A New Systematic Framework for Autonomous Cross-Layer Optimization

Fangwen Fu, *Student Member, IEEE*, and Mihaela van der Schaar, *Senior Member, IEEE*

***Abstract*—Cross-layer optimization solutions have been proposed in recent years to improve the performance of wireless users that operate in a time-varying, error-prone network environment. However, these solutions often rely on centralized cross-layer optimization solutions that violate the layered network architecture of the protocol stack by requiring layers to provide access to their internal protocol parameters to other layers. This paper presents a new systematic framework for cross-layer optimization, which allows each layer to make autonomous decisions to maximize the wireless user's utility by optimally determining what information should be exchanged among layers. Hence, this cross-layer framework preserves the current layered network architecture. Since the user interacts with the wireless environment at various layers of the protocol stack, the cross-layer optimization problem is solved in a layered fashion such that each layer adapts its own protocol parameters and exchanges information (messages) with other layers that cooperatively maximize the performance of the wireless user. Based on the proposed layered framework, we also design a message-exchange mechanism that determines the optimal cross-layer transmission strategies, given the user's experienced environment dynamics.**

***Index Terms*—Autonomous decision making, cross-layer optimization, environmental dynamics, information exchange, layered dynamic programming (DP) operator.**

I. Introduction

The OPEN systems interconnection (OSI) model [1] is a layered abstract organization of various communication and computer network protocols. In layered network architectures, each layer autonomously controls and optimizes a subset of decision variables (i.e., protocol parameters) based on the information (or observations) obtained from other layers to provide services to the layer(s) above. The advantage of layered architectures is that the designer or implementer of the protocol or algorithm at a particular layer can focus on the design of that layer, without being required to consider all the parameters and algorithms of the rest of the stack [3]. However, in current layered network architectures, the information exchange between multiple layers is often implemented in an ad hoc manner. This generally results in suboptimal performance for the users and their applications.

Manuscript received December 15, 2007; revised May 12, 2008 and September 4, 2008. This work was supported by the National Science Foundation through CAREER Award CCF-0541867. The review of this paper was coordinated by Dr. H. Jiang.

The authors are with the Department of Electrical Engineering, University of California at Los Angeles, Los Angeles, CA 90095 USA (e-mail: fwfu@ee.ucla.edu; mihaela@ee.ucla.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2008.2007418

To optimize the different protocol parameters, the wireless users (transmitter and receiver pairs) need to consider the dynamic wireless network "environment" shaped by the repeated interaction with other users, the time-varying channel conditions, and, for delay-sensitive applications, the time-varying traffic characteristics. Moreover, it should be noted that to maximize its utility, a wireless user needs to jointly optimize the protocol parameters selected at each layer of the OSI stack. The joint optimization of the transmission strategies at the various layers is referred to as cross-*layer optimization* [2], [3]. Recently, various cross-layer optimization methods have been proposed to jointly adapt the transmission strategies at each layer to the rapidly varying network environment. A brief review of this work is presented next.

A. Related Work

Application-Specific Solutions: Numerous solutions have been proposed in recent years to provide efficient adaptation of specific applications (e.g., real-time multimedia transmission) to error-prone networks (e.g., Internet and wireless networks) [25]. A majority of these solutions consider the lower layers as a "black box" and adapt the application (APP) layer strategies based on the information fed back from the lower layers (e.g., information about the network congestion and packet loss rates), as shown in Fig. 1(a). These solutions aim at providing applications the information necessary to adapt their own algorithms and parameters, without exposing the details of the lower layers' protocols and algorithms to the applications. These application-specific solutions, however, often ignore the adaptability of lower layers [e.g., transport layer, network layer, media access control (MAC) layer, and physical (PHY) layer].

Layer-Centric Solutions: To jointly consider the lower layers' adaptation, numerous solutions have also been proposed to allow the APP layer to drive the adaptation of network parameters and algorithms by permitting the application to access the internal protocol parameters of the lower layers [2], as shown in Fig. 1(b). Alternative solutions are also developed to allow a certain layer (e.g., the MAC layer) other than the APP layer to drive the cross-layer adaptation by accessing the internal protocol parameters and algorithms of the other layers [4]–[6], as shown in Fig. 1(c). Although these approaches jointly adapt the cross-layer strategies and significantly improve the overall user's performance, they *violate* the layered network architecture, since they require access to the internal variables of other layers. This violation of the layered network architecture has several disadvantages. These disadvantages include creating more dependencies between layers and increasing the

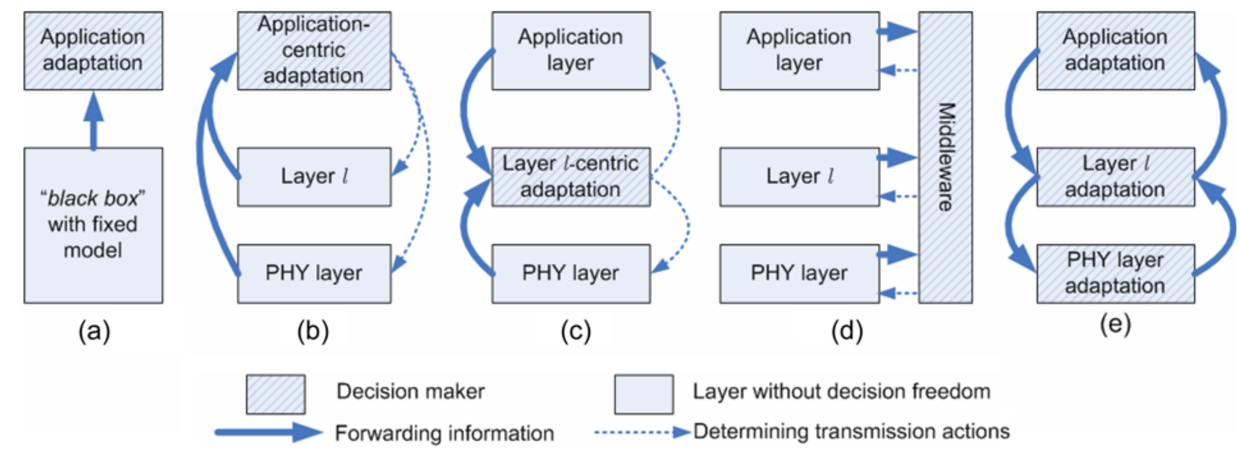

Fig. 1. Conceptual illustration of cross-layer optimization methods. (a) Application adaptation. (b) Application-centric adaptation. (c) Middle layer-centric adaptation. (d) Middleware-based adaptation. (e) Proposed autonomous adaptation with information exchange.

difficulty of independent protocol and algorithm design at the various OSI layers, since one layer needs to be aware of the parameters of the other layers [3].

Centralized Solutions: Another type of cross-layer optimization involves the use of middleware or system-level monitors (centralized optimizers) to estimate resource availability and environmental dynamics, coordinate the allocation of resources across applications and nodes, and adapt the protocols' algorithms and parameters at each layer based on the experienced dynamics [15], as shown in Fig. 1(d). These solutions typically coordinate a subset of the system layers and maximize the user's utility, given all the various resource constraints (e.g., power and delay). First, it is clear that the centralized cross-layer optimization solutions require each layer to forward the complete information about its protocol-dependent dynamics, as well as its possible protocol parameters and algorithms, to the middleware or system-level monitors. Hence, this centralized decision also *violates* the current layered network architecture [3]. Second, the centralized optimization obliges each layer to take the actions (i.e., select the protocol parameters and algorithms) dictated by the central optimizer. The layers have no freedom to adapt their own actions to the environmental dynamics (e.g., source and channel characteristics) that they experience. Hence, inherently, each layer loses the authority to design and select its own suite of protocols and algorithms independently of the other layers, thereby inhibiting the upgrade of the protocols and algorithms at each layer.

In summary, most existing cross-layer design solutions optimize the protocol parameters in an integrated fashion by jointly and simultaneously considering the dynamics at each layer and requiring layers to provide access to their internal protocol parameters to other layers. These cross-layer interactions create the dependencies among the layers, which will affect not only the concerned layer, but also the other layers. Hence, a majority of these integrated approaches violate the layered network architecture of the protocol stack, thereby requiring a complete redesign of current networks and protocols and leading to a high implementation cost [3]. Another limitation of many existing cross-layer solutions is that they react to the experienced network dynamics in a "myopic" way by optimizing the transmission strategies based on the information about the current network dynamics and current application requirements [2], [8], [9]. As shown in our preliminary work [14], to obtain an optimal utility, applications need to adopt *foresighted* adaptation, which considers not only the immediate network status, but how the network dynamics evolve over time as well.

B. Key Features of the Proposed Framework

In this paper, we focus on developing a new systematic framework for cross-layer optimization based on *foresighted* decision making such that the selected transmission strategies at each layer depend not only on the immediate reward, but also on their impact on the future reward. Moreover, the proposed framework *preserves* the current layered architecture of the protocol stack by allowing the layers to make autonomous decisions based on their locally experienced dynamics and message exchanges among the layers, as shown in Fig. 1(e). Thus, the proposed cross-layer solution is compliant with existing protocols and standards available at various layers.

Similar to works in [15], [17], [19], and [20], we model the cross-layer optimization problem as a Markov decision process (MDP) [11] that has as its objective the maximization of the discounted sum of future utility. This way, the impact of the currently selected cross-layer transmission strategy on the future utility (reward) is formulated in a systematic manner. The proposed cross-layer design formulation is presented in Section III.

Traditionally, the MDP problem is solved using value iteration or policy iteration algorithms [12]. The key component of these algorithms is the dynamic programming (DP) operator. In the current cross-layer optimization literature, the DP operator is deployed in a centralized way, i.e., the transmission strategies of all the layers are jointly and simultaneously determined by a central optimizer or a middleware, as shown in Fig. 1(d). The disadvantages of this centralized solution have been discussed in Section I-A. In this paper, we propose a layered DP operator that complies with the layered architecture and protocol design of current wireless networks. Using this layered DP operator, each layer makes its transmission decision [i.e., selects the transmission strategies, e.g., packet scheduling in the APP layer, retransmission in the MAC layer, and modulation selection in the PHY layer] in an autonomous manner by considering the dynamics experienced at that layer, as well as the information available from other layers. Importantly, this layered optimization framework preserves the current layered network architecture and does not require each layer to access the internal protocol parameters of other layers. This feature is desired for the layered network architecture since different layers of the protocol stack may be implemented by different companies, which may not desire to provide access to their parameters and algorithms to other layers that are developed by other companies.

Specifically, to exchange information across multiple layers, we define a message exchange mechanism in which the *content* of the message captures the performed transmission strategies and experienced dynamics at each layer. However, the *format* of the message is independent of the transmission strategies, protocols, and dynamics implemented at each layer and can be implemented using any agreed-upon signaling protocol [18]. Hence, the various protocols can be kept the same, upgraded or entirely modified; the algorithms at the various layers can also be upgraded; and the supported applications can be changed without affecting the proposed cross-layer design framework. Furthermore, certain layers or algorithms can decide not to exchange any messages or not to participate in the cross-layer optimization.

In summary, this paper makes the following contributions.

1) We propose a new theoretic cross-layer optimization framework that provides a systematic, rather than ad hoc, mechanism for dynamically selecting and adapting the transmission strategy at each layer and the message exchange across layers. A layered DP operator is proposed such that each layer autonomously makes its transmission decision by considering its own experienced network dynamics and message exchanges from other layers. This layered optimization framework does not require a central decision maker to consider all the layers' parameters, constraints, protocols, algorithms, etc.
2) A message-exchange mechanism between the layers is developed, in which messages capture the experienced dynamics and the performed transmission strategies, but the format of the message is independent of the transmission strategies, deployed protocols, and dynamics experienced at each layer.

Hence, the proposed cross-layer framework keeps the layered network architecture unaltered and provides network designers the freedom of a scalable, flexible, and easily upgradable network design.

C. *Paper Organization*

The rest of this paper is organized as follows. Section II discusses the problem settings for the cross-layer optimization. Section III briefly reviews the centralized DP operator to solve the MDP-based cross-layer optimization problem. Section IV presents a layered DP operator framework and discusses the advantages of the layered DP operator. Section V gives an illustrative example to verify the efficiency of the layered DP operator. This paper concludes in Section VI.

II. Cross-Layer Problem Formulation

We consider an autonomous wireless user transmitting its time-varying traffic to another wireless user (e.g., base station) over a one-hop wireless network (e.g., wireless local area network and cellular network). We study how this wireless user can autonomously adapt its transmission strategies[1] at the APP, MAC, and PHY layers to maximize its utility. We assume that there are L participating layers[2] in the protocol stack. Each layer is indexed $l \in \{1, \ldots, L\}$, with layer 1 corresponding to the lowest participating layer (e.g., PHY layer) and layer L corresponding to the highest participating layer (e.g., APP layer). In this paper, we focus on user-centric cross-layer adaptation, where the wireless user performs cross-layer adaptation of the L layers to maximize its own utility.

Although the cross-layer optimization framework proposed in this paper is general, can be applied in different wireless network settings, and can involve a variety of network protocols, we would like to first provide a concrete example of a cross-layer optimization problem to help readers become familiar with the concept of actions and states before we formally define them in Sections II-B and C.

A. *Illustrative Cross-Layer Optimization Example*

Similar to [15], in this example, we consider that the wireless user transmitting delay-sensitive data accesses the wireless channel. The channel access can be based on time-division multiple access (TDMA) or on asynchronous code-division multiple access (A-CDMA). In the PHY layer, the wireless user experiences the channel noise (e.g., additive Gaussian noise [1]) and interference from the other users due to imperfect synchronization or code design [1]. In cellular networks, interference can also be incurred from neighboring cells. The channel quality experienced by the wireless user is represented by the signal-to-interference-plus-noise ratio (SINR), which is determined by the transmission power, channel noise, and interference. Given the power allocation, the channel quality is often modeled as a finite-state Markov chain (FSMC) [16], [26]. In this example, we consider a more general case in which the channel quality is modeled as an FSMC with the state transition being controlled by the power allocation. Given the SINR, the wireless user also adapts the modulation schemes to determine the service provided to the upper layers.

In the MAC layer, if the channel access is based on TDMA, the amount of time allocated to the wireless user during one time slot depends on the scheduling algorithm deployed in the network, e.g., the predetermined scheduling in the 802.11e hybrid coordination function [10] or the repeated resource competition discussed in [14]. In the resource competition scenario, the wireless user will need to autonomously and dynamically compete for transmission time with other users. In both resource-management scenarios, we can use an FSMC

[1] In this paper, we focus on wireless transmission over one-hop networks, and thus, the transmission strategies at the transport layer and network layer are not considered.

[2] If one layer does not participate in the cross-layer design, it can simply be omitted. Hence, we consider here only the L participating layers.

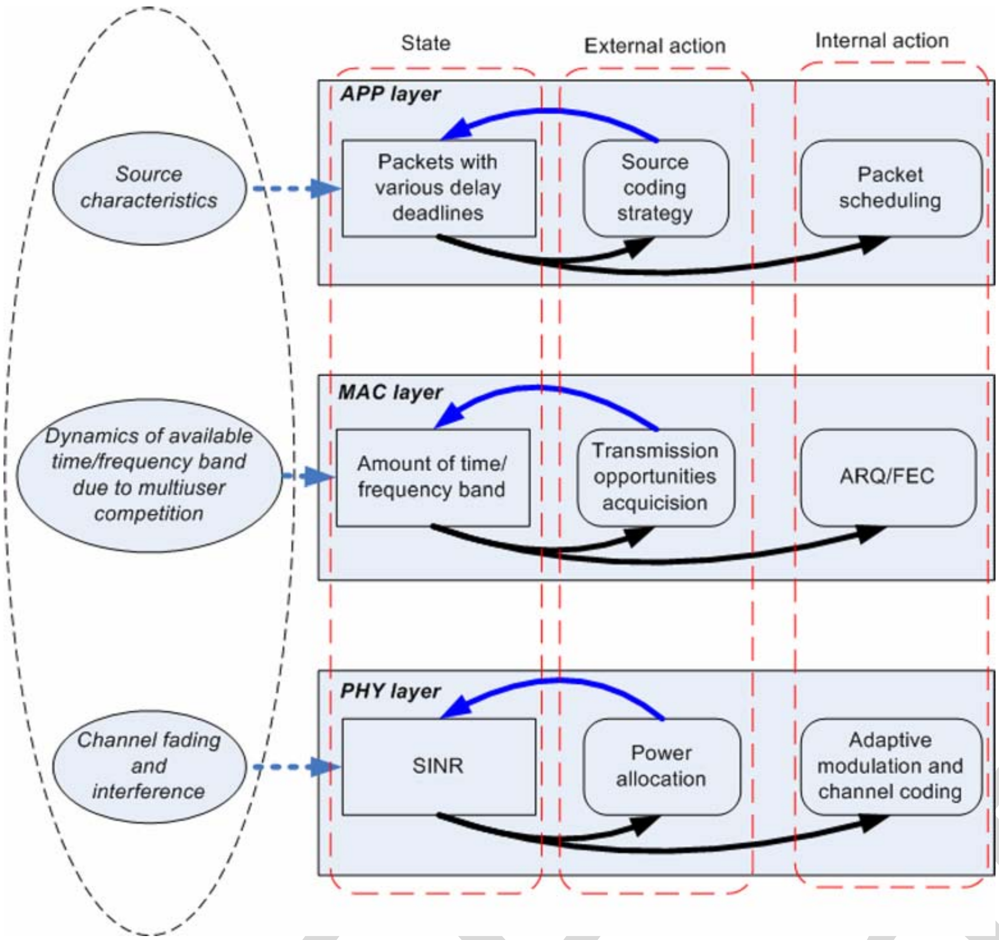

Fig. 2. Internal and external actions and states for the cross-layer optimization in the example.

that has as its states the amount of time allocated to the wireless user to model the resource-allocation process. However, the state transition of the FSMC is determined by the user's strategies to compete for the network resources with other wireless users (e.g., the bid strategy in the resource auction game [14] in the MAC layer). If the resource allocation is predetermined, then the process is then controlled by a constant action. This model can capture the dynamics experienced by a user due to the multiuser interaction. If the channel access is based on A-CDMA, then the wireless users can access the channel all the time. The state transition is a special case of FSMC with the state being constant. In addition to the resource allocation, the MAC can also perform error control algorithms such as Automatic Repeat-reQuest (ARQ) or forward error correction (FEC) to improve the service provided to the upper layers.

In the APP layer, we assume that the wireless user generates delay-sensitive traffic. The delay sensitivity is represented by the delay deadlines after which the packets will expire, and thus, they will not contribute to the wireless user's application quality. As in [15], we can model the number of packets with the various delay deadlines available for transmission as an FSMC. Since the transmission strategies at the lower layers determines the amount of packets to be transmitted and the source coding algorithms determines the amount of packets to arrive for transmission, the state transition is then controlled by the transmission strategies at the lower layers and the source-coding algorithms.

The objective of the wireless user is to jointly adapt the transmission strategies across all the three layers such that the user's utility is maximized.

B. States

In wireless communication, different states can be defined at each layer to capture the currently experienced dynamics [12], [15]. In this paper, the state of the layers is defined such that future transmission strategies can be determined independently of the past history of the transmission strategies and environment, given the current state, i.e., the state is Markovian. To adhere to the layered architecture of current networks, we define a state $s_l \in \mathcal{S}_l$ for each layer l. Then, the state of the entire wireless user is denoted by $s = (s_1, \ldots, s_L) \in \mathcal{S}$, with $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_L$. The states of the cross-layer optimization example are illustrated in Fig. 2.

C. Actions

In a layered architecture, a wireless user takes different transmission actions in each state of each layer. The transmission actions can be classified into two types at each layer l: An *external action* is performed to determine what the next state should be (i.e., state transition) such that the future reward will be improved, and an *internal action* is performed to determine the service provided to the upper layers for the packet(s) transmission in current time slot.

The external actions at each layer l are denoted by $a_l \in A_l$, where A_l is the set of the possible external actions available at layer l. The external actions of the wireless user at all the layers are denoted by $a = (a_1, \ldots, a_L) \in A$, where $A = A_1 \times \cdots \times A_L$. The internal actions are denoted by $b_l \in B_l$, where B_l is the set of the possible internal actions available at layer l. The internal actions are performed by the wireless user to efficiently *utilize* the allocated wireless network resource and its own resource budget (e.g., power constraint) by providing the quality of service (QoS) required by the supported applications. The internal actions of the wireless user across all the layers are denoted by $b = (b_1, \ldots, b_L) \in B$, where $B = B_1 \times \cdots \times B_L$. The action at layer l is the aggregation of external and internal actions, which is denoted by $\xi_l = (a_l b_l) \in \mathcal{X}_l$, where $\mathcal{X}_l = A_l \times B_l$. The joint action of the wireless user is denoted by $\xi = (\xi_1, \ldots, \xi_L) \in \mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_L$. The external and internal actions in the cross-layer optimization example are illustrated in Fig. 2.

Distinguishing between the internal and external transmission actions has the following advantages, which will become clearer in Section IV.

1) The current utility computation based on the internal actions can be computed independently of the state transition that takes place due to the external actions deployed at each layer. This separation enables us to design a cross-layer optimization framework that complies with the current layered architecture of the protocol stack.
2) The separation between the internal actions and external actions enables us to design an interlayer message exchange mechanism that is independent of the specific format of the protocols and algorithms deployed at each layer.

D. Transition Probability

In this section, we examine the structure of the state transition model and the underlying models for environmental dynamics. In general, because states are Markovian, the state transition of the wireless user only depends on the current state s, the current performed external actions, and the environmental dynamics. The corresponding transition probability is denoted by $p(s'|s, \xi)$. This global state transition can be compactly represented using a dynamic decision network [22]. Formally, the transition model is decomposed as

$$p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{\xi}) = \prod_{l=1}^{L-1} p\left(s_l'|parent\left(s_l'\right), action\left(s_l'\right)\right) \qquad (1)$$

where $parent(s_l')$ represents the set of states on which the transition of s_l' depends, and $action(s_l')$ represents the set of actions performed at the current time that affect the transition of s_l'.

In the cross-layer optimization example, the state transition at each layer $l < L$ is only controlled by the external actions at that layer and is independent of the other layers' states and actions. At layer L, the state transition is determined by the external actions at that layer and internal actions of all the layers. Motivated by this example, we can further simplify the transition probability for the cross-layer optimization as

$$p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{\xi}) = \prod_{l=1}^{L-1} p\left(s_l'|s_l, a_l\right) p\left(s_L'|\boldsymbol{s}, a_L, \boldsymbol{b}\right). \qquad (2)$$

Comparing (2) with (1), we note that $parent(s_l') = \{s_l\}$ and $action\{s_l'\} = \{a_l\}$ for $l \in \{1, \ldots, L-1\}$, and $parent(s_L') = \{s\}$ and $action\{s_L'\} = \{a_L, b\}$. In other words, the state transition at the lower layer ($l \in \{1, \ldots, L-1\}$) is driven by the external action a_l at that layer and depends only on its own current state s_l. At layer L, the state transition is determined using both the external action a_L as well as the internal actions b at all the layers. We also allow the state transition at layer L to depend on the current states s of all the layers. We should note that although the state transition in the lower layers ($l < L$) is independent of other layers' state, the external action selection at that layer will depend on the message (e.g., the future reward generated by the upper layer) exchanged with the other layers, which will be specified in Sections IV-C and D. Fig. 3 illustrates how the state transition is determined.

This decomposition is determined such that the cross-layer optimization is complying with the layered network architecture and enables the development of a layered framework for cross-layer optimization, which will be presented in Section IV.

E. Utility Function

The application quality obtained in layer L is based on the states and internal actions at each layer and is denoted by $g(s, b)$. At the same time, performing the internal actions at various layers will incur the internal cost $d(s, b)$, and it will be set to zero if no cost is incurred. The external cost $c_l(s_l, a_l)$ at layer l represents the cost of performing the external action, e.g., the amount of power allocated to determine the channel conditions or the tax (tokens, money) spent for consuming wireless resources [13], [14]. The utility gain and the corresponding costs are depicted in Fig. 3. In this paper, we have defined the reward as

$$R(\boldsymbol{s}, \boldsymbol{\xi}) = g(\boldsymbol{s}, \boldsymbol{b}) - \lambda^b d(\boldsymbol{s}, \boldsymbol{b}) - \sum_{l=1}^{L} \lambda_l^a c_l(s_l, a_l) \qquad (3)$$

where λ^b and λ_l^a are positive parameters that trade off between the application quality and cost incurred by performing certain actions. These parameters can be determined based on the resource budgets available for the wireless user [17] or by the network coordinator to efficiently utilize the network resources [24]. In this paper, we assume that these parameters are known to the wireless users, and we focus on the internal and external action selection for utility maximization. The reward in (3) can be further decomposed into the following two parts: 1) the internal reward, which depends on the internal

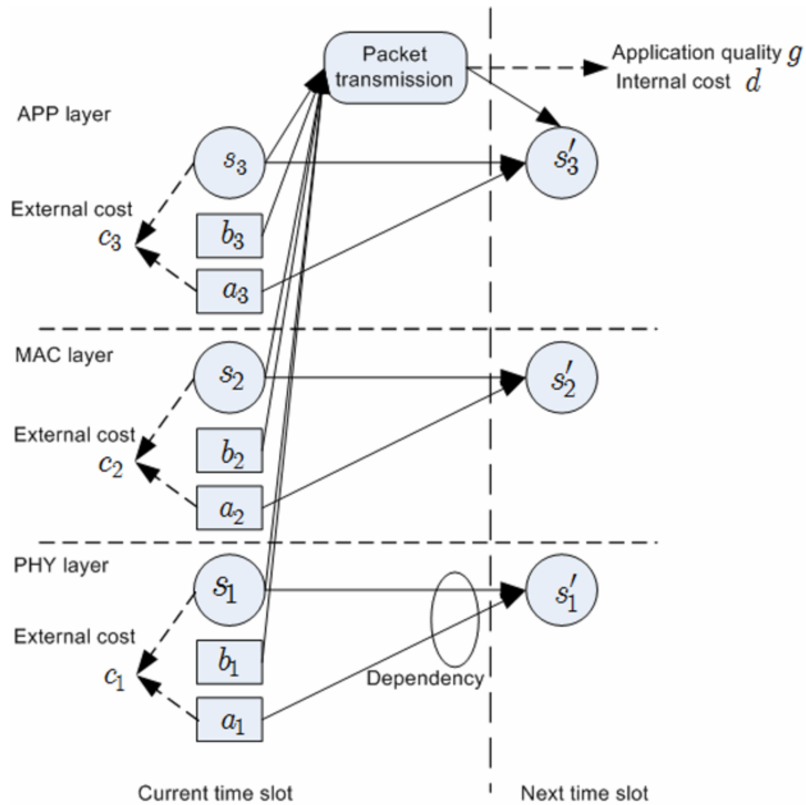

Fig. 3. Layered transition model and components of decomposed utility function.

actions; and 2) the external reward, which depends on the external actions. The internal reward is

$$R_{\text{in}}(\boldsymbol{s}, \boldsymbol{b}) = g(\boldsymbol{s}, \boldsymbol{b}) - \lambda^b d(s, b) \tag{4}$$

and the external reward is

$$R_{\text{ex}}(\boldsymbol{s}, \boldsymbol{a}) = -\sum_{l=1}^{L} \lambda_l^a c_l(s_l, a_l). \tag{5}$$

Hence, the reward is $R = R_{\text{in}} + R_{\text{ex}}$.

F. MDP Formulation for Foresighted Cross-Layer Optimization

As described in Section II-D, the state transition at each layer is controlled by the external actions. For simplicity, we assume that the state transition in each layer is synchronized and operates at the same time scale such that the transition can be discretized into stages during which the wireless user has constant state and performs static actions. The length of the stage is denoted by $\triangle T$ and can be determined based on how fast the environment changes. We use a superscript k to denote stage k. Hence, the state of the wireless user at stage $k \in \mathbb{N}$ is denoted by s^k, with each element s_l^k being the state of layer l; similarly, the joint action performed by the wireless user at stage k is ξ^k, with each element $\xi_l^k = (a_l^k, b_l^k)$. The state transition probability is given by (2), and the stage reward is given by (3).

Unlike the conventional cross-layer adaptation that focuses on maximizing the myopic (i.e., immediate) utility, in the proposed cross-layer framework, the goal is to find the optimal internal and external actions at each stage such that a cumulative function of the rewards is maximized. We refer to this decision process as the *foresighted* cross-layer decision. By maximizing the cumulative reward, the wireless user is able to take into account the impact of the current actions on the future reward. Specifically, we assume that the wireless user will maximize the discounted accumulative reward, which is defined as

$$\sum_{k=0}^{\infty} (\gamma)^k R(\boldsymbol{s}^k, \boldsymbol{\xi}^k | \boldsymbol{s}^0) \tag{6}$$

where γ is a discounted rate with $0 \leq \gamma < 1$, and s^0 is the initial state. Unlike the formulation in [17] and [21], where the time-average reward is considered, we use a discounted accumulated reward with a higher weight on the current reward. The reasons for this are given as follows: 1) For delay-sensitive applications, the data need to be sent out as soon as possible to avoid missing delay deadlines; and 2) since a wireless user may encounter unexpected environmental dynamics in the future, it may care more about its immediate reward. Hence, this needs to be considered when determining the values of γ for a specific cross-layer problem.

The foresighted cross-layer optimization can be formulated using an MDP, which is defined as follows.

Definition 1 (MDP): An MDP is defined [11] as a tuple $M = \langle \mathcal{S}, \mathcal{X}, p, R, \gamma \rangle$, where $\mathcal{S}$ is a joint state space, i.e., $\mathcal{X}$ is a joint action space for each state, p is a transition probability function $\mathcal{S} \times \mathcal{X} \times \mathcal{S} \mapsto [0, 1]$, R is a reward function $\mathcal{S} \times \mathcal{X} \mapsto \Re$, and γ is the discounted factor.

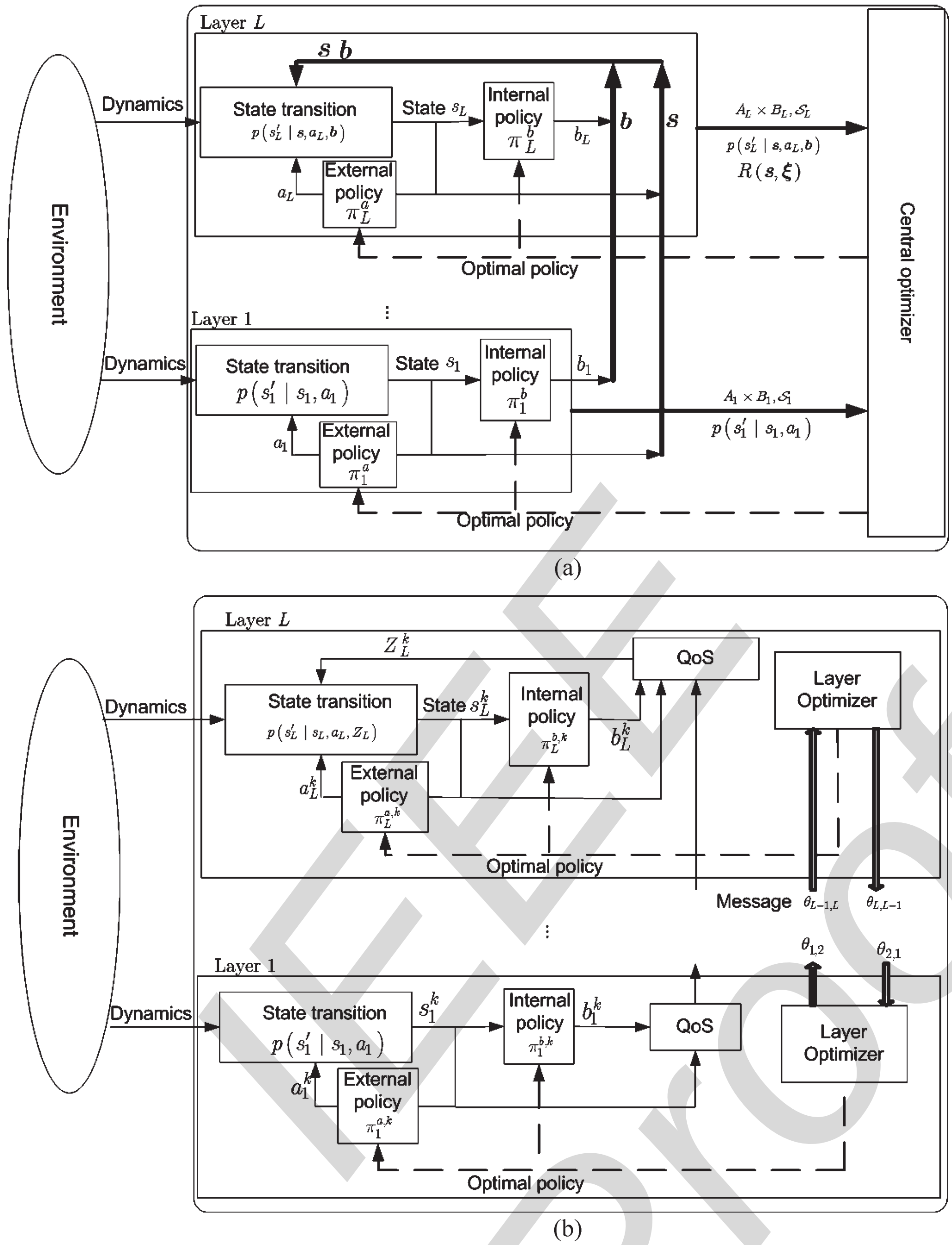

Fig. 4. Comparison of traditional cross-layer optimization framework and proposed cross-layer optimization framework. (a) Centralized cross-layer optimization framework. (b) Layered cross-layer optimization framework.

In our context, the joint state space is $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_L$, the joint action space is given by $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_L$, the transition probability is given by (2), and the reward function is given by (3).

III. Centralized Cross-Layer Solution and Its Disadvantages

A. Centralized Cross-Layer Optimization

Similar to [7], [15], and [17], the foresighted cross-layer optimization can be solved in a centralized way without noticing the structure of the cross-layer optimization. To solve the MDP problem, the central optimizer needs to know the following [see Fig. 4(a)]:

1) the state space at each layer;
2) the action space at each layer;
3) probability distribution describing the state transition (i.e., environmental dynamics);
4) state reward function of the states and performed actions.

Several centralized algorithms (e.g., the policy iteration, value iteration, and linear programming [12]) have been proposed to find the optimal policy that maximizes the discounted sum of future rewards. However, these algorithms neglect the layered structure of the cross-layer optimization.

In both the value-iteration and policy-iteration algorithms, the key step that needs to be performed at each iteration is solving the following optimization:

$$\max_{\xi \in \mathcal{X}} \left\{ R(\boldsymbol{s}, \boldsymbol{\xi}) + \gamma \sum_{s' \in \mathcal{S}} p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{\xi}) V(\boldsymbol{s}') \right\} \tag{7}$$

where $V(s')$ is a state-value function defined as the discounted reward that can be received when starting from state s'.

This optimization is called the DP operator [12]. In Section IV, we will decompose this key step into the layered DP operator such that the MDP problem can be solved in the manner that complies with the network architecture.

B. Limitations Associated With Centralized Cross-Layer Optimization

In the centralized optimization described in Section III-A, the actions at all the layers are simultaneously selected in the DP operator. However, this centralized optimization exhibits the following problems when implemented in the layered network architectures.

First, from Fig. 4(a), it is clear that the centralized cross-layer optimization solution requires each layer to forward the complete information about its protocol-dependent dynamics, as well as its internal and external action space and state space to the central optimizer. This centralized decision violates the current layered network architecture [3]. Specifically, a completely new interface between the central optimizer and all the layers is created. The central optimizer is allowed to access the internal variables at each layer, and hence, it is required to know the details about the protocols and algorithms deployed at each layer.

Second, the centralized optimization obliges each layer to take actions specified by the central optimizer. The layers have no freedom to adapt their own actions to the environmental dynamics that they experience. Hence, inherently, each layer loses the power to design its own protocol independently of other layers, which inhibits the upgrade of the various layers' protocols and algorithms.

IV. Layered Cross-Layer Optimization

To overcome the problems associated with the centralized cross-layer optimization that violates the layered network architecture, in this paper, we design a layered DP operator, which takes advantage of the structure of the cross-layer optimization discussed in Section II and allows each layer to autonomously optimize its own policy, based on the information exchanged with the other layers. This way, the layered architecture is preserved.

We will first discuss in Section IV-A how one layer can abstract the QoS that it provides to its upper layer and how it can compute the internal reward defined in (4). In Section IV-B, we discuss how the DP operator in (7) can be decomposed to comply with the layered architecture of the protocol stack and what messages are required to be exchanged among layers for this decomposition. In Section IV-C, we discuss how the internal and external actions are selected from the layered DP operator.

A. Quality of Service and Internal Reward Computation

In the layered network architecture, each layer selects its own internal actions, which, combined with the service provided by the lower layers, determine the QoS supported to the upper layer. In examples 1 and 2, the QoS levels computed in the PHY layer and provided to the MAC layer at the current time slot include the data throughput (in packets per second), the packet error rate, and the cost for transmitting one packet. The services are determined by the internal actions (e.g., modulation adaptation) and the state [i.e., signal-to-noise ratio (SNR) or SINR]. Based on the services provided by the PHY layer, the MAC layer can then adapt the ARQ scheme (e.g., the internal action) to compute the throughput, the packet error rate, and the cost of transmitting one packet (including the cost in the PHY layer), which are provided to the APP layer.

In this paper, we consider that each layer l provides to the upper layer the QoS, which includes the following: 1) the packet loss probability ε_l, which presents the probability that one packet at layer l is lost due to the imperfect transmission; 2) the transmission time per packet[3] τ_l at layer l; and 3) the transmission cost per packet υ_l at layer l. The QoS at layer l is denoted by $Z_l = (\varepsilon_l, \tau_l, \upsilon_l)$. The QoS Z_l is determined by the internal actions b_l and the QoS Z_{l-1} from the lower layer $l-1$, i.e., $Z_l = (\varepsilon_l, \tau_l, \upsilon_l) = (f_l^{\varepsilon}(s_l, b_l, Z_{l-1}), f_l^{\tau}(s_l, b_l, Z_{l-1}), f_l^{\upsilon}(s_l, b_l, Z_{l-1}))$, where f_l^{ε}, f_l^{τ}, and f_l^{υ} are the functions that map the current state s_l and internal action b_l at layer l and the QoS Z_{l-1} at layer $l-1$ into the packet loss rate ε_l, transmission time τ_l, and transmission cost υ_l, respectively. For notation simplicity, here, we denote the functions compactly as $Z_l = \vec{f}_l(s_l, b_l, Z_{l-1})$. The specific forms of these functions depend on the applications and network protocols. In Section V, we will give the specific forms of these functions for the example illustrated in Section II-A. Given the QoS at layer L, the application quality $g(s,b)$ only depends on the packet loss rate and transmission time and is then computed as $g(s,b) = g(s_L, \varepsilon_L, \tau_L)$. The internal cost $d(s,b)$ is computed as $d(Z_L) = \upsilon_L$. The internal reward function is computed as $R_{\text{in}}(s,b) = R_{\text{in}}(s_L, Z_L) = g(s_L, \varepsilon_L, \tau_L) - \lambda^b \upsilon_L$.

To compute the internal reward function $R_{\text{in}}(s_L, Z_L)$, layer L has to know all the QoS levels jointly determined by the states and internal actions at all the layers. Given the current state s of the wireless user, the set of the possible QoS levels at layer l is denoted by $\mathcal{Z}_l(s)$ and can be computed by enumerating all the combinations of internal actions available at each layer, i.e.,

$$\mathcal{Z}_l(\boldsymbol{s}) = \Big\{ Z_l | Z_l = \vec{f}_l(s_l, b_l, Z_{l-1}), \ldots, Z_1 = \vec{f}_1(s_1, b_1, \varnothing) \quad \forall b_1 \in B_1, \ldots, b_l \in B_l \Big\}. \tag{8}$$

Then, the set of QoS levels $\mathcal{Z}_l(s)$ at layer l captures the necessary information from the lower layers to compute the internal reward. In the layered network architecture, using the QoS set, layer $l+1$ does not need to know the actions and states of the lower layers. However, the size of the set $\mathcal{Z}_l(s)$ is often

[3]The transmission time per packet is the duration (time) for which the packet is being transmitted.

very large and, hence, leads to a high computational burden at the higher layers. In the following, we present a method to reduce the number of QoS levels to be provided to the upper layer without the performance loss.

We first define the relationship between two QoS levels at layer l using the following two terms: 1) "dominated" and 2) "Pareto equivalent."

Definition 2 (Dominated QoS): A QoS $Z_L = (\varepsilon_L, \tau_L, \upsilon_L)$ is dominated with respect to another QoS $Z'_L = (\varepsilon'_L, \tau'_L, \upsilon'_L)$ if $\varepsilon'_L \le \varepsilon_L$, $\tau'_L \le \tau_L$, $\upsilon'_L \le \upsilon_L$, and the equalities do not hold at the same time (i.e., $Z'_l - Z_l \le 0^4$ but $Z'_l \ne Z_l$). We denote this relationship as $Z'_l \overset{d}{\le} Z_l$.

Definition 3 (Pareto-Equivalent QoS): A QoS $Z_L = (\varepsilon_L, \tau_L, \upsilon_L)$ is Pareto equivalent to another QoS $Z'_L = (\varepsilon'_L, \tau'_L, \upsilon'_L)$, which is denoted by $Z'_l \overset{p}{=} Z_l$, if neither of the QoS levels is dominated by the other, i.e., $Z'_l \overset{d}{\le} Z_l$ or $Z_l \overset{d}{\le} Z'_l$.

Based on the relationship definition, we notice that for two QoS levels $Z'_L = (\varepsilon'_L, \tau'_L, \upsilon'_L)$ and $Z_L = (\varepsilon_L, \tau_L, \upsilon_L)$, if $Z'_L \overset{d}{\le} Z_L$, then $g(s_L, \varepsilon'_L, \tau'_L) \ge g(s_L, \varepsilon_L, \tau_L)$, since the lower packet loss probability and smaller transmission time per packet lead to more packets being transmitted and, hence, a higher application quality. Therefore, we have $R_{\text{in}}(s_L, Z'_L) \ge R_{\text{in}}(s_L, Z_L)$.

Furthermore, if layer $l-1$ provides two QoS levels Z_{l-1} and Z'_{l-1}, with $Z'_{l-1} \overset{d}{\le} Z_{l-1}$, then $Z'_l = \vec{f}_l(s_l, b_l, Z'_{l-1}) \le Z_l = \vec{f}_l(s_l, b_l, Z_{l-1})\ \forall s_l \in \mathcal{S}_l, b_l \in B_l$. That is, the functions f_l^ε, f_l^τ, and f_l^υ are nondecreasing functions of Z_{l-1}, given the current state $s_l \in \mathcal{S}_l$ and internal action $b_l \in B_l$. This can be explained as follows: When layer $l-1$ provides lower packet loss rate ε'_{l-1}, lower transmission time per packet τ'_{l-1}, and lower transmission cost per packet υ'_{l-1}, the internal action b_l at the current state s_l at layer l will result in lower packet loss rate ε'_l, lower transmission time per packet τ'_l, and lower transmission cost per packet υ'_l. For example, at the MAC layer, given a lower packet loss rate, a lower transmission time per packet, and a lower transmission cost per packet from the PHY layer, the same ARQ scheme (e.g., the same number of retransmission) will give a lower packet loss rate, a lower transmission time per packet, and a lower transmission cost per packet as well.

Hence, in our cross-layer design framework, the states and actions preserve the "domination" relationship of the QoS levels. That is, the states and actions in each layer have the following property.

Property 1 (Preservation of QoS): If $Z'_{l-1} \overset{d.}{\le} Z_{l-1}$, then $Z'_l = \vec{f}_l(s_l, b_l, Z'_{l-1}) \le Z_l = \vec{f}_l(s_l, b_l, Z_{l-1})\ \forall s_l \in \mathcal{S}_l, b_l \in B_l$.

The preservation of QoS means that the dominated QoS Z_l provided by layer l cannot result in a dominant QoS by performing any internal action at the upper layer. Hence, the dominated QoS Z_l should not be reported to the upper layer. Hence, the preservation of the domination relationship significantly reduces the amount of information exchanged by the lower layers to the upper layers. To describe the QoS levels that must be provided to the upper layer, we first define the optimal QoS frontier.

[4] $X \ge 0$ means that every component of is greater than or equal to 0.

Definition 4 (Optimal QoS Frontier): The optimal frontier of the possible QoS set $\mathcal{Z}_l(s)$ at layer l is the largest subset $\mathcal{Z}_l(s) \subseteq \mathcal{Z}_l(s)$ with each element satisfying the following condition: For any $Z_l \in \mathcal{Z}_l(s)$, there is no existing $\tilde{Z}_l \in \mathcal{Z}_l(s)$ such that $\tilde{Z}_l \overset{d.}{\le} Z_l$.

Hence, each layer l is only required to provide the QoS set $\mathcal{Z}_l(s)$ that represents the optimal frontier instead of all the possible QoS levels (i.e., $\mathcal{Z}_l$). The algorithm to construct the QoS frontier at layer l is presented in Algorithm 1.

Algorithm 1. Method for constructing the optimal QoS frontier $\mathcal{Z}_l$

Input: $\mathcal{Z}_{l-1}$, s_l, and B_l.
Initialize: $\mathcal{Z}_l = \varnothing$, flag $= 0$.
Loop 1: For each $b_l \in B_l$
Loop 2: For each $Z_{l-1} \in \mathcal{Z}_{l-1}$
    flag $= 0$;
    Compute $Z_l = \vec{f}_l(s_l, b_l, Z_{l-1})$.
Loop 3: For each $Z'_l \in \mathcal{Z}_l$
      If $Z'_l \overset{d}{\le} Z_l$
        flag $= 1$; break;
      endif
    endfor //loop 3
   if flag $== 0$
     $\mathcal{Z}_l = \mathcal{Z}_l \cup \{Z_l\}$.
   endif
  endfor //loop 2
 endfor // loop 1

B. Layered DP Operator

The key step of the cross-layer optimization is the DP operator. In the centralized formulation, the DP operator can only be performed in a centralized manner. In this section, we show how to decompose the DP operator into a layered DP with information exchange among the layers.

Considering the structure of the cross-layer optimization explored in Section II, we can rewrite the DP operator in (7) as follows:

$$\max_{\boldsymbol{a}\in\boldsymbol{A},\boldsymbol{b}\in\boldsymbol{B}} \left\{ \underbrace{g(\boldsymbol{s},\boldsymbol{b}) - \lambda^b d(\boldsymbol{s},\boldsymbol{b}) - \sum_{l=1}^{L} \lambda_l^a c_l(s_l, a_l)}_{R(\boldsymbol{s},\boldsymbol{\xi})} + \gamma \underbrace{\sum_{s'_1\in\mathcal{S}_1,\ldots,s'_L\in\mathcal{S}_L} p(s'_1|s_1,a_1)\cdots p(s'_L|\boldsymbol{s},\boldsymbol{b},a_L) V(s'_1,\ldots,s'_L)}_{\sum_{s'\in\mathcal{S}} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{\xi}) V(s')} \right\}. \quad (9)$$

TABLE I
DP OPERATOR AT EACH LAYER

Layer	DP operator at each layer
L	$V_{L-1}(s_1',\cdots,s_{L-1}') = \max_{\substack{a_L\in A_L,\\ Z_L\in\mathcal{Z}_L}}\left[R_{in}(s_L,Z_L) - \lambda_L^a c_L(s_L,a_L) + \gamma\sum_{s_L'\in\mathcal{S}_L} p(s_L' \mid s_L,Z_L,a_L)V(s_1',\cdots,s_L')\right]$
$l\in\{2,\ldots,L-1\}$	$V_{l-1}(s_1',\cdots,s_{l-1}') = \max_{a_l\in A_l}\left[-\lambda_l^a c_l(s_l,a_l) + \sum_{s_l'\in\mathcal{S}_l} p(s_l' \mid s_l,a_l)V_l(s_1',\cdots,s_l')\right]$
1	$V(\boldsymbol{s}) = \max_{a_1\in A_1}\left[-\lambda_1^a c_1(s_1,a_1) + \sum_{s_1'\in\mathcal{S}_1} p(s_1' \mid s_1,a_1)V_1(s_1')\right]$

TABLE II
MESSAGE EXCHANGES BETWEEN LAYERS FOR LAYERED DP OPERATOR

Layer	Upward Message $\theta_{l,l+1}$		Downward Message $\theta_{l,l-1}$	
L	$\varnothing$	None	$\{V_{L-1}(s_1',\cdots,s_{L-1}')\}$	Expected future reward at layer $L-1$
$l\in\{2,\ldots,L-1\}$	$\mathcal{Z}_l$	QoS level set provided to layer $l+1$	$\{V_{l-1}(s_1',\cdots,s_{l-1}')\}$	Expected future reward at layer $l-1$
1	$\mathcal{Z}_1$	QoS level set provided to layer 2	$\varnothing$	None

In the layered DP operator, we allow each layer to select its own internal and external actions to perform the optimization, as shown in (9). From the Appendix, the DP operator can be performed at each layer as shown in Table I, and the message exchanges between layers are shown Table II.

In this layered DP operator, the optimal external action $a_l^\ell(s_1',\ldots,s_{l-1}')$ is selected for each state $(s_1',\ldots,s_{l-1}')$ at the lower layers, and the optimal QoS level $Z_L^\ell(s_1',\ldots,s_{L-1}')$ depends on the state $(s_1',\ldots,s_{L-1}'$. Then, we have the following theorem.

Theorem 1: The state-value functions obtained in the layered DP operator satisfy the follow inequalities:

$$\begin{aligned}
&V_{L-1}\left(s_1',\ldots,s_{L-1}'\right)\\
&= \max_{\substack{a_L\in A_L,\\ Z_L\in\mathcal{Z}_L}}\Bigg[R_{\rm in}(s_L,Z_L) - \lambda_L^a c_L\left(s_L,a_L\right)\\
&\qquad + \gamma\sum_{s_L'\in\mathcal{S}_L} p\left(s_L'|s_L,Z_L,a_L\right)V\left(s_1',\ldots,s_L'\right)\Bigg]\\
&\geq R_{\rm in}\left(s_L,Z_L^*\right) - \lambda_L^a c_L\left(s_L,a_L^*\right)\\
&\quad +\gamma\sum_{s_L'\in\mathcal{S}_L} p\left(s_L'|s_L,Z_L^*,a_L^*\right)V(s_1',\ldots,s_L')\\
&\qquad\qquad \forall\left(s_1',\ldots,s_{L-1}'\right)
\end{aligned}\tag{10}$$

$$\begin{aligned}
&V_{l-1}\left(s_1',\ldots,s_{l-1}'\right)\\
&= \max_{a_l\in A_l}\left[-\lambda_l^a c_l(s_l,a_l) + \sum_{s_l'\in\mathcal{S}_l} p\left(s_l'|s_l,a_l\right)V_l\left(s_1',\ldots,s_l'\right)\right]\\
&\geq -\lambda_l^a c_l\left(s_l,a_l^*\right) + \sum_{s_l'\in\mathcal{S}_l} p\left(s_l'|s_l,a_l^*\right)V_l\left(s_1',\ldots,s_l'\right)\\
&\qquad \forall\left(s_1',\ldots,s_{l-1}'\right),\quad \forall l=1,\ldots,L-1
\end{aligned}\tag{11}$$

where the optimal external actions a_l^* $\forall l$ and optimal QoS level Z_L^* are obtained in the centralized DP operator.

Proof: The inequalities in (10) and (11) result from the fact that a_l^* $\forall l$ and Z_L^* represent the feasible solution to the layered DP operator, and hence, the state-value function obtained by the layered DP operator (which performs the maximization) is greater than or equal to the state-value function of any feasible solution. The detailed proof is omitted here due to space limitations. ■

Theorem 1 shows that the layered DP operator obtains higher state-value functions by performing the mixed actions at each layer, as explained below.

Similar to the centralized DP operator, at layer l, given the next state $(s_1',\ldots,s_{l-1}')$ and current state s, the optimal external action $a_l^\ell(s_1',\ldots,s_{l-1}')$ obtained in the layered DP operator is a pure action. However, the next state $(s_1',\ldots,s_{l-1}')$ is unknown at the current stage and has the probability distribution $p(s_1'|s_1,a_1^\ell), p(s_2'|s_2,a_2^\ell(s')),\ldots, p(s_{l-1}'|s_{l-1},a_{l-1}^\ell(s_1',\ldots,s_{l-1}'))$

TABLE III
MESSAGE EXCHANGE FOR INTERNAL AND EXTERNAL ACTION SELECTION

Layer	Upward Message $\theta_{l,l+1}$		Downward Message $\theta_{l,l-1}$	
L	$\varnothing$	None	$Z_{L-1}^{\dagger}$	The optimal QoS at layer $L-1$
$l \in \{2,\ldots,L-1\}$	$\arg\max_{s_1'} p\left(s_1' \mid s_1, a_1^{\dagger}\right)$ $\vdots$ $\arg\max_{s_l'} p\left(s_l' \mid s_l, a_l^{\dagger}\right)$	The optimal next states at layers $1,\cdots,l$	$Z_{l-1}^{\dagger}$	The optimal Qost at layer $l-1$
1	$\arg\max_{s_1'} p\left(s_1' \mid s_1, a_1^{\dagger}\right)$	The optimal next state	$\varnothing$	None

determined by the external actions performed at layers $1,\ldots,l-1$ and the environmental dynamics. Hence, the optimal external action $a_l^m(s)$ at layer l (computed without knowing the next states at layers $1,\ldots,l-1$) is a mixed action, whose elements $a_l^{\ell}(s_1',\ldots,s_{l-1}')$ have the same probability distribution as that of $(s_1',\ldots,s_{l-1}')$, i.e., $p(s_1'|s_1,a_1^{\ell}), p(s_2'|s_2,a_2^{\ell}(s'))$, $\ldots, p(s_{l-1}'|s_{l-1},a_{l-1}^{\ell}(s_1',\ldots,s_{l-1}'))$. Then, we can represent the mixed external action at layer l as

$$\begin{aligned} a_l^m(\boldsymbol{s}) &= \bigcup_{s_1'\in\mathcal{S}_1,\ldots,s_{l-1}'\in\mathcal{S}_{l-1}} \Big\{ p\left(s_1'|s_1,a_1^{\ell}\right), p\left(s_2'|s_2,a_2^{\ell}\left(s_1'\right)\right),\ldots \\ &\qquad p\left(s_{l-1}'|s_{l-1},a_{l-1}^{\ell}\left(s_1',\ldots,s_{l-1}'\right)\right) \\ &\qquad \circ\, a_l^{\ell}\left(s_l',\ldots,s_{l-1}'\right) \Big\} \end{aligned} \tag{12}$$

where the operator "$\circ$" indicates that action $a_l^{\ell}(s_1',\ldots,s_l')$ is performed with the probability $p(s_1'|s_1,a_1^{\ell}), p(s_2'|s_2, a_2^{\ell}(s_1')),\ldots,p(s_{l-1}'|s_{l-1},a_{l-1}^{\ell}(s_1',\ldots,s_{l-1}'))$. We use the union operator "$\bigcup$" to compactly represent the mixed action. Similarly, the optimal QoS level at layer L is given by

$$\begin{aligned} Z_L^m(\boldsymbol{s}) &= \bigcup_{s_1'\in\mathcal{S}_1,\ldots,s_{l-1}'\in\mathcal{S}_{l-1}} \Big\{ p\left(s_1'|s_1,a_1^{\ell}\right), p\left(s_2'|s_2,a_2^{\ell}\left(s_1'\right)\right),\ldots \\ &\qquad p\left(s_{L-1}'|s_{L-1},a_{L-1}^{\ell}\left(s_1',\ldots,s_{L-1}'\right)\right) \\ &\qquad \circ\, Z_L^{\ell}\left(s_1',\ldots,s_{L-1}'\right) \Big\}. \end{aligned} \tag{13}$$

In summary, compared with the centralized DP operator in which the pure action is chosen for each current state s, the optimal pure action $a_l^{\ell}(s_1',\ldots,s_{l-1}')$ in the layered DP operator is chosen for each current state s and next state $(s_1',\ldots,s_{l-1}')$. In other words, the layered DP operator takes into account the states' information at the next stage [i.e., $(s_1',\ldots,s_{l-1}')$] and performs the mixed actions based on the distribution of the states $(s_1',\ldots,s_{l-1}')$. Hence, the optimal mixed actions can improve the state-value function.

C. Internal and External Actions Selection

In this section, we will illustrate how the internal and external actions are selected without knowing the states at the next stage in the layered DP operator. From (12) and (13), we notice that the layered DP operator can only provide the mixed actions. The mixed action selection at each layer requires the transition probabilities at the lower layers. However, in our proposed layered network architecture, we do not allow the exchange of transition probabilities (i.e., the dynamics model at that layer), since this leads to significantly increased information exchange and requires each layer to access the internal parameters of other layers, thereby violating the OSI layer design. Instead, we restrict the optimal external action and optimal QoS-level selection as follows:

$$\begin{aligned} a_1^{\dagger} &= a_1^{\ell} \\ a_2^{\dagger} &= a_2^{\ell}\left(\arg\max_{s_1'} p\left(s_1'|s_1,a_1^{\dagger}\right)\right) \\ &\vdots \\ a_L^{\dagger} &= a_L^{\ell}\Bigg(\arg\max_{s_1'} p\left(s_1'|s_1,a_1^{\dagger}\right),\ldots \\ &\qquad \arg\max_{s_{L-1}'} p\left(s_{L-1}'|s_{L-1},a_{L-1}^{\dagger}\right)\Bigg) \\ Z_L^{\dagger} &= Z_L^{\ell}\Bigg(\arg\max_{s_1'} p\left(s_1'|s_1,a_1^{\dagger}\right),\ldots, \\ &\qquad \arg\max_{s_{L-1}'} p\left(s_{L-1}'|s_{L-1},a_{L-1}^{\dagger}\right)\Bigg). \end{aligned} \tag{14}$$

From (14), we note that the action and QoS-level selection does not require the information of transition probability but rather the states that maximize the transition probability. However, we should note that this selection is an approximation to the optimal mixed action and QoS level. To select external action and QoS level, the lower layer $l-1$ needs to provide the information $(\arg\max_{s_1'} p(s_1'|s_1,a_1),\ldots,\arg\max_{s_{l-1}'} p(s_{l-1}'| s_{l-1},a_{l-1}))$ to layer l. Given the approximated QoS level $Z_L^{\dagger}$, we obtain the internal action $b_L^{\dagger}$ and the QoS level $Z_{L-1}^{\dagger}$ at layer $L-1$, which generate the QoS level $Z_L^{\dagger}$. Similarly, given the QoS level $Z_l^{\dagger}$, layer l can find the internal action $b_l^{\dagger}$ and the QoS level $Z_{l-1}^{\dagger}$ for layer $l-1$. Hence, to select the internal action, layer l needs to provide the information $Z_{l-1}^{\dagger}$ to layer $l-1$.

D. Advantages of the Layered DP Operator

In this section, we highlight the advantages of the proposed layered DP operator compared with the centralized DP operator illustrated in Section III-A.

As discussed in Section III, the central optimizer is required to completely know the dynamics model (i.e., states, transition probability) and possible internal and external actions of all the layers that are protocol dependent. Hence, the mechanism of information exchange between the central optimizer and the layers is also protocol dependent. In the proposed algorithm, however, the centralized DP operator shown in (7) is decomposed into multiple layered DP operators, each of which is accordingly solved by one layer. From the layered DP operators shown in Table I and the message exchange between layers shown in Tables II and III, we note that our proposed layered DP operator has the following advantages.

First, to perform the layered DP operator, given the information exchanged between layers, each layer is only required to know its own internal and external actions and transition probabilities (corresponding to the dynamics models), but it is not required to know the actions and transition probabilities of other layers.

Second, the format (i.e., QoS optimal frontier for upward messages and the state-value functions for downward message) of the messages exchanged between layers is independent of the protocols deployed in each layer, while the content (i.e., QoS optimal frontier depends on the performed internal actions and state-value function depends on the external actions) of the messages characterizes the dynamics and performed actions at each layer.

Third, the internal and external actions are autonomously selected by each layer. Each layer has its own freedom to determine its own transmission strategies, which is desirable for the case that the protocols at various layers are designed by different companies. This way, upgrading the protocol at one layer does not affect other layers' protocol designs. Hence, our proposed cross-layer optimization solution preserves the current layered network architecture.

V. Simulation Results for the Illustrative Example

In this section, we use the example presented in Section II-A to illustrate the proposed cross-layer design framework. We first discuss the states, actions, and dynamics model used at each layer. Then, we provide simulation results to illustrate the merits of our proposed layered DP operator for cross-layer optimization.

A. APP Layer Models

In the APP layer, we assume that the wireless user deploys a delay-sensitive application (e.g., streaming "Mobile" video sequence with a 30-Hz frame rate at common intermediate format resolution). The data of the APP layer are packetized with an average packet length η in bits. Each packet is associated with a hard delay deadline, i.e., it will expire after $J\Delta T$ seconds (J stages) after they are ready for transmission. Then, we can define the state of the APP layer at stage k as $s_3^k = [s_{3,1}^k, \ldots, s_{3,J}^k]^T$, where $s_{3,j}^k \ (1 \le j \le J)$ is the number of packets waiting for transmission that have a remaining lifetime of j stages.

In the APP layer, the external action a_3^k (i.e., the source coding algorithms) determines the amount of packets arriving into the buffer at the beginning of stage k. For simplicity, we assume that a_3^k is equal to the average number of arriving packets. We denote by Y_3^k the random number of arriving packets. Then, $E[Y_3^k] = a_3^k$. The probability mass function of the random variable Y_3^k is assumed to be independent at each stage and is denoted by $\{P(Y_3^k = y|a_3^k),\ y \in \mathbb{N}\}$.

Given the QoS Z_3^k, the APP layer transmits the packets with lifetime 1. If there are no packets with lifetime 1 remaining for transmission, the packets with lifetime 2 will be transmitted, and so on. The number of packets that can be transmitted is computed as

$$n_3^k\left(Z_3^k\right) = \left\lfloor \frac{\Delta T}{\tau_3^k}\left(1 - \varepsilon_3^k\right) \right\rfloor. \tag{15}$$

The state at stage $k+1$ is updated as

$$\begin{bmatrix} s_{3,1}^{k+1} \\ \vdots \\ s_{3,j}^{k+1} \\ \vdots \\ s_{3,J}^{k+1} \end{bmatrix} = \begin{bmatrix} s_{3,2}^k - \max\left(n_3^k\left(Z_3^k\right) - s_{3,1}^k, 0\right) \\ \vdots \\ s_{3,j+1}^k - \max\left(n_3^k\left(Z_3^k\right) - \sum\limits_{m=1}^{j} s_{3,m}^k, 0\right) \\ \vdots \\ Y_3^k\left(a_3^k\right) \end{bmatrix}. \tag{16}$$

The state transition probability is computed as

$$p\left(s_3^{k+1}|s_3^k, a_3^k, Z_L^k\right) = \begin{cases} P\left(Y_3^k = y|a_3^k\right), & \text{if } s_3^{k+1} \text{ satisfies the relationship in (19) and } Y_3^k = y \\ 0, & \text{o.w.} \end{cases} \tag{17}$$

The application quality for the delay-sensitive application is defined here as

$$g\left(s_3^k, Z_3^k\right) = n_3^k\left(Z_3^k\right) - \lambda_g \max\left\{s_{3,1}^k - n_3^k\left(Z_3^k\right), 0\right\} \tag{18}$$

where λ_g is the parameter to tradeoff the received packets and lost packets. In this simulation, the internal action at layer 3 is empty, and hence, $Z_3^k = Z_2^k$. In this simulation, we reported the video quality in terms of peak SNR (PSNR) to indicate the real received video quality in Section V-E.

B. MAC Layer Model

For the TDMA-based channel access, the MAC layer requests spectrum access by performing the external actions a_2^k, which can be the resource requests values (e.g., taxation). The MAC layer state $s_2^k \in [0, 1]$ is the fraction of one time slot allocated in the current stage and quantized as a discrete value. By taking external action a_2^k, the transition probability is $p(s_2^{k+1}|s_2^k, a_2^k)$, and the external cost introduced is $c_2(s_2^k, a_2^k) = a_2^k$. For the A-CDMA-based channel access, the MAC layer does not need to request spectrum access since the whole spectrum band is available. Hence, the state at the MAC layer is $s_2^k = 1$, and the external action $a_2^k = \varnothing$. The corresponding

TABLE IV
PARAMETERS USED FOR THE SIMULATION AT THE VARIOUS LAYERS

Layer	Parameter	
PHY layer	Channel model parameters	$f_d = 50\text{Hz}$, $T_p = 0.8\text{ms}$, $s_1 \in [0.4, 0.8, \cdots, 4]\text{dB}$
	Modulation level	$m = 1, \ldots, 4$ (BPSK, QPSK, 8PSK, 16PSK)
	Power allocation	$A_1 = \{0.5,\ 1,\ 1.5,\ 2\}$
	Packet loss probability[1]	$\varepsilon_1 = 1 - (1 - BER)^{\eta}$ $BER(s_1, m) = erfc\left(\kappa \Gamma_{s_1} \sin\left(\frac{\pi}{2^m}\right)\right), \kappa = 283.5$
	Transmission time per packet	T_p / m
MAC layer	MAC state	$s_2 \in \{0.2, 0.4, \cdots, 1\}$
	Maximum retransmission limit	$N_{\max} = 5$
	Trade-off parameter λ_2^a	$\lambda_2^a = 0.1$
	Competition bids (external action)	$A_2 = \{0, 1\}$
APP layer	APP state	$s_3 \in \{(0,0), \ldots, (4,4)\}$
	Maximum life time	$J = 2$
	External action	$A_3 = \{1, 2, 3\}$
	Trade-off parameter λ_g	$\lambda_g = 0.1$

external cost is 0. The state transition probability is given by $p(s_2^{k+1} = 1 | s_2^k = 1, a_2^k = \varnothing) = 1$.

The wireless user can perform ARQ to enhance the QoS provided to the APP layer. Hence, the internal action can be $b_2^k \in \{0, \ldots, N_{\max}\}$, where $N_{\max}$ is the maximum retry limit, and b_2^k is the actual retry limit. Given the QoS provided from the PHY layer, e.g., $Z_1^k = (\varepsilon_1^k, \tau_1^k, v_1^k)$, if the internal action b_2^k is performed, then the QoS obtained in the MAC layer becomes

$$Z_2^k = (\varepsilon_2^k, \tau_2^k, v_2^k) = \left(\left(\varepsilon_1^k\right)^{b_2^k + 1}, \frac{\left(1 - \left(\varepsilon_1^k\right)^{b_2^k}\right) \tau_1^k}{\left(1 - \varepsilon_1^k\right) s_2^k}, \frac{\left(1 - \left(\varepsilon_1^k\right)^{b_2^k}\right) v_1^k}{\left(1 - \varepsilon_1^k\right)} \right). \quad (19)$$

It is easy to show that if $Z_1^k \overset{d.}{\leq} \tilde{Z}_1^k$, then $Z_2^k \overset{d.}{\leq} \tilde{Z}_2^k$ for any internal action b_2^k, which means that the preservation of QoS property defined in Section III is satisfied.

C. PHY Layer Model

Similar to the model used in [15] and [16], we assume that the received SINR experienced by a wireless user can be modeled as a discrete time FSMC. The state s_1^k in the PHY layer is the SINR. At each state, the wireless user is able to adapt its modulation and channel coding scheme (i.e., internal action) $b_1 \in B_1$ to determine the QoS level to support upper layer, where B_1 is the set of possible modulation and channel coding schemes. The wireless user also has to adapt the power allocation (i.e., external action) $a_1 \in A_1$ to determine the received SINR (i.e., the state at next time slot), where A_1 is the set of possible power allocations. The external cost is $c_1(s_1^k, a_1^k) = a_1^k$. As shown in [6], the PHY layer state can be determined by partitioning the possible received SINR into $r + 1$ disjoint regions $\mathbb{R}_0, \ldots, \mathbb{R}_r$ by boundary points $\Gamma_0, \ldots, \Gamma_{r+1}$, where $\mathbb{R}_i = [\Gamma_i, \Gamma_{i+1}]$ and $\Gamma_0 < \Gamma_1 < \cdots < \Gamma_{r+1}$. The PHY layer is said to be in the state $s_1^k = \tilde{\Gamma}_i$, where $\tilde{\Gamma}_i$ is the representative channel gain if the real channel gain is in the region $\mathbb{R}_{i-1}$. Similar to [16], the channel gain is assumed to be a Rayleigh-fading channel, which is denoted by Υ and is exponentially distributed with the following probability density function:

$$p_{\Upsilon}(\mu) = \frac{1}{\bar{\mu}(a_1)} \exp\left(-\frac{\mu}{\bar{\mu}(a_1)}\right), \qquad \mu \geq 0 \quad (20)$$

where $\bar{\mu}(a_1)$ is the average SINR, which is determined by the allocated transmission power a_1. The state transition at the PHY layer is computed as

$$p\left(s_1^{k+1} | s_1^k, a_1^k\right) = \begin{cases} \mathcal{N}(\tilde{\Gamma}_{i+1}) \frac{T_p}{\omega_i}, & s_1^k = \tilde{\Gamma}_i,\ s_1^{k+1} = \tilde{\Gamma}_{i+1} \\ \mathcal{N}(\tilde{\Gamma}_i) \frac{T_p}{\omega_i}, & s_1^k = \tilde{\Gamma}_i,\ s_1^{k+1} = \tilde{\Gamma}_{i-1} \\ 1 - \mathcal{N}(\tilde{\Gamma}_{i+1}) \frac{T_p}{\omega_i} - \mathcal{N}(\tilde{\Gamma}_i) \frac{T_p}{\omega_i}, & s_1^k = \tilde{\Gamma}_i,\ s_1^{k+1} = \tilde{\Gamma}_i \\ 0, & \text{o.w.} \end{cases} \quad (21)$$

where $\mathcal{N}(\mu) = (2\pi\mu/\bar{\mu}(a_1))^{1/2} f_d \exp(-\mu/\bar{\mu}(a_1)$, $\omega_i = \exp(-\Gamma_i/\bar{\mu}(a_1)) - \exp(-\Gamma_{i+1}/\bar{\mu}(a_1))$, T_p is the transmission time for one packet, and f_d is the maximum Doppler frequency.

D. Stage Reward Function

In this section, we present the explicit form of the internal reward function. In this example, the internal cost $d(s, b)$ is 0, and the internal reward function is given by $R_{\text{in}}(s_3^k, Z_3^k) = n_3^k(Z_3^k) - \lambda_g \max\{s_{3,1}^k - n_3^k(Z_3^k), 0\}$. It is easy to prove that the internal reward function $R_{\text{in}}(s_3^k, Z_3^k)$ is a nonincreasing function of Z_3^k, i.e., $R_{\text{in}}(s_3^k, Z_3^k) \geq R_{\text{in}}(s_3^k, \tilde{Z}_3^k)$ if $Z_3^k \overset{d.}{\leq} \tilde{Z}_3^k$. This property enables each layer only to report the QoS frontier to its upper layer, as discussed in Section IV-A.

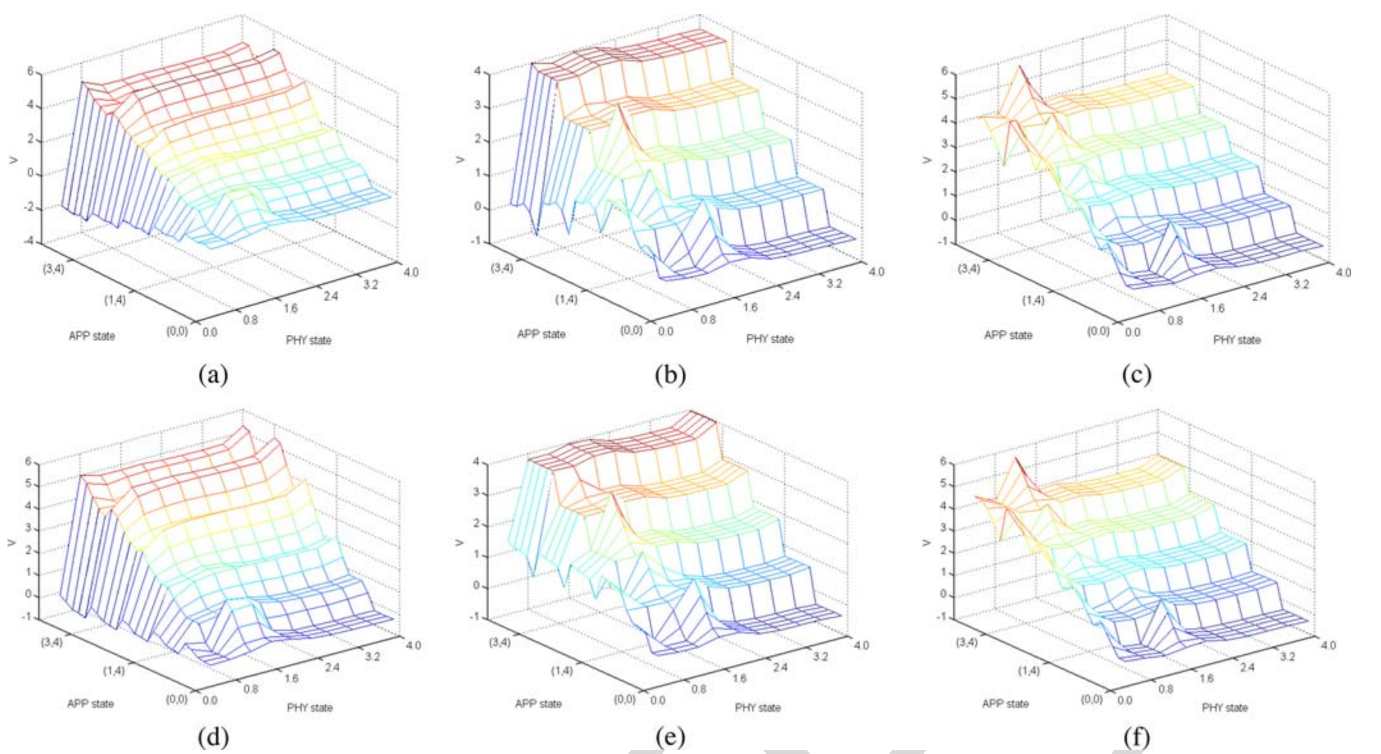

Fig. 5. State-value functions that resulted from the centralized value iteration and proposed layered value iteration. (a)–(c) State-value functions of the centralized DP operator when $s_2 = 0.1$, 0.6, and 1, respectively. (d)–(f) State-value functions of the layered DP operator when $s_2 = 0.1$, 0.6, and 1, respectively.

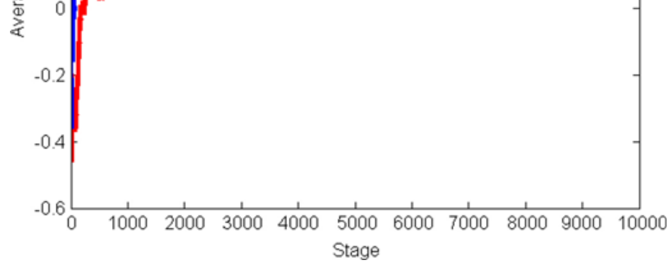

Fig. 6. Average reward obtained using the policies from a centralized DP operator and a layered DP operator.

E. Simulation Results Verifying the Optimality of the Layered DP Operator

We compare the optimal state-value functions obtained using the centralized DP operator and layered DP operator in the simulation presented in this section. Through this comparison, we will verify that the proposed layered DP operator also optimally solves the cross-layer optimization problem defined in Section II. The parameters for the APP, MAC, and PHY layers are shown in Table IV. The state-value functions $V^*(s)$ resulting from the centralized DP operator and proposed layered DP operator are shown in Fig. 5, where we observe that the state-value functions computed based on both algorithms are close, which means that our proposed layered DP operator achieves the performance close to the centralized one, i.e., near-optimally finding the cross-layer transmission strategies. To prove that, we also implement the policy obtained by both algorithms on line. The average rewards are depicted in Fig. 6, which demonstrates that the performance of both algorithms is the same when running for a long time. The transient performance of the layered DP operator in the beginning is worse than the central one, which is because we start from the state in which the centralized DP operator has good performance. The average PSNRs of the video sequence for both layered DP operator and centralized DP operator are 32.5 and 32.8 dB, respectively.

F. Myopic Versus Foresighted Optimization

In this simulation, we use the same parameters as in Section V-E. We compare the performance of the myopic cross-layer optimization (i.e., $\gamma = 0$) versus our proposed foresighted cross-layer optimization. We first run the value iteration to solve the cross-layer optimization off-line and apply the optimal policy on-line. Fig. 7 shows the average reward per stage for both the myopic policy and foresighted policy. The average reward obtained by the foresighted policy is 0.1850, while the average reward by the myopic policy is only -0.1050. Note that this reward value is computed based on the utility function given in Section V-D, and thus, other types of utility functions may have different values. The simulation results demonstrate

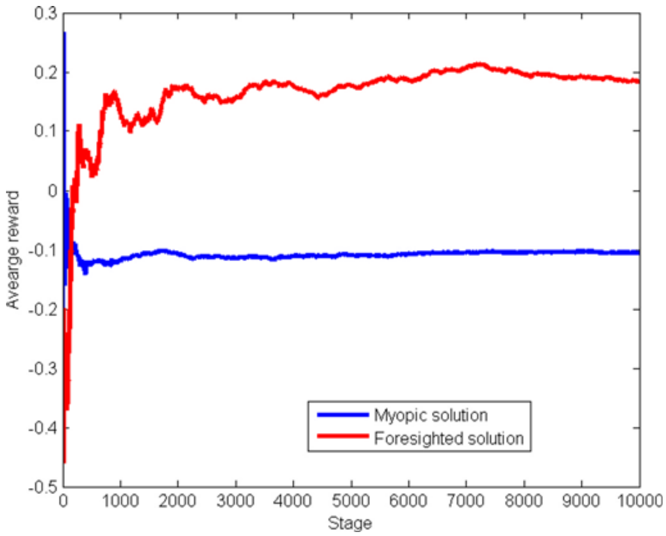

Fig. 7. Average reward per state for myopic cross-layer optimization and foresighted cross-layer optimization.

that the foresighted policy can achieve much better performance than the myopic policy.

VI. Conclusion

In this paper, we have formulated the dynamic cross-layer optimization problem as an MDP in which each layer interacts independently with the environment and experiences different dynamics. We proposed a layered DP operator to solve the cross-layer MDP problem. The layered DP operator allows each layer to perform its own optimization to find the optimal actions in an autonomous manner, given the information exchanges with other layers. Each layer is not required to know the protocols and algorithms implemented at other layers, thereby complying with the current layered network architecture and allowing network designers to build scalable, flexible, and upgradable protocols and algorithms at each layer of the OSI stack. An important topic for future work is the extension of this layered cross-layer framework by explicitly considering the constraints at each layer. Other important topics include implementing this framework for specific cross-layer problems, such as power-optimized transmission of media streams, real-time transmission over different types of channels, and wireless streaming for different video applications exhibiting various delay constraints.

Appendix

In the layered DP operator, the layers cooperatively perform the optimization shown in (9). Given the optimal frontier of QoS levels at layer L, the DP operator is rewritten as

$$\begin{aligned}\max_{a_1\in A_1,\ldots,a_L\in A_L,Z_L\in\mathcal{Z}_L}&\Bigg\{R_{\text{in}}\left(s_L,Z_L\right)-\sum_{l=1}^{L}\lambda_l^a c_l\left(s_l,a_l\right)\\&+\gamma\sum_{s_1'\in\mathcal{S}_1,\ldots,s_L'\in\mathcal{S}_L}p\left(s_1'|s_1,a_1\right),\ldots\\&\qquad p\left(s_L'|s_L,Z_L,a_L\right)V\left(s_1',\ldots,s_L'\right)\Bigg\}.\end{aligned}\tag{22}$$

Instead of simultaneously finding the optimal external actions and QoS levels as in the centralized DP operator, we optimize (22) layer by layer. We rewrite the DP operator in (22) as in (23), shown at the bottom of the page.

For each next state at the lower layers $(s_1',\ldots,s_{L-1}')$, the DP operator at layer L is

$$\begin{aligned}&V_{L-1}\left(s_1',\ldots,s_{L-1}'\right)\\&=\max_{\substack{a_L\in A_L,\\ Z_L\in\mathcal{Z}_L}}\Bigg[R_{\text{in}}(s_L,Z_L)-\lambda_L^a c_L(s_L,a_L)\\&\qquad+\gamma\sum_{s_L'\in\mathcal{S}_L}p\left(s_L'|s_L,Z_L,a_L\right)V\left(s_1',\ldots,s_L'\right)\Bigg].\end{aligned}\tag{24}$$

Then, the optimal external action $a_L(s_1',\ldots,s_{L-1}')$ and QoS level $Z_L(s_1',\ldots,s_{L-1}')$ depend on the next states of the lower layers. We should note that the optimization in (23) is not exactly the same as the one in (22), which were analyzed in Section IV-B. When layer L performs the optimization as in (24) for each state $(s_1',\ldots,s_{L-1}')$, it sends a message $\{V_{L-1}(s_1',\ldots,s_{L-1}')|\forall(s_1',\ldots,s_{L-1}')\}$ to layer $L-1$. At the same time, the DP operator is reduced as

$$\begin{aligned}\max_{a_1\in A_1,\ldots,a_{L-1}\in A_{L-1}}&\Bigg\{-\sum_{l=1}^{L-1}\lambda_l^a c_l\left(s_l,a_l\right)\\&+\sum_{s_1'\in\mathcal{S}_1,\ldots,s_{L-1}'\in\mathcal{S}_{L-1}}\prod_{l=1}^{L-1}p\left(s_l'|s_l,a_l\right)V_{L-1}\left(s_1',\ldots,s_{L-1}'\right)\Bigg\}.\end{aligned}\tag{25}$$

$$\begin{aligned}\max_{a_1\in A_1,\ldots,a_{L-1}\in A_{L-1}}&\Bigg\{-\sum_{l=1}^{L-1}\lambda_l^a c_l\left(s_l,a_l\right)+\sum_{s_1'\in\mathcal{S}_1,\ldots,s_{L-1}'\in\mathcal{S}_{L-1}}\prod_{l=1}^{L-1}p\left(s_l'|s_l,a_l\right)\\&\underbrace{\times\max_{\substack{a_L\in A_L,\\ Z_L\in\mathcal{Z}_L}}\left[R_{\text{in}}\left(s_L,Z_L\right)-\lambda_L^a c_L(s_L,a_L)+\gamma\sum_{s_L'\in\mathcal{S}_L}p\left(s_L'|s_L,Z_L,a_L\right)V\left(s_1',\ldots,s_L'\right)\right]\Bigg\}}_{\text{DP operator at layer } L}\end{aligned}\tag{23}$$

$$\max_{a_1\in A_1,\ldots,a_{L-2}\in A_{L-2}} \left\{ -\sum_{l=1}^{L-2}\lambda_l^a c_l\left(s_l,a_l\right) + \sum_{s_1'\in\mathcal{S}_1,\ldots,s_{L-2}'\in\mathcal{S}_{L-2}} \prod_{l=1}^{L-2} p\left(s_l'|s_l,a_l\right) \right.$$
$$\left. \times \underbrace{\max_{a_{L-1}\in A_{L-1}}\left[-\lambda_{L-1}^a c_{L-1}(s_{L-1},a_{L-1}) + \sum_{s_{L-1}'\in\mathcal{S}_{L-1}} p\left(s_{L-1}'|s_{L-1},a_{L-1}\right) V_{L-1}\left(s_1',\ldots,s_{L-1}'\right)\right]}_{\text{value iteration of layer } L-1} \right\} \tag{26}$$

Similar to (23), the optimization in (25) is rewritten in (26), shown at the top of the page.

For each next state at the lower layers $(s_1',\ldots,s_{L-2}')$, the DP operator at layer $L-1$ is

$$\begin{aligned} &V_{L-2}\left(s_1',\ldots,s_{L-2}'\right) \\ &= \max_{a_{L-1}\in A_{L-1}} \left[-\lambda_{L-1}^a c_{L-1}(s_{L-1},a_{L-1}) \right. \\ &\qquad + \sum_{s_{L-1}'\in\mathcal{S}_{L-1}} p\left(s_{L-1}'|s_{L-1},a_{L-1}\right) \\ &\qquad\qquad \left. \times V_{L-1}\left(s_1',\ldots,s_{L-1}'\right)\right]. \end{aligned} \tag{27}$$

Then, the message from layer $L-1$ to layer $L-2$ is $\{V_{L-2}(s_1',\ldots,s_{L-2}')|\forall(s_1',\ldots,s_{L-2}')\}$.

Similarly, for each state $(s_1',\ldots,s_l')$, layer l performs the DP operator as follows:

$$\begin{aligned} &V_{l-1}\left(s_1',\ldots,s_{l-1}'\right) \\ &= \max_{a_l\in A_l}\left[-\lambda_l^a c_l(s_l,a_l) + \sum_{s_l'\in\mathcal{S}_l} p\left(s_l'|s_l,a_l\right) V_l\left(s_1',\ldots,s_l'\right)\right]. \end{aligned} \tag{28}$$

We can interpret $V_{l-1}(s_1',\ldots,s_{l-1}')$ as a state-value function of state $(s_1',\ldots,s_{l-1}')$ seen at layer $l-1$. The message exchanged from layer l to layer $l-1$ is $\{V_{l-1}(s_1',\ldots,s_{l-1}')|\forall(s_1',\ldots,s_{l-1}')\}$.

At layer 1, the DP operator is

$$V(s) = \max_{a_1\in A_1}\left[-\lambda_1^a c_1(s_1,a_1) + \sum_{s_1'\in\mathcal{S}_1} p\left(s_1'|s_1,a_1\right) V_1\left(s_1'\right)\right]. \tag{29}$$

REFERENCES

[1] D. Bertsekas and R. Gallager, *Data Networks*. Upper Saddle River, NJ: Prentice–Hall, 1987.

[2] M. van der Schaar and S. Shankar, “Cross-layer wireless multimedia transmission: Challenges, principles, and new paradigms,” *IEEE Wireless Commun.*, vol. 12, no. 4, pp. 50–58, Aug. 2005.

[3] V. Kawadia and P. R. Kumar, “A cautionary perspective on cross-layer design,” *IEEE Wireless Commun.*, vol. 12, no. 1, pp. 3–11, Feb. 2005.

[4] X. Wang, Q. Liu, and G. B. Giannakis, “Analyzing and optimizing adaptive modulation coding jointly with ARQ for QoS-guaranteed traffic,” *IEEE Trans. Veh. Technol.*, vol. 56, no. 2, pp. 710–720, Mar. 2007.

[5] Q. Liu, S. Zhou, and G. B. Giannakis, “Cross-layer combining of adaptive modulation and coding with truncated ARQ over wireless links,” *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1746–1755, Sep. 2004.

[6] Y. J. Chang, F. T. Chien, and C. C. Kuo, “Cross-layer QoS analysis of opportunistic OFDM-TDMA and OFDMA networks,” *IEEE J. Sel. Areas Commun.*, vol. 25, no. 4, pp. 657–666, May 2007.

[7] D. Wu, S. Ci, and H. Wang, “Cross-layer optimization for video summary transmission over wireless networks,” *IEEE J. Sel. Areas Commun.*, vol. 25, no. 4, pp. 841–850, May 2007.

[8] R. Hamzaoui, V. Stankovic, and Z. Xiong, “Optimized error protection of scalable image bit streams,” *IEEE Signal Process. Mag.*, vol. 22, no. 6, pp. 91–107, Nov. 2005.

[9] F. Zhai, Y. Eisenberg, and A. K. Katsaggelos, “Joint source-channel coding for video communications,” in *Handbook of Image and Video Processing*, 2nd ed. A. Bovik, Ed. Amsterdam, The Netherlands: Elsevier. AQ2

[10] *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS), Draft Supplement,* IEEE Std. 802.11e/D5.0, Jun. 2003.

[11] M. L. Puterman, *Markov Decision Processes—Discrete Stochastic Dynamic Programming*. New York: Wiley, 1994.

[12] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Belmont, MA: Athena Scientific, 2005.

[13] F. P. Kelly, A. K. Maulloo, and D. K. Tan, “Rate control for communication networks: Shadow prices, proportional fairness, and stability,” *J. Oper. Res. Soc.*, vol. 49, no. 3, pp. 237–252, Mar. 1998.

[14] 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.

[15] T. Holliday, A. Goldsmith, and P. Glynn, “Optimal power control and source-channel coding for delay constrained traffic over wireless channels,” in *Proc. IEEE Int. Conf. Commun.*, May 2002, vol. 2, pp. 831–835.

[16] Q. Zhang and S. A. Kassam, “Finite-state Markov model for Rayleigh fading channels,” *IEEE Trans. Commun.*, vol. 47, no. 11, pp. 1688–1692, Nov. 1999.

[17] D. Djonin and V. Krishnamurthy, “MIMO transmission control in fading channels—A constrained Markov decision process formulation with monotone randomized policies,” *IEEE Trans. Signal Process.*, vol. 55, no. 10, pp. 5069–5083, Oct. 2007.

[18] Q. Wang and M. A. Abu-Rgheff, “Cross-layer signalling for next-generation wireless systems,” in *Proc. IEEE WCNC*, New Orleans, LA, Mar. 2003, vol. 2, pp. 1084–1089.

[19] A. T. Hoang and M. Motani, “Buffers and channel adaptive modulation for transmission over fading channels,” in *Proc. IEEE ICC*, May 2003, vol. 4, pp. 2748–2752.

[20] M. Goyal, A. Kumar, and V. Sharma, “Power constrained and delay optimal policies for scheduling transmission over a fading channel,” in *Proc. IEEE INFOCOM*, Apr. 2003, vol. 1, pp. 311–320.

[21] A. Ekbal, K. B. Song, and J. M. Cioffi, “QoS-constrained physical layer optimization for correlated flat-fading wireless channels,” in *Proc. IEEE ICC*, Jun. 2004, vol. 7, pp. 4211–4215.

[22] T. Dean and K. Kanazawa, “A model for reasoning about persistence and causation,” *Comput. Intell.*, vol. 5, no. 3, pp. 142–150, 1989.

[23] M. Alouini and A. J. Goldsmith, “Adaptive modulation over Nakagami fading channels,” *Wirel. Pers. Commun.*, vol. 13, no. 1/2, pp. 119–143, May 2000.

[24] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, “Layering as optimization decomposition: A mathematical theory of network architectures,” *Proc. IEEE*, vol. 95, no. 1, pp. 255–312, Jan. 2007.

[25] A. Reibman and M.-T. Sun, Eds., *Compressed Video Over Networks*. New York: Marcel Dekker, 2000.

[26] V. Bhaskar and K. H. Adjallah, "Finite-state Markov model for chi-square (central) SNR distribution in A-CDMA systems," in *Proc. Int. Conf. Commun., Internet Inform. Technol.*, Nov. 2005. AQ3

Fangwen Fu (S'08) received the bachelor's and master's degrees from Tsinghua University, Beijing, China, in 2002 and 2005, respectively. He is currently working toward the Ph.D. degree with the Department of Electrical Engineering, University of California, Los Angeles.

During the summer of 2006, he was an Intern with the IBM T. J. Watson Research Center, Yorktown Heights, NY. His research interests include wireless multimedia streaming, resource management for networks and systems, applied game theory, video processing, and analysis.

Mihaela van der Schaar (SM'04) received the Ph.D. degree from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 2001. AQ4

She is currently an Associate Professor with the Department of Electrical Engineering, University of California, Los Angeles. She has been an active participant in the International Standards Organization (ISO) Motion Picture Expert Group Standard since 1999, to which she has made more than 50 contributions. She was an Associate Editor for the *SPIE Electronic Imaging Journal*. She is a coeditor (with P. Chou) of the book *Multimedia Over IP and Wireless Networks: Compression, Networking, and Systems*. She is the holder of 28 granted U.S. patents, with several more pending.

Dr. van der Schaar is a member of the Technical Committee on Multimedia Signal Processing and the Technical Committee on Image and Multiple Dimensional Signal Processing of the IEEE Signal Processing Society. She was an Associate Editor for the IEEE TRANSACTIONS ON MULTIMEDIA. She is currently an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and the IEEE SIGNAL PROCESSING LETTERS. She received the National Science Foundation CAREER Award in 2004, the IBM Faculty Award in 2005, the Okawa Foundation Award in 2006, the Best IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY Paper Award in 2005 and 2007, and the Most Cited Paper Award from the *EURASIP Journal Signal Processing: Image Communication* between 2004 and 2006. She is also a recipient of three ISO recognition awards.