Collaborative Resource Exchanges for Peer-to-Peer Video Streaming Over Wireless Mesh Networks

Nicholas Mastronarde, Deepak S. Turaga, and Mihaela van der Schaar

***Abstract*— Peer-to-peer collaboration paradigms fundamentally change the passive way wireless stations currently adapt their transmission strategies to match available resources, by enabling them to proactively influence system dynamics through exchange of information and resources. In this paper, we focus on delay-sensitive multimedia transmission among multiple peers over wireless multi-hop enterprise mesh networks. We propose a distributed and efficient framework for resource exchanges that enables peers to collaboratively distribute available wireless resources among themselves based on their quality of service requirements, the underlying channel conditions, and network topology. The resource exchanges are enabled by the scalable coding of the video content and the design of cross-layer optimization strategies, which allow efficient adaptation to varying channel conditions and available resources. We compare our designed low complexity distributed resource exchange algorithms against an optimal centralized resource management scheme and show how their performance varies with the level of collaboration among the peers. We measure system utility in terms of the multimedia quality and show that collaborative approaches achieve ~50% improvement over non-collaborative approaches. Additionally, our distributed algorithms perform within 10% system utility of a centralized optimal resource management scheme. Finally, we observe 2-5 dB improvement in decoded PSNR for each peer due to the deployed cross-layer strategy.**

***Index Terms*— Collaborative resource management, multimedia P2P communication, multimedia transmission, peer-to-peer collaboration, resource exchanges for wireless P2P communication, wireless mesh networks.**

I. Introduction

EMERGING wireless multi-hop mesh networks are poised to enable a variety of delay-sensitive multimedia transmission applications, such as videoconferencing, on-demand distributed services and applications, surveillance, remote teaching and training [1]. However, the existing wireless infrastructure often provides dynamically varying resources with only limited support for the Quality of Service (QoS) required by these multimedia applications. Hence, multimedia streaming systems must accommodate variations in bandwidth and other network QoS parameters, introduced by the shared nature of the network and quality of the physical connections. Many of these emerging multimedia applications involve wireless peer-to-peer (P2P) transmission. The focus of this paper is on emerging P2P applications where wireless stations (referred to interchangeably as peers or nodes) located across an enterprise network transmit to each other delay-sensitive video bitstreams across multiple hops. In this distributed transmission scenario, peers need to proactively exchange information as well as collaborate to reserve resources, such 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 or resource allocation strategy deployed by one peer impacts and is impacted by the other peers. In the proposed P2P resource exchange paradigm, information about wireless resources (used, available etc.) and constraints (e.g., QoS requirements, channel conditions etc.) of the peers can be disseminated to all nodes, and used as available optimization criteria for their own communication subsystem (i.e. their own cross-layer optimization strategy).

Prior research on multimedia transmission over multi-hop wireless P2P networks has focused exclusively either on the cross-layer adaptation strategies, or on centralized resource management strategies. The problem of multi-hop wireless video streaming has only recently been studied in a variety of scenarios [2], [3]. However, the majority of this research does not consider mechanisms available at lower layers of the protocol stack and/or optimizes video transport using purely end-to-end metrics, thereby excluding the significant gains achievable by cross-layer design [5], [6], [7]. 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 re-transmissions, the loss rate and bandwidth of each link [7], as well as expected transmission time [8] can significantly impact the actual end-to-end network throughput. In summary, while significant contributions have been made to enhance the separate performance of the various OSI layers, no integrated and realistic cross-layer optimization framework exists for efficient multi-user multimedia transmission over multi-hop mesh networks.

Different centralized and distributed approaches have been adopted to solve the resource management problem for wireless networks. Centralized approaches solve the end-to-end routing and path selection problem as a combined optimization using multi-commodity flow [22] algorithms, as this ensures that the end-to-end throughput is maximized while constraints on individual link capacities are satisfied. In contrast, distributed approaches use fairness or incentive policies to resolve resource allocation issues in a scalable

Manuscript received January 1, 2006; revised August 31, 2006.

Nicholas Mastronarde and Mihaela van der Schaar are with the Electrical Engineering Department, University of California, Los Angeles (e-mail: nhmastro@ucla.edu; mihaela@ee.ucla.edu).

Deepak S. Turaga is with IBM T.J. Watson Research Center (e-mail: turaga@us.ibm.com).

Digital Object Identifier 10.1109/JSAC.2007.070111.

manner [26]-[29]. For instance, in [26], a new solution to the problem of engineering non-monetary incentives for edge based wireless access services was proposed which offers both higher throughput for bursty data and more stable allocation for real-time applications. However, previous research has not considered the benefits of dynamic resource and information exchanges among wireless peers.

In this paper we jointly consider and optimize resource exchanges among the peers sharing the same multi-hop enterprise wireless LAN (WLAN) infrastructure, as well as the cross-layer adaptation at each individual peer, such that the video quality (of simultaneously transmitted streams) is maximized. Unlike commercial P2P systems, where the incentive to collaborate is minimal and there are often free-riders, we investigate this resource collaboration and exchange in an enterprise network setting. In enterprise networks, the number of simultaneous users and variety of deployed applications can be controlled. Typical enterprise mesh networks consist of sparsely distributed wireless nodes where users may be authenticated, and their QoS requirements verified. Finally, the network topology typically changes slowly over time as users join and leave the network in a controlled manner.

We assume that the various nodes of the mesh network employ polling-based (reservation-based) admission control similar to that employed in the emerging 802.11e WLAN standard [10]. Furthermore, we consider that the WLAN uses multiple orthogonal channels for transmission thereby avoiding losses due to interference effects. However, for current wireless networks, investigating the performance degradation due to interference [31], [6], [32] forms an important area of our future research. We deploy an overlay network [11] where the users/applications are required to adhere to various imposed policies (for resource exchange, admission control, fairness, security etc.). An additional benefit of the overlay network is that it can convey information about the expected Signal to Interference-Noise Ratio (SINR), as well as the guaranteed bandwidth under the dynamically changing physical layer modulation at each wireless node. To improve the system utilization (number of admitted peers) as well as the QoS for admitted stations, we deploy scalable video coding schemes that enable each video flow (bitstream) to be divided into several sub-flows (layers) with different priority. This partitioning into multiple sub-flows also enables path diversity gains, since different sub-flows may be transmitted over different paths between the same source and destination peers.

Summarizing, to allow P2P collaboration in wireless networks, we propose a new way of architecting wireless multimedia communications systems by jointly optimizing the protocol stack at each station and resource exchanges among stations based on the underlying channel conditions, network topology, and requirements of each video sub-flow. We first design algorithms for *collaborative resource exchanges*, where given the average (or worst case) underlying channel conditions, source peers collaboratively decide how many sub-flows to admit, and which paths these sub-flows should be transmitted on. We design distributed algorithms for collaborative path partitioning and air-time reservation at intermediate nodes along the paths of the flow. We consider different levels of collaboration among participating peers, and examine the end-to-end performance in terms of the system utility and decoded video quality. Peers collaborate by exchanging information about selected paths, bandwidth requirements, and time reservation at intermediate nodes. Given the solution to the admission control, we then perform *dynamic cross-layer adaptation* at each peer. We modify the packet scheduling at the Application layer, the retransmission limit for each packet at the Medium Access Control (MAC) layer, and the underlying physical layer (PHY) modulation strategy, at each peer dynamically, given the underlying channel conditions.

This paper is organized as follows. We describe the notion of sub-flow, and describe the partitioning of a scalable video flow into multiple sub-flows in Section II. We also describe the quality-rate model that we use to solve the optimization problem. In Section III, we present distributed algorithms for collaborative path partitioning and air-time reservation to solve the collaborative resource exchange problem. In Section IV, we present the cross-layer adaptation strategies. We present simulation results in Section V and conclude in Section VI.

II. Video Sub-flows and Quality-Rate Model

A. Partitioning Scalable Bitstream Into Sub-Flows

It has been shown in [21] that, in 802.11e enabled wireless networks, partitioning a scalable video flow into several prioritized sub-flows can improve the number of admitted flows, as well as the overall quality received by the peers. Each sub-flow may be viewed as a separate *quality layer* of the scalable bitstream. In order to generate our flows we use a 3D wavelet video codec[1] that employs a spatio-temporal wavelet transform followed by embedded coding [14]. The bitstream is partitioned into quality layers based on the decoding distortion of different subbands and their deadlines etc. Examples of partitioning one video flow into multiple quality layers may be obtained from [23]. Each sub-flow has an associated priority based on its distortion impact and deadline constraint. Furthermore, there are dependencies among sub-flows, i.e. the gains from decoding some sub-flows are very limited if other sub-flows are not received. For instance, we observe drift errors when we decode enhancement layers without receiving the base layer. Although the distortion impacts of several sub-flows have dependencies, each sub-flow has its own admission control parameters and is admitted independently by the wireless network. The collection of sub-flows belonging to one end-to-end P2P connection is referred to as an *Aggregate Flow*.

Let us assume that there are N_p aggregate flows in the network. We label aggregate flow y, as set Ψ_y (with $1 \leq y \leq N_p$). We partition these N_p aggregate flows into N total sub-flows and label sub-flow x as f_x $1 \leq x \leq N$. Sub-flow f_x is a part of the aggregate flow Ψ_y, if $f_x \in \Psi_y$.

[1]Note that any other scalable or prioritized video coded bitstream can be used in the proposed framework. For instance, a state-of-the-art H.264 based coder can also be employed and various quality layers can be constructed through data-partitioning or hierarchical B-frames.

B. Quality-Rate Model: Utility for Collaborative Resource Exchange

The incremental quality provided by the decoding of an individual sub-flow (under the assumption of a fine-granular partition) is determined by the following quality-rate model:

$$\Delta Q_x = \lambda_x \log(B_x), \quad (1)$$

where λ_x is a flow-specific parameter that depends on the underlying video characteristics, encoding parameters etc. The bit-rate B_x corresponds to the rate requirement of the sub-flow. Our model is based on the work in [19] that models decoded quality for a scalable codec.[2] The quality parameter λ_x typically increases with the distortion impact of a sub-flow. The quality Q_y received by the aggregate flow Ψ_y may be computed as:

$$Q_y = Q_y^0 + \sum_{f_x \in \Psi_y} \lambda_x \log(B_x), \quad (2)$$

where Q_y^0 is a constant that also depends on the content characteristics, encoding parameters etc. Collaborative resource exchange algorithms are used to admit sub-flows and determine a path for each admitted sub-flow. If a sub-flow f_x is not admitted into the network, we do not add its contribution to the quality of the aggregate flow. Moreover, all sub-flows (enhancement layers) that depend on this sub-flow are not included in the contribution to the decoded quality. In particular, for sub-flow f_x we define indicator variable $\omega_x \in \{-1, 0, 1\}$ that takes value -1 before the decision on admitting the sub-flow is made, and takes values 1 and 0 based on whether the sub-flow is successfully admitted into the network, or not. Hence, the decoded video quality for aggregate flow Ψ_y based on all its sub-flows that are either admitted or denied admission is:

$$\widehat{Q}_y(P_y) = Q_y^0 + \sum_{\substack{f_x \in \Psi_y \\ \omega_x \neq -1}} \omega_x \lambda_x \log(B_x). \quad (3)$$

where P_y is the set of paths *selected* for its sub-flows (one path per sub-flow)[3] during path provisioning, as described in Section III. The *available* routing paths for any source-destination pair, from which set P_y is selected, may be determined using existing path discovery algorithms for multi-hop wireless networks or using protocols such as the Ad-hoc On-demand Multipath Distance Vector (AOMDV) [33] protocol.

We combine the individual quality experienced by each aggregate flow into one end-to-end system utility function. Different objectives can lead to various utility functions. For instance, we may wish to maximize the total quality (MTQ) received by all users, and the utility function may be written as a sum of the qualities of all N_p aggregate flows as

$$U_{MTQ} = \sum_{y=1}^{N_p} \widehat{Q}_y(P_y) = \sum_{y=1}^{N_p} \left[Q_y^0 + \sum_{\substack{f_x \in \Psi_y \\ \omega_x \neq -1}} \omega_x \lambda_x \log(B_x) \right]. \quad (4)$$

Alternately, instead of maximizing the total quality, we can also maximize the minimum quality (MMQ) experienced by any aggregate flow in the system. The MMQ utility may be written as:

$$U_{MMQ} = \min_y \left\{ \widehat{Q}_y(P_y) \right\} = \min_y \left\{ Q_y^0 + \sum_{\substack{f_x \in \Psi_y \\ \omega_x \neq -1}} \omega_x \lambda_x \log(B_x) \right\}. \quad (5)$$

We design resource exchange algorithms that maximize these utilities. We note that U_{MTQ} and U_{MMQ} serve as two different system optimization policies that resource managers might deploy for their system based on their application.

III. Collaborative Resource Exchanges

We allow peers to collaboratively determine which sub-flows they can admit into the network, and what paths they should select for each admitted sub-flow, based on the average underlying channel conditions, the bit-rate requirements of each sub-flow, and their contribution to the different utilities as defined in equations (4) and (5). Decisions on path provisioning and admission control are made only when a new flow arrives, an existing flow departs, or the network topology changes. In this paper, the path provisioning and routing of packets is not dynamically adapted. This limits the complexity and delays involved whenever performing sophisticated in-network processing. This is especially important for video streaming applications, where the delay associated with discovering new paths and performing dynamic routing are often not tolerable.

In Section III.A, we present our system specification. Our algorithms for collaborative admission control and path provisioning are presented in Sections III.B-III.D. Finally, in Section III.E, we quantify the worst-case overheads of the information exchanges needed for these algorithms.

A. System Specification

Let us consider an intermediate node v_a and assume that it receives a set of sub-flows from its neighbors, where each sub-flow f_x has an associated rate B_x, and experiences channel conditions based on which neighboring node it is transmitted from. Based on the experienced channel condition (and the modulation strategy), flow f_x is transmitted over a link with an expected goodput $\overline{G}_x^a$ (that depends on the PHY layer transmission rate T_x^a and the packet loss probability ϵ_x^a experienced by the link).

[2]In general we can use a different quality-rate model without modifying our optimization algorithms. Other such quality-rate models for scalable video have been presented in [24], [25] and they could have been also deployed for the proposed resource exchange framework.

[3]With a fine-granular partitioning into sub-flows, the overheads associated with transmitting a sub-flow over multiple paths are likely to outweigh the gains from additional path diversity.

We assume that the channel exhibits independent packet losses.[4] Let the neighboring node transmitting packets of sub-flow f_x to node v_a select physical layer mode θ_x^a. The corresponding bit error rate $e\left(\theta_x^a\right)$ over this hop is [13]:

$$e\left(\theta_x^a\right) = \frac{1}{\left(1 + e^{\mu(s-\delta)}\right)}. \tag{6}$$

where s is the underlying SINR, and μ and δ are constants. The PHY layer transmission rate T_x^a is the maximum achievable rate for a given mode θ_x^a. Then, the packet error probability ϵ_x^a that a packet of size L_x experiences during transmission [4] is:

$$\epsilon_x^a\left(L_x, \theta_x^a\right) = 1 - \left(1 - e\left(\theta_x^a\right)\right)^{L_x}. \tag{7}$$

Hence, the expected goodput experienced by sub-flow f_x, when transmitted to node v_a is

$$\overline{G}_x^a = \left(1 - \epsilon_x^a\left(L_x, \theta_x^a\right)\right) T_x^a\left(\theta_x^a\right). \tag{8}$$

We assume that the service interval t_{SI} for each node is partitioned into a listening service interval $t_{SI}^{(RX)}$ and a transmission service interval $t_{SI}^{(TX)}$. In this paper we assume that t_{SI}, $t_{SI}^{(RX)}$ and $t_{SI}^{(TX)}$ are determined a priori and are the same at each node. Furthermore, we assume that $t_{SI}^{(RX)} \ll t_{SI}^{(TX)}$ such that congestion exists mainly because of the limited listening time.

Each receiving node polls its neighbors for a different amount of time (within its fixed listening service interval $t_{SI}^{(RX)}$) based on the number of sub-flows competing for this air-time, their priorities, and rate requirements. In particular, a fraction $r_x^a (0 \le r_x^a \le 1)$ of the listening time of node v_a is allocated to f_x. Hence, the equivalent rate of flow f_x expected to arrive at node v_a is $\min\left\{\overline{G}_x^a r_x^a \frac{t_{SI}^{(RX)}}{t_{SI}}, B_x\right\}$. Clearly, if $\overline{G}_x^a r_x^a \frac{t_{SI}^{(RX)}}{t_{SI}} < B_x$ then the sub-flow f_x does not receive its desired rate requirement. When sub-flows are fine granular partitions of the aggregate flow, it is often better to not admit a sub-flow that does not receive its desired rate. Consequently, this subflow should not be admitted over this link or over any path containing this link.

B. Sub-Flow Admission Control

Let us first assume that for each aggregate flow Ψ_y we have a set of paths P_y, with path $p_x \in P_y$ assigned to sub-flow $f_x \in \Psi_y$. Given this selection, we need to solve the air-time reservation per node and determine which sub-flows can be admitted into the network such that we maximize the desired utility. The order in which sub-flows should be admitted depends on their impact on the end-to-end utility function. From (1), the impact of sub-flow f_x is directly related to its quality parameter λ_x, its bit-rate B_x, and the *quality layer* it belongs to.

[4]In [18] it was shown that the Gilbert-Elliot model for bit-errors in wireless channels generates packet loss patterns that are statistically similar (within ~0.4%) to naïve independent bit-by-bit simulations. Using this argument we translate the independent loss model to an independent packet loss probability and generate results using one Bernoulli experiment per packet. This also reduces the computational complexity of our experiments significantly (by a factor of 100).

TABLE I
PSEUDOCODE FOR SUB-FLOW ADMISSION CONTROL FOR A GIVEN SET OF PATHS (ONE PATH PER SUB-FLOW)

1. if maximizing U_{MTQ}
Sort all flows f_x in descending order of the fraction $\frac{\lambda_x}{B_x}$.
else if maximizing U_{MMQ}
Group flows into quality layers starting with the base layer. Within each quality layer sort flows in descending order of the fraction $\frac{\lambda_x}{B_x}$.
end
2. For all nodes v_a set $\rho^a = 1$
*3. For all flows set indicator variable $\omega_x = -1$ /*undefined value: sub-flow not considered yet*/*
4. for x =1:N (in order of the sorted list)
if $\omega_x = -1$
Traverse path p_x to determine if p_x can support f_x, i.e. for each node $v_b \in p_x$
if $\frac{B_x}{\overline{G}_x^b} \frac{t_{SI}}{t_{SI}^{(RX)}} > \rho^b$
*/*This node (and therefore path) cannot support bit-rate B_x.*/*
Drop sub-flow f_x. Set $\omega_x = 0$
Drop other sub-flows that depend on it. Set corresponding $\omega_z = 0$ for f_z if it depends on f_x.
Goto 4.
end
end
*/*This path can support the flow*/*
Traverse path p_x to reserve appropriate time fractions at intermediate nodes, i.e. for each node $v_b \in p_x$
Set $r_x^b = \frac{B_x}{\overline{G}_x^b} \frac{t_{SI}}{t_{SI}^{(RX)}}$
Set the indicator variable $\omega_x = 1$
$\rho^b \leftarrow \rho^b - r_x^b$
end
end
end
5. Determine end-to-end utility U_{MTQ} or U_{MMQ}

a. In order to maximize U_{MTQ}, we admit sub-flows in descending order of the fraction $\frac{\partial(\Delta Q_x)}{\partial B_x} = \frac{\lambda_x}{B_x}$ as it represents the tradeoff between quality (proportional to $\log B_x$) and resources allocated (proportional to B_x). With this order, we may admit more sub-flows from one aggregate flow than the other, potentially leading to a large variation in quality across the different aggregate flows.

b. In order to maximize U_{MMQ}, it is essential that the variation in quality across the different aggregate flows is minimized. Hence we admit sub-flows quality layer by quality layer, i.e. we first admit *quality layer* 1 sub-flows for all aggregate flows before we admit any *quality layer* 2 sub-flows. Within the same *quality layer* we admit sub-flows in descending order of the fraction $\frac{\lambda_x}{B_x}$.

For each sub-flow, in order of this sorted list, we determine the required time reservation fraction at all intermediate nodes along its path. Hence, at each intermediate node v_a along path p_x, we determine the desired time fraction $\frac{B_x}{\overline{G}_x^a} \frac{t_{SI}}{t_{SI}^{(RX)}}$. If that is less than the remaining available listening time fraction $\rho^a (0 \le \rho^a \le 1)$, given the reservations for sub-flows with higher priority, then the flow is admitted at node v_a. If the flow can be admitted at all nodes along its path then it is admitted into the network, and the time reservation fractions at each intermediate node are modified appropriately, i.e. the fraction $\frac{B_x}{\overline{G}_x^a} \frac{t_{SI}}{t_{SI}^{(RX)}}$ is reserved at node v_a, and the available listening time fraction ρ^a is decreased. If the bandwidth requirement for the sub-flow is violated at any intermediate node, then we discard it from the list along with all other sub-flows that depend on it. The pseudocode for this sub-flow admission control algorithm to determine which flows receive their bandwidth end-to-end, given a selection of one path per sub-flow, is presented in Table 1.

C. Centralized Algorithm for Optimal Sub-Flow Admission Control and Path Provisioning

We may use the sub-flow admission control algorithm in Table 1, in conjunction with a centralized algorithm, to determine the optimal path for each sub-flow. An exhaustive approach to determining the optimal set of paths is to consider all sets of possible path-sub-flow pairings (with one path per sub-flow) and then pick the set that maximizes the appropriate end-to-end utility. Clearly, if there are a total of K_x paths per sub-flow, the total complexity of an exhaustive search is $\prod_{x=1}^{N} K_x$ (in terms of the number of path combinations). Since the total number of subflows, and correspondingly paths per sub-flow can be large, an exhaustive search can be prohibitively expensive. Note that this complexity may be reduced through intelligent pruning of paths based on the given static channel conditions.

D. Distributed Algorithms for Optimal Sub-Flow Admission Control and Path Provisioning

Alternately, we design distributed solutions to the wireless P2P admission control and path provisioning. In these approaches, we solve the path provisioning for each sub-flow individually, in a greedy manner. We decompose the joint optimization into a set of successive path selection problems, where in each step we determine one path per sub-flow given the paths selected for other sub-flows. Using this greedy approach, at each stage we need to consider only K_x paths and the overall complexity decreases significantly to $\sum_{x=1}^{N} K_x$ (in terms of the number of path combinations). Hence, the optimization can be implemented in a distributed manner at each source peer. The success of such a distributed approach depends heavily on the amount of collaboration among source peers. We consider two sets of distributed algorithms, collaborative and non-collaborative. In the collaborative approach, source peers exchange information about the relative importance (in terms of the quality layer and the quality-rate slopes) of each sub-flow and perform the optimization sub-flow by sub-flow in order of the appropriate sorted list. In the non-collaborative approach each source peer performs the optimization independently. In both cases, source peers are aware of the air-time reservations at all intermediate nodes, made for all sub-flows.

1) Collaborative Distributed Path Provisioning Algorithm: Source peers collaboratively determine the sorted list of sub-flows in decreasing order of their contribution to the utility function being considered. Hence, sub-flows are sorted in descending order of the fraction $\frac{\lambda_x}{B_x}$ for maximizing U_{MTQ}. For maximizing U_{MMQ}, sub-flows are first grouped based on which quality layer they correspond to, and then sorted within each quality layer in descending order of the fraction $\frac{\lambda_x}{B_x}$. For each sub-flow in this sorted order, the corresponding source peer determines if there exists any path that provides it with its desired bandwidth. If no such path exists, the source node discards this sub-flow and all other sub-flows that depend on it. If multiple paths exist that provide the required end-to-end bandwidth for a sub-flow, the source node selects the path that leads to the smallest amount of congestion (in terms of contention for shared resources at different nodes). The congestion at any node v_a in the network is measured in terms of the fraction of its listening time that is already reserved for sub-flows that pass through it. The larger this total fraction is, the greater the congestion at the node. Hence, the congestion at node v_a may be written as $1 - \rho^a$, where ρ^a is the fraction of available listening time. The source peer selects among multiple paths that can provide the required end-to-end bandwidth for the sub-flow, based on two different congestion metrics:[5]

a. *Bottleneck air-time congestion*. Bottleneck congestion for any path is defined as the maximum congestion experienced at any node along it. Specifically, for any node v_a that lies along the examined path, we compute $1 - \rho^a$ (assuming that the current sub-flow is admitted, and given information about other sub-flows already admitted). The bottleneck congestion ξ_x^i for path p_x^i (the i-th path of sub-flow x) may be defined as $\xi_x^i = \max_{v_a \in p_x^i} \{1 - \rho^a\}$, where we use $v_a \in p_x^i$ to represent all nodes v_a that lie along the path p_x^i.

b. *Mean end-to-end air-time congestion*. Mean congestion for any path is the mean congestion experienced across all nodes that lie along it. The mean congestion ϕ_x^i for path p_x^i is defined as $\phi_x^i = \frac{1}{|p_x^i|} \sum_{v_a \in p_x^i} (1 - \rho^a)$, where $\left|p_x^i\right|$ is the number of nodes in the path. The source node then selects the path with the smallest end-to-end congestion given the chosen congestion metric. Bottleneck end-to-end path congestion is preferable when the network topology contains bottleneck nodes (i.e. nodes through which several sub-flows must necessarily pass, as in a star topology). Otherwise, it is preferable to use mean end-to-end path congestion metrics. Note that, regardless of the congestion metric, the sub-flow under consideration will *always* be admitted if there exists at least one path with enough resources to support it. The congestion metric, however, may significantly influence how subsequent sub-flows are admitted. The pseudocode for this algorithm is presented in Table II.

2) Non-Collaborative Distributed Path Provisioning Greedy Algorithm: We design a non-collaborative distributed path provisioning greedy algorithm to compare with the collaborative solution. In the non-collaborative path provisioning, each source peer determines the paths for all of its sub-flows (in order of their importance) before other peers are allowed to determine paths for their sub-flows. Hence, collaborative sorting of all sub-flows is not performed; instead, each source peer sorts only its sub-flows in the order of decreasing $\frac{\lambda_x}{B_x}$ (which is the same as sorting into quality layers). Source peers then perform the optimization independently and non-collaboratively in the order of their arrival. The simulation results in Section V confirm that the non-collaborative approach is not capable of admitting as many high priority flows as the collaborative approach.

[5]Other measures of congestion such as intersection weighted congestion (where each node's reserved fraction is weighted by the number of flows that pass through it) may also be considered.

TABLE II
PSEUDOCODE FOR COLLABORATIVE DISTRIBUTED PATH PROVISIONING ALGORITHM

1. if maximizing U_{MTQ}
  Source peers sort sub-flows f_x *in descending order of the fraction* $\frac{\lambda_x}{B_x}$.
  else if maximizing U_{MMQ}
  Source peers sort sub-flows into quality layers starting with the base layer. Within each quality layer source peers sort sub-flows in descending order of $\frac{\lambda_x}{B_x}$.
  end
2. for all nodes v_a *set* $\rho^a = 1$
3. For all flows set indicator variable $\omega_x = -1$ */*undefined value: sub-flow not considered yet*/*
4. For x=1:N (in order of the sorted list)
    */*The corresponding source peer performs this optimization*/*
    if $\omega_x = -1$
        Set found_one_path = false;
        ***Next**: for each available path* p_x^i *for sub-flow* f_x
            Set congestion fractions $\xi_x^i = \infty$ *and* $\bar{\varphi}_x^i = \infty$
            Traverse path p_x^i *to determine if path can support sub-flow, i.e. for each node* $v_b \in p_x^i$
                if $\frac{B_x}{\bar{G}_x^b}\frac{t_{SI}}{t_{SI}^{(RX)}} > \rho^b$
                    */*This path cannot support bit-rate* B_x*.*/*
                    Goto Next.
                end
            end
            */*This path can support the flow*/*
            Set found_one_path = true
            if minimizing bottleneck congestion
                Set bottleneck congested reservation fraction for this path $\xi_x^i = 0$
                Traverse path p_x^i *to compute the largest reserved time fraction, i.e. for each node* $v_b \in p_x^i$
                    Set $\xi_x^i \leftarrow \max\left(\xi_x^i, (1-\rho^b) + \frac{B_x}{\bar{G}_x^b}\frac{t_{SI}}{t_{SI}^{(RX)}}\right)$
                end
            else
                Set mean air-time reservation fraction for this path $\bar{\varphi}_x^i = 0$
                Traverse path p_x^i *to compute the largest reserved time fraction, i.e. for each node* $v_b \in p_x^i$
                    Set $\bar{\varphi}_x^i \leftarrow \bar{\varphi}_x^i + (1-\rho^b) + \frac{B_x}{\bar{G}_x^b}\frac{t_{SI}}{t_{SI}^{(RX)}}$
                end
                Set $\bar{\varphi}_x^i \leftarrow \frac{\bar{\varphi}_x^i}{|p_x^i|}$, *i.e. divide by the number of nodes in the path*
            end
        end
        if found_one_path = false
            Drop sub-flow f_x. *Set* $\omega_x = 0$
            Drop all other sub-flows that depend on it. Set corresponding $\omega_z = 0$ *for* f_z *if it depends on* f_x
            Goto 3.
        else
            Set indicator variable $\omega_x = 1$
            if minimizing bottleneck congestion
                Select path p_x^i *with smallest congestion reservation fraction* ξ_x^i.
            else
                Select path p_x^i *with smallest mean reservation fraction* $\bar{\varphi}_x^i$.
            end
            Traverse path p_x^i *to reserve appropriate time fractions at intermediate nodes, i.e. for each node* $v_b \in p_x^i$
                Set $r_x^b = \frac{B_x}{\bar{G}_x^b}\frac{t_{SI}}{t_{SI}^{(RX)}}$
                $\rho^b \leftarrow \rho^b - r_x^b$
            end
        end
    end
end
5. Determine end-to-end utility U_{MTQ} *or* U_{MMQ}

E. Information Exchange Overheads

Collaborative distributed resource exchanges operate under the assumption that certain control information can be propagated throughout the network using the overlay infrastructure [11]. In this section we develop a worst-case bound on the total overhead introduced by information exchange between peer nodes.

During path provisioning three different sets of messages need to be exchanged. The first set consists of messages from each source peer containing the sub-flow bit-rates B_x, priorities λ_x, and corresponding quality layer tags in each aggregate flow. The second set of messages includes information about the link and channel conditions (SINR). The third set consists of messages about the path p_x selected for each sub-flow f_x and the modified available listening time fraction ρ^a for every node v_a in p_x. We define $\overline{O}_0$ as the average overhead incurred when transmitting one information packet over one hop. It is important to note that, $\overline{O}_0$ is likely to be small due to the small size of information packets.

Consider a P2P scenario where N_p aggregate flows (and N total sub-flows) are pending admission. Let the maximum number of paths available for any flow be K and the maximum number of nodes along any path be M. Let V be the maximum number of hops between any two nodes ($V = M - 1$). A worst-case bound on the overhead costs for dispersing sub-flow bitrate and priority requirements, $O_{flowreq}$, is,

$$O_{flowreq} \leq \overline{O}_0 V N_p (N_p - 1) \tag{9}$$

since each peer needs to send messages to $N_p - 1$ other peers over at most V hops. The above assumes each packet contains rate and priority information for all sub-flows of one aggregate flow. Overhead costs to collect channel information $O_{channel}$ may be bounded by:

$$O_{channel} \leq \overline{O}_0 V^2 K N_p. \tag{10}$$

since each source peer needs information about at most V hops for at most K paths, and each of these messages has to traverse at most V hops to get to the peer. Finally, overhead costs for exchanging path provisioning information $O_{pathprov}$, may be bounded by,

$$O_{pathprov} \leq \overline{O}_0 V N \left(N_p - 1\right) M \tag{11}$$

since information about each of the at most M nodes along the path selected for each subflow needs to be communicated to the $N_p - 1$ other source peers.

The above derived bounds are worst case and assume no optimization in terms of message forwarding, multicast message passing [11], incremental information exchange etc. Significant savings in the total overhead may be achieved by considering these schemes, but these are beyond the scope of this paper. However, even with these worst case assumptions, the total overhead is limited, and does not significantly limit the scalability of the algorithms.

In a dynamic scenario, if link conditions change beyond a preset threshold, some sub-flows may require re-routing. In the worst-case, distributing the new network information for re-routing requires MN_p message transmissions. Additionally, re-routing requires the path provision overhead of (11). We note that, re-routing may be done in the background while the higher priority sub-flows continue to be transmitted over their previously assigned paths. Consideration of such dynamic routing scenarios based on the priorities of the flows is a topic for future research.

IV. CROSS-LAYER ADAPTATION STRATEGIES AT THE WIRELESS PEERS

During collaborative resource exchange, the peers determine which sub-flows can be admitted and what path each sub-flow should take. However, this decision is made based on static (average or worst case) channel conditions and the rate requirements of each sub-flow. In real transmission scenarios, these underlying channel conditions vary based on the SINRs experienced by the different peers. Hence, in order to adapt to varying channel conditions each peer needs to dynamically modify its cross-layer transmission strategy. In this section we use Application layer packet scheduling (at the source peer) in conjunction with dynamic MAC retransmission limit adaptation and PHY layer modulation mode selection (at all peers) to maximize the end-to-end video quality.

A. Application Layer Packet Scheduling and MAC Retransmission Strategy

Given the set of video packets for each sub-flow f_x as well as their transmission deadlines, the source peer determines which subset should be transmitted and in which order. Additionally, each intermediate peer determines the maximum permissible number of video-packet retransmissions under losses. In our discussion in [20] we have shown that if we schedule packets appropriately, then the optimal retransmission limit (in terms of maximizing the decoded video quality) for any packet may be directly computed, given the packet deadline, the time needed to transmit the packet, and the current time (which includes time taken by packets transmitted before the current packet). Based on the results presented in [16], [17] that determine the redundancy rate of forward error correction (FEC) codes for scalably encoded data, we can show that, if packets are sorted in order of their distortion impacts, then the optimal retransmission strategy involves transmitting a particular packet as many times as possible (without violating its delay constraint) until it is successfully received. This is because retransmission codes form a special class of FEC codes for which the redundancy rate is directly proportional to the probability of successfully receiving the data. In particular, we have shown in [20] that for a scalable codec, the packets need to be scheduled as follows:

- Packets are first ordered in increasing order of packet decoding deadlines.
- Packets with the same decoding deadline are ordered in terms of their impact on the decoded distortion, i.e. for a 3D wavelet coder in order of increasing spatio-temporal frequency, and within one subband starting with the most significant bitplane.

Once this scheduling is done, video packets are placed in MAC frames (packets) and these are passed to the MAC layer in the specified order. Hence, the optimal retransmission limit $M_{j_x}^{opt}$ for packet j_x (from sub-flow f_x being transmitted to node v_a) may be computed as:

$$M_{j_x}^{opt} = \left\lfloor \frac{Deadline_{j_x} - t_{cur}}{\frac{L_{j_x}}{T_x^a} + t_{overhead}} \right\rfloor, \qquad (12)$$

where $Deadline_{j_x}$ is the decoding deadline for the packet, T_x^a is the PHY layer transmission rate, and $\lfloor \circ \rfloor$ is the floor operation. The time t_{cur} is the current time, which includes the service interval duration, the time between transmission opportunities, and finally the transmission time for packets before the current packet. Lastly, the time $t_{overhead}$ represents the duration of the required overheads corresponding to polling and acknowledgment policies.

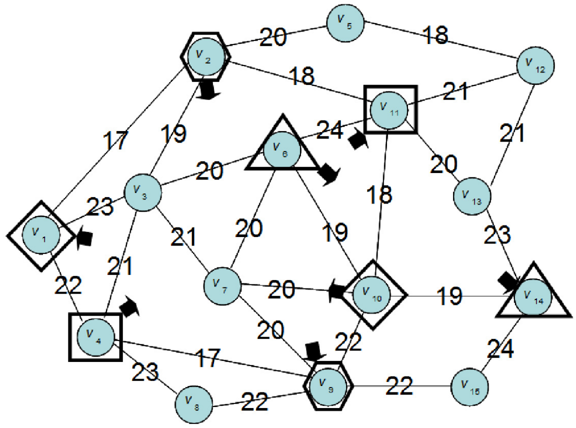

Fig. 1. Network Topology and Average Channel SINRs (in dB).

B. Adaptive PHY Layer Modulation Mode Selection

As mentioned earlier, the PHY layer modulation mode θ_x^a determines the error probability ϵ_x^a as well as the transmission rate T_x for sub-flow f_x being transmitted to node v_a. Hence, implicitly, the retransmission limit selected in equation (12) is a function of θ_x^a. The goal of the adaptive modulation strategy is to select the physical layer modulation mode θ_x^a that maximizes the number of packets of f_x that are successfully received at v_a. Importantly, we perform this optimization once for every transmission opportunity that sub-flow f_x receives, to limit the underlying computational complexity. Sub-flow f_x is allowed a transmission opportunity[6] $r_x^a t_{SI}^{(RX)}$ within each service interval t_{SI} of node v_a. At the start of each transmission opportunity, we determine θ_x^a that maximizes the number of packets that can be successfully transmitted to node v_a.

Given the underlying packet loss rate ϵ_x^a and the retransmission limit M_{j_x} for packet j_x, the expected number of transmissions[7] equals:

$$\begin{aligned}\overline{m}_{j_x} &= \sum_{i=1}^{M_{j_x}} i\,[1-\epsilon_x^a]\,[\epsilon_x^a]^{i-1} + M_{j_x}\,(\epsilon_x^a)^{M_{j_x}} \\ &= \frac{1-(\epsilon_x^a)^{M_{j_x}}}{1-\epsilon_x^a}, \end{aligned} \qquad (13)$$

where we drop (θ_x^a, L_{j_x}) for ease of notation, with L_{j_x} being the length of the packet. Hence, the expected time taken to transmit packet j_x (i.e. expected time before it is received at node v_a) is:

$$\overline{\tau}_{j_x} = \overline{m}_{j_x} \left(\frac{L_{j_x}}{T_x^a} + t_{overhead} \right). \qquad (14)$$

All packets are assumed to be transmitted their expected number of times in order to determine the time taken by any

[6] Note that we maintain separate queues at the MAC for each sub-flow at a node, and hence packets from one sub-flow do not experience any queuing delay due to packets from another sub-flow.

[7] We use the expected number of transmissions because we cannot determine the actual transmissions without actually transmitting the packets.

TABLE III
SUB FLOW CHARACTERISTICS

	Ψ_1 **Coastguard** 1490 kbps $v_4 \rightarrow v_{11}$	Ψ_2 **Mobile** 2000 kbps $v_6 \rightarrow v_{14}$	Ψ_3 **Foreman** 1500 kbps $v_{10} \rightarrow v_1$	Ψ_4 **Hall** 470 kbps $v_2 \rightarrow v_9$
Quality Layer 1	f_1, $\lambda_1 = 124.7$ $B_1 = 450$	f_5, $\lambda_5 = 157.5$ $B_5 = 640$	f_9, $\lambda_9 = 114.0$ $B_9 = 430$	f_{13}, $\lambda_{13} = 150.0$ $B_{13} = 230$
Quality Layer 2	f_2, $\lambda_2 = 14.2$ $B_2 = 340$	f_6, $\lambda_6 = 16.5$ $B_6 = 490$	f_{10}, $\lambda_{10} = 11.8$ $B_{10} = 320$	f_{14}, $\lambda_{14} = 4.3$ $B_{14} = 80$
Quality Layer 3	f_3, $\lambda_3 = 4.9$ $B_3 = 390$	f_7, $\lambda_7 = 4.9$ $B_7 = 560$	f_{11}, $\lambda_{11} = 4.1$ $B_{11} = 390$	f_{15}, $\lambda_{15} = 2.2$ $B_{15} = 80$
Quality Layer 4	f_4, $\lambda_4 = 1.8$ $B_4 = 310$	f_8, $\lambda_8 = 1.4$ $B_8 = 310$	f_{12}, $\lambda_{12} = 1.8$ $B_{12} = 360$	f_{16}, $\lambda_{16} = 1.3$ $B_{16} = 80$

packet transmission. Hence, the total expected time to transmit the first η_x packets for the flow f_x is:

$$\overline{\tau}_{\eta_x}^{trans} = \sum_{j_x=1}^{\eta_x} \overline{m}_{j_x} \left(\frac{L_{j_x}}{T_x^a} + t_{overhead} \right). \tag{15}$$

The maximum expected number of packets transmitted in the transmission opportunity $t_{SI}^{(RX)} r_x^a$ is:

$$\eta_x^{\max} = \arg\max_{\eta_x} \left\{ \overline{\tau}_{\eta_x}^{trans} | \overline{\tau}_{\eta_x}^{trans} \leq t_{SI}^{(RX)} r_x^a \right\}. \tag{16}$$

We thus need to select the modulation mode θ_x^a that maximizes $\eta_x^{\max}$. In a typical 802.11a network this involves searching through the available 8 PHY modulations to select the appropriate mode.

V. SIMULATION RESULTS

A. Network Topology and Multi-Hop Wireless Mesh Test-Bed

The network topology we used for our experiments, as proposed in [30], consists of 15 wireless peers connected to each other, and is shown in Fig. 1. We simulated the multi-hop mesh network under predetermined transmission intervals for each link. Our simulation took into account different parameters for the various layers, such as varying SINR, transmission overheads at the MAC layer due to packet acknowledgements and polling overheads, as well as queuing and propagation delays in the various links of the mesh network. To incorporate the effect of noise and interference [9], we performed a number of simulations, setting the mean SINR of each link, between 15 and 25 dB and allowing random perturbations around the mean to simulate the effects of a time-varying channel. A path with a bad link (i.e. low SINR) is often not viable compared to other paths available to an aggregate flow. In this way, the proposed framework can accommodate the case when a peer is out of range or has left the network. The mean SINR for each link is listed (in dB) in Fig. 1, and is inversely proportional to the distance between the peers (represented by the length of the link). Network feedback via the overlay network is conveyed to each hop whenever a significant change in the experienced channel condition occurs. Our wireless mesh network test-bed simulates the 802.11a network with a maximum PHY layer rate of 54 Mbps, and 8 modulation modes. It also includes a realistic time reservation scheme, dynamic retransmissions at the MAC layer, adaptive modulation at the PHY layer using realistic channel models, and optimal distortion based ordering of the aggregate flows' packets at the APP layer.

TABLE IV
OPTIMIZED TOTAL QUALITY METRIC U_{MTQ}

	Congestion Metric	Optimized U_{MTQ}	Number of Admitted Sub-Flows Ψ_1	Ψ_2	Ψ_3	Ψ_4	Resulting U_{MMQ}
Centralized Optimal	-	964.80	2	2	4	4	211.31
Collaborative Distributed	Bottleneck	963.44	3	2	2	4	204.02
	Mean	955.54	2	2	2	4	204.02
Non-collaborative Distributed	Bottleneck	660.21	4	0	4	4	0
	Mean	451.65	4	0	4	0	0

B. Flow Characteristics

We consider the transmission of four different aggregate flows, with different sequence characteristics and bit-rates, over this network infrastructure. The sequences selected are CIF (352×288) with 300 frames at 30 frames per second. We use a Group Of Pictures (GOP) structure with 16 frames in each GOP, and a temporal decomposition with four temporal levels. The end-to-end tolerable delay was set to one GOP, i.e. 0.54 seconds, and the MSDU frames were allowed a maximum size of 1000 bytes.

Each aggregate flow is split into four quality layer sub-flows by grouping packets from the same temporal level into one sub-flow. The characteristics of the sub-flows, including their rate requirements (in kbps), their quality parameters, and their source and destination peers, are listed in Table III. The quality parameter λ_x is determined for each sub-flow as the average of the priorities of the packets within the sub-flow. Each packet's priority is determined by the importance of its content in terms of distortion impact upon loss or gain of the packet. The distortion impact is predicted based on the spatial and temporal level the packet belongs to, dependencies between packets, and packet content (i.e. motion vector, texture, color channel information etc.). Within each aggregate flow, we normalize the individual packet priorities to a maximum of 1000. More details on this may be obtained from [21].

The source and destination peers are shown in Fig. 1, where communicating peers share the same bounding box, with outgoing and incoming arrows representing source and destination respectively. From Table III, we notice that whether we order sub-flows purely in terms of $\frac{\lambda_x}{B_x}$ or first in terms of quality layers and then in terms of $\frac{\lambda_x}{B_x}$, the resulting sorted lists are very similar (due to the near monotonic relationship between $\frac{\lambda_x}{B_x}$ and the quality layer). This indicates that results are likely to be similar whether we optimize U_{MTQ} or U_{MMQ}. We have observed similar results using different sets of video sequences encoded at several bit-rates (500-2000 Kb/s) and using 2 to 5 quality layers (temporal decomposition levels) per flow.

C. Results

In Table IV and Table V, we present results on collaborative resource sharing among peers determined based on the average channel SINRs shown in Fig. 1. In Table IV, we present results obtained by optimizing U_{MTQ}, and in Table V, we present results from optimizing U_{MMQ}. We show the overall resulting utility (both metrics), as well as the discarded sub-flows for both congestion metrics. The distributed collaborative algorithms have a performance similar to the centralized

TABLE V
OPTIMIZED MINIMUM QUALITY METRIC U_{MMQ}

	Congestion Metric	Optimized U_{MMQ}	Number of Admitted Sub-Flows				Resulting U_{MTQ}
			Ψ_1	Ψ_2	Ψ_3	Ψ_4	
Centralized Optimal	-	211.31	2	1	4	4	964.80
Collaborative Distributed	Bottleneck	204.02	2	2	2	4	955.54
	Mean	204.02	2	2	2	4	955.54
Non-collaborative Distributed	Bottleneck	0	4	0	4	4	660.21
	Mean	0	4	0	4	0	451.65

TABLE VI
AVERAGE PSNR (DB) RESULTS (3 RUNS WITH 300 VIDEO FRAMES PER RUN)

	Coastguard $v_4 \to v_{11}$	**Hall** $v_2 \to v_9$
Dynamic Cross-Layer	27.31	34.57
Fixed Retry and PHY mode	24.96	29.09

TABLE VII
NUMBER OF ADMITTED SUB-FLOWS FOR THE ORIGINAL FLOWS (USERS) WHEN TWO ADDITIONAL FLOWS (USERS) ARE ENTERING THE NETWORK

	Congestion Metric	Admitted Sub-flows (4 flows)				Admitted Sub-flows (6 flows)					
		Ψ_1	Ψ_2	Ψ_3	Ψ_4	Ψ_1	Ψ_2	Ψ_3	Ψ_4	Ψ_5	Ψ_6
Collaborative Distributed U_{MTQ}	Bottleneck	3	2	2	4	2	1	3	2	4	2
	Mean	2	2	2	4	2	1	3	3	4	1
Collaborative Distributed U_{MMQ}	Bottleneck	2	2	2	4	2	1	3	2	4	2
	Mean	2	2	2	4	2	1	3	3	4	1

exhaustive approach (within 3.5%) in terms of both utilities, which corresponds to at most admitting two less sub-flows. At the same time, they significantly outperform the non-collaborative approach (by > 30%) which often discards all sub-flows of an aggregate flow. We note that in all tests we have performed, with quality layers determined by temporal levels, the distributed collaborative algorithms have near optimal performance and always significantly outperform the non-collaborative approach. We omit a comprehensive set of results due to space constraints, however, the presented results are indicative.

Interestingly, we observe that the use of bottleneck congestion metrics leads to better performance for the collaborative algorithm. This is because our selected topology and source destination peers require that a large number of sub-flows pass through common bottleneck nodes (e.g. v_7, v_3). As expected, whether we optimize U_{MTQ} or U_{MMQ}, we achieve similar results, as may be seen by comparing Table IV against Table V. We reiterate that U_{MTQ} and U_{MMQ} serve as two different system optimization policies that resource managers might deploy for their system based on their application. In general, optimizing U_{MTQ} does not achieve optimal U_{MMQ} and U_{MMQ} admits sub-flows from different aggregate flows more equitably.

We repeat our experiments for a network with greater congestion, and observe that the distributed collaborative algorithms perform within 10% of the centralized algorithm. Also, collaboration during resource sharing always leads to a benefit (~50% in this case) in terms of the overall system utility.

We also quantify the importance of using dynamic cross-layer optimization on the average decoded PSNR for the Hall and Coastguard sequences. We compare our dynamic cross-layer optimization strategies (as described in Sections IV.A. and IV.B.) against a strategy that uses fixed MAC retransmission limits and a fixed modulation at the PHY layer, and the same optimal packet ordering at the APP layer as in the cross-layer case. The MAC retransmission limit is set to 2 for all packets, as it corresponds to the mean number of retransmissions experienced with a dynamic strategy, given the delay constraint of 1 GOP. The PHY layer mode is selected based on the mean SINR of each link. Table VI shows that with dynamic cross-layer optimization, we achieve significant gains (2-5dB) over using fixed retransmission limits and modulation strategies. These improvements are due to the optimal use of allocated bandwidth by the cross-layer optimized transmission strategies. All of the PSNRs in Table VI are for 300 frames, averaged over 3 sample runs that each use the flow specifications in Table III, the average network conditions in Figure 1, and the time reservations and path assignments obtained using the collaborative distributed algorithm for optimization of U_{MTQ} (with bottle-neck congestion metric).

Table VII shows how the admitted sub-flows for 4 aggregate flows change when two additional flows are introduced in the network and also the admitted sub-flows of the new aggregate flows. Our additional two flows are at 1.5Mb/s and 1.0Mb/s, with source-destination peers v_7-v_{15} and v_3-v_{12}, respectively. From the results, we note that the addition of two aggregate flows into the network significantly changes the admitted sub-flows of the four original aggregate flows. Our admission control and path-provisioning algorithms gracefully reduce the quality of the current P2P streams in order to allow more P2P streams to simultaneously coexist. In a non-collaborative scheme, where the original video streams do not adjust their quality, there may not have been enough available bandwidth to accommodate any sub-flows from the two additional P2P streams.

VI. CONCLUSION

In this paper, we propose multi-user collaborative algorithms for optimizing P2P multimedia transmission over wireless multi-hop enterprise networks. We consider a scenario with multiple pairs of peers transmitting scalably encoded video (partitioned into multiple sub-flows) to each other over a shared infrastructure. We design distributed algorithms for P2P resource exchange, including collaborative admission control, path provisioning, and air-time reservation at intermediate nodes. We compare the performance of these distributed resource exchange algorithms under different levels of collaboration against centralized optimal allocation strategies. We show that the collaborative distributed resource exchange algorithm performs within 10% (in terms of system utility) of the centralized approach with significantly reduced complexity. Furthermore, the collaborative strategy significantly outperforms (by ~50%) the noncollaborative strategy that is typically employed in current streaming systems.

We also design real-time cross-layer optimization strategies, including Application layer scheduling, MAC layer retransmission and PHY layer modulation mode selection at all source and intermediate peers such that the end-to-end multimedia quality is maximized under dynamically varying channel conditions. We show that for channel variations of 1-2 dB per link we can improve the end-to-end multimedia quality received by any sub-flow by approximately 2- 5 dB in PSNR.

There are several possible extensions of this research. In this paper we ignore variations in channel SINR due to interference from transmitting neighboring peers. A direct extension is to use an interference matrix, or a simplified elementary rate matrix approach to schedule peer transmissions such that the interference is minimized. This will also correspondingly require modifications to the resource sharing and cross-layer adaptation strategies. Furthermore, these distributed resource management schemes may also be combined with P2P routing algorithms such as flooding to dynamically discover the optimal path between source and destination peers given information about other flows in the network, and the underlying channel conditions. Finally, these algorithms may also be extended to support multicast routing by decomposing multicast flows into several "virtual" unicast flows and using these virtual flows to interface with the proposed algorithms.

ACKNOWLEDGMENT

We would like to acknowledge the support of the NSF CAREER Award CCF-0541867, NSF CCF-0541453 and NSF CNS-0509522 and grants from Intel IT Research and UC Micro. We are also grateful to the reviewers that helped improve the paper through their insightful comments.

REFERENCES

[1] K. Holt, "Wireless LAN: Past, present, and future," in *Proc. IEEE Design, Automation, and Testing*, 2005, vol. 3, pp. 92-93.

[2] W. Wei and A. Zakhor, "Multipath unicast and multicast video communication over wireless ad hoc networks," in *Proc. Int. Conf. Broadband Networks, Broadnets*, 2002, pp. 496-505.

[3] 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 Commun. Mag.*, vol. 12, no. 4, pp. 59-65, Aug. 2005.

[4] M. van der Schaar, S. Krishnamachari, S. Choi, and X. Xu, "Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 WLANs," *IEEE J. Select. Areas Commun.*, vol. 21, no. 10, pp. 1752-1763, Dec. 2003.

[5] A. Butala and L. Tong, "Cross-layer design for medium access control in CDMA ad-hoc networks," *EURASIP J. on Applied Signal Processing*, to appear.

[6] Y. Wu, P. A. Chou, Q. Zhang, K. Jain, W. Zhu, and S.-Y. Kung, "Network planning in wireless ad hoc networks: A cross-layer approach," *IEEE J. Select. Areas Commun.*, vol. 23, no. 1, pp. 136-150, Jan. 2005.

[7] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high throughput path metric for multi-hop wireless routing," in *Proc. ACM Conf. Mob. Computing and Networking*, 2003, pp. 134-146.

[8] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proc. ACM Conf. Mob. Computing and Networking*, 2004, pp. 114-128.

[9] R. L. Cruz, and A. V. Santhanam, "Optimal routing, link scheduling and power control in multihop wireless networks," in *Proc. IEEE INFOCOM*, 2003, vol. 1, pp. 702-711.

[10] 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.

[11] D. Krishnaswamy and J. Vicente, "Scalable adaptive wireless networks for multimedia in the proactive enterprise," *Intel Technology J.*. [Online] Available: `http://developer.intel.com/technology/itj/2004/volume08issue04/art04_scalingwireless/p01_abstract.htm`

[12] S. Roy, M. Covell, J. Ankcorn, S. Wee, and T. Yoshimura, "A system architecture for managing mobile streaming media services," in *Proc. IEEE Distrib. Computing Syst. Workshop*, 2003, pp. 408-413.

[13] D. Krishnaswamy, "Network-assisted link adaptation with power control and channel reassignment in wireless networks," in *Proc. 3G Wireless Conference*, 2002, pp. 165-170.

[14] 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*, vol. 19, no. 7, pp. 653-673, Aug. 2004.

[15] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks." Chapter in *Mobile Computing* (T. Imielinski and H. Korth, eds.) Kluwer Acad. Pub., 1996.

[16] J. Thie, "Optimal Erasure Protection Assignment for Scalably Compressed Data over Packet-Based Networks." Ph.D. thesis, Aug. 2003.

[17] J. Thie and D. Taubman, "Optimal erasure protection assignment for scalable compressed data with small packets and short channel codewords," *EURASIP Journal on Applied Signal Processing–Special issue on Multimedia over IP and Wireless Networks*, no. 2, pp. 207-219, Feb. 2004.

[18] J.-P. Ebert and A. Willig, "A Gilbert-Elliot bit error model and the efficient use in packet level simulation," Technical Report TKN-99-002, Telecommunication Networks Group, Technische Universität Berlin, Mar. 1999.

[19] M. Dai, D. Loguinov, and H. Radha, "Rate-Distortion modeling of scalable video coders," in *Proc. IEEE International Conference on Image Processing*, Oct. 2004.

[20] M. van der Schaar and D. S. Turaga, "Cross-layer packetization and retransmission strategies for delay-sensitive wireless multimedia transmission," *IEEE Trans. Multimedia*, under review.

[21] S. Shankar, Z. Hu, and M. van der Schaar, "Cross-layer optimized transmission of wavelet video over IEEE 802.11a/e WLANs," in *Proc. Packet Video*, Dec. 2004.

[22] B. Awerbuch and T. Leighton, "Improved approximation algorithms for the multicommodity flow problem and local competitive routing in dynamic networks," in *Proc. 26th ACM Symposium on Theory of Computing*, May 1994.

[23] D. S. Turaga and M. van der Schaar, "Cross-layer aware packetization strategies for optimized wireless multimedia transmission," in *Proc. IEEE ICIP*, Sep. 2005.

[24] M. Wang and M. van der Schaar, "Rate-Distortion modeling for wavelet video coders," in *Proc. IEEE ICASSP*, 2005.

[25] X. Zhu and B. Girod, "Distributed rate allocation for multi-stream video transmission over ad-hoc networks," in *IEEE ICIP*, Sep. 2005.

[26] R.-F. Liao, R. Wouhaybi, and A. Campbell, "Wireless incentive engineering," *IEEE J. Select. Areas Commun., Special Issue on Recent Advances in Multimedia Wireless*, 4th quarter, 2003.

[27] S. Shenker, "Efficient network allocations with selfish users," in *Proc. Perform.*, Sep. 1990, pp. 279-285.

[28] J. MacKie-Mason and H. Varian, "Pricing congestable network resources," *IEEE J. Select. Areas Commun.*, vol. 13, no.7, pp. 1141-1149, Sep. 1995.

[29] R. La and V. Anantharam, "Optimal routing control: Repeated game approach," *IEEE Trans. Automat. Contr.*, vol. 47, no. 3, pp. 437-450, Mar. 2002.

[30] M. Kodialam and T. Nandagopal, "Characterizing achievable rates in multi-hop mesh networks with orthogonal channels," *IEEE/ACM Trans. Networking*, vol. 13, no. 4, pp. 868-80, Aug. 2005.

[31] S. Toumpis and A. J. Goldsmith, "Capacity regions for wireless ad hoc networks," *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 736-748, July 2003.

[32] L. Xiao, M. Johansson, and S. P. Boyd, "Simultaneous routing and resource allocation via dual decomposition," *IEEE Trans. Commun.*, vol. 52, no. 7, pp. 1136-1144, July 2004.

[33] M. K. Marina and S. R. Das, "Ad hoc on-demand multipath distance vector routing (AOMDV)," in *Proc. ACM SIGMOBILE Mobile Computing and Communications*, July 2002, vol. 6, no. 3.

Nicholas Mastronarde is currently a Ph.D. student in the Electrical Engineering Department at UCLA. He finished his M.Sc. degree in 2006 in the Electrical and Computer Engineering at UC Davis. His research interests are in multimedia compression, signal processing and networking.

Deepak S. Turaga received the B.Tech. degree in Electrical Engineering from Indian Institute of Technology, Bombay in 1997 and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from Carnegie Mellon University, Pittsburgh in 1999 and 2001, respectively. He is currently a Research Staff Member in the Exploratory Stream Processing department at the IBM T. J. Watson Research Center in Hawthorne. He was at Philips Research during 2001-2003 and with Sony Electronics in 2003-2004. His research interests lie primarily in statistical signal processing, multimedia coding and streaming, machine learning and computer vision applications. In these areas he has published over 35 journal and conference papers and two book chapters. He received the CSVT 2006 Transactions Best Paper Award (with M. van der Schaar and B. Pesquet-Popescu), and is a coauthor for the 2006 IEEE ICASSP Best Student Paper (with H. Tseng, O. Verscheure and U. Chaudhari). He has filed over fifteen invention disclosures, and has participated actively in MPEG standardization activities. He is an Associate Editor of the IEEE Transactions on Multimedia.

Mihaela van der Schaar received the Ph.D. degree from Eindhoven University of Technology, Eindhoven, The Netherlands, in 2001. Prior to joining the UCLA Electrical Engineering Department faculty on July 1st, 2005, she was between 1996 and June 2003 a senior researcher at Philips Research in the Netherlands and USA, where she led a team of researchers working on multimedia coding, processing, networking, and streaming algorithms and architectures. From January to September 2003, she was also an Adjunct Assistant Professor at Columbia University. From July 1st, 2003 until July 1st, 2005, she was an Assistant Professor in the Electrical and Computer Engineering Department at University of California, Davis. She has published extensively on multimedia compression, processing, communications, networking and architectures and holds 27 granted US patents. Since 1999, she was an active participant to the ISO Motion Picture Expert Group (MPEG) standard to which she made more than 50 contributions and for which she received three ISO recognition awards. She was also chairing for three years the ad-hoc group on MPEG-21 Scalable Video Coding, and also co-chairing the MPEG ad-hoc group on Multimedia Test-bed. She was an Associate Editor of IEEE Transactions on Multimedia and SPIE Electronic Imaging Journal from 2002-2005. Currently, she is an Associate Editor of IEEE Transactions on Circuits and System for Video Technology and an Associate Editor of IEEE Signal Processing Letters. She received the NSF CAREER Award in 2004, IBM Faculty Award in 2005, Okawa Foundation Award in 2006, and the Best Paper Award for her paper published in 2005 in the IEEE Transactions on Circuits and Systems for Video Technology.