Video Dissemination over Hybrid Cellular and Ad Hoc Networks

Ngoc Minh Do, *Student Member*, *IEEE*, Cheng-Hsin Hsu, *Member*, *IEEE*, and Nalini Venkatasubramanian, *Senior Member*, *IEEE*

Abstract—We study the problem of disseminating videos to mobile users by using a hybrid cellular and ad hoc network. In particular, we formulate the problem of optimally choosing the mobile devices that will serve as gateways from the cellular to the ad hoc network, the ad hoc routes from the gateways to individual devices, and the layers to deliver on these ad hoc routes. We develop a Mixed Integer Linear Program (MILP)-based algorithm, called POPT, to solve this optimization problem. We then develop a Linear Program (LP)-based algorithm, called MTS, for lower time complexity. While the MTS algorithm achieves close-to-optimum video quality and is more efficient than POPT in terms of time complexity, the MTS algorithm does not run in real time for hybrid networks with large numbers of nodes. We, therefore, propose a greedy algorithm, called THS, which runs in real time even for large hybrid networks. We conduct extensive packet-level simulations to compare the performance of the three proposed algorithms. We found that the THS algorithm always terminates in real time, yet achieves a similar video quality to MTS. Therefore, we recommend the THS algorithm for video dissemination over hybrid cellular and ad hoc networks.

Index Terms—Wireless networks, video streaming, quality optimization, resource allocation

1 Introduction

Mobile devices, such as smartphones and tablets, are getting increasingly popular, and continue to generate record-high amount of mobile data traffic. For example, a Cisco report indicates that mobile data traffic will increase 39 times by 2015. Sixty six percent of the increase is due to video traffic [1]. Unfortunately, existing cellular networks were designed for unicast voice services, and do not natively support multicast and broadcast. Therefore, cellular networks are not suitable for large-scale video dissemination. This was validated by a measurement study, which shows that each HSDPA cell can only support up to six mobile video users at 256 kbps [2]. Thus, disseminating videos to many mobile users over cellular networks could lead to network congestion and degraded user experience. This network capacity issue may be partially addressed by deploying more cellular base stations, installing dedicated broadcast networks (such as Digital Video Broadcast-Handheld, DVB-H [3]), or upgrading the cellular base stations to support Multimedia Broadcast Multicast Service (MBMS) [4]. However, these approaches all result in additional costs for new network infrastructure, and might not be fully compatible with existing mobile devices. Hence, a better way to disseminate videos to many mobile users is critical to the profitability of cellular service providers.

- *N.M. Do and N. Venkatasubramanian are with the Donald Bren School of Information and Computer Sciences, University of California, Irvine, 6210 Donald Bren Hall, Irvine, CA 92697-3425. E-mail: nmdo@uci.edu, nalini@ics.uci.edu.*
- *C.-H. Hsu is with the Department of Computer Science, National Tsing Hua University, No. 101, Section 2, Kuang-Fu Road, Hsinchu, Taiwan 30013. E-mail: chsu@cs.nthu.edu.tw.*

Manuscript received 9 Feb. 2012; revised 5 Aug. 2012; accepted 3 Nov. 2012; published online 29 Nov. 2012.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2012-02-0065.
Digital Object Identifier no. 10.1109/TMC.2012.246.

In this paper, we study video dissemination in a hybrid cellular and ad hoc network. Fig. 1 depicts the underlying network, consisting of one or several base stations and multiple mobile devices equipped with heterogeneous network interfaces. Mobile devices not only connect to the base station over the cellular network, but also form an ad hoc network using short-range wireless protocols such as WiFi and Bluetooth. Mobile devices relay video traffic among each other using ad hoc links, leveraging such a *free* spectrum to alleviate bandwidth bottlenecks and cut down the expense of cellular service providers. Throughout the paper, we denote mobile devices that directly receive video data over the cellular network and relay the receiving data to other mobile devices over the ad hoc network as *gateways*. Notice that although we do not explicitly consider centralized access points in the short-range network, our formulation and solutions are general enough, and can be readily applied to WiFi and Bluetooth access points.

Disseminating videos over a hybrid cellular and ad hoc network is not an easy task because transmission of video data must adhere to timing constraints inherent in the delivery and playback of video content. Traditionally, video servers use computationally complex transcoders [5] to reduce the video coding rates to guarantee on time delivery of video data. However, in a hybrid network, real-time transcoding is not feasible on resource-constrained mobile devices. Thus, we employ scalable videos [6] for in-network video adaptation [7]. More precisely, at the base station, scalable coders encode each video into a scalable stream consisting of multiple layers, and each mobile device can selectively forward some layers to other mobile devices in a timely fashion.

To deliver the highest possible video quality, we study an optimization problem that determines: 1) the mobile devices that will serve as gateways and relay video data

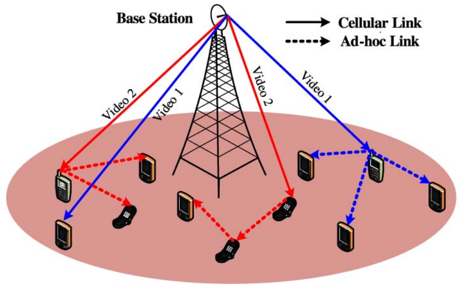

Fig. 1. A hybrid cellular and ad hoc network.

from the cellular network to the ad hoc network, 2) the multihop ad hoc routes for disseminating video data, and 3) the subsets of video data each mobile device relays to the next hops under capacity constraints. We formulate the optimization problem into a Mixed Integer Linear Program (MILP), and propose an MILP-based algorithm, called POPT, to optimally solve it. POPT has a rather high time complexity, and thus, we also propose two heuristic algorithms: MTS and THS. MTS is a Linear Program (LP)-based algorithm while THS is a greedy algorithm. We conduct extensive packet-level simulations to evaluate the performance of the proposed algorithms. We found that both MTS and THS achieve a similar video quality, which is close-to-optimum video quality with at most a 2 dB gap observed. More importantly, the THS algorithm has a much lower time complexity than POPT and MTS. It always terminates in real time, and supports large hybrid networks with 70+ mobile devices. Hence, we recommend the THS algorithm for video streaming over hybrid cellular and ad hoc networks. Last, we also build a real video dissemination system among multiple Android smartphones over a live cellular network. Via actual experiments, we demonstrate the practicality and efficiency of the proposed THS algorithm.

The rest of this paper is organized as follows: The related work is reviewed in Section 2. We give our system's overview, build up notations, define, and formulate our optimization problem in Section 3. This is followed by the proposed algorithms presented in Section 4. We evaluate the algorithms using extensive simulations and experiments in Sections 5 and 6, respectively. The paper is concluded in Section 7.

2 Related Work

Using an auxiliary ad hoc network to augment a cellular network has been considered in prior literature for enhancing data transfer and video dissemination.

2.1 Data Transfer over Hybrid Networks

We classify the existing approaches to speed up data transfer over a cellular network using an ad hoc network into two groups: unicast and multicast data transfers. In the former group [8], [9], data are unicast from a base station to mobile devices over a cellular network, and these devices relay the data to other mobile devices over an ad hoc network. In the latter group [10], [11], [12], [13], via multicast-enabled base stations, data are simultaneously sent to multiple mobile devices, which then propagate the data to other devices through multihop paths in an ad hoc network. Although, multicast data transfer may improve the system performance, current cellular networks only employ multicast for disseminating short messages (at most 90 characters) [14], which is inapplicable for disseminating videos.

Unicast Data Transfer. Luo et al. [8] design a hybrid network that uses a WiFi ad hoc network to route cellular data via other mobile devices with higher cellular data rates. Two neighbor discovery and routing protocols, proactive and on-demand, are proposed. With the former protocol, all devices proactively maintain the states of their immediate neighbors. When a device wants to discover a route to the base station, it issues a route discovery message to a neighbor with the highest cellular data rate. The message is further relayed by the neighbor to its highest rate neighbor until there is no neighbor with higher rate than the relayer or the hop count limit is reached. The final relayer is the one that receives data from the cellular network and propagates data to the original requester. With the on-demand protocol, devices do not maintain their neighbors' states. A requester discovers a route to the base station by flooding a route discovery message to all its neighbors within a given range. Devices with higher data rates than that of the previous hops forward the message to the base station, which eventually selects the best path to the requester. Simulation results show that the on-demand protocol typically incurs higher traffic overhead on the cellular network, while the proactive protocol consumes more energy. Through simulations, Hsieh and Sivakumar [9] show that generic ad hoc protocols do not work well in hybrid cellular and WiFi ad hoc networks, and may lead to: 1) degraded overall throughput, 2) unfair resource allocation, and 3) low resilience to mobility. They propose two approaches to improve the efficiency of ad hoc protocols. First, the base station can run optimization algorithms for the WiFi ad hoc network, for example, to build optimized routes. Second, mobile devices connected to other access networks can offload traffic from the cellular network to those access networks, so as to avoid network congestion around the base station.

Multicast Data Transfer. Law et al. [13] evaluate a hybrid network in which a cellular base station reduces its transmission range to achieve a higher data rate for mobile devices inside its range. Some mobile devices act as gateways and relay data to mobile devices outside the range via a multihop ad hoc network. The analysis and simulation results indicate that up to 70 percent downlink capacity improvement over pure cellular networks is possible. Lao and Cui [12] propose a hybrid network, in which each multicast group is either in the cellular mode or in the ad hoc mode. Initially, all multicast groups are in ad hoc mode, and when the bandwidth requirement of a group exceeds the ad hoc network capacity, the base station picks up that group and switches it into the cellular mode. Park and Kasera [10] consider the gateway node discovery problem, and model the ad hoc interference as a graph coloring problem. Solving this problem allows them to approximate the number of other mobile devices in the transmission range of a specific mobile device. Bhatia et al. [11] consider the ad hoc routing problem for multicast services, and also abstract ad hoc interference as a graph. They formulate a problem of finding the relay forest to maximize the overall data rate, and they propose an approximation algorithm.

Unlike the above works [8], [9], [10], [11], [12], [13], we focus on delay sensitive live video distribution over a hybrid network.

2.2 Video Dissemination over Hybrid Networks

The problem of video dissemination over hybrid networks has been recently considered [15], [16], [17], [18]. Qin and Zimmermann [15] present an adaptive strategy for live video distribution to determine the number of quality layers to be transmitted between two mobile devices. They also propose a technique that helps mobile nodes retrieve missing frames when nodes get reconnected after disconnections. Their solution, however, is only applicable to individual links, while live video distribution usually utilizes multihop paths. Hua et al. [16] formulate an optimization problem in a hybrid network to determine the cellular broadcast rate of each quality layer. In the ad hoc network, a flooding routing protocol is used to discover neighbors and a heuristic is employed to forward video data. Our work differs from Hua et al. [16] in several aspects: 1) we propose a unified optimization problem that jointly finds the optimal gateway mobile devices, ad hoc routes, and video adaptation, 2) we consider existing cellular base stations that may not natively support multicast, and 3) we employ Variable-Bit-Rate (VBR) streams.

Preliminary results of this paper were published in Do et al. [17]. The current paper makes several enhancements. First, we propose a new greedy algorithm, called THS, which has a much lower time complexity than the algorithms proposed in [17], yet still achieves close-to-optimum video quality. Second, we carry out packet-level simulations for more convincing evaluation results. Third, we evaluate our algorithms in a real hybrid network testbed [18].

3 Video Dissemination in Hybrid Networks

In this section, we first describe our system's overview and notations used frequently in the paper. We then state our problem that schedules to stream videos optimally, and formulate this problem as an MILP problem.

3.1 System Overview and Notations

We consider a hybrid network (see Fig. 1), which consists of a cellular base station and several mobile devices. Table 1 summarizes the notations used in the paper. The base station concurrently transmits K videos to U mobile devices, where each mobile device receives and renders a video chosen by its user. Throughout this paper, we use *node* to refer to both the base station and mobile devices. All mobile devices are equipped with two network interfaces for cellular and ad hoc networks, respectively. Examples of cellular networks include EDGE, 3G, and 4G cellular networks, and examples of ad hoc networks are WiFi ad hoc and Bluetooth networks. Mobile devices can always receive video data from the base station via cellular links. They also form an ad hoc network and exchange video data over it. Unlike cellular networks, ad hoc connectivity is not guaranteed because ad hoc networks, such as WiFi ad hoc and Bluetooth networks, have a rather short range, less than a few hundreds of meters, and are prone to disconnections due to user mobility.

TABLE 1
Notations Used Frequently in This Paper

U	Number of users.
K	Number of videos.
S	Number of segments.
L	Number of layers.
W	Number of segments in a scheduling window.
W'	Number of segments the scheduling window is shifted to the right from the previous scheduling time ($1 \leq W' \leq W$).
D	Duration of a segment (e.g., 2 seconds).
H	The maximum number of hops a unit can travel in each scheduling window.
$t_{k,s,l}$	Transmission unit of video k, segment s and layer l. It is the smallest data unit considered for transmission by our system.
$q_{k,s,l}$	Quality improvement of transmission unit $t_{k,s,l}$.
$z_{k,s,l}$	Size of transmission unit $t_{k,s,l}$.
$y^u_{k,s,l}$	= 1 if mobile device u already has $t_{k,s,l}$; = 0 otherwise.
$x^{a,v,u}_{k,s,l}$	A decision variable which is set to 1 if the scheduler decides to send $t_{k,s,l}$ from device v to device u, and set to 0 otherwise.
$m^{a,v}_{k,s,l}$	The number of descendants of mobile device v in the breadth-first tree rooted at node a for unit $t_{k,s,l}$.
$A_{k,s,l}$	A set of mobile devices already holding $t_{k,s,l}$.
$N^{a,h}_{k,s,l}$	A set of mobile devices that are h hops away from mobile device a, which is in $A_{k,s,l}$.
δ_u	Cellular link air time allocated for streaming units from base station to mobile device u. $\delta \geq \sum_{u=1}^{U} \delta_u$ where δ is the maximum resource allocated for the streaming application.
I_q	A maximum independent set of ad hoc links in which the links are able to be concurrently active.
λ_q	Ad hoc link air time allocated for streaming units within a maximum independent set I_q.
$c_{u,v}$	Channel data rate from mobile device u to v. We use $u = 0$ to denote the base station. Therefore, cellular data rate is denoted as $c_{0,v}$.
ω_u	Location of mobile device u.
$G(\mathbf{V},\mathbf{E})$	Network graph; $\mathbf{V}$ are nodes representing mobile devices, and $\mathbf{E}$ are edges representing links between mobile devices.
$G(\bar{\mathbf{V}},\bar{\mathbf{E}})$	Conflict graph; $\bar{\mathbf{V}}$ are nodes corresponding to the links in network graph, and $\bar{\mathbf{E}}$ are edges representing conflict between links in network graph.

Distributing videos in a hybrid network is challenging because: 1) wireless networks are dynamic in terms of connectivity, latency, and capacity, and 2) video data require high throughput and low latency. To cope with these challenges, we employ layered video coding [6], such as H.264/SVC [19], to encode each video into L layers. Layer 1 is referred to as the base layer, which provides a basic video quality. Layers $2, 3, \ldots, L$ are enhancement layers, which provide incremental quality improvements. An enhancement layer is decodable if all layers below it are received. With layered videos, we can dynamically adjust the number of layers sent to each mobile device. While the adjustments may be done very frequently, a subject user study [20] reveals that frequent quality changes lead to degraded viewer experience. Therefore, we divide each video into multiple D sec video segments, where D is a small number (e.g., 1 or 2 seconds). Quality changes are only allowed at boundaries of segments. We let S be the

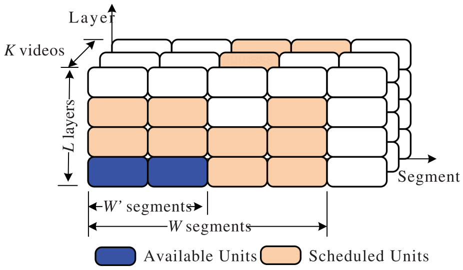

Fig. 2. Scheduling window of a specific link.

total number of segments of every video, and we let $t_{k,s,l}$ $(1 \leq k \leq K, 1 \leq s \leq S, 1 \leq l \leq L)$ be the *transmission unit* of video k, segment s, and layer l.

We study an optimization problem of selecting transmission units from W consecutive segments to transmit to mobile devices over the hybrid network. The W consecutive segments considered for selection form a scheduling window. We refer to a solution as a *schedule* and we call an algorithm that runs at the base station to compute schedules as a scheduler. The scheduler on the base station takes feedback from networks, and computes a new schedule every DW' secs $(1 \leq W' \leq W)$. That is, every DW' seconds, the scheduling window is shifted to the right for W' segments. The feedback from mobile device u includes transmission unit availability $y^u_{k,s,l}$ at u and u's location $\omega_u = (\omega_{u,x}, \omega_{u,y})$. We let $y^u_{k,s,l} = 1$ if u holds unit $t_{k,s,l}$, and $y^u_{k,s,l} = 0$ otherwise. We use $\omega_{u,x}$ and $\omega_{u,y}$ to denote the longitude and latitude of u, which can be derived from Global-Positioning-System (GPS) functionality, cellular network triangulations, and WiFi fingerprints, which work both outdoors and indoors [21]. Each mobile device u reports its $y^u_{k,s,l}$ and ω_u to the base station, and the base station maintains the state of availability and device location for all mobile devices $1 \leq u \leq U$.

Given that the base station maintains a global view of the hybrid cellular and ad hoc network, the scheduler on the base station computes the schedule for all cellular and ad hoc links. The base station sends a new schedule to mobile devices every DW' secs. The mobile devices then distribute transmission units following the schedule. To maintain the tractability, our schedule does not explicitly specify the transmission time of each transmission unit. Rather, the order of transmission units is determined by the importance of transmission units. We denote the list of transmission units sorted by their importance as a *precedence list*. Mobile devices skip transmission units that have not been received, and check their availability again whenever a transmission unit is completely sent.

Fig. 2 illustrates a scheduling window. Each rectangle represents a transmission unit. The dark units are the units already available at the receiving mobile device. The shaded units are the units scheduled by the schedulers. As the figure illustrates, the scheduler considers for scheduling only transmission units that are: 1) not available at the receiver and 2) in the scheduling window W.

Fig. 3 presents a different view on a schedule, in which circles represent mobile devices, rectangles are transmission units, and arrows indicate multicast trees for dissemination. This figure illustrates that, depending on unit availability, a schedule may specify different multicast trees for different transmission units. For example, mobile device 4 receives unit $t_{1,1,3}$ through the path consisting of base station 0 and devices 6, 7, while it receives unit $t_{1,1,1}$ through the path including only device 5. Figs. 2 and 3 show the generality of our scheduling problem.

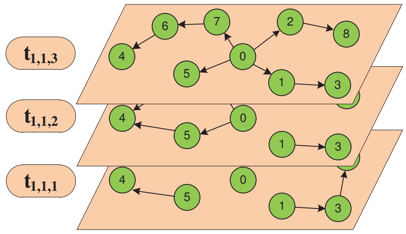

Fig. 3. Multicast trees built by the scheduler.

3.2 Problem Statement

With the above notations, we next formally describe the considered problem.

Problem 1 (Scheduling in a Hybrid Network). *Assume K videos are concurrently distributed from a cellular base station to a large number of mobile devices over a hybrid cellular and ad hoc network. Each video k is coded into multiple transmission units, while each unit $t_{k,s,l}$ represents layer l of segment s. Every DW' secs, we compute the schedule for a scheduling window of W segments and for every network link, to maximize the overall video quality across all mobile devices. The resulting schedule should be feasible in the sense that the scheduled units can be delivered in DW' secs.*

This scheduling problem is fairly general because: 1) any mobile device can relay any transmission unit to other mobile devices and 2) each transmission unit can be disseminated over different multicast trees.

Lemma 1 (Hardness). *The scheduling problem in a hybrid cellular and ad hoc network is NP-hard.*

Proof. The problem is to decide which transmission units to be transmitted over individual network links, where network links have diverse capacities and transmission units carrying variable-size video data with diverse quality improvement. Let us describe the NP-Complete 0/1 Multidimensional Knapsack (MDK01) problem [22]: Given a set of items $\mathbf{N} = 1, 2, \ldots, N$, where item n $(1 \leq n \leq N)$ has an M-dimensional weight $(w_{n,1}, w_{n,2}, \ldots w_{n,M})$ and a profit value p_n, find a subset $\mathbf{N}^* \subset \mathbf{N}$ to maximize the total profit and fit into an M-dimensional Knapsack with capacities $(c_1, c_2, \ldots, c_M)$.

For any MDK01 problem, we can construct a scheduling problem as follows: We first create a Hamilton path visiting all M mobile devices. We let cellular link capacity of the first mobile device to be c_1, while all other mobile devices have zero cellular link capacity. We set the capacity of the ad hoc link into the mth mobile device to be c_m $(2 \leq m \leq M)$. Next, for each item n, we create M transmission units; one for each mobile device. That is, we have MN transmission units in total. Last, we use profit p_n to assign the average quality improvement delivered by each transmission unit, and we use the weight $w_{n,m}$ to determine the size of transmission unit corresponding to item n and mobile device m. It is clear that an optimal solution of the constructed scheduling problem can be converted into an optimal solution for MDK01 in polynomial time. Thus, if there is an efficient algorithm for our scheduling problem, it can be used to solve MDK01 problem and all other NP-hard problems. This completes our NP-hard proof. □

Since the scheduling problem in a hybrid network is NP-hard, we formulate it as an MILP problem in the next section.

3.3 Optimization Problem Formulation

We first build up the video and network models. Then, we formulate the considered scheduling problem.

3.3.1 Rate-Distortion (R-D) Model

Our objective is to maximize the video quality under network bandwidth constraints. A popular method to achieve such quality-optimized system is to use a rate-distortion model, which describes the mapping between video rates and degrees of quality degradation in reconstructed videos. R-D models capture the diverse video characteristics and enable media-aware resource allocation. Due to flexible and complicated prediction structures of layered video streams, existing scalable R-D models [23] are fairly complex and may not be suitable for real-time applications. Hence, we adopt a low-complexity discrete R-D model below. The distortion caused by not sending a transmission unit $t_{k,s,l}$ to a mobile device can be divided into two parts [24], [25]: 1) truncation distortion and 2) drifting distortion. Truncation distortion refers to the quality degradation of pictures in segment s itself, and drifting distortion refers to the quality degradation of pictures in other segments due to imperfect reconstruction of reference pictures. We assume each segment s contains multiple groups-of-picture (GoPs) and, thus, can be independently decoded. This practical assumption eliminates the needs to model drifting distortion.

We let $q_{k,s,l}$ be the quality improvement when receiving $t_{k,s,l}$ in addition to the previously received $t_{k,s,l'}$, where $l' = 1, 2, \ldots, l-1$. While quality improvement can be in any video quality metric, we use peak signal-to-noise ratio (PSNR) throughout this paper for concrete discussions. PSNR is a widely used objective video quality metric, which is in dB scale and inversely related to mean-squared error [6]. We let $z_{k,s,l}$ be the size of $t_{k,s,l}$. The sets $\mathbf{Q}_k = \{q_{k,s,l} \mid 1 \leq s \leq S, 1 \leq l \leq L\}$ and $\mathbf{Z}_k = \{z_{k,s,l} \mid 1 \leq s \leq S, 1 \leq l \leq L\}$ model the R-D characteristics of video stream k. $\mathbf{Q}_k$ and $\mathbf{Z}_k$ are computed during the encoding time, and sent to the base station as metadata along with the video stream k itself. The base station uses them as inputs to solve the scheduling problem. Notice that $q_{k,s,l}$ and $z_{k,s,l}$ are merely two numbers and, thus, sending them along with the video data, typically in the order of kilo-bytes, leads to negligible overhead.

3.3.2 Network Capacity Model

We let $c_{u,v}$ be the link capacity between nodes u and v under the assumption of minimized interference. In practical systems, $c_{u,v}$ may be derived by various approaches. For example, Riiser et al. [26] propose to look up link capacity based on each node's location. In our simulations and experiments, we adopt this location-based approach, and use the device location ω_u to estimate the link capacity in both cellular and ad hoc networks. More specifically, we empirically measure the mapping between the node location and link capacity several times, and use the resulting values for capacity estimation. In the following paragraphs, we explain how the interference is minimized in: 1) cellular networks and 2) ad hoc networks.

Cellular networks control the interference via various multiple access methods (such as FDMA, TDMA, and CDMA) and via proper network planning (to avoid intercell interference). At a high level, the base station runs a centralized algorithm to allocate δ_u air-time to mobile device u, where $1 \leq u \leq U$, and $\delta = \sum_{u=1}^{U} \delta_u$ is the total air-time reserved for mobile data, which is a system parameter. Let node 0 be the base station, the effective cellular capacity between it and node u is $\delta_u c_{0,u}$, where δ_u ($1 \leq u \leq U$) is a variable of our optimization problem.

Interference in ad hoc networks is harder to control as the air-time allocation is done by distributed media access control (MAC) protocols. We model the air-time allocation using the conflict graphs [11], [27], [28]. A conflict graph is used to learn links that cannot be simultaneously activated due to interference. This happens in ad hoc networks because mobile devices use the same frequency for transmission. Two links interfere each other if at least one end of a link is in the transmission range of one or two ends of the other link. Let $G(\mathbf{V}, \mathbf{E})$ be the network graph, where $\mathbf{V}$ and $\mathbf{E}$ are nodes and edges. Its corresponding conflict graph $G(\bar{\mathbf{V}}, \bar{\mathbf{E}})$ is constructed as follows: We first create a vertex $\bar{\nu}_{u,v}$ in $\bar{\mathbf{V}}$ for each edge $\epsilon_{u,v} \in \mathbf{E}$, and we add an edge connecting $\bar{\nu}_{u,v}$ and $\bar{\nu}_{k,l}$ to $\bar{\mathbf{E}}$ if node u or v is in node k or l's transmission range.

Each independent set selected from a conflict graph $G(\bar{\mathbf{V}}, \bar{\mathbf{E}})$ corresponds to a set of edges in the network graph $G(\mathbf{V}, \mathbf{E})$ that can be simultaneously activated without interfering with each other. An independent set refers to a subset of vertices where no two of them are adjacent. An independent set is called a maximal independent set if adding any vertex to it leads to a nonindependent set. We let $\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_Q$ be all maximal independent sets. Given Q maximal independent sets, distributed MAC protocols allocate λ_q air-time to maximal independent set $\mathbf{I}_q$, where $\sum_{q=1}^{Q} \lambda_q \leq 1$ [28], [29]. λ_q ($1 \leq q \leq Q$) are variables of our optimization problem. For a link $\epsilon_{u,v} \in \mathbf{E}$ between two mobile devices u and v, the effective ad hoc capacity is therefore, $\sum_{1 \leq q \leq Q,\ \bar{\nu}_{u,v} \in \mathbf{I}_q} \lambda_q c_{u,v}$.

3.3.3 Controlling Dissemination Latency

Our optimization problem only determines which transmission units to send in the current scheduling window, but does not model the fine-grained delivery time of each transmission unit. We should mention that the unit delivery time could be modeled using time-indexed Integer Linear Program (ILP) [30]. In time-indexed ILP formulations, all time intervals are expressed as (rounded to) multiples of a sufficiently small time slot. In these formulations, short time slots are essential for good performance, but short time slots also lead to a large number of decision variables and render the formulation computationally intractable.

We do not employ time-indexed ILP in our formulation, but use two other approaches to control latency. First, we limit each unit to be sent over at most H hops in each scheduling window, where H is a small integer and a system parameter. Second, we employ the paths on the breadth-first trees for unit delivery, which is detailed in the following: Let $\mathbf{A}_{k,s,l}$ be the set of nodes that already have unit $t_{k,s,l}$. Nodes in $\mathbf{A}_{k,s,l}$ are potential sources for distributing $t_{k,s,l}$ and all other mobile devices are receivers of that transmission unit. For a source $a \in \mathbf{A}_{k,s,l}$ and an arbitrary receiver u, there are many paths between them for distributing $t_{k,s,l}$. To avoid inefficient paths, we only consider the paths that follow the breadth-first tree from the source a to all mobile devices not in $\mathbf{A}_{k,s,l}$. We let $\mathbf{N}^{a,h}_{k,s,l}$ be the receiving mobile devices that are h hops away from $a \in \mathbf{A}_{k,s,l}$ in the breadth-first tree, where $1 \le h \le H$. Mathematically, we write

$$\mathbf{N}^{a,h}_{k,s,l} = \bigcup_{u\in \mathbf{N}^{a,h-1}_{k,s,l}} \mathbf{N}^{u,1}_{k,s,l} \setminus \mathbf{N}^{a,h-1}_{k,s,l} \setminus \mathbf{N}^{a,h-2}_{k,s,l}, \qquad (1)$$

where $\mathbf{N}^{u,1}_{k,s,l}$ includes u's neighbors (note that $\mathbf{N}^{u,0}_{k,s,l} = \{u\}$). Equation (1) defines nodes in level h by finding all nodes that are neighbors of nodes in level $h-1$ ($\bigcup_{u\in \mathbf{N}^{a,h-1}_{k,s,l}} \mathbf{N}^{u,1}_{k,s,l}$) and then excluding nodes in levels $h-1$ ($\setminus N^{a,h-1}_{k,s,l}$) and $h-2$ ($\setminus N^{a,h-2}_{k,s,l}$) from the found nodes.

Fig. 4 presents an example of a breadth-first tree, which is formed to deliver unit $t_{k,s,l}$ from device 1 in $\mathbf{A}_{k,s,l}$ to four receivers. Device 1 is the root of the tree, and nodes 2 and 3 are in level 1. Using (1), our technique determines nodes in level 2 (i.e., $h = 2$) to be receivers 4 and 5 because

$$\mathbf{N}^{a,h-1}_{k,s,l} = \{2,3\},\ \bigcup_{u\in \mathbf{N}^{a,h-1}_{k,s,l}} \mathbf{N}^{u,1}_{k,s,l} = \{1,2,3,4,5\}, \mathbf{N}^{a,h-l}_{k,s,l} = \{2,3\},$$

and $\mathbf{N}^{a,h-2}_{k,s,l} = \{1\}$. Using breadth-first trees for path selection helps eliminate long and inefficient paths. For example, path 1-2-4-5 should not be chosen to deliver to data from 1 to 5 because there is a shorter path 1-3-5.

Distributing transmission units over breadth-first trees not only limits the distribution latency and avoids loops, but also reduces the complexity of the considered problem without sacrificing the solutions' quality. This is because paths that do not follow breadth-first trees are inefficient and should be avoided.

3.3.4 Formulation

We define $x^{a,v,u}_{k,s,l} \in \{0,1\}$ to be a decision variable: $x^{a,v,u}_{k,s,l} = 1$ if transmission unit $t_{k,s,l}$ is scheduled to be sent from node v to node u over the breadth-first tree rooted at node a; $x^{a,v,u}_{k,s,l} = 0$ otherwise. The scheduling problem in a hybrid cellular and ad hoc network is formulated in (2a)-(2h). In this formulation, we refer to the base station as node 0. The objective function in (2a) is the average video quality achieved by all U mobile devices. The objective function contains two terms (within the square brackets): the first term considers the breadth-first tree rooted at the base station, and the second term considers the breadth-first trees rooted at gateways, which directly receive the transmission unit from the base

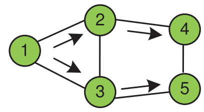

Fig. 4. An illustrative breadth-first tree for unit delivery.

station. We note that if the objective function only has the first term, a new transmission unit would only have a breadth-first tree rooted at the base station with height one because all mobile devices are one hop away from the base station. Consequently, the transmission units would never be exchanged over ad hoc networks. Hence, the second term is critical to the utilization of the ad hoc network, and overall video quality.

$$\max \frac{1}{U}\sum_{k=1}^{K}\sum_{s=s_c}^{s_c+W}\sum_{l=1}^{L} q_{s,k,l}\Bigg[\sum_{a\in \mathbf{A}_{k,s,l}\setminus\{0\}}\sum_{h=1}^{H}\sum_{v\in\mathbf{N}^{a,h-1}_{k,s,l}}\sum_{u\in\mathbf{N}^{a,h}_{k,s,l}} x^{a,v,u}_{k,s,l} + \sum_{a'\in\mathbf{N}^{0,1}_{k,s,l}}\sum_{h=1}^{H-1}\sum_{v\in\mathbf{N}^{a',h-1}_{k,s,l}}\sum_{u\in\mathbf{N}^{a',h}_{k,s,l}} x^{a',v,u}_{k,s,l}\Bigg] \qquad (2a)$$

$$\text{s.t.} \quad \sum_{k=1}^{K}\sum_{s=s_c}^{s_c+W}\sum_{l=1}^{L}\frac{z_{k,s,l}x^{0,0,\hat u}_{k,s,l}}{c_{0,\hat u}DW'} - \delta_{\hat u} = 0; \qquad (2b)$$

$$\sum_{u=1}^{U}\delta_u - \delta \le 0; \qquad (2c)$$

$$\sum_{k=1}^{K}\sum_{s=s_c}^{s_c+W}\sum_{l=1}^{L}\sum_{a\in\mathbf{A}_{k,s,l}}\frac{z_{k,s,l}x^{a,\hat u,\hat v}_{k,s,l}}{c_{\hat u,\hat v}DW'} - \sum_{1\le q\le Q,\ \bar\nu_{\hat u,\hat v}\in \mathbf{I}_q}\lambda_q = 0; \qquad (2d)$$

$$\sum_{q=1}^{Q}\lambda_q - 1 \le 0; \qquad (2e)$$

$$\sum_{a'\in\mathbf{A}_{\hat k,\hat s,\hat l+1}}\sum_{h=1}^{H}\sum_{t\in\mathbf{N}^{a',h-1}_{\hat k,\hat s,\hat l+1}} x^{a',t,\hat u}_{\hat k,\hat s,\hat l+1} - y^{\hat u}_{\hat k,\hat s,\hat l} - \sum_{a\in\mathbf{A}_{\hat k,\hat s,\hat l}}\sum_{h=1}^{H}\sum_{t\in\mathbf{N}^{a,h-1}_{\hat k,\hat s,\hat l}} x^{a,t,\hat u}_{\hat k,\hat s,\hat l} \le 0; \qquad (2f)$$

$$\sum_{a\in\mathbf{A}_{\hat k,\hat s,\hat l}}\sum_{h=1}^{H}\sum_{t\in\mathbf{N}^{a,h-1}_{\hat k,\hat s,\hat l}} x^{a,t,\hat u}_{\hat k,\hat s,\hat l} - 1 \le 0; \qquad (2g)$$

$$\sum_{a'\in\mathbf{A}_{\hat k,\hat s,\hat l}}\sum_{t'\in\mathbf{N}^{a',\hat h}_{\hat k,\hat s,\hat l}} x^{a',t',\hat u}_{\hat k,\hat s,\hat l} - \sum_{a\in\mathbf{A}_{\hat k,\hat s,\hat l}}\sum_{t\in\mathbf{N}^{a,\hat h-1}_{\hat k,\hat s,\hat l}} x^{a,t,\hat u}_{\hat k,\hat s,\hat l} \le 0; \quad \forall\, 1\le \hat u,\hat v\le U,\ 1\le\hat k\le K,\ 1\le\hat s\le S,\ 1\le\hat l\le L,\ 1\le\hat h\le H. \qquad (2h)$$

Constraints in (2b) and (2c) model the air-time allocation in the cellular network to guarantee that the capacity of the cellular network is not exceeded. Constraints in (2d) and (2e) model the air-time allocation in the ad hoc network to ensure that the capacity of the ad hoc network is not exceeded. The constraint in (2f) guarantees the dependency among layers. That is, a unit with layer l will not be sent to a device unless the device has units of all lower layers ($l' < l$) or will receive them in the current scheduling window. The constraint in (2g) ensures a mobile device receives each transmission unit from a single sender over a single breadth-first tree. It helps avoid transmission redundancy. The constraint in (2h) makes sure that a mobile device sends a transmission unit only if it receives that unit in current or earlier scheduling windows.

4 Scheduling Algorithms

In this section, we present three algorithms to solve the scheduling problem in a hybrid cellular and ad hoc network.

4.1 An MILP-Based Algorithm: POPT

The formulation in (2a)-(2h) consists of linear objective functions and constraints with integer decision variables ($x^{a,v,u}_{k,s,l}$) and real-value variables (δ_u and λ_q). Hence, it is an MILP problem and may be solved by MILP solvers. However, observe that constraints in (2d) and (2e) include all the maximal independent sets $\mathbf{I}_q$ ($1 \leq q \leq Q$) in the conflict graph, and finding all $\mathbf{I}_q$ itself is an NP-Complete problem [22]. Therefore, it is computationally impractical to consider all Q maximal independent sets. Jain et al. [31] propose a random search algorithm for deriving a subset of maximal independent sets that is sufficient for optimal schedulers. Li et al. [29] show that this random search algorithm is inefficient, and propose a priority-based algorithm to find the maximal independent sets that will be used in the optimal schedule with high probability. While the priority-based algorithm is defined for the throughput optimization problem in a multiradio, multichannel wireless network, it can be extended to other conflict graph-based optimization problems by revising the definition of the scheduling priority.

The priority-based algorithm works as follows: First, the shortest path between the source-destination pair of each flow is calculated. Then, the number of shortest paths traversing through each link is used as its priority. Next, the algorithm uses an anchor link to iterate through the links from high to low priority. For each anchor link, the algorithm scans through all links that are not its neighbors in the conflict graph, and creates a set of new maximal independent sets, where every maximal independent set contains the anchor link. A link covered by any maximal independent set will not be considered as an anchor link. The algorithm stops once all links are covered by at least a maximal independent set. Readers are referred to Li et al. [29] for more details on this algorithm.

We define a new priority function for each ad hoc link to achieve the following four design goals:

1. The links into mobile devices with more descendants in breadth-first trees are given higher priorities.
2. The links into mobile devices on breadth-first trees of transmission units with higher quality improvement values are given higher priorities.
3. The links with higher ad hoc link capacities are given higher priorities.
4. The links from mobile devices with higher cellular link capacities are given higher priorities.

Specifically, we define the priority function $f(u,v)$ of an edge from u to v as

$$f(u,v) = f_a(v) + f_c(u), \tag{3}$$

where $f_a(v)$ and $f_c(u)$ are the "importance" factors due to the ad hoc network and the cellular network, respectively. They are computed as

$$f_a(v) = c_{u,v} \sum_{k=1}^{K} \sum_{s=s_c}^{s_c+W} \sum_{l=1}^{L} \sum_{a \in \mathbf{A}_{k,s,l} \setminus \{0\}} m^{a,v}_{k,s,l} q_{k,s,l}, \tag{4}$$

$$f_c(u) = \delta c_{0,u} \sum_{k=1}^{K} \sum_{s=s_c}^{s_c+W} \sum_{l=1}^{L} m^{u,u}_{k,s,l} q_{k,s,l}, \tag{5}$$

where $m^{a,v}_{k,s,l}$ is the number of descendants of mobile device v on the breadth-first tree rooted at node a for video k, segment s, and layer l.

With the priority function $f(u,v)$, we leverage the priority-based algorithm [29] to generate a small set of $\hat{Q}$ ($1 \leq \hat{Q} \leq Q$) maximal independent sets that will be employed by the optimal schedules with high probability. We then apply a practical simplification on the formulation in (2a)-(2h) by only considering the $\hat{Q}$ maximal independent sets in the constraints in (2d) and (2e). Unlike the original formulation that may consist of exponentially many maximal independent sets, the simplified formulation has a reasonable number of maximal independent sets, and can be solved by MILP solvers. We use an MILP solver to solve the simplified formulation, and we refer to it as the Prioritized Optimization (POPT) algorithm.

4.2 A Throughput-Based Heuristic Algorithm: MTS

Since MILP problems are NP-Complete, the POPT algorithm does not scale well with the number of mobile devices. Hence, we develop a heuristic algorithm, called Maximum Throughput Scheduling (MTS) algorithm that was first presented in Do et al. [17].

This algorithm consists of two steps. In step 1, we derive the *demand capacity* $\hat{c}_{u,v}$ for each link from mobile device u to v. We iterate through the transmission units following the precedence list, which generally starts from lower to higher layers and from earlier to later segments. For each transmission unit, we first schedule it to be delivered to all mobile devices that have not received that unit yet, over the ad hoc links. Mobile devices that cannot receive the transmission unit from peer mobile devices are scheduled to receive the unit from the base station over the cellular network if their cellular data rate is enough to do so. More specifically, among mobile devices that do not have the unit, the base station selects a device with the highest number of children in an ad hoc tree rooted at that device, and sends it the unit. The selected device then propagates the unit along the tree. The iteration stops once there is a device that cannot get the unit via neither networks.

Trees are formed in the ad hoc network with interference consideration. We employ a simple and practical passive interference control. For each ad hoc link from u to v, we define its utilization factor as

$$\tau_{u,v} = \frac{\sum_{t_{k,s,l} \in \mathbf{X}_{u,v}} z_{k,s,l}}{c_{u,v} \cdot D \cdot W'},$$

where $\mathbf{X}_{u,v}$ is a set of transmission units scheduled for dissemination from u to v, $\sum_{t_{k,s,l}\in\mathbf{X}_{u,v}} z_{k,s,l}$ is the total size of transmission units scheduled from u to v during DW' secs, and $c_{u,v}$ is data rate of link (u,v). Whenever the scheduler wishes to transmit a unit $t_{k,s,l}$ from sender u to receiver v, it must ensure that the following constraint holds:

$$\sum \tau_{p,q} + \tau_{u,v} + \tau_{v,u} \leq 1 \tag{6}$$

for any nodes p or q in the transmission range of u or v. This constraint prevents the schedule from over utilizing the ad hoc network, which leads to network congestion. For the cellular network, we divide the amount of traffic over the cellular network to mobile device u by the cellular capacity for the air-time δ_u, and stop scheduling units over the cellular network once $\sum_{u=1}^{U} \delta_u \geq \delta$.

During the iteration, we accumulate the required ad hoc capacity $\tilde{c}_{u,v}$ for each ad hoc link by the ratio between the total transmission unit size sent over link $u-v$ during DW' secs and DW'. Upon getting demand capacity $\hat{c}_{u,v}$, we compute the maximum ad hoc network capacity by solving a Linear Program:

$$\max \sum_{1\leq q\leq \hat{Q}} \sum_{\bar{\nu}_{u,v}\in\mathbf{I}_q} c_{u,v}\lambda q \tag{7a}$$

$$\text{s.t.} \sum_{q=1}^{\hat{Q}} \lambda_q \leq 1; \tag{7b}$$

$$c_{u,v} \sum_{1\leq q\leq \hat{Q}, \bar{\nu}_{u,v}\in\mathbf{I}_q} \lambda_q \geq \hat{c}_{u,v}. \tag{7c}$$

This formulation maximizes the total ad hoc capacity in (7a), while guaranteeing the demand capacity is met in constraint in (7c). Formulation in (7) can be efficiently solved by LP solvers. Let λ_q^* $(1 \leq q \leq \hat{Q})$ be the optimum air-time allocation, we compute the optimum effective ad hoc link capacity between mobile device u and v as

$$c_{u,v}^* = c_{u,v} \sum_{1\leq q\leq \hat{Q}, \bar{\nu}_{u,v}\in\mathbf{I}_q} \lambda_q^*. \tag{8}$$

We next traverse through the precedence list, and we go through the transmission units of different videos on the descending order of the ratio of quality improvement and transmission unit size. We consider the transmission units with higher ratios earlier to achieve higher quality improvement under the same link capacity. Next, for each transmission unit, we sort the mobile devices that already hold that transmission unit on the numbers of descendants on their breadth-first trees. We iterate through these mobile devices, and schedule the transmission unit as long as the remaining link capacity permits. We stop once the current transmission unit is distributed to all mobile devices. If the transmission unit cannot be received by some mobile devices, we instruct the base station to transmit the unit to these devices over the cellular network. The algorithm stops upon both maximum ad hoc link capacity and cellular data air-time allocation δ are saturated.

The next lemma shows that the MTS algorithm runs in polynomial time.

Lemma 2 (Complexity of MTS Algorithm). *The MTS algorithm runs in polynomial time in the worst-case, if the formulation in (7) is solved by a polynomial time LP solver. For example, with Karmarkar's interior point method [32], the MTS algorithm has a time complexity of* $O[\hat{Q}^{5.5}E^2 + WLKUE^2]$, *where* $E = |\mathbf{E}|$ *is the number of edges in the network graph.*

4.3 A Tree-Based Heuristic Algorithm: THS

Both POPT and MTS algorithms employ optimization problem solvers. Although commercial and open-source solvers are available, these solvers might lead to long running time in the worst-case scenarios. Hence, we next propose a greedy scheduling algorithm that does not rely on any solvers. We call it Tree-Based Heuristic Scheduling (THS) algorithm, and it works as follows: We first sort all the transmission units in the W-segment scheduling window in descending order of importance, by layer, segment, and video. We then go through these WL units, and sequentially schedule the transmissions to all mobile devices. For each transmission unit, we first consider dissemination over the ad hoc network. If the ad hoc network cannot deliver this unit to all mobile devices in time, we fall back to the cellular network. The scheduler sends the unit to a device with highest cellular data rate among those which have not received the unit. The unit is further relayed by the device through the ad hoc network. THS handles ad hoc interference similarly to MTS, i.e., using (6). The algorithm stops when (6) does not hold and the cellular air-time is saturated.

We give the pseudocode of THS in Fig. 5. Lines 3-30 iterate each unit in the sorted list of units in the scheduling window, and make sure that all devices will receive the unit if they have not received it yet. Lines 5-16 schedule transmissions over the ad hoc network. More specifically, if there are mobile nodes owning unit $t_{k,s,l}$ in their buffer, the scheduler will request them to transmit the unit to other mobile devices over breadth-first ad hoc trees rooted at themselves. Function *findRoots(*$t_{k,s,l}$*)* returns a list of devices already having unit $t_{k,s,l}$. Function *getRootWithLargestTree(rootSet,* $t_{k,s,l}$*)* returns a root in *rootSet* whose number of descendants is the largest. Function *isFeasible(*$s, r, t_{k,s,l}$*)* checks if sending unit $t_{k,s,l}$ from sender s to receiver r exceeds ad hoc network capacity using the interference control method based on the inequation in (6). Function *getDevicesWithoutUnit(*$t_{k,s,l}$*)* returns a list of devices that have not received or been scheduled to receive unit $t_{k,s,l}$ yet. Lines 17-29 schedule transmissions over the cellular network. The cellular network is employed only when the ad hoc network cannot deliver units. The mobile device with the highest cellular data rate will be chosen as a gateway if it has not received the transmission unit. This gateway then forms a breadth-first ad hoc tree rooted at itself to forward the transmission units to all descendants.

The next lemma reports the complexity of THS. Compared to the MTS algorithm, THS has a smaller time complexity.

Lemma 3 (Complexity of THS Algorithm). *Assuming* H *is a small constant, the THS algorithm given in Fig. 5 has a time complexity of* $O[WLKU^2E]$, *where* $E = |\mathbf{E}|$ *is the number of edges in the network graph.*

```
schedule = ∅ % entry format: {sender, receiver, unit}
time = 0 % time to transmit units from server to gateways
for t_{k,s,l} in the list in order
  rootSet = findRoots(t_{k,s,l}) % set of devices having unit t_{k,s,l}
  while rootSet ≠ ∅
    root = getRootWithLargestTree(rootSet, t_{k,s,l})
    for h = 1 to H
      r = descendants(root, h) % get dest at level h
      for each r_i in r
        s = parent(r_i, root) % get r_i's parent
        if isFeasible(s, r_i, t_{k,s,l}), schedule += {s, r_i, t_{k,s,l}}
      end for
    end for
    remove root from rootSet
    if getDevicesWithoutUnit(t_{k,s,l}) == ∅, break
  end while
  while (B = getDevicesWithoutUnit(t_{k,s,l})) ≠ ∅
    g = getGW(B) % gw in B with max 3G rate
    if time + z_{k,s,l}/rate(server, g) <= W' · D
      schedule += {server, g, t_{k,s,l}} % add gateway
      time = time + z_{k,s,l}/rate(server, g)
      for h = 1 to H % form the tree rooted at g
        r = descendants(g, h) % get dest at level h
        for each r_i in r
          s = parent(r_i, g) % get r_i's parent
          if isFeasible(s, r_i, t_{k,s,l}), schedule += {s, r_i, t_{k,s,l}}
        end for
      end for
  end while
end for
```

Fig. 5. Tree-based heuristic scheduling algorithm.

5 Simulation-Based Evaluation

We conduct extensive packet-level simulations to evaluate our proposed algorithms in this section.

5.1 Settings

We employ a well-known network simulator, Qualnet 5.0 [33]. We emphasize that Qualnet captures many more details in a hybrid cellular and ad hoc network, and provides simulator results closer to real life, compared to the flow-based simulations used in our earlier work [17]. We implement all the proposed scheduling algorithms in the simulator. Some algorithms use an optimization solver, CPLEX [34], to solve the MILP and LP problems. Specifically, whenever an optimization problem needs to be solved, we pause Qualnet temporarily and call CPLEX. Once CPLEX completes, we resume Qualnet. That is, we ignore the impact of the algorithms' running time on the video streaming performance. The goal here is to use POPT as a benchmark to evaluate the performance of MTS and THS. In practical systems, such as the testbed implementation presented in Section 6, we do not pause the dissemination system.

For comparisons, we also implement a cellular only optimal scheduler, which is referred to as Current* in the figures. We emphasize that Current* is not a naive algorithm. Rather, it achieves optimal streaming quality without leveraging on the auxiliary ad hoc network.

TABLE 2
Videos Used for Evaluations

Video	Layer	Mean Bit Rate (bps)	Avg Quality (dB)
Crew	1	28377.8	27.66
	2	74026.7	31.48
	3	260084.9	37.44
	4	925342.7	44.12
Star War	1	14059.6	32.3
	2	25605.3	34.7
	3	43556.4	39.04
	4	155763.4	42.27

In our simulations, we employ WiMAX to establish connections between base station and mobile devices, and use IEEE 802.11b to form an ad hoc network. WiMAX supports a peak bit rate of 1.2 Mbps, while the maximum data rate of 802.11b is 6 Mbps. Mobile devices are randomly placed in a network with a terrain of $1{,}000 \times 1{,}000\ \mathrm{m}^2$. We vary the number of mobile devices $U = 10, 20, 30, 40, 50, 60, 70$ watching live videos. WiMAX is configured so that the base station can reach all mobile devices and vice versa while 802.11 range is set to be 200 m. We employ the Two-Ray propagation model [35] in the simulations. UDP is used as the underlying transport protocol. We use the Random Waypoint model [36] to simulate mobile device mobility. By default, maximum speed is 2 m/s with 60 second pause time, unless it is otherwise specified. The cellular network reserves air-time fraction $\delta = 0.75$ for data traffic. Maximum hop count H is set to be 3 by default. All the experiments are run on a Linux workstation with a dual-core AMD 2.2 GHz CPU and 8 GB RAM.

We adopt the video traces of H.264/SVC layered videos from an online video library [37]. The mean bit rate and average video quality for each layer of the considered videos are given in Table 2. In this paper, we report sample simulation results of distributing *Crew*. However, the proposed formulation and solutions are general and also work for the scenarios where mobile devices watch different videos. The video is divided into $D = 2$ sec segments. Each simulation lasts for 90 secs, so the number of segments streamed is 45. We let $W = 6$ and $W' = W/2$ if not otherwise specified. We assume an initial buffering time of 3 secs when determining whether a transmission unit misses its playback deadline.

We consider the following performance metrics:

- *PSNR:* received video quality in dB.
- *Running time*: the time an algorithm takes to solve a scheduling problem.
- *Transmitted data amount*: the total traffic amount in each simulation.
- *Decodable segment ratio*: the ratio of the number of received segments with at least the base-layer over the total number of segments in the video.
- *Delivery ratio*: the ratio of the number of units successfully delivered to mobile devices before deadline over the number of units scheduled for streaming.

5.2 Simulation Results

Performance Improvement. We investigate the performance improvement achieved by the hybrid network compared to

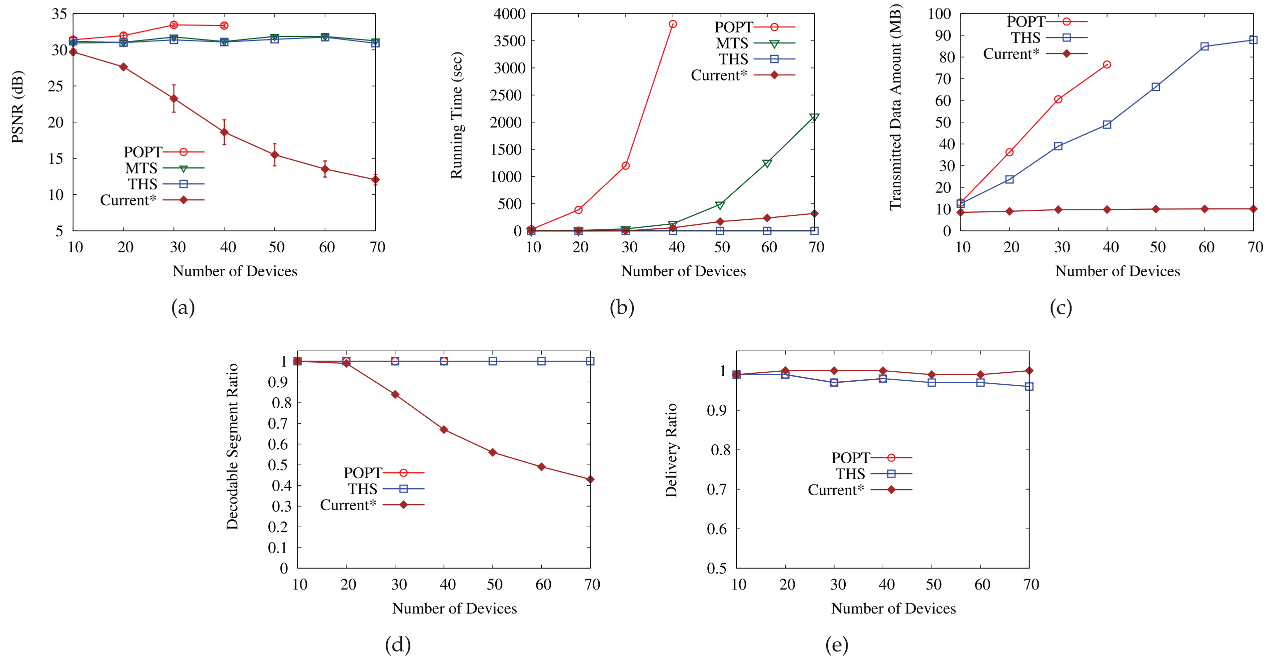

Fig. 6. Performance improvement: (a) video quality with 95 percent confidence interval, (b) running time, (c) transmitted data amount, (d) decodable segment ratio, and (e) delivery ratio.

the cellular-only network with varied number of mobile devices U. Fig. 6a shows with 95 percent confidence intervals that, for a PSNR requirement of 30 dB, the Current* scheduler can only support 10 mobile devices. POPT, MTS, and THS algorithms all achieve that quality with any investigated number of mobile devices. Note that these schedulers provide an advantage over Current*—mobile devices receive almost equally—good PSNR. That is, the longest range of 95 percent confidential interval achieved by POPT, MTS, and THS is merely 0.30 dB, while Current* suffers from a much larger range of up to 3.78 dB.

As shown in Fig. 6a, POPT achieves the highest PSNR, but we have to stop at $U = 40$ because it takes prohibitively long time when solving a problem with more mobile devices. We plot the average running time of different algorithms in Fig. 6b, which shows that it took more than 1 hour on average for POPT to come up with a schedule for 40 device networks. We observe that two algorithms, MTS and THS, achieve similar PSNR, at most 2 dB lower than POPT. Fig. 6b indicates that MTS is more efficient than POPT, but MTS' running time still increases prohibitively with the increase of device density. With a 70-device network, MTS takes more than half an hour to generate a schedule. In contrast, the THS algorithm always terminates in very short time under any number of devices. This shows that the THS algorithm achieves a good tradeoff between complexity and solution quality. Because MTS and THS achieve similar PSNR, but THS runs faster than MTS, we do not consider MTS in the remaining comparisons.

To have a deeper look at how much the ad hoc network contributes to the success of the proposed systems, we show the transmitted data amount due to video streaming in Fig. 6c. The amount of data transmitted through the cellular network by Current* is less than 10 MB regardless to the increasing number of devices. This is because the cellular network's resources are used up. With a 70-device network, the achieved PSNR of THS is 20+ dB higher than that of Current* because the data amount THS transmits in the hybrid network is more than 90 MB. This clearly shows the benefit of forming a hybrid network.

Next, we plot the decodable segment ratio in Fig. 6d. This figure shows that the fraction of decodable segments with Current* decreases from 99 (20 devices) to 83 percent (30 devices), and eventually to 42 percent (70 devices). This reveals that without ad hoc networks, the cellular only network cannot even deliver the base layer units to all mobile devices, once the network is larger than 20 devices.

Fig. 6e plots the delivery ratio with different scheduling algorithms. During streaming, units scheduled for transmission are lost due to packet loss and missed deadline. It is observed that the schedulers employing hybrid networks lose more transmission units. There are two reasons: 1) the ad hoc network is more vulnerable for packet loss due to interference and mobility, 2) with the schedulers employing the hybrid network, the average number of hops per unit is higher, which contributes a higher probability of the unit loss. Nonetheless, because ad hoc networks provide additional network capacity, after accounting for these lost units, POPT and THS algorithms still outperform Current*, both in terms of average video quality and quality variation among mobile devices, as shown in Fig. 6a.

Summarizing, the THS algorithm achieves almost-optimal video quality and terminates in real time. Therefore, we no longer present the results of Current* and POPT in the rest of this section.

Mobility and Window Size. We first investigate the performance of the system with varied device speed when W and W′ are set to 6 and 3, respectively. The number of devices investigated is varied among: 30, 50, and 70. Fig. 7 shows that with $W = 6$ and $W' = 3$, the PSNR achieved by

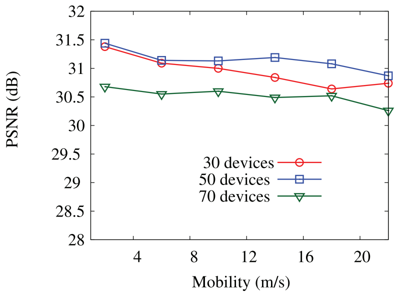

Fig. 7. THS performs well even under high mobility.

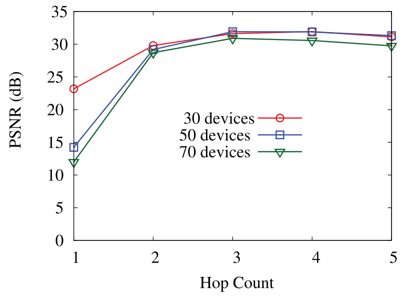

Fig. 8. THS is sensitive to the maximum hop count setting.

the THS algorithm does not drop significantly when the nodes move faster. Its PSNR drops only 0.5 dB when the nodes' speed increases from 2 to 22 m/s. We attribute this merit to a short time period between two consecutive scheduling times (6 secs). This short time period allows the THS algorithm to adapt to the dynamic network conditions, and the ad hoc links are likely to remain connected.

Next, we evaluate the impact of W and W′ on the performance of 50-device networks. Table 3 presents PSNR of different W and W′ values. With $W = 1$ and $W' = 1$, each segment is considered for scheduling one time, but the scheduling duration is very short, just 2 secs. Thus, its PSNR does not drop much as mobility speed increases. However, its PSNR is lower than that of $W = 12$ and $W' = 6$ at low speeds. For example, at 2 m/s, PSNR is 30.67 dB with $W = 1$ and $W' = 1$, while it is 31.96 dB with $W = 12$ and $W' = 6$. With $W = 12$ and $W' = 6$, the streaming system suffers from a sharper PSNR drop at high mobility due to the longer scheduling duration. This table shows the importance of selecting W and W' values. Based on the experimental results, we recommend the users to select $W > W'$, which helps the system adapt to dynamic network conditions better because every unit may be considered several times. In low-mobility scenarios, we recommend larger W and W' for lower update overhead. In high-mobility scenarios, we recommend smaller W and W' so that the scheduler is more aware of the current network conditions.

Maximum Hop Count. We study the implication of maximum hop count H. We plot the video quality in Fig. 8, which shows that the best H value is 4 for $U = 30$ and $U = 50$, and is 3 for $U = 70$. With a small H, the scheduling algorithms only consider short dissemination paths in the ad hoc network, and may underutilize the network resources. With a large H, the scheduling algorithms consider to use long dissemination paths, which increases not only the utilization of the ad hoc network, but also the probability of packet loss due to interference and user mobility. This figure sheds some lights on good H values in a hybrid network. We recommend H values between 2 and 4.

TABLE 3
PSNR (in dB) under Different Mobility and Window Size

Speed	$W = 1; W' = 1$	$W = 6; W' = 3$	$W = 12; W' = 6$
2 m/s	30.67	31.44	31.96
10 m/s	30.44	31.13	30.56
18 m/s	30.17	31.04	29.84

6 Testbed Experiments

We next report the experimental results gathered from a real testbed. We implement a Linux streaming server on a 2-GHz Pentium IV machine with 1-GB RAM. The server connects to the Internet via a Fast Ethernet link. We also implement a streaming client on Android, and deploy it on five Android smartphones. This testbed is located at the University of California Irvine campus. The smartphones connect to the streaming server over T-Mobile 3G network, and are equipped with GPS readers. In our testbed, we adopt a location-based interference control method inspired by Evensen et al. [38]. More precisely, a history database containing data rates at locations on our campus is built up at the streaming server. We employ the same videos used in Section 5 (see Table 2). The server follows the traces to generate UDP packets. We stream a video in each experiment, but our testbed is general and can handle multiple videos concurrently. Each video is split into 2 sec segments, each segment is encoded into four layers. We use $W = 10$ and $W' = W/2$. The initial buffering time is 5 sec. We consider the following metrics:

- *PSNR:* video quality in dB. We consider two PSNR metrics in this section. *Expected* PSNR indicates the video quality scheduled to be received at mobile devices by the scheduler. *Received* PSNR indicates the video quality achieved at mobile devices after transmissions. If not explicitly specified, by PSNR, we refer to received PSNR.
- *Decodable segment ratio:* see Section 5.1.
- *Delivery ratio:* see Section 5.1.
- *Delay:* the average time duration for a unit to be delivered from the streaming server to mobile device.

We first report the performance of the proposed THS algorithm. Fig. 9a shows the average PSNR. In this figure, Current*, and Current*_S refer to the received and expected PSNR values over the cellular-only network, respectively. Moreover, THS and THS_S are the received and expected PSNR values of the THS algorithm, respectively. With both videos, the hybrid network outperforms the cellular network. The gap between THS and Current* when streaming *Crew* (8.6 dB) is higher than that of *Star War* (4.5 dB) because *Crew* has a higher bit rate. This shows that the ad hoc network significantly improves the video streaming quality, compared to cellular-only networks. Fig. 9b partially explains why the cellular network suffers from such a low PSNR when streaming *Crew*: there are only 93 percent of decodable segments. Fig. 9c depicts the delivery ratios in

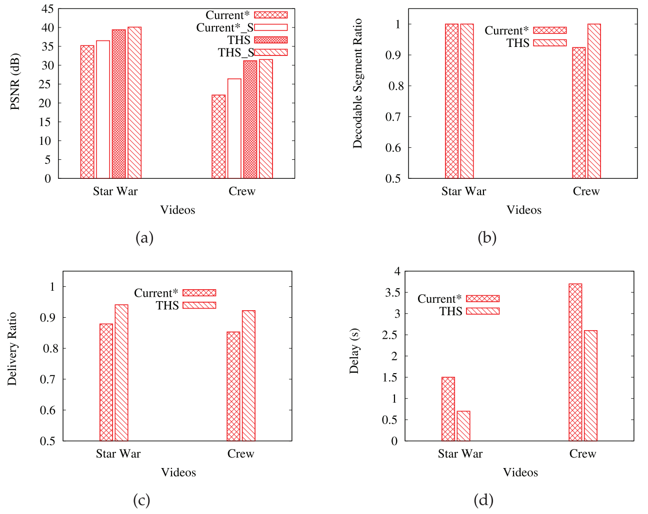

Fig. 9. Experimental results from the testbed: (a) PSNR, (b) decodable segment ratio, (c) delivery ratio, and (d) delay.

both networks when streaming videos. The number of received units in the hybrid network is higher because WiFi links have much higher bandwidth than cellular links. This observation also explains lower delays in the hybrid network shown in Fig. 9d.

We next compare THS against POPT algorithm. In our experiments, POPT runs on average 1.22 secs to generate a schedule. In contrast, THS runs in less than 100 msecs, i.e., in real time. Table 4 shows the PSNR, decodable segment ratio, and delivery ratio when streaming *Crew* over the hybrid network. Results from *Star War* show similar trends. This table reveals that although POPT tries to optimize the received PSNR at all receivers, it results in lower PSNR. This is because POPT has a much longer running time and, thus, cannot keep up with real-time transmissions.

The experimental results from the actual testbed confirm the observations we made in Qualnet simulations: the THS algorithm clearly outperforms the Current* algorithm. Furthermore, the THS algorithm may outperform POPT in real systems, which can be attributed to the long running time of the POPT algorithm. This demonstrates that the THS algorithm is practical and efficient.

7 Conclusion

We studied the problem of optimally leveraging an auxiliary ad hoc network to boost the overall video quality of mobile users in a cellular network. We formulated this problem as an MILP problem to jointly solve the gateway selection, ad hoc routing, and video adaptation problems for a global optimum schedule. We proposed three algorithms: 1) an MILP-based algorithm, POPT, 2) an LP-based algorithm, MTS, and 3) a greedy algorithm, THS. Via packet-level simulations, we found that neither POPT nor MTS scale to large hybrid networks. This is because they both employ numerical methods to solve optimization problems. Therefore, we recommend the THS algorithm, which terminates in real time even when there are 70+ mobile devices in the hybrid network.

TABLE 4
Experimental Results from the Testbed: POPT versus THS Algorithms

Metric	POPT	THS
PSNR (dB)	16.8	30.9
Decodable segment ratio (%)	60.6	100
Delivery ratio (%)	54.3	91.7

The simulation results indicate that the THS algorithm not only runs fast, but also achieves overall video quality close to the optimum: at most 2 dB difference is observed, compared to the POPT algorithm. In contrast, optimum schedules over the cellular network achieves much lower video quality compared to POPT: more than 15 dB difference is observed. We also validated the practicality and efficiency of the THS algorithm using a real testbed in a live cellular network. The experimental results confirm that the THS algorithm result in high video quality. Moreover, the THS algorithm could outperform the POPT algorithm in real systems. This is because although POPT could generate optimal schedules, its high running time may lead to many late segments, which in turn render inferior video quality.

We believe that massive delivery of rich information is useful in a range of mission critical scenarios such as military command-and-control and emergency response applications where existing infrastructure may be damaged, inaccessible, or overloaded. For example, customized notifications in emergency alerting situations will make it possible for users to receive rich alerts (such as evacuation maps and traffic routes) based on their current context for a more effective response. The ability to combine multiple infrastructure and ad hoc connections to achieve faster and improved information exchange on a societal scale, as demonstrated in this paper, is key to enabling richer mobile applications for the future.

References

[1] "Cisco Visual Networking Index: Forecast and Methodology," http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html, 2012.

[2] F. Hartung, U. Horn, J. Huschke, M. Kampmann, T. Lohmar, and M. Lundevall, "Delivery of Broadcast Services in 3G Networks," *IEEE Trans. Broadcasting,* vol. 53, no. 1, pp. 188-199, Mar. 2007.

[3] M. Kornfeld and G. May, "DVB-H and IP Datacast - Broadcast to Handheld Devices," *IEEE Trans. Broadcasting,* vol. 53, no. 1, pp. 161-170, Mar. 2007.

[4] S. Parkvall, E. Englund, M. Lundevall, and J. Torsner, "Evolving 3G Mobile Systems: Broadband and Broadcast Services in WCDMA," *IEEE Comm. Magazine,* vol. 44, no. 2, pp. 30-36, Feb. 2006.

[5] J. Xin, C. Lin, and M. Sun, "Digital Video Transcoding," *Proc. IEEE,* vol. 93, no. 1, pp. 84-97, Jan. 2005.

[6] Y. Wang, J. Ostermann, and Y. Zhang, *Video Processing and Communications,* first ed. Prentice Hall, 2001.

[7] I. Kofler, M. Prangl, R. Kuschnig, and H. Hellwagner, "An H.264/SVC-Based Adaptation Proxy on a WiFi Router," *Proc. ACM 18th Int'l Workshop Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '08),* pp. 63-68, May 2008.

[8] H. Luo, R. Ramjee, P. Sinha, L. Li, and S. Lu, "UCAN: A Unified Cellular and Ad-Hoc Network Architecture," *Proc. ACM MobiCom 2003,* pp. 353-367, Sept. 2003.

[9] H. Hsieh and R. Sivakumar, "On Using Peer-to-Peer Communication in Cellular Wireless Data Networks," *IEEE Trans. Mobile Computing,* vol. 3, no. 1, pp. 57-72, Mar. 2004.

[10] J. Park and S. Kasera, "Enhancing Cellular Multicast Performance Using Ad Hoc Networks," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '05),* pp. 2175-2181, Mar. 2005.

[11] R. Bhatia, L. Li, H. Luo, and R. Ramjee, "ICAM: Integrated Cellular and Ad Hoc Multicast," *IEEE Trans. Mobile Computing,* vol. 5, no. 8, pp. 1004-1015, Aug. 2006.

[12] L. Lao and J. Cui, "Reducing Multicast Traffic Load for Cellular Networks Using Ad Hoc Networks," *IEEE Trans. Vehicular Technology,* vol. 55, no. 3, pp. 317-329, May 2006.

[13] L. Law, K. Pelechrinis, S. Krishnamurthy, and M. Faloutsos, "Downlink Capacity of Hybrid Cellular Ad Hoc Networks," *IEEE Trans. Networking,* vol. 18, no. 1, pp. 243-256, Feb. 2010.

[14] E. Stanley and J. Sutton, *Public Response to Alerts and Warnings on Mobile Devices.* Nat'l Academies, Apr. 2011.

[15] M. Qin and R. Zimmermann, "An Adaptive Strategy for Mobile Ad Hoc Media Streaming," *IEEE Trans. Multimedia,* vol. 12, no. 4, pp. 317-329, June 2010.

[16] S. Hua, Y. Guo, Y. Liu, H. Liu, and S. Panwar, "Scalable Video Multicast in Hybrid 3G/Ad-Hoc Networks," *IEEE Trans. Multimedia,* vol. 13, no. 2, pp. 402-413, Apr. 2011.

[17] N. Do, C. Hsu, S. Jatinder, and N. Venkatasubramanian, "Massive Live Video Distribution over Hybrid Cellular and Ad Hoc Networks," *Proc. IEEE Int'l Symp. World of Wireless, Mobile and Multimedia Networks (WoWMoM '11),* pp. 1-9, June 2011.

[18] N. Do, C. Hsu, X. Huang, and N. Venkatasubramanian, "An Experimental Study on Scalable Video Streaming over Hybrid Cellular and Ad Hoc Networks," *Proc. ACM/IFIP/USENIX Int'l Conf. Middleware (Middleware '11),* Dec. 2011.

[19] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 17, no. 9, pp. 1103-1120, Sept. 2007.

[20] P. Ni, A. Eichhorn, C. Griwodz, and P. Halvorsen, "Fine-Grained Scalable Streaming from Coarse-Grained Videos," *Proc. Int'l Workshop Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '09),* pp. 103-108, Sept. 2009.

[21] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Rev.,* vol. 37, no. 6, pp. 1067-1080, Nov. 2007.

[22] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness,* first ed. W.H. Freeman, 1979.

[23] C. Hsu and M. Hefeeda, "On the Accuracy and Complexity of Rate-Distortion Models for FGS-Encoded Video Sequences," *ACM Trans. Multimedia Computing, Communications, and Applications,* vol. 4, no. 2, pp. 15:1-15:22, May 2008.

[24] C. Hsu, N. Freris, J. Singh, and X. Zhu, "Rate Control and Stream Adaptation for Scalable Video Streaming over Multiple Access Networks," *Proc. Int'l Packet Video Workshop (PV '10),* pp. 1-8, Dec. 2010.

[25] N. Freris, C. Hsu, X. Zhu, and J. Singh, "Resource Allocation for Multihomed Scalable Video Streaming to Multiple Clients," *Proc. IEEE Int'l Symp. Multimedia (ISM '10),* pp. 9-16, Dec. 2010.

[26] H. Riiser, T. Endestad, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Video Streaming Using a Location-Based Bandwidth-Lookup Service for Bitrate Planning," *ACM Trans. Multimedia Computing, Comm. and Applications,* vol. 8, no. 3, pp. 24:1-24:19, July 2012.

[27] P. Gupta and P. Kumar, "The Capacity of Wireless Networks," *IEEE Trans. Information Theory,* vol. 46, no. 2, pp. 388-404, Mar. 2000.

[28] Y. Wang, W. Wang, X. Li, and W. Song, "Interference-Aware Joint Routing and TDMA Link Scheduling for Static Wireless Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 19, no. 12, pp. 1709-1726, Dec. 2008.

[29] H. Li, Y. Cheng, C. Zhou, and P. Wan, "Multi-Dimensional Conflict Graph Based Computing for Optimal Capacity in MR-MC Wireless Networks," *Proc. IEEE 30th Int'l Conf. Distributed Computing Systems (ICDCS '10),* pp. 774-783, Aug. 2010.

[30] M. Pinedo, *Scheduling: Theory, Algorithms, and Systems,* third ed. Springer, 2008.

[31] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of Interference on Multi-Hop Wireless Network Performance," *Proc. ACM MobiCom,* pp. 66-80, Sept. 2003.

[32] N. Karmarkar, "A New Polynomial-Time Algorithm for Linear Programming," *Combinatorica,* vol. 4, no. 4, pp. 373-395, Dec. 1984.

[33] "Qualnet," http://www.scalable-networks.com/, 2013.

[34] "IBM ILOG CPLEX," http://www.ibm.com/software/integration/optimization/cplex-optimizer, 2013.

[35] T. Rappaport, *Wireless Communications: Principles & Practice,* first ed. Prentice Hall, Jan. 1996.

[36] C. Bettstetter, G. Resta, and P. Santi, "The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks," *IEEE Trans. Mobile Computing,* vol. 2, no. 3, pp. 257-269, Sept. 2003.

[37] "Video Trace Files and Statistics," http://trace.eas.asu.edu, 2013.

[38] K. Evensen, A. Petlund, H. Riiser, P. Vigmostad, D. Kaspar, C. Griwodz, and P. Halvorsen, "Mobile Video Streaming Using Location-Based Network Prediction and Transparent Handover," *Proc. ACM 21st Int'l Workshop Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '11),* pp. 21-26, June 2011.

Ngoc Minh Do received the BS degree in information technology from the Hanoi University of Technology, Vietnam, in 2004. He is currently working toward the PhD degree in the Department of Information and Computer Science, University of California, Irvine. He was a research intern at Deutsche Telekom R&D Labs in 2010 and Alcatel-Lucent Bell Labs in 2011. His research interests include multimedia, wireless communication, mobile computing, crowdsourcing, and algorithms. He is a student member of the IEEE.

Cheng-Hsin Hsu received the BSc degree in mathematics and the MSc degree in computer science and information engineering from National Chung-Cheng University, Taiwan, in 1996 and 2000, respectively. He received the MEng degree in electrical and computer engineering from the University of Maryland, College Park, in 2003 and the PhD degree in computing science from Simon Fraser University, Burnaby, British Columbia, Canada, in 2009. He is an assistant professor at National Tsing Hua University, Hsin Chu, Taiwan. His research interests are in the areas of multimedia networking and distributed systems. He is a member of the IEEE and ACM.

Nalini Venkatasubramanian received the MS and PhD degrees in computer science from the University of Illinois, Urbana-Champaign. She is a professor in the School of Information and Computer Science, University of California, Irvine. Her research interests include distributed and parallel systems, middleware, mobile environments, multimedia systems/applications, and formal reasoning of distributed systems. She was a member of technical staff at Hewlett-Packard Laboratories in Palo Alto, California, for several years, where she worked on large-scale distributed systems and interactive multimedia applications. She has also worked on various database management systems and on programming languages/compilers for high performance machines. She is a senior member of the IEEE and ACM.