(A_{m_h})$, the learning approach provides a more accurate MDP delay value for an agent that is closer to its destination node, which is also verified in the simulation results in Section V.D.

V. SIMULATION RESULTS

In this section, two groups of delay-sensitive applications are sent with different priorities ($K = 8$). The characteristic parameters of these delay-sensitive applications are given in Table 3.3. In the simulation, the packet length L_k is 1000 bytes for all classes. The application delay deadline is set to $D_k = 1$ seconds for all packets in different classes. We analyze the performance of our cross-layer transmission policy using the proposed distributed MDP framework in terms of the discounted end-to-end delay of the delay-sensitive applications.

TABLE 3.3. THE CHARACTERISTIC PARAMETERS OF THE DELAY-SENSITIVE APPLICATIONS.

	Group 1 delay-sensitive applications V_1				Group 2 delay-sensitive applications V_2			
C_k	C_1	C_4	C_6	C_8	C_2	C_3	C_5	C_7
λ_k	0.0170	0.0064	0.0042	0.0031	0.0105	0.0064	0.0048	0.0042
R_k (Kbps)	556	333	334	445	500	300	300	400

A. Simulation results for different network topologies

We simulate the proposed model-based reinforcement learning for solving the distributed MDP for the delay-sensitive applications in a 6-hop multi-hop network. The network topology is shown in Figure 3.5 (a) with two ASs and 18 active ARs. Group 1 delay-sensitive applications are sent through the AS m_1 to the destination node $D1$ and group 2 delay-sensitive applications are sent from the other AS to its destination node $D2$. The agents are assumed to be able to select a set of modulation and coding schemes that support a transmission rate $T = 1$ Mbps for all the transmission links in the network [Kri02]. Each receiver of the transmission links receives a random SINR x that results in a packet error rate ranging from 5% to 30%. We assume that the nodes are exchanging hello messages (as in DSDV [PB94]) with the required information exchange every 10

ms (each service interval is 10 ms). Figure 6(b) shows the MDP delay values from the ASs to the destination nodes for the first 120 service intervals. Only the results of the first five priority classes are shown. The higher priority traffic has a smaller MDP delay value $V_{k,m}^t$. The results of centralized optimization are analytically computed by assuming that the global network information is known by a central controller, which is unrealistic in practice. On the other hand, the proposed model-based reinforcement learning determines the cross-layer transmission policy at each agent based on local information. We set $\gamma = 0.75$, which is appropriate for highly time-varying multi-hop networks (after 10 service intervals, the future is only about 5% of the cost). Note that our model-based learning provides the MDP delay values close to the centralized optimization results, especially for the priority classes C_1, C_2, C_3 that satisfy the condition $E[W_{k,m_h}] \leq D_{k,m_h}^{rem}$. These three priority classes converge to a steady state after $t = 40$, since their end-to-end delays are within the delay deadline of the applications (the required performance level is set as $\sum_{t=1}^{\infty} \gamma^{t-1} D_k = \frac{D_k}{1-\gamma} = 4$ when the delay deadline of each future service interval is considered) and no packets are dropped. The results also show that the higher priority traffic converges faster than the lower priority traffic. This is because the queuing delay of the lower priority class traffic is impacted by the higher priority class traffic.

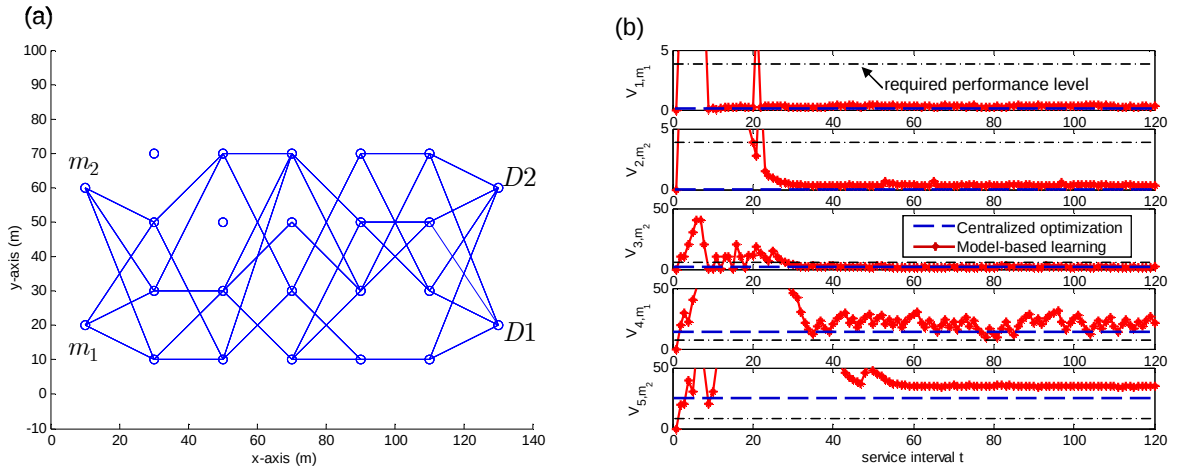


Fig. 3.5 (a) 6-hop network topology (b) MDP delay values of the first five priority classes.

Next, we simulate a skewed network topology that has two clusters of nodes shown in Figure 3.6 (a). Such network topology with clusters of nodes can be common in the multi-hop network due to landscape requirements. The network connections between the two clusters usually form a bottleneck to transmit the delay-sensitive applications. Figure 3.6 (b) shows that the discounted end-to-end delays $V_{k,m}^t$ of all the priority classes increase. Only the first two priority classes converge to the steady state results and the convergence rates decrease in the skewed network.

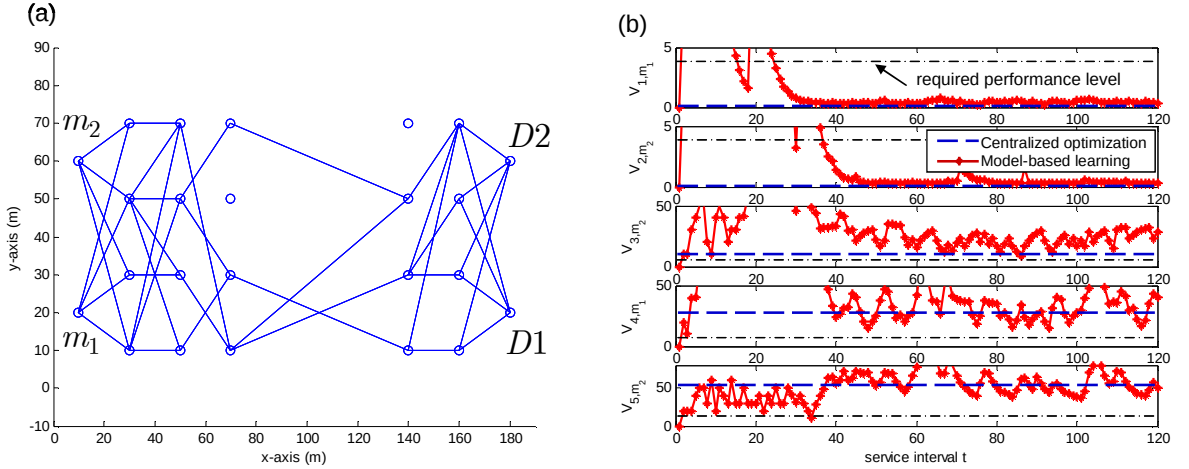


Fig. 3.6 (a) 2-cluster skewed network topology (b) MDP delay values of the first five priority classes.

B. Comparisons of the learning approaches

In this subsection, we compare the proposed model-based reinforcement learning approach with Q-learning in [WD92] (a model-free reinforcement learning approach) and the myopic self-learning approach in the previous chapter ($\gamma = 0$). We adopt the same network conditions as the previous simulations and the network topology shown in Figure 6(a). In Figure 8, the simulation results show that the proposed model-based reinforcement learning approach outperforms the other two learning approaches in terms of the MDP delay values for all the priority classes. Although Q-learning has the lowest

computational complexity, it has the worst performance in terms of both the MDP delay value $V_{k,m}^t$ and the convergence rate. The delay of the C_1 traffic converges after $t = 20$ for the proposed model-based learning approach and converges only after $t = 40$ for Q-learning approach. The convergence is not guaranteed for the lower priority class traffic, especially for the myopic self-learning solution. Moreover, although the myopic approach has the fastest convergence rate, it results in a worse performance than the proposed model-based reinforcement learning approach.

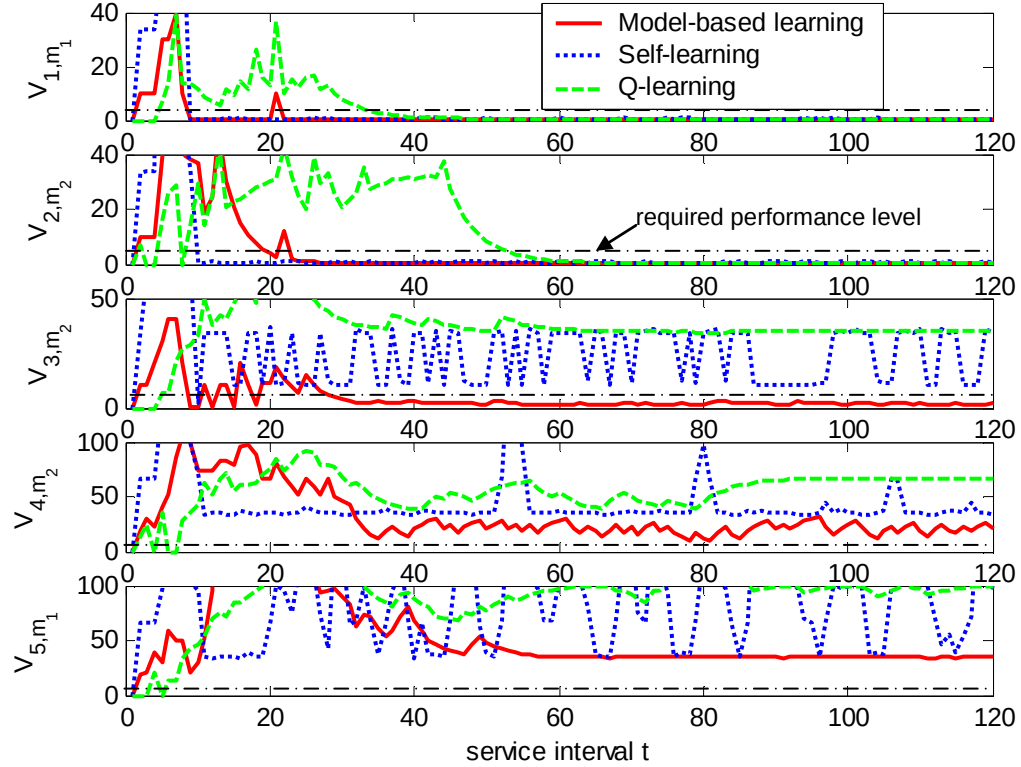


Fig 3.7. Comparisons of the MDP delay values using different learning approaches.

In addition to the discounted end-to-end delays $V_{k,m}^t$, we directly compare the undiscounted expected end-to-end delays $E[Delay_k^t]$ of the delay-sensitive applications

from the ASs to the destination nodes. The acceptance level for $E[Delay_k^t]$ is $D_k = 1$. In Figure 3.8, the simulation results show that by using the proposed model-based learning approach, the multi-hop network is able to support up to three delay-sensitive applications, since the end-to-end delay must be within the delay deadline of the applications ($E[Delay_k^t] \leq D_k$), while by using the other two learning approaches, the network can only support two delay-sensitive applications.

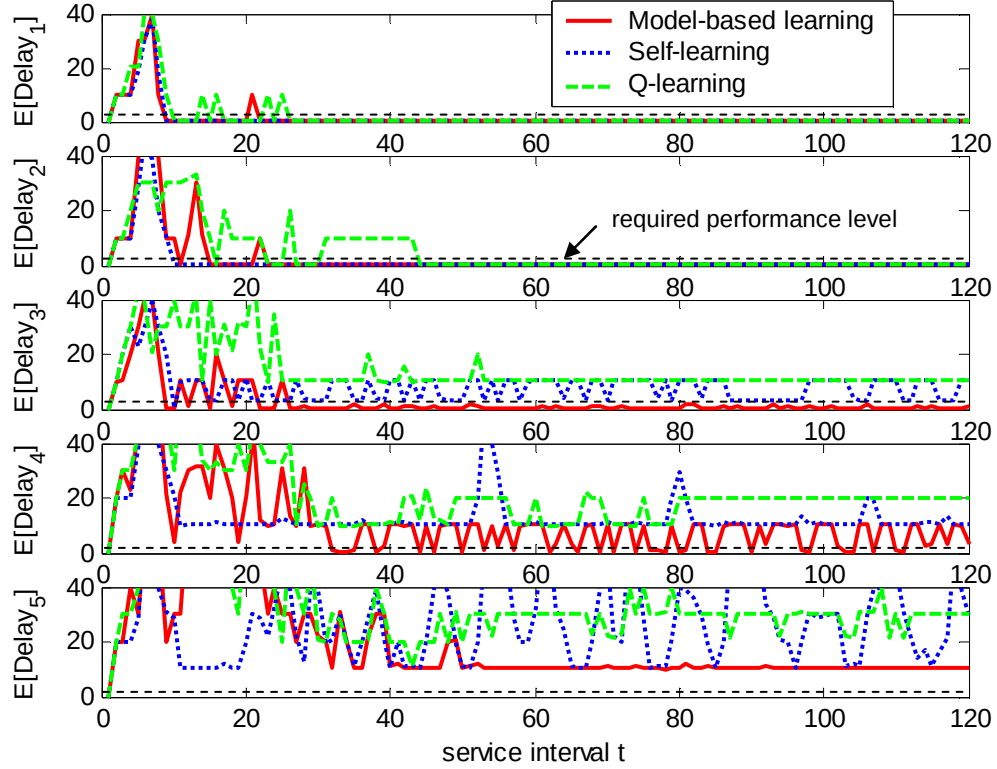


Fig 3.8. Comparisons of the expected end-to-end delay using different learning approaches.

Next, we simulate the expected delay of different classes in a source variation scenario, where the AS m_1 disappears right after service interval $t = 60$. Figure 3.9 shows the

changes of expected delays over time for different classes using various learning approaches. Since the AS m_1 is the source node of packets in classes $\{C_1, C_4, C_6, C_8\}$, the expected delays $E[Delay_1]$ and $E[Delay_4]$ in Figure 3.8 vanish after $t = 60$. We can observe that if the Q-learning is applied, before $t = 60$, only class C_1 from m_1 can be delivered in time ($E[Delay_{y_1}] \leq D_1$). However, after $t = 60$, the class C_2 from m_2 can be supported by the multi-hop network due to the alleviation of the traffic loading. By applying the proposed model-based learning approach, before $t = 60$, both class C_1, C_2 from m_1 and m_2 can be delivered in time, and after $t = 60$, not only the class C_2 but also the class C_3 from m_2 can be supported by the multi-hop network. This shows that the proposed model-based learning approach increases the capability of the agents in the multi-hop network to support more delay-sensitive applications.

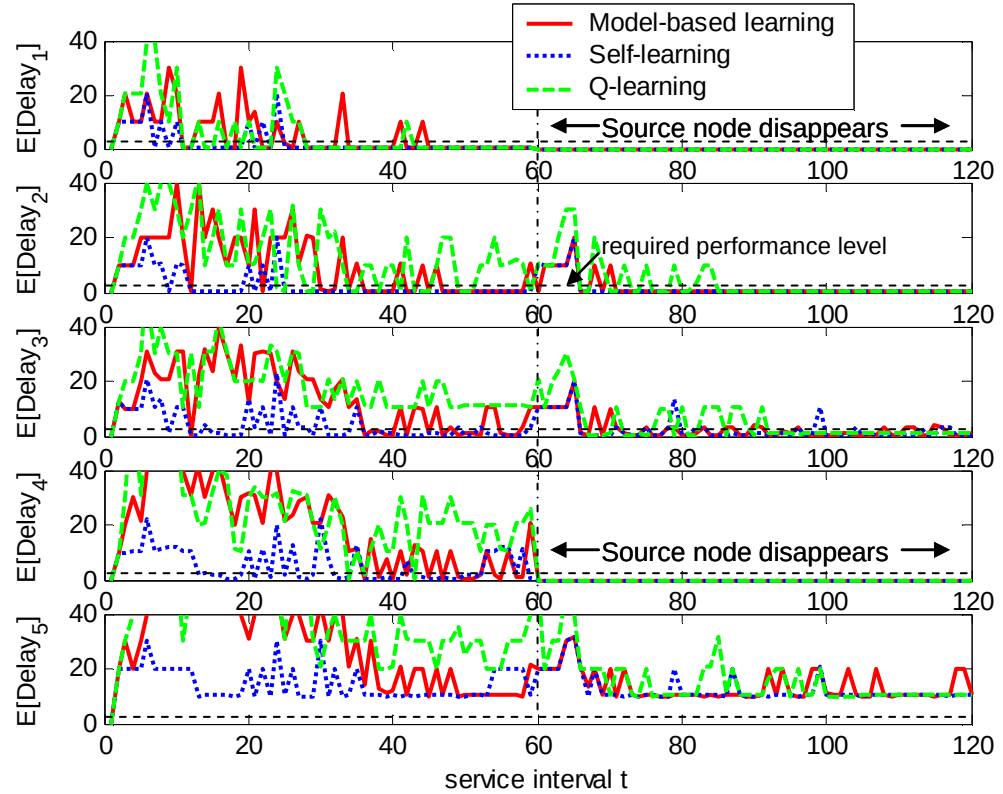


Fig 3.9. Source node of packets in class C_1, C_4 disappears after $t = 60$.

C. Heterogeneous learning

In the previous simulations, we assume that all the network nodes adopt the same learning approach to solve the distributed MDP. However in reality, the agents can adopt different learning approaches. We simulated different scenarios in which the agents have heterogeneous learning capabilities using the same network conditions as the previous simulation and the same network topology shown in Figure 3.5 (a).

TABLE 3.4 THE RESULTS OF HETEROGENEOUS LEARNING SCENARIOS.

Scenario	Learning method of the nodes within 2 hops away from ASs	Learning method of the nodes outside 2 hops away from ASs	Expected discounted end-to-end delay of the first class traffic (sec)	Expected discounted end-to-end delay of the second class traffic (sec)
1	Model-based	Model-based	0.34	0.4535
2	Model-based	Both (random)	0.3411	1.5841
3	Model-based	Model-free	0.3461	1.9785
4	Model-free	Model-based	1.5507	2.9401
5	Model-free	Both (random)	1.6886	7.4319
6	Model-free	Model-free	1.8401	7.7301

In Table 3.4, we assume that the agents in the same hop using the same learning method. The model-based learning refers to the proposed model-based reinforcement learning approach and the model-free learning refers to the Q-learning in [WD92]. The simulation results show that adopting a model-based learning approach near the ASs is very important. The discounted delays are smaller no matter what type of learning approaches the rest of the nodes adopt. This is because the model-based learning approach provides a more accurate estimate of the expected delay feedback than the model-free learning approach. Also, the model-based learning approach converges faster than the model-free learning approach. Hence, the more remaining nodes adopt the model-based learning approach, the higher the improvement in the delay performance.

Moreover, the discounted delays of the second priority class traffic vary more than the first priority class. This shows that the learning methods adopted by the agents may not impact the high priority delay-sensitive applications but can significantly impact the delay-sensitive applications with low priorities. Importantly, the learning approaches also impact the number of delay-sensitive applications supported by the multi-hop network.

D. Simulation results for the upper and the lower bounds

In this subsection, we provide simulation results to show the upper bound and the lower bound of the model-based reinforcement learning. We adopt the same network conditions and the 2-cluster network topology shown in Figure 3.6 (a). Figure 3.10 shows the MDP delay values of the first priority class traffic at different hops. Since the real delay is proven to be bounded between the upper and the lower bounds, the result shows that the model-based reinforcement learning provides end-to-end delays that are more and more accurate over time as well as when the agents are getting closer to the destination nodes.

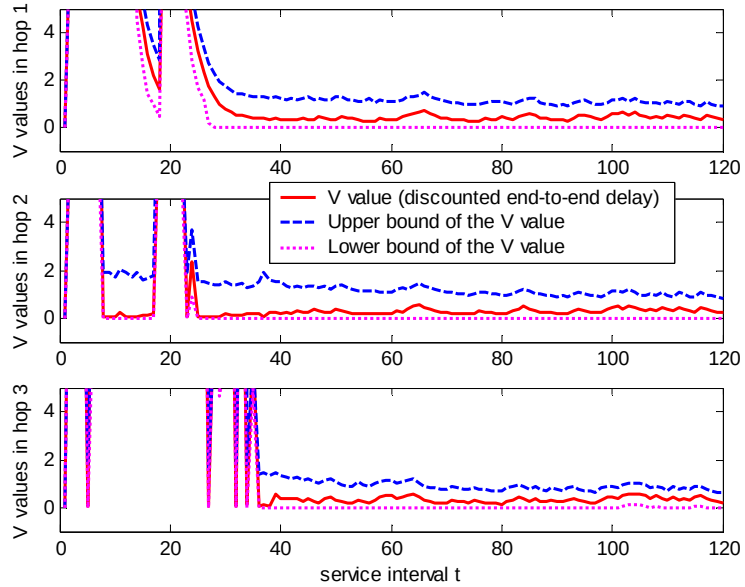


Fig. 3.10 The upper and the lower bounds of the MDP delay values for the first priority class traffic at different hops.

VI. CONCLUSIONS

In this chapter, we investigate how the agents select optimal cross-layer transmission actions in the multi-hop network to minimize the end-to-end delays of the delay-sensitive applications. To consider both the spatial and temporal dependency in the multi-hop network, we formulate the network delay minimization problem using MDP. We decompose the centralized MDP into a distributed MDP framework that is suitable for delay-sensitive applications and prove that they converge asymptotically to the same optimal policy. We propose an online model-based reinforcement learning approach for solving the distributed MDP in practice. Unlike the model-free reinforcement learning approaches, the proposed model-based reinforcement learning approach has a faster convergence rate, since it takes less time for the autonomous node to explore different states to evaluate the Q-values. Our simulation results verify that the proposed model-based learning approach is more suitable for the autonomous nodes to support delay-sensitive applications in the multi-hop network.

APPENDIX A

Proof of Lemma: For the higher priority class, if the condition $E[W_{k,m_h}] \leq D_{k,m_h}^{rem}$ holds, the packet will not be dropped in the network. Therefore, the updating equation (10) will not be impacted by the information feedforward from the agents in the previous hops. Based on this, we can write the updating equation (10) as a contraction mapping $V_{k,m_h}^{t+1} = CM(s_{m_h}^t, F_{h+1}^{b,t}, V_{k,m_h}^t)$. Define $s_h = [s_{m_h}, \forall m_h \in \mathbf{M}_h]$ as the states of the agents in the h-th hop and $V_{k,h}^t = \sum_{m_h \in \mathbf{M}_h} V_{k,m_h}^t$ as the overall MDP delay value of the agents in the h-th hop. The contraction mapping of the overall MDP delay value $V_{k,h}^{t+1} = \sum_{m_h \in \mathbf{M}_h} CM(s_{m_h}^t, F_{h+1}^{b,t}, V_{k,m_h}^t) = CM(s_h^t, F_{h+1}^{b,t}, V_{k,h}^t)$ can be constructed. Given that the feedback information $F_{h+1}^{b,t}$ is a function of the current states s_h^t , the feedback value can then be regarded as part of the cost function. Based on this, equation (10) leads to a value iteration update of a regular Bellman equation and hence, for $\forall s_h$,

$\|CM(s_h, F_{h+1}^b, V_{k,h}) - CM(s_h, F_{h+1}^b, V'_{k,h})\|_\infty \leq \gamma \|V_{k,h} - V'_{k,h}\|_\infty$ holds for the contraction mapping.

This contraction guarantees that the updating equation (10) will converge. ■

Proof of Theorem: For the higher priority class, if the condition $E[W_{k,m_h}] \leq D_{k,m_h}^{rem}$ holds, the packet will not be dropped in the network. Since the node m_{H-1} in the last hop has no information feedback value and the designation node of the class C_k packets are predetermined, the ARs in the $H-1$ -th hop converge to the value $V_{k,m_{H-1}}^*(s_{m_{H-1}})$. Moreover, the cross-layer transmission policy $\mu_{k,m_{H-1}}^t(s_{m_{H-1}})$ will also converge. The information feedback $F_{H-1}^{b,t}$ becomes only a function of the agents' states in the previous hop. Hence, the conditions in the Lemma are satisfied from the last hop to the first hop and the convergence of the distributed MDP is proven.

Assume that the AS is able to gather global information in real-time. Let $\mathbf{s}_h = [s_h, \dots, s_{H-1}]$ represent the states of all the network nodes beyond the h -th hop (denote $\mathbf{s}_0 = \mathbf{s}$) and $\mathbf{A}_h = [A_{m_h}, \forall m_h, h' = h, \dots, H-1]$ represents the cross-layer transmission actions of all the network nodes beyond the h -th hop (denote $\mathbf{A}_0 = \mathbf{A}$). From the AS m_0 's point of view, equation (7) can be rewritten as:

$$\begin{aligned} V_{k,m_0}^*(s_{m_0}, F_1^b) &= \min_{A_{m_0} \in \mathcal{A}_{m_0}} \left\{ E[W_{k,m_0}(s_{m_0}, A_{m_0})] + F_1^b(A_{m_0}) + \gamma \sum_{s_{m_0}'} T_{s_{m_0}s_{m_0}'}(A_{m_0}) V_{k,m_0}^*(s_{m_0}', F_1^b) \right\} \\ &= \min_{A_{m_0} \in \mathcal{A}_{m_0}} \left\{ E[W_{k,m_0}(s_{m_0}, A_{m_0})] + \min_{\mathbf{A}_1 \in \mathcal{A}} \left\{ F_1^b(\mathbf{s}_1, \mathbf{A}_1, A_{m_0}) + \gamma \sum_{s_{m_0}'} T_{s_{m_0}s_{m_0}'}(A_0) \underbrace{\left\{ \sum_{\mathbf{s}_1'} T_{\mathbf{s}_1\mathbf{s}_1'}(\mathbf{A}) V_{k,m_0}^*(s_0', \mathbf{s}_1') \right\}}_{V_{k,m_0}^*(s_{m_0}', F_1^b(\mathbf{s}_1, \mathbf{A}_1, A_{m_0}))} \right\} \right\} \end{aligned} \quad (20)$$

Note that the next relay m_1 will feed back the second term in the equation (20) as the expected end-to-end delay value from the next hop to the destination, i.e. $F_1^b(\mathbf{s}_1, \mathbf{A}_1, A_{m_0}) = \sum_{h=1}^{H-1} E[W_{k,m_h}(s_h, A_h)]$. The dependencies of $F_1^b(\mathbf{s}_1, \mathbf{A}_1, A_0)$ address the fact that the expected delay from the next hop to the destination node only depends on the states and actions $(\mathbf{s}_1, \mathbf{A}_1)$. Denote the last term of equation (20) as $V_{k,m_0}(s_{m_0}', F_1^b) = \sum_{\mathbf{s}_1'} T_{\mathbf{s}_1\mathbf{s}_1'}(\mathbf{A}) V(s_{m_0}', \mathbf{s}_1')$ to represent the MDP delay value at the AS m_0 . Then, equation (20) can be equivalently rewritten as

$$V_{k,m_0}^*(s) = \min_{\mathbf{A} \in \mathcal{A}} \left\{ E[\text{Delay}_k(s, \mathbf{A})] + \gamma \sum_{s'} T_{ss'}(\mathbf{A}) V_{k,m_0}^*(s') \right\}. \quad (21)$$

We denote the Q-value of taking a cross-layer transmission action $\mathbf{A}$ at the state s as $Q_{k,m_0}(s, \mathbf{A}) = E[\text{Delay}_k(s, \mathbf{A})] + \gamma \sum_{s'} T_{ss'}(\mathbf{A}) V_{k,m_0}^*(s')$. We define the centralized stationary policy as $\mu_k(s) = \arg \min_{\mathbf{A} \in \mathcal{A}} Q_{k,m_0}(s, \mathbf{A})$.

Note that for the agent m_h , the cross-layer transmission policy $\mu_{k,m_h}^d(s_{m_h})$ minimizes $\sum_{h'=h}^{H-1} \sum_{t=0}^{\infty} r^t E[W_{k,m_h}^t(s_{m_h}^t, \mu_{k,m_h}^d(s_{m_h}^t))]$ $= \sum_{t=0}^{\infty} r^t \sum_{h'=h}^{H-1} E[W_{k,m_h}^t(s_{m_h}^t, \mu_{k,m_h}^d(s_{m_h}^t))]$. Due to the principle of the optimality [Ber95] the relay nodes selected from $\mu_{k,m_h}^d(s_{m_h})$ must also lie on the shortest route specified by the $\mu_k(s)$. From equation (20), we can conclude that $\sum_{t=0}^{\infty} r^t \sum_{h'=0}^{H-1} E[W_{k,m_h}^t(s_{m_h}^t, \mu_{k,m_h}^d(s_{m_h}^t))] = \sum_{t=0}^{\infty} r^t E[\text{Delay}_k^t(s^t, \mu_k(s^t))]$. Hence, the distributed MDP solution $[\mu_{k,m_h}^d(s_{m_h}), h = 0, \dots, H-1]$ converges to the same policy of the centralized MDP.

For the necessary condition, if the condition $E[W_{k,m_h}] \leq D_{k,m_h}^{\text{rem}}$ does not hold for every node in the route σ_k , this results in an infinite $E[W_{k,m_h}]$ at the node m_h and an infinite feedback value $F_h^{b,t+1}$ to notify the AS to reroute the packets and no convergence can be guaranteed in this case. ■

Proof of Proposition: We apply Hoeffding inequality [Hoe63] to obtain the confidence interval ε , which basically states that given random variables $\{X_1, \dots, X_m\}$ in range $[0, X_{\max}]$, the inequality holds:

$$\text{Prob}\left(\frac{1}{m} \sum_{i=1}^m X_i - \frac{1}{m} \sum_{i=1}^m E[X_i] \geq \varepsilon\right) \leq e^{-2m \left(\frac{\varepsilon}{X_{\max}}\right)^2}. \quad (22)$$

From the first condition, we have $\varepsilon = V_{\max} \sqrt{\frac{-\ln\left(\frac{\delta}{|\mathcal{A}_{m_h}| |\mathcal{E}_{m_h}|}\right)}{2n_{s_{m_h}}^t(A_{m_h})}}$. Denote

$E[\bar{V}(s_{m_h}, A_{m_h})] = \sum_{s_{m_h}'} \hat{T}_{s_{m_h} s_{m_h}'}^t(A_{m_h}) \bar{V}_{k,m_h}^t(s_{m_h}', F_{h+1}^{b,t-1})$ as the average MDP delay upper bound based on estimated $\hat{T}_{s_{m_h} s_{m_h}'}^t(A_{m_h})$ evaluated whenever state s_{m_h} is visited and action A_{m_h} is taken, and denote $E[V(s_{m_h}, A_{m_h})] = \sum_{s_{m_h}'} T_{s_{m_h} s_{m_h}'}^t(A_{m_h}) V_{k,m_h}^t(s_{m_h}', F_{h+1}^{b,t-1})$ as the average expected MDP delay value based on real $T_{s_{m_h} s_{m_h}'}^t(A_{m_h})$. Similar to the proof of lemma 3.2 in [EMM03], equation (22) can be rewritten as:

$$\text{Prob}(E[\bar{V}(s_{m_h}, A_{m_h})] - E[V(s_{m_h}, A_{m_h})] \geq \varepsilon) \leq e^{-2n_{s_{m_h}}^t(A_{m_h}) \left(\frac{1}{V_{\max}} \right)^2 \left(V_{\max} \sqrt{\frac{-\ln\left(\frac{\delta}{|\mathcal{A}_{m_h}| |\mathcal{B}_{m_h}|}\right)}{2n_{s_{m_h}}^t(A_{m_h})}} \right)^2} = \left(\frac{\delta}{|\mathcal{A}_{m_h}| |\mathcal{B}_{m_h}|} \right). \quad (23)$$

Hence, $\text{Prob}(\bar{V}_{k,m_h}^{t+1}(s_{m_h}, F_{h+1}^{b,t}) - V_{k,m_h}^{t+1}(s_{m_h}, F_{h+1}^{b,t}) \geq \varepsilon) \leq \delta$ for each state-action pair (the total number of the state-action pairs is $|\mathcal{A}_{m_h}| |\mathcal{B}_{m_h}|$). Similar proof can be applied to the lower bound. Since $n_{s_{m_h}}^t(A_{m_h})$ in the last term of equations (18) and (19) goes to infinity as $t \rightarrow \infty$, we can show that both the upper bound and the lower bound converge under the same conditions, i.e. $\bar{V}_{k,m_h}^*(s_{m_h}, F_{h+1}^b) = \lim_{t \rightarrow \infty} \bar{V}_{k,m_h}^t(s_{m_h}, F_{h+1}^{b,t})$ and $V_{k,m_h}^*(s_{m_h}, F_{h+1}^b) = \lim_{t \rightarrow \infty} V_{k,m_h}^t(s_{m_h}, F_{h+1}^{b,t})$. Due to the symmetric structure of $\bar{V}_{k,m_h}^*(s_{m_h}, F_{h+1}^b)$ and $V_{k,m_h}^*(s_{m_h}, F_{h+1}^b)$, we apply the union bound as in [EMM03] to show that the probability $\text{Prob}(|\bar{V}_{k,m_h}^*(s_{m_h}, F_{h+1}^b) - V_{k,m_h}^*(s_{m_h}, F_{h+1}^b)| \geq \varepsilon) \leq 2\delta$ and complete the proof. ■

APPENDIX B

Algorithm 3.1: Model-free reinforcement learning at node m_h

Input: $\{F_{h+1}^{b,t}, \forall t\}$, $\{F_{h-1}^{f,t}, \forall t\}$, γ , τ ; **Output:** μ_{k,m_h}^t , $F_h^{b,t}$, $F_h^{f,t}$;

Initialization: μ_{k,m_h}^0 , $F_{h+1}^{b,0}$, $F_{h-1}^{f,0}$, $s_{m_h}^0$;

set $t \leftarrow 0$, $Q_{k,m_h}^0(s_{m_h}, A_{m_h}) = 0, \forall s_{m_h} \in \mathcal{X}_{m_h}, \forall A_{m_h} \in \mathcal{A}_{m_h}$;

Step 1: Verify the head of line packet class and the delay deadline.

Get the class C_k packet that has the highest priority in the queue, and check the packet header.

If $Delay_{k,h-1}^{PASS} > D_k$, drop the packet and repeat step 1, $t \leftarrow t + 1$; otherwise, go to step 2.

Step 2: Select an action $A_{m_h}^t$ based on policy μ_{k,m_h}^t .

Randomly select the action $A_{m_h}^t$ according to the probability distributions

$[\pi_{k,m_h}^t(s_{m_h}^t, A_{m_h}), \forall A_{m_h} \in \mathcal{A}_{m_h}]$.

Step 3: Transmit the packet and observe the current cost $Cost_{k,m_h}^t$ and the new state $s_{m_h}^{t+1}$

Step 4: Update the Q-value

For $\forall s_{m_h} \in \mathcal{X}_{m_h}$, update the Q-values $Q_{k,m_h}^{t+1}(s_{m_h}, A_{m_h}^t)$ using equation (13).

Step 5: Update the policy

For $\forall s_{m_h} \in \mathcal{X}_{m_h}$, update the policy $\mu_{k,m_h}^{t+1}(s_{m_h})$ using equation (14).

Step 6: Update the feedback values and exchange information with the neighboring nodes.

Update $F_h^{b,t+1}$ and $F_h^{f,t+1}$ as in equation (11) and (12).

$t \leftarrow t + 1$; go back to step 1.

Algorithm 3.2: Model-based reinforcement learning at node m_h

Input: $\{F_{h+1}^{b,t}, \forall t\}, \{F_{h-1}^{f,t}, \forall t\}, \gamma, \tau$; **Output:** $\mu_{k,m_h}^t, F_h^{b,t}, F_h^{f,t}$;

Initialization: $\mu_{k,m_h}^0, F_{h+1}^{b,0}, F_{h-1}^{f,0}, s_{m_h}^0$;

set $t \leftarrow 0, Q_{k,m_h}^0(s_{m_h}, A_{m_h}) = 0, \forall s_{m_h} \in \mathcal{X}_{m_h}, \forall A_{m_h} \in \mathcal{A}_{m_h}$;

Step 1: Verify the head of line packet class and the delay deadline.

Get the class C_k packet that has the highest priority in the queue, and check the packet header.

If $Delay_{k,h-1}^{PASS} > D_k$, drop the packet and repeat step 1, $t \leftarrow t + 1$; otherwise, go to step 2.

Step 2: Select an action $A_{m_h}^t$ based on policy μ_{k,m_h}^t .

Randomly select the action $A_{m_h}^t$ according to the probability distributions $[\pi_{k,m_h}^t(s_{m_h}^t, A_{m_h}^t), \forall A_{m_h}^t \in \mathcal{A}_{m_h}]$.

Step 3: Transmit the packet, observe the new state $s_{m_h}^{t+1}$, and update the number of state transition.

$$n_{s_{m_h}^t s_{m_h}^{t+1}}^{t+1}(A_{m_h}^t) = n_{s_{m_h}^t s_{m_h}^{t+1}}^t(A_{m_h}^t) + 1; \quad n_{s_{m_h}^t}^t(A_{m_h}^t) = \sum_{s_{m_h}^{t+1} \in \mathcal{X}_{m_h}} n_{s_{m_h}^t s_{m_h}^{t+1}}^t(A_{m_h}^t).$$

$$\hat{T}_{s_{m_h}^t s_{m_h}^{t+1}}^t(A_{m_h}^t) = \frac{n_{s_{m_h}^t s_{m_h}^{t+1}}^t(A_{m_h}^t)}{n_{s_{m_h}^t}^t(A_{m_h}^t)}.$$

Step 4: Evaluate the local queuing delay

Calculate $E[W_{k,m_h}^t(s_{m_h}^t, A_{m_h}^t)]$ using equation (16).

Step 5: Update the Q-value

For $\forall s_{m_h} \in \mathcal{B}_{m_h}$, update the Q-values $Q_{k,m_h}^{t+1}(s_{m_h}, A_{m_h}^t)$ using equation (15).

Step 6: Update the policy

For $\forall s_{m_h} \in \mathcal{B}_{m_h}$, update the policy $\mu_{k,m_h}^{t+1}(s_{m_h})$ using equation (14).

Step 7: Update the feedback values and exchange information with the neighboring nodes.

Update $F_h^{b,t+1}$ and $F_h^{f,t+1}$ as in equation (11) and (12).

$t \leftarrow t + 1$; go back to step 1.

Chapter 4

Adapting the Information Horizon – Risk-Aware Scheduling for Multimedia Streaming

I. INTRODUCTION

As discussed in the previous chapter, the majority of the multimedia-centric research focuses on a centralized optimization and optimizes the video streaming using purely end-to-end metrics and does not consider the protection techniques available at the lower layers of the protocol stack. Hence, they do not take advantage of the significant gains provided by cross-layer design [SYZ05][BT05][WCZ05]. In [AMV06], an integrated cross-layer optimization framework was proposed that considers the video quality impact based on different information horizons. However, the proposed solution in [AMV06] considers only the single user case, where a set of paths and transmission opportunities are statically pre-allocated for each video application. This leads to a sub-optimal, non-scalable solution for the multi-user case. Importantly, the overhead induced by the various information horizons are not investigated in [AMV06], which have essential impact for the delay-sensitive multimedia applications. To enable efficient distributed multi-user video streaming over a wireless multi-hop infrastructure, nodes need to timely collect and disseminate network information based on which, the various nodes can collaboratively adapt their cross-layer transmission strategies. For instance, based on the available information feedback, a network node can timely choose an alternate (less congested) route for streaming the packets that have a higher contribution to the overall distortion or a more imminent deadline.

Although the information feedback is essential to the cross-layer optimization, the cost

of collecting the information is seldom discussed in the literature. Due to the informationally decentralized nature of the multi-hop wireless network, it is impractical to assume that the global network information and the time-varying application requirements can be relayed to the central (overlay) network manager in a timely manner. Distributed suboptimal solutions that adapt the transmission strategies based on well-designed localized information feedback should be adopted for the delay-sensitive applications.

In summary, no integrated framework has been developed that explicitly considers the impact of accurate and frequent network information feedback from various horizons, when optimizing the resource allocation and the cross-layer transmission strategies for multiple collaborating users streaming real-time multimedia over a wireless multi-hop network. In this chapter, we investigate the impact of this information feedback on the distributed cross-layer transmission strategies deployed by the multiple video users. We assume a directed acyclic overlay network as the previous chapters that can be superimposed over any wireless multi-hop network to convey the information feedback. Our solution relies on the users' agreement to collaborate by dynamically adapting the quality of their multimedia applications to accommodate the flows/packets of other users with a higher quality impact and/or higher probability to miss their decoding deadlines. Unlike commercial multi-user systems, where the incentive to collaborate is minimal, we investigate the proposed approach in an enterprise network setting where source and relay nodes exchange accurate and trustable information about their applications and network statistics.

To increase the number of users that can simultaneously share the same wireless multi-hop infrastructure as well as to improve their performance given time-varying network conditions, we deploy scalable video coding schemes [VAH06] that enable a fine-granular adaptation to changing network conditions and a higher granularity in assigning the packet priorities. We assume each receiving node performs polling-based

contention-free media access control (MAC) [IEE03] that dynamically reserves a transmission opportunity interval in a service interval. The network topology and the corresponding channel condition of each link are assumed to remain unchanged within the service interval.

In this chapter, we discuss the required information/parameter exchange among network nodes/layers for implementing a distributed solution for selecting the following cross-layer transmission strategies at each intermediate node – the packet scheduling, and the next-hop relay (node) selection based on routing policies similar to the Bellman-Ford routing algorithm [BG87], and the retransmission limit at the MAC layer. In performing the cross-layer adaptation, we explicitly consider the packet deadlines and the relative priorities (based on the quality impact of the packets) encapsulated in the packet headers. Each intermediate node maintains a queue of video packets from various users and determines the cross-layer transmission strategies in a distributed fashion through the information feedback from other intermediate nodes within a certain network horizon and with a certain frequency. While a larger horizon/frequency can provide more accurate network information, this also results in an increased transmission overhead that can have a negative impact on the video performance. Hence, we aim at quantifying the video quality benefit derived by the various users for different network conditions and video application characteristics based on various information feedbacks.

Our chapter makes the following contributions:

- **Decentralized information feedback driven cross-layer adaptation**

In this chapter, we show how the various cross-layer strategies can be adapted based on the information feedback. The solutions of centralized flow-based optimizations [WZ02][SYZ05][AL94] have several limitations. First, the video bitstreams are changing over time in terms of required rates, priorities and delays. Hence, it is difficult to timely allocate the necessary bandwidths across the wireless network infrastructure to match these time-varying application requirements. Second, the delay constraints of the various

packets are not explicitly considered in centralized solutions, as this information cannot be relayed to a central resource manager in a timely manner. Third, the complexity of the centralized approach grows exponentially with the size of the network and number of video flows. Finally, the channel characteristics of the entire network (the capacity region of the network) need to be known for this centralized, oracle-based optimization. This is not practical as channel conditions are time-varying, and having accurate information about the status of all the network links is not realistic.

Alternatively, we focus on a fully distributed packet-based solution, where timely information feedback can efficiently drive the cross-layer adaptation for each individual multimedia stream as well as the multi-user collaborations in sharing the wireless infrastructure. To cope with the delay sensitivity of the video traffic, we explicitly consider the delay deadlines of the various packets (packets are dropped whenever their deadlines expire) and estimate the remaining transmission time based on the available information feedback. This approach is better suited for the informationally decentralized nature of the investigated multi-user video transmission problem over multi-hop infrastructures.

- **Impact of various information horizons/frequencies**

We define the mechanism of information feedback conveyed through a multi-hop overlay infrastructure and investigate the impact of different information horizons/frequencies on the video quality derived by the various multimedia users. We discuss the tradeoff between the increased transmission overhead and the benefit of larger information horizons, which result in improved predictions of network conditions. More information allows nodes in the network to better estimate the time for each packet to reach its destination and hence, the chance of missing its deadline.

- **Information feedback driven packet scheduling and retransmission strategies**

We introduce the concept of *risk estimation* based on the available information feedback that determines the probability that a packet will miss its delay deadline. Based

on the estimated risk and the quality impact of the video packet, we proposed novel information feedback driven scheduling and retransmission strategies for each node in the network.

The chapter is organized as follows. Section II defines the video and network specification for multi-user video transmission over multi-hop wireless networks and provides a cross-layer distributed optimization scheme based on the information feedback. In Section III, we discuss the impact of the information feedback with different information horizons and present an integrated cross-layer adaptation algorithm for the real-time multi-user streaming problem. Section IV introduces a novel information feedback driven scheduling algorithm that takes advantage of the larger information horizons. Section V introduces our information feedback driven retransmission limit calculation. In Section VI, we discuss the overheads of the information feedback of various parameters. Simulation results are given in Section VII. Section VIII concludes the chapter.

II. PROBLEM FORMULATION AND SYSTEM DESCRIPTION

We assume that V video users with distinct source and destination nodes are sharing the same multi-hop wireless infrastructure. Similarly, we adopt an embedded 3D wavelet codec [AMB04] as Chapter 2 and construct video classes by truncating the embedded bitstream (see Chapter 3.II.A). Here we define that N_k represents the number of packets in the class f_k in one GOP duration of the corresponding video sequence and hence, $R_k = N_k L_k$ represents the rate requirement.

At the client side, the expected quality improvement for video v in one GOP can be expressed as:

$$Q_v^{rec} = \sum_{f_k \in v} (\lambda_k \cdot L_k) \cdot (N_k \cdot P_k^{succ}), \quad (1)$$

Here, we assume that the client implements a simple error concealment scheme, where the lower priority packets are discarded whenever the higher priority packets are lost

[VT07]. Recall that the end-to-end probability P_k^{succ} depends on the network resource, competing users' priorities as well as the deployed cross-layer transmission strategies. In addition, at the intermediate node m , we assume that the video packets are scheduled in a specific order π_m according to the prioritization associated with the video content characteristics.

A. Overlay network specification

We assume the same directed acyclic multi-hop wireless network as in Figure 2.1. Importantly, note that the deployed structure is very general and various multi-hop network that can be modeled as a directed acyclic graph can be modified to fit into this overlay structure by simply adding virtual nodes (virtual hops for different users) [EM93]. We introduce virtual nodes with zero service time for users that have a smaller number of hops, and fix the path for particular classes to pass through the virtual node (by enforcing $\beta_{k,m_h,m_{h+1}}$). Figure 4.1 gives an example of a 3-hop overlay network with two users ($V = 2, H = 3, M_0 = M_3 = 2, M_1 = 4, M_2 = 2$). Methods to construct such overlay structures given a specific multi-hop network and a set of transmitting-receiving pairs can be found in [WR03][Jan02]. Through the multi-stage overlay infrastructure, the information feedback is performed from the intermediate nodes to all the connected nodes ($\beta_{k,m_h,m_{h+1}} \neq 0$) in the previous hop.

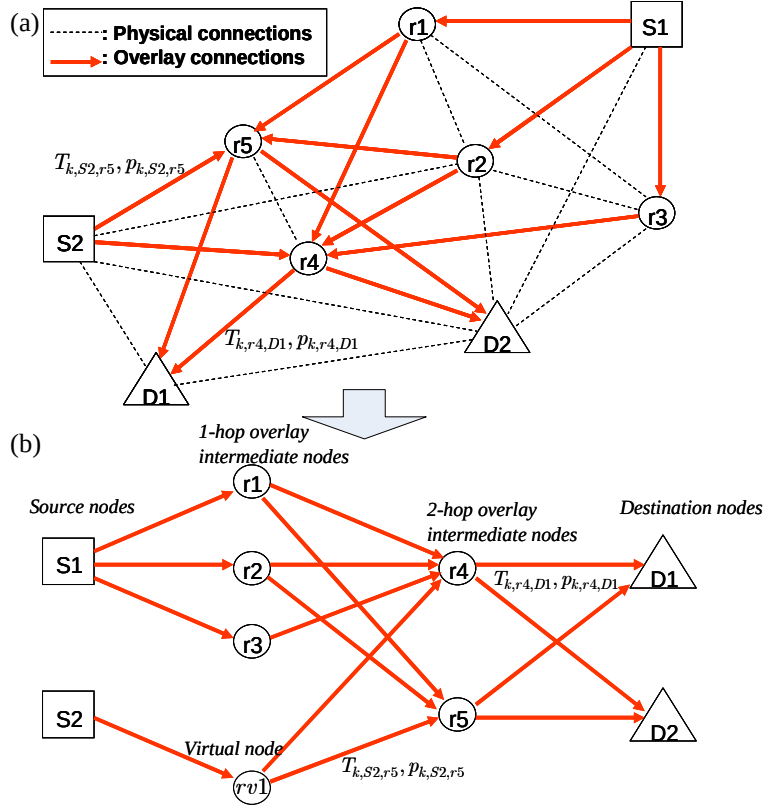


Fig. 4.1. The directed acyclic multi-hop overlay network for an exemplary wireless infrastructure.
(a) Actual network topology that has 2 source-destination pairs, 5 relay nodes. (b) Overlay network topology that has 2 source-destination pairs, 6 relay nodes (with one virtual node in the 1-hop intermediate nodes).

B. Centralized cross-layer optimization for multi-user wireless video transmission

We define STR_{m_h} as the cross-layer transmission strategy vector for packets at the node m_h consisting of the packet scheduling policy π_{m_h} , the relay selecting parameters $\beta_{k,m_h,m_{h+1}}$ for routing, the MAC retransmission limit $\gamma_{k,m_h,m_{h+1}}^{MAX}$ per link, i.e. $STR_{m_h} = [\pi_{m_h}, \beta_{k,m_h,m_{h+1}}, \gamma_{k,m_h,m_{h+1}}^{MAX}] \in \mathcal{A}_{tot}$. And $\mathcal{A}_{tot} = \mathcal{A}_{APP} \times \mathcal{A}_{NET} \times \mathcal{A}_{MAC}$ represents the set including all the feasible cross-layer transmission strategy vector, where $\mathcal{A}_{APP}$ is the set of all feasible packet scheduling strategies, $\mathcal{A}_{NET}$ is the set of all possible selections of relays, and $\mathcal{A}_{MAC}$ is an integer set from 0 to the maximum retransmission limit supported by the MAC protocol. Then, assuming the global information $\mathcal{I}_{global}$ is available, the investigated multi-user wireless video transmission problem can be

formulated as a centralized delay-driven cross-layer optimization:

$$\begin{aligned} \mathbf{STR}^{opt} &= \arg \max_{\mathbf{STR} \in \mathcal{A}_{tot}^{|\mathbf{M}|}} \sum_{v=1}^V Q_v^{rec}(\mathbf{STR}, \mathcal{I}_{global}) \\ &= \arg \max_{\mathbf{STR} \in \mathcal{A}_{tot}^{|\mathbf{M}|}} \sum_{v=1}^V \sum_{f_k \in v} \lambda_k L_k N_k P_k^{succ}(\mathbf{STR}, \mathcal{I}_{global}) \end{aligned} \quad (2)$$

where $\mathbf{STR} = [STR_{m_h} \mid m_h \in \mathbf{M}]$, and $\mathbf{M}$ represents the set of nodes at which the transmission strategies decisions can be made for the video packets. $|\mathbf{M}| = \sum_{h=0}^{H-1} M_h$ is the number of the nodes in $\mathbf{M}$. Since the successfully received packets of each class f_k must have their end-to-end delay D_k smaller than their corresponding delay deadline d_k , the constraint of the optimization is $D_k(\mathbf{STR}) < d_k, k = 1, \dots, K$. Due to the priority queuing and the error concealment scheme, the optimal solution of equation (2) serves the more important packets instead of transmitting as many packets as possible. Although the centralized optimization provides optimal solution for the multi-user video streaming problem, it suffers from the unrealistic assumption of collecting timely global information across the multi-hop network for the delay-sensitive applications. Due to the informationally decentralized nature of the multi-hop wireless networks, the centralized solution is not practical for the multi-user video streaming problem. For instance, the optimal solution depends on the delay incurred by the various packets across the hops, which cannot be timely relayed to a central controller. Moreover, the complexity of the centralized optimization grows exponentially with number of classes and nodes in the network. Hence, the optimization might require a large amount of time to process and the collected information might no longer be accurate by the time transmission decisions need to be made.

C. Proposed distributed cross-layer adaptation based on information feedback

Instead of gathering the global information $\mathcal{I}_{global}$, we propose a distributed suboptimal solution that collects the local information feedback $\mathcal{I}_{local}$ at the node m_h to maximize the expected quality of the various users sharing the same multi-hop wireless infrastructure:

$$STR_{m_h}^{opt} = \arg \max_{STR \in \mathcal{A}_{tot}} \sum_{\forall f_k \text{ at } m_h} \lambda_k L_k N_{k,m_h} E[P_k^{succ}(STR_{m_h}, \mathcal{I}_{local})], \quad (3)$$

where N_{k,m_h} represents the number of packets of class f_k present in the queue at the node m_h .

In this chapter, we define $\mathcal{I}_{local}$ with the following information feedback parameters:

- $SINR$, the SINR to calculate the channel conditions over each link of the overlay network.
- P_{k,m_h} , the packet loss probability of the class f_k through the intermediate node m_h . The parameter illustrates the bottleneck identification for various video classes. This information can be used by the application layer to decide how many quality layers are transmitted or to adapt its encoding parameters (in the case of real-time encoding) to improve its video quality performance given the current number of users, priorities of the competing streams and network conditions, but also, importantly, to alleviate the network congestion.
- $E[Delay_{k,m_h}]$, the expected delay from the intermediate node m_h to the destination node of the class f_k to convey the congestion information of the network, which is essential for the delay-sensitive applications.

Let us consider the simple example in Figure 4.2 that illustrates how information feedback is deployed. The term information horizon will be defined in Section III. In this example, node n1 is an intermediate node that needs to relay multiple video classes from various users. In order for the relay n1 to determine the optimized cross-layer transmission strategies, at least 1-hop information feedback is required. The network status information can be disseminated at frequent intervals over the overlay infrastructure, and it is considered to be known at the decision relay n1. However, in certain cases, feedback information from some hops (beyond the information horizon) may arrive with an intolerable delay, and may be unreliable due to the rapidly-changing network conditions.

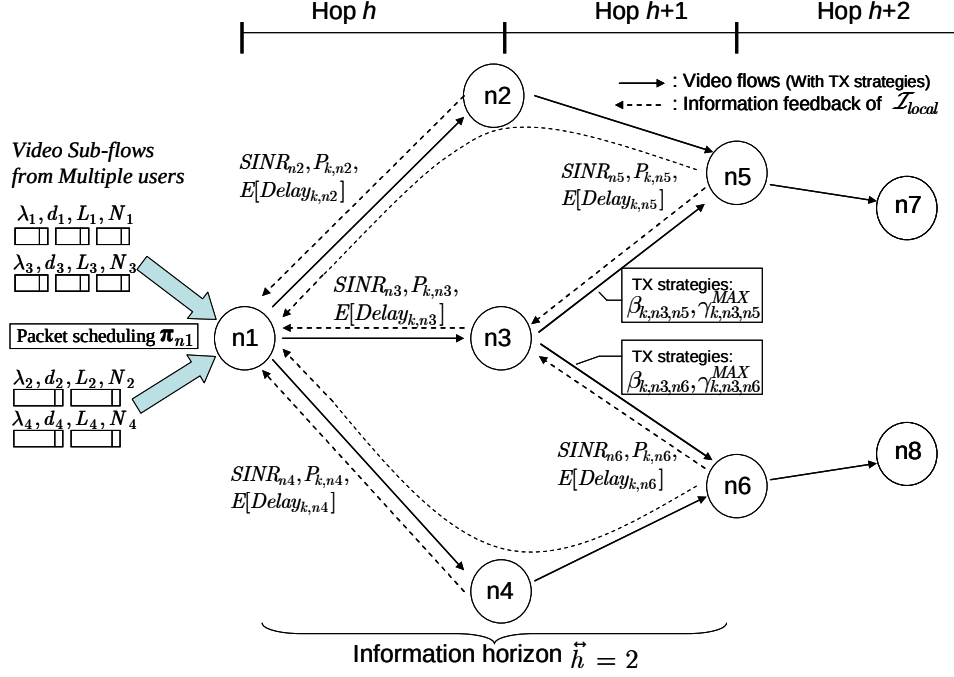


Fig. 4.2. Illustrative example of an application layer overlay network with information horizon $\vec{h} = 2$.

In this chapter, we make the following assumptions for performing the information feedback and the delay estimation $E[Delay_{k, m_h}]$. First, we assume a polling-based contention-free media access (which is similar to the deployed IEEE 802.11e [IEE03] and 802.11s [FWK06] standards) that dynamically reserves transmission opportunities within a service interval t_{SI} [IEE03], and the network status (such as the topology, the transmission rate $T_{k, m_h, m_{h+1}}$ and the packet error rate $p_{k, m_h, m_{h+1}}$ for each link) remains unchanged in t_{SI} . Second, because of the retransmission in the MAC layer protection, the effective packet transmission time can be formulated as a geometric distribution [Kon80] with $T_{k, m_h, m_{h+1}}$, $p_{k, m_h, m_{h+1}}$, and packet length L_k (as discussed in Section III.B). Third, for simplification, the arrival of the packets at each intermediate node is regarded as a Poisson arrival process, which is reasonable if the number of intermediate nodes is large enough and the selection of paths is relatively balanced. Fourth, we assume that the queue waiting time dominates the overall delay. Under these assumptions, we can estimate the *risk* that packets from different priority classes will not arrive at their

destination before their decoding deadline expires (see Section IV for more detail). The adaptation of π_{m_h} , $\gamma_{k,m_h,m_{h+1}}^{MAX}$, and the dynamic routing policies for $\beta_{k,m_h,m_{h+1}}$ can be deployed in a distributed manner based on the information feedback. Next, we discuss the mechanism of performing the information feedback through the directed acyclic overlay network.

III. IMPACT OF ACCURATE NETWORK STATUS

Since the network conditions can rapidly vary in multi-hop network infrastructures, the performance of any video streaming solution will significantly depend on the availability of accurate network information. Three key aspects for multi-user video streaming are influenced by the availability, accuracy and timeliness of this information feedback.

- *Decentralized decision making* - network nodes can be implemented to improve their adopted cross-layer strategies based on information feedback about the channel conditions and regional network congestion to avoid unnecessary queuing delay and hence, packet drops.
- *Timely adaptation* - information feedback enables timely adaptation to network changes (e.g. nodes leaving or sources of interference appearing or disappearing), which is essential for delay-sensitive multimedia transmission.
- *Inter-user collaboration* - based on information feedback, network resources can be effectively managed and users are able to effectively collaborate to achieve the desired global optimal utility. For instance, in the absence of such information, an intermediate node may waste precious resources by allocating time to packets from classes that will miss their deadlines, thereby preventing other classes which can meet their delay constraint from being transmitted.

A. Information feedback frequencies and information horizon

The information feedback should be performed in a distributed (per hop) fashion that explicitly considers the dissemination delay. We assume that the information feedback is

periodically transmitted to the previous hop every t_{info}^1 seconds during each t_{SI} ($0 < t_{\text{info}} \leq t_{SI}$). We define $f_{\text{info}}(1)$ as the frequency of the information feedback within one hop:

$$f_{\text{info}}(1) = \frac{1}{t_{\text{info}}}. \quad (4)$$

We also define the vector $\mathbf{b} = (b_1, b_2, \dots, b_H)$ of the dissemination factors over the network. Let $t_{\text{info}}(h)$ represent the time it takes for the information to be disseminated over h hops:

$$t_{\text{info}}(h) = b_h \times t_{\text{info}}, \text{ where } b_h \geq 1, \text{ for } h \geq 1. \quad (5)$$

Since the network information requires time to pass through the various hops, we have $b_h > b_{h-1}$. We set $b_1 = 1$. Because the information is conveyed hop by hop, $t_{\text{info}}(h)$ also depends on the per-hop information feedback frequency $f_{\text{info}}(1)$. We define $f_{\text{info}}(h)$ as the information feedback frequency when the information is conveyed over h hops in the following way:

$$f_{\text{info}}(h) = \frac{1}{t_{\text{info}}(h)} = \frac{c}{t_{SI} \times b_h}, \quad (6)$$

where c is defined as t_{SI}/t_{info} . Since the network conditions are assumed to be unchanged within the service interval t_{SI} , we define the information horizon $\vec{h}$ as the number of hops from which the information feedback can be accurately disseminated during t_{SI} :

$$\begin{aligned} \vec{h}(\mathbf{b}, f_{\text{info}}(1)) &= \text{maximize } h \\ \text{subject to } t_{\text{info}}(h) &\leq t_{SI}, \\ h &= 1, \dots, H. \end{aligned} \quad (7)$$

In [AMV06][Kri02], the dissemination time for the information feedback is proportional to the number of hops across which the information feedback is traversed, i.e. $\mathbf{b}_p = (1, 2, 3, \dots, H-1, H)$, and if we assume that c is an integer, the relationship between $\vec{h}$ and $f_{\text{info}}(1)$ becomes a linear function:

¹ The time interval is not the time fraction for transmitting the information feedback in a service interval, but rather the time between two subsequent information feedbacks (which includes time for transmitting the video packet, the information feedback and also the protocol overheads).

$$\vec{h}(\mathbf{b}_p, f_{\text{info}}(1)) = f_{\text{info}}(1) \times t_{SI} = c. \quad (8)$$

We focus on the impact of different information horizons directly on the video qualities of multiple users sharing the same multi-hop wireless network. Note that $f_{\text{info}}(1)$ can be converted into an information horizon based on equation (7), as long as the information dissemination factors (i.e. $\mathbf{b}$ vector) are given. Thus, for simplicity, in the remainder part of the chapter, we denote the information horizon $\vec{h}(\mathbf{b}, f_{\text{info}}(1))$ by simply $\vec{h}$. An example of $\vec{h} = 2$ is shown in Figure 4.2. The local information feedback in equation(3) for a larger information horizon becomes a vector $\mathcal{Z}_{\text{local}} = [SINR_{m_h}, P_{k,m_h}, E[\text{Delay}_{k,m_h}] \mid m_h \in \mathbf{M}_{k,h+1} \cup \dots \cup \mathbf{M}_{k,h+\vec{h}}, \forall k]$, where $\mathbf{M}_{k,h+1}$ represents a set of nodes in the $h+1$ -th hop that feedback the information for the class f_k traffic to the decision nodes (e.g. node “n1” in the example in Figure 4.2).

B. The impact of various information horizons

With a larger information horizon, more accurate network status can be obtained, which can be used to adapt the cross-layer transmission strategies at various layers. A larger information horizon ensures that the information can be obtained in a timely manner and network status can be estimated more accurately. For example, a better routing decision can be determined to avoid the congested regions in the network. This decreases the packet loss probability P_k for each class, thus increasing the P_k^{succ} for the important classes and improving the received video qualities. However, the penalty of the overhead is seldom jointly considered in the prior works. Let $E[\text{Time}_{k,m_h}^{\text{packet}}(\vec{h})]$ represent the expected transmission time for a video packet in class f_k at node m_h to the next hop with the information feedback of horizon h^{horizon} . Based on the geometric assumption, we can write:

$$E[\text{Time}_{k,m_h}^{\text{packet}}(\vec{h})] = \sum_{m_{h+1}=1}^{M_{h+1}} \beta_{k,m_h,m_{h+1}} \left(\frac{1 - p_{k,m_h,m_{h+1}}^{\gamma_{k,m_h,m_{h+1}}^{\text{MAX}} + 1}}{1 - p_{k,m_h,m_{h+1}}} \right) \left(\frac{L_k + L_{\text{header}}}{T_{k,m_h,m_{h+1}}} + \text{Time}^{\text{over}}(\vec{h}) \right), \quad (9)$$

which is calculated as an average transmission time over all the possible relays m_{h+1} in the next hop. $\text{Time}^{\text{over}}(\vec{h})$ denotes the time overhead introduced by the various protocols [IEE03] including the time of waiting for the MAC acknowledgements etc., and also the

information feedback. Consequently, a larger information horizon can induce larger overheads for the packet transmission time, and hence increases the end-to-end delay D_k , which can lead to higher packet losses P_k as they can miss their deadline. In this chapter, we assume that the time overhead is a known function of the information horizon and we will discuss this in more detail in Section VII.

In general, the information horizon might be different for various users or classes and also can vary per node, depending on its location, congestion level, etc. Thus, a scalable information feedback can be implemented (e.g. the information horizon can depend on class f_k and node m_h). For instance, to reduce the overhead associated with the information feedback, some less important classes can have smaller horizons. However, for simplicity, the information horizon is assumed to be the same for all classes (users) in the rest of the chapter. The topic of implementing the scalable information feedback and the analysis of its impact form a topic of our future research.

C. Distributed cross-layer adaptation based on the information feedback with larger information horizons

Instead of performing the exhaustive search for the distributed optimization in equation (3), we present the following iterative cross-layer adaptation to solve the multi-user video streaming problem. Based on the information feedback, the goal of the distributed cross-layer adaptation is to determine an optimal packet present in the queue (from π_{m_h}) to be transmitted through the optimal relay m_{h+1} (from $\beta_{k,m_h,m_{h+1}}$) in the next hop with the optimal retransmission limit (from $\gamma_{k,m_h,m_{h+1}}^{MAX}$).

1. To determine a packet of a specific class f_k for transmission, the packet scheduling policy π_{m_h} in the queue of the intermediate node m_h is optimized to first transmit the video packets with larger λ_k , since they have a higher impact on the overall video quality. With a larger information horizon, such packet scheduling can be improved as we will discuss it in Section IV.
2. To solve the routing problem, we deploy a priority queuing approach based on the

information feedback and apply dynamic routing policies similar to the Bellman-Ford routing algorithm [BG87]. We exploit the $E[Delay_{k,m_h}]$ in the local information feedback $\mathcal{Z}_{local}$. The selection of $\beta_{k,m_h,m_{h+1}}$ is based on the $E[Delay_{k,m_{h+1}}]$ value that minimizes the end-to-end packet loss probability P_k for the transmitted packet. We will discuss the routing problem in detail in Section VI.

3. At the MAC layer, we choose the appropriate retransmission limit $\gamma_{k,m_h,m_{h+1}}^{MAX}$ per packet based on the $T_{k,m_h,m_{h+1}}^{goodput}$ such that its delay constraint is satisfied. Based on our prior results [VT07] in one-hop network, the optimal retransmission strategy is to send the highest priority packet until it is successfully received by the next relay or until its delay deadline expires. Specifically, let d^{curr} represent the current delay incurred by a particular packet at the current node m_h . The maximum retransmission limit for the packet of class f_k over the link from m_h to m_{h+1} is determined based on the delay deadline d_k (where $\lfloor \cdot \rfloor$ is the floor operation):

$$\gamma_{k,m_h,m_{h+1}}^{MAX} = \left\lfloor \frac{T_{k,m_h,m_{h+1}}^{goodput} (d_k - d^{curr})}{L_k} \right\rfloor - 1. \quad (10)$$

With a larger information horizon, the retransmission limit can be improved as we will discuss it in Section V.

4. Then, we measure the $SINR$ and estimate the corresponding $E[Delay_{k,m_h}]$ and P_{k,m_h} for each class f_k at the node m_h and feed back this information to the nodes in the previous hops within the information horizon $\vec{h}$.

IV. RISK-AWARE SCHEDULING FOR MULTIMEDIA STREAMING

At each intermediate node m_h , in order to optimize the scheduling of the various video packets, we determine the risk $Risk_{k,m_h}$ ($0 \leq Risk_{k,m_h} \leq 1$) that the packets of class f_k will miss their delay deadline, based on the probability that the estimated received time at the destination is *after* their delay deadlines. Higher probabilities of packet loss over the network (due to interference, congestion, nodes leaving etc.) will lead to higher risks of packets missing their delay deadlines. Based on this risk, the scheduling of the

various packets of the different classes can be determined to ensure a maximized system quality.

To compute the risk estimation for a packet, we need to consider both the delay deadlines d_k as well as the expected delay $E[Delay_{k,m_h}]$ in the information feedback $\mathcal{Z}_{local}$ conveyed from the intermediate nodes within the information horizon $\vec{h}$. The video packets at an intermediate node can be divided into three categories:

- Packets that will certainly be dropped (“dropped” packets).
- Packets that have very high probability to be dropped (“almost-dropped” packets).
- Packets that have low probability to be dropped (“seldom-dropped” packets).

“Dropped” packets are video packets with a current cumulative delay d^{curr} exceeding their delay deadline ($d^{curr} > d_k$). These packets will be dropped at the current node and hence, there is no need to compute their risk. The “almost-dropped” packets have not yet exceeded their delay deadline ($d^{curr} < d_k$), but their current cumulative delay plus the expected delay to reach the destination does exceed their delay deadline, i.e. $d^{curr} + E[Delay_{k,m_h}] > d_k$. We set the risks for these “almost-dropped” packets to be 0, as they have a very high probability of being dropped and hence, they will unnecessarily waste resources that could be used for the successful transmission of “seldom-dropped” packets. The remaining video packets are “seldom-dropped” packets. Their current cumulative delay plus the expected delay from the current node to the destination is lower than the delay deadline, i.e. $d^{curr} + E[Delay_{k,m_h}] < d_k$. Hence, these packets have a high probability of arriving at the destination on time and their scheduling needs to be optimized to maximize the video quality across the various users. Next, we discuss how to estimate the risk for these seldom-dropped packets.

A. Risk estimation based on priority queuing analysis

The risk estimation for the seldom dropped packets is determined based on the priority queuing analysis, by using the approximation of the waiting time tail distribution. Let W_{k,m_h} represent the queue waiting time for class f_k at intermediate node m_h . The

waiting time tail distribution can be approximated as [JTK01][ACW95]:

$$\text{Prob}(W_{k,m_h} > t) \approx \left(\sum_{i=1}^k \eta_{i,m_h} E[X_{i,m_h}] \right) \exp\left(-\frac{\sum_{i=1}^k \eta_{i,m_h} E[X_{i,m_h}]}{E[W_{k,m_h}]} \times t\right), \quad (11)$$

where η_{k,m_h} is the measured average input rate and $E[X_{i,m_h}]$ is the average service time of class f_k at the intermediate node m_h . The expected average queue waiting time of the priority queue is:

$$\int_{t=-\infty}^{\infty} \text{Prob}(W_{k,m_h} > t) dt = E[W_{k,m_h}] = \frac{\sum_{i=1}^k \eta_{i,m_h} E[X_{i,m_h}^2]}{2 \left(1 - \sum_{i=1}^{k-1} \eta_{i,m_h} E[X_{i,m_h}] \right) \left(1 - \sum_{i=1}^k \eta_{i,m_h} E[X_{i,m_h}] \right)}. \quad (12)$$

equation(12) is determined based on the Mean Value Analysis (MVA) of a preemptive-priority M/G/1 queue [BG87]. Until now, we do not consider the interference incurred in wireless multi-hop networks (orthogonal transmission channels are available for adjacent wireless links), the average service time $E[X_{k,m_h}]$ is the average packet transmission time $E[Time_{k,m_h}^{packet}(\vec{h})]$ in equation(9). If the influence of interference is considered, the average service time $E[X_{k,m_h}]$ can be approximated using a virtual queue analysis similar to the “service on vacation” concept in queuing theory [Kle75][BG87]. Using equation(11), the proposed risk estimation² for the packets in class f_k can be computed as:

$$\begin{aligned} Risk_{k,m_h}(Time^I) &= \begin{cases} \text{Prob}(W_{k,m_h} + Time^I > E[d_k^{left}]), & \text{if } E[d_k^{left}] > 0 \text{ (seldom-dropped packets)} \\ 0 & \text{,if } E[d_k^{left}] \leq 0 \text{ (almost-dropped packets)} \end{cases} \\ &= \begin{cases} \left(\sum_{i=1}^k \eta_{i,m_h} E[X_{i,m_h}] \right) \exp\left(\frac{\sum_{i=1}^k \eta_{i,m_h} E[X_{i,m_h}]}{E[W_{k,m_h}]} \times (Time^I - E[d_k^{left}]) \right), & \text{if } E[d_k^{left}] > 0 \\ 0 & \text{, if } E[d_k^{left}] \leq 0 \end{cases}, \end{aligned} \quad (13)$$

where $E[d_k^{left}] = d_k - d^{curr} - E[Delay_{k,m_h}]$ represents the expected time remaining after a

² The higher risk packets should be sent earlier, since they are with high probability to exceed their deadlines. However, we do not want to waste our resources on those almost-drop packets, hence the risk estimation for these packets are set to zero.

packet reaches its destination. We can determine the probability that the waiting time W_{k,m_h} plus a pre-determined time duration $Time^I$, which is a general variable for risk estimation, exceeds the expected time left $E[d_k^{left}]$, and thus, that the packet will be lost. The time duration $Time^I$ can be viewed as an extension of the waiting time for the packet. Larger $Time^I$ values lead to higher risks. An example of the risk estimation is given in Section IV.B. Note that the accuracy of computing the expected time left $E[d_k^{left}]$ increases with a larger information horizon. Thus, the $Risk_{k,m_h}(Time^I, \vec{h})$ also depends on the information horizon $\vec{h}$ and can be better estimated given a larger $\vec{h}$.

B. Feedback-driven scheduling

In a priority queue, the packet scheduler at an intermediate node transmits first the most important packets (i.e. the packets with the largest λ_k). Each packet is transmitted until the packet is successfully received by the next hop node or until its deadline expires. Assume that there are L total video packets at the intermediate node m_h . Let the application layer packet scheduling $\pi_{m_h} = (\pi_1, \dots, \pi_l, \dots, \pi_L)$, where π_l represents the scheduling order for the video packet $l \in \{1, \dots, L\}$. The basic priority scheduling can be written as:

$$\begin{aligned} \pi_{m_h}^{PRI} = \arg \max_{\pi_{m_h}} & \left\{ \sum_{k=1}^K \lambda_k \times N_{k,m_h}(\pi_{m_h}, Time_{\pi}^I) L_k \right\} \\ \text{subject to } \pi_{m_h} &= (\pi_1, \dots, \pi_l, \dots, \pi_L), \\ \pi_l &= drop, \text{ if } l \in f_k, \text{ and } d_l^{curr} \geq d_k \end{aligned} \quad (14)$$

where $N_{k,m_h}(\pi_{m_h}, Time_{\pi}^I)$ is the number³ of packets of the class f_k that are transmitted during a period of time $Time_{\pi}^I$ using a specific packet scheduling π_{m_h} . The notation $\pi_l = drop$ indicates that the packet l is not scheduled due to its deadline expiration.

A packet could be dropped in the future hops, as its deadline is exceeded at these hops, and the transmission time of this packet is wasted. This may results in the loss of other packets that would have arrived on time at their destination. Thus, enabled by the

³ Packet loss is considered in this number due to the delay constraint that drops packets.

information feedback, an intermediate node gathers the network status and makes a scheduling decision. Instead of always transmitting the most important packet in the queue, some other video packets of the different users that are less important but have a higher packet loss probability (risk) can be sent first. Based on this, we propose a novel Information Feedback Driven packet Scheduling (IFDS). The system map of the IFDS scheduling at an intermediate node is illustrated in Figure 4.3. The risk is estimated using the information feedback $E[Delay_{k,m_h}]$ and the waiting time distribution (see equation (13)).

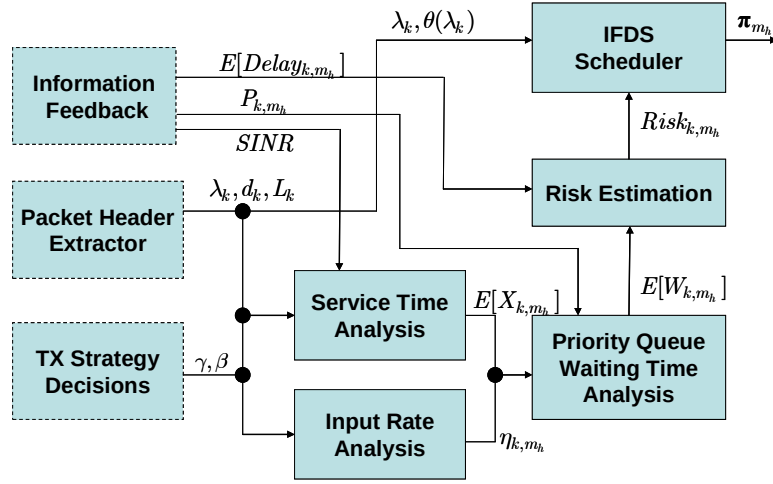


Fig. 4.3 System map for the IFDS packet scheduling.

For the IFDS scheduling, the video packets ordered in $\pi_{m_h}^{IFDS}$ are transmitted for a pre-determined period of time $Time_{\pi}^{Interval}$. The IFDS scheduling is determined as:

$$\begin{aligned} \pi_{m_h}^{IFDS}(\vec{h}) &= \arg \max_{\pi_{m_h}} \left\{ \sum_{k=1}^K \lambda_k \times N_{k,m_h}(\pi_{m_h}, Time_{\pi}^I) L_k \times Risk_{k,m_h}(Time_{\pi}^I, \vec{h}) \right\} \\ \text{subject to } \pi_{m_h} &= (\pi_1, \dots, \pi_l, \dots, \pi_L), \\ \pi_l &= drop, \text{ if } l \in f_k, \text{ and } d_l^{curr} \geq d_k \end{aligned} \quad (15)$$

As opposed to the priority queuing scheduling (equation (14)), the risk of losing a certain class $Risk_{k,m_h}(Time_{\pi}^I, \vec{h})$ is considered jointly with the packet quality impact. The scheduler sends the packets in the order that maximizes the output video quality weighted

by $\lambda_k Risk_{k,m_h}$ within the time interval $Time_\pi^I$. Since different traffic classes have different packet transmission times $E[Time_{k,m_h}^{packet}]$ (see equation (9)), the number of packets being transmitted per class $N_{k,m_h}(\pi_{m_h}, Time_\pi^I)$ depends on which packets are sent (scheduling decision). However, the $Risk_{k,m_h}(Time_\pi^I, \vec{h})$ remains constant and is independent of the scheduling decision within $Time_\pi^I$. Recall that with a larger information horizon $\vec{h}$, the risk is estimated more accurately because the node is able to obtain more accurate information from nodes which are closer to the destination. Hence, the packet scheduling policy $\pi_{m_h}^{IFDS}$ is more accurate and adaptive to the network changes than the priority scheduling strategy of equation (14). Finally, the IFDS scheduling has the following constraint:

$$\pi_{m_h}^{Final} = \left\{ \pi_{m_h}^{IFDS} = (\pi_1, \dots, \pi_l, \dots, \pi_{l'}, \dots, \pi_L) \mid \pi_l \ll \pi_{l'}, \right. \\ \left. \text{only if } l \in f_k, l' \in f_{k'} \text{ and } \lambda_k > \theta(\lambda_{k'}) \right\}, \quad (16)$$

where the notation $\pi_l \ll \pi_{l'}$ represents that packet l is scheduled before packet l' . If λ_k belongs to user v , the $\theta(\lambda_k)$ is a class dependent threshold, which can be defined as:

$$\theta(\lambda_k \in \text{user } v) = \max_{u \in \{k+1..K\}} \{\lambda_u \mid f_u \in \text{the same user } v\}. \quad (17)$$

Equation (17) provides a threshold for a particular class, which is the quality impact value of the next important class of the same user. The reason for the constraint in equation (16) is to avoid sending an unimportant class with high risk (i.e. for the classes of the same user, packets with higher λ_k must be sent first). This is important since the less important classes depend on the more important classes of the same user and hence, their distortion will be significantly impacted if the higher priority packets are lost [VT07].

An example of the risk estimation at an intermediate node m_h with fixed $\vec{h}$ is given in Figure 4.4 for a case of two users and four classes with the quality impact parameters $\lambda_1 > \lambda_2 > \lambda_3 > \lambda_4$. User 2 (with classes f_2 and f_3) has a smaller expected time left $E[d_k^{left}]$ than user 1 (having classes f_1 and f_4). Note that when $Time_\pi^I \geq E[d_k^{left}]$, $Risk_{k,m_h}(Time_\pi^I) = 1$ for all the classes, because they miss their deadlines after waiting for

$Time_\pi^I$. Let us now adopt the IFDS packet scheduling algorithm, and set the $Time_\pi^I$ between $E[d_1^{left}]$ and $E[d_2^{left}]$. From Figure 4.4, we can observe that $Risk_{2,m_h}(Time_\pi^I) = 1$ and $Risk_{1,m_h}(Time_\pi^I) \cong 0$. Hence, the packets of class f_1 can wait for $Time_\pi^I$ without significantly increasing the packet loss, while the packets of class f_2 that are less important ($\lambda_2 < \lambda_1$) are transmitted.

From the example, we see that the setting of $Time_\pi^I$ affects the risk estimation and hence the scheduling decision. Note that if we set $Time_\pi^I$ larger than the maximum delay deadline of all the users, the risk will be 1 for all the seldom-dropped packets, and thus the information feedback driven scheduling will only depend on λ_k . If $Time_\pi^I$ is set too small, the risk estimations will not affect the original priority decision. Thus, we define a lower and an upper bound of the $Time_\pi^I$:

$$\min_k E[d_k^{left}] \leq Time_\pi^I \leq \max_k E[d_k^{left}], \text{ where } E[d_k^{left}] > 0 \text{ (for seldom-dropped packets)}, \quad (18)$$

since the risk estimations are large enough to take effect within this interval. For the example in Figure 4.4, $E[d_3^{left}] \leq Time_\pi^I \leq E[d_1^{left}]$.

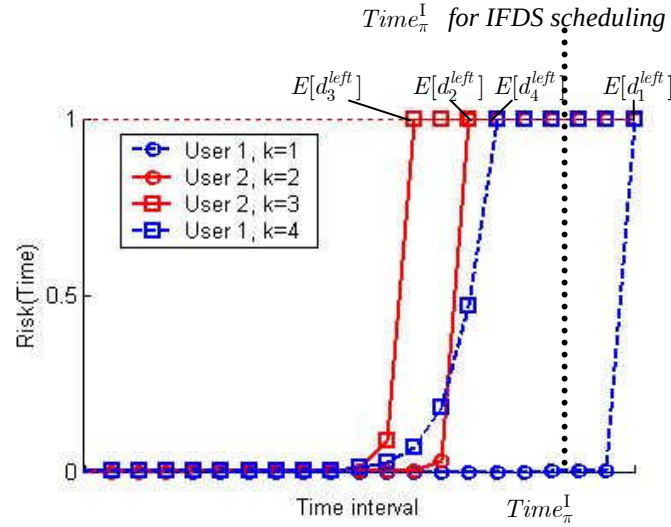


Fig. 4.4. Risk estimation vs. time interval for 2 users.

V. RISK-AWARE MAC LAYER RETRANSMISSION STRATEGY

For protection over an error-prone wireless link, a retransmission scheme at the MAC layer is adopted. In [VT07], it was shown that for the scalable video coders such as [AMB04], the video packets should be retransmitted by the MAC until they are received without error or their deadline expires in order to maximize the received video quality. However, if a packet approaches its delay deadline, the risk that it will not reach its destination increases. Hence, similarly to the application layer scheduling strategies discussed in the previous section, we propose a MAC layer information feedback driven retransmission strategy $\gamma_{k,m_h,m_{h+1}}^{IFDS}(\vec{h})$ that explicitly considers the risk of losing a packet based on the available information feedback $\mathcal{I}_{local}$.

Let $retry$ be an integer variable that represents the number of retransmissions for a packet. If the transmission of the packet repeatedly fails, the retransmission should last only until another class of video packets starts to have a higher impact in terms of overall video quality. In both scheduling policies in the previous section, the scheduler will send packets of class f_k having a larger $\lambda_k Risk_{k,m_h}$ value (see equation (15)). Therefore, the information feedback driven retransmission limit becomes:

$$\begin{aligned} \gamma_{k,m_h,m_{h+1}}^{IFDS}(\vec{h}) &= \text{maximize } \gamma \\ \text{subject to } \lambda_k Risk_{k,m_h}(Time_\gamma^I, \vec{h}) &\geq \lambda_j Risk_{j,m_h}(Time_\gamma^I, \vec{h}), \text{ for all } j \text{ that } \lambda_j > \theta(\lambda_k), \\ Time_\gamma^I &= (\gamma + 1) \times Time_{k,m_h,m_{h+1}}^{packet}, \gamma \in \mathbb{N}, \end{aligned} \quad (19)$$

which states that the retransmission limit is the maximum number of retries such that the transmitting packet (of class f_k) has a greater $\lambda_k Risk_{k,m_h}$ than other packets in the queue. Due to the scheduling constraint in equation (16), we only need to check the classes that have a quality impact value larger than the threshold $\theta(\lambda_k)$ in equation (19). Note that the information feedback driven retransmission limit is always smaller than the retransmission limit in equation (10) ($\gamma_{k,m_h,m_{h+1}}^{IFDS} \leq \gamma_{k,m_h,m_{h+1}}^{MAX}$), since when a packet approaches the deadline, it will first belong to the “almost-dropped” packets class ($d^{curr} + E[Delay_{k,m_h}] > d_k$), for which $Risk_{k,m_h} = 0$. Thus, another class of packets will be

transmitted, thereby terminating the retransmission of the current packet. Consequently, a packet retransmission will first reach the information feedback driven retransmission limit $\gamma_{k,m_h,m_{h+1}}^{IFDS}$ before the delay deadline. Thus, other packets that have a better chance to reach the destinations could be sent earlier.

VI. OVERHEAD ANALYSIS FOR INFORMATION FEEDBACK

The information feedback can enable the cross-layer adaptation of video streaming over a multi-hop network. As the information horizon increases, the network status can be estimated more timely and accurately, and the cross-layer strategies can be improved for the delay-sensitive applications. However, a larger information horizon also consumes more network resources for video transmission and results in an increased time overhead per packet transmission, $Time_k^{over}(\vec{h})$ (see equation (9)). Various information feedback parameters have different transmission overheads. In this chapter, we take the three information feedback parameters illustrated in Figure 4.2 as examples.

Assuming a certain topology, let us perform a worst-case analysis to quantify the maximum information feedback. We assume that the information feedback overheads are $I_{SINR}, I_{Ploss}, I_{E[Delay]}$ for the three information feedback parameters, respectively. We assume that the average number of nodes in one hop is M , the number of total classes is K , and we set the information horizon as $\vec{h}_{\max}$ for all users (classes). The SINR information is fed back from potential receivers to the transmitters to enable the link adaptation as well as to facilitate the polling control signaling. Thus, an information horizon of only 1 hop is sufficient for the adopted overlay infrastructure, and the overhead in terms of the information feedback unit is $M^2 I_{SINR}$. As for the other two information feedback parameters, the parameters are required across the whole information horizon and different for all the classes. An aggregation scheme $\mathcal{G}$ can be applied to reduce the repeated information (as in e.g. [KV04][KEW02]). The worst-case overheads in terms of the information feedback unit are $\vec{h}_{\max} KM \cdot I_{Ploss}(\vec{h}_{\max}, \mathcal{G})$ and $\vec{h}_{\max} KM \cdot I_{E[Delay]}(\vec{h}_{\max}, \mathcal{G})$, respectively. $I_{Ploss}(\vec{h}_{\max}, \mathcal{G})$ and $I_{E[Delay]}(\vec{h}_{\max}, \mathcal{G})$ represents the

functions of aggregated information feedback over $\vec{h}_{\max}$ hops for these two information feedback parameters. In conclusion, the information feedback overhead increases with the information horizon.

VII. SIMULATION RESULTS

To assess the importance of information feedback, we consider several multi-user video transmission scenarios. Two video sequences, “Mobile” and “Coastguard” (16 frames per GOP, frame rate of 30 Hz, CIF format) compressed using a scalable video codec [AMB04] are sent from distinct sources to their corresponding destinations through the multi-hop wireless network shown in Figure 4.5. We consider four different scenarios with various information horizons and information feedback overheads as stated in Table 4.1. Each video sequence is divided into four classes ($C_v = 4, K = 8$). The quality impact parameters λ_k and the number of packets N_k in one group of picture for each class are the same as the previous chapter in Table 3.1.

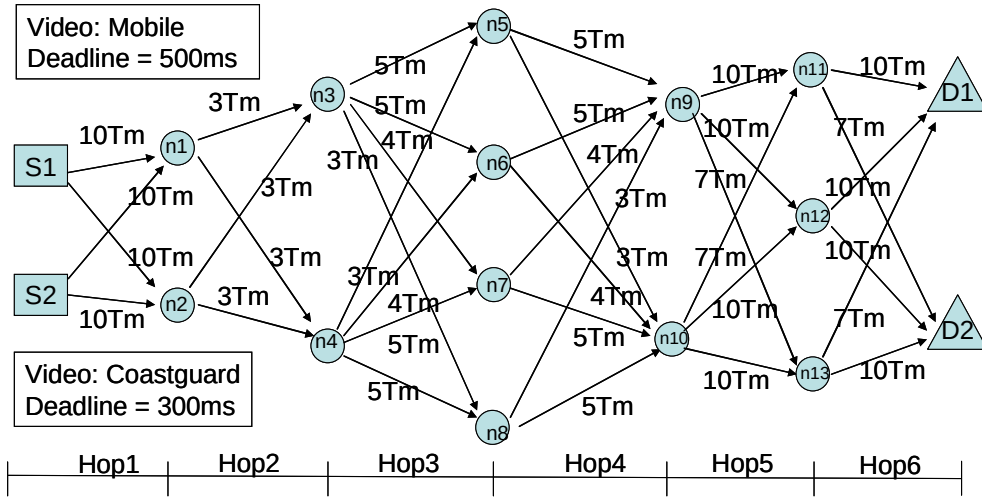


Fig. 4.5. Simulation settings of a 6-hop overlay network with 2 video sequences.

TABLE 4.1 DESCRIPTIONS FOR THE FOUR CASES OF THE SIMULATION RESULTS ($t_{SI} = 100$ ms).

Scenario	Information horizon $\vec{h}$	Information feedback interval t_{info}	Overhead per packet $Time^{over}(\vec{h})$	Equivalent overhead/packet ratio of “Coastguard”	Equivalent overhead/packet ratio ($T_m=300$ Kbps)
$c=1$	1 hop	100ms	0.1 ms	$2.1 \times 10^{-8} T_m$	6.3/1000
$c=2$	2 hops	50ms	0.2 ms	$4.2 \times 10^{-8} T_m$	12.6/1000
$c=3$	3 hops	33ms	0.3 ms	$6.3 \times 10^{-8} T_m$	18.9/1000
$c=4$	4 hops	25ms	0.4 ms	$8.4 \times 10^{-8} T_m$	25.2/1000

In our simulation, we captured the packet-loss pattern under different channel conditions (described in the chapter by the link SINR) using our wireless streaming test-bed [KV04]. In this way, we can assess the efficiency of our system under real wireless channel conditions and link adaptation mechanisms currently deployed in state-of-the-art 802.11a/g wireless cards with 802.11e extension. Link adaptation selects one appropriate physical-layer mode (modulation and channel coding) depending on the link condition, in order to continuously maximize the experienced goodput [KV04]. Hence, each link in our network settings shown in Figure 4.5 is assigned with an effective transmission rate measured from the test-bed. The parameter T_m represents the streaming efficiency of the network. The various efficiency levels are represented by varying the available time fraction for the contention-free period in the polling-based MAC protocol, which induces the various available transmission rates for the video packets over the links. In our event-driven simulation, these network efficiency levels range from 300 Kbps to 500 Kbps. A larger T_m gives higher network efficiency. We set $t_{SI} = 100$ ms, $\mathbf{b} = \mathbf{b}_p$ (see Section III) and $\vec{h}$ varies from 1 to 4 for the four scenarios. The information feedback overheads are set as $Time^{over}(\vec{h}) = t_{SI} / 1000 \times \vec{h}$ for all the classes. Note that the time overhead is limited, i.e. 2.5% of the average packet transmit time when $T_m=300$ Kbps, and $\vec{h} = 4$.

Note that the effect of the IFDS scheduling depends on many factors, such as the network topology, application characteristics, network transmission efficiency, and congestion/interference conditions, etc. Here, we would like to assess the importance of

the risk consideration in resource-constrained networks. We set the application playback delay deadlines are set to 500 ms and 300 ms for the classes of the two video sequences respectively. The transmission rates of the links in the first hop are, relatively higher than the subsequent links. Consequently, most of the packets of the various classes will be queued at the specific intermediate nodes $n1$ and $n2$ (some of them will still be left in the source queues), and the effect of risk can be highlighted for two streams with different delay deadlines.

We adopt the IFDS scheduling and the retransmission limit algorithm in Section IV and V for cases with larger information horizons ($\vec{h} \geq 2$). In scenario 1, we make the packet scheduling first transmit the packets with the highest quality impact parameter λ_k until the transmission success or delay deadline expiration (i.e. equation(14)). In scenario 2, the risk estimation is considered jointly with the quality impact parameters using equation(15). In scenarios 3 and 4, larger information horizons are used in equation(15) for risk estimation. However, with larger information horizon, the performance degrades due to larger information feedback overheads. The simulation results of the packet loss rate of each class at their destinations are shown in Table 4.2 under various network transmission efficiencies. Since the delay deadline of the “Coastguard” sequence is smaller, it has higher packet loss rate, especially in networks with low transmission efficiency. However, it is shown that as the information horizon increases, the IFDS scheduling sends more “Coastguard” packets to improve its video quality without degrading significantly the video quality of the “Mobile” sequence.

To observe the impact of the various information horizons on the overall video quality, the average Y-PSNR decoded at the destinations of the two sequences are shown in Figure 4.6. It shows that the optimal choice of information horizon varies with the network transmission efficiency. For networks with high transmission efficiency, a larger information horizon ($\vec{h} \geq 3$) makes the IFDS scheduling more efficient, and improves the video qualities. However, for a network with low transmission efficiency that is more

congested, a shorter information horizon ($\vec{h} \leq 2$) results in better performance since the limited network resource can be focused on the video transmission (payload).

TABLE 4.2 SIMULATION RESULTS FOR IFDS SCHEDULING WITH VARIOUS INFORMATION HORIZONS AND DIFFERENT NETWORK EFFICIENCIES.

Mobile (1668 Kbps)	Packet loss probability P_k (delay deadline $d_k = 500\text{ms}$) for different Tm (Kbps)												
f_k	Scenario 1			Scenario 2			Scenario 3			Scenario 4			Opt. Value
	300	400	500	300	400	500	300	400	500	300	400	500	
f_1	0%	0.3%	0%	5.5%	2.0%	0%	7.8%	1.2%	0.2%	1.3%	0.8%	0%	0%
f_3	21%	8.1%	3.3%	62%	18%	3.0%	68%	16%	4.1%	51%	18%	4.2%	0%
f_6	79%	30%	12%	100%	69%	15%	100%	52%	19%	100%	55%	19%	0%
f_8	100%	95%	83%	100%	100%	83%	100%	100%	82%	100%	100%	90%	0%
Y-PSNR (dB)	29.46	30.98	31.66	28.66	30.02	31.24	28.30	30.16	31.22	28.23	29.97	31.16	33.12
Coastguard (1500 Kbps)	Packet loss probability P_k (delay deadline $d_k = 300\text{ms}$) for different Tm (Kbps)												
f_k	Scenario 1			Scenario 2			Scenario 3			Scenario 4			Opt. Value
	300	400	500	300	400	500	300	400	500	300	400	500	
f_2	33%	10%	11%	7.9%	5.7%	1.7%	8.1%	4.8%	3.9%	8.0%	6.4%	1.2%	0%
f_4	100%	96%	41%	95%	65%	51%	97%	67%	27%	99%	43%	38%	0%
f_5	100%	100%	96%	100%	100%	100%	100%	100%	98%	100%	99%	96%	0%
f_7	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	0%
Y-PSNR (dB)	28.51	29.35	30.34	29.92	30.49	31.13	29.81	30.76	31.84	29.73	30.68	31.67	35.61

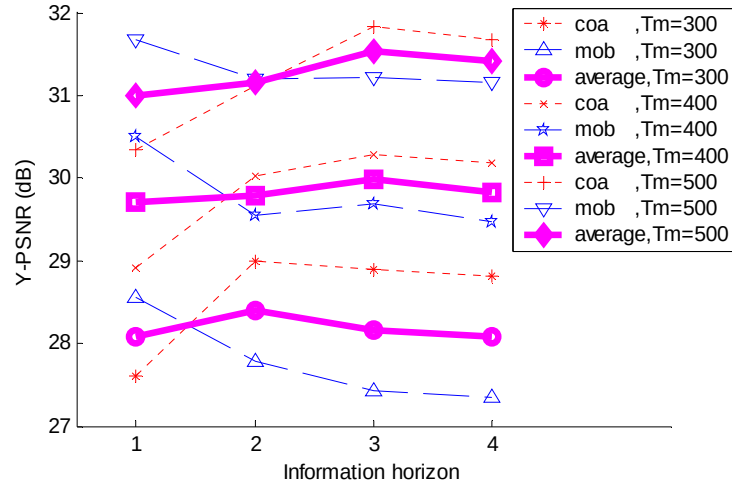


Fig. 4.6 Y-PSNR vs. various information horizon cases under different network transmission efficiencies.

VIII. CONCLUSIONS

In this chapter, we investigate the impact of information feedback with different network horizons on the video quality of multiple users sharing the same multi-hop wireless network. We illustrate how the various cross-layer strategies can be adapted to take advantage of the available information feedback from a larger network horizon through the proposed information feedback driven scheduling, retransmission limit and the dynamic priority hybrid routing algorithm. Unlike the end-to-end feedback that exists in today's networking protocols (such as the rate control in TCP), the information feedback is performed in a distributed (per hop) fashion that explicitly considers the instantaneous delays, which is essential for supporting delay-sensitive multimedia applications. We investigate the tradeoff between the increased transmission overhead and the benefit of larger information horizons leading to an improved prediction of network conditions. The results show that in a network with higher transmission efficiency, a larger information horizon can lead to an improved performance in terms of video quality, which leads to more than 2 dB improvement in video quality as a result of balancing the effect of different delay deadline among users. However, with lower transmission efficiency, smaller information horizon performs better by ensuring limited overhead of the information feedback.

Chapter 5

Feedback-Driven Interactive Learning in Mobile Ad Hoc Networks

I. INTRODUCTION

Power control is an important problem in wireless networks. Prior literature has investigated such dynamic resource management jointly with routing [NMR05], time sharing [KOG07], frequency channel selection [SB97][ZZY05], power allocation [GKG07][YGC02][LZL07], etc. In this chapter, we focus on the non-cooperative decentralized setting, where autonomous users make decisions on accessing resources based on their current knowledge about their opponents as determined from information feedback. Such information feedback is essential for decentralized dynamic resource management, since in informationally-decentralized wireless networks, it is impossible for a user to know the exact actions of the other users sharing the network. Hence, it is important to investigate how users can dynamically adapt their current decisions to maximize the expected utility based on available information feedback. We focus on the joint *power-spectrum* allocation problem for dynamic resource management in wireless network, since the interference at the physical layer results in a strong coupling between the transmission actions (i.e. the power/frequency channel selections) of the competing users. However, the proposed solution can also be used in other decentralized dynamic resource management problems.

Joint power and spectrum resource allocation research has attracted a lot of attention in recent years [YL06][YGC02][MCP06][GM00][SMG02][XSC03]. For the multi-user case to maximize the overall throughput, the resource allocation problem becomes very complicated since the wireless mutual interference among users results in a nonconvex

optimization problem [YL06]. The computational complexity of the centralized approaches becomes prohibitive as the number of users grows. Moreover, the centralized approaches require the propagation of global control information back and forth to a common coordinator, thereby incurring heavy signaling overhead [GKG07]. Hence, decentralized solutions, such as the “iterative water filling” approach [YGC02], are more desirable in practice.

Recently, game-theoretic concepts have been applied to deal with the decentralized resource allocation problem [MCP06][GM00][SMG02][XSC03][LZL07] using various utility functions. For example, in [MCP06], non-cooperative power control games were constructed where each user possesses an energy-efficient utility function. The existence and uniqueness of Nash equilibrium in such non-cooperative game was extensively studied. In [MCP06][GM00], other than maximizing the throughput, users maximize a ratio of throughput over the transmitted power (measured in bits/joule). In [SMG02][XSC03], a pricing mechanism was employed to provide Pareto-efficient solutions by adopting an additional penalty term associated with the power consumption in the utility function. In [LZL07], a reinforcement learning approach for the non-cooperative game is proposed and the convergence property of the reinforcement approach was studied.

In short, previous research mainly concentrates on studying the existence and performance of Nash equilibrium in non-cooperative games or developing efficient algorithms to approach the Pareto boundary. However, prior research does not consider the users’ availability of information feedback from various users and ignores the performance degradation when the actions of the other users are not accurately modeled. Note that without a central coordinator, multiple users sharing the same wireless network need to manage their local resources based on the available information feedback. Hence, the best response strategy of a selfish user making decisions in the non-cooperative game based on “limited” (incomplete) information feedback [GKG07] still needs to be

determined. Intuitively, a “foresighted” user with more information should be able to gain more benefits in such a non-cooperative game. However, such information feedback is not costless. In practical systems, heavy signaling overhead can degrade the users’ performance [SV07b]. Therefore, it is important to investigate what is the benefit that a user can derive from gathering more information feedback, which allows it to better model the competing wireless users, while *explicitly* considering the cost of feeding back the information.

In this chapter, we investigate two types of information feedback for autonomous self-interested users (transmitter-receiver pairs) in the power control problem. The transmitters will select the transmitting power levels and the frequency channels by maximizing the utility function based on two types of information feedback:

- 1) *Private information feedback* – To evaluate the utility function, transmitters actually require their receivers to provide important channel state information, the Signal-to-Interference-Noise Ratio (SINR). The SINR value contains the aggregate effect of other users’ actions and this value can only be measured at the receiver side. Such information needs to be fed back to the transmitter to make decisions. This information feedback between the transmitter-receiver pair is referred to as the private information feedback.
- 2) *Public information feedback* – When non-cooperative users have incentives to exchange information (depending on the communication protocols, such as in [LS99]), explicit information feedback about the other users’ actions enables a user to directly model the other users efficiently and hence, improve the accuracy of the utility evaluation resulting from taking different actions. Even when users are non-cooperative, they can still reveal their action information to others in order to maximize their own utilities [FL98]. This explicit information feedback among users is referred to as the public information feedback.

Note that the private information feedback contains implicit information about the other

users' actions in the network. On the other hand, by gathering public information feedback, users can explicitly model their opponents. Due to the informationally-decentralized nature of the wireless network, when a user makes decisions, the user does not know the exact transmission actions that its interfering neighbors will take. If a user is foresighted, meaning that it can predict the exact actions of its competing users by exploiting the experienced information feedback, its performance can be improved [SB97][FL98]. This requires the user to learn the transmission strategies of its major interferers through interactive learning [You04] based on the available information feedback. Figure 5.1 illustrates the differences of the conventional distributed power control and the proposed power control using interactive learning. We discuss two classes of interactive learning schemes are compared – payoff-based learning and model-based learning, depending on the type of the information feedback. In this chapter, we assume that the information feedback is truthful and error-free¹, and investigate how to adapt the information feedback to enable a user to maximize its utility in different network scenarios through interactive learning.

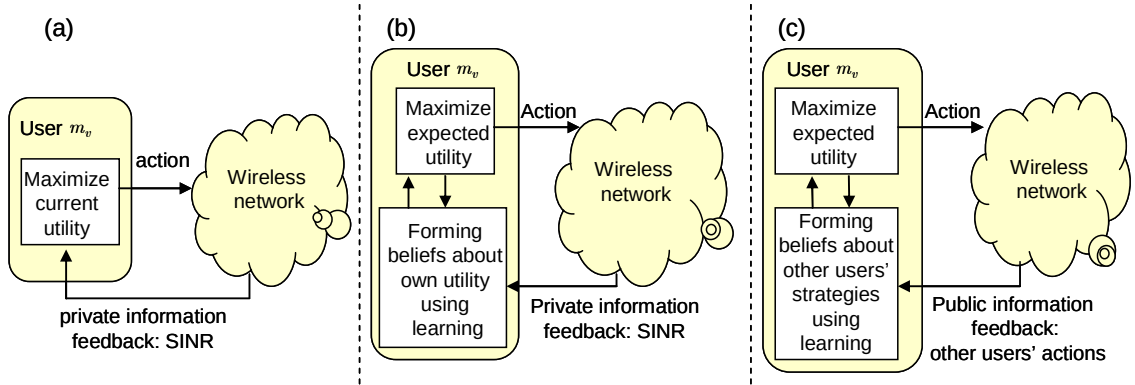


Fig. 5.1 (a) Conventional distributed power control. (b) Payoff-based interactive learning with private information feedback. (c) Model-based interactive learning with public information feedback.

¹ In this chapter we will assume that the public information is accurately transmitted. However, if it is believed that malicious users are presented in the system, mechanism design can be used to compel users to declare their information truthfully.

We focus on the problem of delay-sensitive applications sharing the same wireless network. Due to the delay sensitivity, the utility of a user is dramatically impacted by the applications of other users. This provides the user an additional incentive to adopt a better learning scheme, since it cannot wait a long time to transmit the packets. To cope with the delay sensitivity, we need to consider not only the impact of the effective throughput over the wireless network, but also the source traffic characteristics, including the source rates and the delay deadlines of the applications.

In summary, this chapter aims to make the following contributions:

- 1) **Feedback-driven interactive learning framework that outperforms the Nash equilibrium performance.** We develop a feedback-driven learning framework for distributed power control of delay sensitive users that outperforms the Nash equilibrium performance, which is achieved when users deploy myopic best response such as iterative water filling [YGC02]. Note that the users are self-interested, meaning that they tend to maximize their own utility in a fully distributed manner.
- 2) **Cost-efficiency tradeoff of interactive learning.** We consider learning solutions based on both the private and public information feedback, and characterize the cost of information feedback by explicitly considering – a) from whom (i.e. from which transmitters or receivers) this information is obtained, and b) how often such information is obtained (i.e. the frequency of getting feedback). We quantify the cost-efficiency tradeoff when learning from different information feedback and show how to adapt the information feedback to maximize the learning efficiency.
- 3) **Analytical upper bounds based on interactive learning.** We also quantify the utility upper bounds that can be achieved by a user through learning based on private or public information feedback.

The chapter is organized as follows. In Section II, we discuss the considered network settings and formulate the studied informationally-decentralized dynamic resource management problem among wireless users competing for resources with incomplete

information. In Section III, we characterize the information feedback and discuss the cost-efficiency tradeoff of the information feedback. Based on the type of information feedback, we introduce two classes of interactive learning solutions and discuss how to adjust the information feedback to improve the learning efficiency. In Section IV, payoff-based learning is discussed, which employs only private information feedback. In Section V, we introduce model-based learning, which requires public information feedback. Section VI presents simulation results and Section VII concludes the chapter.

II. NETWORK SETTINGS AND PROBLEM FORMULATION

A. Network settings

We assume that there are V users $(m_1, \dots, m_V)$ that are simultaneously transmitting delay sensitive applications over the same wireless infrastructure. A network user m_v is composed of a source node n_v^s (transmitter) and a destination node n_v^d (receiver) that can establish a direct communication connection, i.e. $m_v = \{n_v^s, n_v^d\}$. We assume that there are multiple frequency channels for users to transmit their applications and $\mathcal{F}$ is the set of all channels. An illustrative network example is depicted in Figure 5.2.

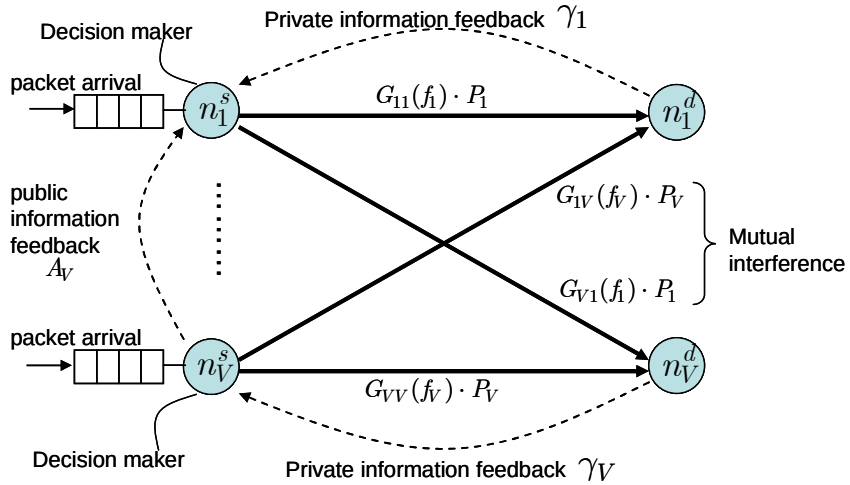


Fig. 5.2 System diagram of the dynamic joint power-spectrum resource allocation.

B. Actions and strategies

We consider a fully distributed setting where each user attempts to maximize its own utility function by selecting the optimal frequency channels and transmitted power levels in the selected channels. We assume that only frequency channels in the set $\mathcal{F}_v \subseteq \mathcal{F}$ are available to the user m_v . Network user m_v transmits its application through one of the available frequency channels $f_v \in \mathcal{F}_v$ with a power level $0 \leq P_v \leq P_v^{\max}$. In this chapter, we assume that the transmit power level can take a discrete set of values in the set $\mathcal{P}_v$. Hence, we define the action of a user m_v as $A_v = [f_v, P_v] \in \mathcal{A}_v = \mathcal{F}_v \times \mathcal{P}_v$. We assume that $S_v(A_v)$ represents the probability that a user m_v takes A_v as its action. The *strategy*² of user m_v is defined as a probability distribution $\mathbf{S}_v = [S_v(A_v), \text{ for } A_v \in \mathcal{A}_v] \in \mathcal{S}_v$, where $\mathcal{S}_v$ is a set of probability distributions over all feasible actions $A_v \in \mathcal{A}_v$.

Let $G_{vv'}(f_v)$ represent the channel gain from the transmitter $n_{v'}^s$ of the user $m_{v'}$ to the receiver n_v^d of the users m_v , which is related to the distance of the two nodes and channel characteristics. The SINR γ_v experienced by user m_v in frequency channel f_v depends on the user's action A_v and the actions of all the other users, denoted as A_{-v} :

$$\gamma_v(A_v, A_{-v}) = \frac{G_{vv}(f_v)P_v}{N_{f_v} + \sum_{v' \neq v, f_{v'} = f_v} G_{vv'}(f_v)P_{v'}}, \quad (1)$$

where N_{f_v} represents the AWGN noise level in the frequency channel f_v . The term $\sum_{v' \neq v, f_{v'} = f_v} G_{vv'}(f_v)P_{v'}$ represents the mutual interference coupling from the other users. The effective throughput available at a transmitter n_v^s depends on the experienced SINR γ_v and it is denoted as $B_v(A_v, A_{-v}) = T_v(f_v)(1 - p_v(\gamma_v))$, where $T_v(f_v)$ and $p_v(\gamma_v)$ represent the maximum transmission rate and packet error rate of user m_v using the frequency channel f_v . In Appendix A, we provide Table 5.5 to summarize the notation used in this chapter.

² The strategy defined in this chapter can be regarded as a mixed strategy and the action defined in this chapter can be regarded as a pure strategy in game theory.

C. Utility function definition

We assume that users are transmitting delay sensitive applications. The packet arrival process of a user m_v is assumed to be Poisson with the mean arrival rate λ_v . The delay deadline of the packets of user m_v is d_v . We assume that each user maintains a buffer at its transmitter and that the arriving packets which cannot be transmitted immediately will be queued in the buffer. The effective throughput $B_v(A_v, A_v)$ is independent of the packet arrival process. Hence, there will be queuing delay and transmission delay. We denote the total delay as D_v , which is a random variable depending on both arrival rate λ_v and the effective throughput $B_v(A_v, A_v)$. The packet loss rate is defined as the probability when this delay exceeds the packet delay deadline, i.e. $\text{Prob}\{D_v(\lambda_v, B_v(A_v, A_v)) > d_v\}$. Therefore, the rate of successfully received packets is $\lambda_v \times \text{Prob}\{D_v(\lambda_v, B_v(A_v, A_v)) \leq d_v\}$.

We assume the users attempt to maximize their energy-efficient utility functions (measured in bits/joule) similar to [MCP06]. The difference is that we also consider the packet loss due to the expiration of the delay deadline for delay sensitive applications. The utility function of a user m_v is

$$u_v(A_v, A_v) = \frac{\lambda_v \times \text{Prob}\{D_v(\lambda_v, B_v(A_v, A_v)) \leq d_v\}}{P_v}. \quad (2)$$

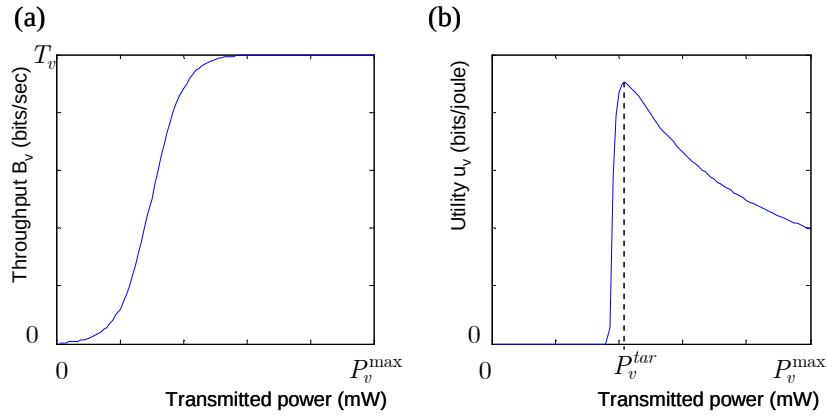


Fig. 5.3 (a) Throughput B_v vs. P_v in a selected frequency channel f_v with fixed interference.

(b) Utility u_v vs. P_v in a selected frequency channel f_v with fixed interference.

The utility function reflects the expected number of packets that is successfully received

(rather than transmitted as in [MCP06]) per joule of energy consumed for delay sensitive users. More details about how this utility function can be computed in a practical communication setting can be found in Appendix B. Figure 5.3 illustrates the utility function of a user m_v using different power $0 \leq P_v \leq P_v^{\max}$ in a selected frequency channel f_v with fixed interference. We denote the power of user m_v that maximizes the utility function when transmitting in channel f_v as $P_v^{tar}(f_v)$.

D. Problem formulation

Let $\mathbf{A}_{-v}^{myop}$ represent the latest actions of the other users observed by a user m_v in the network. Conventionally, the user m_v adopts a myopic distributed optimization, which can be formulated as:

$$A_v^{myop} = [f_v^{myop}, P_v^{myop}] = \arg \max_{A_v \in \mathcal{A}_v} u_v(A_v, \mathbf{A}_{-v}^{myop}). \quad (3)$$

In [MCP06], it was shown that the myopic best response A_v^{myop} converges to the Nash equilibrium under certain conditions on channel gains. However, if a foresighted user m_v knows the exact response actions of other users $\mathbf{A}_{-v}^{fors}(A_v)$, a better performance can be achieved [FT91]. Let $\mathbf{A}_{-v}^{fors}(A_v)$ represent the actions of the other users given that the action A_v is taken by user m_v . The optimization performed by a foresighted user can be formulated as [FT91]:

$$A_v^{fors} = [f_v^{fors}, P_v^{fors}] = \arg \max_{A_v \in \mathcal{A}_v} u_v(A_v, \mathbf{A}_{-v}^{fors}(A_v)). \quad (4)$$

Let us assume that only one user is foresighted, and all the other users in the network still adopt a myopic best response. Given the exact response actions $\mathbf{A}_{-v}^{fors}(A_v)$, the foresighted decision making based on the complete information of the other users will converge to the Stackelberg equilibrium [FT91] and the optimal utility is denoted as

$$U_v(\mathbf{A}_{-v}^{fors}(A_v)) = \max_{A_v \in \mathcal{A}_v} u_v(A_v, \mathbf{A}_{-v}^{fors}(A_v)). \quad (5)$$

However, due to the informationally-decentralized nature of the wireless networks, it is impossible for each user to know in practice the exact response actions $\mathbf{A}_{-v}^{fors}(A_v)$. Hence, accurately modeling the actions $\mathbf{A}_{-v}^{fors}(A_v)$ based on the information feedback is necessary.

Definition 1: Denote the information feedback of user m_v at time slot t as $\mathcal{Z}_v^t$, regardless whether the information feedback is private or public. We define the *observed information history* of user m_v at time slot t as $o_v^t = \{\mathcal{Z}_v^t, o_v^{t-1}\}$.

Assume that the strategy of user m_v at time slot t is denoted as $\mathbf{s}_v^t$. We use the notation $\mathbf{M}_{-v}$ to indicate the set of all users except user m_v . The strategy of all users in the network except user m_v is $\mathbf{s}_{-v}^t = \{\mathbf{s}_u^t, \text{for } m_u \in \mathbf{M}_{-v}\}$.

Definition 2: Since the exact response actions of other users $\mathbf{A}_{-v}^{fors}$ are not available to user m_v in real time, user m_v estimates $\mathbf{A}_{-v}^{fors}$ by building a *belief* on the other users' strategies $\mathbf{s}_{-v}^t$. The belief of user m_v is defined as $\tilde{\mathbf{s}}_{-v}^t(A_v) = \{\tilde{s}_{-v}^t(A_{-v} | A_v), \text{for all } A_v \in \mathcal{A}_v\}$, where $\tilde{s}_{-v}^t(A_{-v} | A_v)$ ³ is the estimated strategies of the other users given that user m_v decides to take the action A_v .

In other words, user m_v estimates the other users' strategies $\tilde{\mathbf{s}}_{-v}^t(A_v)$ for each of its action $A_v \in \mathcal{A}_v$ ⁴.

Definition 3: Assume Λ_v represents the interactive learning scheme adopted by user m_v . A *learning scheme* Λ_v is defined as a method that allows user m_v to build a belief $\tilde{\mathbf{s}}_{-v}^t = \Lambda_v(o_v^t)$ ⁵ based on the observed information history o_v^t , in order to estimate the actions of the other users $\mathbf{A}_{-v}^{fors}$.

Specifically, by learning from the observed information history o_v^t , user m_v builds its *belief* $\tilde{\mathbf{s}}_{-v}^t$ on the other users' strategies and determine its own best response strategy $\mathbf{s}_v^t$. Figure 5.4 illustrates how a delay sensitive user makes decisions based on the observed information history o_v^t and the mutual interference coupling in the dynamic wireless environment. The problem in equation (4) can be now reformulated as:

³ $\tilde{\mathbf{s}}_{-v}^t(A_v) = \{\tilde{s}_u^t(A_u \in \mathcal{A}_u | A_v), \text{for } m_u \in \mathbf{M}_{-v}\}$. $\tilde{\mathbf{s}}_u^t(A_u \in \mathcal{A}_u | A_v) = [\tilde{s}_u^t(A_u | A_v), \text{for } A_u \in \mathcal{A}_u]$

is the conditional probability distribution when user m_v takes the action A_v .

⁴ Based on different types of information feedback, user m_v may implicitly model the other users by only estimating the aggregate effect of the other users. See Section IV for more detail.

⁵ For representation convenience, we use the simplified notation $\mathbf{A}_{-v}^{fors}$ to represent $\mathbf{A}_{-v}^{fors}(A_v), A_v \in \mathcal{A}_v$ as the exact response actions of other users. And also use $\tilde{\mathbf{s}}_{-v}^t$ to represent $\tilde{\mathbf{s}}_{-v}^t(A_v)$ in the rest of the chapter.

$$\mathbf{S}_v^t(\tilde{\mathbf{S}}_{-v}^t) = \arg \max_{\mathbf{S}_v \in \mathcal{S}_v} E_{(\mathbf{S}_v, \tilde{\mathbf{S}}_{-v}^t)}[u_v(\mathbf{S}_v, \tilde{\mathbf{S}}_{-v}^t)]. \quad (6)$$

Based on the determined $\mathbf{S}_v^t$, user m_v selects an action A_v at time slot t .

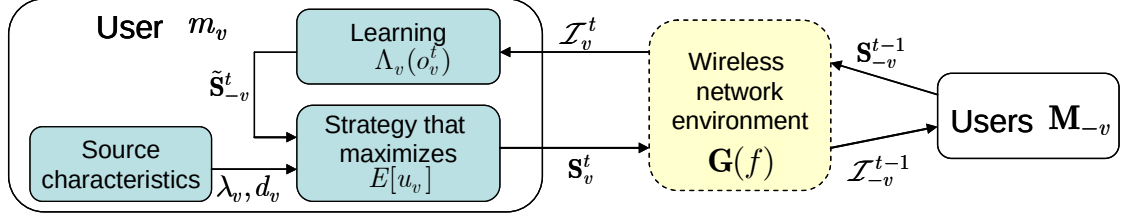


Fig. 5.4 Interactions among users and the foresighted decision making based on information feedback.

E. Learning efficiency

The performance of an interactive learning approach depends on how accurate the belief $\tilde{\mathbf{S}}_{-v}^t = \Lambda_v(o_v^t)$ can predict the actions $\mathbf{A}_{-v}^{fors}$. A more accurate prediction of $\mathbf{A}_{-v}^{fors}$ can lead to a better learning efficiency. We define the learning efficiency $J_v(\Lambda_v(o_v^t))$ of the learning approach Λ_v (based on the observed information history o_v^t) by quantifying its impact on the expected utility, i.e.

$$J_v(\Lambda_v(o_v^t)) \triangleq E_{(\mathbf{S}_v^t, \tilde{\mathbf{S}}_{-v}^t)}[u_v(\mathbf{S}_v^t, \Lambda_v(o_v^t))], \text{ where} \quad (7)$$

$$E_{(\mathbf{S}_v^t, \tilde{\mathbf{S}}_{-v}^t)}[u_v(\mathbf{S}_v^t, \Lambda_v(o_v^t))] = \sum_{A_v \in \mathcal{A}} \left[S_v^t(A_v) \times \left(\sum_{A_{-v} \in \mathcal{A}^{V-1}} \tilde{S}_{-v}^t(A_{-v} | A_v) \times u_v(A_v, A_{-v}) \right) \right]. \quad (8)$$

The notation $\tilde{S}_{-v}^t(A_{-v} | A_v)$ is used to represent the joint probability that the users $m_u \in \mathbf{M}_{-v}$ take actions A_{-v} , given that user m_v took the action A_v .

Since the belief $\tilde{\mathbf{S}}_{-v}^t$ is only a prediction for $\mathbf{A}_{-v}^{fors}$, we define the Price of Imperfect Belief (PIB) for using the learning scheme Λ_v based on the observed information history o_v^t as the performance difference between the Stackelberg equilibrium [FT91] $U_v(\mathbf{A}_{-v}^{fors})$ (where the user m_v knows the exact response of the other users) and the practical learning efficiency $J_v(\Lambda_v(o_v^t))$, i.e.

$$\Delta_P(\Lambda_v(o_v^t)) \triangleq U_v(\mathbf{A}_{-v}^{fors}) - J_v(\Lambda_v(o_v^t)). \quad (9)$$

In the next sections, we quantify the cost of the information feedback $\mathcal{Z}_v^t$ and study two classes of interactive learning approaches Λ_v^{priv} and Λ_v^{pub} based on different types of information feedback.

III. INFORMATION FEEDBACK FOR INTERACTIVE LEARNING

A. Characterization of information feedback

In this chapter, we define the *entire* information history from all users until time slot t as

$$h^t = \{\gamma_v^s, \mathbf{G}_v^s, A_v^s, \text{for } v = 1, \dots, V, s = 0, \dots, t\}. \quad (10)$$

Note that a user m_v observes only a subset of the entire history through information feedback, i.e. $o_v^t \subseteq h^t$. The observed information history o_v^t can be characterized in three distinct categories:

- **Types of information feedback** – As mentioned before, there are two types of information that a user m_v can observe at a certain time slot t , i.e. the private information feedback $\mathcal{Z}_v^{t,priv} = \{\gamma_v^{t-1}\}$ or the public information feedback $\mathcal{Z}_{-v}^{t,pub} = \{\mathbf{G}_u^{t-1}, A_u^{t-1}, \text{for } m_u \in \mathbf{M}_{-v}\}$. Recall that $o_v^t = \{\mathcal{Z}_v^t, o_v^{t-1}\}$ in Definition 1.
- **Information zone** – We define the information zone $\mathbf{V}_v^t$ as a set of users that are able to feed back information to the transmitter of user m_v at time slot t . In the wireless communication networks, the information from further users is less significant, since the effect of mutual interference coupling decreases ($G_{vv'}$ decreases in equation (1)) as the distance increases [Rap02]. Hence, user m_v can selectively collect the information only from a set of neighboring (e.g. within an information horizon as in the previous chapter) users $m_u \in \mathbf{V}_v^t$, i.e. $\mathcal{Z}_{-v}^{t,pub} = \{\mathbf{G}_u^{t-1}, A_u^{t-1}, \text{for } m_u \in \mathbf{V}_v^t\}$. Since the information zone of the private information feedback only contains user m_v itself, we define $|\mathbf{V}_v^t| = 0$ for $\mathcal{Z}_v^{t,priv} = \{\gamma_v^{t-1}\}$.
- **Information feedback frequency** – In our problem formulation in equation (6), user

m_v can obtain the information feedback and make decisions during every time slot. However, in practice, user m_v can obtain the information feedback at different time scales. Assume that user m_v observes the information feedback for every τ_v time slots ($\tau_v \in \mathbb{Z}^+$). Define $\omega_v = 1/\tau_v$ as the frequency of the information feedback, $0 \leq \omega_v \leq 1$. Let $\omega_v = 0$ represent the case when no information feedback is obtained. Let $\mathbf{T}_v^t$ represent the set of time slots before time slot t at which the user m_v obtains information and makes decisions, i.e. $\mathbf{T}_v^t = \{(s_v^0 + k\tau_v), k = 0, 1, \dots, K_v^t\}$, where s_v^0 is the initial time slot that a user m_v obtains information and starts making decisions. The number of decisions made by user m_v up to time t equals $K_v^t = \lfloor (t - s_v^0)/\tau_v \rfloor$ represents, where $\lfloor \cdot \rfloor$ is the floor operation. The observed information history now becomes $\mathcal{O}_v^t = \{\mathcal{Z}_v^s, \text{ for } s \in \mathbf{T}_v^t\}$.

Figure 5.5 illustrates examples of the overhead for different types of information feedback. Note that the information overhead varies over time, since the information feedback from different users depends on the time-varying network environment. Depending on the type of information feedback that user m_v observes, two classes of interactive learning approaches are developed. We will discuss them in more detail in Section IV and V, respectively.

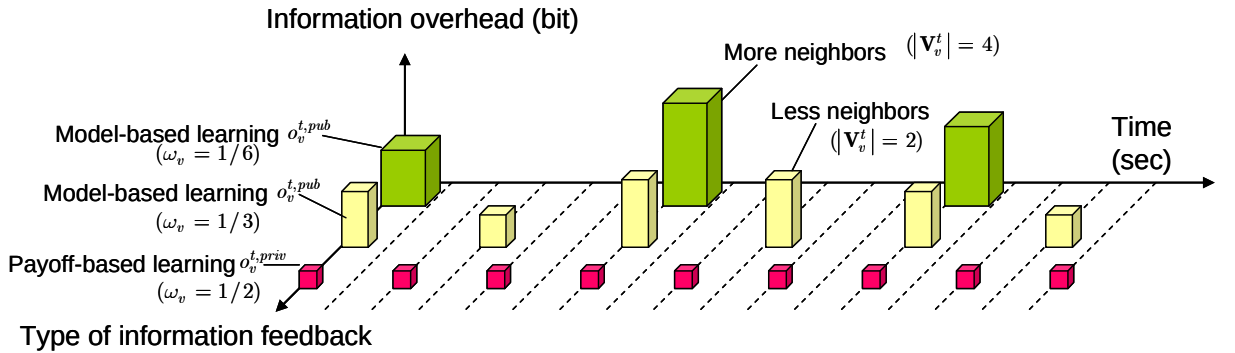


Fig. 5.5 Examples of different types of information feedback $\mathcal{Z}_v^t$.

B. Cost-efficiency tradeoff when adjusting the information feedback

Let us denote the information feedback overhead of user m_v as $\sigma_v(\omega_v, |\mathbf{V}_v|)$ ⁶, which is a function of the information feedback frequency ω_v and the number of the neighboring users $|\mathbf{V}_v|$. In general, with more frequent information feedback (i.e. a larger ω_v) or feedback from more users (i.e. a larger $|\mathbf{V}_v|$), a user can obtain more information from the entire information history h^t and hence, this results in a more accurate belief. On the other hand, a large information overheads $\sigma_v(\omega_v, |\mathbf{V}_v|)$ can degrade the learning efficiency $J_v(\Lambda_v(o_v^t))$. In this chapter, we assume that the packet transmission and the information feedback are multiplexed in the same frequency channel. Hence, considering the information overhead, the effective throughput can be represented as $B'_v(\sigma_v, A_v, A_{-v}) = B_v(A_v, A_{-v}) \times \theta(\sigma_v)$, where $0 < \theta(\sigma_v) \leq 1$ represents the fraction of time dedicated to the packet transmission, and it is a decreasing function of σ_v .

We now focus on how the learning efficiency $J_v(\Lambda_v(o_v^t))$ in equation (7) changes with different value of σ_v . If σ_v is large, the belief $\tilde{\mathbf{S}}_{-v}^t$ provides an accurate model on $\mathbf{A}_{-v}^{fors}$. Given $\mathbf{A}_{-v}^{fors}$, the utility function in equation (2) can be derived as (see Appendix C for more detail):

$$u_v(\sigma_v, A_v, A_{-v}) = \begin{cases} \frac{\lambda_v}{P_v} (1 - \frac{1}{F_v(\sigma_v, \gamma_v(A_v, A_{-v}))}), & \text{if } B'_v(\sigma_v, A_v, A_{-v}) / L_v > \lambda_v \\ 0, & \text{otherwise} \end{cases}, \quad (11)$$

$$\text{where } F_v(\sigma_v, \gamma_v(A_v, A_{-v})) \equiv \exp(B'_v(\sigma_v, \gamma_v(A_v, A_{-v})) \frac{d_v}{L_v} - \lambda_v d_v). \quad (12)$$

L_v represents the average packet length of user m_v . Note that both $B'_v(\sigma_v, A_v, A_{-v})$ and $F_v(\sigma_v, \gamma_v(A_v, A_{-v}))$ are decreasing functions of σ_v . Hence, the utility function is a non-increasing function of σ_v . In other words, the PIB Δ_P is a non-decreasing function of σ_v when σ_v is large. On the other hand, if σ_v is small, the belief $\tilde{\mathbf{S}}_{-v}^t$ provides an inaccurate model on $\mathbf{A}_{-v}^{fors}$. By having more information $o_v^t \subseteq h^t$, increasing σ_v can improve the learning efficiency and hence, Δ_P decreases. In other words, Δ_P is a

⁶ Note that for private information feedback, the information overhead σ_v only depends on ω_v ($|\mathbf{V}_v| = 0$).

non-increasing function of σ_v when σ_v is small. Note that this efficiency-cost tradeoff occurs when adjusting either ω_v or $|\mathbf{V}_v|$.

Proposition 1: For a given learning scheme Λ_v , there exists at least one optimal information feedback overhead σ_v^* such that

$$\sigma_v^*(\Lambda_v) = \arg \min_{\sigma} \Delta_P(\Lambda_v(o_v^t(\sigma))) \quad (13)$$

Proof: Note that minimizing Δ_P is the same as maximizing $J_v(\Lambda_v(o_v^t))$. Since $0 \leq J_v(\Lambda_v(o_v^t)) \leq U(\mathbf{A}_{-v}^{fors})$ is bounded, there must exist a minimum value with a certain σ_v^* . ■

Based on Proposition 1, we propose an adaptive interactive learning that adapts the information feedback parameters for user m_v to improve its learning efficiency J_v . Figure 5.6 presents the system block diagram of our adaptive interactive learning framework. Due to the consideration of the source characteristics, the interactive learning framework is operated at the application layer. The goal of user m_v in the adaptive interactive learning framework is to build the belief $\tilde{\mathbf{S}}_{-v}^t$ based on o_v^t for determining the best response strategy $\mathbf{S}_v^t$ and adjust the information feedback $\mathcal{I}_v^{t+1}(\sigma_v)$ to improve the learning efficiency $J_v(\Lambda_v(o_v^t))$. In the following sections, we will discuss the adaptive interactive learning schemes based on different types of information feedback in more details.

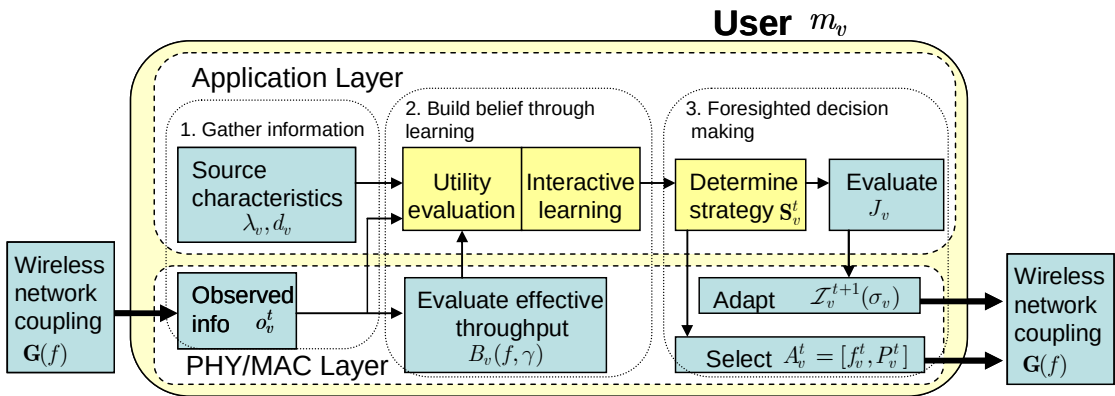


Fig. 5.6 System block diagram for the adaptive interactive learning for dynamic resource management.

IV. INTERACTIVE LEARNING WITH PRIVATE INFORMATION FEEDBACK

In the case where user m_v only observes the private information feedback $\mathcal{Z}_v^{t,priv}$, it can only model the aggregate effect of other users' actions through the experienced SINR value γ_v . Hence, it cannot model the exact response actions of the other users $\mathbf{A}_{-v}^{fors}$ explicitly. Note that the observed information history in this case is $o_v^t(\omega_v) = \{\gamma_v^{s-1}, s \in \mathbf{T}_v^t(\omega_v)\}$. Based on this observed information history $o_v^t(\omega_v)$, user m_v is aware of its past actions $A_v^{s-1}, s \in \mathbf{T}_v^t$ and the past resulting utilities $u_v(A_v^{s-1}, \gamma_v^{s-1}), s \in \mathbf{T}_v^t$. Let $\tilde{u}_v(A_v, o_v^t(\omega_v))$ represent the estimated utility of user m_v if the action A_v is taken. Instead of predicting the exact response actions $\mathbf{A}_{-v}^{fors}$ explicitly, user m_v builds a belief on the utility and determines its best strategy $\mathbf{S}_v^t$ based on its past experienced action-utility pairs $[A_v^{s-1}, u_v(A_v^{s-1}, \gamma_v^{s-1})], s \in \mathbf{T}_v^t$. Hence, user m_v does not try to estimate the probability $\tilde{S}_{-v}^t(A_{-v} | A_v)$ in equation (8). Instead, user m_v builds directly its belief on what will be the average utility impact that it will experience if it takes action A_v , i.e. $\tilde{u}_v(A_v, o_v^t(\omega_v))$ substitutes the term $\sum_{A_{-v} \in \mathcal{A}^{V-1}} \tilde{S}_{-v}^t(A_{-v} | A_v) \times u_v(A_v, A_{-v})$ in equation (8).

Let $\mathbf{S}_v^t(\omega_v) = \Lambda_v^{priv}(o_v^t(\omega_v))$ be the strategy of user m_v at time slot t learned from the observed information history $o_v^t(\omega_v)$. From equation (7), the learning efficiency of user m_v is

$$J_v(\Lambda_v^{priv}(o_v^t(\omega_v))) = \sum_{A_v \in \mathcal{A}} S_v^t(A_v) \times \tilde{u}_v(A_v, o_v^t(\omega_v)). \quad (14)$$

To minimize Δ_P in equation (9), the best response strategy is:

$$\mathbf{S}_v^t(\omega_v) = \arg \max_{\mathbf{S}_v \in \mathcal{S}_v} \sum_{A_v \in \mathcal{A}} S_v(A_v) \times \tilde{u}_v(A_v, o_v^t(\omega_v)). \quad (15)$$

The payoff-based learning based on private information feedback can be represented equation (15). After the strategy $\mathbf{S}_v^t$ is determined, the action of user m_v at time slot t is determined by

$$A_v^t = \text{Rand}(\mathbf{S}_v^t), \quad (16)$$

where $\text{Rand}(\mathbf{S}_v^t)$ represents a random selection based on the probabilistic strategy

$\mathbf{s}_v^t \in \mathcal{S}_v$. Payoff-based learning [You04] provides a method to learn the strategy $\mathbf{s}_v^t$ from the past experienced action-utility pairs $[A_v^{s-1}, u_v(A_v^{s-1}, \gamma_v^{s-1})], s \in \mathbf{T}_v^t$. A simple example of a payoff-based learning method will be provided in Section IV.A.

If the private information feedback is costless (i.e. $B'_v = B_v$ in equation (11)), the utility upper bound of the payoff-based learning can be calculated based on the resulting strategy $\mathbf{s}_v^* = [S_v^*(A_v), \text{ for all } A_v \in \mathcal{A}_v]$ at convergence.

Proposition 2 For a payoff-based learning with private information feedback, if the information feedback is costless, the upper bound of the learning efficiency $J_v(\Lambda_v^{priv})$ is $(1 - \varepsilon_v(\Lambda_v^{priv}))U_v(\mathbf{A}_{-v}^{fors})$, and $0 \leq \varepsilon_v(\Lambda_v^{priv}) < 1$.

$$\varepsilon_v(\Lambda_v^{priv}) = \frac{1}{U_v(\mathbf{A}_{-v}^{fors})} \sum_{A_v \in \mathcal{A}_v} g(A_v) u_v(A_v, \mathbf{A}_{-v}^{fors}), \text{ where } g(A) = \begin{cases} 1 - S_v^*(A), & \text{for } A = A_v^{fors} \\ -S_v^*(A), & \text{otherwise} \end{cases}. \quad (17)$$

Proof: By substituting equation (14) into equation (9), the PIB becomes $\Delta_P(\Lambda_v^{priv}) = \varepsilon_v(\Lambda_v^{priv})U_v(\mathbf{A}_{-v}^{fors})$. Since $u_v(A_v, \mathbf{A}_{-v}^{fors})$ has costless information feedback, substituting $\tilde{u}_v(A_v, o_v^t(\omega_v))$ by $u_v(A_v, \mathbf{A}_{-v}^{fors})$ provides a lower bound on $\Delta_P(\Lambda_v^{priv})$, which is $\sum_{A_v \in \mathcal{A}_v} g(A_v) u_v(A_v, \mathbf{A}_{-v}^{fors})$. ■

From equation (17), in order to decrease $\varepsilon_v(\Lambda_v^{priv})$, user m_v needs to increase the accuracy of the best response strategy $\mathbf{s}_v^*$ such that it approaches A_v^{fors} . Next, let us give a simple example using a well-known reinforcement learning solution [You04].

A. Reinforcement learning based on private information feedback

In this subsection, let us assume $\omega_v = 1$. By applying typical reinforcement learning, user m_v models its best response strategy $\mathbf{s}_v^t$ as

$$S_v^t(A_v) = \frac{r_v^t(A_v)}{\sum_{A_v \in \mathcal{A}_v} r_v^t(A_v)}, \quad (18)$$

where $r_v^t(A_v)$ represents the *propensity* [You04] of user m_v choosing an action A_v at time slot t . Let us define $\mathbf{r}_v^t = [r_v^t(A_v), \text{ for } A_v \in \mathcal{A}_v]$ as a vector of propensity of all feasible actions. The user updates $\mathbf{r}_v^t$ based on the experienced utility, $u_v(A_v^{t-1}, \gamma_v^{t-1})$ when the action A_v^{t-1} is taken at time slot $t - 1$. Here, we adopt the cumulative payoff

matching [You04]:

$$\mathbf{r}_v^t = \alpha \times \mathbf{r}_v^{t-1} + (1 - \alpha) u_v(A_v^{t-1}, \gamma_v^{t-1}) \times \mathbf{I}(A_v^{t-1}), \quad (19)$$

where α is the discount factor for the history value of the cumulative propensity.

$\mathbf{I}(A_v^t) = [I(A = A_v^t), \text{ for } A \in \mathcal{A}_v]$ represents an indicator vector such that

$$I(A = A_v^t) = \begin{cases} 1, & \text{if } A = A_v^t \\ 0, & \text{if } A \neq A_v^t \end{cases}. \quad (20)$$

B. Adaptive reinforcement learning

The reinforcement learning in the previous subsection fixes $\omega_v = 1$, i.e. user m_v obtains information feedback at each time slot. From Proposition 1, we know that by adjusting information feedback frequency ω_v to ω_v^* , user m_v can minimize its PIB Δ_P . Hence, we introduce the adaptive reinforcement learning that adjusts ω_v to maximize the learning efficiency $J_v(\Lambda_v^{priv})$. Specifically, for $\omega_v < 1$, user m_v will not receive the private information feedback at each time slot with probability $1 - \omega_v$. If there is no information feedback, user m_v takes the *baseline action* A_v^{base} , which is the past action that ever provides the best payoff value. Smaller ω_v means that the user is more reluctant to deviate from its baseline action and leads to a lower information feedback overhead. With probability ω_v , the user will receive the information feedback and perform the same reinforcement learning as in the previous subsection. After user m_v selects an action A_v^t , it compares the payoff value u_v and then updates the record of the baseline action A_v^{base} and the baseline payoff value u_v^{base} :

$$A_v^{base} = \begin{cases} A_v^{t-1}, & \text{if } u_v(A_v^{t-1}, \gamma_v^{t-1}) > u_v^{base} \\ A_v^{base}, & \text{otherwise} \end{cases}. \quad (21)$$

$$u_v^{base} = \max(u_v^{base}, u_v(A_v^{t-1}, \gamma_v^{t-1})). \quad (22)$$

Finally, user m_v evaluates the learning efficiency $J(\Lambda_v(o_v^t))$ and changes the information feedback frequency ω_v by $\Delta\omega_v$ until the maximum $J(\Lambda_v(o_v^t))$ is found. The details of the proposed adaptive reinforcement learning can be found in Algorithm 5.1 in Appendix D.

V. INTERACTIVE LEARNING WITH PUBLIC INFORMATION FEEDBACK

Unlike the payoff-based learning, when user m_v observes public information feedback $\mathcal{Z}_{-v}^{t, pub} = \{\mathbf{G}_u^{t-1}, A_u^{t-1}, \text{ for } m_u \in \mathbf{M}_{-v}\}$, the observed information history is $o_v^t = \{\mathcal{Z}_{-v}^{s, pub}, s \in \mathbf{T}_v^t\}$. Based on this, user m_v can directly model the strategy of other users and build belief $\tilde{\mathbf{S}}_{-v}^t$ on it explicitly.

Let $\tilde{\mathbf{S}}_{-v}^t(\sigma_v) = \Lambda_v^{pub}(o_v^t(\sigma_v))$. From equation (7), the learning efficiency is

$$J_v(\Lambda_v^{pub}(o_v^t(\sigma_v))) = \sum_{A_v \in \mathcal{A}} \left(S_v^t(A_v) \times \left(\sum_{A_{-v} \in \mathcal{A}^{V-1}} \tilde{S}_{-v}^t(A_{-v} | A_v) \times u_v(\sigma_v, A_v, A_{-v}) \right) \right). \quad (23)$$

To minimize the Δ_P in equation (9), the best response strategy of user m_v is to take the action ($\mathbf{S}_v^t = \mathbf{I}(A_v^t)$):

$$A_v^t(\sigma_v) = \arg \max_{A_v \in \mathcal{A}_v} E_{\tilde{\mathbf{S}}_{-v}^t} [u_v(A_v, \tilde{\mathbf{S}}_{-v}^t(\sigma_v))]. \quad (24)$$

Model-based learning [You04] provides a method to build the belief on $\tilde{\mathbf{S}}_{-v}^t(\sigma_v)$ of other users' actions from the past experienced public information $A_u^{s-1}, s \in \mathbf{T}_v^t$. We present the action learning that performs equation (24) as an example in Section V.A.

Similarly, if the public information feedback is costless (i.e. $B_v' = B_v$ in equation (11)), the utility upper bound of the model-based learning can be calculated as discussed below.

Proposition 3 For the model-based learning based on the public information feedback, if the information feedback is costless, the upper bound of the learning efficiency $J_v(\Lambda_v^{pub})$ is $U_v(\mathbf{A}_{-v}^{fors})$.

Proof: Substitute equation (24) into equation (23) and substitute $u_v(\sigma_v, A_v, A_{-v})$ by $u_v(A_v, \mathbf{A}_{-v}^{fors})$. And this provides an upper bound on $J_v(\Lambda_v^{pub})$, since $u_v(A_v, \mathbf{A}_{-v}^{fors}) \geq u_v(\sigma_v, A_v, A_{-v})$. Equation (23) then becomes

$$\max_{A_v \in \mathcal{A}_v} \left(u_v(A_v, \mathbf{A}_{-v}^{fors}) \times \left(\sum_{A_{-v} \in \mathcal{A}^{V-1}} \tilde{S}_{-v}^t(A_{-v} | A_v) \right) \right) = u_v(A_v^{fors}, \mathbf{A}_{-v}^{fors}) = U_v(\mathbf{A}_{-v}^{fors}). \quad \blacksquare \quad (25)$$

The reason why the model-based learning with public information feedback has a higher upper bound compared to the payoff-based learning with private information

feedback is because it enables the user to explicitly model the actions of other users and hence, the user can directly choose the action that maximizes its expected utility. Next, we provide a simple model-based learning – action learning, which is similar to the well-known fictitious play [You04].

A. Action learning based on public information feedback

Recall that in order to build the belief $\tilde{\mathbf{S}}_{-v}^t$ from $\mathcal{O}_v^t = \{\mathcal{Z}_{-v}^{s, pub}, s \in \mathbf{T}_v^t\}$, user m_v maintains a set of strategy vectors $\tilde{S}_{-v}^t(A_{-v} | A_v) = \{\tilde{S}_u^t(A_u \in \mathcal{A}_u | A_v), \text{ for } m_u \in \mathbf{M}_{-v}\}$ for all possible actions $A_v \in \mathcal{A}_v$, where $\tilde{S}_u^t(A_u \in \mathcal{A}_u | A_v) = [\tilde{r}_u^t(A_u | A_v), \text{ for } A_u \in \mathcal{A}_u]$ represents the estimated strategy of the user $m_u \in \mathbf{M}_{-v}$ given that user m_v taking action A_v at time slot t . Hence, in the action learning, whenever action A_v is taken by the user m_v , we set

$$\tilde{S}_u^t(A_u | A_v) = \frac{r_u^t(A_u | A_v)}{\sum_{A \in \mathcal{A}_u} r_u^t(A | A_v)}, \quad (26)$$

where $r_u^t(A_u | A_v)$ is the *propensity* of user m_u at time t . The propensity represents the number of times that user m_u takes action A_u given that user m_v took action A_v . Hence, whenever the action A_v is taken by user m_v , the vector $\mathbf{r}_u^t(A_u \in \mathcal{A}_u | A_v) = [r_u^t(A_u | A_v), \text{ for all } A_u \in \mathcal{A}_u]$ is updated by:

$$\mathbf{r}_u^t(A_u \in \mathcal{A}_u | A_v) = \mathbf{r}_u^{t-1}(A_u \in \mathcal{A}_u | A_v) + \mathbf{I}(A_u^{t-1}). \quad (27)$$

Then, the probability $\tilde{S}_u^t(A_u | A_v)$ represents the *empirical frequency* that user m_u will take an action $A_u \in \mathcal{A}_u$ given that user m_v took an action A_v .

Next, we show how to maximize $E_{\tilde{\mathbf{S}}_{-v}^t}[u_v(A_v, \tilde{\mathbf{S}}_{-v}^t(\sigma_v))]$ in equation (24) analytically given the belief $\tilde{\mathbf{S}}_{-v}^t$. First, we show the necessary condition for user m_v to maximize its utility function.

Proposition 4: For a certain frequency channel f , in order to maximize $u_v(f)$, user m_v needs to transmit at the target SINR value $\gamma_v^{tar}(f)$, which is the unique positive solution of $\gamma \frac{\partial B'_v(\gamma)}{\partial \gamma} = \frac{L_v}{d_v}(F_v(\gamma) - 1)$ ($F_v(\gamma)$ is in equation (12)).

Proof: See Appendix C.

Proposition 4 suggests that if user m_v is using the frequency channel f , it should adapt the target power level $P_v^{tar}(f)$ accordingly to the interference from the other users using the same frequency channel to support the target SINR value $\gamma_v^{tar}(f)$. Since the power level in our setting is discrete, we choose the $P_v^{tar}(f) \in \mathcal{P}_v$ as the power that provides the nearest SINR value to $\gamma_v^{tar}(f)$. If the target SINR $\gamma_v^{tar}(f)$ requires a power higher than $P_v^{\max}$ (when the interference in the channel is too high), set $P_v^{tar}(f)$ to $P_v^{\max}$.

Next, given the target $P_v^{tar}(f)$, we further determine the optimal frequency channel selection of the user m_v .

Proposition 5: Let $F_v^{tar}(f) = F_v(f, \gamma_v^{tar}(f))$ in equation (12). Given the corresponding target $P_v^{tar}(f)$, the optimal action A_v^* of a user m_v is

$$f_v^* = \arg \min_{f \in \mathcal{F}_v} \{P_v^{tar}(f) \times \frac{F_v^{tar}(f)}{F_v^{tar}(f) - 1}\} \text{ and } P_v^* = P_v^{tar}(f_v^*). \quad (28)$$

Proof: From Proposition 4, maximizing $u_v = \lambda_v / P_v (1 - 1/F_v)$ leads to equation (28).

In summary, user m_v selects the frequency channel f_v^* and power level P_v^* to support the target SINR $\gamma_v^{tar}(f_v^*)$, which maximizes the utility function in equation (2). This requires user m_v to estimate the interference from other users, which can be computed by user m_v based on its belief Θ_{-v}^t . Specifically, denote the estimated interference of user m_v as $\Omega_v(A_v)$, when the action A_v is taken. Given Θ_{-v}^t , $\Omega_v(A_v)$ can be computed as:

$$\Omega_v(A_v) = \sum_{\substack{u \neq v \\ A_u \in \mathcal{A}_u}} G_{uv}(f_v) [\tilde{S}_u^t(A_u | A_v) \times P_u \times I(f_u = f)]. \quad (29)$$

Then, the resulting SINR value $\gamma_v(A_v)$ is ($A_v = [f_v, P_v]$):

$$\gamma_v(f_v, P_v) = \frac{G_{vv}(f_v)P_v}{N_{f_v} + \Omega_v(A_v)}. \quad (30)$$

By applying Proposition 4, we calculate the target power $P_v^{tar}(f)$ in different frequency channels:

$$P_v^{tar}(f) = \min_{P \in \mathcal{P}_v} |\gamma_v^{tar}(f) - \gamma_v(f, P)|. \quad (31)$$

Then we apply Proposition 5 to determine $A_v^t = [f_v^*, P_v^*]$.

B. Adaptive action learning

For the action learning in the previous subsection, the public information feedback $\mathcal{Z}_{-v}^{t, pub} = \{\mathbf{G}_u^{t-1}, A_u^{t-1}, m_u \in \mathbf{M}_{-v}\}$ is required from every user in the network, during each time slot. This results in heavy information overhead. Moreover, the overall action space $\mathcal{A}^{V-1}$ makes the computational complexity prohibitive to model all the users in the network. To approach the upper bound $U_v(\mathbf{A}_{-v}^{fors})$ of the model-based learning efficiency, we need to adjust the information overhead $\sigma_v(\omega_v, |\mathbf{V}_v|)$ by changing the information feedback parameters ω_v and $|\mathbf{V}_v|$.

Hence, in our proposed active action learning, to reduce the overhead, we classify the neighboring users of user m_v into H groups ($1 \leq H \leq |\mathbf{M}_{-v}|$) and assign different information feedback frequency ω_v^i to different groups (i.e. $1 \geq \omega_v^1 \geq \omega_v^2 \geq \dots \geq \omega_v^H \geq 0$). For the dynamic power/spectrum management problem in this chapter, the neighboring users can be classified based on their average channel gains $\bar{G}_{uv}$ over the frequency channels, i.e. $\bar{G}_{uv} = (1/|\mathcal{F}|) \sum_{f \in \mathcal{F}} G_{uv}(f)$ (from the transmitter of the neighboring user m_u to the receiver of the foresighted user m_v), since these channel gains directly impact the user's utility (see equation (1) and (2)). For instance, a neighboring user m_u with a larger channel gain $\bar{G}_{uv}$ will have more impact on u_v .

Let X_v^i represents the number of users in the group $H_i, i = 1, \dots, H$. Assume the neighboring users are relabeled according to its average channel gain value, i.e. $\bar{G}_{[1]v} \geq \bar{G}_{[2]v} \geq \dots \geq \bar{G}_{[V-1]v}$. Then,

$$m_{[u]} \in H_i, \text{ iff } \sum_{j=1}^{i-1} X_v^j \leq [u] \leq \sum_{j=1}^i X_v^j \quad (32)$$

In Algorithm 5.2 in Appendix D, we provide our adaptive action learning approach for the extreme case when $H = 2$ as an example. In this case, we only need to adapt $|\mathbf{V}_v|$ ($X_v^1 = |\mathbf{V}_v|$ and $X_v^2 = V - 1 - |\mathbf{V}_v|$). If the neighboring users $m_u \in \mathbf{V}_v$, we set $\omega_v = 1$, otherwise, $\omega_v = 0$. Meaning that user m_v only needs to model the users in set $\mathbf{V}_v$ based on $\mathcal{Z}_{-v(\mathbf{V}_v)}^{t, pub} = \{\mathbf{G}_u^{t-1}, A_u^{t-1}, m_u \in \mathbf{V}_v\}$. In Table 5.1, we compare the two

proposed interactive learning algorithms.

TABLE 5.1 COMPARISONS OF THE PROPOSED LEARNING ALGORITHMS.

	Information feedback	Build belief on	Adapt to	Performance upper bounds
Adaptive Reinforcement Learning (payoff-based)	Private	Own utility $\tilde{u}_v$	Other users' actions A_{-v} , information feedback frequency ω_v	$(1 - \varepsilon_v)U_v(\mathbf{A}_{-v}^{fors})$
Adaptive Action Learning (model-based)	Public	Other users' strategies $\tilde{\mathbf{S}}_{-v}$	Other users' actions A_{-v} , number of neighbor users $ \mathbf{V}_v $	$U_v(\mathbf{A}_{-v}^{fors})$

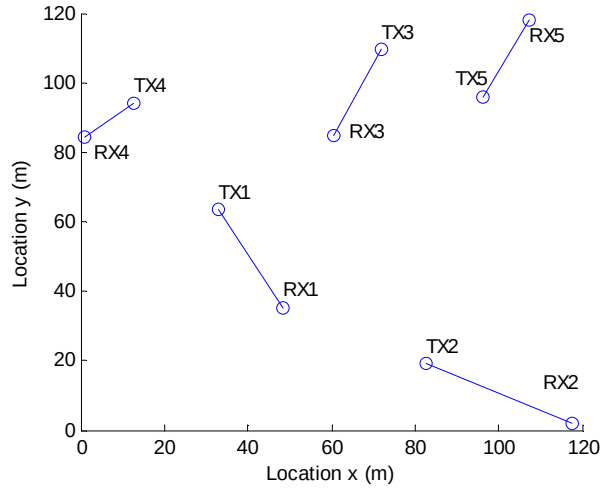


Fig. 5.7 Topology settings for the simulation.

VI. SIMULATION RESULTS

We simulate an ad hoc wireless network environment shown in Figure 5.7 with 5 users (distinct transmitter-receiver pairs) and 3 frequency channels. The frequency channels are accessible for all the users, i.e. $\mathcal{F}_v = \mathcal{F}$, for $\forall m_v$. Each user can choose its power level P_v from a set $\mathcal{P} = \{20, 40, 60, 80, 100\}$ (mW). Hence, there are a total of 15 actions A_v for users to adapt. At the physical layer, we model the channel gain between different network nodes using $G_{vv'} = K_0 \times (dis_{vv'} / dis_0)^{-\alpha}$ for all frequency channels, where $dis_{vv'}$ represents the distance from the transmitter of the user m_v to the receiver of the user $m_{v'}$, and $K_0 = 5 \times 10^{-4}$, $N_f = 1 \times 10^{-5}$, $dis_0 = 10$, $\alpha = 2$ are constants. For the

application layer parameters, we set the average packet length $L_v = 1000$ bytes, input rate $R_v = 500$ Kbps ($\lambda_v = R_v / L_v$), and delay deadline $d_v = 200$ msec for all the users. The effective transmission rate $B'_v(\gamma_v) = T \times (1 - p_v(\gamma_v)) \times \sigma_v(\beta_v)$, where $p_v(\gamma_v)$ represents the packet error rate (see Appendix B).

A. Comparison among different learning approaches

We show the simulation results using five different schemes when the physical transmission rate are $T = 700$ Kbps and 2100 Kbps in Table II and III, respectively. The five schemes are – 1) the centralized optimal (CO) 2) the theoretical upper bound $U(\mathbf{A}_{-v})$ (UB) 3) myopic best response without learning (NE), 4) user m_1 adopting adaptive reinforcement learning with private information feedback in Algorithm 1 (AR), and 5) user m_1 adopting adaptive action learning with public information feedback in Algorithm 2 (AA). The CO scheme provides the global optimal results for the overall utilities. In the NE scheme, each user attempt to maximize its current utility function based on the actions they observe in the previous time slot as in equation (3). The UB is computed from equation (4) for m_1 given the exact response of the other four users ($u_1 = U(\mathbf{A}_{-1}^{fors})$). Since, the user m_1 is in the middle of the topology, we select m_1 to be the foresighted user who learns from the information feedback. Each simulation result is averaged over 500 time slots in the dynamic network settings with mutual interference in equation (1).

Table 5.2 shows that user m_1 stays in channel 1 in both the CO and UB scheme while the other four users using the rest of two channels. However, since users are self-interested, NE scheme shows that user m_5 also attempts to transmit in channel 1 and hence, the utility u_1 decreases and forces user m_1 to increase its power level. If user m_1 becomes foresighted, as shown in the AR scheme, it will keep using the highest power level to prevent user m_5 from using its channel. The resulting utility u_1 is higher than the NE scheme. Using the AA scheme, users are able to exploit the spectrum more efficiently, due to the ability that the users can better model the strategies of other

interference sources in the network. However, this requires significant information overhead, which results in a worse performance at low bandwidth, i.e. when $T = 700$ Kbps. Note that although only user m_1 is learning, the average utility of using interactive learning schemes outperforms the myopic NE scheme. Even in a non-cooperative setting, this foresighted user actually benefits the overall system performance.

TABLE 5.2 SIMULATION RESULTS OF THE FIVE SCHEMES WHEN $T = 700$ KBPS.

$T = 700$ Kbps		Actions $A_v = [f_v, P_v]$ (or strategies S_v)	u_v (Kbit/joule)	$\sum_{v=1}^5 u_v / 5$
1) Centralized Optimal (CO)	m_1	[1,3]	1022.8	1420.8
	m_2	[2,1]	0	
	m_3	[3,2]	1479.5	
	m_4	[2,1]	3096.7	
	m_5	[2,2]	1499.8	
2) Theoretical Upper Bound (UB)	m_1	[1,3]	$U_1(\mathbf{A}_{-v}^{fors}) = 1022.8$	1285.0
	m_2	[2,4]	0	
	m_3	[2,4]	765.3	
	m_4	[3,1]	3100.8	
	m_5	[3,2]	1536.1	
3) Myopic Best Response (NE)	m_1	[1,3] x 65%, [1,5] x 35%	519.0	890.15
	m_2	[2,5] x 65%, [3,5] x 35%	195.2	
	m_3	[3,2]x33%, [3,3]x33%, [3,5]x33%	530.6	
	m_4	[2,1]x65%, [3,1]x35%	2073.0	
	m_5	[2,2]x33%, [2,3]x33%, [1,3]x33%	1132.9	
4) Adaptive Reinforcement Learning at m_1 (AR)	m_1	[1,5]	555.2	1005.6 ($\omega_v = 0.7$)
	m_2	[2,5]	113.5	
	m_3	[3,5]	345.6	
	m_4	[2,1]	2830.2	
	m_5	[2,3]	1183.7	
5) Adaptive Action Learning at m_1 (AA)	m_1	[1,3]x65%, [1,4]x27%, [1,5]x8%	529.3	1039.3 ($ \mathbf{V}_v = 2$)
	m_2	[2,5] x 85%, [3,5] x 15%	445.6	
	m_3	[3,2]x45%, [3,3]x45%, [3,5]x10%	446.8	
	m_4	[2,1]x50%, [3,1]x50%	2771.2	
	m_5	[2,2]x10%, [2,3]x10%, [1,3]x80%	1003.3	

When $T = 2100$ Kbps, Table 5.3 shows that users are now selecting a lower power levels, since the physical transmission bandwidth is sufficient. Using the AR scheme, user m_1 again occupies channel 1 by using higher power level compared to the UB

scheme. Note that using the AA scheme, user u_1 can almost reach the theoretical upper bound, since the cost of information feedback is comparatively small when $T = 2100$ Kbps. Again, the average utilities of the adaptive interactive learning schemes outperform the myopic NE scheme. The higher T gives a better learning environment for the user m_1 using AA scheme to approach the theoretical upper bound $U_1(\mathbf{A}_{-v}^{fors})$ than using AR scheme. Since all the users are selfish (including user m_1 who is learning), the learning user m_1 will benefit itself by suppressing the utility of m_2 as shown in Table 5.3. This situation is not seen in Table 5.2, since the learning environment is bad for AA scheme when the T is small.

TABLE 5.3 SIMULATION RESULTS OF THE FIVE SCHEMES WHEN $T = 2100$ KBPS.

$T = 2100$ Kbps		Actions $A_v = [f_v, P_v]$ (or mixed strategies $\mathbf{S}_v$)	u_v (Kbit/joule)	$\sum_{v=1}^5 u_v / 5$
1) Centralized Optimal (CO)	m_1	[1,2]	1562.2	1718.7
	m_2	[2,4]	781.2	
	m_3	[3,2]	1562.5	
	m_4	[2,1]	3125.0	
	m_5	[2,2]	1562.5	
2) Theoretical Upper Bound (UB)	m_1	[1,2]	$U_1(\mathbf{A}_{-v}^{fors}) =$ 1562.2	1458.3
	m_2	[2,3]	76.8	
	m_3	[2,3]	1041.7	
	m_4	[3,1]	3125.0	
	m_5	[3,2]	1562.5	
3) Myopic Best Response (NE)	m_1	[1,2]x25%, [1,3]x25%, [2,2]x25%, [2,3]x25%	523.4	1380.7
	m_2	[1,3]x25%, [1,4]x25%, [2,3]x25%, [2,4]x25%	390.6	
	m_3	[1,2]x25%, [1,3]x25%, [2,2]x25%, [2,3]x25%	1302.1	
	m_4	[3,1]	3125.0	
	m_5	[3,2]	1562.5	
4) Adaptive Reinforcement Learning at m_1 (AR)	m_1	[1,3]	1018.2	1503.6 ($\omega_v = 1$)
	m_2	[2,4]	757.8	
	m_3	[2,3]	1054.7	
	m_4	[3,1]	3125.0	
	m_5	[3,2]	1562.5	
5) Adaptive Action Learning at m_1 (AA)	m_1	[1,2]x50%, [2,2]x50%	1549.1	1455.7 ($ \mathbf{V}_v = 4$)
	m_2	[1,3] x50%, [2,3]x50%	0	
	m_3	[1,3] x50%, [2,3]x50%	1041.7	
	m_4	[3,1]	3125.0	
	m_5	[3,2]	1562.5	

B. Convergence of the learning approaches

In order to show the convergence of the proposed learning approaches, in Figure 5.8, we simulate the time plot of the two proposed learning algorithms (AR and AA) and the best response scheme without learning (NE). The network settings are the same as Table II when $T = 700$ Kbps. It is shown that both the two proposed learning schemes outperform the myopic best response scheme in terms of the average utility. The convergence speed of the AR scheme is about three times slower than the myopic best response (which converges to Nash equilibrium in about 5 time slots), while the AA scheme is about six times slower. The convergence speed of the AR scheme is faster than the AA scheme, since the AR scheme only need to build belief on its own utility. The AA scheme needs to build beliefs on its neighboring users' strategies, which leads to a slower convergence speed.

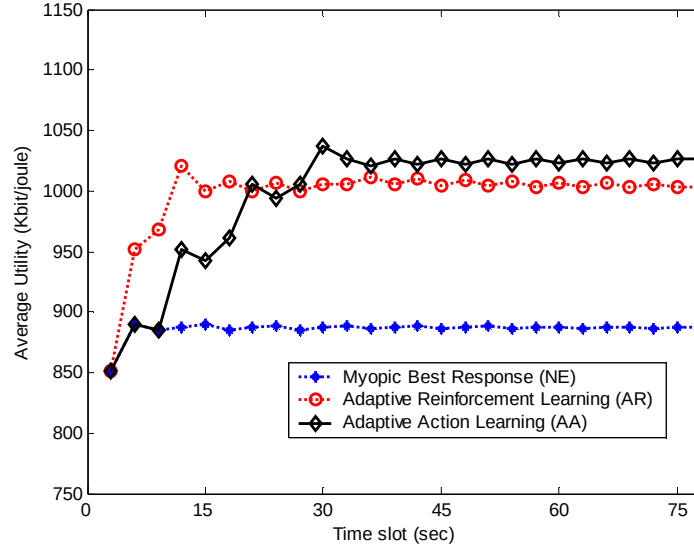


Fig. 5.8 Average utility vs. time slot of the proposed algorithms when $T = 700$ Kbps.

C. Adaptive reinforcement learning using different time scales

The reinforcement learning is very sensitive to the initial status of users' actions. Hence, in our simulations, we first train the user m_1 's initial strategy by performing

myopic best response in the first 20 time slots. Then, we simulate the reinforcement learning with different values of ω_v in Figure 5.9 for different T . Since the input rates of the applications are fixed to 500 Kbps, the utility will saturate as the bandwidth increases. The UB scheme has another saturation when T becomes larger than 1.1 Mbps, since the larger bandwidth enables another set of actions for the users. Note that when $\omega_1 = 1$, the reinforcement learning learns the transmission strategy S_1^t at every time slot. The simulation results show that the performance of $\omega_1 = 0.8$ is better than $\omega_1 = 1$ when the physical bandwidth is lower than 1Mbps, since learning at a slower pace can reduce the overhead of the private information feedback. The results in Figure 9 show that the proposed adaptive reinforcement learning operates on the envelope of the solutions obtained for different ω_1 , with $\omega_1 \in [0.5, 1]$. Hence, the performance of user m_1 using the adaptive reinforcement learning becomes closer to the upper bound.

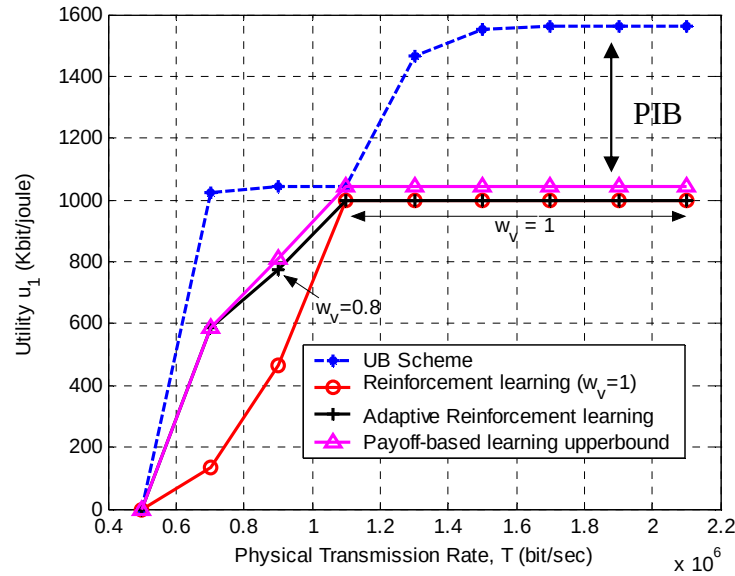


Fig. 5.9 Performance of user m_1 adopting adaptive reinforcement learning with private information feedback using different ω_1 .

D. Adaptive action learning from different neighboring users

In Figure 5.10, we also simulate the case that the action learning models the strategy of the nearest $|\mathbf{V}_v| = 2$ users instead of $|\mathbf{M}_{-v}| = 4$ users. With smaller $|\mathbf{V}_v|$, fewer neighbors need to feed back information and hence, results in less information overhead. The simulation results show that modeling users from public information feedback can improve the performance for user m_1 . However, when the physical transmission rate is lower than 1.1 Mbps, the required information overhead degrades the performance significantly and hence, it is essential to adapt the number of neighbors in the action learning to model less users in the network. The results show that using the proposed adaptive action learning, the performance of user m_1 with public information feedback becomes closer to the upper bound.

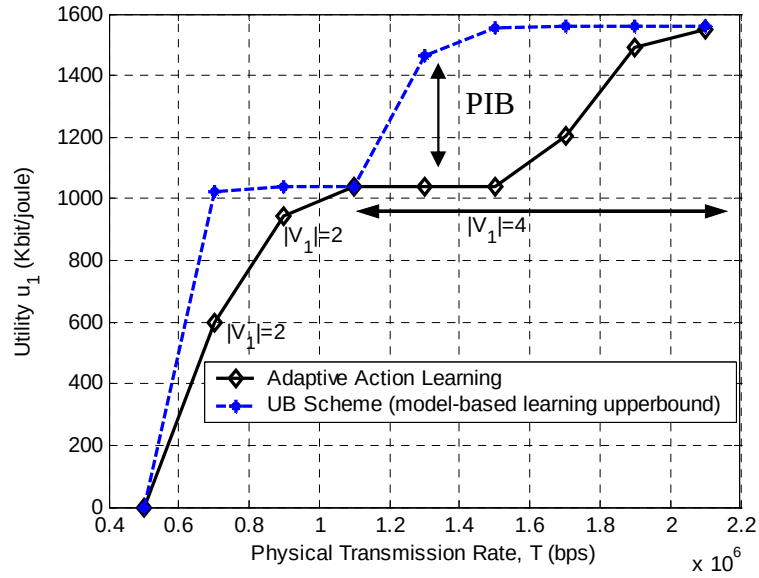


Fig. 5.10 Performance of user m_1 adopting adaptive action learning with public information feedback using different $|\mathbf{V}_1^t|$.

E. Mobility effect on the interactive learning efficiency

In the previous subsections, all the simulation results are based on the fixed topology shown in Figure 5.7. In this subsection, we simulate the case that all 5 receivers moves according to a well-known mobility model “random walk” [CBD02] – receivers randomly select a direction at each time slot and move at a fixed speed ν . Starting from the topology in Figure 5.7, Figure 5.11 shows the learning efficiency over time of the AR, AA, and NE schemes for $\nu=0.5, 1, 2$ (meter/time slot) with $T = 2100$ Kbps. It is shown that the AA scheme has higher learning efficiency on average, since user m_1 is able to obtain the channel gain information (which is directly affected by the mobility) of the other users from the public information feedback. Moreover, as expected, as the mobility increases, the learning efficiency decreases because the receivers are moving further apart. Especially for the reinforcement learning without explicit channel gain information, the results show that the performance can be worse than myopic best response, since the learning cannot keep up with the topology changes and the user’s belief about the other users becomes inaccurate when the mobility is high.

VII. CONCLUSIONS

In this chapter, we provide an adaptive interactive learning framework for delay sensitive users to adapt their frequency channel selections and power levels in wireless networks in a decentralized manner. We show that a foresighted user can improve its utility significantly learning from the information feedback. We determine performance upper bounds for the user’s utility when learning from private or public information feedback, respectively. The simulation results show that the proposed adaptive interactive learning can significantly improve the performance of delay sensitive users compared to the myopic best response. It is shown that even when only one user learns from its information feedback, the overall performance can be better than the Nash equilibrium resulting from the myopic best response. Especially, if the available system bandwidth is not limited, the proposed adaptive action learning with public information feedback

approaches the utility upper bound.

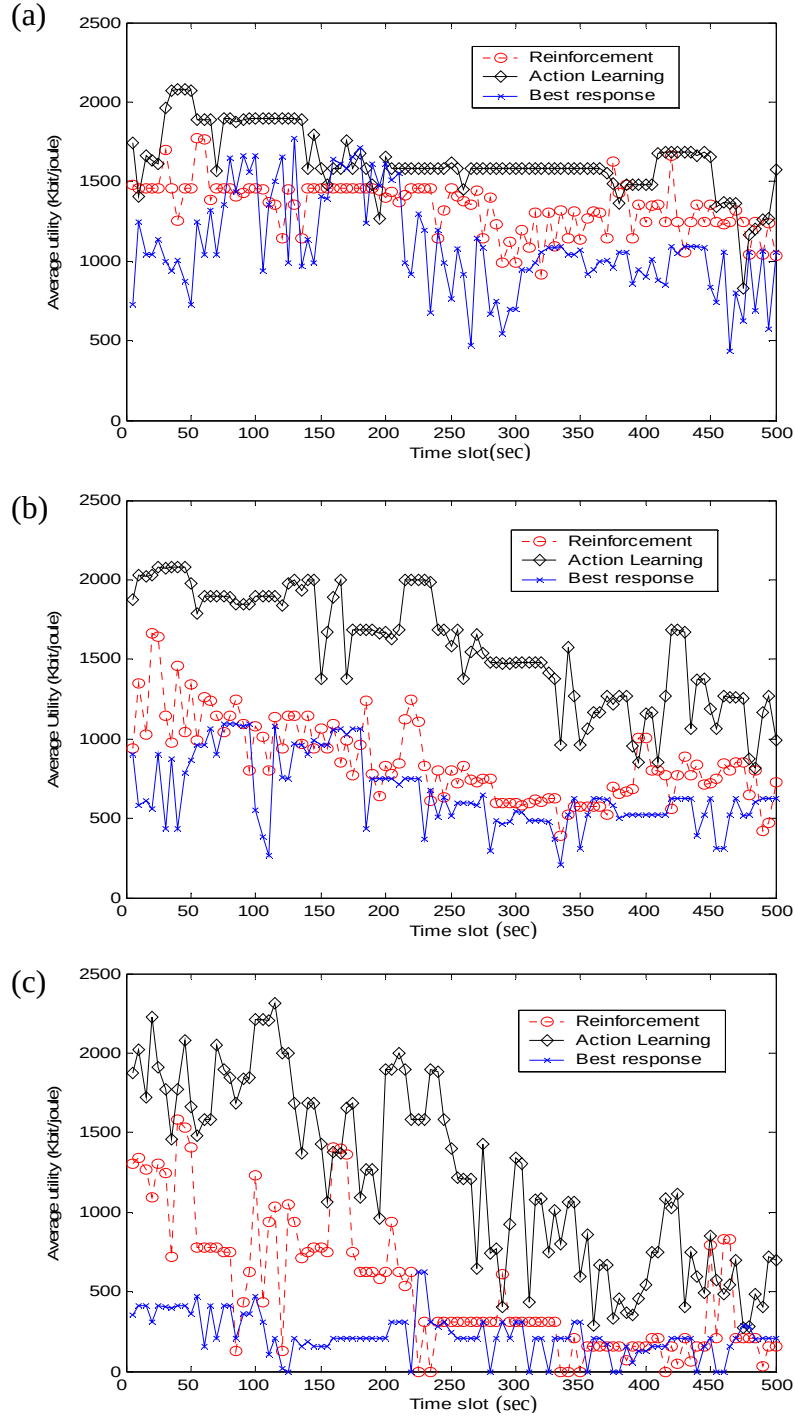


Fig. 5.11 Average utility over time using the adaptive interactive learning when receivers have mobility ($T = 2100$ Kbps) (a) $\nu = 0.5$, (b) $\nu = 1$, (c) $\nu = 2$ (m/time slot).

VIII. APPENDIX A

TABLE 5.5 SUMMARY OF THE USED NOTATIONS OF CHAPTER 5.

Notation	Description
Related to Users' Actions	
m_v	User m_v is composed by a source node n_v^s and a destination node n_v^d , i.e. $m_v = \{n_v^s, n_v^d\}$.
$\mathbf{M}_{-v}$	A set of users except the user m_v .
f_v	Frequency channel selected by user m_v .
P_v	Power level selected by user m_v .
$A_v = [f_v, P_v]$	A variable represents the action selected by user m_v , including the frequency channel selection and power level selection.
A_{-v}	A variable represents the actions taken by all the other users except user m_v .
$\mathbf{A}_{-v}^{fors}$	The exact response actions of the other users given that A_v is taken by m_v .
$S_v^t(A_v)$	The probability for user m_v to select an action A_v at time t .
$\mathbf{S}_v^t$	The strategy of user m_v at time t . $\mathbf{S}_v^t = [S_v^t(A_v), \forall A_v]$
$\mathbf{S}_{-v}^t$	The strategies of all the other users except user m_v .
$\tilde{\mathbf{S}}_{-v}^t$	The belief of user m_v on the strategies of all the other users $\tilde{\mathbf{S}}_{-v}^t$.
Related to Users' Utilities	
$u_v(A_v, A_{-v})$	The utility of user m_v .
$U_v(\mathbf{A}_{-v}^{fors})$	The utility upper bound when the exact response actions are available.
$\gamma_v(A_v, A_{-v})$	The SINR sensed by the receiver of user m_v .
λ_v	The packet arrival rate for the traffic of user m_v .
d_v	The delay deadline for the traffic of user m_v .
$D_v(A_v, A_{-v})$	The delay experienced by user m_v .
$B_v(A_v, A_{-v})$	The effective throughput experienced by user m_v .
$T_v(f_v)$	The transmission rate experienced by user m_v , when it select a frequency channel f_v .
$p_v(\gamma_v)$	The packet error rate experienced by user m_v , given the sensed SINR γ_v .
Related to Information Exchange and Learning	
$\mathcal{Z}_v^t$	The information gathered by user m_v at time t .
o_v^t	The observed information history of user m_v at time t .
Λ_v	The learning scheme adopted by user m_v that results in a certain belief, i.e. $\tilde{\mathbf{S}}_{-v}^t = \Lambda_v(o_v^t)$.
$J_v(\Lambda_v(o_v^t))$	The learning efficiency of adopting a certain learning scheme Λ_v and having observed information history o_v^t .
$\Delta_P(\Lambda_v(o_v^t))$	The price of imperfect belief for using the learning scheme Λ_v based on o_v^t .
σ_v	The information overhead experienced by user m_v .
$\mathbf{V}_v$	A set of the neighbors of user m_v .
ω_v	The information feedback frequency of user m_v .
$B'_v(\sigma_v, A_v, A_{-v})$	The reduced effective throughput experienced by user m_v given the information overhead σ_v .

IX. APPENDIX B

Recall that T_v and p_v represent the maximum transmission rate and packet error rate of user m_v using the frequency channel f_v . T_v and p_v are estimated by the MAC/PHY layer link adaptation [Kri02], which can be modeled as sigmoid functions of the SINR $\gamma_v(A_v, A_v)$ for user m_v :

$$p_v(f_v, \gamma_v(A_v, A_v)) = \frac{1}{1 + \exp(\zeta(\gamma_v(A_v, A_v) - \delta))}, \quad (33)$$

$$B_v(A_v, A_v) = T_v(f_v) \times (1 - p_v(f_v, \gamma_v(A_v, A_v))), \quad (34)$$

$B'_v(\sigma_v, A_v, A_v) = B_v(A_v, A_v) \times \theta(\sigma_v)$, and $\sigma_v(\omega_v, |\mathbf{V}_v|) = 1 - \rho\omega_v(|\mathbf{V}_v| + 1)$, where ζ , δ , and $\rho > 0$ are empirical constants corresponding to the modulation and coding schemes for a given packet length.

Assume that a delay sensitive application is sent by the user m_v through the network with the average input rate R_v (bits/sec). Assume that the user m_v maintains a queue with infinite buffer size in the application layer. We model the packet arrival process using Poisson process. The packet arrival rate is assumed as $\lambda_v = R_v / L_v$ (packet/sec). Considering the packet protection scheme similar to the Automatic Repeat Request protocol in IEEE 802.11 networks [IEE03], the transmission time of a packet can be modeled as a geometric distribution. For simplicity, we approximate the queuing model as M/M/1 queue with the service rate $\mu_v(\sigma_v, A_v, A_v) = B'_v(\sigma_v, A_v, A_v) / L_v$ (packet/sec). Denote the delay of transmitting the delay sensitive application through the network as $D_v(\sigma_v, A_v, A_v)$. The average delay can be obtained by

$$E[D_v(\sigma_v, A_v, A_v)] = \frac{1}{\mu_v(\sigma_v, A_v, A_v) - \lambda_v}, \text{ for } \mu_v(\sigma_v, A_v, A_v) > \lambda_v. \quad (35)$$

Using the M/M/1 queuing model, the probability that the packet of the user m_v can be received before the delay deadline d_v is

$$\text{Prob}\{D_v(\sigma_v, A_v, A_v) \leq d_v\} = \begin{cases} 1 - \exp(-\frac{d_v}{E[D_v(\sigma_v, A_v, A_v)]}), & \text{for } \mu_v(\sigma_v, A_v, A_v) > \lambda_v \\ 0, & \text{otherwise} \end{cases}, \quad (36)$$

The utility function in (2) equals to 0 unless the transmitted power is high enough to

support a sufficient throughput $B'_v(\sigma_v, A_v, A_v)) / L_v > \lambda_v$ to keep the probability $\text{Prob}\{D_v(\sigma_v, A_v, A_v) \leq d_v\} > 0$ (see Figure 2). Substituting equation (35) and (36) into equation (2), we have equation (11). Since $B'_v(\sigma_v)$ is a decreasing function of σ_v , the utility function is a non-increasing function of σ_v .

X. APPENDIX C

Proof of Proposition 4: Given the channel model $B_v(f, \gamma)$ for the frequency channel f in equation (34), user m_v , $m_v \in \Omega_f$ can apply queuing analysis with the application characteristics R_v , L_v and d_v . From equation (35) and (36), we have $\text{Prob}\{D_v \leq d_v\} = 1 - \frac{1}{F_v(\gamma_v)}$. The optimality condition of $\frac{\partial u_v}{\partial P_v} = 0$ becomes $-P_v \times \frac{\partial}{\partial P_v} \frac{1}{F_v(\gamma_v)} = 1 - \frac{1}{F_v(\gamma_v)}$. The left hand side can be derived as $\gamma_v \frac{\partial B_v(\gamma_v)}{\partial \gamma_v} \times \frac{d_v}{L_v} \frac{1}{F_v(\gamma_v)}$, since $P_v \frac{\partial \gamma_v}{\partial P_v} = \gamma_v$. By multiplying F_v to both sides, we have the optimality condition in Proposition 4 and the corresponding γ_v^{tar} that maximizes the utility function u_v .

XI. APPENDIX D

Algorithm 5.1 Adaptive reinforcement learning with private information feedback

For user m_v at time slot t , assume $U(0,1)$ represents a uniform distribution from 0 to 1.

Initialization: Set $J_v^{prev} = 0$, $\omega_v = 1$, $\Delta\omega_v = 0.05$.

Step 1. If $Rand(U(0,1)) < 1 - \omega_v$, keep using action $A_v^t = A_v^{base}$, $t \leftarrow t + 1$, and repeat Step 1, otherwise go to Step 2.

Step 2. Calculate $u_v(A_v^{t-1}, \gamma_v^{t-1})$ from previous action $A_v^{t-1} = [f_v^{t-1}, P_v^{t-1}]$ and the private information feedback $\mathcal{Z}_v^{t,priv} = \{\gamma_v^{t-1}\}$.

Step 3. Update the propensity r_v^t and the strategy S_v^t .

Step 4. Determine the action from $A_v^t = Rand(S_v^t)$.

Step 5. Update the baseline action A_v^{base} and baseline payoff value u_v^{base} as in equation (21) and (22).

Step 6. Evaluate J_v . If $J_v > J_v^{prev}$, then
 If $\omega_v - \Delta\omega_v > 0$, $\omega_v \leftarrow \omega_v - \Delta\omega_v$, else if $\omega_v - \Delta\omega_v \leq 0$, keep ω_v .
 Otherwise,
 If $\omega_v + \Delta\omega_v \leq 1$, $\omega_v \leftarrow \omega_v + \Delta\omega_v$, else if $\omega_v + \Delta\omega_v > 1$, keep ω_v .

Step 7. Set $J_v^{prev} \leftarrow J_v$, $t \leftarrow t + 1$, and go back to Step 1.

Algorithm 5.2 Adaptive action learning ($H=2$) with public information feedback

For user m_v at time slot t ,

Initialization: Set $J_v^{prev} = 0$, $|\mathbf{V}_v| = |\mathbf{M}_{-v}|$, $\Delta|\mathbf{V}_v| = 1$.

Step 1. Observe the public information feedback $\mathcal{Z}_{-v(V_v)}^{t,pub} = \{\mathbf{G}_u^{t-1}, A_u^{t-1}, m_u \in \mathbf{V}_v\}$ fed back from the users $m_u \in \mathbf{V}_v$.

Step 2. Update the propensity r_u^t for users $m_u \in \mathbf{V}_v$ and calculate the strategy vector $\tilde{\mathbf{S}}_{-v}^t(A_v)$.

Step 3. Calculate the target power $P_v^{tar}(f)$ from equation (31) and find the action $A_v^t = [f_v^*, P_v^*]$ using Proposition 5.

Step 4. Evaluate J_v . If $J_v > J_v^{prev}$, then
 If $|\mathbf{V}_v| - \Delta|\mathbf{V}_v| > 0$, $|\mathbf{V}_v| \leftarrow |\mathbf{V}_v| - \Delta|\mathbf{V}_v|$, else if $|\mathbf{V}_v| - \Delta|\mathbf{V}_v| \leq 0$, keep $|\mathbf{V}_v|$.
 Otherwise,
 If $|\mathbf{V}_v| + \Delta|\mathbf{V}_v| \leq |\mathbf{M}_{-v}|$, $|\mathbf{V}_v| \leftarrow |\mathbf{V}_v| + \Delta|\mathbf{V}_v|$, else if $|\mathbf{V}_v| + \Delta|\mathbf{V}_v| > |\mathbf{M}_{-v}|$, keep $|\mathbf{V}_v|$.

Step 5. Set $J_v^{prev} \leftarrow J_v$, $t \leftarrow t + 1$, and go back to Step 1.

Chapter 6

Resource Management in Single-Hop Cognitive Radio Networks

I. INTRODUCTION

The demand for wireless spectrum has increased rapidly in recent years due to the emergence of a variety of applications, such as wireless Internet browsing, file downloading, streaming, etc. In the foreseeable future, the requirements for wireless spectrum will increase even more with the introduction of multimedia applications such as YouTube, peer to peer multimedia networks, and distributed gaming. However, scanning through the radio spectrum reveals its inefficient occupancy [SW04] in most frequency channels. Hence, the Federal Communications Commission (FCC) suggested in 2002 [FCC02] improvements on spectrum usage to efficiently allocate frequency channels to license-exempt users without impacting the primary licensees. This forms cognitive radio networks that 1) enhance the spectrum usage of the traditional licensing system, and 2) release more spectrum resources for the unlicensed allocations in order to fulfill the required demand.

The emergence of cognitive radio networks have spurred both innovative research and ongoing standards [CCB06][Hay05][MM99]. Cognitive radio networks have the capability of achieving large spectrum efficiencies by enabling *interactive* wireless users to *sense* and *learn* the surrounding environment and correspondingly *adapt* their transmission strategies. Three main challenges arise in this context. The first problem is how to sense the spectrum and model the behavior of the primary licensees. The second problem is how to manage the available spectrum resources and share the resource to the license-exempt users to satisfy their transmission requirements while not interfering with

the primary licensees. The third problem is how to maintain seamless communication during the transition (hand-off) of selected frequency channels. In this chapter, we focus on the second challenge and rely on the existing literature for the remaining two challenges [ALV06][Bro05].

Prior research such as [CCB06][ZL06] focus on centralized solutions for the resource management problem in cognitive radio networks. However, due to the informationally-decentralized nature of wireless networks, the complexity of the optimal centralized solutions for spectrum allocation is prohibitive [WP02] for delay-sensitive multimedia applications. Moreover, the centralized solution requires the propagation of private information back and forth to a common coordinator, thereby incurring delay that may be unacceptable for delay-sensitive applications. Hence, it is important to implement decentralized solutions for dynamic channel selection by relying on the wireless multimedia users' capabilities to sense and adapt their frequency channel selections. Moreover, unlike most of the existing research on resource management in the cognitive radio networks [TJ91][SCC05] that ignores the multimedia traffic characteristics in the application layer and assumes that all competing users in the networks are of the same type (applications, radio capabilities), we consider *heterogeneous* users in this chapter, meaning that the users can have 1) different types of utility functions and delay deadlines, 2) different traffic priorities and rates, and 3) experience distinct channel conditions in different frequency channels. For example, the multimedia users can differ in their preferences of utility functions, priorities of accessing the frequency channels, traffic rate requirements, capabilities of transmitting data in different frequency channels. Note that in the informationally-decentralized wireless network, these utility functions, traffic characteristics, and the channel conditions are usually considered as private information of the users. Hence, the main challenge here is how to coordinate the spectrum sharing among heterogeneous multimedia users in a decentralized manner.

To do this, information exchange across the multimedia users is essential. Since the

decisions of a user will impact and be impacted by the other users selecting the same frequency channel, without explicit information exchange, the heterogeneous users will consume additional resources and respond slower to the time-varying environment [Luc06]. The key questions are what information exchanges are required, and how autonomous users adapt their channel selections based on the limited information exchange to efficiently maximize their private utilities. In this chapter, we propose a novel priority virtual queue interface to abstract multimedia users' interactions and determine the required information exchange according to the priority queuing analysis. Note that such information exchanges can rely on a dedicated control channel for all users, or can use a group-based scheme without a common control channel [ZZY05].

In this chapter, we model the traffic of the users (including the licensed users and the license-exempt users) and the channel conditions (e.g. Signal-to-Noise Ratio, Bit-Error-Rate) by stationary stochastic models similar to [SCC05]. Our approach endows the primary licensees with the priority to preempt the transmissions of the license-exempt users in the same frequency channel. Based on the priority queuing analysis, each wireless user can evaluate its utility impact based on the behaviors of the users deploying the same frequency channel (including the primary licensees, to which the highest priority is assigned). The behavior of a user is represented by its probability profile for selecting different frequency channels, which is referred as the channel selection strategy in this chapter. Based on the expected utility evaluation, we propose a Dynamic Strategy Learning (DSL) algorithm for an autonomous multimedia user to adapt its channel selection strategy.

In summary, our chapter addresses the following important issues:

a) Separation of the utility evaluation and channel selection using the priority virtual queue interface.

We propose a novel priority virtual queue interface for each autonomous user to exchange information and maximize its private utility in cognitive radio networks.

Through the interface, the user can model the strategies of the other users with higher priorities and evaluates the expected utility of selecting a certain frequency channel. Importantly, the interface provides a simple model that facilitates the user's learning of what is the best channel selection strategy.

b) Priority virtual queuing analysis for heterogeneous multimedia users.

Unlike prior works on cognitive radio networking, which seldom consider multimedia traffic characteristics and delay deadlines in the application layer, our priority virtual queue framework enables the autonomous multimedia users to consider 1) priorities of accessing the frequency channels, 2) different traffic loads and channel conditions in different frequency channels, and 3) heterogeneous preferences for various types of utility functions based on the deployed applications. Note that the priority queuing model allows the primary licensees to actively share the occupied channels instead of excluding all the other wireless users. However, by assigning highest preemptive priorities to the licensees, the unlicensed users do not impact the licensees.

c) DSL algorithm for dynamic channel selections by wireless stations.

Based on the expected utility evaluation from the interface, we propose a decentralized learning algorithm that dynamically adapts the channel selection strategies to maximize the private utility functions of users. Note that a frequency channel can be shared by several users. A wireless user can also select multiple frequency channels for transmission. Our learning algorithm addresses how multimedia users distribute traffic to multiple available frequency channels to maximize their own utility functions.

The rest of this chapter is organized as follows. Section II provides the specification of cognitive radio networks and models the dynamic resource management problem as a multi-agent interaction problem. In Section III, we give an overview of our dynamic resource management for the heterogeneous multimedia users, including the priority virtual queue interface and the dynamic channel selection. In Section IV, we provide the queuing analysis for the priority virtual queue interface and determine the required

information exchange. In Section V, we focus on the dynamic channel selection and propose the DSL algorithm to adapt the channel selection strategy for the multimedia users. Simulation results are given in Section VI. Section VII concludes the chapter.

II. MODELING THE COGNITIVE RADIO NETWORKS AS MULTI-AGENT INTERACTIONS

A. Agents in a cognitive radio network

In this chapter, we assume that the following agents interact in the cognitive radio network:

- **Primary Users** are the incumbent devices possessing transmission licenses for specific frequency bands (channels). We assume that there are M channels in the cognitive radio network, and that there are several primary users in each frequency channel. These primary users can only occupy their assigned frequency channels. Since the primary users are licensed users, they will be provided with an interference-free environment [Hay05][ALV06].
- **Secondary Users** are the autonomous wireless stations that perform channel sensing and share the available spectrum holes [CCB06]. We assume that there are N secondary users in the system. These secondary users are able to transmit their traffic using various frequency channels. If multiple users select the same frequency channel, they will time share the chosen frequency channel. Moreover, these secondary users are license-exempt, and hence, they cannot interfere with the primary users.

In this chapter, we consider the users sharing a single-hop wireless ad-hoc network. Figure 6.1 provides an illustration of the considered network model. We assume the secondary users as transmitter-receiver pairs with information exchange among these pairs. In order to maintain stationary property, we assume that these network agents are static (i.e. we do not consider mobility effects). Next, we model the interaction among secondary users accessing the same frequency channel.

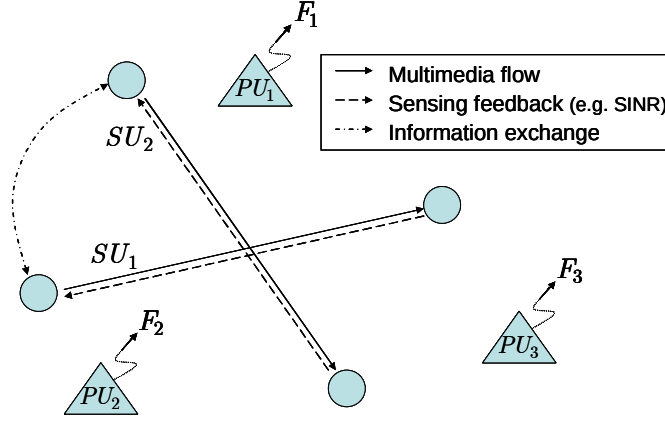


Fig. 6.1 An illustration of the considered network model.

B. Models of the dynamic resource management problem

- **Users:** As indicated above, there are two sets of users – aggregate primary users in each channel $\mathbf{PU} = \{PU_1, \dots, PU_M\}$ ¹ and the secondary users $\mathbf{SU} = \{SU_1, \dots, SU_N\}$. The priorities of users in cognitive radio networks are pre-assigned depending on their Quality of Service (QoS) requirements and their right to access the frequency channels.
- **Resources:** The resources are the frequency channels $\mathbf{F} = \{F_1, \dots, F_M\}$. Multiple users can time share the same frequency channel. Note that even if the same time sharing fraction is assigned to the users choosing the same frequency channel, their experienced channel conditions may differ.
- **Actions:** The considered actions of the secondary users are the selection of the frequency channel for each packet transmission. We denote the actions of a secondary user SU_i using $\mathbf{a}_i = [a_{i1}, a_{i2}, \dots, a_{iM}] \in \mathcal{A}^M$, where $a_{ij} \in \mathcal{A}$ ($\mathcal{A} = \{0, 1\}$). $a_{ij} = 1$ indicates that SU_i chooses the frequency channel F_j . Otherwise, $a_{ij} = 0$. Let $\mathbf{a}_{-i}$ denote the actions of the other secondary users except SU_i . Let

¹ From the secondary users' point of view, there is no need to differentiate different primary users in one frequency channel. Hence, we reduce the primary users in one frequency channel into one aggregate primary user. A secondary user needs to back-off and wait for transmission or select another frequency channel, once any of the primary users starts to transmit in the same frequency channel.

$\mathbf{A} = [\mathbf{a}_1^T, \dots, \mathbf{a}_N^T] \in \mathcal{A}^{M \times N}$ denote the total action profile across all secondary users.

- **Strategies:** A strategy of a secondary user SU_i is a vector of probabilities $\mathbf{s}_i = [s_{i1}, s_{i2}, \dots, s_{iM}] \in \mathcal{S}^M$, where $s_{ij} \in \mathcal{S}$ ($\mathcal{S} \in [0,1]$) represents the probability of the secondary user SU_i to take the action a_{ij} (i.e. to choose the frequency channel F_j). Hence, the summation over all the frequency channels is $\sum_{j=1}^M s_{ij} = 1$. Note that s_{ij} can also be viewed as the fraction of data from SU_i transmitted on frequency channel F_j , and hence, multiple frequency channels are selected for a secondary users with $s_{ij} > 0$. Let $\mathbf{S} = [\mathbf{s}_1^T, \dots, \mathbf{s}_N^T] \in \mathcal{S}^{M \times N}$ denote the total strategy profile across all secondary users.
- **Utility functions:** Each secondary user has its own utility function. Based on the adopted actions of the secondary users, we denote the utility function of SU_i as u_i . Conventionally, the utility function of a specific user is often modeled solely based on its own action, i.e. $u_i(\mathbf{a}_i)$ without modeling the other secondary users [WP02][VS05]. However, the utility function for multimedia users relates to the effective delay and throughput that a secondary user can derive from the selected frequency channel, which is coupled with the actions of other secondary users. Hence, the utility function u_i is also influenced by the action of other secondary users that select the same frequency channel. In other words, the utility function can be regarded as $u_i(\mathbf{a}_i, \mathbf{a}_{-i})$. We will discuss this utility function in detail in Section III.C.
- **Expected utility function with dynamic adaptation:** In an informationally-decentralized cognitive wireless network that consists of heterogeneous secondary users, the secondary user SU_i may not know the exact actions of other secondary users $\mathbf{a}_{-i}$. Moreover, even if all the actions are known, it is unrealistic to assume that the exact action information can be collected timely to compute and maximize the actual utility function $u_i(\mathbf{a}_i, \mathbf{a}_{-i})$. Hence, a more practical solution is to dynamically model the other secondary users' behavior by updating their probabilistic strategy profile of actions $\mathbf{s}_{-i}$ based on the observed information,

and then compute the optimal channel selection strategy s_i that maximizes the expected utility function of SU_i , i.e.

$$U_i(s_i, s_{-i}) = E_{(s_i, s_{-i})}[u_i(a_i, a_{-i})], \quad (1)$$

where $E_{(s_i, s_{-i})}[u_i(a_i, a_{-i})]$ is the expected utility function, given a fixed strategy profile $S = (s_i, s_{-i})$. In the next section, we discuss how secondary users perform dynamic resource management that maximizes the expected utility function $U_i(s_i, s_{-i})$ by modeling the strategy (behavior) s_{-i} of the other users in cognitive radio networks.

III. DYNAMIC RESOURCE MANAGEMENT FOR HETEROGENEOUS SECONDARY USERS USING PRIORITY QUEUING

In this section, we provide our dynamic resource management solution using the multi-agent interaction settings in the previous section. We first emphasize the heterogeneity of the secondary users in cognitive radio networks and then introduce our solution with the priority queuing interface and adaptive channel selection strategies.

A. Prioritization of the users

We assume that there are K priority classes of users in the system. The highest priority class C_1 is always reserved for the primary users PU in each frequency channel. The heterogeneous secondary users SU can be categorized into the rest of $K - 1$ priority classes $(C_2, \dots, C_K)$ to access the frequency channels². We assume that the users in higher priority classes can preempt the transmission of the lower priority classes to ensure an interference-free environment for the primary users [Kle75]. The priority of a user affects its ability of accessing the channel. Primary users in the highest priority class C_1 can always access their corresponding channels at any time. Secondary users, on the other hand, need to sense the channel and wait for transmission opportunities for transmission (when there is no higher priority users using the channel) based on their

² The prioritization of the secondary users can be determined based on their applications, prices paid for spectrum access, or other mechanism design based rules. In this chapter, we will assume that the prioritization was already performed.

priorities. We assume that there are N_k users in each of the class C_k . Hence, $N_1 = M$ (number of aggregate primary users) and $\sum_{k=2}^K N_k = N$ (number of secondary users).

Various multiple access control schemes can be adopted for the secondary users to share the spectrum resource. For simplicity, in this chapter, we consider a MAC protocol similar to IEEE 802.11e HCF [IEE03]³ to assign transmission opportunities (i.e. TXOP) and ensure that a secondary users in the lower priority class will stop accessing the channel and wait in the queue or change its action (channel selection) if a higher priority user is using the frequency channel. Note that for secondary users, they not only can have different priorities to access the frequency channels, but they can also have different channel conditions and possess their own preferences for a certain type of utility function, which is discussed in the following subsections.

B. Heterogeneous channel conditions

For a certain frequency channel F_j , the secondary users can experience various channel conditions for the same frequency channel. We denote T_{ij} and p_{ij} as the resulting physical transmission rate and packet error rate for the secondary user SU_i transmitting through a certain frequency channel F_j . Let $R_{ij} = [T_{ij}, p_{ij}] \in \mathcal{R}$ be the channel conditions of the channel F_j for the secondary user SU_i . We denote the channel condition matrix as $\mathbf{R} = [R_{ij}] \in \mathcal{R}^{M \times N}$. The expected physical transmission rate and packet error rate can be approximated as sigmoid functions of measured Signal-to-Interference-Noise-Ratio (SINR) and the adopted modulation and coding scheme as in [Kri02]. Note that the expected T_{ij} and p_{ij} of the same frequency channel can be different for various secondary users.

C. Goals of the heterogeneous secondary users

In general, the utility function u_i is a non-decreasing function of the available

³ Either the polling-based HCCA or contention-based EDCA protocols can be applied, as long as the priority property of the users is provided. However, a more sophisticated MAC protocols can also be considered to deal with the spectrum heterogeneity (such as HD-MAC in [ZZY05]). Different MAC protocols will have different overheads including the time of waiting for the MAC acknowledgement, contention period, etc. that affect the service time distribution of the M/G/1 queuing model.

transmission rates. Several types of objectives for the secondary users can be considered in practice, such as minimizing the end-to-end delay, loss probability, or maximizing the received quality, etc. For simplicity, we assume only two types of utility functions⁴ in this chapter.

- **The delay-based utility** for *delay-sensitive* multimedia applications.

Let $D_i(\mathbf{a}_i, \mathbf{a}_{-i})$ represent the end-to-end packet delay (transmission delay plus the queuing delay) for the secondary user SU_i . Let d_i represent the delay deadline of the application of secondary user SU_i . We consider this type of utility function as (as in [JCO02]):

$$u_i^{(1)}(\mathbf{a}_i, \mathbf{a}_{-i}) = \text{Prob}(D_i(\mathbf{a}_i, \mathbf{a}_{-i}) \leq d_i), \quad (2)$$

which depends on the end-to-end delay $D_i(\mathbf{a}_i, \mathbf{a}_{-i})$ and the delay deadline d_i imposed by the application.

- **The throughput-based utility** for *delay-insensitive* applications.

Let T_i^{eff} represent the effective available throughput for the secondary user SU_i . The second type of utility function is assumed to be directly related to the throughput (as in [ZP05]). In this chapter, we define it as:

$$u_i^{(2)}(\mathbf{a}_i, \mathbf{a}_{-i}) = \begin{cases} \frac{T_i^{eff}(\mathbf{a}_i, \mathbf{a}_{-i})}{T_i^{\max}}, & \text{if } T_i^{eff}(\mathbf{a}_i, \mathbf{a}_{-i}) \leq T_i^{\max} \\ 1 & , \text{ if } T_i^{eff}(\mathbf{a}_i, \mathbf{a}_{-i}) > T_i^{\max} \end{cases}, \quad (3)$$

where $T_i^{\max}$ is the physical throughput required by the secondary user SU_i .

We assume that a secondary user can possess multiple applications that can be either delay-sensitive multimedia traffic or delay-insensitive data traffic. Hence, we define the utility function of a secondary user as a multi-criterion objective function (as in [ZL06][TJZ03]) of these two types of utility functions. Different secondary users can have different preferences θ_i ⁵ ($0 \leq \theta_i \leq 1$). Specifically, the goal of a secondary user

⁴ This model can be easily extended to more types of utility functions. Moreover, our utility function can also be easily modified to a quality-type utility function using different priorities. For simplicity, we do not consider the quality impact of different multimedia packets in our utility function.

⁵ In this chapter, we assume that the preferences θ_i are predetermined by the secondary users. The preferences θ_i of the multi-criterion optimization can be determined based on the applications. See e.g. [SP85].

SU_i is to maximize the following utility function:

$$u_i(\mathbf{a}_i, \mathbf{a}_{-i}) = \theta_i \cdot u_i^{(1)}(\mathbf{a}_i, \mathbf{a}_{-i}) + (1 - \theta_i) \cdot u_i^{(2)}(\mathbf{a}_i, \mathbf{a}_{-i}). \quad (4)$$

Note that, in this setting, $0 \leq u_i(\mathbf{a}_i, \mathbf{a}_{-i}) \leq 1$.

D. Example of three priority classes with different utility functions

Let $\mathbf{A}_k$ be the action set of the secondary users in the classes $C_2, \dots, C_k$, i.e. $\mathbf{A}_k = \{\mathbf{a}_i \mid SU_i \in C_l, l = 2, \dots, k\}$. Note that $\mathbf{A}_{k-1} \subseteq \mathbf{A}_k \subseteq \mathbf{A}$. Due to the priority queuing structure, the actions of the secondary users with lower priority will not affect the users in the higher priority class [BG87]. Hence, the decentralized optimizations are performed starting from the higher priority classes to the lower priority classes. In other words, the decentralized optimization of a secondary user in a lower priority class also needs to consider the actions of the users in higher priority classes. For example, three classes can be assumed ($K = 3$) – the first priority class is composed by the primary users whose actions are fixed (no channel selection capability). The second priority class C_2 is composed by the secondary users transmitting delay-sensitive multimedia applications, and the third priority class C_3 is composed by the secondary users transmitting regular data traffic, which requires throughput maximization. The objective function for each of the secondary users in priority class C_2 is ($\theta_i = 1$, for $SU_i \in C_2$):

$$\begin{aligned} & \text{maximize } U_i^{(1)}(\mathbf{s}_i, \mathbf{s}_{-i}) \\ \Rightarrow & \text{maximize } E_{(\mathbf{s}_i, \mathbf{s}_{-i})}[\text{Prob}(D_i(\mathbf{A}_2) \leq d_i)] \end{aligned} \quad (5)$$

Then, the objective function for the secondary users in the class C_3 is ($\theta_i = 0$, for $SU_i \in C_3$):

$$\begin{aligned} & \text{maximize } U_i^{(2)}(\mathbf{s}_i, \mathbf{s}_{-i}) \\ \Rightarrow & \text{maximize } E_{(\mathbf{s}_i, \mathbf{s}_{-i})}[T_i^{\text{eff}}(\mathbf{A})] \end{aligned} \quad (6)$$

with the constraint that $\mathbf{A}_2 \subseteq \mathbf{A}$ are predetermined by (5). The effective transmission rate of each secondary user can be expressed as:

$$E_{(\mathbf{s}_i, \mathbf{s}_{-i})}[T_i^{\text{eff}}(\mathbf{A})] = \sum_{j=1}^M s_{ij} T_{ij} (1 - p_{ij}). \quad (7)$$

From the above three classes example, note that delay analysis is essential for the

heterogeneous secondary users with delay-sensitive applications in a cognitive radio network.

To maximize the expected utility function as stated in equation (1), a secondary user needs to consider the impact of the other secondary users. In order to efficiently regulate the information exchange among heterogeneous users and efficiently provide expected utility evaluation, a coordination interface must be developed. Based on this interface, the secondary users can interact with each other in a decentralized manner. In the next subsection, we propose a novel dynamic resource management with such an interface for a secondary user SU_i to adapt its frequency selection strategy s_i .

E. Priority virtual queue interface

The resource management for delay-sensitive multimedia applications over cognitive radio networks needs to consider the heterogeneous wireless users having various utility functions, priorities of accessing the channel, traffic rates, and channel conditions. Specifically, the main challenge is how to coordinate the spectrum sharing among competing users and select the frequency channel to maximize the utility functions in a decentralized manner. For this, we propose a novel priority virtual queue interface. Unlike prior research assuming that secondary users apply 2-state “spectrum holes” (on-off model [SCC05]) for spectrum access [Hay05] in our priority virtual queue interface, we allow secondary users to obtain transmission opportunities once the primary user in a specific channel stops transmitting. The primary users have the highest priority, thereby being able to preempt the transmission of the secondary users’ transmission.

The priority virtual queue interface has two main tasks – 1) determines the required information exchange and 2) evaluates the utility impact from the wireless environment as well as the competing users’ behaviors in the same frequency channel. In the priority virtual queue interface of a user, the virtual queues are preemptive priority queues [Kle75] for each of the frequency channels. They are emulated by each multimedia user to estimate the delay of selecting a specific frequency channel for transmission. Figure 6.2

illustrates the architecture of the proposed dynamic resource management with priority virtual queue interface that exchanges information and emulates the expected delay. Note that these virtual queues are in fact distributed (physically located) at the secondary users.

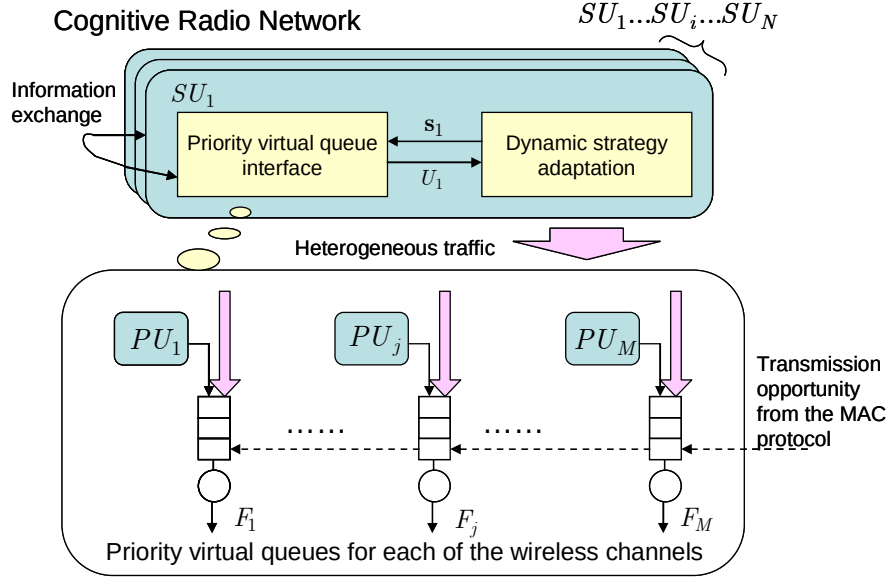


Fig. 6.2 The architecture of the proposed dynamic resource management with priority virtual queue interface.

The implementation of the dynamic resource management with priority virtual queue interface of the secondary users is presented below:

1. **Information exchange collection:** The secondary user SU_i collect the required information from other secondary users through the priority virtual queue interface. The required information exchange will be discussed in Section IV.D based on the queuing analysis.
2. **Priority queuing analysis:** The interface estimates s_{-i} and performs priority queuing analysis based on the observed information to evaluate the expected utility $U_i(s_i, s_{-i})$. The priority queuing analysis will be discussed in details in Section IV.
3. **Dynamic strategy adaptation:** Based on the expected utility $U_i(s_i, s_{-i})$, the

secondary user adapts its channel selection strategy s_i . We propose a dynamic strategy learning algorithm, which will be discussed in detail in Section V.

4. **Assign actions for each packet based on the strategy:** Based on current channel selection strategy s_i , SU_i can assign to each packet an action (select frequency channel according to the probability profile). As the channel selection strategy adapts to the network changes, the behavior of a secondary user selecting the frequency channels for its packets will also change.
5. **Wait for the transmission opportunity and transmit the packets:** The packets wait in queues to be transmitted. Based on the priorities of the users, the higher priority secondary users will have a better chance to access the channel and transmit their packets.

Note that the primary users will transmit whenever needed in their corresponding frequency channels.

Next, we present the priority queuing analysis for delay-sensitive multimedia users to evaluate $U_i(s_i, s_{-i})$.

IV. PRIORITY QUEUING ANALYSIS FOR DELAY-SENSITIVE MULTIMEDIA USERS

In this section, we discuss the priority queuing analysis for delay-sensitive multimedia applications. It is important to note that the packets of the competing wireless users are physically waiting at different locations. Figure 6.3 gives an example of the physical queues for the case of M frequency channels and N secondary users. Each secondary user maintains M physical queues for the various frequency channels, which allows users to avoid the well-known head-of-line blocking effect [WZF04]. The channel selection decisions are based on the queuing analysis, which will be discussed in detail in Section V. In this section, we focus on the priority queuing analysis from the perspective of each secondary user to evaluate $U_i(s_i, s_{-i})$.

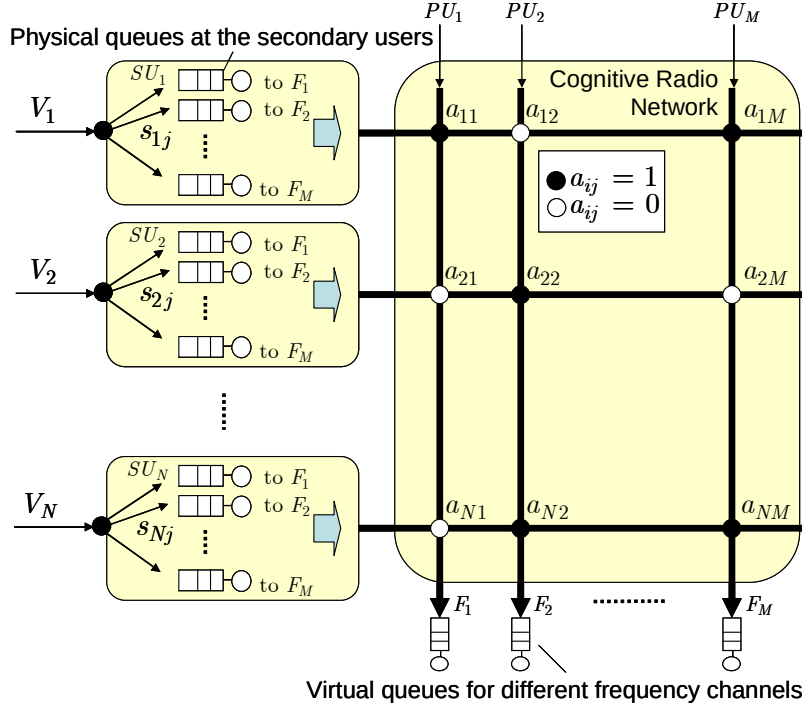


Fig. 6.3 Actions of the secondary users a_{ij} and their physical queues for each frequency channel.

A. Traffic models

- **Traffic model for primary users**

We assume that the stationary statistics of the traffic patterns of primary users can be modeled by all secondary users. The packet arrival process of a primary user is modeled as a Poisson process with average packet arrival rate λ_j^{PU} for the primary user PU_j using the frequency channel F_j . Note that the aggregation of Poisson processes of primary users in the same frequency channel is still Poisson. We denote the m^{th} moments of the service time distribution of the primary user PU_j in frequency channel F_j as $E[(X_j^{PU})^m]$. We adopt an M/G/1 model for the traffic descriptions. Note that this traffic model description is more general than a Markov on-off model [SCC05], which is a sub-set of our queuing model with an exponential idle period and an exponential busy period.

- **Traffic model for secondary users**

We assume that the average rate requirement for the secondary user SU_i is B_i (bit/s). Let λ_{ij} denote the average packet arrival rate of the secondary user SU_i using the frequency channel F_j . Since the strategy s_{ij} represents the probability of the secondary user SU_i taking action a_{ij} (transmitting using the frequency channel F_j), we have

$$\lambda_{ij} = s_{ij} \frac{B_i}{L_i}, \quad (8)$$

where L_i denotes the average packet length of the secondary user SU_i . If a certain secondary user SU_i can never use the frequency channel F_j , we fix its strategy to $s_{ij} = 0$, and hence, $\lambda_{ij} = 0$. For simplicity, we also model the packet arrival process of the secondary users using a Poisson process. Note that the average arrival rate is the only sufficient statistics required to describe a Poisson process.

Since packet errors are unavoidable in a wireless channel, we assume that packets will be retransmitted, if they are not correctly received. This can be regarded as a protection scheme similar to the Automatic Repeat Request protocol in IEEE 802.11 networks [IEE03]. Hence, the service time of the users can be modeled as a geometric distribution [Kon80]. Let $E[X_{ij}]$ and $E[X_{ij}^2]$ denote the first two moments of the service time of the secondary user SU_i using the frequency channel F_j . We have:

$$E[X_{ij}] = \frac{L_i + L_o}{T_{ij}(1 - p_{ij})}, \quad (9)$$

$$E[X_{ij}^2] = \frac{(L_i + L_o)^2(1 + p_{ij})}{T_{ij}^2(1 - p_{ij})^2}, \quad (10)$$

where L_i is the average packet length of the secondary user SU_i and L_o represents the effective control overhead including the time for protocol acknowledgement⁶, information exchange, and channel sensing delay, etc. (see [IEE03] for details). Let us denote $\mathbf{X}_i = [E[X_{ij}] \mid j = 1, \dots, M]$ and $\mathbf{X}_i^2 = [E[X_{ij}^2] \mid j = 1, \dots, M]$. To describe the traffic model, we define the traffic specification⁷ for the secondary user SU_i as $\mathbf{TS}_i = [C_k, B_i, L_i, \mathbf{X}_i, \mathbf{X}_i^2]$, if $SU_i \in C_k$. This information needs to be exchanged among the

⁶ Here we only consider retransmission due to channel errors. We consider the protocol overhead in the MAC layer including possible contention period, time for acknowledgement, etc. in the effective control overhead.

⁷ The traffic specification is similar to the TSPEC in current IEEE 802.11e [IEE03] for multimedia transmission.

secondary users, which will be discussed in detail in Section IV.D.

B. Priority virtual queuing analysis

In order to evaluate the expected utility $U_i(\mathbf{s}_i, \mathbf{s}_{-i})$ for delay-sensitive multimedia applications, we need to calculate the distribution of the end-to-end delay $D_i(\mathbf{a}_i, \mathbf{a}_{-i})$ for the secondary user SU_i to transmit its packets. The expected end-to-end delay⁸ $E[D_i]$ of the secondary user SU_i can be expressed as:

$$E[D_i(\mathbf{a}_i, \mathbf{a}_{-i})] = \sum_{j=1}^M s_{ij} \cdot E[D_{ij}(R_{ij}(\mathbf{A}))], \quad (11)$$

where $E[D_{ij}(R_{ij}(\mathbf{A}))]$ is the average end-to-end delay if the secondary user SU_i chooses the frequency channel F_j . Note that s_{ij} is the strategy of the action a_{ij} in $\mathbf{A}$.

Using the queuing model in Figure 6.3, each arriving packet of SU_i will select a physical queue to join (action a_{ij}) according to the strategy s_{ij} . Note that there are N physical queues from N secondary users for a frequency channel F_j . Only one of them can transmit its packets at any time. Hence, we form a “virtual queue” for the same frequency channel as illustrated in Figure 6.3. In a virtual queue, the packets of the different secondary users wait to be transmitted. Importantly, the total sojourn time (queue waiting time plus the transmission service time) of this virtual queue now becomes the actual service time at each of the physical queues. The concept is similar to the “service on vacation” [BG87] in queuing theory, and the waiting time of the virtual queue can be regarded as the “vacation time”.

Since the number of the secondary users in a regular cognitive radio network is usually large, we can approximate the virtual queue using prioritized M/G/1 queuing model (i.e. when $N \rightarrow \infty$, the input traffic of the virtual queue can be modeled as a Poisson process). The average arrival rate of the virtual queue of the frequency channel F_j is $\sum_{i=1}^N \lambda_{ij}$. Let us denote the first two moments of the service time for the virtual queue of the

⁸ In order to simplify the notation, we use simple expectation notation $E[\cdot]$ instead of the expectation over the action strategies $E_{(\mathbf{s}_i, \mathbf{s}_{-i})}[\cdot]$ hereafter in this chapter.

frequency channel F_j as $E[\tilde{X}_j]$ and $E[\tilde{X}_j^2]$. For a packet in the virtual queue of frequency channel F_j , we determine the probability of the packet coming from the secondary user SU_i as:

$$f_{ij} = \frac{\lambda_{ij}}{\sum_{k=1}^N \lambda_{kj}}. \quad (12)$$

Hence,

$$E[\tilde{X}_j] = \sum_{i=1}^N f_{ij} \times E[X_{ij}], \quad E[\tilde{X}_j^2] = \sum_{i=1}^N f_{ij} \times E[X_{ij}^2]. \quad (13)$$

Since there are K priority classes among users ($K > 2$, $PU \in C_1$, $SU \in \{C_2, \dots, C_K\}$), we assume that μ_{jk} represents the normalized traffic loading of all the class C_k secondary users using the frequency channel F_j . By the definition of the normalized traffic loading [BG87], we have:

$$\mu_{jk} = \sum_{\forall SU_i \in C_k} \lambda_{ij} \times E[\tilde{X}_j], \text{ and } \mu_{jk}^2 = \sum_{\forall SU_i \in C_k} \lambda_{ij} \times E[\tilde{X}_j^2]. \quad (14)$$

Assume that $E[\tilde{D}_{jk}]$ and $E[\tilde{W}_{jk}]$ represent the average virtual queuing delay and average virtual queue waiting time experienced by the secondary users in class C_k in the virtual queue of the frequency channel F_j . By applying the Mean Value Analysis (MVA) [Kle75], we have:

$$E[\tilde{D}_{jk}] = E[\tilde{W}_{jk}] + E[\tilde{X}_j] = \frac{\rho_j^2 + \sum_{l=2}^k \mu_{jl}^2}{2(1 - \rho_j - \sum_{l=2}^{k-1} \mu_{jl})(1 - \rho_j - \sum_{l=2}^k \mu_{jl})} + E[\tilde{X}_j], \quad (15)$$

where ρ_j represents the normalized loading of the primary user PU_j for the frequency channel F_j , and

$$\rho_j = \lambda_j^{PU} E[X_j^{PU}], \quad \rho_j^2 = \lambda_j^{PU} E[(X_j^{PU})^2]. \quad (16)$$

Recall that the average input rate of the primary user PU_j is λ_j^{PU} , and the first two moments of the service time is $E[X_j^{PU}]$ and $E[X_j^{PU2}]$.

Since the average virtual queuing delay $E[\tilde{D}_{jk}]$ is the average service time of the physical queue, the average end-to-end delay of the secondary user SU_i sending packets through frequency channel F_j is approximately:

$$E[D_{ij}] = \frac{E[\tilde{D}_{jk}]}{1 - \lambda_{ij}E[\tilde{D}_{jk}]}, \text{ for } \lambda_{ij}E[\tilde{D}_{jk}] < 1, SU_i \in C_k. \quad (17)$$

Strategies (s_i, s_{-i}) such that $\lambda_{ij}E[\tilde{D}_{jk}] \geq 1$ will result in an unbounded delay $E[D_{ij}]$, which is undesirable for delay-sensitive applications. The advantage of this approximation is that once the average delay of the virtual queue $E[\tilde{D}_{jk}]$ is known by the secondary user SU_i , the secondary user can immediately calculate the expected end-to-end delay $E[D_{ij}]$ of a packet transmitting using the frequency channel F_j . Note that in equation (17), we assume that once a packet selects a physical queue, it cannot switch to another queue (change position to the other queues). However, by considering current physical queue size q_{ij} for user SU_i using the frequency channel F_j , a packet can change its channel selection after it is put in the physical queue. The switching probability from a longer queue q_{ia} to a shorter queue q_{ib} in a time interval t can be defined as $1 - \exp(-t \times (q_{ia} - q_{ib}))$. To evaluate such expected end-to-end delay $E[D_{ij}]$, a more sophisticated queuing model with jockey impatient customers [Koe66] needs to be considered.

Let $P_{ij}(s_i, s_{-i})$ represent the probability of packet loss for the secondary user SU_i sending packets through frequency channel F_j . By applying G/G/1 approximation based on the work of [JTK01], we have:

$$P_{ij}(s_i, s_{-i}) = \begin{cases} \lambda_{ij}E[\tilde{D}_{jk}] \exp\left(-\frac{\lambda_{ij}E[\tilde{D}_{jk}] \times d_i}{E[D_{ij}]}\right), & \text{for } \lambda_{ij}E[\tilde{D}_{jk}] < 1, SU_i \in C_k \\ 1, & \text{for } \lambda_{ij}E[\tilde{D}_{jk}] \geq 1 \end{cases}. \quad (18)$$

For a delay-sensitive secondary user SU_i , the objective function in (5) becomes:

$$\begin{aligned} & \underset{s_i}{\text{maximize}} \quad U_i^{(1)}(s_i, s_{-i}) \\ \Rightarrow & \underset{s_i}{\text{maximize}} \quad \sum_{j=1}^M s_{ij}(1 - P_{ij}(s_i, s_{-i})) \\ \Rightarrow & \underset{s_i}{\text{minimize}} \quad \sum_{j=1}^M s_{ij} \lambda_{ij} E[\tilde{D}_{jk}] \exp\left(-\frac{\lambda_{ij}E[\tilde{D}_{jk}] \times d_i}{E[D_{ij}]}\right), \text{ for } SU_i \in C_k \end{aligned} \quad (19)$$

C. Information overhead and the aggregate virtual queue effects

In the previous subsection, we calculate $P_{ij}(s_i, s_{-i})$, the packet loss probability for a

packet of the secondary user SU_i transmitting using the frequency channel F_j . In a general case, we can calculate the expected utility function of equation (4) as:

$$\begin{aligned} E[u_i(\mathbf{a}_i, \mathbf{a}_{-i})] &= \theta_i \cdot U_i^{(1)} + (1 - \theta_i) \cdot U_i^{(2)} \\ &= \theta_i \sum_{j=1}^M s_{ij} \cdot (1 - P_{ij}(\mathbf{s}_i, \mathbf{s}_{-i})) + (1 - \theta_i) \sum_{j=1}^M s_{ij} T_{ij} (1 - p_{ij}) / T_i^{\max}, \\ &= \sum_{j=1}^M s_{ij} \cdot E[V_{ij}(\mathbf{a}_i, \mathbf{a}_{-i})] \end{aligned} \quad (20)$$

where $E[V_{ij}(\mathbf{a}_i, \mathbf{a}_{-i})] = \theta_i(1 - P_{ij}(\mathbf{s}_i, \mathbf{s}_{-i})) + (1 - \theta_i)T_{ij}(1 - p_{ij}) / T_i^{\max}$. $E[V_{ij}(\mathbf{a}_i, \mathbf{a}_{-i})]$ represents the aggregate virtual queue effect for the secondary user SU_i of class C_k transmitting using the frequency channel F_j . Note that $E[V_{ij}(\mathbf{a}_i, \mathbf{a}_{-i})] \leq 1$.

The aggregate virtual queue effect $E[V_{ij}(\mathbf{a}_i, \mathbf{a}_{-i})]$ can be regarded as a metric of the dynamic wireless environment and the competing wireless users' behaviors [Hay05][MM99], which reflects the impact of the time-varying environment and the impact of the other users (including the primary user and the other secondary users) on the secondary user SU_i in the specific frequency channels F_j . To evaluate $E[V_{ij}(\mathbf{a}_i, \mathbf{a}_{-i})]$, modeling other secondary users is necessary⁹. Our priority virtual queue interface requires the following information to compute μ_{jl} and μ_{jl}^2 in (15):

1. Priority: the secondary users' priorities.
2. Normalized loading: the secondary users' normalized loading parameters $\lambda_{ij} \times E[\tilde{X}_j]$, which not only include the information of s_i , but also reflects the input traffic loading and the expected transmission time using a specific frequency channel.
3. Variance statistics: the secondary users' variance statistics with the normalized parameter $\lambda_{ij} \times E[\tilde{X}_j^2]$.

To determine the above information, two kinds of information need to be exchanged:

- Information exchange of other secondary users' traffic specification $\mathbf{TS}_{-i}$ (see Section IV.A).
- Information exchange of the action of the other secondary users $\mathbf{a}_{-i}$ (to model the

⁹ Although we apply M/G/1 priority queuing analysis, more sophisticated queuing models can be applied for evaluating the aggregate virtual queue effects, if using different traffic model description.

strategies $\mathbf{s}_{-i}$).

Since the traffic specification TS_i only varies when the frequency channels change dramatically (we do not consider mobility effects and this information exchange is assumed to be truthfully revealed), the traffic specification can be exchanged only when a secondary user joins the network to reduce the overhead. On the other hand, the action information can be observed (sensed) more frequently (once per packet/service interval [IEE03]). Note that since the users in the higher priority classes will not be affected by the users in the lower priority classes, they do not need the information from the users in a lower priority class. Hence, higher priority secondary users will have small information exchange overhead and computational complexity. In conclusion, the information overheads for higher importance secondary users are limited.

Based on the action information observation, the interface updates the strategies $(\mathbf{s}_i, \mathbf{s}_{-i})$ and compute all the required information to evaluate the aggregate virtual queue effect $E[V_{ij}(\mathbf{a}_i, \mathbf{a}_{-i})]$. Next, we discuss how to make use of $E[V_{ij}(\mathbf{a}_i, \mathbf{a}_{-i})]$ to determine the frequency channel selection.

V. DYNAMIC CHANNEL SELECTION WITH STRATEGY LEARNING

From Section III, we know that the goals of the secondary users are to maximize their utility functions. We define the *best response strategy* for the decentralized optimization by considering the strategy that yields the highest utility U_i of the secondary user SU_i . To simplify the description, we now consider all the secondary users in one class¹⁰. The decentralized optimization is:

$$\mathbf{s}_i^* = \arg \max_{\mathbf{s}_i \in \mathcal{S}_i^M} E_{(\mathbf{s}_i, \mathbf{s}_{-i})}[u_i(\mathbf{a}_i, \mathbf{a}_{-i})]. \quad (21)$$

From equation (20), the decentralized optimization problem in equation (21) can be written as:

¹⁰ For multiple priority classes' case, the same algorithm can be applied consecutively from higher priority classes to lower priority classes without losing generality.

$$\mathbf{s}_i^* = \arg \max_{\mathbf{s}_i \in \mathcal{S}^M} \sum_{j=1}^M s_{ij} \cdot E[V_{ij}(\mathbf{a}_i, \mathbf{a}_{-i})]. \quad (22)$$

Based on the strategy $\mathbf{s}_i^*$, a secondary user can choose its action (frequency channel), and then the secondary user models $\mathbf{s}_{-i}$ based on the action information exchange revealed by the other secondary users (i.e. $\mathbf{a}_{-i}$) in order to evaluate a new $E[V_{ij}(\mathbf{a}_i, \mathbf{a}_{-i})]$. The concept is similar to the fictitious play [FL98] in multi-agent learning in game theory. The difference is that a user not only models the strategies of the other users, but also explicitly calculates the aggregate virtual queue effect $E[V_{ij}(\mathbf{a}_i, \mathbf{a}_{-i})]$ that directly impacts the utility function. Based on the priority queuing analysis in Section IV, the aggregate virtual queue effect $E[V_{ij}(\mathbf{a}_i, \mathbf{a}_{-i})]$ can be evaluated using equation (20) by each of the secondary users. The iterative learning algorithm based on $E[V_{ij}(\mathbf{a}_i, \mathbf{a}_{-i})]$ can be written as:

$$\begin{aligned} \mathbf{s}_i^*(n) &= \arg \max_{\mathbf{s}_i \in \mathcal{S}^M} U_i(\mathbf{s}_i, \mathbf{s}_{-i}(n-1)) \\ &= \arg \max_{\mathbf{s}_i \in \mathcal{S}^M} \sum_{j=1}^M s_{ij} \cdot E[V_{ij}(\mathbf{a}_i(n-1), \mathbf{a}_{-i}(n-1))], \end{aligned} \quad (23)$$

where the initial stage is $\mathbf{s}_i(0)$. We show the system diagram of a secondary user in Figure 6.4. The optimal strategy $\mathbf{s}_i^*$ can be determined by the secondary user SU_i for a given $E[V_{ij}(\mathbf{a}_i, \mathbf{a}_{-i})]$ from the interface. Then, based on the best response strategy $\mathbf{s}_i^*(n)$, a packet of the secondary user SU_i selects an action $\mathbf{a}_i(n)$.

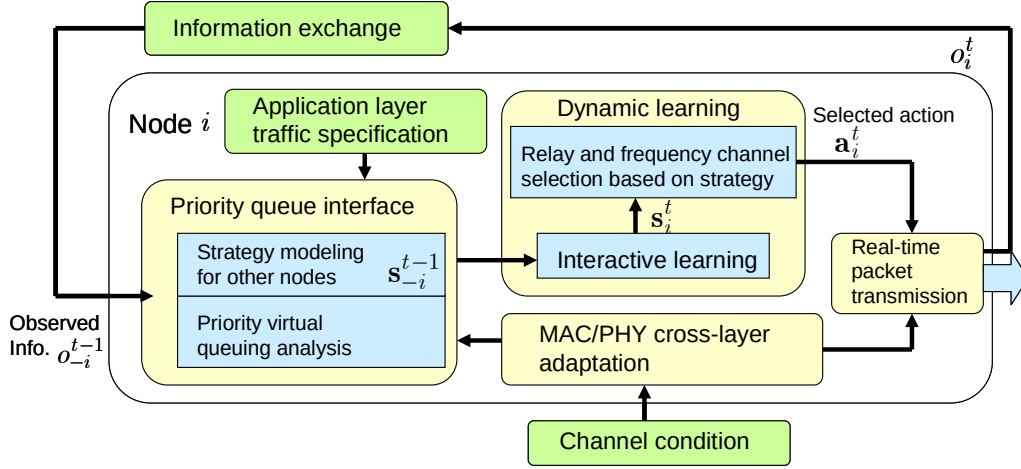


Fig 6.4. The block diagram of the priority virtual queue interface and dynamic strategy learning of a secondary user.

Let the frequency channel with the largest $E[V_{ij}(\mathbf{A}(n-1))]$ be $F^*(n)$, i.e. $F^*(n) = \arg \max_{F_j \in \mathbf{F}_i} \{E[V_{ij}(\mathbf{A}(n-1))]\}$. Recall that $\mathbf{A}(n-1) = [\mathbf{a}_i(n-1), \mathbf{a}_{-i}(n-1)]$. The solution of (23) is:

$$\mathbf{s}_i^*(n) = \begin{cases} s_{ij} = 1, & \text{if } F_j = F^*(n) \\ s_{ij} = 0, & \text{otherwise} \end{cases}. \quad (24)$$

For a specific frequency channel F_j , even though the corresponding primary user's traffic is stationary, it is not guaranteed that the secondary users' strategies will converge to a steady state, since the secondary users mutually impact each other. Hence, our solution adopts a multi-agent learning which resembles the gradient play [FL98] in game theory. Our approach does not employ a best response strategy, but rather adjusts a strategy in the direction of the perceived "better" response. In addition, due to the cost of frequency hopping and the hardware limitations, only a limited set of selectable frequency channels can be selected by a secondary user for transmission. Hence, we assume that the selectable frequency channels for the secondary user SU_i are in a set $\mathbf{F}_i \subseteq \mathbf{F}$. Let us denote $\mathbf{H}_i = \{F_j \mid s_{ij} > 0\} \subseteq \mathbf{F}_i$ as the set of frequency channels with $s_{ij} > 0$. The maximum number of selected frequency channel is H_i , i.e. $|\mathbf{F}_i| \leq H_i$.

Note that changing the selected frequency channels requires channel sensing, control signaling, and also additional incurred delays, etc. for the spectrum handoff [ALV06]. In

Appendix, we discuss the convergence properties of the proposed algorithm considering the cost of changing the frequency selection strategy. We refer to this cost for the secondary user SU_i as $\chi_i(s_i(n), s_i(n-1))$, which is a function of the difference between the selected strategy and the previous strategy (see Appendix for more detail). The utility function of SU_i now becomes

$$U_i(s_i(n), s_{-i}(n-1)) = \sum_{j=1}^M s_{ij}(n) \times E[V_{ij}(\mathbf{A}(n-1))] - \chi_i(s_i(n), s_i(n-1)).$$

The steps in our DSL algorithm are summarized below:

Algorithm 6.1 DSL algorithm

Step 1. Model the strategy matrix from the action information exchange:

The priority virtual queue interface collects the action information from the other users and accordingly updates the strategy matrix.

Step 2. Calculate virtual queue effects:

Given the strategy matrix of the previous stage, $\mathbf{S}(n-1) = [s_i(n-1), s_{-i}(n-1)]$ and the channel loading specification, we calculate the aggregate virtual queue effects $E[V_{ij}(\mathbf{A}(n-1))]$ based on equation (18) and (20).

Step 3. Determine the set of selected frequency channels:

Determine the set $\mathbf{H}_i$ of selected frequency channels from $\mathbf{F}_i$:

$$\mathbf{H}_i(n) = \arg \max_{F_j \in \mathbf{F}_i}^{(H_i)} \{E[V_{ij}(\mathbf{A}(n-1))]\}, \quad (25)$$

where we denote the operation $\max^{(N)}(X)$ as the largest N choices from a set X . Recall that the frequency channel with the largest $E[V_{ij}(\mathbf{A}(n-1))]$ be $F^*(n)$.

Step 4. Determine the channel selection strategies:

Based on $\mathbf{H}_i(n)$, we determine the strategy $\tilde{s}_{ij}(n)$ using the following policy:

$$\tilde{s}_{ij}(n) = \begin{cases} \max(0, s_{ij}(n-1) - \sigma) & , \text{ if } F_j \in \mathbf{H}_i(n), F_j \neq F^*(n) \\ 1 - \sum_{F_j \neq F^*(n)} \max(0, s_{ij}(n-1) - \sigma) & , \text{ if } F_j \in \mathbf{H}_i(n), F_j = F^*(n), \\ 0 & , \text{ if } F_j \notin \mathbf{H}_i(n) \end{cases} \quad (26)$$

where σ is a constant step size of changing the strategies such that the policy favors a frequency channel leading to a larger $V_{ij}(\mathbf{S}(n-1))$. Specifically, the policy concentrates

the traffic distribution to the frequency channel $F^*(n)$ from the other frequency channels in H_i , while learning from the previous strategy $s_{ij}(n-1)$.

Step 5. Update the new strategy:

Update the new strategy $s_{ij}(n)$ if the strategy $\tilde{s}_{ij}(n)$ leads to an improved utility.

$$s_{ij}(n) = \begin{cases} \tilde{s}_{ij}(n), & \text{if } U_i(\tilde{s}_i(n), \mathbf{s}_{-i}(n-1)) > U_i(\mathbf{s}_i(n-1), \mathbf{s}_{-i}(n-1)) \\ s_{ij}(n-1), & \text{otherwise} \end{cases}. \quad (27)$$

Step 6. Determine a frequency channel for packet transmissions based on the strategy.

The proposed dynamic channel selection algorithm has the following advantages:

1. Decentralized decision making allows heterogeneous secondary users (in terms of their priorities, utilities, source traffic and channel conditions) to optimize their own utility functions based on the information exchanges.
2. Virtual queuing analysis provides the expected utility impacted by other users using the same frequency channel and hence, simplifies the required information exchange.
3. The iterative algorithm provides real-time adaptation to the changing network conditions and source traffic variations of the primary users or other secondary users.

VI. SIMULATION RESULTS

First, we simulate a simple network with two secondary users and three frequency channels (i.e. $N = 2$, $M = 3$) in order to show the results of our solution using a simple example such that the behavior of the proposed cognitive radio model can be clearly understood. We assume that each secondary user can choose all the frequency channels, i.e. $H_i = 3$. The two secondary users are in the same priority class. The simulation parameters of the secondary users are presented in Table 6.1 including the channel conditions $R_{ij} = [T_{ij}, p_{ij}]$, and initial strategies $s_i(0)$, etc. The average packet lengths are assumed to be 1000 bytes and the delay deadlines are assumed to be 0.5 sec for all users. The normalized traffic statistics of the primary users are in Table 6.2. Given these statistics, Figure 6.5 provides the analytical experienced delays $E[D_{ij}]$ (using equation

(17)) that are bounded by the delay deadlines for the two secondary users using different strategy pairs (s_{1j}, s_{2j}) in the three frequency channels. Importantly, a strategy pair (s_{1j}, s_{2j}) that results in an unbounded $E[D_{ij}]$ will make the utility function drop abruptly for delay-sensitive applications (see equation (2)), which is undesirable for these secondary users. Hence, equation (17) provides the analytical operation points for the strategy pairs. In the following subsection, each secondary user applies the proposed DSL algorithm from a uniform traffic distribution over the three channels to find the channel selection strategies.

TABLE 6.1 SIMULATION PARAMETERS OF THE SECONDARY USERS.

Secondary users	Physical transmission rate T_{ij} (Mbps)			Physical packet error rate p_{ij}			Initial strategy $s_{ij}(0)$			Satisfaction rate $T_i^{\max} = 3B_i$ (Mbps)	Rate requirement B_i (Mbps)
	F_1	F_2	F_3	F_1	F_2	F_3	F_1	F_2	F_3		
SU_1	1.90	1.21	1.78	0.09	0.16	0.12	1/3	1/3	1/3	2.77	0.92
SU_2	0.46	0.97	1.52	0.01	0.09	0.15	1/3	1/3	1/3	2.21	0.74

TABLE 6.2 SIMULATION PARAMETERS OF THE PRIMARY USERS.

Primary users	Normalized loading ρ_j	Second moment normalized loading ρ_j^2
PU_1	0.2	1×10^{-4}
PU_2	0.1	1×10^{-4}
PU_3	0.3	1×10^{-4}

A. Impact of the delay sensitivity preference of the applications

In this simulation, we show that the delay sensitivity preferences of the secondary users affect the stability of utility and also the resulting channel selection strategies. Figure 6.6 gives the strategies and the resulting utilities of the two secondary users with two different θ_i (applications that care less about delay with $\theta_i = 0.2$, $i = 1, 2$ in Figure 6.6 (a) and applications that care more about delay with $\theta_i = 0.8$, $i = 1, 2$ in Figure 6.6 (b)).

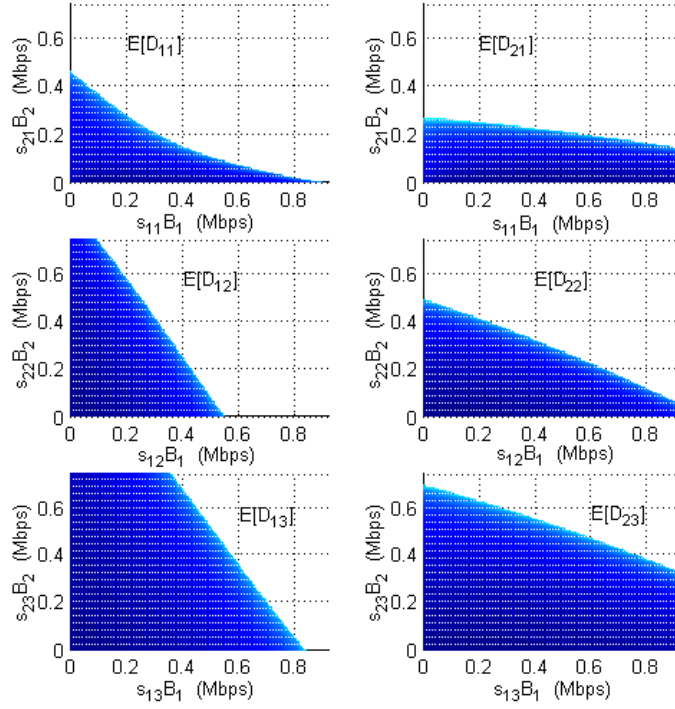


Fig. 6.5 Analytical expected delay of the secondary users with various strategies in different frequency channels, shadow part represents a bounded delay below the delay deadline (stable region).

The delay-sensitive applications in Figure 6.6 (b) do not achieve a steady state, since the small changes in the channel selection strategies can push the experienced delay over the delay deadline and hence, impact the utility function dramatically. Moreover, compared with the resulting strategies of the applications in Figure 6.6 (a), Figure 6.6 (b) shows that the delay-sensitive applications prefer a channel without other secondary users to transmit the data – SU_1 transmits most of its data through channel F_1 , while SU_2 transmits through F_2 and F_3 (i.e. $s_{11} \cong 1$, $s_{21} \cong 0$). This is because for a secondary user with delay sensitive applications, the utility function is more sensitive to the traffic in a frequency channel. The data traffic from other secondary users can increase the uncertainty of the channel, which makes such channel undesirable for the delay sensitive applications. Moreover, the resulting utility is more unstable for the applications with a larger θ_i . The resulting strategy $(s_{11}, 0)$, $(0, s_{22})$, and $(0, s_{23})$ of Figure 6.6 (b) are closer to the region with unbounded delay for $E[D_{11}]$, $E[D_{22}]$, and $E[D_{23}]$ (see Figure 6.5).

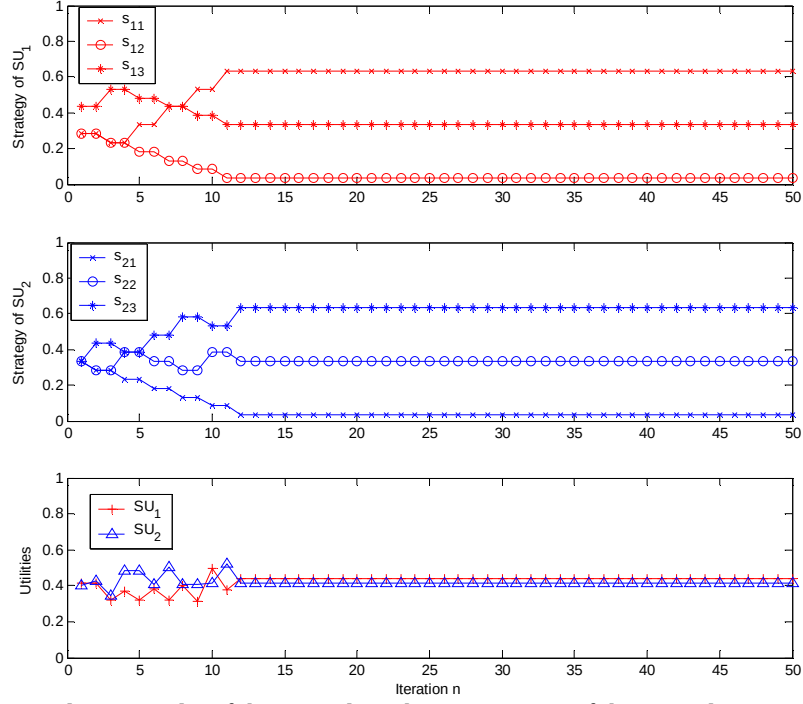


Fig. 6.6 (a) Simulation results of the DSL algorithm – strategies of the secondary users and the utility functions of less delay-sensitive applications ($\theta_i = 0.2, \sigma = 0.05, \chi_{ij} = 0$).

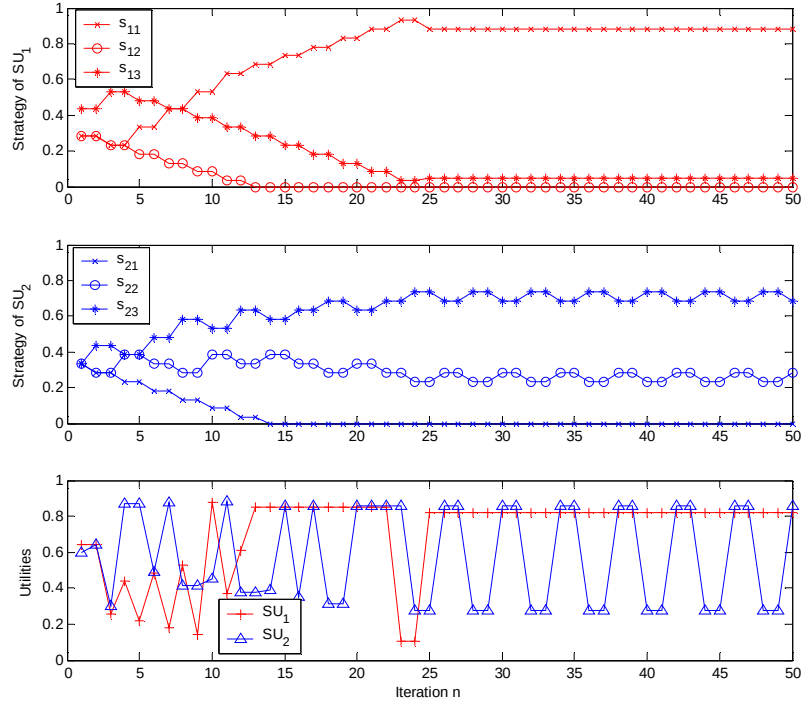


Fig. 6.6(b) Simulation results of the DSL algorithm – strategies of the secondary users and the utility functions of delay-sensitive applications ($\theta_i = 0.8, \sigma = 0.05, \chi_{ij} = 0$).

B. Impact of the primary users

Next, we simulate the impact of the highest priority users – the primary users in Figure 6.7. We change the normalized traffic loading of PU_1 in the frequency channel F_1 from 0 to 1 and fix the normalized loading of the other two primary users as in Table II. Due to the priority queuing, we know that once ρ_1 reaches 1, frequency channel F_1 is not accessible for the secondary users. For different normalized loading of PU_1 , Figure 6.7 shows the resulting strategies and the utilities of the two secondary users after convergence. Both s_{11} and the utility value U_1 decreases when the available resource from F_1 decreases ($\rho_1 > 0.6$). Interestingly, even though SU_2 does not utilize channel F_1 ($s_{21} \cong 0$) and the resulting strategies do not change with ρ_1 , U_2 also decreases. This is because more traffic from SU_1 will now be distributed to F_2 and F_3 . This simple example illustrates that the traffic of a higher priority class user can still affect the utilities of the secondary users in lower priority classes even when these secondary users avoid selecting the same channels as the higher priority class user.

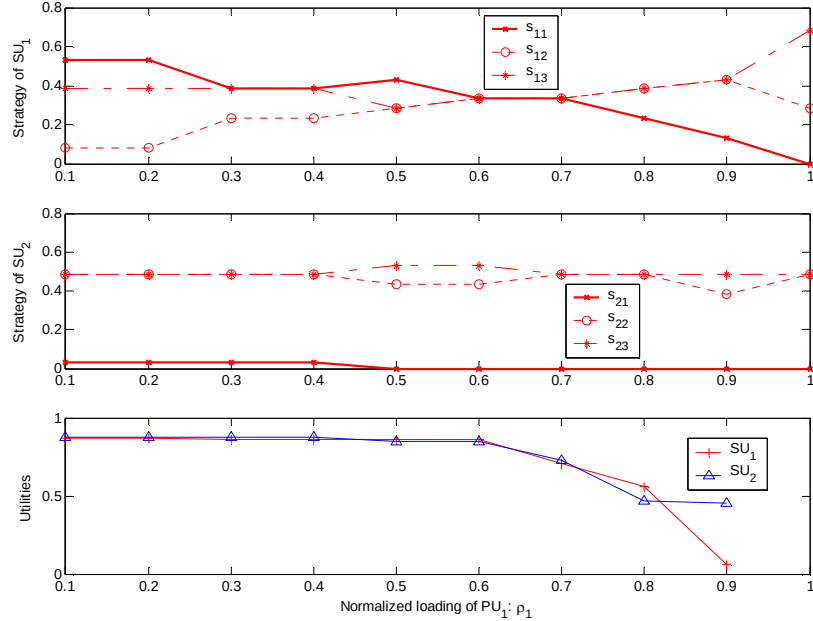


Fig. 6.7 Steady state strategies of the secondary users and the utility functions vs. the normalized loading of PU_1 for delay-sensitive applications ($\theta_i = 0.8$, $\sigma = 0.05$, $\chi_{ij} = 0.02$).

C. Comparison with other cognitive radio resource management solutions

In this subsection, we simulate a larger number of secondary users and a larger number of frequency channels. First, we look at the case with 6 secondary users with video streaming applications (“Coastguard”, frame rate of 30Hz, CIF format, delay deadline 500ms) sharing 10 frequency channels ($N = 6, M = 10, \theta_i = 1$). We compare our DSL algorithm with other two resource allocation algorithms – the “Static Assignment” [TJ91] and the “Dynamic Least Interference” [KP99]. In the “Static Assignment” algorithm, a secondary user will statically select a frequency channel with the best effective transmission rate without interacting with other secondary users. This work has the drawback that it is merely a decentralized scheme without any information exchange. In the “Dynamic Least Interference” algorithm, a secondary user will dynamically select a single frequency channel that has the least interference from the other users (both secondary users and primary users), which is also similar to the rule D in [ZC05]. This work has the drawback of considering only the interference and the resulting throughput in the physical layer. We simulate 100 different frequency channel conditions as well as the traffic loadings and then compute the average the video PSNR and the standard deviation of the PSNR over the one hundred cases in Table 6.3 for the 6 video applications. Unlike the “Dynamic Least Interference” that only considers the interference and the resulting throughput in the physical layer, our proposed multi-agent learning algorithm tracks the strategies of the other users through information exchange and adequately adapts the channel selection to maximize the multimedia utility in the application layer. The results show that our DSL algorithm outperforms the other two algorithms for delay-sensitive multimedia applications in terms of packet loss rate (PLR) and video quality (PSNR).

Next, we simulate the case with 20 secondary users with video streaming applications ($\theta_i = 1$) mixed with r secondary users with delay insensitive ($\theta_i = 0$) applications. These secondary users are in the same priority class and share 10 frequency channels.

The average T_{ij} of the frequency channels is now set to 3 Mbps, instead of 1.25 Mbps and 1 Mbps in the previous simulation. Table 6.4 shows the average packet loss rate and the average PSNR over the 20 video streams (instead of over 100 different channel conditions in the previous simulation) with different r for the three solutions. Larger r reduces the available resources that can be shared by the video streams, and hence, decreases the received video quality. The results show that the video streaming of the “Static Assignment” is impacted severely by the different channel conditions to the secondary users. The standard deviations of the “Static Assignment” are larger than the results of the “Dynamic Least Interference” and our DSL algorithm. The results again show that our DSL algorithm outperforms the other two algorithms for multimedia applications in terms of packet loss rate and video quality.

TABLE 6.3 COMPARISONS OF THE CHANNEL SELECTION ALGORITHMS FOR DELAY-SENSITIVE APPLICATIONS WITH $N = 6, M = 10$.

Medium bandwidth case: (average $T_{ij} = 1.25$ Mbps)	“Static Assignment – Largest Bandwidth”			“Dynamic Least Interference”			“Dynamic Learning Algorithm”		
	PLR	Average Y-PSNR (dB)	Y-PSNR Standard Deviation	PLR	Average Y-PSNR (dB)	Y-PSNR Standard Deviation	PLR	Average Y-PSNR (dB)	Y-PSNR Standard Deviation
SU_1	15.24 %	32.93	3.92	17.44 %	32.55	3.49	7.61 %	34.17	1.52
SU_2	25.38 %	31.48	4.31	19.80 %	32.20	3.45	8.74 %	33.97	1.82
SU_3	21.34 %	32.03	4.24	15.45 %	32.86	3.50	11.85 %	33.44	2.28
SU_4	20.38 %	32.17	4.35	12.98 %	33.26	3.40	8.22 %	34.06	1.77
SU_5	27.17 %	31.21	4.29	20.56 %	32.09	3.55	12.61 %	33.32	2.21
SU_6	19.26 %	32.32	4.33	12.86 %	33.27	3.61	9.38 %	33.86	2.27
Low bandwidth case: (average $T_{ij} = 1$ Mbps)	“Static Assignment -Largest-Bandwidth”			“Dynamic Least Interference”			“Dynamic Learning Algorithm”		
	PLR	Average Y-PSNR (dB)	Y-PSNR Standard Deviation	PLR	Average Y-PSNR (dB)	Y-PSNR Standard Deviation	PLR	Average Y-PSNR (dB)	Y-PSNR Standard Deviation
SU_1	42.01 %	29.48	4.94	38.16 %	29.89	4.32	18.30 %	32.42	1.97
SU_2	38.21 %	29.90	4.89	34.07 %	30.35	4.29	17.02 %	32.62	2.42
SU_3	39.97 %	29.69	5.02	33.85 %	30.37	4.41	18.76 %	32.36	2.26
SU_4	32.30 %	30.59	4.98	29.74 %	30.87	4.37	16.12 %	32.75	2.31
SU_5	42.19 %	29.48	4.98	38.34 %	29.87	4.41	18.45 %	32.40	2.33
SU_6	37.07 %	30.01	5.04	31.52 %	30.65	4.46	19.40 %	32.26	2.67

TABLE 6.4 COMPARISONS OF THE CHANNEL SELECTION ALGORITHMS FOR DELAY-SENSITIVE APPLICATIONS WITH $N = 20 + r, M = 10$, WHERE r IS THE SECONDARY USERS WITH DELAY INSENSITIVE $\theta_k = 0$ APPLICATIONS.

Average T_{ij} = 3 Mbps	“Static Assignment – Largest Bandwidth”			“Dynamic Least Interference”			“Dynamic Learning Algorithm”		
	PLR	Average Y-PSNR (dB)	Y-PSNR Standard Deviation	PLR	Average Y-PSNR (dB)	Y-PSNR Standard Deviation	PLR	Average Y-PSNR (dB)	Y-PSNR Standard Deviation
$r = 2$	20.00%	28.49	14.24	12.64%	33.76	2.59	0.06%	35.60	0.0013
$r = 5$	35.00%	23.15	16.98	15.81%	33.30	2.83	2.86%	35.23	1.64
$r = 10$	50.00%	17.81	17.80	24.34%	32.32	3.36	8.12%	34.50	2.55

VII. CONCLUSIONS

In this chapter, we propose a priority virtual queue interface for heterogeneous multimedia users in cognitive radio networks, based on which they can exchange information and time share the various frequency channels in a decentralized fashion. Based on the information exchange, the secondary users are able to evaluate the expected utility impact from the dynamic wireless environment as well as the competing wireless users’ behaviors and learn how to efficiently adapt their channel selection strategies. We focus on delay-sensitive multimedia applications, and propose a dynamic learning algorithm based on the priority queuing analysis. Importantly, unlike conventional channel allocation schemes that select the least interfered channel merely based on the channel estimation, the proposed multi-agent learning algorithm allows the secondary users to track the actions of the other users and adequately adapt their own strategies and actions to the changing multi-user environment. The results show that our proposed solution outperforms the fixed channel allocation and the dynamic channel allocation that selects the least interfered channel, in terms of video quality. Without primary users using the highest priority class, the proposed approach can also be used to support QoS for general multi-radio wireless networks. This situation also emerges in wireless systems such as those discussed in [ALV06], where the secondary users are competing in the

unlicensed band (i.e. ISM band) and there is no primary user. The proposed DSL algorithm can be implemented by the secondary users to switch channels, suspend/resume channel operation, and add/remove channels, etc., while complying with emerging MAC solutions for cognitive radio networks [CCB06].

VIII. APPENDIX

CONVERGENCE OF THE DECENTRALIZED APPROACH

If we consider the additional cost (penalty) $\chi_i(\mathbf{s}_i, \mathbf{s}_i(n-1))$ when the channel selection strategies are not the same, equation (23) can be rewritten as:

$$\begin{aligned} \mathbf{s}_i^*(n) &= \arg \max_{\mathbf{s}_i \in \mathcal{S}^M} U_i(\mathbf{s}_i, \mathbf{s}_{-i}(n-1)) \\ &= \arg \max_{\mathbf{s}_i \in \mathcal{S}^M} \left(\sum_{j=1}^M s_{ij} \cdot E[V_{ij}(\mathbf{A}(n-1))] - \chi_i(\mathbf{s}_i, \mathbf{s}_i(n-1)) \right), \end{aligned} \quad (28)$$

For example, the penalty function can be

$$\chi_i(\mathbf{s}_i, \mathbf{s}_i(n-1)) = \begin{cases} \chi_i^+, & \text{if } s_{ij}(n-1) = 0, s_{ij} > 0 \\ \chi_i^-, & \text{if } s_{ij}(n-1) > 0, s_{ij} = 0, \\ 0, & \text{otherwise} \end{cases} \quad (29)$$

where χ_i^+ and χ_i^- represent the cost of selecting a new channel and the cost of hopping away from a used frequency channel.

From equation (28), the secondary user SU_i will keep updating its channel for transmission, unless the utility difference of selecting a new strategy $\mathbf{s}_i^*(n)$ becomes small. Hence, in the proposed DSL algorithm in Section V, assume the difference between the estimated strategy $\tilde{s}_{ij}(n)$ and the previous strategy $s_{ij}(n-1)$ is $e_{ij}(n)$ for SU_i using the frequency channel F_j , i.e. $e_{ij}(n) = \tilde{s}_{ij}(n) - s_{ij}(n-1)$. Let $U_i^{\text{diff}}(n) = \sum_{j=1}^M e_{ij}(n) \times E[V_{ij}(\mathbf{A}(n-1))]$ be the utility difference between the estimated strategy and the previous strategy.

Claim 1: If $e_{ij}(n)$ satisfies the following condition:

$$U_i^{\text{diff}}(n) \leq \chi_i(\tilde{\mathbf{s}}_i(n), \mathbf{s}_i(n-1)), \quad (30)$$

for all the secondary users, the channel selection strategies converge to a steady state.

Proof: Equation (30) can be derived as:

$$\begin{aligned}
\chi_i(\tilde{s}_i(n), s_i(n-1)) &\geq \sum_{j=1}^M (\tilde{s}_{ij}(n) - s_{ij}(n-1)) \times E[V_{ij}(\mathbf{A}(n-1))] \\
&\Rightarrow \sum_{j=1}^M s_{ij}(n-1) \times E[V_{ij}(\mathbf{A}(n-1))] \geq \sum_{j=1}^M \tilde{s}_{ij}(n) \times E[V_{ij}(\mathbf{A}(n-1))] - \chi_i(\tilde{s}_i(n), s_i(n-1)) \\
&\Rightarrow U_i(s_i(n-1), s_{-i}(n-1)) \geq U_i(\tilde{s}_i(n), s_{-i}(n-1))
\end{aligned}$$

From Step 5 of the DSL algorithm in Section V, the strategies will remain unchanged and converge to a steady state. ■

Claim 2: If the penalty function $\chi_i(s_i, s_i(n-1))$ is a convex function of s_i , when the DSL algorithm converges to a steady state, the channel selection strategy s_i^* is the best response strategy that maximizes U_i .

Proof: As long as the penalty function $\chi_i(s_i, s_i(n-1))$ is a convex function of s_i , the utility function $U_i(s_i, s_{-i}(n-1))$ is a concave function, since for each iteration, the $E[V_{ij}(\mathbf{A}(n-1))]$ in equation (28) does not change with s_i . Hence, when the DSL algorithm converges to a steady state, the local optimum in equation (28) converges to the global optimum. ■

Chapter 7

Resource Management in Multi-hop Cognitive Radio Networks

I. INTRODUCTION

The majority of the resource management research in cognitive radio networks has focused on a single-hop wireless infrastructure [CCB06][ZTS07][NH07][HPR07]. In this chapter, we focus on the resource management problem in the more general setting of multi-hop cognitive radio networks. A key advantage of such flexible multi-hop infrastructures is that the same infrastructure can be re-used and reconfigured to relay the content gathered by various transmitting users (e.g. sources nodes) to their receiving users (e.g. sinks nodes). These users may have different goals (application utilities etc.) and may be located at various locations. For the multi-hop infrastructure, there are three key differences as opposed to the single-hop case. First, the users have as available network resources not only the vacant frequency channels (spectrum holes or spectrum opportunities [Hay05][CCB06]) as in the single-hop case, but also the routes through the various network relays to the destination nodes. Second, the transmission strategies will need to be adapted not only at the source nodes, but also at the network relay nodes. In cognitive radio networks, network nodes are generally capable of sensing the spectrum and modeling the behavior of the primary users and thereby, identifying the available spectrum holes. In multi-hop cognitive radio networks, the network nodes will also need to model the behavior of the other neighbor nodes (i.e. other secondary users) in order to successfully optimize the routing decisions. In other words, network relays also require a learning capability in the multi-hop cognitive radio network. Third, to learn and efficiently adapt their decisions over time, the wireless nodes need to possess accurate

(timely) information about the channel conditions, interference patterns and other nodes transmission strategies. However, in a distributed setting such as a multi-hop cognitive radio network, the information is decentralized, and thus, there is a certain delay associated with gathering the necessary information from the various network nodes. Hence, an effective solution for multi-hop cognitive radio networks will need to tradeoff the “value” of having information about other nodes versus the transmission overheads associated with gathering this information in a timely fashion across different hops, in terms the utility impact.

In this chapter, we aim at learning the behaviors of interacting cognitive radio nodes that use simple interference graph (similar to the spectrum holes used in [CCB06][ZTS07]) to sequentially adjust and optimize their transmission strategies. We apply a multi-agent learning algorithm – the fictitious play (FP) [FL98] to model the behavior of neighbor nodes based on the information exchange among the network nodes. We focus on delay-sensitive applications such as real-time multimedia streaming, i.e. the receiving users need to get the transmitted information within a certain delay. Due to the informationally decentralized nature of the multi-hop wireless networks, a centralized resource management solution for these delay-constrained applications is not practical [SV07b], since the tolerable delay does not allow propagating information back and forth throughout the network to a centralized decision maker. Moreover, the complexity and the information overhead of the centralized optimization grow exponentially with the size of the network. The problem is further complicated by the dynamic competition for wireless resources (spectrum) among the various wireless nodes (i.e. source nodes/relays). The centralized optimization will require a large amount of time to process and the collected information will no longer be accurate by the time transmission decisions need to be made. Hence, a distributed resource management solution, which explicitly considers the availability of information, the transmission overheads and incurred delays, as well as the value of this information in terms of the utility impact is necessary.

The chapter is organized as follows. In Section II, we discuss the main challenges of the dynamic resource management in multi-hop cognitive radio networks and the related works. Section III provides the multi-hop cognitive radio network settings and strategies and Section IV gives problem formulation of the distributed resource management for delay sensitive transmission in such networks. In Section V, we determine how to quantify the rewards and costs associated with various information exchanges in the multi-hop cognitive radio networks. In Section VI, we propose our distributed resource management algorithms with the information exchange and introduce the adopted multi-agent learning approach – adaptive fictitious play in the proposed algorithms. Simulation results are in Section VII. Finally, Section VIII concludes the chapter.

II. MAIN CHALLENGES AND RELATED WORKS

A. *Main challenges in multi-hop cognitive radio networks*

To design such a distributed resource management in multi-hop cognitive radio networks, several main challenges need to be addressed:

- **Dynamic adaptation to a time-varying network environment**

Multi-hop cognitive radio networks are generally experiencing the following dynamics: 1) the primary users directly affect the spectrum opportunities available for the secondary users, 2) the mobility of the network relays that affects the network topology, 3) the traffic load variation due to multiple applications simultaneously sharing the same network infrastructure, and 4) the time-varying wireless channel conditions. Given the dynamic nature of the cognitive radio networks, wireless nodes need to learn, dynamically self-organize and strategically adapt their transmission strategies to the available resources without interfering the primary licensees. Due to these time-varying dynamics, the outcomes of these interactions do not need to converge to an equilibrium, i.e., disequilibrium and perpetual adaptation of strategies may persist, as long as the performance of the delay sensitive application is maximized [FL98]. Hence, repeated

information exchange among network nodes is required for nodes to efficiently learn and keep adapting to the changing network dynamics.

- **Information availability in multi-hop infrastructures**

Due to the informationally-decentralized nature of the multi-hop infrastructure, the exchanged network information is only useful when it can be conveyed in time. The timeliness constraint of the information exchange depends on the delay deadline of the applications, the information overhead, and the condition of the network links, etc. Hence, the value of information in terms of its impact on the users' utilities will need to be quantified for the different settings of the multi-hop cognitive radio network. This information will impact the accuracy with which the wireless nodes can model the behavior of other nodes (including the primary users) and hence, the efficiency with which they can respond to this environment by adequately optimizing their transmission strategies.

B. Related work

Distributed dynamic spectrum allocation is an important issue in cognitive radio networks. Various approaches have been proposed in recent years. In [ZTS07], a decentralized cognitive MAC protocols are proposed based on the theory of Partially Observable Markov Decision Process (POMDP), where a secondary user is able to model the primary users through Markovian state transition probabilities. In [NH07], the authors investigated a game-theoretic spectrum sharing approach, where the primary users are willing to share spectrum and provide a determined pricing function to the secondary users. In [HPR07], a no-regret learning approach is proposed for dynamic spectrum access in cognitive radio networks. However, these studies focus on dynamic spectrum management for the single-hop network case.

Exploiting frequency diversity in wireless multi-hop networks has attracted enormous interests in recent years. In [LL06], the authors propose a distributed allocation scheme of sub-carriers and power levels in an orthogonal frequency-division multiple-access-based

(OFDMA) wireless mesh networks. They proposed a fair scheduling scheme that hierarchically decouples the sub-carrier and power allocation problem based on the limited local information that is available at each node. In [WYT06], the authors focus on the distributed channel and routing assignment in heterogeneous multi-radio, multi-channel, multi-hop wireless networks. The proposed protocol coordinates the channel and route selection at each node, based on the information exchanged among two-hop neighbor nodes. However, these studies are not suitable for cognitive radio networks, since they ignore the dynamic nature of spectrum opportunities and users (network nodes) need to estimate the behavior of the primary users for coexistence. To the best of our knowledge, the dynamic resource management problem in multi-hop cognitive radio networks has not been addressed in literature.

In summary, the chapter makes the following contributions.

- a) We propose a dynamic resource management scheme in multi-hop cognitive radio network settings based on periodic information exchange among network nodes. Our approach allows each network nodes (secondary users and relays) to exchange their spectrum opportunity information and select the optimal channel and next relay to transmit delay sensitive packets.
- b) We investigate the impact of the information exchange collected from various hops on the performance of the distributed resource management scheme. We introduce the notion of an “information cell” to explicitly identify the network nodes that can convey timely information. Importantly, we investigate the case that the information cell does not cover all the interfering neighbor nodes in the interference graph.
- c) The proposed dynamic resource management algorithm applies FP, which allows various nodes to learn their spectrum opportunity from the information exchange and adapt their transmission strategies autonomously, in a distributed manner. Moreover, we discuss the tradeoffs between the cost of the required information exchange and the learning efficiency of the multi-agent learning approach in terms of the utility impact.

Next, we present our network settings of the multi-hop cognitive radio networks.

III. MULTI-HOP COGNITIVE RADIO NETWORK SETTINGS

A. Network entities

In this chapter, we assume that a multi-hop cognitive radio network involves the following network entities and their interactions:

- **Primary Users (PUs)** are the incumbent devices that possess transmission licenses for specific frequency bands (channels). Without loss of generality, we assume that there are M frequency channels in the considered cognitive radio network. We also assume that the maximum number of primary users that can be present in the network equals M . Note that these primary users can only occupy their assigned (licensed) frequency channels and not other primary users' channels. Since the primary users are licensed users, they will be guaranteed an interference-free environment [Hay05][ALV06]. When a primary user is not transmitting data using its assigned frequency channel, a spectrum hole is formed at the corresponding frequency channel.
- **Secondary Users (SUs)** are the autonomous wireless stations that perform channel sensing and access the existing spectrum holes in order to transmit their data. The secondary users can occupy the spectrum holes available in the various frequency channels. In this chapter, the secondary users are deploying delay sensitive applications. Specifically, we assume that there are V delay sensitive applications simultaneously sharing the cognitive radio network infrastructure, having unique source and destination nodes. These secondary users are able to deploy their applications across various frequency channels and routes.
- **Network Relays (NRs)** are autonomous wireless nodes that perform channel sensing and access the existing spectrum holes in order to relay the received data to one of its neighboring nodes or SUs. Hence, unlike in the SUs case, there is no source or

destination present at the NRs. Note that multiple applications can use the same NR using different frequency channels.

B. Source traffic characteristics

Let V_i denote the delay sensitive application of the i -th SU. Assume that the application V_i consists of packets in K_i priority classes. The total number of applications is V . We assume that there are a total of $K = \sum_{i=1}^V K_i + 1$ priority classes (i.e., $C = \{C_1, \dots, C_K\}$). The reason for adding an additional priority class is because the highest priority class C_1 is reserved for the traffic of the primary users. The rest of the classes $C_k, k > 1$ can be characterized by:

- λ_k , the impact factor of a class C_k . For example, this factor can be obtained based on the money paid by a user (different service levels can be assigned for different SUs by the cognitive radio network), based on the distortion impact experienced by the application of each SU or based on the tolerated delay assigned by the applications. The classes of the delay sensitive applications are then prioritized based on this impact factor, such that $\lambda_k \geq \lambda_{k'}$ if $k < k', k = 2, \dots, K$. The impact factor is encapsulated in the header (e.g. RTP header) of each packet.
- D_k , the delay deadline of the packets in a class C_k . In this chapter, a packet is regarded useful for the delay sensitive applications only when it is received before its delay deadline.
- L_k , the average packet length in the class C_k .

A variety of delay sensitive applications can use the cognitive radio set-up discussed in this chapter. Multimedia transmission such as video streaming or video conferencing can be examples of such applications as discussed in the first three chapters. We assume in this chapter that an application layer scheduler is implemented at each network node to send the most important packet first based on the impact factor encapsulated in the packet header.

C. Multi-hop cognitive radio network specification

We consider a multi-hop cognitive radio network, which is characterized by a general topology graph $\mathcal{G}(\mathbf{M}, \mathbf{N}, \mathbf{E})$ that has a set of primary users $\mathbf{M} = \{m_1, \dots, m_M\}$, a set of network nodes $\mathbf{N} = \{n_1, \dots, n_N\}$ (include SUs and NRs) and a set of network edges (links) $\mathbf{E} = \{e_1, \dots, e_L\}$ (connecting the SUs and NRs). There are a total of N nodes and L links in this network. Each of these N network nodes is either a secondary user (as a source or a destination node) or a network relay.

We assume that $\mathbf{F} = \{f_1, \dots, f_M\}$ is the set of frequency channels in the network, where M is the total number of the frequency channels. To avoid interference to the primary users, the network nodes can only use spectrum holes for transmission. Hence, to establish a link with its neighbor nodes, each network node $n \in \mathbf{N}$ can only use the available frequency channels in a set $\mathbf{F}_n \subseteq \mathbf{F}$. Note that these wireless nodes in a cognitive radio network will continuously sense the environment and exchange information and hence, $\mathbf{F}_n$ may change over time depending on whether the primary users are transmitting in their assigned frequency channels.

The network resource for a network node $n \in \mathbf{N}$ of the multi-hop cognitive radio network includes the routes composed by the various links and frequency channels. We define the resource matrix $\mathbf{R}_n = [R_{ij}] \in \{0, 1\}^{L \times M}$ for the network node n as follows:

$$R_{ij} = \begin{cases} 1, & \text{if link } e_i \text{ is connected to the node } n \\ & \text{and the frequency channel } f_j \text{ is available.} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Whether or not the resource R_{ij} is available to node $n \in \mathbf{N}$ depends not only on the topology connectivity, but also on the interference from other traffic using the same frequency channel. Next, we discuss the interference from other users (including the primary users).

D. Interference characterization

Recall that the highest priority class C_1 is always reserved in each frequency channel

for the traffic of the primary users. The traffic of the SUs can be categorized into $K - 1$ priority classes ($C_2, \dots, C_K$) for accessing frequency channels. The traffic priority determines its ability of accessing the frequency channel. Primary users in the highest priority class C_1 can always access their corresponding channels at any time. The traffic of the SUs can only access the spectrum holes for transmission. Hence, we define two types of interference to the secondary users in the considered multi-hop cognitive radio network:

1) *Interference from primary users.*

In practical cognitive networks, even though primary users have the highest priority, secondary users will cause some level of interference to the primary users due to their imperfect awareness (sensing) of the primary users. The primary users' interference depends on the location of the M primary users. We rely on methods such as in [Bro05] that consider the power and location of the secondary users to ensure that the secondary users do not exceed some critical interference level to the primary users. We also assume that the spectrum opportunity map is available to the secondary users as in [CCB06][HPR07]. Since the primary users will block all the neighbor links using its frequency channel, a network node n will sense the channel and obtain the Spectrum Opportunity Matrix (SOM) of the primary users:

$$\mathbf{Z}_n = [Z_{ij}] \in \{0,1\}^{L \times M}, \text{ with } Z_{ij} = \begin{cases} 1, & \text{if the primary user is occupying frequency channel } f_j \\ & \text{and the link } e_i \text{ can interfere with the primary user.} \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

A simple example is illustrated in Figure 7.1, which indicates the SOM of the primary users and the resource matrix of each network node in the multi-hop cognitive radio network.

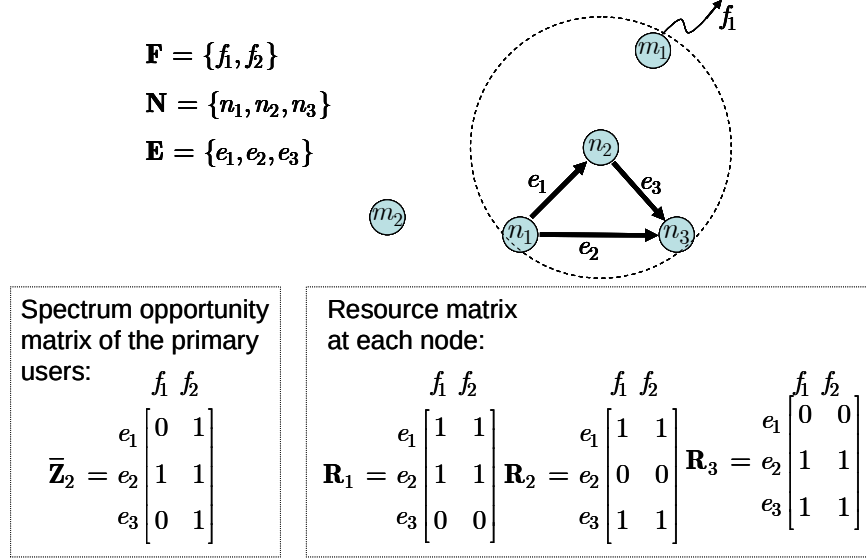


Fig. 7.1. A simple multi-hop cognitive radio network with three nodes and two frequency channels.

2) Interference from competing secondary users.

We define $\mathbf{I}_k = [I_{ij}] \in \{0,1\}^{L \times M}$ as the Interference Matrix (IM) for the traffic in priority class $C_k, k \geq 2$.

$$I_{ij} = \begin{cases} 1, & \text{if link } e_i \text{ using frequency channel } f_j \text{ can be} \\ & \text{interfered by the traffic of priority class } C_k. \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

The interference caused by the traffic in priority class C_k can be determined based on the interference graph of the nodes that transmit the traffic (as in [HPR07]). The interference graph is defined as the corresponding links that are interfered by the transmission of the class C_k traffic¹. The IM can be computed by the information exchange among the neighbor nodes.

The available resource matrix can be masked out by the SOM and IM of the higher priority classes, i.e. $\mathbf{R}_{nk}^{(I)} = \mathbf{R}_n \otimes \bar{\mathbf{I}}_{k-1} \otimes \dots \otimes \bar{\mathbf{Z}}_n$, where the notation $\otimes$ represents

¹ In a wireless environment, the transmission of *neighbor links* can interfere with each other and significantly impact their effective transmission time. Hence, the action of a node can impact and be impact by the action of the other relay nodes. In order to coordinate these neighboring nodes, we construct the interference matrix with binary “1” and “0”.

element-wise multiplication of the matrixes and $\bar{\cdot}$ denotes the inverse operation, which turns 1 into 0 and 0 into 1. The resulting resource matrix $\mathbf{R}_{nk}^{(I)}$ represents the *available resource* around the network node n for the class C_k traffic under the interference of other higher priority traffic (classes). Next, we define the actions available to the network nodes in a multi-hop cognitive radio network.

E. Actions of the nodes

We define the action of the network node n in order to relay the delay sensitive application V_i as $A_n = (e \in \mathbf{E}_n, f \in \mathbf{F}_n)$. We assume that a network relay n can select a set of links to its neighbor nodes (links connected to node n) $\mathbf{E}_n \subseteq \mathbf{E}$. Corresponding to the actions, we define the transmission strategy vector of the network node n as $\mathbf{s}_n = [s_A \mid A = (e \in \mathbf{E}_n, f \in \mathbf{F}_n)]$, where s_A represent the probability that the network node n will choose an action A . We refer to an action at a node n as a “feasible action” for transmitting a class C_k traffic, if $A = (e, f)$ is an “available resource” in $\mathbf{R}_{nk}^{(I)}$ (i.e. element $R_{ef} = 1$ in $\mathbf{R}_{nk}^{(I)}$), since in this case the selected link and frequency channel do not interfere with the traffic in the higher priority classes. That is,

$$\hat{\mathbf{A}}_n(k) = \{A = (e, f) \mid \mathbf{R}_{nk}^{(I)} = [R_{ef}]^{L \times M}, R_{ef} = 1\}. \quad (4)$$

We denote the set of all the feasible actions for node n as $\hat{\mathbf{A}}_n(k)$ for class C_k traffic. We next determine the corresponding delay based on different actions, which considers the deployed cross-layer transmission strategies in order to compute the Effective Transmission Time (ETT) [DPZ04] over the transmission links.

Each network node n computes the ETT $ETT_{nk}(e, f)$, with $e \in \mathbf{E}_n, f \in \mathbf{F}_n$ for transmitting delay sensitive applications in priority class C_k :

$$ETT_{nk}(e, f) = \frac{L_k}{T_n(e, f) \times (1 - p_n(e, f))}. \quad (5)$$

$T_n(e, f)$ and $p_n(e, f)$ represent the transmission rate and the packet error rate of the network node n using the frequency channel f over the link e . $T_n(e, f)$ and $p_n(e, f)$ can be estimated by the MAC/PHY layer link adaptation [Kri02]. Specifically, we assume that the channel condition of each link-frequency channel pair can be modeled

using a continuous-time Markov chain [BG87] with a finite number of states $\mathbf{S}_{(e,f)}^n$. The time a channel condition spends in state $i \in \mathbf{S}_{(e,f)}^n$ is exponentially distributed with parameter ν_i (rate of transition at state i in transitions/sec). We assume that the maximum transition rate² of the network is ν and the variation of the channel conditions in a time interval $\tau \leq 1/\nu$ is regarded negligible.

Define the action vector $\mathbf{A}_i = [A_n \mid n \in \sigma_i]$ as the vector of the actions of all the network relay nodes for transmitting V_i . Assume that the i th delay sensitive application V_i are transmitted from the source node $n_i^s \in \mathbf{N}$ to the destination node $n_i^d \in \mathbf{N}$ with a total of q_i packets. The routes of V_i are denoted as $\sigma_i = \{\sigma_{ij} \mid j = 1, \dots, q_i\}$, where σ_{ij} is the route of the j th packet in V_i . A route σ_{ij} is a set of link-frequency pairs that the packets flow through, i.e.

$$\sigma_{ij} = \{(e, f) \mid \text{the } j\text{th packet of } V_i \text{ flows through link } e \text{ using frequency channel } f\}. \quad (6)$$

Note that if the action of a certain relay node changes, the corresponding route $\sigma_{ij}(\mathbf{A}_i)$ of relaying V_i also changes. We denote the end-to-end delay of the packets transmitted using the route $\sigma_{ij}(\mathbf{A}_i)$ as $d_{ij}(\sigma_{ij}(\mathbf{A}_i))$. Based on the topology, each network relay node receiving a packet can decide where to relay the packet to and using which frequency channel, in order to minimize its end-to-end delay $d_{ij}(\sigma_{ij}(\mathbf{A}_i))$. Finally, to calculate $d_{ij}(\sigma_{ij}(\mathbf{A}_i))$, the source node need to obtain the delay information from other nodes according to the actions taken by the relay nodes, i.e.

$$d_{ij}(\sigma_{ij}(\mathbf{A}_i)) = \sum_{n \in \sigma_{ij}} ETT_{nk}(\mathbf{A}_i), \text{ for } V_i \in C_k. \quad (7)$$

IV. RESOURCE MANAGEMENT PROBLEM FORMULATION

By examining the cumulated ETT values, the objective of a delay sensitive application is to minimize its own end-to-end packet delay. The centralized and proposed distributed

² In case that some of the channel conditions change severely in the network, a threshold ν_{th} can be set by protocols to avoid these fast-changing nodes and the ν is hence selected as the maximum transition rate below this threshold value.

problem formulations are subsequently provided.

- **Centralized problem formulation with global information available at the sources**

If we assume that the global information³ $\mathcal{G}_i$ is available to the source node n_i^s for the delay sensitive application V_i , the route $\sigma_{ij}(\mathbf{A}_i, \mathcal{G}_i)$ can be determined for each packet j of V_i . The centralized optimization can be performed at every source node in order to maximize the utility u_i . Hence, for application V_i we have:

$$\begin{aligned} \mathbf{A}_i^{opt} &= \arg \max u_i(\mathbf{A}_i, \mathcal{G}_i) \\ &\text{subject to } A \in \hat{\mathbf{A}}_n \text{ for all } A \in \mathbf{A}_i, \end{aligned} \quad (8)$$

$$\begin{aligned} \text{where } u_i(\mathbf{A}_i, \mathcal{G}_i) &= \sum_{j=1}^{q_i} \lambda_{ij} \cdot \text{Prob}\{d_{ij}(\sigma_{ij}(\mathbf{A}_i, \mathcal{G}_i)) \leq D_{ij}\}, \\ D_{ij} &= D_k \text{ and } \lambda_{ij} = \lambda_k \text{ if } j \in C_k. \end{aligned} \quad (9)$$

However, due to the limited wireless network resource, the end-to-end delay constraint $d_{ij}(\sigma_{ij}(\mathbf{A}_i, \mathcal{G}_i)) \leq D_k$ can make the optimization solution infeasible. Hence, a sub-optimal greedy algorithms that perform optimizations sequentially from the highest priority class to the lowest priority class are commonly adopted [CF06][SV07b]. Specifically, for class C_k , the following optimization is considered:

$$\begin{aligned} \mathbf{A}_{ik}^{opt} &= \arg \min \sum_{j \in C_k} d_{ij}(\sigma_{ij}(\mathbf{A}_{ik}, \mathcal{G}_i)) \\ &\text{subject to } d_{ij}(\sigma_{ij}(\mathbf{A}_{ik}, \mathcal{G}_i)) \leq D_k, \quad , \\ &A \in \hat{\mathbf{A}}_n \text{ for all } A \in \mathbf{A}_{ik}. \end{aligned} \quad (10)$$

where $\mathbf{A}_{ik} = [A_n \mid n \in \sigma_{ij}, j \in C_k]$.

Due to the informationally decentralized nature of the multi-hop wireless networks, the centralized solution is not practical for the multi-user delay sensitive applications, as the tolerable delay does not allow propagating the global information $\mathcal{G}_i$ back and forth throughout the network to a centralized decision maker. For instance, the optimal solution depends on the delay d_{ij} incurred by the various packets across the hops, which cannot

³ The word “global information” means the information gathered from every node throughout the network. We discuss the required information in Section V.

be timely relayed to a source node. For instance, when the network environment is time-varying, the gathered global information $\mathcal{G}_i$ can be inaccurate due to the propagation delay for this information. Moreover, the complexity of the centralized optimization grows exponentially with the number of classes and nodes in the network. The problem is further complicated by the dynamic adaptation of the transmission strategies deployed by the wireless nodes, which impacts their spectrum access and hence, implicitly, the performance of their neighbor nodes. The optimization will require a large amount of time to process and the collected information might no longer be accurate by the time transmission decisions need to be made.

In summary, in the studied dynamic cognitive radio network, the decisions on how to adapt the aforementioned actions at sources and relays need to be performed in a distributed manner due to these informational constraints. Hence, a “decomposition” of the optimization problem into distributed strategic adaptation based on the available local information is necessary.

- **Proposed distributed problem formulation with local information at each node:**

Instead of gathering the entire global information $\mathcal{G}_i$ at each source, we propose a distributed suboptimal solution that collects the local information $\mathcal{L}_n$ at node n to minimize the expected delay of the various applications sharing the same multi-hop wireless infrastructure. Note that at each node n , the end-to-end delay for sending a packet $j \in C_k$ in equation (10) can be decomposed as:

$$d_{ij}(\sigma_{ij}) = d_n^P(\sigma_{ij}) + E[\tilde{d}_n(k, \sigma_{ij})], \quad (11)$$

where $d_n^P(\sigma_{ij})$ represents the past delay that packet j has experienced before it arrives at node n and $E[\tilde{d}_n(k, \sigma_{ij})]$ represents the expected delay from the node n to the destination of the packet $j \in C_k$. The sending packet $j \in C_k$ is determined by the application layer scheduler according to the impact factor λ_k . The information about λ_k can be encapsulated in the packet header and $d_n^P(\sigma_{ij})$ can be calculated based on the

timestamp available in the packet header. The priority scheduler at each node ensures that the higher priority classes are not influenced by the lower priority classes (see equation (10)). Since at the node n the value of $d_n^P(\sigma_{ij})$ is fixed, the optimization problem at the node n becomes:

$$\begin{aligned} A_n^{opt} &= \arg \min E[\tilde{d}_n(k, \sigma_{ij}(A_n, \mathcal{L}_n))] \\ \text{subject to } &E[\tilde{d}_n(k, \sigma_{ij}(A_n, \mathcal{L}_n))] \leq D_k - d_n^P(\sigma_{ij}) - \rho, j \in C_k, \\ &A_n \in \hat{\mathbf{A}}_n \end{aligned} \quad (12)$$

where $E[\tilde{d}_n(k, \sigma_{ij}(A_n, \mathcal{L}_n))]$ represents the expected delay from the relay node n to the destination of the packets in class C_k . ρ represents a guard interval such that the probability $\text{Prob}\{E[\tilde{d}_n(k, \sigma_{ij}(A_n, \mathcal{L}_n))] + d_n^P(\sigma_{ij}) > D_k\}$ is small (as in [JF06]). To estimate the expected delay $E[\tilde{d}_n(k, \sigma_{ij}(A_n, \mathcal{L}_n))]$ in equation (12), each network node n maintains an estimated transmission delay $E[\tilde{d}_n(k)]$ from itself to the destination for each class of traffic using the Bellman-Ford shortest-delay routing algorithm [BG87]. We assume that each node n maintains and updates a delay vector $\mathbf{d}_n = [E[\tilde{d}_n(2)], \dots, E[\tilde{d}_n(K)]]$ (note that the first priority class is reserved for the primary users) with elements for each priority class. Each network node exchanges such information to its neighbor nodes and selects the best action A_n^{opt} for the highest priority packet in the buffer of the network node n . We will discuss the minimum-delay routing/channel selecting algorithm in Section VI. Note that a group of packets in the buffer of a node n can take the action A_n , since the action is determined based on local information $\mathcal{L}_n$. Since in the cognitive radio networks, the available channel is time-variant, the information needs to be timely conveyed to the network node for the distributed optimization. Compared to the centralized approach in equation (8), the distributed resource management in equation (12) can adapt better to the dynamic wireless environment by periodically gathering local information. Next, we discuss the distributed resource management with information constraints in more detail.

V. DISTRIBUTED RESOURCE MANAGEMENT WITH INFORMATION CONSTRAINTS

A. Considered medium access control

In this chapter, we assume that the required local information $\mathcal{L}_n$ is exchanged using a designated coordination control channel similar to [BRB05]. Such a coordination channel can be selected from the existing ISM bands, since there is no primary licensee in these bands to interfere with. The transmission is time slotted and the time slot structure of a node is provided in Figure 7.2. We denote the time slot duration as t_I . The action A_n are selected at each node, during each time slot, after the coordination interval (that includes the channel sensing for SOM and the information exchange for IM). We denote the coordination interval at the network node n as $d_I(\mathcal{L}_n)$. The goal of the coordination interval at each time slot is to provide the feasible action set $\hat{A}_n$ for the channel access and the relay selection of the packet transmission. We will discuss how to obtain $\hat{A}_n$ based on the SOM and the IM among the neighboring nodes when we introduce the proposed algorithm, in Section VI.

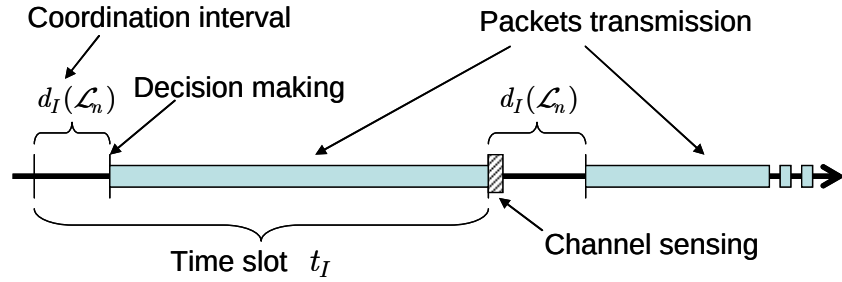


Fig. 7.2. Transmission time line at the node n with local information $\mathcal{L}_n$.

Besides the SOM and IM, the information required in the coordination interval should also include the delay vectors $\mathbf{d}_n$ and the control messages for RTS/CTS coordination [ZTS07][WYT06]. Note that the local information $\mathcal{L}_n$ does not need to include all these information in each time slot (except the control messages). For example, the SOM and IM can be collected in a different period, depending on the sensing and information exchange mechanism. Hence, the coordination duration $d_I(\mathcal{L}_n)$ will vary for different

time slots, which will be discussed in more detail in Section V.C. Next, we investigate the benefit of acquiring information from different h -hop neighbor nodes, which also affects the duration of the coordination interval $d_I(\mathcal{L}_n)$.

B. Benefit of acquiring information and information constraints

For the network node n , the local information $\mathcal{L}_n$ gathered from different network nodes has different impact on decreasing the objective function $E[\tilde{d}_n(k, \sigma_{ij}(A_n, \mathcal{L}_n))]$ in equation (12). Let $\mathcal{Z}_n(x) = \{\mathbf{I}_k(n_x, A_{n_x}), A_{n_x}, \mathbf{d}_{n_x} \mid n_x \in \mathbf{N}_x^n\}$ denote the set of local information gathered from the neighbor nodes, which is x hops away from node n , where $\mathbf{N}_x^n$ represents a set of nodes that is x hops away from node n . We define $\mathcal{L}_n(x) = \{\mathcal{Z}_n(l) \mid l = 1, \dots, x\}$ as the local information gathered from all of these neighbor nodes. Given the local information $\mathcal{L}_n(x)$, we define the optimal expected delay as $K_n(k, x) = E[\tilde{d}_n(k, \sigma_{ij}(A_n^{opt}, \mathcal{L}_n(x)))]$. The larger x will have a smaller expected delay $K_n(k, x)$. The benefit (reward) of the information $\mathcal{Z}_n(x)$ for the class C_k traffic is denoted as $J_n(k, \mathcal{Z}_n(x))$. In a static network case, $J_n(k, \mathcal{Z}_n(x))$ is defined as:

$$J_n(k, \mathcal{Z}_n(x)) \triangleq K_n(k, x-1) - K_n(k, x), \text{ if } x > 1. \quad (13)$$

We define $J_n(k, \mathcal{Z}_n(1)) = K_n(k, 1)$ since $\mathcal{L}_n(1) = \mathcal{Z}_n(1)$. The reward of information $J_n(k, \mathcal{Z}_n(x))$ can be regarded as the benefit (decrease of the expected delay) in terms of the expected delay $E[\tilde{d}_n(k, \sigma_{ij})]$ if the information $\mathcal{Z}_n(x)$ is received by node n . Note that the optimal expected delay $K_n(k, x)$, given the information $\mathcal{L}_n(x)$:

$$K_n(k, x) = K_n(k, 1) - \sum_{l=2}^x J_n(k, \mathcal{Z}_n(l)). \quad (14)$$

Equation (14) states that the optimal expected delay is a decreasing function of x , meaning that smaller expected delays can be achieved as more information is gathered. The improvement is quantified by the reward of the information $J_n(k, \mathcal{Z}_n(l))$. Here, we ignore the cost of exchanging such information, which will be defined in the next subsection. Figure 7.3 shows a simple illustrative example of reward of information at node n , which is five hops away from the destination node of class C_k traffic. The more information $\mathcal{Z}_n(x)$ available from nodes that is x hops away, the smaller optimal

expected delay $K_n(k, x)$ can be obtained.

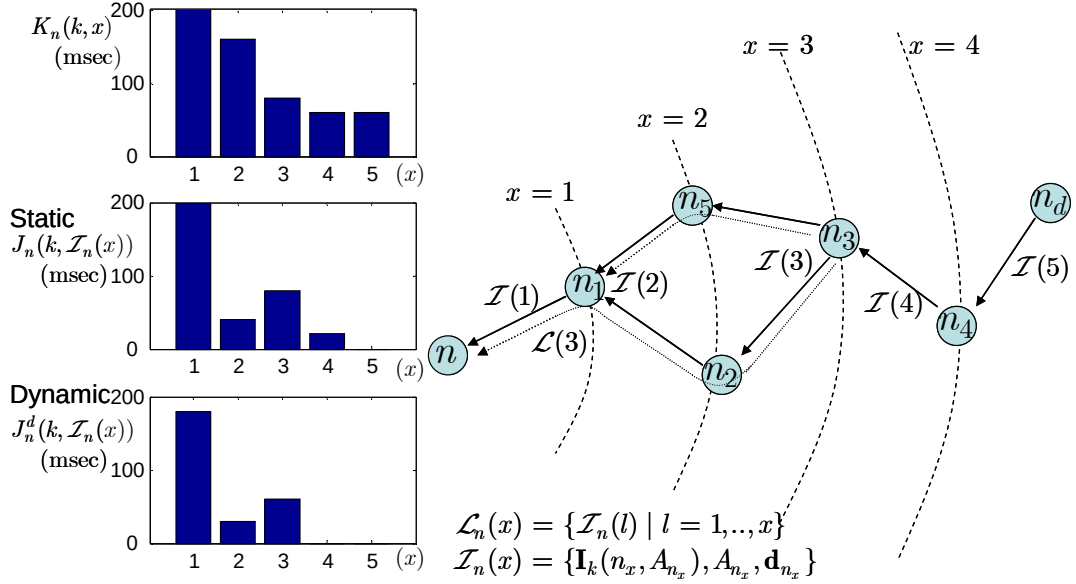


Fig. 7.3. Example of the static reward of information $J_n(k, \mathcal{I}_n(x))$, dynamic reward of information $J_n^d(k, \mathcal{I}_n(x))$ and optimal expected delay $K_n(k, x)$ (where the information horizon $h_n(k, \nu) = 3$, average packet length $L_k = 1000$ bytes, and average transmission rate $T = 6$ Mbps over the multi-hop network).

Let $\mathbf{J}_n(k) = [J_n(k, \mathcal{I}_n(x))]$, for $1 \leq x \leq H_n$ denote the reward vector from 1-hop information to H_n -hop information, where $H_n = \max\{H_n^I, H_n^d\}$. H_n^d represents the shortest hop counts from the node n to the destination node of the class C_k traffic and H_n^I represents the interference range in terms of hop counts for node n . We also need to consider the hop count H_n^I in case that the destination node is close to the node n within the interference range. We assume that the reward vector $\mathbf{J}_n(k)$ is obtained when the network is first deployed and only updated infrequently, when SUs join or leave the network. Note that all the elements in $\mathbf{J}_n(k)$ are nonnegative, i.e. $J_n(k, \mathcal{I}_n(x)) \geq 0$, for $1 \leq x \leq H_n$, due to the fact that knowing additional information cannot increase the expected delay $E[\tilde{d}_n(k, \sigma_{ij})]$ in a static network. However, if we consider the propagation delay of such information exchange across the network in the dynamic network, the dynamic reward of information $J_n^d(k, \mathcal{I}_n(x))$ decreases as the hop

count x increases. When the information of the further nodes reaches the decision node n , the information is more likely to be out-of-date (i.e. the information cannot reflect the exact network situation in a dynamic setting, since the network conditions and traffic characteristics are time-varying). Once the information is out-of-date, $J_n^d(k, \mathcal{Z}_n(x)) = 0$, i.e. there is no benefit from gathering information that is out-of-date. Note that in a dynamic network, once $J_n^d(k, \mathcal{Z}_n(x)) = 0$, $J_n^d(k, \mathcal{Z}_n(x')) = 0$ for $x \leq x' \leq H_n$.

Therefore, in the dynamic network, we define the information horizon $h(k, \nu)$ such that

$$h_n(k, \nu) \triangleq \arg \max x \quad \text{subject to } J_n^d(k, \mathcal{Z}_n(x)) > \phi(k, \nu), 1 \leq x \leq H_n. \quad (15)$$

where $\phi(k, \nu) \geq 0$ represents a minimum delay variation specified by the application which determines the minimum benefit of receiving local information for class C_k traffic. In fact, $h_n(k, \nu)$ depends on the variation speed ν of the wireless network condition (i.e. the transition rate of the Markovian channel condition model, see Section III.E). In a dynamic network with higher variation speeds ν (e.g. with high mobility), a higher threshold $\phi(k, \nu)$ is needed to guarantee that the information $\mathcal{Z}_n(x)$ is still valuable and it should be exchanged. This results in a smaller information horizon $h_n(k, \nu)$. We illustrate this mobility issue in Section VII. Note that the information horizon $h_n(k, \nu)$ varies for different classes of traffic at different locations in the network. Since higher priority class traffic has more network resources than the lower priority class (i.e. they are scheduled first for optimization in equation (12)), the threshold value $\phi(k, \nu) \leq \phi(k', \nu)$, if $k < k'$ and thereby, $h_n(k, \nu) \geq h_n(k', \nu)$, if $k < k'$. In other words, the information horizon $h_n(k, \nu)$ of a higher priority class C_k is larger than the information horizon $h_n(k', \nu)$ of a lower priority class $C_{k'}$.

Although the information horizon $h_n(k, \nu)$ can vary at different locations for different priority classes depending on the applications, the complexity of such implementation is high and the adaptation of the information horizon itself can be an interesting topic. Hence, we will leave the information horizon adaptation problem to our future research.

For simplicity, we assume in this chapter that the information horizon is only a function of the network variation speed ν , i.e. $h_n(k, \nu) = h(\nu)$. The information horizon $h(\nu)$ is determined for the most important class among the SUs in the network. This definition of the information horizon $h(\nu)$ is the same as in Chapter 3, in which $h(\nu)$ is defined as the maximum number of hops that the information can be conveyed in τ , such that the network is considered unchanged (recall that any network changes within the interval $\tau(\nu) \leq 1/\nu$ can be regarded negligible).

Based on this information horizon $h(\nu)$, we assume that the network nodes within the $h(\nu)$ hops form an *information cell*. Only the local information $\mathcal{L}_n(h)$ within the information cell is useful to the node n , since the reward of information is zero, i.e. $J_n(k, \mathcal{L}_n(x)) = 0$ for $\forall x > h(\nu)$. In the dynamic network, network node n determines its action at time slot t based on the acquired information at the previous time slot $t - 1$.

The optimization problem in equation (12) can be written as:

$$\begin{aligned} A_n^{opt}(t) &= \arg \min E[\tilde{d}_n(k, \sigma_{ij}(A_n, \mathcal{L}_n(h, t - 1)))] \\ \text{subject to } &E[\tilde{d}_n(k, \sigma_{ij}(A_n, \mathcal{L}_n(h, t - 1)))] \leq D_k - d_n^P(\sigma_{ij}) - \rho, j \in C_k. \\ &A_n \in \hat{\mathbf{A}}_n(t - 1) \end{aligned} \quad (16)$$

Recall that the neighbor nodes of the node n are defined as the nodes that can interfere or can be interfered by the node n (within H_n^I hops), which may not align with the range of the information cell (within $h(\nu)$ hops). If all neighbor nodes are within the h -hop information cell, all necessary information are timely conveyed to the node n . Otherwise, the neighbor nodes that are too far away cannot convey the interference information to the node n in time. Since the required information cannot be acquired in time, the solution in equation (16) becomes suboptimal. We refer to this problem as “information exchange mismatch” problem.

Figure 7.4 illustrates two simple network examples with and without the mismatch problem. Note that in Figure 4(b), since the information cell does not cover all the interfering neighbor nodes, the center node n_2 will still be interfered by other secondary users. In fact, due to the nature of the multi-hop wireless environment, the network nodes

that are far away from the node n have limited interference impact on node n_2 . Hence, even though the information horizon h does not match the interference range, the performance degradation of the optimization problem in equation (16) using the local information $\mathcal{L}_n(h)$ is limited.

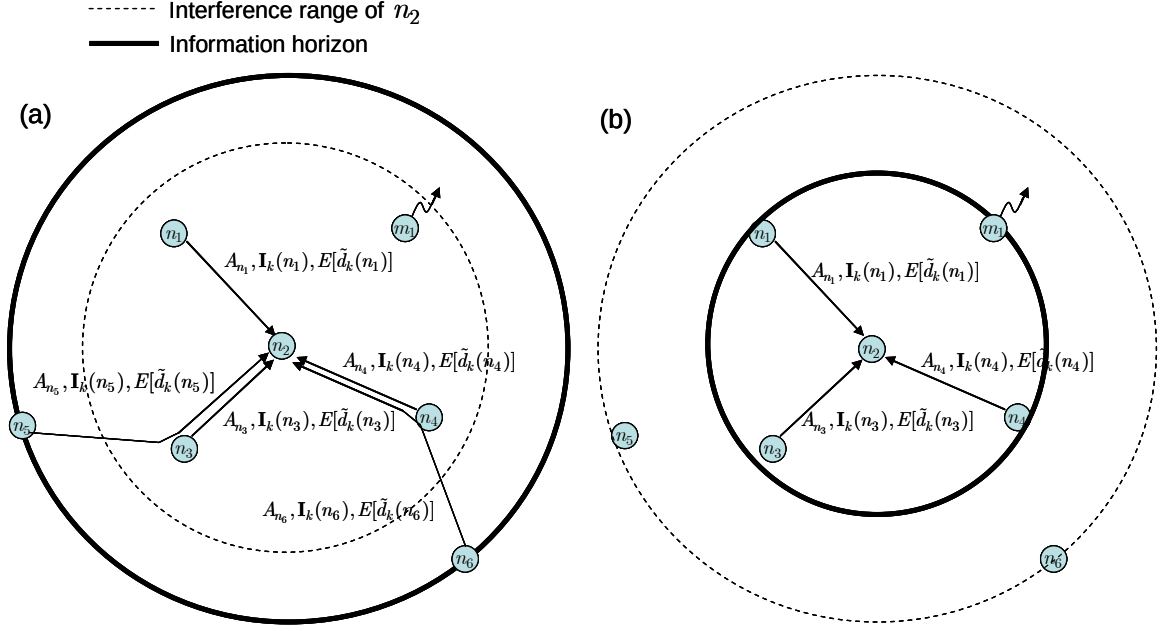


Fig 7.4. (a) 2-hop information cell network without information exchange mismatch problem.
(b) 1-hop information cell network with information exchange mismatch problem.

C. Cost of information exchange

In the previous subsection, we discuss the reward of information in an h -hop information cell while ignoring the negative impact of the information exchange. In this section, we discuss the cost (increase of the expected delay) due to this information exchange. Recall that the duration of the time slot is $t_I(\nu)$, which is also the interval between the repeated information exchanges in the network. We define there are c time slots in τ seconds, i.e.

$$t_I(\nu) = \frac{\tau(\nu)}{c}. \quad (17)$$

c defines the frequency of the decision making as well as the learning process, which will be discussed in detail in Section VI. Note that decisions can be made every t_I and this time slot duration is short enough compared to τ . Hence, the network changes in t_I is also negligible.

Recall that the coordination duration in a time slot for the network node n is $d_I(\mathcal{L}_n(h))$. Assume the information unit for the required information is $U^{(I)}$, $U^{(A)}$, and $U^{(d)}$ per class, respectively. Assume the average number of nodes in an h -hop information cell is $\bar{N}(h)$. The information time overhead of $\mathcal{L}_n(h)$ is on average $d_I(\mathcal{L}_n(h)) = \bar{N}(h)[(K-1)(U^{(d)} + U^{(I)}) + U^{(A)}]$.

Note that even though the information exchange is implemented in a designated coordination channel [BRB05], a network node with a single antenna cannot transmit both the data and the control signals at the same time. This information exchange time overhead decreases the effective transmission rate at node n using the line e and frequency channel f :

$$T'_n(e, f) = \frac{t_I(\nu) - d_I(\mathcal{L}_n(h))}{t_I(\nu)} \times T_n(e, f). \quad (18)$$

Hence, the effective transmission time at a node n using the link e and frequency channel f to transmit a packet in class C_k becomes:

$$ETT'_{nk}(e, f) = \frac{t_I(\nu)}{t_I(\nu) - d_I(\mathcal{L}_n(h))} \times ETT_{nk}(e, f). \quad (19)$$

In conclusion, the increase of the effective transmission time degrades the performance of the delay sensitive applications. The degradation depends on the content of the local information exchange $\mathcal{L}_n(h)$, and the network variation speed ν . Hence, the benefit $J_n^d(k, \mathcal{Z}_n(x))$ in equation (15) will decrease due to this cost of the information. Hence, we denote the value of information with this cost consideration as $J_n^c(k, \mathcal{Z}_n(x))$:

$$\begin{aligned} J_n^c(k, \mathcal{Z}_n(x)) &= K'_n(k, x-1) - K'_n(k, x) \\ &= K_n(k, x-1) \times \frac{t_I(\nu)}{t_I(\nu) - d_I(\mathcal{L}_n(x-1))} - K_n(k, x) \times \frac{t_I(\nu)}{t_I(\nu) - d_I(\mathcal{L}_n(x))}. \end{aligned} \quad (20)$$

And the optimal information horizon $h_n(k, \nu)$ in equation (15) also decreases due to the cost. Next, we discuss the proposed distributed resource management algorithm based

on the information exchanges and learning capabilities to tackle the optimization problem in equation (16).

VI. DISTRIBUTED RESOURCE MANAGEMENT ALGORITHMS

Figure 7.5 provides a system diagram of the proposed distributed resource management. First, a packet $j \in C_k$ is selected from the application scheduler at the node n based on the impact factor λ_k of the packet and an action A_n is taken for that packet. The application layer information including C_k, L_k, D_k is conveyed to the network layer for this action decision. Network conditions $T_n(e, f), p_n(e, f)$ are then conveyed from the MAC/PHY layer for computing the ETT values using equation (5).

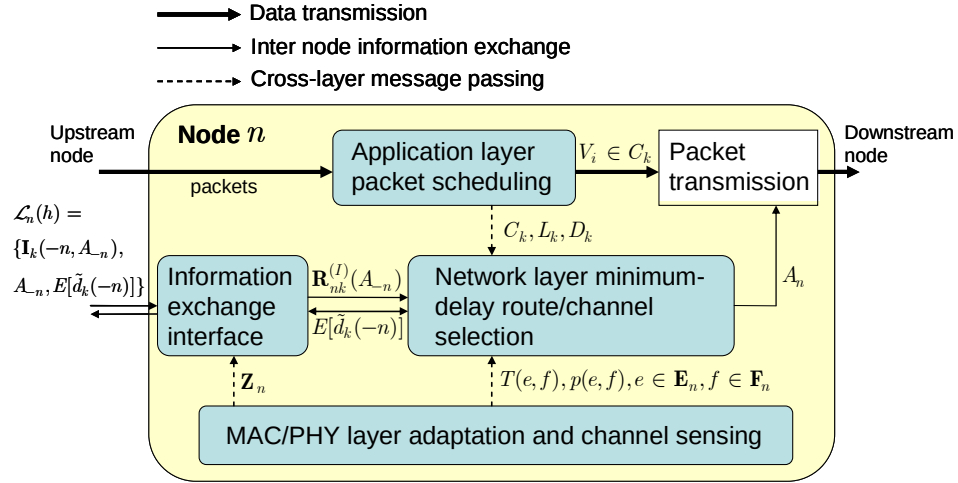


Fig. 7.5. System diagram of the proposed distributed resource management.

In addition to the $T_n(e, f), p_n(e, f)$, the action selection is impacted by the interference induced from the action of these neighbor nodes and hence, the information received from the neighbor nodes in the information cell. Recall that $\mathcal{L}_n(h) = \{\mathcal{Z}_n(l) \mid l = 1, \dots, h\}$. We use the notation $-n(h)$ to represent the set of the neighbor nodes of the network node n in the h -hop information cell. Hence, the local information exchanged $\mathcal{L}_n(h) = \{\mathbf{I}_k(-n(h), A_{-n(h)}), A_{-n(h)}, \mathbf{d}_{-n(h)}\}$ across the network nodes is required. Hence, the

node n knows the estimated delay $\mathbf{d}_{-n(h)}$ from its neighbor nodes to the destinations, so as the actions $A_{-n(h)}$ of its neighbor nodes and their IM $\mathbf{I}_k(-n(h), A_{-n(h)})$. Based on the delay information from the neighbor nodes $\mathbf{d}_{-n(h)}$, a network node can update its own estimated delay to the various destinations and determine the minimum-delay action based on Bellman-Ford algorithm [BG87].

We separate the distributed resource management into two blocks at the node n as in Figure 7.5 – the information exchange interface block that regularly collects required local information and the route/channel selection block to determine the optimal action. We now discuss the role of the exchanged information and the two algorithms implemented in these blocks, respectively.

A. Resource management algorithms

The next algorithm is performed at network node n at the information exchange interface in Figure 7.5.

Algorithm 7.1. Periodic information exchange algorithm:

Step 1. Collect the required information – the node n first collects the required information the SOM $\mathbf{Z}$ from channel sensing and $\mathcal{L}_n(h) = \{\mathbf{I}_k(-n(h), A_{-n(h)}), A_{-n(h)}, \mathbf{d}_{-n(h)}\}$ from the neighbor nodes in the information cell.

Step 2. Learn the behavior of the neighbor nodes – by continuously monitoring the actions of the neighbor nodes, node n can model the behavior of the neighbor nodes or learn a better transmission strategy using strategy vectors $\mathbf{s}(n') = [s_A(n') \mid A = (e \in \mathbf{E}_{n'}, f \in \mathbf{F}_{n'})]$, $n' \in -n(h)$, where $s_A(n')$ represents the probability (strategy) of selecting an action A by the node n' , which will be discussed in the next subsection.

Step 3. Estimate the resource matrix – from the SOM and the IM $\mathbf{I}_k(n', A_{n'})$ gathered from the neighbor node n' , the resource matrix can be obtained for each class of traffic by $\mathbf{R}_{nk}^{(I)} = \mathbf{R}_n \otimes \bar{\mathbf{I}}_{k-1} \otimes \dots \otimes \bar{\mathbf{Z}}_n$, which will be explained in Section VI.A in more details. Then the available resource $\mathbf{R}_{nk}^{(I)}(A_{-n})$ are provided to the network layer route/channel

selection block stated in the Algorithm 7.2.

Step 4. Update information $\{\mathbf{I}_k(n, A_n), A_n, \mathbf{d}_n\}$ – based on the recently selected action A_n , the latest delay vector $\mathbf{d}_n$, and the IM $\mathbf{I}_k(n, A_n)$. Two types of interference model are considered in this chapter when constructing the IM $\mathbf{I}_k(n, A_n)$ from equation (3):

- 1) A network node can transmit and receive packets at the same time – Note that a node cannot reuse a frequency channel $f \in \mathbf{F}_n$ used by its neighbor nodes. If a frequency channel is used by its neighbor nodes, all the elements in the column of the interference $\mathbf{I}_k(n, A_n)$ that is associated with the frequency channel are set to 1. Then the IM is exchanged to the nodes within the pre-determined information horizon h .
- 2) A network node cannot transmit and receive packets at the same time – In this case, if the frequency channel $f \in \mathbf{F}_n$ is used, all the elements in the column of the IM $\mathbf{I}_k(n, A_n)$ associated with the frequency channel are set to 1. In addition, if a network link $e \in \mathbf{E}_n$ is used by its neighbor nodes, all the elements of the IM $\mathbf{I}_k(n, A_n)$ that is associated with the node n are also set to 1, no matter what frequency channel it uses. Then the IM is exchanged to the nodes within the pre-determined information horizon h .

Step 5. Broadcast the information $\{\mathbf{I}_k(n, A_n), A_n, \mathbf{d}_n\}$ **and repeat the algorithm periodically in every $t_I(\nu)$ seconds.**

The next algorithm is performed at the network node n at the network layer minimum-delay route/channel selection block in Figure 7.5.

Algorithm 7.2. Minimum-delay route/channel selection algorithm:

Step 1. Determine the packet to transmit – based on the impact factor, one packet j in the buffer at the node n is scheduled to be transmitted. Assume the packet $j \in C_k$, and the information of C_k , L_k , $D_k - d_n^P$ are extracted or computed from the application layer.

Step 2. Construct the feasible action set – construct the feasible action set $\hat{\mathbf{A}}_n(k)$ from the resource matrix $\mathbf{R}_{nk}^{(I)}$ given from the information exchange interface for the priority

class C_k at the node n (see equation (4)).

Step 3. Estimate the channel condition – the transmission rate $T_n(e, f)$ and packet error rate $p_n(e, f)$ for each link-frequency channel pair $(e \in \mathbf{E}_n, f \in \mathbf{F}_n)$ are provided from the PHY/MAC layer through link adaptation [Kri02].

Step 4. Calculate the expected delay toward the destination – for each action $A_n \in \hat{\mathbf{A}}_n(k)$ of the traffic class C_k :

$$E[\tilde{d}_n(k, A_n)] = ETT_{nk}(A_n) + E[\tilde{d}_{n'(A_n)}(k)], \text{ for } \forall A_n \in \hat{\mathbf{A}}_n(k), \quad (21)$$

where $E[\tilde{d}_{n'(A_n)}(k)]$ represents the corresponding element for the class C_k in the delay vector $\mathbf{d}_{-n}$ from the neighbor node $n'(A_n)$. $ETT_{nk}(A_n)$ can be calculated based on L_k , $T_n(e, f)$, and $p_n(e, f)$ using equation (5).

Step 5. Check the delay deadline – if $E[\tilde{d}_n(k)] \geq D_k - d_n^P - \rho$, drop the packet.

Step 6. Select the minimum delay action – if $E[\tilde{d}_n(k)] < D_k - d_n^P - \rho$, find the minimum-delay route and frequency channel selection, i.e. determine the optimal action A_n^{opt} from the feasible action set $\hat{\mathbf{A}}_n(k)$. In other words, the goal here is to solve equation (16) at node n :

$$A_n^{opt} = \arg \min_{A_n \in \hat{\mathbf{A}}_n(k)} E[\tilde{d}_n(k, A_n)]. \quad (22)$$

Note that the feasible action set $\hat{\mathbf{A}}_n(k)$ in equation (22) depends on the actions of other neighbor nodes A_{-n} . It is important for the network nodes to adopt learning approaches for modeling the behaviors of these network nodes to decrease the complexity of the dynamic adaptation. This will be discussed in the next subsection.

Step 7. Send RTS request – after determining the next relay and frequency channel, send RTS request indicating the determined action information A_n^{opt} to the next relay.

Step 8. Wait for CTS response and transmit the packets.

Step 9. Update the delay and the current action information – after selecting the optimal action, update the estimated delay $E[\tilde{d}_n(k)]$ using exponential moving average with a smoothing factor α :

$$E[\tilde{d}_n(k)] = \alpha \times E[\tilde{d}_n(k)]^{old} + (1 - \alpha) \times E[\tilde{d}_n(k, A_n^{opt})], \quad (23)$$

and provide the updated delay vector $\mathbf{d}_n = [E[\tilde{d}_n(2)], \dots, E[\tilde{d}_n(K)]]$ to Algorithm 7.1 at the information exchange interface. In Figure 7.6, we provide a block diagram of the proposed distributed resource management. For the blocks that beyond the scope of this chapter, we refer to [ALV06][Bro05] for channel sensing, [ZTS07][WYT06] for RTS/CTS coordination, and [BG87] for the delay vectors.

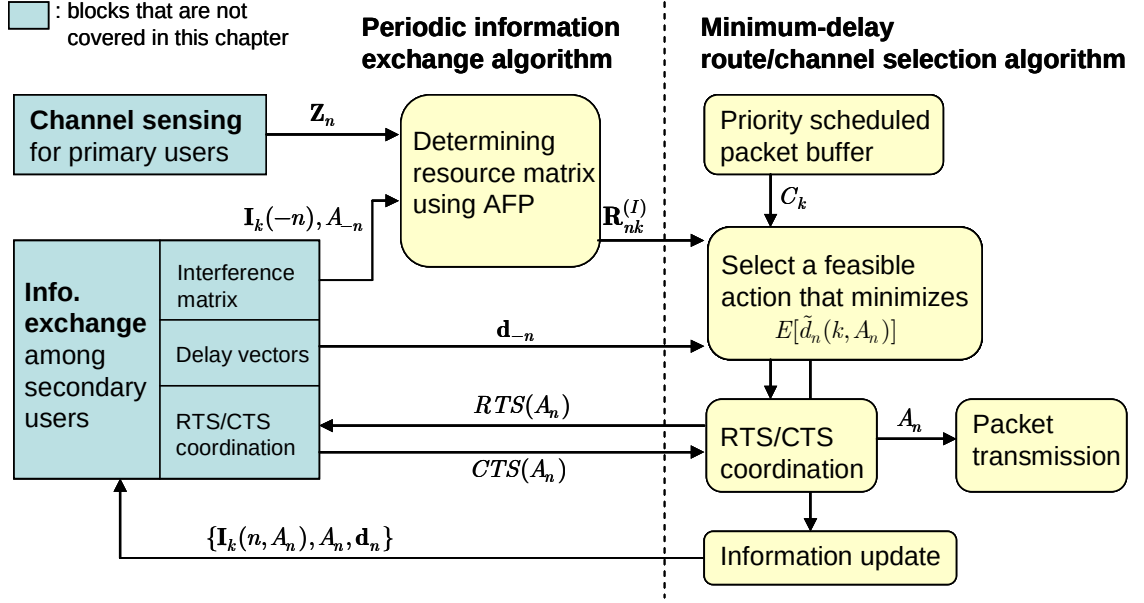


Fig. 7.6. Block diagram of the proposed distributed resource management at network node n .

B. Adaptive fictitious play (AFP)

We now provide a learning approach for the SUs to learn the feasible action set $\hat{\mathbf{A}}_n(k)$ in equation (22) for our distributed resource management algorithms. Specifically, based on the information exchange $\mathcal{L}_n(h)$, the behaviors of the neighbor nodes in the information cell can be learned (Step 2 of Algorithm 7.1) and based on the behaviors, the feasible action set $\hat{\mathbf{A}}_n(k)$ is determined. This motivates us to apply a well-known

learning approach – fictitious play [FL98], applied when the SUs are willing⁴ to reveal their current action information and thereby, they are able to model the behaviors (strategies) of other SUs (a model-based learning [SPG07]). However, due to the information constraint discussed in the previous section, only the information from the neighbor nodes in the information cell is useful. Hence, we adapt the fictitious play learning approach to our considered network setting. Figure 7.7(a) provides a block diagram of the proposed distributed resource management algorithm using the adaptive fictitious play.

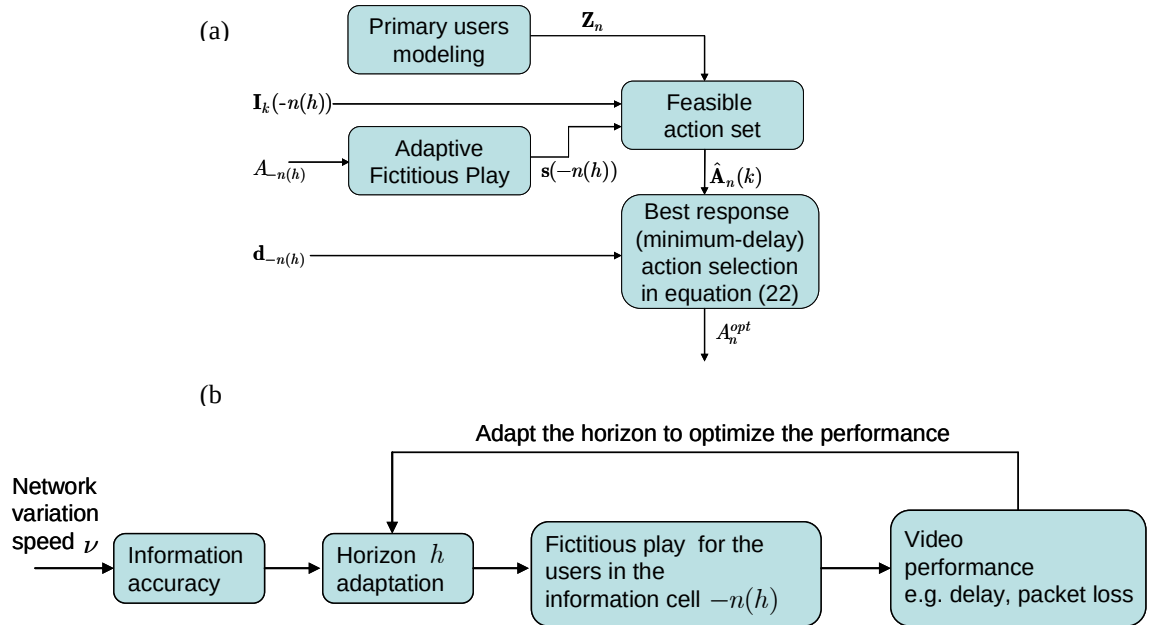


Fig. 7.7 (a). Block diagram of the proposed distributed resource management algorithm using the AFP.
(b). Impact of the network variation on the FP and the video performance.

⁴ If the action information is not provided by the other secondary users, a node can learn its own strategy from its action payoffs – the estimated delay $E[\tilde{d}_n(k)]$. The learning approach refers to the reinforcement learning (a model-free learning or a payoff-based learning).

Note that only part of the SUs can be modeled via the learning approach depending on the information horizon. Specifically, a node n maintains a strategy vector over time $\mathbf{s}(n', t) = [s_A(n', t) \mid A = (e \in \mathbf{E}_{n'}, f \in \mathbf{F}_{n'})]$ for each of its neighbor nodes $n' \in -n(h)$ in the information cell. $s_A(n', t)$ represents the frequency selection strategy of the node n' making action A at time t , which is obtained using:

$$s_A(n', t) = \frac{r_A(n', t)}{\sum_{A \in (\mathbf{E}_{n'}, \mathbf{F}_{n'})} r_A(n', t)}, \quad (24)$$

where $r_A(n', t)$ is the *propensity* [You04] of node n' for taking action A at time t , which can be computed by:

$$r_A(n', t) = \alpha \times r_A(n', t-1) + I(A_{n'}(t) = A), \quad (25)$$

where $\alpha < 1$ is a discount factor quantifying the importance of the history value.

$I(A_{n'}(t) = A)$ represents an indicator function such that,

$$I(A_{n'}(t) = A) = \begin{cases} 1, & \text{if the action of the node } n' \text{ at time } t \text{ is } A \\ 0, & \text{otherwise} \end{cases}. \quad (26)$$

Figure 7.7(b) shows how the network variation speed ν affects the size of the information cell and ultimately, the video performance. We will consider the mobility of the network relays to show this network variation impact in the next section.

As stated in Section III.E, $s_A(n', t)$ represent the probability that the network node n' will choose an action A . Hence, the probability $s_A(n', t)$ for modeling the node n' making an action A at time t will increase with the actual times that the action A is selected. Based on the strategy $s_A(n', t)$, the adaptive fictitious play provides the estimated IM $\mathbf{I}_k$, and then the feasible action set $\hat{\mathbf{A}}_n(k)$ can be computed.

From the gathered IM $\mathbf{I}_k(n', A_{n'})$ from the neighbor node $n' \in -n(h)$, the node n can compute the expected IM from

$$\mathbf{I}_k^e = [I_{ij}^e] = \sum_{n' \in -n(k)} \mathbf{I}_k(n') = \sum_{n' \in -n(k)} \sum_A s_A(n') \mathbf{I}_k(n', A). \quad (27)$$

Then, the node n can estimate the IM $\mathbf{I}_k$ for the traffic in class C_k :

$$\mathbf{I}_k = [I_{ij} \mid I_{ij} = \begin{cases} 1, & \text{if } I_{ij}^e \geq \mu \\ 0, & \text{if } I_{ij}^e < \mu \end{cases}], \quad (28)$$

where μ represents a threshold value that determines whether or not a link-frequency-channel pair (e, f) is considered to be occupied. Feasible action set $\hat{\mathbf{A}}_n(k)$ can hence be learned based on the resource matrix $\mathbf{R}_{nk}^{(I)} = \mathbf{R}_n \otimes \bar{\mathbf{I}}_{k-1} \otimes \dots \otimes \bar{\mathbf{Z}}_n$ using equation (4). By learning the feasible action set $\hat{\mathbf{A}}_n(k)$, the best response actions are computed using equation (22).

C. Information exchange overhead reduction

The fictitious play suffers from a large information overhead, since it requires all the local information $\mathcal{L}_n(h) = \{\mathbf{I}_k(-n(h), A_{-n(h)}), A_{-n(h)}, \mathbf{d}_{-n(h)}\}$ in the h -hop information cell. From the cost of information exchange in equation (20), we know that the overhead can increase the expected delay, especially when the network changes slowly (i.e. with a large information cell). Hence, the overhead reduction is required to mitigate the performance degradation.

(1) Reducing the information horizon.

Recall that the information overhead of $\mathcal{L}_n(h)$ is $\bar{N}(h)[(K-1)(U^{(d)} + U^{(I)}) + U^{(A)}]$ in average ($\bar{N}(h)$ is the average number of nodes in an h -hop information cell). With an information horizon $h' < h$, the overhead becomes $\bar{N}(h')[(K-1)(U^{(d)} + U^{(I)}) + U^{(A)}]$, where $\bar{N}(h') < \bar{N}(h)$. Note that it is not always beneficial to decrease the overhead by reducing the information horizon. There exists a trade-off as discussed in Section V. The reward of information $J_n^d(k, \mathcal{Z}_n(x))$, $x < h$ in equation (15) provides a metric to select the most valuable information from the nodes within the information cell.

(2) Reducing the number of classes.

From equation (12), we know that the higher priority classes will not be influenced by the lower priority classes. Hence, the information overhead can be reduced by ignoring the information exchange of the lower priority classes. The overhead becomes $\bar{N}(h)[(k'-1)(U^{(d)} + U^{(I)}) + U^{(A)}]$, $k' < K$.

(3) Reducing the frequency of learning.

Although we divide c time slots in τ seconds, a network node n does not have to

learn in all these c time slots. In other words, the periodic learning process of the node n does not have to be aligned with the information exchange (decision making). In order to avoid simultaneous learning among network neighbors in a distributed manner, at each time slot, the network node n updates the strategy vector $s_A(n', t)$ with probability $\varepsilon_n = b_n / c$ ($b_n \leq c$), and keeps the same strategy vector with probability $1 - \varepsilon_n$. In other words, the network node n chooses b_n time slots out of c time slots in τ seconds to model the behavior of other neighbor nodes. Note that the parameter b_n characterize the speed of learning at different network node n . The larger b_n gives the network node n faster learning capability. The information overhead of $\mathcal{L}_n(h)$ becomes $b_n / c \times \bar{N}(h)[(K - 1)(U^{(d)} + U^{(I)}) + U^{(A)}]$.

VII. SIMULATION RESULTS

We simulate two video streaming applications that are transmitting videos V_1 “Coastguard” and V_2 “Mobile” (16 frames per GOP, frame rate of 30Hz, CIF format) over the same multi-hop cognitive radio network. Each video sequence is divided into four priority classes ($K_i = 4, K = 9$) with average packet length $L_k = 1000$ bytes and delay deadline $D_k = 500$ millisecond. Although the first priority class C_1 is reserved for the primary users, let us first consider the case when there are no primary users, i.e. only the SUs and NRs are transmitting. We assume that there are two frequency channels ($M=2$). The wireless network topology is shown in Figure 7.8 in a 100x100 meters region with $N = 15$ nodes and $L = 22$ links similar to the network settings in [KSV06]. A link is established as long as the channel condition (described in the chapter by the link SINR) is acceptable within the transmission distance (approximately 36 meters). Note that this transmission distance is not aligned with the interference range H_n^I . Neighbor nodes that are beyond the transmission distance can still interfere with each other.

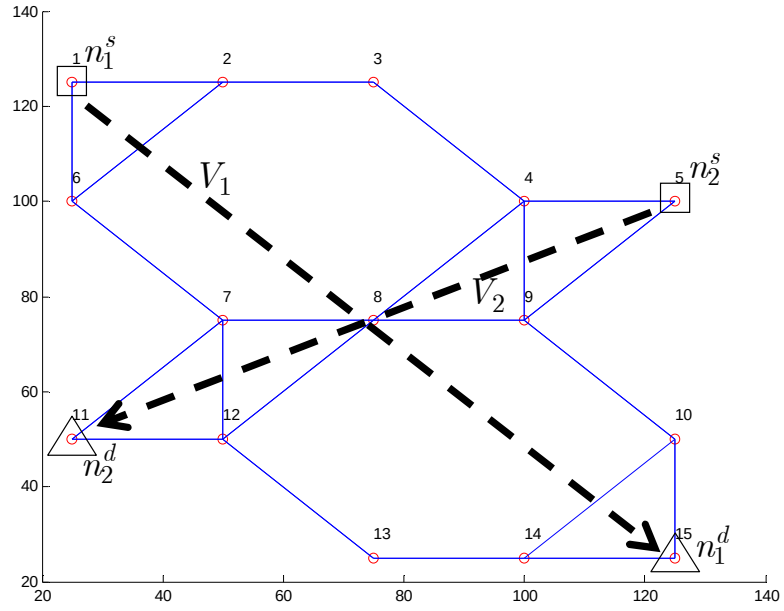


Fig. 7.8. Wireless network settings for the simulation of two video streams.

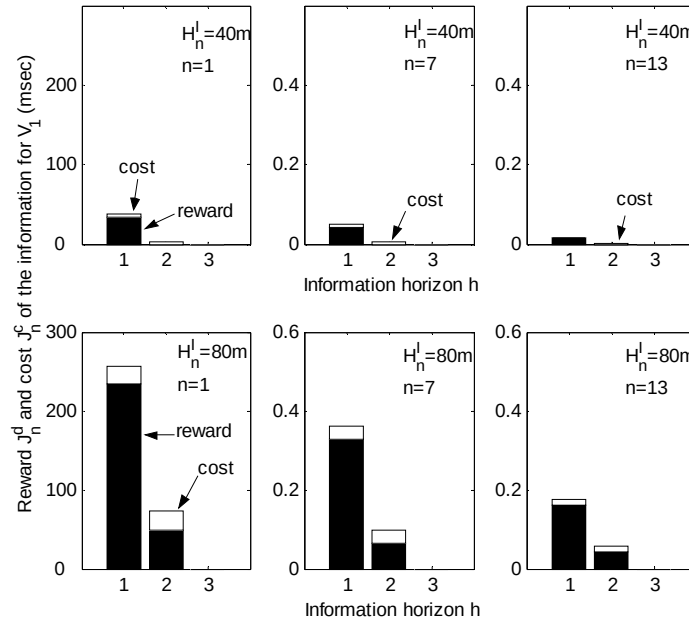


Fig.7.9. Reward J_n^d and cost J_n^c of different information horizon at different node for video V_1 .

A. Reward and cost of the information exchange

First, we simulate the impact of the information including the reward J_n^d (see equation (13)) and cost J_n^c (see equation (20)) from the expected delay $E[\tilde{d}_n]$ using the adaptive fictitious play in Section VII with different information horizons. Figure 7.9 shows the resulting reward and cost of information at different locations for streaming video V_1 (at node $n = 1, 7$, and 13 on one of the routes of video V_1). The results show that a 1-hop information cell is enough when the interference range is 40 meters, since only the nodes that are 1 hop away can interfere with each other. If the interference range is 80 meters, the information exchange mismatch problem (see Section V) occurs and the appropriate information horizon for information exchange is then increased to 2.

B. Application layer performance with different information horizons and interference ranges

We next compare the proposed dynamic resource management algorithm using adaptive fictitious play (AFP) with two other resource management methods – AODV [PR99] with load balancing over the two available frequency channels (AODV/LB) and the Dynamic Least Interference Channel Selection [KP99] (DCS) extended to a network setting. Table 7.1 and 7.2 show the results of the Y-PSNR of the two video sequences using different approaches. The results show that the proposed algorithm using learning from the nodes within the information cell outperforms the alternative approaches. Especially, when the interference range is large ($H_n^I = 80$ meters), the proposed AFP approach significantly improves the video quality (X represents PSNR below 26 dB, which is unacceptable for a viewer).

TABLE 7.1. Y-PSNR OF THE TWO VIDEO SEQUENCES USING VARIOUS APPROACHES ($H_n^I = 40$ METERS).

Network Bandwidth		Y-PSNR (dB)		
		AODV/LB	DCS	AFP (1-hop information cell)
Average $T = 5.5$ Mbps	V_1	32.47	35.21	35.61
	V_2	31.70	33.32	33.32

TABLE 7.2. Y-PSNR OF THE TWO VIDEO SEQUENCES USING VARIOUS APPROACHES ($H_n^I = 80$ METERS).

Network Bandwidth		Y-PSNR (dB)			
		AODV/LB	DCS	AFP (1-hop information cell)	AFP (2-hop information cell)
Average $T = 5.5$ Mbps	V_1	X	X	28.19	29.80
	V_2	X	X	31.26	31.70
Average $T = 10$ Mbps	V_1	30.47	34.46	35.61	35.61
	V_2	31.92	33.08	33.32	33.32

For delay sensitive applications, we measure the packet loss rate (i.e. the probability that the end-to-end delay exceeds the delay deadline) for different approaches in Figure 7.10(a). The results of both applications are shown. The AODV represents the on-demand routing solution with only 1 frequency channel. The AODV/LB approach randomly distributes packets over the two available frequency channels. The DCS approach with cognitive ability selects a better frequency channel based on the link measurements and hence, improves the performance opposed to the AODV/LB. The AFP further improves the performance of both applications by learning the behaviors of the neighbor nodes. Interestingly, the benefit brought by the learning capability decreases as the network bandwidth increases. In other words, it is not worthy to be too intelligent in an environment with plenty of resource. Moreover, as shown in Figure 7.10(b), the improvement of 2-hop information cell is limited when the interference range is 40 meters. This is because the nodes that are two hops away have no impact on the current node and their information is not valuable (i.e. it does not impact the utility).

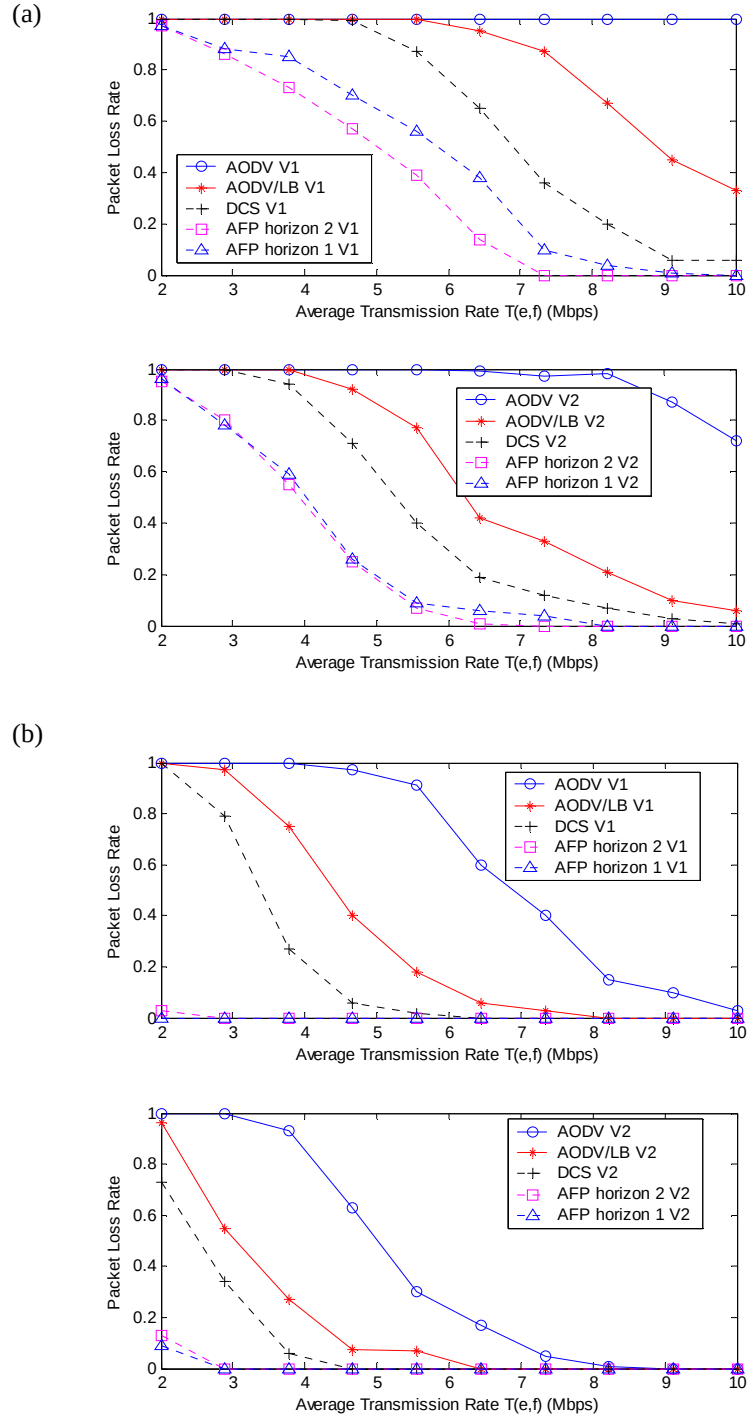


Fig. 7.10. (a) Packet loss rate vs. average transmission bandwidth using different approaches ($H_n^I = 80$ meters). (b) Packet loss rate vs. average transmission bandwidth using different approaches ($H_n^I = 40$ meters).

C. Reducing the frequency of learning

When the interference range is 40 meters, Figure 7.10(b) shows that the AFP with 1-hop information cell is better than with 2-hop information cell, since 1-hop information cell has smaller cost of information exchange. In addition to reducing the information horizon, reducing the frequency of learning b_n/c at all the nodes can also reduce the cost of information exchange. Figure 7.11 shows the packet loss rate of the two applications with different information horizon when b_n/c changing from 1 to 0.5. As the learning frequency b_n/c decreases, the packet loss rate decreases with the cost of information exchange. However, it is shown that when $b_n/c < 0.6$, the AFP becomes inefficient and the packet loss rate starts increasing for both applications. In other words, changing the frequency of learning will also lead to a trade-off between the learning efficiency and the information overhead. The information overhead decreases when the learning frequency b_n/c decreases and hence, the packet loss rate decreases. However, when the learning frequency is too slow ($b_n/c < 0.6$), the learning efficiency decreases and this results in an increasing packet loss rate.

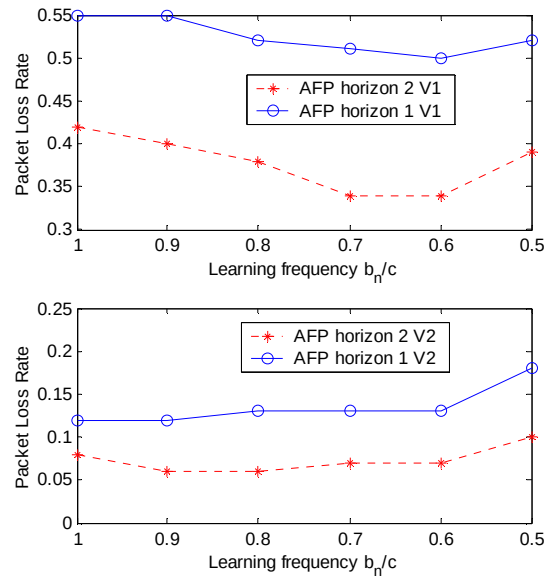


Fig. 7.11. Packet loss rate vs. learning frequency b_n/c (average $T = 5.5$ Mbps, $H_n^I = 80$ meters).

D. Impact of the primary users

The simulation implies that the reward of information is also impacted by the existence of the primary users. Next, we consider the impact of the primary users, which always have higher priority to access the pre-assigned frequency channels than the network nodes in Figure 7.8. Assume that the frequency channel F_1 is occupied by the primary users with time fraction $\rho = 0\%, 20\%, 40\%, 60\%$, and 80% around a certain congestion region (network nodes $n = 7, 11, 12$) in Figure 7.8. Figure 7.12 shows the packet loss rate for the two video streams using the AFP with various information horizons. The average transmission rate is set to 5.5 Mbps, $b_n / c = 1$, and the interference range is 80 meters.

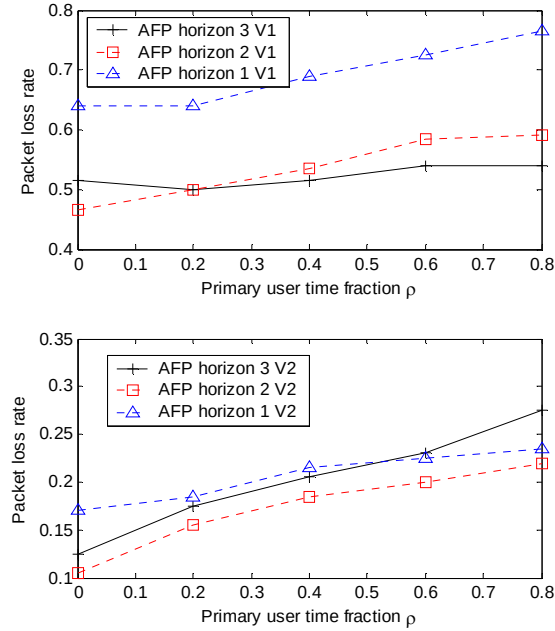


Fig. 7.12. Packet loss rate vs. time fraction ρ of the primary users occupying frequency channel F_1 around network node $n = 7, 11, 12$ (average $T = 5.5\text{Mbps}$, $b_n / c = 1$, $H_n^I = 80$ meters).

The results show that as the time fraction ρ increases, the packet loss rates of both applications increase, since fewer resources are available for the secondary users to transmit the packets. As the simulation in the previous subsection, when the interference

range is 80 meters, AFP with 2-hop information cell still performs better than 1-hop information cell case. Interestingly, for application V_1 , AFP with 3-hop information cell performs even better in a large ρ case, even though more cost of information is needed. This is because the congestion region are more likely to be discovered at the source node $n=1$ and detour the packets through other routes. However, such advantage is not exploited by the application V_2 , since its destination node is affected by the primary users and there is no way to detour the packets. Note that when there is no primary user ($\rho = 0$), AFP with 3-hop information cell performs worse than 2-hop case due to the larger cost of information exchange.

E. Impact of mobility

In this subsection, we consider the impact of mobility on the video performance. We adopt a well-known mobility model, the “random walk” [CBD02], in which the relay nodes (secondary users) shown in Figure 8 randomly select a direction at each time slot and move at a fixed speed v . We simulate the speed v ranging from 0 to 1 meters/sec. We assume that there is no primary user, i.e. $\rho = 0$. The average transmission rate is set to 8 Mbps, $b_n/c = 1$, and the interference range is 80 meters. Figure 7.13 illustrates the packet loss rate as the mobility changes for different information horizons. The results show that the mobility degrades the performance of both applications. When the mobility v is small, AFP with information horizon $h = 2$ performs better than with information horizon $h = 1$ as in the previous simulations with $H_n^I = 80$ meters. However, for video V_2 , when the mobility exceeds 0.6 meters/sec, the best information horizon changes from $h = 2$ to $h = 1$. This is because the increased mobility will decrease the information accuracy and hence, the required information horizon also decreases. Note that for video V_1 , the AFP with information horizon $h = 2$ still performs better than with information horizon $h = 1$. This is because the video V_1 has a longer route and thus, modeling more interfering neighbor nodes, using a larger information horizon, is still beneficial.

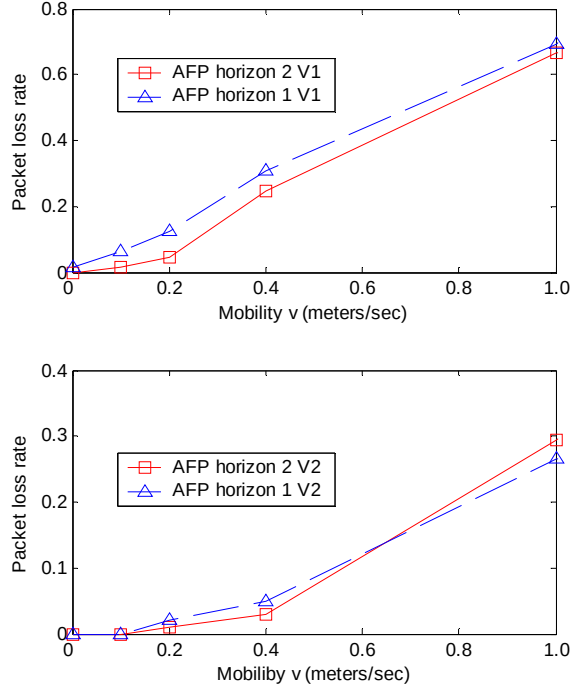


Fig. 7.13. Packet loss rate vs. mobility v of the secondary users (network relays) (average $T = 8\text{Mbps}$, $\rho = 0$, $b_n / c = 1$, $H_n^I = 80$ meters).

VIII. CONCLUSIONS

In this chapter, we show that the distributed resource management solution using adaptive fictitious play significantly improves the performance of delay sensitive applications transmitted over a multi-hop cognitive radio network. We assume that the autonomous secondary users are able to learn the spectrum opportunities based on the information exchange. The proposed approach can also be used to support QoS for general multi-radio wireless networks, when there is no primary user. This situation is also brought up in [ALV06], when the secondary users are competing in the unlicensed band (i.e. ISM band), where there is no primary user. Importantly, based on the value of the obtained information (i.e. the impact on decreasing the expected end-to-end delay), we define the information horizon in our adaptive fictitious play. In addition to the reward, the cost of the information exchange is also considered in terms of transmission

time overheads. Various approaches of decreasing this time overhead are discussed, and their performance impact is quantified. Our simulation results show that the benefit from various information horizons can be different for distinct applications with various delays and quality impacts, especially when primary users are present in the network at different locations.

Chapter 8

Conjecture-Based Channel Selection in Multi-Channel Wireless Networks

I. INTRODUCTION

In this chapter, we provide a fundamental view of channel selection in multi-channel MAC protocols that aims to minimize the delays of delay-sensitive users transmitting their packets through the multi-channel wireless network. Since the delay of a user is impacted by the channel selection strategies of the other network users, it is important that users consider the impact of these other users while determining their own channel selection strategy. We endow the users with the ability to build beliefs about the aggregate response of the other users to their actions (the aggregate response in this chapter is the remaining capacity in each channel that can be measured based on the throughput estimation method [SCN03]) and efficiently minimize their expected future delays in a foresighted manner. Specifically, we model the multi-user interaction as a channel selection game played by users who are capable of making conjectures about how their transmission actions (i.e. their channel selection) will impact other users and eventually impact their future performance. We investigate the performance of the resulting ε -consistent conjectural equilibrium obtained when these users interact based on their conjectures about the future remaining capacities when selecting channels. The proposed ε -consistent conjectural equilibrium is a relaxed version of the conventional conjectural equilibrium [Hah77], which allows us to characterize the equilibrium obtained when network users are able to build near-accurate conjectures.

The channel selection problem was first studied in cellular networks. Various channel assignment schemes have been proposed (see e.g. [KN96] for an excellent survey).

However, most of these channel assignment schemes are based on centralized solutions, which do not scale to the network size and/or are not suitable for wireless networks without a fixed infrastructure, such as ad hoc wireless networks. Moreover, centralized approaches are especially not desirable for delay-sensitive applications as considered in this chapter. The reason is that these centralized solutions require propagating control messages back and forth to a network coordinator, thereby incurring delays that are often unacceptable for delay-sensitive applications [SV08].

To cope with these challenges, distributed channel selection schemes without a network manager have also been proposed in various types of wireless networks, such as wireless ad hoc networks [NZD02][JDN01][SV04], wireless mesh networks [RC05], and cognitive radio networks [CZ05][ZC05][HBH05][SV08], etc. For instance, in wireless ad hoc networks, Nasipuri et al. [NZD02] proposed a multi-channel carrier sense multiple access (CSMA) protocol that identifies the set of idle channels and selects the best channel for transmission based on the channel condition observed at the transmitter side. Jain et al. [JDN01] assumed a separate control channel and proposed an alternate multi-channel CSMA protocol that selects the best channel based on the channel condition observed at the receiver side. So and Vaidya [SV04] proposed a solution that allows users to perform request-to-send (RTS)/clear-to-send (CTS) negotiation without a separate control channel. However, these solutions are myopic, because the autonomous users only adapt to their latest network measurement (e.g. idle channel set, channel condition). These solutions can be inefficient, since the users only react to the latest contention measurements experienced in the different wireless channels.

In emerging cognitive radio networks, a key challenge is how the secondary users can select their transmission channels in order to optimize their performance. Zheng and Cao [ZC05] provided five rule-based spectrum management schemes where users measure local interference patterns and act independently according to the prescribed rules. J. Huang et al. [HBH05] proposed a spectrum sharing scheme where users can select

multiple channels to transmit packets and exchange interference prices for each channel. These distributed schemes assume that users cooperate in order to efficiently coordinate their channel selection strategies. However, as discussed in e.g. [RHA04], users can decide to deviate from the rules prescribed by the MAC protocols as long as they derive a higher utility when deviating. That is, users in the network may not have incentives to cooperate and maximize a network/system performance, because this would not maximize their own utilities. Non-cooperative games were proposed to characterize and analyze the performance of self-interested users interacting in different communication systems. For example, Lee et al. [LTH07] showed that the current back-off based MAC protocols can be modeled as a non-cooperative channel access game. The distributed channel selection problem was studied by Felegyhazi et al. [FCB07], who showed that users autonomously selecting channels in non-cooperative multi-channel wireless networks converge to the Nash Equilibrium (NE). However, it is well-known that the NE can often be Pareto-inefficient. For instance, it is possible that some of the selfish users will improve their performance at the cost of degrading the system-wide performance. To optimize the multi-user system utility, a Network Utility Maximization (NUM) framework has been introduced in [LCC07]. It has been shown that by allowing users to exchange messages, they can determine a wireless channel access strategy that reaches a Pareto-efficient solution in a distributed manner. Similar concepts have been proposed in [WZQ08] for distributed channel selection, where pricing has been deployed in order to enable users to maximize the system throughput in a distributed manner. To determine the resource price, message exchanges among users are necessary. However, such message exchanges among users can be undesirable due to their increased computational and communication overhead, or simply due to security issues, protocol limitations, etc. Moreover, the incentives for the users to add a penalty term in their utility functions in order to collaborate with each other are not addressed. Alternatively, a distributed channel access scheme using simple random access algorithms without message

exchanges was discussed in [PYC08]. However, this solution can only achieve a near optimal system-wise throughput if there are no message exchanges among the participating users.

In this chapter, we develop a distributed channel selection scheme for multi-channel wireless networks. We show that it is possible for users to achieve a system-wise optimal solution without the need for message exchanges when users are able to make foresighted decisions based on their future expected utilities. Their foresighted interaction also provides them the necessary incentives to collaborate, because they can now determine their own performance benefits resulting from their voluntary collaboration with the other users. We investigate in this chapter the multi-user communication scenarios under which a system-wise optimal solution can be reached by the autonomous users.

This chapter considers how autonomous users can transmit delay-sensitive traffic over the same multi-channel wireless network. The autonomous users will dynamically select the channels in which they should send their traffic in a distributed and strategic manner, by estimating their expected utilities from taking various transmission actions based on their available conjectures about the communication system. Specifically, we discuss two new concepts that enable the network users to make strategic decisions and maximize their own utilities in distributed wireless networks, without the need of message exchanges with other users:

- *Foresighted channel selection strategies.* As mentioned previously, the users' strategies are coupled in multi-user wireless environments since the channel selection of a user impacts and is impacted by the other users. Thus, users need to select their channels by considering not only the impact of their actions on their immediate experienced utilities, but also on their long term utilities. For instance, a user's aggressive strategy may be rewarded in the short term, but this will trigger the other users to adapt their own strategies, which will impact its long term reward. Hence, foresighted users need to build accurate models (conjectures) about how their actions

are coupled with that of the other users and, based on these models, make foresighted decisions on how to adapt their transmission strategies in real-time.

- *Learning accurate coupling models based on local information.* To build these coupling models, the foresighted users can adopt interactive learning approaches to update their beliefs about the expected response of the other users to their actions. Specifically, we propose learning approaches for foresighted users to build their beliefs in a distributed manner, given only their local information (i.e. their own measurement history).

We provide foresighted channel selection strategies for the following two communication scenarios – 1) when the system has only one foresighted user, and 2) when the system has multiple foresighted users. We are able to analytically show that when the system has only one foresighted user, this user can deploy a linear belief function to model the aggregate response of the other users. In [WH98], a foresighted user is assumed to model the market price also as a linear function of its desired demand. However, we note that using the linear model is purely heuristic in [WH98]. In this chapter, we will show that such a linear belief function is able to capture the specific structure of the considered multi-user interaction. When there is only one foresighted user, we investigate two different situations. We show that when the foresighted user is altruistic (e.g. whenever it acts as a network leader), it can drive the system to the system-wise Pareto optimal solution by modeling the reactions of the other myopic users. Alternatively, if the foresighted user is self-interested, we show that this user will benefit itself at the expense of (some of) the myopic users increased delays. If the system has an increased number of foresighted users building beliefs simultaneously, these users' beliefs will become inconsistent and users will experience performance degradation. To enable multiple foresighted users to build consistent beliefs about each other, they need to obey the rules prescribed by the MAC protocol. We also show how these autonomous users can comply with the rule-based solution, such that the distributed channel selection

reaches the system-wise Pareto optimal solution when all users decide their channel selection strategies in an autonomous manner.

The chapter is organized as follows. Section II discusses the considered wireless network model and formulates the foresighted channel selection problem for autonomous delay-sensitive users. In Section III, we define the conjecture-based channel selection game for the foresighted users and the ε -consistent conjectural equilibrium of the game. In Section IV, we investigate the case when there is only one foresighted user in the network. We provide a learning algorithm for the foresighted user to update its belief. In Section V, we further discuss the case when there are multiple foresighted users in the network. The numerical results are shown in Section VI and Section VII concludes the chapter.

II. PROBLEM FORMULATION FOR FORESIGHTED CHANNEL SELECTION

A. Network model

We assume that there are M autonomous network users sharing the same multi-channel wireless network. Let $\mathbf{V} = \{v_i, i = 1, \dots, M\}$ represent the set of these users. User v_i is composed by a source-destination pair, i.e. $v_i = (v_i^s, v_i^d)$. We assume that there are N non-overlapping channels for these users to transmit their delay-sensitive applications. Let $\mathbf{r} = \{r_j, j = 1, \dots, N\}$ represents the set of all these non-overlapping frequency channels.

We assume that each user v_i wants to serve an application with traffic rate x_i (bps). Each frequency channel r_j has a capacity W_j ¹ (bps). In this chapter, we assume an unsaturated network condition, in which the total capacity is more than the total traffic rate of the users, i.e. $\sum_{j=1}^N W_j > \sum_{i=1}^M x_i$. Each wireless channel access can then be modeled as a queue [CW05]. Such unsaturated condition can ensure that a user can always find an unsaturated channel to transmit its traffic, and hence, the queuing delays

¹ For simplicity, we assume that each virtual queue has the same capacity for every user. However, the analysis provided in this chapter can be generalized to the case when each virtual queue has different capacities for different users by adopting a more sophisticated queuing model.

can be bounded. The network queuing model is illustrated in Figure 8.1. For each wireless channel, the maximum channel service rate is $C_j = W_j / L$ (packets/second), where L is the average packet length. When more users access the same channel, the channel service rate reduces due to the contention. The resulting service rate is measured by user v_i when accessing channel r_j and it is referred to the remaining capacity C_{ij} in this chapter. This is regarded as the local information of user v_i , e.g. the throughput estimation method proposed in [SCN03], based on which it makes its channel selection decision.

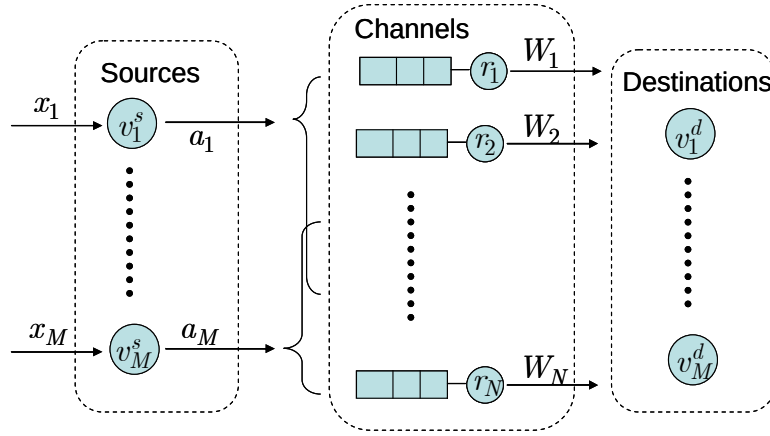


Fig. 8.1 Considered queuing model for multi-user channel access.

Next, we define the distributed wireless channel selection problem in more detail. An autonomous wireless user needs to autonomously determine its traffic rate to transmit on each frequency channel. We denote the probability of user v_i to select the channel r_j as $a_{ij} \in [0, 1]$. Let $\mathbf{a}_i = [a_{i1}, \dots, a_{iN}] \in [0, 1]^N$ be the channel selection probability distribution of user v_i , where $\sum_{j=1}^N a_{ij} = 1$. The traffic rate from user v_i through the channel r_j is denoted as λ_{ij} (packet/second), where $\lambda_{ij} = x_i a_{ij} / L$ and $\sum_{j=1}^N \lambda_{ij} = x_i / L$, and we denote $\boldsymbol{\sigma}_i = [\lambda_{ij}, j = 1, \dots, N]$ as the traffic distribution of user v_i , and $\boldsymbol{\sigma}_{-i}$ as the traffic distribution for the other users except v_i ($\boldsymbol{\sigma} = [\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_{-i}]$). The total traffic rate on the

channel r_j is denoted as λ_j and $\lambda_j = \sum_{i=1}^M \lambda_{ij}$.

As in [CW05] we assume that each user deploys an application generating a Poisson packet arrival. We assume that the delay through each frequency channel can be modeled using an M/M/1 queuing model. The expected delay through the channel r_j can then be expressed as:

$$E[D_j] = \begin{cases} \frac{1}{C_j - \lambda_j}, & \text{if } C_j > \lambda_j \\ \infty, & \text{otherwise} \end{cases}. \quad (1)$$

The delay of user v_i is defined as:

$$U_i(\sigma_i, \sigma_{-i}) = \sum_{j=1}^N a_{ij} E[D_{ij}] = \frac{L}{x_i} \sum_{j=1}^N \frac{\lambda_{ij}}{C_{ij}(\sigma_{-i}) - \lambda_{ij}}, \quad (2)$$

where $C_{ij}(\sigma_{-i})$ is the measured remaining capacity (an aggregate response of the other users' channel selection) for a specific user v_i using channel r_j . Since in a wireless channel $E[D_{ij}] = E[D_j]$, following equation (1) and (2), we have $C_{ij}(\sigma_{-i}) = C_j - \sum_{i' \neq i} \lambda_{i'j}$ ². Note that in the considered network, there is no information exchange among the users. We assume that if user v_i changes its traffic $\Delta\lambda_{ij}$ in channel r_j , another user $v_{i'}$ can measure the resulting changes in the remaining capacity of channel r_j as $C_{i'j}(\Delta\lambda_{ij}) = C_{i'j}^0 - \Delta\lambda_{ij}$, where $C_{i'j}^0$ is the remaining capacity when $\Delta\lambda_{ij} = 0$.

Next, we first discuss how the multi-user multi-channel selection problem can be solved using a conventional, centralized approach, which requires message exchanges (between the users and a central network manager).

B. Conventional centralized decision making

In general, centralized resource management methods aim at implementing Pareto efficient solutions, which optimize the “system welfare”, e.g. they minimize the weighted summation of users' utilities, i.e. $U(\sigma) = \sum_{i=1}^M w_i U_i(\sigma)$, where w_i represents the weighting

² We assume that this remaining capacity can be measured by user v_i based on the throughput estimation method as in [SCN03]. This value is analytically true when the M/M/1 queuing model in each channel is valid.

parameters.

Definition 1: Pareto boundary. Given different users' weights $\mathbf{w} = [w_i, i = 1, \dots, M \mid w_i > 0, \sum_{i=1}^M w_i = 1]$, the Pareto boundary is formed by the solutions of the following multi-user multi-channel selection problem:

$$\begin{aligned} \sigma^P(\mathbf{w}) = \arg \min_{\sigma \geq 0} & \sum_{i=1}^M w_i U_i(\sigma) \\ \text{s.t.} & \sum_{j=1}^N \lambda_{ij} = x_i / L, \text{ for } \forall v_i \end{aligned} \quad (3)$$

In order to perform the above centralized optimization, the network manager needs to collect the global network information $\mathcal{Z}_g = [\{C_j, \forall r_j\}, \{x_i, \forall v_i\}, \{w_i, \forall v_i\}]$. Specifically, in this chapter, we define the system-wise utility as

$U^{fair}(\sigma) = \sum_{i=1}^M x_i U_i(\sigma) = \sum_{j=1}^N \frac{L \sum_{i=1}^M \lambda_{ij}}{C_j - \sum_{i=1}^M \lambda_{ij}}$. Based on Little's formula [Kle75], this utility represents the total queue size of these N M/M/1 queues for the N channels.

Definition 2: System-wise Pareto optimal solution. The system-wise Pareto optimal solution is then defined as:

$$\begin{aligned} \sigma^P = \arg \min_{\sigma \geq 0} & U^{fair}(\sigma) \\ \text{s.t.} & \sum_{j=1}^N \lambda_{ij} = x_i / L, \text{ for } \forall v_i \end{aligned} \quad (4)$$

The system-wise Pareto optimal solution is on the Pareto boundary where the users' weights are proportional to the traffic rates of the users. However, such centralized approach may be undesirable in many settings due to two reasons: a) high message overhead required for exchanging the control information and b) users may not have incentives to comply with the allocation solution σ^P imposed by the central manager. These motivate the adoption of distributed resource management approaches, which do not require any message exchanges.

C. Conventional distributed decision making

In a distributed resource management, the objective of a user v_i is to minimize its

delay over all possible wireless channels that it chooses. The traffic distribution of the other users σ_{-i} may not be observable for user v_i , but v_i can measure the aggregate response $C_{ij}(\sigma_{-i})$. To perform equation (5), user v_i needs to observe the local information $\mathcal{Z}_i = [\{C_{ij}, \forall r_j\}, \{C_j, \forall r_j\}, x_i]$. Based on it, the following best response is adopted by every user in the network:

$$\begin{aligned} \pi_i(\mathcal{Z}_i) = \arg \min_{\sigma_i \geq 0} U_i(\sigma_i, \mathcal{Z}_i) \\ \text{s.t. } \sum_{j=1}^N \lambda_{ij} = x_i / L \end{aligned} \quad (5)$$

where π_i represents the myopic policy for channel selection. The solution to the problem in equation (5) will lead to a NE, as proven in [KLO97] for a network routing scenario, similar to the considered channel selection setting. Based on [KLO97], the optimal channel selection probability for user v_i to transmit in channel r_j can be expressed as

$$a_{ij}^* = \lambda_{ij}^* L / x_i, \text{ and } \lambda_{ij}^* = \max\{0, C_{ij} - \alpha_{ij} R_i\}, \quad (6)$$

where $R_i = \sum_{r_j \in \Omega_i} C_{ij} - x_i / L$ represents the overall remaining capacity after user v_i sends its traffic x_i , Ω_i represents the set of channels for which $\lambda_{ij} > 0$, and

$\alpha_{ij} = \frac{\sqrt{C_{ij}}}{\sum_{r_j \in \Omega_i} \sqrt{C_{ij}}}$ represents the optimal fraction (in terms of minimizing U_i), based on which R_i is allocated to channel r_j . The difference between the measured remaining capacity C_{ij} and $\alpha_{ij} R_i$ is the optimal λ_{ij}^* for user v_i to put on channel r_j .

Note that $\{C_j, \forall r_j\}$ and x_i is time-invariant, and $\{C_{ij}, \forall r_j\}$ is time-variant. To reach the NE, users repeatedly measure the remaining capacities $\{C_{ij}, \forall r_j\}$ and interact with each other using the best response in equation (5). Specifically, user v_i will update its traffic rate on the channel r_j as:

$$\lambda_{ij}^t = \min\{0, C_{ij}^{t-1} - \alpha_{ij}(C_{ij}^{t-1})R_i\}, \alpha_{ij} = \frac{\sqrt{C_{ij}^{t-1}}}{\sum_{r_j \in \Omega_i} \sqrt{C_{ij}^{t-1}}}. \quad (7)$$

However, the resulting NE is Pareto inefficient [KLO95]. Hence, in this chapter, we investigate how to improve the efficiency of the multi-user interaction to achieve the

system-wise Pareto optimal solution in a distributed manner. We endow users with the ability to build belief functions $\mathbf{B}_i(\boldsymbol{\sigma}_i)$ on the remaining capacities C_{ij} (instead of using the latest measurement) for user v_i to take into account the impact of $\boldsymbol{\sigma}_i$ on C_{ij} . We refer to this approach as foresighted decision making because it enables users to predict how their channel selection will impact the decision of the other users and thereby, impact the future remaining capacities. Next, we discuss this distributed foresighted resource management approach.

D. Foresighted decision making

By adopting a belief function $\mathbf{B}_i(\boldsymbol{\sigma}_i)$, the distributed optimization in equation (5) is formulated as

$$\begin{aligned} \pi_i^f(\mathbf{B}_i, \mathcal{I}_i) = \arg \min_{\boldsymbol{\sigma}_i \geq 0} U_i(\boldsymbol{\sigma}_i, \mathbf{B}_i(\boldsymbol{\sigma}_i), \mathcal{I}_i) \\ \text{s.t. } \sum_{j=1}^N \lambda_{ij} = x_i / L \end{aligned} \quad (8)$$

where $\mathbf{B}_i(\boldsymbol{\sigma}_i)$ represents the conjecture (belief) of user v_i on the expected remaining capacity over each frequency channel when the traffic distribution $\boldsymbol{\sigma}_i$ is taken. This belief is built based on the measurement history $o_i^t = \{(\lambda_{ij}^{t-k}, C_{ij}^{t-k}), k = 1, \dots, S, j = 1, \dots, N\}$, where S is the observation window size. In Figure 8.2, we provide a block diagram to highlight the main differences between the myopic channel selection approaches and the proposed foresighted channel selection. Comparing the optimal foresighted policy in equation (8) and the policy in equation (5), there are two main differences.

- 1) Unlike in equation (5), the provided policy in equation (8) does not depend only on the current remaining capacities $\{C_{ij}^{t-1}, j = 1, \dots, N\}$. Alternatively, user v_i can determine its expected remaining capacities when it takes a certain traffic distribution $\boldsymbol{\sigma}_i$ by learning and updating its belief $\mathbf{B}_i(\boldsymbol{\sigma}_i)$ based on its measurement history o_i^t .
- 2) The delays in equation (5) are based on the latest measurements of the remaining capacities. Hence, instead of minimizing the delay in equation (5) in a myopic manner, the expected delay $U_i(\boldsymbol{\sigma}_i, \mathbf{B}_i)$, which is considered as the future delay, is minimized in

equation (8).

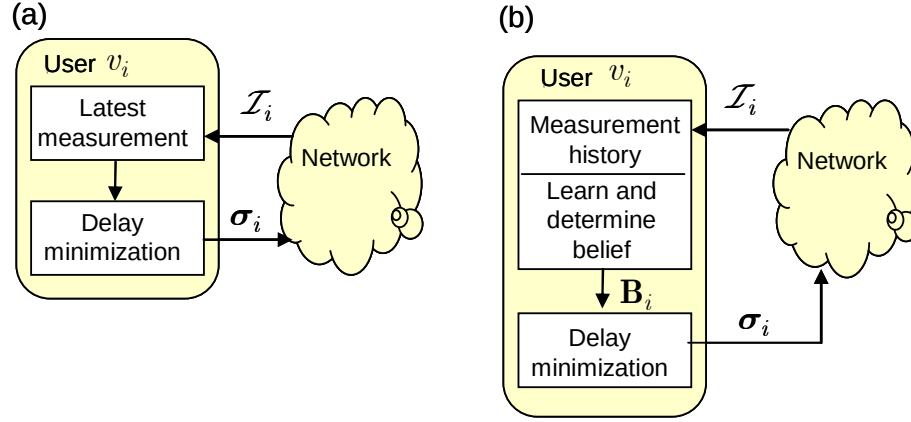


Fig. 8.2 Block diagram of the (a) myopic channel selection and (b) foresighted channel selection.

III. CONJECTURE-BASED CHANNEL SELECTION GAME AND THE CONJECTURAL EQUILIBRIUM

In a network, there are users who adopt the myopic channel selection or adopt the foresighted channel selection. We formalize the multi-user interaction in a multi-channel network using the following repeated game.

Definition 3: Conjecture-based channel selection game. We consider the conjecture-based channel selection game as a stage game represented by the following tuple $\langle \mathbf{V}, \Lambda, \mathcal{S}, \mathbf{U} \rangle$.

- $\mathbf{V}$ is the set of players (users), and we assume that there are two types of users in the network: a set of foresighted users in $\mathbf{V}^F$ and a set of myopic users in $\mathbf{V}^M$, i.e. $\mathbf{V} = \{\mathbf{V}^F, \mathbf{V}^M\}$.
- Λ is the action space of the system, where $\Lambda = \Lambda_1 \times \dots \times \Lambda_M$. The action of user v_i is defined as the traffic distribution $\sigma_i = [\lambda_{ij}, \forall r_j] \in \Lambda_i$.
- $\mathcal{S}$ is the conjecture space of all the users, i.e. $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \dots \times \mathcal{S}_M$. The *conjecture* of

user v_i is defined as its belief about the expected remaining capacities $\mathbf{B}_i = [\tilde{C}_{ij}(\lambda_{ij}), \forall r_j] \in \mathcal{S}_i$. We will discuss how to construct the function $\tilde{C}_{ij}(\lambda_{ij})$ in Section IV. B. This function models the remaining capacities for user v_i . Such models implicitly provide the user v_i with an aggregate belief regarding the coupling of its actions to that of the other users.

- $\mathbf{U}$ is a delay vector of the users, i.e. $\mathbf{U} = [U_i(\sigma_i, \mathbf{B}_i), \forall v_i]$.

The stage game is played repeatedly by the users with the following two types of belief updating methods:

a) *Myopic users*: A myopic user v_i will update its belief function using $\mathbf{B}_i^t = [C_{ij}^{t-1}, \forall r_j]$ in the repeated game. As a result, user v_i will select its new action σ_i^t based on the latest measurements obtained about the remaining capacities, using the myopic best response in equation (7).

b) *Foresighted users*: A foresighted user v_i will update its belief function using $\mathbf{B}_i^t(\sigma_i^t) = [\tilde{C}_{ij}^t(\lambda_{ij}^t), \forall r_j]$ in the repeated game and select its new action σ_i^t using equation (8). We will discuss how to learn the belief function $\tilde{C}_{ij}^t(\lambda_{ij}^t)$ in Section IV.

Note that the actual (real) remaining capacities $[C_{ij}(\sigma), j = 1, \dots, N]$ depend on σ . However, user v_i 's conjecture is the expected remaining capacities on the various channels $\tilde{C}_{ij}(\sigma_i)$ given only σ_i . Based on these conjectures, we can define the concept of a Conjectural Equilibrium (CE) for the considered channel selection game. The CE was first discussed by Hahn in the context of a market model [Hah77]. A general multi-agent framework is proposed in [WH98] to study the existence of and the convergence to CE in market interactions.

Definition 4: Conjectural equilibrium of the channel selection game. Following the definition in [Hah77], the conjectural equilibrium (CE) is defined as $\sigma^* \in \Lambda$, if for each user $v_i \in \mathbf{V}$, the following two conditions are satisfied:

- (a) The expected remaining capacities at the equilibrium are the actual remaining capacities, i.e. $\tilde{C}_{ij}^*(\sigma_i^*) = C_{ij}^*(\sigma^*), \forall r_j$.

(b) The action at the equilibrium σ_i^* minimizes $U_i(\sigma_i, \{\tilde{C}_{ij}^*(\sigma_i^*), j = 1, \dots, N\})$.

The belief function $B_i^t(\sigma_i^t)$ may not be perfectly estimated at the equilibrium in practice. However, a user can still keep selecting the same action with imperfect belief estimation, as long as that action consistently minimizes the expected utility. For this, we define an extension to the well-known CE, where users' actions converge to the equilibrium based on their “imperfect” beliefs.

Definition 5: ε -consistent conjectural equilibrium of the channel selection game. The ε -consistent conjectural equilibrium (ε -CE) is defined as $\sigma^* \in \Lambda$, if for each user $v_i \in \mathbf{V}$, the following two conditions are satisfied:

(a) The expected remaining capacities at the equilibrium approximate the actual remaining capacities, i.e.

$$\max_{v_i \in \mathbf{V}} \max_{r_j \in A_i} \left(\tilde{C}_{ij}^*(\sigma_i^*) - C_{ij}^*(\sigma^*) \right)^2 \leq \varepsilon. \quad (9)$$

(b) The action at the equilibrium σ_i^* minimizes its expected delay $U_i(\sigma_i, B_i(\sigma_i^*))$.

Note that as the CE, ε -CE may not exist and, even if it exists, it may not be a unique equilibrium. Next, we will discuss how a user should build its conjecture (belief) that leads to the ε -CE and compare the resulting performance with the system-wise Pareto optimal solution in various scenarios. In Section IV, we investigate the case when the system has only one foresighted user and in Section V, the case when multiple foresighted users interact.

IV. DISTRIBUTED CHANNEL SELECTION WHEN THERE IS ONLY ONE FORESIGHTED USER

A. Belief function when only one user is foresighted

In this subsection, we assume that only user v_1 is foresighted and the other users are myopic in the conjecture-based channel selection game. We then discuss how to construct the belief function $B_1(\sigma_1) = [\tilde{C}_{1j}(\sigma_1), j = 1, \dots, N]$ in equation (8). Given the traffic distribution of the user v_1 , the channel selection game of the other myopic users will

reach NE. Note that when user v_1 puts more traffic³ $\Delta\lambda_{1j}$ into channel r_j , the lower remaining capacity $C_{i'j}$ will be measured by the other users, which leads to another NE.

Proposition 1: Linearity of the belief function in the case of one foresighted user. The belief function $\mathbf{B}_1(\boldsymbol{\sigma}_1) = [\tilde{C}_{1j}(\lambda_{1j}), \forall r_j]$ can be approximately modeled as a linear belief function when there is only one foresighted user in the wireless network.

Proof: From equation (6), the remaining capacity $C_{1j}(\Delta\lambda_{1j}) = C_j - \sum_{v_i \neq v_1} \lambda_{i'j}(\Delta\lambda_{1j})$ can be expressed as:

$$\begin{aligned} C_{1j}(\Delta\lambda_{1j}) &= C_j - \sum_{v_i \in \mathbf{V}^M} (C_{i'j}(\Delta\lambda_{1j}) - \alpha_{i'j}(\Delta\lambda_{1j})R_{i'}) \\ &= \underbrace{C_j - \sum_{v_i \in \mathbf{V}^M} C_{i'j}^0}_{\text{constant}} + \sum_{v_i \in \mathbf{V}^M} \Delta\lambda_{1j} + \sum_{v_i \in \mathbf{V}^M} \alpha_{i'j}(\Delta\lambda_{1j})R_{i'}. \end{aligned}$$

Note that the last term can be written as follows using the Taylor expansion:

$$\begin{aligned} \alpha_{i'j}(\Delta\lambda_{1j})R_{i'} &= \frac{\sqrt{C_{i'j}(\Delta\lambda_{1j})}}{\sum_{j' \neq j} \sqrt{C_{i'j'}} + \sqrt{C_{i'j}(\Delta\lambda_{1j})}} R_{i'} \\ &\cong \frac{b}{a+b} R_{i'} + \frac{d\alpha_{i'j}(0)}{d\Delta\lambda_{1j}} \Delta\lambda_{1j} R_{i'} + \frac{d^2\alpha_{i'j}(0)}{d^2\Delta\lambda_{1j}} (\Delta\lambda_{1j})^2 R_{i'} + \dots \end{aligned}$$

where $a = \sum_{j' \neq j} \sqrt{C_{i'j'}}$, $b = \sqrt{C_{i'j}^0}$. The magnitude of the second order term is bounded as follows:

$$\begin{aligned} \left| \frac{d^2\alpha_{i'j}(0)}{d^2\Delta\lambda_{1j}} (\Delta\lambda_{1j})^2 R_{i'} \right| &= \frac{a}{2b^2} \left(\frac{1}{2b(a+b)^2} + \frac{1}{(a+b)^3} \right) (\Delta\lambda_{1j})^2 R_{i'} \\ &\cong \frac{a}{4(a+b)^2 b^3} (\Delta\lambda_{1j})^2 R_{i'} \leq \frac{R_{i'} (\Delta\lambda_{1j})^2}{4ab^3}. \end{aligned}$$

In our network settings, since the value of ab^3 in the denominator is much larger than the value of $R_{i'}$ in the nominator, it can be shown that all the higher order terms of $\alpha_{i'j}(\Delta\lambda_{1j})R_{i'}$ can be negligible and only the linear terms are significant. ■

Based on this, we define the linear belief function for the foresighted users.

³ $\Delta\lambda_{1j}$ can be set as the smallest difference of the foresighted user v_i 's traffic in channel r_j when v_i changing its belief parameters $\beta_{ij} \in \mathcal{B}_i$

Definition 6: Linear belief function for the foresighted user. The linear belief function on the remaining capacities of a foresighted user v_i can be expressed by a two-parameter linear function:

$$\tilde{C}_{ij}(\lambda_{ij}) = \beta_{ij}^{(0)} + \beta_{ij}^{(1)}\lambda_{ij}, \quad (10)$$

where $\beta_{ij} = [\beta_{ij}^{(0)}, \beta_{ij}^{(1)}] \in \mathcal{B}_i$ and $\mathcal{B}_i$ represents a finite set of positive parameters with $0 \leq \beta_{ij}^{(1)} < 1$, $0 \leq \beta_{ij}^{(0)} \leq C_j$. The condition $0 \leq \beta_{ij}^{(1)} < 1$ implies that when the foresighted user increases the traffic that it transmits through a certain channel λ_{ij} , the other myopic users will avoid using the same channel and move their traffic to other channels. This increases the expected remaining capacity $\tilde{C}_{ij}$ for the foresighted user v_i . In the next subsection, we provide a reinforcement learning method for the foresighted user v_i to learn these parameters $\beta_{ij} = [\beta_{ij}^{(0)}, \beta_{ij}^{(1)}]$ based on the measurement history o_i^t .

B. Linear regression learning to model the belief function

The foresighted user v_i repeatedly updates its belief function $\tilde{C}_{ij}(\lambda_{ij}) = \beta_{ij}^{(0)} + \beta_{ij}^{(1)}\lambda_{ij}$ at every time slot⁴. In this chapter, we make the foresighted user update the parameters $\beta_{ij}^t = [\beta_{ij}^{(0)t}, \beta_{ij}^{(1)t}] \in \mathcal{B}_i$ using the following update rule:

$$\begin{aligned} \beta_{ij}^t &= \arg \min_{\beta_{ij} \in \mathcal{B}_i} \|\beta_{ij} - \hat{\beta}_{ij}\|, \\ \text{where } \hat{\beta}_{ij} &= (1 - \rho_i) \beta_{ij}^{t-1} + \rho_i \tilde{\beta}_{ij}(o_{ij}^t) \end{aligned} \quad (11)$$

ρ_i is the learning rate, which determines how rapidly a user is willing to change its belief on the remaining capacities. $\tilde{\beta}_{ij}(o_{ij}^t) = [\tilde{\beta}_{ij}^{(0)}, \tilde{\beta}_{ij}^{(1)}]$ will be estimated based on the linear regression from the samples o_{ij}^t , where o_{ij}^t represents the latest S measured remaining capacities and input traffic pairs for a certain channel in o_i^t (i.e. $\{(C_{ij}^{t-k}, \lambda_{ij}^{t-k}), k = 1, \dots, S\}$). For this, we can adopt standard least square error linear regression [KSH00]. To estimate the error due to deploying a linear model, denote $e^{t-k}(\tilde{\beta}_{ij}, o_{ij}^t)$ as the residual error of the linear regression at time slot $t - k$. The mean residual error is then defined as

⁴ Different time scale can be applied for the foresighted users to make sure that the measured remaining capacities C_{ij}^t are the stable results of the other myopic users played in the game.

$$\bar{e}(\tilde{\beta}_{ij}, o_{ij}^t) = \frac{1}{S} \sum_{k=1}^S e^{t-k}(\tilde{\beta}_{ij}, o_{ij}^t).$$

Proposition 2: Reaching the ε - CE using the linear regression learning. When there is only one foresighted user, the linear regression learning results in ε - CE of the conjecture-based channel selection game with $\varepsilon \leq \max_{r_j} \{\bar{e}(\tilde{\beta}_{ij}, o_{ij}^t)\}$.

Proof: The foresighted user can determine an optimal action based on the linear belief function using the linear regression learning method. Given the optimal action of the foresighted user, the other myopic users will reach their NE equilibrium. If ε is selected as the worst case mean residual error, i.e. $\varepsilon \leq \max_{r_j} \{\bar{e}(\tilde{\beta}_{ij}, o_{ij}^t)\}$, the two conditions in Definition 5 are satisfied. Hence, such equilibrium will be the ε -CE. ■

In the simulation section, we also verify that the mean residual error for the belief function linearization is indeed very small, i.e. $\bar{e}(\tilde{\beta}_{ij}, o_{ij}^t) \leq \sum_{v_i \in \mathbf{V}^M} \frac{R_i (\Delta \lambda_{ij})^2}{4ab^3} \cong 0$, when there is only one foresighted user in the network.

Next, we discuss in more detail the ε -CE in two different cases: when the foresighted user is altruistic and when the foresighted user is self-interested.

C. Altruistic foresighted user

An altruistic foresighted user is usually the leader in a clustered network [CZ05], e.g. the access point in IEEE 802.11 network, or the routing leader in a hierarchical ad hoc network [Bel04]. An altruistic foresighted user will have an objective function that is aligned with the system goal, $U^{fair}(\sigma)$. As the foresighted user v_i 's belief $\tilde{C}_{ij}(\lambda_{ij})$ reflects the aggregate traffic distribution of all the other users $\sum_{v_i \in \mathbf{V}^M} \lambda_{ij}$, $U^{fair}(\sigma)$ can be rewritten as:

$$\begin{aligned} \sum_{j=1}^N \frac{\sum_{i=1}^M \lambda_{ij}}{C_j - \sum_{i=1}^M \lambda_{ij}} &= \sum_{j=1}^N \frac{C_j - \tilde{C}_{ij}(\lambda_{ij}) + \lambda_{ij}}{\tilde{C}_{ij}(\lambda_{ij}) - \lambda_{ij}} \\ &= \sum_{j=1}^N \frac{C_j - \beta_{ij}^{(0)} - \beta_{ij}^{(1)} \lambda_{ij} + \lambda_{ij}}{\beta_{ij}^{(0)} + \beta_{ij}^{(1)} \lambda_{ij} - \lambda_{ij}} \equiv U^{fair}(\sigma_i, \mathbf{B}_i(\sigma_i)) \end{aligned} \quad (12)$$

Then, the altruistic foresighted user v_i performs the following optimization:

$$\begin{aligned} & \underset{\sigma_i \geq 0}{\text{minimize}} U^{fair}(\sigma_i, \mathbf{B}_i(\sigma_i)) \\ & \text{s.t. } \sum_{j=1}^N \lambda_{ij} = x_i / L \end{aligned}, \quad (13)$$

while the rest of the myopic users adopt equation (7). Note that only the system-wise Pareto optimal solution on the Pareto boundary can be approached by the altruistic foresighted user⁵. For the other solutions on the Pareto boundary, the foresighted user needs to know the traffic rate x_i as well as the weights w_i of the other users. However, the foresighted user adopts a linear belief function in Section IV.B, which provides an imperfect belief by approximating the remaining capacities. There will be a performance penalty (gap) experienced by the foresighted user between the resulting ε -CE σ_{alt}^* and the system-wise Pareto optimal solution σ^P based on the user's perfect beliefs, which is defined as:

$$GAP(\sigma_{alt}^*, \sigma^P) = U^{fair}(\sigma_{alt}^*) - U^{fair}(\sigma^P). \quad (14)$$

Proposition 3: Reaching system-wise Pareto optimal solution when only one user is foresighted. When there is only one altruistic foresighted user v_i in the conjecture-based channel selection game, the gap between the resulting ε -CE σ_{alt}^* and the σ^P will be bounded by:

$$GAP(\sigma_{alt}^*, \sigma^P) \leq \sqrt{\varepsilon} \sum_{r_j \in \Omega_i} \frac{C_j}{(C_{ij,alt}^* - \lambda_{ij,alt}^*)^2}, \quad (15)$$

where Ω_i represents a set of channels whose $\lambda_{ij} > 0$.

Proof: Since the foresighted user can access all the channels, the foresighted user's action can directly influence all the other myopic users in the network. Since the foresighted user will approximate $\tilde{C}_{ij}(\lambda_{ij})$ to the actual remaining capacities to satisfy equation (9) at the ε -CE σ_{alt}^* , the worst case $\tilde{C}_{ij}^*(\sigma_i^*) \geq C_{ij}^*(\sigma_{alt}^*) - \varepsilon'$ ($\varepsilon' = \sqrt{\varepsilon}$) can be considered to bound the $GAP(\sigma_{alt}^*, \sigma^P)$. The worst case gap is bounded by

⁵ For the solution of equation (13) to approach the system-wise Pareto optimal solution, the ratio of the traffic rate of the foresighted user and the total traffic rate is required to be above a certain threshold, which is discussed in [KLO95]. In the following discussion, we assume that the ratio is above such threshold.

$$\begin{aligned}
GAP(\sigma_{alt}^*, \sigma^P) &\leq \sum_{r_j \in \Omega} \frac{C_j + \lambda_{ij}^* - C_{ij}^* + \varepsilon'}{C_{ij}^* - \lambda_{ij}^* - \varepsilon'} - \sum_{r_j \in \Omega} \frac{C_j + \lambda_{ij}^* - C_{ij}^*}{C_{ij}^* - \lambda_{ij}^*} . \quad \text{Let } K_{ij} = C_j + \lambda_{ij}^* - C_{ij}^* \quad \text{and} \\
J_{ij} &= C_{ij}^* - \lambda_{ij}^* . \text{ For a small } \varepsilon , \text{ the first term of the right hand side can be simplified as} \\
\sum_{r_j \in \Omega_i} \frac{K_{ij} + \varepsilon'}{J_{ij} - \varepsilon'} &\cong \sum_{r_j \in \Omega_i} \frac{K_{ij}}{J_{ij}} + \sum_{r_j \in \Omega_i} \frac{K_{ij} + J_{ij}}{(J_{ij})^2} \varepsilon' , \quad \text{and the gap will be bounded by} \\
GAP(\sigma_{alt}^*, \sigma^P) &\leq \varepsilon' \sum_{r_j \in \Omega_i} \frac{K_{ij} + J_{ij}}{(J_{ij})^2} = \varepsilon' \sum_{r_j \in \Omega_i} \frac{C_j}{(C_{ij}^* - \lambda_{ij}^*)^2} . \quad \blacksquare
\end{aligned}$$

In other words, the foresighted user is able to drive σ_{alt}^* to the system-wise Pareto optimal solution for an arbitrary small ε . Proposition 3 also implies that given the same total capacities, i.e. $\sum_{j=1}^N C_j$ is fixed, the uniform capacities among the frequency channels will result in a minimum gap from the ε - CE to the system-wise Pareto optimal solution.

D. Self-interested foresighted user

Note that reaching the system-wise Pareto optimal solution will not minimize the delay of the foresighted user itself (as will be shown in the Section VI). Thus, a self-interested foresighted user has no incentive to optimize the system-wise delay. Importantly, the foresighted users will have to sacrifice its own delay in order to minimize the system-wise delay. Hence, we now consider the case when the foresighted user is self-interested and only intends to minimize its own delay. If the foresighted user is self-interested, the objective function of the foresighted user is then minimizing

$U_i(\sigma_i, \mathbf{B}_i(\sigma_i)) = \frac{L}{x_i} \sum_{j=1}^N \frac{\lambda_{ij}}{\tilde{C}_{ij}(\lambda_{ij}) - \lambda_{ij}}$. Specifically, with the linear belief functions, the self-interested foresighted user v_i performs:

$$\begin{aligned}
&\underset{\sigma_i \geq 0}{\text{minimize}} \sum_{j=1}^N \frac{\lambda_{ij}}{\beta_{ij}^{(0)} + \beta_{ij}^{(1)} \lambda_{ij} - \lambda_{ij}} . \\
&\text{s.t.} \quad \sum_{j=1}^N \lambda_{ij} = x_i / L
\end{aligned} \tag{16}$$

The following proposition provides the optimal action for the self-interested foresighted user.

Proposition 4: Solution of the self-interested foresighted user

Given the belief of the remaining capacity $\tilde{C}_{ij}(\lambda_{ij}) = \beta_{ij}^{(0)} + \beta_{ij}^{(1)}\lambda_{ij}$, with $0 \leq \beta_{ij}^{(1)} < 1$, $0 \leq \beta_{ij}^{(0)} \leq C_j$, the optimal action that minimizes U_i for the foresighted user to transmit on channel r_j is $a_{ij}^* = \lambda_{ij}^* L / x_i$,

$$\lambda_{ij}^* = \max\{0, \tilde{D}_{ij} - \alpha_{ij}^{(f)} \left(\sum_{r_j \in \Omega_i} \tilde{D}_{ij} - x_i / L \right)\}, \quad (17)$$

where $\tilde{D}_{ij} = \beta_{ij}^{(0)} / (1 - \beta_{ij}^{(1)})$. The portion $\alpha_{ij}^{(f)}$ now becomes $\kappa_{ij} / \sum_{r_j \in \Omega} \kappa_{ij}$, where $\kappa_{ij} = \sqrt{\beta_{ij}^{(0)}} / (1 - \beta_{ij}^{(1)})$ and Ω_i represents the channels whose $\lambda_{ij} > 0$.

Proof: See Appendix A.

While the other users are myopic, the best performance from the self-interested foresighted user's perspective is to achieve the Stackelberg Equilibrium (SE) σ^S [FL98]. Note that if the foresighted user is able to build a perfect belief on the remaining capacities (i.e. $\varepsilon = 0$), the resulting conjectural equilibrium is the same as the SE of the game, since the foresighted user knows the exact reactions of the myopic users. Hence, we use the SE σ^S instead of the system-wise Pareto optimal solution σ^P to benchmark the self-interested foresighted user. Denoting the solution in Proposition 3 as σ_{self}^* , the corresponding performance gap is defined as $GAP(\sigma_{self}^*, \sigma^S) = U_i(\sigma_{self}^*) - U_i(\sigma^S)$.

Proposition 5: Reaching SE when only one user is foresighted. When there is only one self-interested foresighted user v_i in the conjecture-based channel selection game, the gap between the resulting ε - CE and the SE will be bounded by:

$$GAP(\sigma_{self}^*, \sigma^S) \leq \sqrt{\varepsilon} \sum_{r_j \in \Omega_i} \frac{1}{(C_{ij, self}^* - \lambda_{ij, self}^*)^2}, \quad (18)$$

where Ω_i represents a set of channels whose $\lambda_{ij} > 0$.

Proof: The gap can be shown to be bounded using a similar proof as Proposition 3. Note that the foresighted user is now minimizing its own delay instead of U^{fair} in Proposition 3. Hence, the $GAP(\sigma_{self}^*, \sigma^S)$ is calculated with respect to the foresighted user v_i 's delay U_i , and the resulting upper bound changes accordingly.

In other words, the foresighted user is able to drive the ε - CE σ_{self}^* to the SE σ^S for an arbitrary small ε . Proposition 4 provides the optimal channel selection of the self-interested foresighted user v_i when applying a linear belief function as described in equation (10) and Proposition 5 implies that the performance of the foresighted user at ε - CE can be as good as the SE when the self-interested foresighted user can approximate the future remaining capacities. In Appendix C, Algorithm 8.1 provides the channel selection algorithm that will be followed by the self-interested foresighted user. An illustrative example is given in Figure 8.3 for the solutions introduced in Section IV.C and IV.D in 2-user case (v_i is the foresighted user and v_{-i} is the myopic user). Note that the SE σ^S provides a smaller delay compared to σ^P for the foresighted user v_i at the cost of increasing the delay of the myopic user. This is because it selfishly minimizes its own delay given that it knows the reaction of the other user, which is the best that a self-interested foresighted user can achieve.

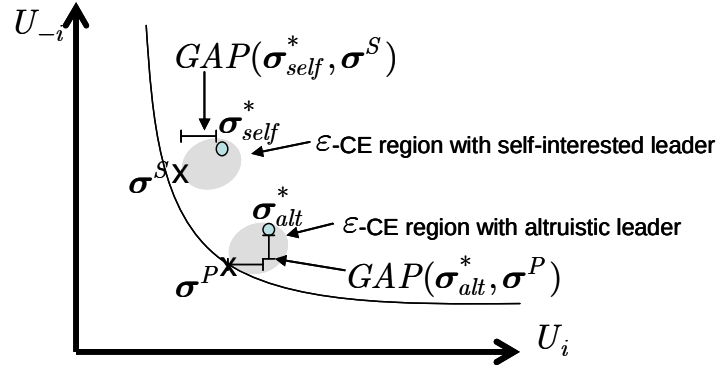


Fig. 8.3 An illustrative example of the solutions in the utility domain for a 2-user case (v_i is the foresighted user).

V. DISTRIBUTED CHANNEL SELECTION WHEN THERE ARE MULTIPLE FORESIGHTED USERS

A. Performance degradation when multiple users learn

In this section, we investigate the case when multiple users are foresighted. Unlike in Section IV, the coexistence of the multiple foresighted users now complicates the prediction of the other users' reaction (not only the myopic users). The linear belief function in Section IV cannot accurately model the aggregate response of the other users. Without a valid belief function, the ε - CE σ^* does not exist. This is because the other foresighted users will continuously modify their decisions and thus, the condition in equation (9), which is necessary for reaching the ε - CE, will not be satisfied. Such autonomous learning solution can result in users' utility degradation as well as a system utility degradation as shown in [WH98], and it is also illustrated in our simulation results in Section VI. We quantify the performance degradation of a wireless system where foresighted users are autonomously learning using a time average gap $GAP([\sigma^t, t = 1, \dots, T], \sigma^P)$ to the system-wise Pareto optimal solution, which is defined as:

$$\begin{aligned} & GAP([\sigma^t, t = 1, \dots, T], \sigma^P) \\ &= \frac{1}{T} \sum_{t=1}^T \sum_{i=1}^M x_i U_i(\sigma^t) - U^{fair}(\sigma^P), \end{aligned} \tag{19}$$

where σ^t represents the traffic distribution of users at time slot t . We substitute the first term in equation (14) by the time average delay of all the users in the network. This is because in this setting, there is no guarantee that the system will converge to an equilibrium.

To close the gap to the system-wise Pareto optimal solution, these foresighted users cannot form their own beliefs independently [WH98]. Instead, they need to obey the coordination rules prescribed by MAC protocol to specify their beliefs. Such rules can be collaborative solutions derived from solving the centralized optimization in equation (4) or using a NUM-type framework [WZQ08]. In this chapter, we assume that a self-interested

foresighted user will *only* choose to comply with such a prescribed rule when doing so is more beneficial (in terms of delay) for itself than when not doing so. We will find the sufficient condition for the collaboration/coordination among the users to be self-enforcing. Next, we show how the foresighted users can collaboratively build beliefs according to the rules to reach the system-wise Pareto optimal solution.

B. Reaching system-wise Pareto optimal solution when every user builds belief using a prescribed rule

In this subsection, we propose an alternative rule-based belief function for the foresighted users, other than the linear regression learning method proposed in Section IV.B. We prove that this rule-based belief enables the foresighted users to reach the system-wise optimal solution in a distributed manner, based on their local information.

Proposition 6: Rule-based solution that reaches the system-wise Pareto optimal solution.

A family of belief functions $\mathcal{B}_i^* = \{\beta_{ij}^*\} \subseteq \mathcal{B}_i$ leads to the rule-based solution $\sigma_{rule}^* = \{\lambda_{ij}^*, i = 1, \dots, M, j = 1, \dots, N\}$, where $\lambda_{ij}^* = \max\{0, C_{ij} - \frac{\sqrt{C_j}}{\sum_{r_j \in \Omega_i} \sqrt{C_j}} R_i\}$.

This solution satisfies the optimality conditions of minimizing $U^{fair}(\sigma)$ and results in $GAP(\sigma_{rule}^*, \sigma^P) = 0$.

Proof: See Appendix B.

A straightforward example for the belief functions in Proposition 6 can be $\beta_{ij}^{*(0)} = (C_{ij})^2 / C_j$, $\beta_{ij}^{*(1)} = 1 - C_{ij} / C_j$ for $\forall v_i \in \mathbf{V}$. By forcing the users to use this $\beta_{ij}^* = [\beta_{ij}^{*(0)}, \beta_{ij}^{*(1)}]$ ⁶, the rule-based solution σ_{rule}^* can be obtained by the users in one iteration based on their current remaining capacities C_{ij} .

So far, two different approaches are provided for a foresighted user to build belief functions: 1) using $\beta_{ij}^t = [\beta_{ij}^{(0)t}, \beta_{ij}^{(1)t}]$ that applies the linear regression learning in equation

⁶ Such rule-based solution is not a equilibrium of the channel selection game. It is derived as an optimal rate allocation using the utility function defined in this chapter, where all users prefer to minimize the delays experienced from different queues. With other types of utility functions (e.g. users may have conflict of interests), such rule-based solution may not be derivable.

(11) and 2) using $\beta_{ij}^* = [\beta_{ij}^{*(0)}, \beta_{ij}^{*(1)}]$ that applies the rule-based solution. The reason why we consider these two approaches is because of their low computational complexities and because they act only on the local information. The first approach allows the foresighted users to learn their beliefs using a linear model, which has only two parameters, and the linear regression of these two parameters can be easily performed. The second approach can reach the system-wise Pareto optimal solution by appropriately providing these two parameters to the foresighted users. Importantly, there are two differences between these two approaches:

- a) The first approach allows the foresighted users to build their beliefs about the *aggregate response of the other users* (C_{ij} in this chapter) based on only local information. However, the second approach builds the beliefs for users to follow the *optimal rate allocation* (in the sense of manipulating users' utility functions, and done in the NUM-type approaches in [WZQ08]), which results in minimizing the system's utility.
- b) As discussed in the previous subsection, the first approach is not suitable for the scenario when multiple foresighted users build their beliefs simultaneously, because the linear belief is no longer valid. The resulting delay minimization is inefficient. On the contrary, applying the second approach is efficient, but only when all users comply with the rule-based solution. Hence, it is important to investigate the incentives for the foresighted users to comply with the rule-based solution.

To do this, we first consider the case where a new user joins a network and the other users present in the network are already complying with the rules (choosing β_{ij}^*). The following condition ensures that no users will have incentive to deviate from the rule-based solution.

Proposition 7: Sufficient condition (incentive) for users to comply with the rule-based solution. When all users in the network are foresighted, no users will deviate from the rule-based solution σ_{rule}^* , if the users are sharing all the channels, i.e. $\lambda_{ij} > 0, \forall v_i, \forall r_j$.

Proof: When the other users select the rule-based solution and all the channels are shared by all the users (i.e. $\lambda_{ij} > 0, \forall v_i, \forall r_j$), the fraction $\alpha_{ij} = \frac{\sqrt{C_{ij}}}{\sum_{r_j \in \Omega_i} \sqrt{C_{ij}}}$ in the user's best response (see equation (6)) will coincide with $\frac{\sqrt{C_j}}{\sum_{r_j \in \Omega_i} \sqrt{C_j}}$. Hence, the rule-based solution σ_{rule}^* is the best response for user v_i , for $\forall v_i \in \mathbf{V}$ when the other users select the rule-based solution. ■

In a more general case where some of the users will deviate from the rule-based solution (when the condition in Proposition 7 is not satisfied), the foresighted users will then select the alternative linear regression learning method $\beta_{ij}^t = [\beta_{ij}^{(0)t}, \beta_{ij}^{(1)t}]$ to build their beliefs and minimize autonomously their own delays, and end up having the undesirable gap in equation (19). This is similar to the prisoner's dilemma game [FT91], where users also have two actions. In this case, an on-line coordination procedure will be needed for the foresighted users to discover the benefit of using the rule-based solution σ_{rule}^* , which will be discussed in the next subsection. Importantly, message exchanges among the users are still informationally inefficient and thus, undesirable, for the on-line procedure. Instead of allowing the foresighted users to directly reveal their willingness to comply with the rule-based solution, the on-line procedure only allows the foresighted users to test and conjecture on the willingness of the other users by observing their own delay performance, which can be computed based on their local information.

C. On-line coordination of the foresighted channel selection

We now discuss the on-line coordination procedure from a self-interested foresighted user's point of view. First, a foresighted user needs to identify whether or not it is the only foresighted user in the network, in which case it should build its own belief by using the linear regression learning method β_{ij}^t by applying the self-interested channel selection in Algorithm 8.1. Second, if there are multiple foresighted users in the network, the foresighted user needs to identify whether or not it should comply with the rule-based

belief β_{ij}^* to enforce the rule-based solution σ_{rule}^* . We propose the following procedures for the self-interested foresighted user to identify these two conditions in two stages:

a) Stage 1: identify whether or not it is the only foresighted user in the network. To identify whether or not the foresighted user is the only foresighted user present in the network, we propose a probing approach to conjecture the existence of any other foresighted users. Specifically, the foresighted user can deliberately deplete a certain frequency channel r_j (thereby making the remaining capacity equal to $C_j - \lambda_{ij} = \delta$, where $\delta \ll 1$) and observe the change in the remaining capacity C_{ij} . Depending on whether the other users are foresighted or myopic, they will react to this action differently. The myopic user will immediately avoid this channel, since the small remaining capacity of that channel is undesirable for them to minimize their delays (since $C_{ij}^{t-1} = \delta$ in equation (7)). If there exist any foresighted users $v_{i'}$ in the network, its $\lambda_{i'j}^*$ will be determined according to the belief function parameter $\beta_{i'j}^t$. Note that for a foresighted user, the parameters $\beta_{i'j}^t$ will not immediately react to the channel depletion due to the learning rate $\rho_{i'}$ in equation (11). Hence, the foresighted user $v_{i'}$ will still put traffic into the channel r_j , i.e. $\lambda_{i'j}^* > 0$. By examining the change of the subsequent remaining capacity C_{ij} , user v_i can conjecture whether or not there are other foresighted users in the network. Note that a foresighted user can test this condition when it first joins the network, and we assume that the probability that more than two foresighted users simultaneously join the network is very small.

b) Stage 2: determine whether or not to comply with the prescribed rule. In the second stage, if there are multiple foresighted users in the network, the foresighted user should also identify whether or not complying with the rule-based solution can result in a better delay performance for itself. Of course, the foresighted user would like to build its belief autonomously to minimize its own delay. However, the autonomous learning solution can result in undesired performance degradation when there are multiple foresighted users in the network [WH98]. Hence, the coordinated rule-based solution can be a better choice

for the foresighted users. This is similar to the situation arising in the Prisoners' dilemma, where users will align their actions to maximize their payoffs if they can coordinate with each other [FT91]. However, as discussed in Section V.B, if some of the foresighted users deviate from the prescribed rule, the performance degradation can once again provide incentives for these foresighted users to not comply with the rule. Hence, in this stage, the foresighted users test the willingness of each other for complying with the rule-based solution σ_{rule}^* . Following the procedure, the foresighted users perform the rule-based solution σ_{rule}^* at the same time (immediately after a predetermined T time slots⁷). A foresighted user will be willing to comply with the rule-based solution only if the resulting delay performance $U_i(\sigma_{rule}^*)$ is better than the time average delay $U_i^{TA} = \frac{1}{T} \sum_{t=1}^T U_i(\sigma^t)$ during the first T time slots, when the foresighted user selfishly performs Algorithm 8.1 to optimize its own utility.

The detailed steps of the proposed procedure are provided in Algorithm 8.2. This algorithm allows the foresighted user to test the resulting performance of different channel selection solutions without any message exchange to notify what types of users it is interacting with. These other users can be myopic users (who just myopically react to the latest remaining capacity measurement) or other self-interested foresighted users who do not want to comply with the coordinated rule-based solution (e.g. users who are not satisfied with their resulting delays when complying with the prescribed rules). The algorithm provides a method for the foresighted users who are willing to apply this algorithm to discover which solution (the rule-based solution or the self-interested solution in Algorithm 8.1) leads to the best performance in terms of their experienced delay. Figure 8.4 provides a flowchart of the proposed on-line procedure. The procedure can be deployed periodically at the beginning of each MAC super-frame [IEE03] in current standards to test the abovementioned two conditions. Then, during the super-

⁷ Such T time slots directly related to the duration of the test procedures. The small T cannot guarantee that the time average delay U_i^{TA} is representative enough. On the other hand, the large T results in a large protocol overhead for the test procedures, which can be undesirable in the case where all users prefer coordinated rule-based solution.

frame, users can transmit their data using a particular solution (i.e. the coordinated rule-based solution or the selfish foresighted solution). The protocol overhead of the procedure can be set relatively small compared to the data transmission period in a MAC super-frame.

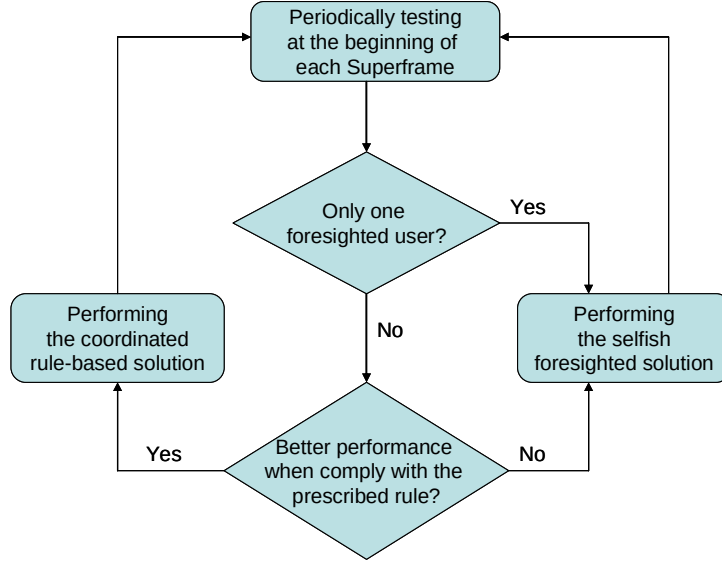


Fig. 8.4 Flowchart of the on-line foresighted channel selection procedure.

VI. SIMULATION RESULTS

In this section, we simulate the conjecture-based channel selection game in two network settings, which are shown in Table 8.1. We assume an asymmetric network where the capacities of the channels are $W_1 = 8$ Mbps and $W_i = 2$ Mbps, $i = 2, \dots, N$. The users are assumed to have traffic with Poisson arrival rates $x_1 = 3.8$ Mbps, $x_i = 0.6$ Mbps, $i = 2, \dots, M$. The average packet length is $L = 1000$ bits.

TABLE 8.1. CONSIDERED NETWORK SETTINGS.

Network setting	Number of channels N	Number of users M	Total channel capacities (Mbps)	Total traffic rates (Mbps)
1 (Large network)	10	30	26	21.2
2 (Small network)	2	8	10	8

A. Single foresighted user scenario

We first simulate the case when there is only one foresighted user. User v_1 is assumed to be the foresighted user, and the rest of the users are myopic users. Figure 8.5(a) shows the evolution of user v_1 's action $\mathbf{a}_1$ (i.e. its channel selection probabilities) until the system reaching the NE in network setting 1 (the large network). Since channel r_1 has a larger capacity, more traffic will be distributed to channel r_1 than to the other channels. Using the learning method proposed in Section IV.B, the foresighted user v_1 can determine its belief functions on the remaining capacities. The circles in Figure 8.5(b) represent the measured remaining capacities C_{11} at different channel selection probability a_{11} (the samples o_1^t). The solid line represents the resulting linear regression. The resulting parameters of the linear belief function are $\beta_{11} = [0.375, 4962]$. The residual mean square error is 0.051 and the computed bound is $\sum_{v_i \in \mathbf{V}^M} \frac{R_i (\Delta \lambda_{ij})^2}{4ab^3} \cong 10.85$, which are in agreement with Proposition 1. Figure 8.5(c) shows similar results in channel r_2 . Similarly in network setting 2 (the small network), Figure 8.5(d) shows again the evolution of $\mathbf{a}_1$ in a network. The channel selection converges faster in this setting, since the number of users is smaller. The resulting parameters of the linear belief function are $\beta_{11} = [0.52, 4718]$. The residual mean square error is 0.012 and the computed bound is $\sum_{v_i \in \mathbf{V}^M} \frac{R_i (\Delta \lambda_{ij})^2}{4ab^3} \cong 4.34$, which are again in agreement with Proposition 1. Based on the linear belief functions, user v_1 can perform the foresighted channel selection.

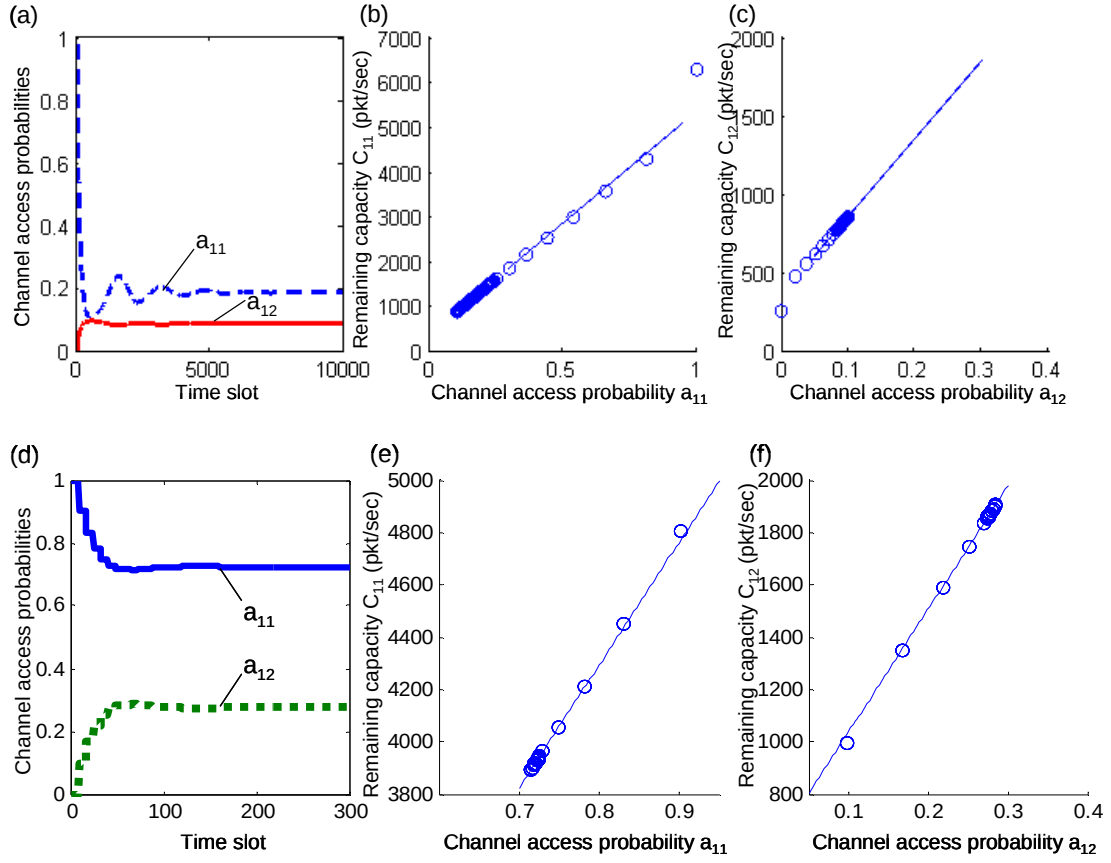


Fig. 8.5(a)(d) The action of the foresighted user v_1 over time, while participating in the channel selection game [(a) in network setting 1, (d) in network setting 2].
 (b)(c)(e)(f) The actual remaining capacity C_{1j} and the estimated linear belief function $\tilde{C}_{1j}$, $j = 1, 2$ [(b)(c) in network setting 1, (e)(f) in network setting 2].

In order to show clearly the intuition behind the foresighted channel selection, we now focus on the small network setting. Figure 8.6 shows the utility domain in terms of delay. The x-axis is the delay of the foresighted user and the Y-axis is the average delay of the myopic users. By using the belief function, the simulation results show that the altruistic foresighted user is able to drive the system from the (system) inefficient NE to the system-wise Pareto optimal solution (in which the system queue size U^{fair} is minimized)

by using the belief function. If the foresighted user is selfish, it will drive the system from NE to SE. Table 8.2 shows the results at different equilibriums. When the foresighted user is selfish, it puts more traffic into the efficient channel τ_1 and forces the other myopic users to select the other channel, thereby benefiting its own utility. On the contrary, if the foresighted user is altruistic, it puts less traffic into channel τ_1 and allows the other users myopically select the efficient channel τ_1 , which will result in an optimal system performance.

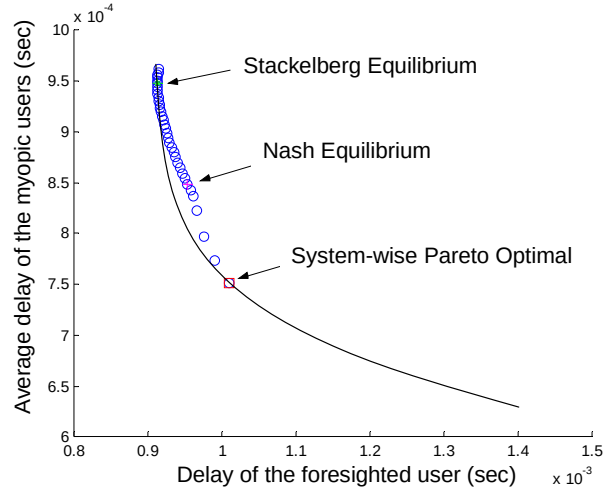


Fig 8.6. Reaching the system-wise Pareto optimal solution and the Stackelberg Equilibrium.

TABLE 8.2. RESULTS AT DIFFERENT EQUILIBRIUMS

	Action of the foresighted user a_{11}	Action of the myopic user a_{i1}	Delay of the foresighted user	Average delay of the myopic users	System Performance
NE	0.72	0.97	0.955 ms	0.848 ms	7.19
SE	0.95	0.78	0.914 ms	0.947 ms	7.45
System-wise optimal	0.66	1	1.011 ms	0.752 ms	7.00

Next, we highlight the impact in terms of delay for the foresighted user and the myopic users, when different numbers of myopic users are active in the network. Figure 8.7 shows the delay of the foresighted user at different equilibriums when there are various

numbers of myopic users in the network. The results show that, as the number of myopic users in the network increases, the altruistic foresighted user will have a higher delay impact to reach the system-wise Pareto optimal solution. Beyond 10 myopic users, the system-wise Pareto optimal solution is not reachable. This situation is also observed in network setting 1 (large network setting). This is because the traffic ratio of the foresighted user to the total traffic in the network is not sufficient enough to drive the equilibrium to the system-wise Pareto optimal solution (as discussed in [KLO95]). On the contrary, the foresighted user can benefit more in terms of delay when the number of the myopic users in the network increases.

B. Multiple foresighted user scenario

In this subsection, we simulate the result when there are multiple foresighted users in the network. We simulate the resulting delays of the conjecture-based channel selection game using the small network setting in the previous subsection. The only difference is that we now assume that the 8 users all have traffic with Poisson arrival rate $x_i = 1$ Mbps. Hence, the total traffic rate is still 8 Mbps. These users can select three different channel selection solutions: 1) the rule-based solution (RB) in Section V.B, 2) the self-interested

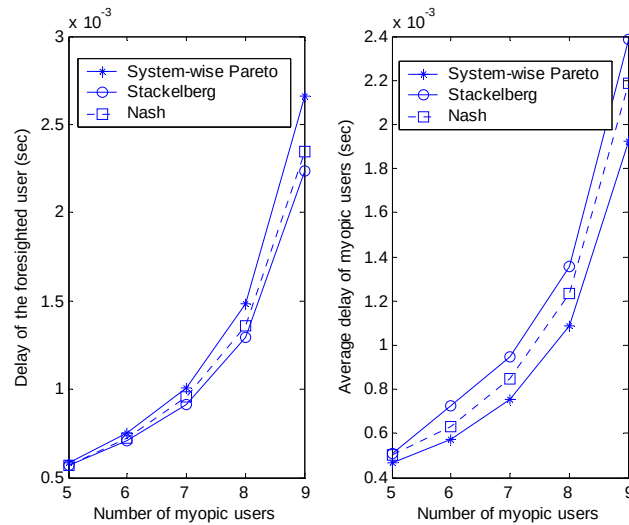


Fig. 8.7 Delay of the foresighted user at different equilibrium for various numbers of myopic users in the network.

foresighted solution (SF) in Section IV.D, and 3) the myopic solution (MY) in Section II.C. We discuss 8 different scenarios in Table 8.3. First, we simulate the case when all users are myopic (scenario 1). As simulated in the previous subsection, a self-interested foresighted user can have a smaller delay when the rest of the users are myopic. However, when the number of these self-interested foresighted users is larger than 3, the average delay of these selfish foresighted users can be even worse than the average delay which they experience when they adopt a myopic channel selection strategy. Hence, this gives incentives for these foresighted users to collaborate with each other by adhering to the proposed Algorithm 8.2, which allows the users to test the rule-based solution. The rule-based solution (scenario 5) provides the minimum average delay for all the foresighted users and the minimum queue size of the system (minimum U^{fair}). However, we can see that, once a selfish user deviates from the rule, both the delay of the selfish user as well as the system queue size U^{fair} increase (scenario 6). Thus, if a foresighted user joins a network where the other users already comply with the rule-based solution, the users should collaborate with each other for their own benefit. Hence, their collaboration is self-enforcing rather than mandated by a protocol designer. Moreover, from scenario 3 and 8, we see that even when the rest of the users are myopic, the 3 foresighted users will still have incentive to perform collaborated rule-based solution. However, the delay performance degrades seriously when some foresighted users deliberately deviate from the prescribed rules (we set 2 users to select SF in scenario 7). In this case, these foresighted users will not have incentives to comply with the rule-based solution anymore. They will all become self-interested and perform Algorithm 8.1 (as in scenario 4).

TABLE 8.3. NUMERICAL RESULTS IN DIFFERENT SCENARIOS

Scenarios	Number of users using different solutions	Average delay of the foresighted users (ms)	Normalized system queue size ($U^{fair}/\text{total traffic rate}$)	GAP to the optimal system performance
1	All MY	0.90	0.9	0.025
2	1 SF, 7 MY	0.852	0.91	0.035
3	3 SF, 5 MY	0.877	0.918	0.043
4	5 SF, 3 MY	0.933	0.953	0.078
5	All RB	0.80	0.875	0
6	1 SF, 7RB	1.00	1.00	0.125
7	3 RB, 2 SF, 3 MY	1.164	1.164	0.289
8	3 RB, 5 MY	0.864	0.911	0.034

VII. CONCLUSIONS

In this chapter, we study the distributed channel selection problem in multi-channel wireless networks. Although we use a multi-channel wireless network setting, it is important to note that the proposed method can be applied to other load balancing resource sharing system. We model the multi-user interaction using a conjecture-based channel selection game where myopic users and foresighted users coexist in the network. Based on the analysis of the conjecture-based channel selection game, we investigate two different operation scenarios. In the single foresighted user scenario, we find that achieving the Pareto-efficient solution is possible without any message exchanges among users, as long as the foresighted user is not selfish. In the scenario where multiple users are foresighted, we show that the resulting performance degrades when users are learning in an autonomous manner. Hence, we discuss a rule-based solution for the foresighted users to collaboratively build the conjectures that optimize the system queue size in this chapter. In order to benefit themselves, these foresighted users can either build their own conjectures autonomously, based on their local information, or they can comply with a prescribed rule-based solution. We propose an on-line procedure for the foresighted users to select a solution to minimize their delay. The results show that in such multi-channel network, delay-sensitive users can minimize their delays if there is only one self-interested foresighted user managing the network. If multiple foresighted users are

present in the network, they benefit from complying with the rule prescribed by the MAC protocols.

VIII. APPENDIX A

Proof of Proposition 4. First, we see that the objective function is a convex function, given that $0 \leq \beta_{ij}^{(1)} \leq 1$, $\beta_{ij}^{(0)} \geq 0$. Assume μ as the Lagrange multiplier. For $\forall r_j \in F_i$, the optimality conditions:

$$\frac{\beta_{ij}^{(0)}}{(\beta_{ij}^{(0)} + \beta_{ij}^{(1)}\lambda_{ij} - \lambda_{ij})^2} = \mu \Rightarrow \lambda_{ij} = \tilde{D}_{ij} - \sqrt{\frac{1}{\mu}}\kappa_{ij}. \quad (20)$$

From the constraint $\sum_{j=1}^N \lambda_{ij} = x_i$, we have

$$\sqrt{1/\mu} = \left(\sum_{r_j \in \Omega} \tilde{D}_{ij} - x_i / L \right) / \sum_{r_j \in \Omega} \kappa_{ij}. \quad (21)$$

By substituting equation (21) into equation (20), we have $\lambda_{ij} = \tilde{D}_{ij} - \alpha_{ij}^{(f)} \left(\sum_{r_j \in \Omega} \tilde{D}_{ij} - x_i / L \right)$ for $\lambda_{ij} > 0$ case. ■

IX. APPENDIX B

Proof of Proposition 6 Denote the total traffic through r_j as $\lambda_j = \sum_{i=1}^M \lambda_{ij}$. Assume $\mu = [\mu_i, i = 1, \dots, M]$ as the Lagrange multipliers. The Lagrange function of equation (4) can be written as:

$$\mathcal{L}(\sigma, \mu) = \sum_{j=1}^N \frac{\sum_{i=1}^M \lambda_{ij}}{C_j - \sum_{i=1}^M \lambda_{ij}} + \sum_{i=1}^M \mu_i (x_i / L - \sum_{j=1}^N \lambda_{ij}). \quad (22)$$

For those $\lambda_{ij} > 0$, the optimality conditions are:

$$\frac{C_j}{(C_j - \lambda_j)^2} = \mu_i \Rightarrow \lambda_j = C_j - \sqrt{\frac{C_j}{\mu_i}}, \text{ for } \forall v_i \in \mathbf{V}. \quad (23)$$

Since we assume the non-saturated condition, the condition $\sum_{j=1}^N \lambda_j = \sum_{i=1}^M x_i / L$ holds.

Based on this, we can calculate the Lagrange multipliers:

$$\sqrt{\frac{1}{\mu_i}} = \frac{(\sum_{r_j \in \Omega} C_j - \sum_{r_j \in \Omega} \lambda_j)}{\sum_{r_j \in \Omega} \sqrt{C_j}}, \text{ for } \forall v_i. \quad (24)$$

Hence, the Pareto optimum solution will be:

$$\lambda_j^* = C_j - \frac{\sqrt{C_j}}{\sum_{r_j \in \Omega} \sqrt{C_j}} (\sum_{r_j \in \Omega} C_j - \sum_{r_j \in \Omega} \lambda_j). \quad (25)$$

From the given $\beta_{ij}^{*(0)} = \frac{(C_{ij})^2}{C_j}$, $\beta_{ij}^{*(1)} = 1 - \frac{C_{ij}}{C_j}$, we have $\tilde{D}_{ij} = C_{ij}$, $\kappa_{ij} = \sqrt{C_j}$ (see the definitions in Proposition 4). We see that $\lambda_{ij}^* = \max\{0, C_{ij} - \frac{\sqrt{C_j}}{\sum_{r_j \in \Omega} \sqrt{C_j}} R_i\}$

is realized for all users. Then,

$$\begin{aligned} \sum_{i=1}^M \lambda_{ij}^* &= \sum_{v_i \in \Psi} \left(C_{ij} - \frac{\sqrt{C_j}}{\sum_{r_j \in \Omega} \sqrt{C_j}} R_i \right) \\ &= \sum_{v_i \in \Psi} C_{ij} - \frac{\sqrt{C_j}}{\sum_{r_j \in \Omega} \sqrt{C_j}} \left(\sum_{v_i \in \Psi} (\sum_{r_j \in \Omega} C_{ij} - x_i / L) \right), \end{aligned} \quad (26)$$

where Ψ represents a set of users whose $\lambda_{ij}^* > 0$. Denote $P = |\Psi|$ as the size of this set.

Then equation (26) can be viewed as:

$$\begin{aligned} \lambda_j &= \sum_{i=1}^M \lambda_{ij}^* = PC_j - P\lambda_j + \lambda_j \\ &\quad - \frac{\sqrt{C_j}}{\sum_{r_j \in \Omega} \sqrt{C_j}} \left(\sum_{r_j \in \Omega} PC_j - \sum_{r_j \in \Omega} P\lambda_j + \sum_{r_j \in \Omega} \lambda_j - \sum_{v_i \in \Psi} x_i / L \right). \\ &\Rightarrow \lambda_j = C_j - \frac{\sqrt{C_j}}{\sum_{r_j \in \Omega} \sqrt{C_j}} (\sum_{r_j \in \Omega} C_j - \sum_{r_j \in \Omega} \lambda_j) = \lambda_j^* \end{aligned} \quad (27)$$

Hence, we showed that the solution is the Pareto optimal solution. ■

X. APPENDIX C

Algorithm 8.1 Self-interested foresighted channel selection

For user v_i at time slot t

Initialization: Set $t = 1$, $C_{ij}^0 = C_j$

Step 1. For all channel r_j , measure the remaining capacity C_{ij}^{t-1} and record it to memory o_{ij}^t .

Step 2. Update β_{ij}^t .

Calculate $\tilde{\beta}_{ij}$ using least square error linear regression from samples $o_{ij}^t = \{(C_{ij}^{t-k}, \lambda_{ij}^{t-k}), k = 1, \dots, S\}$. Then set $\beta_{ij}^t \in \mathcal{B}_i$ as in equation (11).

Step 3. Calculate the self-interested foresighted channel solution $\sigma_{self}^t = [\lambda_{ij}^t, j = 1, \dots, N]$.

λ_{ij}^t is calculated according to equation (17).

Step 4. Find the $\mathbf{a}_i^t = \sigma_{self}^t L / x_i$.

Step 5. $t \leftarrow t + 1$, and go back to Step 1.

Algorithm 8.2 On-line procedure of the foresighted channel selection

For a self-interested foresighted user v_i in the test period

Stage 1: Identify whether or not there is only one foresighted user.

Set $\lambda_{ij} = C_j - \delta$ to deplete the channel r_j .

Measure the subsequent remaining capacity $\tilde{C}_{ij}$.

If $\tilde{C}_{ij} = \delta$, there is only one foresighted user. Apply Algorithm 8.1 in the data transmission period.

Otherwise, go to Stage 2.

Stage 2: Identify whether or not to comply with the rule-based solution

Step 1. Perform Algorithm 8.1 for T time slots and measure the time average delay U_i^{TA} .

Step 2. Perform the rule-based solution $\sigma_{rule}^* = [\lambda_{ij}^*, j = 1, \dots, N]$ as in Proposition 6. Measure the resulting delay $U_i(\sigma^*)$.

Step 3. Compare the utilities. If $U_i(\sigma^*) \geq U_i^{TA}$, keep using the rule-based solution σ_{rule}^* in the data transmission period. Otherwise, apply Algorithm 8.1.

Chapter 9

Conclusions

This dissertation focused on developing mathematical tools, theoretical and statistical analysis, and algorithms, to understand, improve, and preserve the performance of delay-sensitive applications over multi-hop wireless networks. Current networks are primarily designed to communicate delay-insensitive information— they are not designed to handle delay-sensitive traffic. The goal of this dissertation was to study and propose mechanisms that endow wireless networks with the ability to reconfigure and adjust priorities in response to a time-varying network environment in order to deliver desired levels of performance for delay-sensitive applications. System-theoretic and optimization tools, as well as advanced networking, routing, communications, learning and game-theoretic concepts were integrated in order to model and control the behavior of complex interconnected wireless networks.

Most existing networking research aims at improving the goodput and robustness to various attacks or vulnerabilities of multi-hop wireless networks by redesigning existing transport, network or MAC protocols. However, such solutions are only successful if these new protocols are widely adopted by both international standardization bodies and industry. Also, such solutions often demand backwards compatibility and interoperability with existing infrastructures. In contrast, the research in this dissertation focused on improving the resilience and robustness of wireless networks, with minimal or no change to existing protocols and network infrastructure. This objective was accomplished as follows.

In order to fulfill stringent requirements of communicating delay-sensitive traffic over a multi-hop network, nodes in a network should be able to learn the behavior and status

of neighboring nodes and, subsequently, self-organize and adapt their cross-layer strategies in order to maximize the performance metrics and improve network performance. The foundations in this dissertation therefore rely on recognizing and exploiting the following three essential features: **(A)** the need to endow nodes in a network with multi-agent interactive learning abilities; **(B)** the need for these cognitive nodes to continuously adjust their operation in response to a dynamically changing (as opposed to static or stationary) network environment, and **(C)** the need for the individual nodes to adjust their own expectations in response to the state of the network by assigning different risk levels to different portions of their data. The discussion in the sequel expands on features (A)-(C), which were developed in this dissertation.

(A) Multi-agent interactive learning. To begin with, the individual nodes in a wireless network must be endowed with *adaptation and learning abilities* in order to be able to assess the conditions in their neighborhood on various levels such as (1) identifying nodes that have been compromised due to node failure, or other effects, (2) identifying node behavior and their history of resource usage, and (3) identifying nodes with poor signal-to-noise or power conditions.

By adding a cognitive dimension to the nodes, a wireless network becomes better enabled to evaluate the expected responses of neighboring nodes to various types of interference. They also become better prepared to support delay-sensitive applications. The “cognition” (i.e., learning of the environment and the actions from competing nodes) will allow the network nodes to dynamically self-organize and strategically adapt their transmission strategies to maximize the utility defined by the application and, more importantly, be able to provide improved robustness to network attacks. The nodes will be expected to behave as cognitive agents competing for resources within a stochastic game formulation. Within this game, different levels of collaboration can be allowed depending on the “smartness” and “risk” attitudes of the different wireless entities. The cognitive nodes would be able to react to information collected from their neighbors and

compete for resources by adjusting their local cross-layer strategies in relation to dynamic routing, transmission power, channel selection, scheduling, interference avoidance, spectrum allocation, traffic shaping, source-level error resilience, etc. By doing so, the nodes are able to deliver improved performance in terms of minimizing transmission delay for delay-sensitive applications. These objectives can be achieved by developing accurate queuing models for the various cross-layer transmission algorithms and protocols, and by relying on sophisticated queuing concepts, such as service-on-vacation, to accurately model interference effects among simultaneously transmitting nodes within a unifying delay-aware framework.

(B) Adaptation to a dynamic heterogeneous network environment. Additionally, this research recognizes that a wireless network is fundamentally a dynamic system as opposed to a static interconnection of nodes. The states of the network, as well as its topology, are continuously changing due, for example, to the varying levels of resources that are available in the network. The dynamic nature of the wireless network is a strong reason why individual nodes should be able to continuously adjust their operation in response to a dynamically changing (as opposed to static or stationary) network environment. In this manner, nodes are able to match available wireless resources and successfully cope with network dynamics.

The merits of the proposed approaches can be understood from the perspective of multi-agent learning versus single-agent learning. In existing cross-layer optimization solutions, single agent learning is deployed whereby an agent repeatedly interacts with its environment (in this case, the wireless medium). In the case of a stationary environment, repeated interactions should ideally lead to a better model of the environment and, hence, to the opportunity to optimize the agents' strategy over the long run. In multi-agent learning, on the other hand, the environment is composed of other agents, which are simultaneously adapting their strategies [SL09]. Consequentially, from the perspective of any single agent or wireless node, the environment appears to be *non-stationary*. Hence,

unlike conventional cross-layer solutions, the research will focus on how to exploit the interactions among network entities with the objective of maximizing network and users' utilities at reasonable complexity under a broad set of operating scenarios.

The merits of the approaches in this dissertation can also be examined from the perspective of game theory. Earlier investigations on the application of game theory to networking problems have concentrated on characterizing properties of equilibrium conditions. However, an equilibrium operating condition reflects optimality from the perspective of any single agent and it can lead to potential loss of efficiency. In contrast, multi-agent learning leads to a dynamic network operating under non-stationary conditions. As such, the interactions among the nodes need not lead to a state of equilibrium for the network. Perpetual adaptation of strategies may persist, as long as the performance of delay-sensitive applications is maximized. Critical questions that were answered in this dissertation are how quickly conditions and behaviors can be learned and estimated, and how to optimally manage resources and adapt given the speed of change in the interferences, the channel conditions, and application-layer traffic characteristics. For example, in multiple-access radio systems, one cannot change the channel dynamics but, fortunately, one can heavily influence the interference dynamics and the signal-to-interference ratio by adapting the cross-layer transmission strategies. Thus, this research enabled modeling the various network dynamics in order to strategically design adaptation strategies at the various layers of the protocol stack to effectively respond and proactively counteract network dynamics.

(C) Dynamic risk and reward assessment. Adaptation and learning allows each cognitive node to assess the network conditions based on information collected from its neighborhood. Based on this assessment, the nodes can adjust their own utility functions. This feature is particularly relevant under emergency situations since it enables nodes to tag some parts of their data as more critical than others and to request different levels of QoS for the partitioned data. For example, some parts of the data may need to experience

a much shorter delay than other parts. Allowing for such solutions for delay-sensitive applications is an effective means to permit network survivability and resilience in dynamically changing environments because of time-varying source characteristics, wireless network conditions and infrastructure, and mission goals.

In the proposed framework, each node can estimate the *risk* that packets containing delay-sensitive content of various priorities will not arrive on time at their destination. Each node can also observe partial historic information of the outcome of the resource allocation procedure, through which the nodes can estimate the expected rewards in the future. Subsequently, the transmission strategies can be adapted to jointly consider the estimated risk of losing the packets (or not receiving them on time), as well as the impact in terms of content distortion based on the various mission goals. Preliminary results show that such cross-layer transmission strategies and dynamic routing policies based on information exchanges significantly outperform existing state-of-the-art on-demand routing solutions designed for ad-hoc wireless networks.

Additionally, existing research on learning in games assumes that either everything (the utility functions are fully known) or nothing (only the payoffs of each individual player is known) about the utility functions is known. However, in the proposed framework, a middle ground exists where there is partial knowledge of the functional form of the utility function but subject to uncertain parameters that can be estimated online. Existing approaches do not exploit the availability of such partial information.

In summary, the foundation of the research developed in this dissertation was to investigate how the autonomic nodes interact with each other to compete for resources and how and what the nodes can learn from their observed transmission history in order to improve their own strategies to interact with other wireless users. The proposed approach is to model the various wireless nodes as a collection of selfish, autonomous agents that make their own decisions and strategically interact in order to acquire wireless resources. A key aspect of the proposed solution for building robustness for

delay-sensitive applications is the decentralization of the decision-making process among the participating autonomic nodes and their ability to *comprehend* and *consciously* influence the wireless network dynamics based on the gathered information about other network nodes.

Bibliography

- [ACW95] J. Abate, G. L. Choudhury, and W. Whitt. "Exponential approximations for tail probabilities in queues I: Waiting times", *Operations Research*, vol. 43, no. 5, pp 885-901, 1995.
- [AMB04] Y. Andreopoulos, A. Munteanu, J. Barbarien, M. van der Schaar, J. Cornelis, and P. Schelkens, "In-band Motion Compensated Temporal Filtering," *Signal Processing: Image Communication* (Special Issue on "Subband/Wavelet Interframe Video Coding"), vol. 19, no. 7, pp. 653-673, Aug. 2004.
- [AMV06] Y. Andreopoulos, N. Mastronarde, and M. van der Schaar, "Cross-layer Optimized video Streaming over wireless multi-hop Mesh Networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 11, Nov 2006, pp. 2104-2115.
- [ALV06] I. F. Akyildiz, W. -Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey," *Computer Networks: The International Journal of Computer and Telecommunication Networking*, vol. 50, no. 13, Sep 2006.
- [AL94] B. Awerbuch and T. Leighton, "Improved Approximation Algorithms for the Multi-commodity Flow Problem and Local Competitive Routing in Dynamic Networks," *Proc. 26th ACM Symposium on Theory of Computing*, May 1994.
- [BBS95] A. G. Barto, S. J. Bradtke and S. P. Singh, "Learning to act using real-time dynamic programming", *Artificial Intelligence*, vol. 72, no. 1-2, Jan 1995, pp. 81-138.
- [Bel04] E. M. Belding-Royer, "Multi-level Hierarchies for Scalable Ad Hoc Routing," *ACM/Kluwer Wireless Networks (WINET)*, vol. 9, no. 5, Sept. 2004, pp. 461-478.
- [Ber82] D. P. Bertsekas, "Distributed dynamic programming", *IEEE Trans. Autom. Control*, vol. 27, no. 3, pp. 610-616, Jun 1982.
- [Ber95] D. P. Bertsekas, *Dynamic programming and Optimal Control*. vol. I, Belmont, MA: Athena Scientific, 1995.
- [BG87] D. Bertsekas, R. Gallager, *Data Networks*, Prentice Hall, Inc. Upper Saddle River, NJ, 1987.

- [BL94] J. A. Boyan and M. L. Littman, "Packet routing in dynamically changing networks: A reinforcement learning approach," in *Advances in NIPS 6*, J. D. Cowan *et al.*, Eds. San Francisco, CA: Morgan Kauffman, 1994, pp. 671–678.
- [BRB05] V. Brik, E. Rozner, S. Banarjee, P. Bahl, "DSAP: a protocol for coordinated spectrum access," in *Proc. IEEE DySPAN 2005*, Nov. 2005, pp. 611-614.
- [Bro05] T. X. Brown, "An analysis of unlicensed device operation in licensed broadcast service bands," in *Proc. IEEE DySPAN 2005*, Nov 2005, pp. 11-29.
- [BT05] A. Butala, L. Tong, "Cross-layer Design for Medium Access Control in CDMA Ad-hoc Networks," *EURASIP J. on Applied Signal Processing*, vol. 2, pp. 129-143, 2005.
- [CBD02] T. Camp, J. Boleng, V. Davies, "A survey of mobility models for ad hoc network research," in *Wireless Communications and Mobile Computing (WCMC)*, vol. 2, no. 5, pp. 483-502, 2002.
- [CCB06] C. Cordeiro, K. Challapali, D. Birru and S. Shankar N, "IEEE 802.22: An Introduction to the First Wireless Standard based on Cognitive Radios," *Journal of Communications, Academy Publishers*, vol. 1, no. 1, Apr 2006.
- [CF06] J. Chakareski and P. Frossard, "Rate-Distortion Optimized Distributed Packet Scheduling of Multiple Video Streams Over Shared Communication Resource," *IEEE Transactions on Multimedia*, vol. 8, no. 2, Apr, 2006.
- [CM06] P. A. Chou, and Z. Miao, "Rate-Distortion Optimized Streaming of Packetized Media," *IEEE Transactions on Multimedia*, vol. 8, no. 2, pp. 390-404. April 2006.
- [CW05] S. T. Cheng, M. Wu, "Performance Evaluation of Ad-Hoc WLAN by M/G/1 Queuing Model," *IEEE International Conference on Information Technology : Coding and Computing (ITCC'05)*, pp. 681-686, 2005.
- [CZ00] G. Cheung, A. Zakhor, "Bit Allocation for Joint Source/Channel Coding of Scalable Video," *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 340-356, Mar 2000.

- [CZ05] L. Cao and H. Zheng, "Distributed Spectrum Allocation via Local Bargaining," in *2nd Ann. IEEE Comm. Soc. Conf. On Sensor and Ad Hoc Comm. and Networks (SECON 2005)*, pp. 475-486, 2005.
- [CZ06] P. A. Chou, and M. Zhouong, "Rate-distortion Optimized Streaming of packetized media," *IEEE Transactions on Multimedia*, vol. 8, no. 2, pp. 390-404. April 2006.
- [DAB03] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A High Throughput Path Metric for Multi-hop Wireless Routing," *Proc. ACM Conf. Mob. Computing and Networking, MOBICOM*, pp. 134-146, 2003.
- [DCC05] J. Dowling, E. Curran, R. Cunningham, and V. Cahill, "Using Feedback in Collaborative Reinforcement Learning to Adaptively Optimize MANET Routing," *IEEE Transactions on System, Man, and Cybernetics – Part A: Systems and Humans*, vol. 35, no. 3, pp. 360-372, May 2005.
- [DHP03] Y. Diao, J. L. Hellerstein, S. Parekh, J. P. Bigus, "Managing Web Server Performance with Autotune Agens," *IBM System Journal*, vol 42, no. 1, 2003.
- [DPZ04] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proc. ACM Internat. Conf. on Mob. Computing and Networking (MOBICOM)*, 2004, pp. 114-128.
- [EM93] J. R. Evans and E. Minieka, *Optimization Algorithms for Networks and Graphs*, NY: Marcel Dekker, 1993.
- [EMM03] E. Even-Dar, S. Mannor, Y. Manour, "Action elimination and stopping conditions for reinforcement learning," *Proc. Of the International Conference on Machine Learning (ICML 2003)*, 2003.
- [FCB07] M. Felegyhazi, M. Cagalj, S. S. Bidokhti, and J.-P. Hubaux, "Noncooperative multi-radio channel allocation in wireless networks," in *IEEE INFOCOM '07*, May 2007.
- [FCC02] Federal Communications Commission (FCC), "Spectrum Policy Task Force," *ET Docket* no. 02-135, Nov 15, 2002.
- [FL98] D. Fudenberg and D.K. Levine, *The Theory of Learning in Games*, MIT Press, Cambridge, MA, 1998.
- [FT91] D. Fudenberg and J. Tirole, *Game Theory*. MIT Press, Cambridge, MA, 1991.

- [FV07] F. Fu and M. van der Schaar, "Non-collaborative resource management for wireless multimedia applications using mechanism design," *IEEE Trans. Multimedia*, vol. 9, no. 4, pp. 851-868, Jun. 2007.
- [FWK06] S. M. Faccin, C. Wijting, J. Kenckt, A. Damle. "Mesh WLAN Networks: Concept and System Design", *IEEE Wireless Communications Mag.*, pp. 10-17, Apr 2006.
- [GJ07] P. Gupta and T. Javidi, "Towards Throughput and Delay-Optimal Routing for Wireless Ad-Hoc Networks," *Asilomar Conference on Signals, Systems and Computers*, Nov. 2007.
- [GFX01] Y. Guan, X. Fu, D. Xuan, P. U. Shenoy, R. Bettati, and W. Zhao, "NetCamo: Camouflogging Network Traffic for QoS-Guaranteed Mission Critical Applications," *IEEE Transactions on System, Man, and Cybernetics – Part A: Systems and Humans*, vol. 31, no. 4, pp. 253-265, July 2001.
- [GKG07] D. Gesbert, S. G. Kiani, A. Gjendemsjo, and G. E. Oien, "Adaptation, Coordination, and Distributed Resource Allocation in Interference-Limited Wireless Networks," *Proceeding of IEEE*, vol. 95, no. 12, pp. 2393-2409, 2007.
- [GM00] D. J. Goodman and N. B. Mandayam, "Power control for wireless data," *IEEE Personal Communications*, vol. 7, pp. 48-54, Apr 2000.
- [Hah77] F. H. Hahn, "Exercises in conjectural equilibrium analysis," *Scandinavian Journal of Economics*, vol. 79, pp. 210-226, 1977.
- [Hay05] S. Haykin, "Cognitive Radio: Brain-Empowered Wireless Communications," in *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, Feb 2005.
- [HBH05] J. Huang, R. A. Berry, M. L. Honig, "Spectrum Sharing with Distributed Interference Compensation," in *Proc. IEEE DySPAN 2005*, Nov. 2005, pp. 88-93.
- [HHN08] Y. Huang, W. He, K. Nahrstedt, W. C. Lee, "Dos Resistant Broadcast Authentication with Low End-to-end Delay," *IEEE INFOCOM 2008*, April 2008.

- [HPR07] Z. Han, C. Pandana, and K. J. Ray Liu, "Distributive Opportunistic Spectrum Access for Cognitive Radio using Correlated Equilibrium and No-regret Learning", in *Proceedings of IEEE Wireless Communications and Networking Conference*, 2007.
- [Hoe63] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 31-30, Mar. 1963.
- [Hor01] P. Horn, "Autonomic Computing: IBM Perspective on the State of Information Technology," <http://www.research.ibm.com/autonomic> , Oct 2001.
- [IEE03] IEEE 802.11e/D5.0, Draft Supplement to Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS), June 2003.
- [Jan02] J. Jannotti, "Network-layer support for overlay networks," in *Proc. IEEE Conf. Open Architectures and Network Programming*, NY, June 2002.
- [JB07] T. Jiang, J. S. Baras, "Fundamental Tradeoffs and Constrained Coalitional Games in Autonomic Wireless Networks," *IEEE WiOpts*, 2007.
- [JCO02] D. Julian, M. Chiang, D. O'Neill, and S. Boyd, "QoS and fairness constrained convex optimization of resource allocation for wireless cellular and ad hoc networks," *IEEE INFOCOM 2002*, pp. 477-486.
- [JDN01] N. Jain, S. Das, and A. Nasipuri, "A multi-channel MAC protocol with receiver based channel selection for multi-hop wireless networks," *ICCCN 2001*, Oct. 2001.
- [JF06] D. Jurca and P. Frossard, "Media Streaming with Conservative Delay on Variable Rate Channels," in *Proceedings of IEEE international Conference on Multimedia and Expo (ICME, 2006)*, 2006.
- [JF07] D. Jurca, P. Frossard, "Packet Selection and Scheduling for Multipath video streaming," *IEEE Transactions on Multimedia*, vol. 9, no. 2, Apr. 2007.
- [JM96] D. B. Johnson, and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," Chapter in *Mobile Computing*, Kluwer Acad. Pub., 1996.

- [JTK01] T. Jiang, C. K. Tham, and C. C. Ko, "An approximation for waiting time tail probabilities in multiclass systems", *IEEE Communications Letters*, vol. 5, no. 4, pp 175-177. April 2001.
- [KC03] J. O. Kephart and D. M. Chess, "*The vision of autonomic computing*," *IEEE Computer Magazine*, vol. 36, no.1, pp.41-50, 2003.
- [KEW02] B. Krishnamachari, D. Estrin, S. Wicker, "The Impact of Data Aggregation in Wireless Sensor Networks," *IEEE Proc. of International Conference on Distributed Computing Systems Workshop*, pp. 575-578, 2002.
- [Kle75] L. Kleinrock, *Queuing Systems Volume I: Theory*, NY: Wiley-Interscience Publication, 1975.
- [Koe66] E. Koenigsberg, "On jockeying in queues," *Manag. Sci.* vol. 12, pp. 412-436, 1966.
- [Kon80] A. G. Konheim, "A Queuing Analysis of Two ARQ Protocols," *IEEE Transactions on Communications*, vol. com-28, no. 7, July 1980.
- [KOG07] S. G. Kiani, G. E. Oien, D. Gesbert, "Maximizing multi-cell capacity using distributed power allocation and scheduling," *IEEE Wireless Communications and Networking Conference, WCNC 2007*, pp. 1690-1694, Mar 2007.
- [Kri02] D. Krishnaswamy, "Network-assisted Link Adaptation with Power Control and Channel Reassignment in Wireless Networks," *3G Wireless Conference*, pp. 165-170, 2002.
- [KLO95] Y. A. Korilis, A. A. Lazar, and A. Orda, "Architecting Noncooperative Networks," *IEEE Journal on Selected Areas in Comm.*, vol. 13, no. 7, Sep 1995.
- [KLO97] Y. A. Korilis, A. A. Lazar, and A. Orda, "Achieving Network Optima Using Stackelberg Routing Strategies," *IEEE/ACM Transactions on Networking*, vol. 5. no. 1, Feb 1997.
- [KMT98] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237-252, Mar. 1998.

- [KN96] I. Katzela and M. Naghshineh, "Channel assignment schemes for cellular mobile telecommunications: A comprehensive survey," *IEEE Personal Communications*, vol. 3, pp. 10-31, Jun. 1996.
- [KP99] G. D. Kondylis and G. J. Pottie, "Dynamic Channel Allocation Strategies for Wireless Packet Access," *IEEE VTC*, Amsterdam, Sep 1999.
- [KSH00] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*, Prentice-Hall, NJ, 2000.
- [KSV06] D. Krishnaswamy, H. -P. Shiang, J. Vincente, V. Govindan, W. S. Conner, S. Rungta, W. Chan, K. Miao "A cross-layer cross-overlay architecture for proactive adaptive processing in mesh networks," *2nd IEEE Workshop on Wireless Mesh Networks (WiMesh 2006)*, 2006. pp. 74-82.
- [KV04] D. Krishnaswamy, and J. Vicente, "Scalable Adaptive Wireless Networks for Multimedia in the Proactive Enterprise," *Intel Technology Journal*, see online at:
http://developer.intel.com/technology/itj/2004/volume08issue04/art04_scalingwireless/p01_abstract.htm, 2004.
- [LCC07] J. W. Lee, M. Chiang, A. R. Calderbank, "Utility-Optimal Random-Access Control," *IEEE Transactions on Wireless Comm.*, vol. 6, no. 7, pp. 2741-2750. July 2007.
- [LL06] K.-D. Lee and V.C.M. Leung, "Fair Allocation of Subcarrier and Power in an OFDMA Wireless Mesh Network", *IEEE J. Sel. Areas in Commun.*, vol. 24, no. 11, pp. 2051-2060, Nov. 2006.
- [Low03] S. H. Low, "A duality model of TCP and queue management algorithms," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 525-536, 2003.
- [LS99] S. Lal, E. S. Sousa, "Distributed resource allocation for DS-CDMA-based multimedia ad hoc wireless LANs," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 5, pp. 947-967, May 1999.
- [LTH07] J. W. Lee, A. Tang, J. Huang, M. Chiang, A. R. Calderbank, "Reverse-Engineering MAC: A Non-cooperative Game Model," *IEEE Journal on Selected Areas in Comm.*, vol. 25, no. 6, pp. 1135-1147, Aug 2007.
- [Luc06] Robert W. Lucky, "Tragedy of the commons," *IEEE Spectrum*, vol. 43 no.1, pp. 88, Jan 2006.

- [LZL07] C. Long, Q. Zhang, B. Li, H. Yang, and X. Guan, "Non-cooperative power control for wireless ad hoc networks with repeated games," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 25, no. 6 pp. 1101-1112, Aug. 2007.
- [MCP06] F. Meshkati, M. Chiang, H. V. Poor, and S. C. Schwartz, "A game-theoretic approach to energy-efficient power control in multi-carrier CDMA systems," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 24, pp. 1115-1129, June 2006.
- [MD01] M. K. Marina, S. R. Das, "Ad hoc on-demand multi-path distance vector routing (AOMDV)," *Proc. of International Conference on Network Protocols (ICNP)*, pp. 14-23, 2001.
- [MGM05] M. Machado, O. Goussevskaia, R. Mini, C. G. Rezende, A. Loureiro, G. Mateus, J. Nogueira, "Data Dissemination in Autonomic Wireless Sensor Networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 12, pp. 2305-2319, Dec 2005.
- [ML99] J. R. Moorman and J. W. Lockwood, "Implementation of the multiclass priority fair queuing (MPFQ) algorithm for extending quality of service in existing backbones to wireless endpoints," *IEEE Global Telecommunications Conference*, 1999, vol. 5, pp. 2752-2757.
- [MM99] J. Mitola, G. Q. Maguire Jr., "Cognitive radio: Making software radios more personal," *IEEE Pers. Commun.*, vol. 6, no. 4, pp. 13-18, Aug. 1999.
- [MTO04] D. Marsh, R. Tynan, D. O'Kane, G. M. P. O'Hare, "Autonomic Wireless Sensor Networks," *Artificial Intelligence*, vol. 17, pp. 741-748, 2004.
- [NMR05] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic Power Allocation and Routing for Time-Varying Wireless Networks", *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, Jan 2005. pp. 89-103.
- [NH07] D. Niyato and E. Hossain, "A game-theoretic approach to competitive spectrum sharing in cognitive radio networks," in *Proc. IEEE WCNC'07*, Hong Kong, 11-15 March, 2007.
- [NZD02] A. Nasipuri, J. Zhuang, and S. R. Das, "A multi-channel MAC protocol with power control for multi-hop mobile ad hoc networks," *The Computer Journal*, 45, 2002.

- [NZT02] S. Nelakuditi, Z. Zhang, R. P. Tsang, D. H. C. Du, "Adaptive Proportional Routing: A Localized QoS Routing Approach," *IEEE/ACM Transactions on Networking*, vol. 10, no. 6, pp. 790-804, Dec 2002.
- [OR98] A. Ortega, and K. Ramchandran, "Rate-distortion Methods for Image and Video Compression," *IEEE Signal Processing Mag.*, vol. 15, no. 6, pp. 23-50, Nov, 1998.
- [PB94] C. E. Perkins, P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 4, pp. 234-244, Oct. 1994.
- [PR99] C. E. Perkins, E. M. Royer, "Ad hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90-100, Feb 1999.
- [Put94] M. L. Puterman, *Markov Decision Process: Discrete Stochastic Dynamic Programming*, John Wiley & Sons, Inc. New York, 1994.
- [PYC08] A. Proutiere, Y. Yi, M. Chiang, "Throughput of Random Access without Message Passing," *CISS 2008*, Mar. 2008, pp. 509-514.
- [QCS02] D. Qiao, S. Choi and K. G. Shin, "Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LAN", *IEEE Transactions on Mobile Computing*, vol.1, no. 4, 2002.
- [Rap02] T. S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 2002.
- [RC05] A. Raniwala, T. Chiueh, "Architecture and Algorithms for an IEEE 802.11-based Multi-channel Wireless Mesh Network", *INFOCOM 2005*.
- [RE07] A. Rezgui and M. Eltoweissy, "Service-Oriented Sensor-Actuator Networks," *IEEE Communications*, vol. 45, no. 12, pp 92-100, Dec 2007.
- [RHA04] M. Raya, J. -P. Hubaux, and I. Aad, "DOMINO: a system to detect greedy behavior in IEEE 802.11 hotspot," in *MobiSys'04*, 2004.
- [RT02] T. Roughgarden, E. Tardos, "How Bad is Selfish Routing?" *Journal of the ACM*, vol. 49, no. 2, pp. 236-259, March 2002.

- [QCS02] D. Qiao, S. Choi and K. G. Shin, "Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LAN", *IEEE Transactions on Mobile Computing*, vol.1, no. 4, 2002.
- [SB97] S. Singh, D. Bertsekas, "Reinforcement learning for dynamic channel allocation in cellular telephone systems," In *Advances in Neural Information Processing Systems*, pp. 974-980, Cambridge MA, 1997.
- [SCC05] S. Shankar, C. T. Chou, K. Challapali, and S. Mangold, "Spectrum agile radio: capacity and QoS implementations of dynamic spectrum assignment," *Global Telecommunications Conference*, Nov. 2005.
- [SCN03] S. H. Shah, K. Chen, and K. Nahrstedt, "Available Bandwidth Estimation in IEEE 802.11-Based Wireless Networks," in *ISMA/CAIDA 1st Bandwidth Estimation Workshop (BES 2003)*, 2003.
- [SL09] Y. Shoham, K. Leyton-Brown, *Multi-Agent System: Algorithms, Game-Theoretic, and Logical Foundations*, Cambridge University Press, 2009.
- [SMG02] C. U. Saraydar, N. B. Mandayam, and D. J. Goodman, "Efficient power control via pricing in wireless data networks," *IEEE Trans. on Commun.*, vol. 50, no. 2, pp. 291-303, Oct 2002.
- [SP85] S. Sabri and B. Prasada, "Video Conferencing Systems," *Proc. of the IEEE*, vol. 73, no. 4, pp. 671-688, 1985.
- [SPG07] Y. Shoham, R. Powers, and T. Grenager, "If multi-agent learning is the answer, what is the question?" *Artificial Intelligence*, vol. 171, no. 7, pp. 365-377, May 2007.
- [SPI05] C. Shen, D. Pesch, J. Irvine, "A Framework for Self-management of Hybrid Wireless Networks using Autonomic Computing Principles," *IEEE Comm. Networks and Services Research Conference*, pp. 261-266, May 2005.
- [Sut88] R. S. Sutton, "Learning to predict by the method of temporal differences," *Machine Learning*, vol. 3, no. 1, pp. 9-44, Aug. 1988.
- [SV04] J. So, N. H. Vaidya, "Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals using a Single Transceiver," *ACM International Symp. Mobile Ad Hoc Net. And Comp (MOBIHOC)*, May 2004, pp. 222-233.

- [SV06] H. -P. Shiang, and M. van der Schaar, "Multi-user Video Straming over Multi-hop Wireless Networks: A Cross-layer Priority Queuing Approach," in *IEEE Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2006)*, pp. 255-258, Dec, 2006.
- [SV07a] H. Shiang and M. van der Schaar, "Multi-user video streaming over multi-hop wireless networks: A distributed, cross-layer approach based on priority queuing," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 4, pp. 770-785, May 2007.
- [SV07b] H.-P. Shiang and M. van der Schaar, "Informationally Decentralized Video Streaming over Multi-hop Wireless Networkss," *IEEE Trans. Multimedia*, vol. 9, no. 6, pp. 1299-1313, Sep 2007.
- [SV08] H. P. Shiang and M. van der Schaar, "Queuing-Based Dynamic Channel Selection for Heterogeneous Multimedia Applications over Cognitive Radio Networks," *IEEE Trans. Multimedia*, Vol. 10, no.5, pp. 896-909, Aug. 2008.
- [SW04] G. Staple and K. Werbach, "The End of Spectrum Scarcity," *IEEE Spectrum*, vol. 41, no. 3, pp. 48-52, Mar 2004.
- [SYZ05] E. Setton, T. Yoo, X. Zhu, A. Goldsmith, and B. Girod, "Cross-layer design of Ad hoc Networks for real-time video streaming," *IEEE Wireless Communications Mag.*, pp. 59-65, Aug 2005.
- [TB00] H. Tong, T. X. Brown, "Adaptive Call Admission Control under Quality of Service Constraints: A Reinforcement Learning Solution," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 2, pp. 209-221, Feb 2000.
- [TG03] S. Toumpis, A. J. Goldsmith, "Capacity Regions for wireless Ad Hoc Network", *IEEE Transactions on Wireless Communications*, vol. 2, no. 4, pp. 736-748, July 2003.
- [TJZ03] X. Tan, W. Jin, and D. Zhao, "The application of multi-criterion satisfactory optimization in computer networks design," *IEEE Proc. of the 40th International Conference on Parallel and Distributed Computing, Applications and Technologies*, Aug 2003, pp. 660-664.
- [TL08] A. Tizghadam, A. Leon-Garcia, "On Congestion in Delay-sensitive Networks," *IEEE INFOCOM 2008*, April 2008.
- [TO98] P. Tadepalli and D. Ok, "Model-based average reward reinforcement learning", *Artificial Intelligence*, vol. 100, no. 1-2, Jan 1998, pp. 177-224.

- [TJ91] S. Tekinay and B. Jabbari, "Handover and Channel Assignment in Mobile Cellular Networks," *IEEE Communication Magazine*, vol. 29, pp. 42-46, Nov 1991.
- [VAH06] M. van der Schaar, Y. Andreopoulos, Z. Hu, "Optimized Scalable Video Streaming over IEEE 802.11a/e HCCA Wireless Networks under Delay Constraints," *IEEE Trans. On Mobile Computing*, vol. 5, no. 6, pp. 755 – 768, June 2006.
- [VCS03] A. Vetro, C. Christopoulos, H. Sun, "Video Transcoding Architectures and Techniques: An Overview," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 18-29, Mar 2003.
- [VT07] M. van der Schaar, D. S. Turaga, "Cross-layer Packetization and Retransmission Strategies for Delay-sensitive wireless Multimedia Transmission," *IEEE Transactions on Multimedia*, vol. 9, no. 1, pp. 185-197, Jan 2007.
- [VS05] M. van der Schaar and S. Shankar, "Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms," *IEEE Wireless Commun. Mag.*, vol. 12, no. 4, pp. 50-58, Aug. 2005.
- [WCZ05] Y. Wu, P. A. Chou, Q. Zhang, K. Jain, W. Zhu, S.Y. Kung, "Network Planning in Wireless Ad Hoc Networks: A Cross-Layer Approach", *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 136-150, Jan. 2005.
- [WD92] C. J. C. H. Watkins, P. Dayan, "Q-learning", *Machine Learning*, vol. 8, no. 3-4, pp. 279-292, May 1992.
- [WH98] M. P. Wellman and J. Hu, "Conjectural equilibrium in multiagent learning," *Machine Learning*, vol. 33, pp. 179-200, 1998.
- [WP02] C. C. Wang and G. J. Pottie, "Variable Bit Allocation for FH-CDMA Wireless Communication Systems," *IEEE Transactions on Communications*, vol.50, no. 6, Oct 2002.
- [WPT03] R. Want, T. Pering, D. Tennenhouse, "Comparing autonomic and proactive computing," *IBM Systems Journal*, vol. 42, no. 1, 2003.
<http://www.research.ibm.com/journal/sj/421/want.html>
- [WR03] M. Waldvogel and R. Rinaldi. "Efficient Topology-Aware Overlay Network," *ACM SIGCOMM Computer Comm. Review*, vol. 33, no. 1, pp. 101-106, Jan 2003.

- [WV06] M. Wang and M. van der Schaar, "Operational Rate-Distortion Modeling for Wavelet Video Coders," *IEEE Transactions on Signal Processing*, vol. 54, no. 9, pp. 3505-3517, Sep. 2006.
- [WYT06] H. Wu, F. Yang, K. Tan, J. Chen, Q. Zhang, Z. Zhang, "Distributed Channel Assignment and Routing in Multi-radio Multi-channel Multi-hop Wireless Networks", in *IEEE JSAC special issue on multi-hop wireless mesh networks*, vol. 24, no. 11, pp. 1972-1983, Nov 2006.
- [WZ02] W. Wei, and A. Zakhor, "Multipath unicast and multicast video communication over wireless ad hoc networks," *Proc. Int. Conf. Broadband Networks*, Broadnets, pp. 496-505, 2002.
- [WZF04] J. Wang, H. Zhai, and Y. Fang, "Opportunistic Packet Scheduling and Media Access Control for Wireless LANs and Multi-hop Ad Hoc Networks," *IEEE Wireless Communications and Networking Conference*, vol. 2, pp. 1234-1239, Mar 2004.
- [WZQ08] F. Wu, S. Zhong, C. Qiao, "Globally optimal channel assignment for non-cooperative wireless networks," *INFOCOM 2008*, pp. 2216-2224.
- [XCR08] D. Xu, M. Chiang, and J. Rexford, "Link-state Routing with Hop-by-Hop Forwarding Achieves Optimal Traffic Engineering", *Proc. IEEE INFOCOM*, 2008.
- [XJB04] L. Xiao, M. Johansson, S. P. Boyd, "Simultaneous Routing and Resource Allocation Via Dual Decomposition," *IEEE Transactions on Communications*, vol. 52, no. 7, pp. 1136-1144, July 2004.
- [XSC03] M. Xiao, N. B. Shroff, and E. J. P. Chong, "A Utility-Based Power-Control Scheme in Wireless Cellular Systems," *IEEE/ACM Transactions on Networking*, vol. 11, pp. 210-221, Apr 2003.
- [YGC02] W. Yu, G. Ginis, and J. M. Cillfi, "Distributed Multi-user Power Control for Digital Subscriber Lines," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 5, pp. 1105-1115, Jun. 2002.
- [YL06] W. Yu, R. Lui, "Dual Methods for Nonconvex Spectrum Optimization of Multi-carrier Systems," *IEEE Transactions on Communications*, vol. 54, no. 7, July 2006.
- [You04] H. P. Young, *Strategic learning and its Limits*, Oxford University Press, NY 2004.

- [ZC05] H. Zheng, and L. Cao, "Device-Centric Spectrum Management," in *Proc. IEEE DySPAN 2005*, Nov. 2005, pp. 56-65.
- [ZL06] S. A. Zekavat, and X. Li, "Ultimate Dynamic Spectrum Allocation via User-central Wireless Systems," *Journal of Communications, Academy Publishers*, vol. 1, no. 1, pp. 60-67, Apr 2006.
- [ZP05] H. Zheng, and C. Peng, "Collaboration and Fairness in Opportunistic Spectrum Access," In *Proc. 40th annual IEEE International Conference on Communications*, Jun 2005.
- [ZTS07] Q. Zhao, L. Tong, A. Swami, and Y. Chen, "Decentralized Cognitive MAC for Opportunistic Spectrum Access in Ad Hoc Networks: A POMDP Framework" *IEEE Journal on Selected Areas in Communications (JSAC): Special Issue on Adaptive, Spectrum Agile and Cognitive Wireless Networks*, vol. 25, no. 3, pp. 589-600, April, 2007.
- [ZZY05] J. Zhao, H. Zheng, G.-H. Yang, "Distributed Coordination in Dynamic Spectrum Allocation Networks," in *Proc. IEEE DySPAN 2005*, Nov 2005, pp. 259-268.