EMS: Efficient Multicast Streaming Scheme for Multicasting within Wi-Fi Hotspot

Wan-Seon Lim
ID-based Networking Section
Electronics and Telecommunications Research Institute
(ETRI) 218 Gajeongno, Yuseong-gu, Dejeon, Korea
Email: wnsn.lim@gmail.com

Kang G. Shin
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, MI 48109-2121
Email: kgshin@eecs.umich.edu

***Abstract*— We have discovered two main problems when a multicast sender and multicast receivers are associated with the same AP. First, the multicast sender cannot determine a proper sending rate with the traditional end-to-end rate-adaptation schemes. Second, the multicast sender wastes its power due to the unique characteristics of power saving mode of the IEEE 802.11 standard. We propose efficient multicast streaming (EMS) to solve these problems without modifying the AP and the multicast receivers. With EMS, the multicast sender adapts its sending rate at the application layer and adjusts the sleep cycle at the MAC layer by monitoring multicast packets sent from the AP. Our experimentation and simulation results show that EMS can improve video quality at the receivers and reduce power consumption of the multicast sender significantly.**

Keywords— IEEE 802.11, multicast, video streaming, rate-adaptation, power saving.

I. INTRODUCTION

The IEEE 802.11 wireless local area network (WLAN) [1] has been one of the most popular wireless Internet access technologies. Although the data rate of IEEE 802.11 WLAN has been increased with new standards and PHY technologies, mobile data traffic has been outgrowing the technology advances [2]. Since it is an efficient way of delivering traffic to multiple receivers, multicast over IEEE 802.11 WLANs has received significant attention. Numerous protocols have been proposed for efficient multicast in IEEE 802.11 WLANs [3-8]. These protocols assume that an access point (AP) receives multicast data packets over the Internet and the WLAN stations (i.e., multicast receivers) are associated with the AP as shown in Fig. 1(a). For this scenario, there have been proposals for efficient transmission of multicast data packets from the AP to the multicast receivers.

In this paper, as shown in Fig. 1(b), we focus on a new and interesting scenario where a multicast sender is one of WLAN stations associated with the same AP as the multicast receivers. We will henceforth call it 'Multicast Within Hotspot' (MIH). MIH can be used for various types of applications as follows.

Contents sharing: displaying stored digital contents from servers or portable devices (i.e., laptop, smartphone, and camcorder) to multiple display devices.

Real-time streaming: live streaming to multiple users (e.g., infant monitoring and environmental surveillance.)

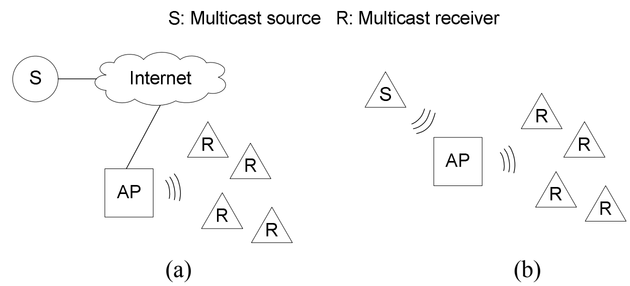


Fig. 1. Multicast scenarios: (a) traditional scenario; (b) target scenario (multicast within hotspot).

Group presentation: sharing documents and presentation slides for a group discussion during a meeting.

In MIH, a multicast packet is sent from the multicast source or streamer which unicasts packets to the AP, which then multicasts the packets to the receivers. Note that multicast transmission between the AP and the receivers can be handled by existing multicast protocols for WLANs [3-8]. Thus, we focus on the characteristics of a multicast sender in MIH which has not been addressed before. We have discovered two main problems with a multicast sender in MIH. First, the multicast sender cannot decide a proper sending rate with the traditional end-to-end rate-adaptation schemes due to the lack of feedback from multicast receivers. Second, the multicast sender unnecessarily wastes its power due to the unique characteristics of power saving mode of the IEEE 802.11 standard.

We propose *Efficient Multicast Streaming* (EMS) in order to solve these problems in MIH. EMS is composed of three modules: *monitoring module*, *rate-adaptation module*, and *sleep-scheduling module*. The monitoring module monitors the packet drop rate and the end-to-end (e2e) delay of multicast data and periodically reports the results to the rate-adaptation module. The rate-adaptation module chooses a proper sending rate for multicast based on the packet drop rate. The sleep-scheduling module adjusts a sleep cycle of WLAN interface according to the amount of multicast data to avoid wasting power. EMS does not require any modification on the AP or multicast receivers, thus facilitating its deployment and coexistence with any existing WLAN multicast protocols. We implement a prototype of EMS in a Linux-based tastbed and conduct an in-depth experimental evaluation. We also use simulation to evaluate the efficiency of EMS. Our experimental and simulation results show that 1) EMS can improve the QoS

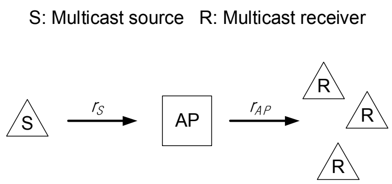


Fig. 2. Transmission rates at PHY layer in MIH scenario.

of receivers by enabling rate-adaptation in MIH and 2) reduce the power consumption of the multicast sender significantly.

The rest of the paper is organized as follows. In Section II, we motivate the need for a new solution such as EMS. We describe the design of EMS in Section III while presenting its detailed evaluation in Section IV. We discuss the related work in Section V. Finally, we discuss some future extensions to EMS and draw conclusions in Section VI.

II. Background and Motivation

A. Rate adaptation

In MIH, as shown in Fig. 2, the multicast sender sends packets to the AP by unicast transmission, and then the AP multicasts the packets to the multicast receivers. The multicast sender dynamically adjusts the PHY rate (r_S) according to the channel condition and/or feedbacks such as ACKs in order to maximize throughput. In contrast, according to the IEEE 802.11 standards, all multicast packets should be transmitted at one of the rates included in the basic PHY rate set. Therefore, the AP's PHY rate (r_{AP}) is usually lower than r_S. In addition, in the IEEE 802.11-based WLANs, the AP accesses the wireless link with the same authority as stations, even though the AP aggregates multiple downlink (i.e., AP to stations) flows. As a result, the AP easily becomes the bottleneck in MIH.

In order to avoid packet loss at the AP, the rate of the multicast sender should be adjusted at the application layer. There have been many approaches to adapting the sending rates of applications, especially for video applications, over wireless networks. One of the approaches is e2e adaptation where the sender node adjusts its sending rate to optimize the video quality subject to the bandwidth constraint. This is a traditional way to adapt video rate and can be applied to any type of network [9-12]. It requires an e2e bandwidth estimation mechanism such as TCP Friendly Rate Control (TFRC) [13] which runs at the network layer. Most of e2e bandwidth estimation mechanisms are designed for unicast-based video streaming, and the sender estimates available bandwidth based on the feedback from the receiver. However, in the case of multicast-based video streaming in MIH, they cannot be directly applied since there are multiple nodes receiving a video stream from the sender. To apply e2e adaptation to MIH, we need a new feedback mechanism for estimating the available bandwidth of all multicast receivers and its implementation by modifying the multicast sender as well as multicast receivers.

Another approach for improving video quality over WLANs is MAC-level adaptation [14-16]. In this approach, an AP transmits more important video packets with a higher probability over wireless links when the link capacity is not sufficient to transmit all video packets. Bianchi *et al.* [14] proposed a scheduling algorithm for video streaming which runs at WLAN APs. For IEEE 802.11e WLANs, several QoS-mapping architectures and algorithms have been proposed [15-16]. They aim to reduce the loss of important video packets at an AP by assigning the packets to different ACs according to their importance. Although MAC-level adaptation can respond quickly to changes of wireless channel capacity compared to e2e adaptation, it has several problems. First, since it operates at the AP side, the video sending rate from a sender cannot be adjusted. If the video sending rate is too low or too high, then video quality may be degraded or network utilization may be reduced. Second, it requires additional implementation and computation overheads in APs. Especially in the case of WLANs where most of APs are independently deployed and managed, it is impractical to modify existing APs. Third, when security protocols above the MAC layer (e.g., IPsec) are used, MAC-layer adaptation cannot be applied since APs cannot handle encrypted packets. Moreover, since it exploits the unique characteristic of video packets, it cannot be used for other types of applications over MIH.

B. Power saving in IEEE 802.11

The IEEE 802.11 standard defines two power management modes, *power saving mode* (PSM) and *active mode* (AM). A station in PSM toggles between *awake* and s*leep* states. In *awake* state, the station can transmit or receive signals while in *sleep* state the station turns off the radio components to save power. In contrast, a station in AM always runs in *awake* state. The goal of PSM is to put the WLAN interface in *awake* state only for the time periods necessary to send or receive data, and in *sleep* state when the network is idle.

The AP handles multicast and unicast packets differently when PSM is enabled. Each station informs its associated AP of whether it utilizes PSM or not. When the AP receives a unicast packet destined for a station in PSM, it buffers the packet. At every beacon interval (typically 100ms), the AP broadcasts a beacon frame that contains a Traffic Indication Map (TIM) which indicates that there is at least one frame buffered at the AP for the station. Stations in PSM switch from *sleep* state to *awake* state periodically to receive beacons. If a station recognizes the existence of a buffered frame for itself, the station transmits a power save (PS) poll frame and then waits for receiving the buffered frame. Otherwise, it returns to *sleep* state.

For multicast transmissions, the AP also buffers multicast packets when at least one station associated with it operates in PSM, even if all the multicast receivers do not use PSM. The buffered multicast packets are transmitted after sending a special beacon that contains delivery TIM (DTIM). DTIM is generated at every DTIM interval, which is usually set to 1 to 3 times of the beacon interval. In order to announce the existence of the multicast frames to stations, the AP sets an indication bit in the DTIM beacon frame to 1. After sending the DTIM beacon, the AP consecutively transmits the multicast frames until all the buffered multicast frames are sent.

The buffering mechanism in PSM does not apply to uplink transmissions (i.e., transmissions from stations to an AP) since

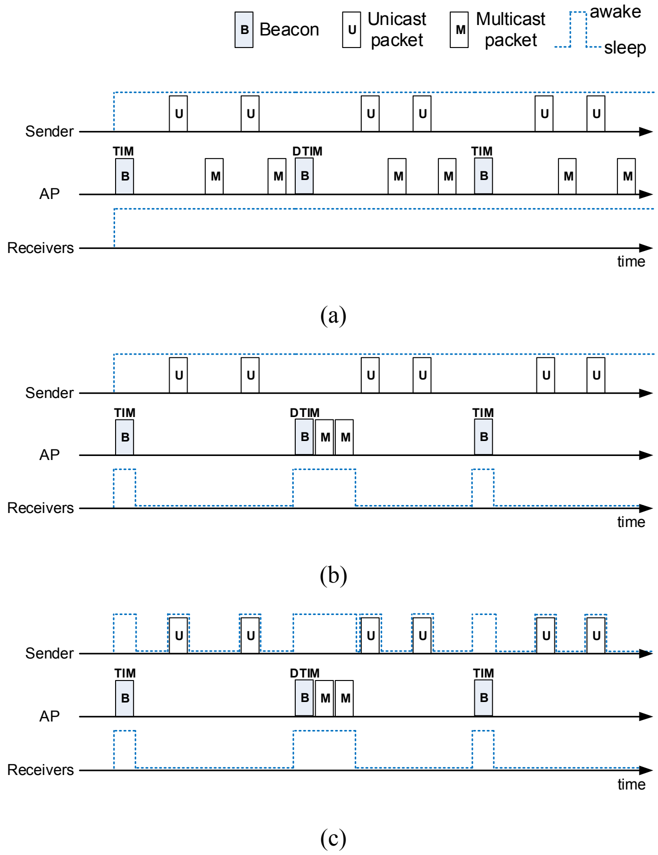


Fig. 3. Operations of multicast sender, AP, and multicast receivers under PSM: (a) The multicast sender and all other stations associated with the AP are working in AM; (b) The multicast sender is working in AM and at least one station

the AP never sleeps. If a station in *sleep* state has a packet to send, it immediately puts its WLAN interface in *awake* state and sends the packet to the AP. If a station in *sleep* state has many packets to send, then it frequently wakes up and sleeps. However, frequent transitions between *sleep* and *awake* states consume a large amount of power.

C. Motivation

As discussed above, the multicast source in MIH has two main problems. First, it cannot decide a proper sending rate with the traditional e2e rate-adaptation schemes which are designed for unicast-based streaming. To apply the existing rate-adaptation schemes to MIH, we need to modify each multicast receiver to send an individual feedback to the sender after receiving multicast packets and the sender to utilize it for the rate-adaptation. However, modifying all multicast receivers is not practical, and such an individual feedback may cause the feedback implosion problem over the wireless channel [17].

EMS aims to enable a sending rate adaptation at the multicast sender without any feedbacks from the AP or multicast receivers. In particular, the rate adaptation in EMS exploits the unique feature of MIH that the multicast sender and receivers are associated with the same AP. In MIH, the multicast sender can estimate the available bandwidth for multicast by overhearing on-going transmissions. The rate adaptation of EMS is a cross-layer approach between the application layer and the MAC layer. A rate-adaptation module at the application layer decides a proper sending rate based on the report from the monitoring module at the MAC layer.

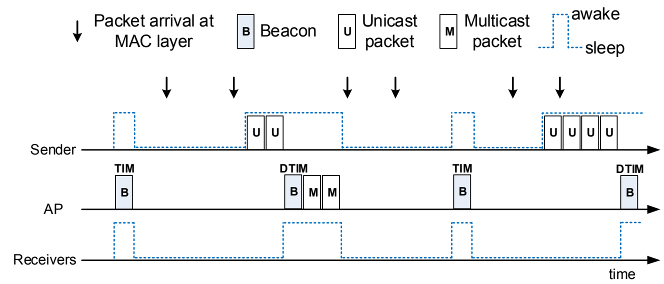


Fig. 4. Ideal operation of multicast sender under PSM

The second problem of MIH is unnecessary waste of the multicast sender's power due to the buffering mechanism at the AP. Traffic patterns in MIH under PSM can be categorized into three cases:

Case 1: The multicast source/sender and all other stations associated with the AP are working in active mode (AM).

Case 2: The multicast sender is operating in AM and at least one station associated with the AP is working in PSM.

Case 3: The multicast sender is operating in PSM.

Fig. 3(a) illustrates the operations of the multicast sender and multicast receivers in Case 1. There is no additional delay due to buffering at the AP, but the multicast sender and multicast receivers do not save power with PSM. In Case 2, the multicast sender directly sends packets to the AP since its WLAN interface is always on as shown in Fig. 3(b). The problem in Case 2 is that even if the AP receives video packets with no delay at the sender side, multicast transmissions from the AP to receivers are delayed due to buffering at the AP. In Case 3, when the sender has a packet to send, it puts its WLAN interface in *awake* state and sends the packet to the AP. If there are no more packets to send, it puts the interface in *sleep* state as shown in Fig. 3(c). Note that the AP always buffers the multicast packet in Case 3 since the sender, which is one of stations associated with it, is working in PSM. Compared to Case 2, the video sender is found able to save its power in Case 3 but frequent changes of interface state (i.e., *sleep→awake→sleep…*) wastes sender's power

Fig. 4 shows an ideal case for minimizing power consumption of the multicast sender. When a video packet arrives at the MAC layer of the sender, it stores the packet to a buffer while keeping the interface in *sleep* state. The buffered packet stays buffered for the remaining time until the next DTIM beacon reception becomes the expected transmission time of all buffered packets. Then, the multicast sender puts the interface in *awake* state and it starts sending the buffered packets. If all buffered packets are delivered to the AP before the AP transmits the DTIM beacon as shown in Fig. 4, delayed transmissions at the multicast sender do not increase the e2e delay compared to Case 3 in Fig. 3(c).

EMS aims to provide an efficient power saving mechanism for

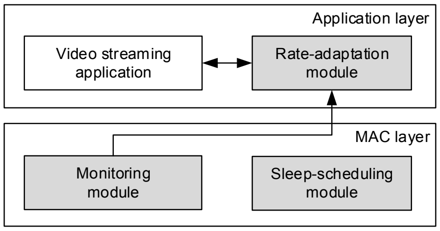


Fig. 5. System architecture of EMS. Gray boxes are new module proposed in EMS.

the multicast sender, by putting the sender's WLAN interface in *sleep* state for a longer time as shown in Fig. 4. Note that, since channel access of IEEE 802.11 WLAN is based on CSMA/CA, we cannot achieve ideal scheduling for packet transmissions. We design sleep scheduling of EMS to minimize the additional e2e delay while keeping the backward compatibility to the legacy 802.11 devices. WiFi-Direct specification which was recently introduced could be a good solution for MIH scenario since it helps devices easily deliver video with each other. However, existing Wi-Fi devices including most laptops and smartphones do not support WiFi-Direct.

One of main advantages of EMS is that it requires *modifying the multicast sender only.* This facilitates the feasibility and deployability of EMS and guarantees its cooperation/coexistence with any existing WLAN multicast protocols proposed for enhancement of multicast transmission between the AP and multicast receivers.

III. Design of EMS

In this section, we first provide a high-level overview of EMS and then describe details of its main components. A video streaming scenario is assumed for the design of EMS since it is the key application exploiting multicast. Note, however, that EMS can be applied any kind of multicast-based applications unlike video streaming-specific approaches [14-16].

A. Overview

Fig. 5 shows the system architecture of EMS which consists of *monitoring*, *rate-adaptation* and *sleep-scheduling* modules. The monitoring module measures e2e delay and the number of packets dropped at the sender itself and the AP due to queue overflow in the MAC layer. At the multicast sender, monitoring its own queue is relatively easy but monitoring the AP's queue is not, since there is no feedback from the AP. The AP's queue is estimated by overhearing multicast transmissions from the AP and obtaining the collision probability at the MAC layer. The monitoring module periodically reports the result to the rate-adaptation module.

The rate-adaptation module is placed at the application layer for easy implementation and deployment. The rate-adaptation module calculates a proper video rate based on the input from the monitoring module and sends it to the video streaming application that generates video packets from a video stream. Note that the rate-adaptation module in EMS can cooperate with any video streaming application if the video

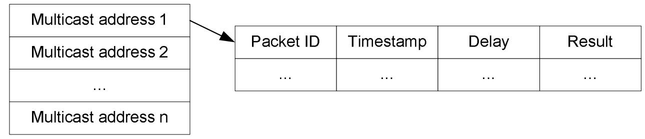


Fig. 6. Multicast maintenance table used by the monitoring module.

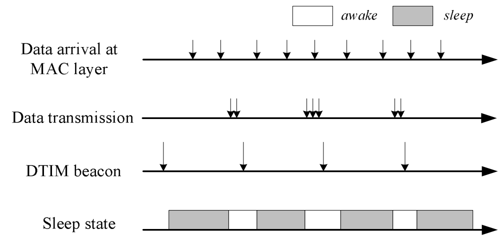


Fig. 7. Packet transmission and reception timing of video sender and change of video sender's state with the sleep-scheduling module.

streaming application has an option to adjust the video sending rate according to the available network bandwidth.

The sleep-scheduling module operates at the MAC layer and does not interact with other modules. It determines the sleep cycle of the sender's WLAN interface according to the video rate and the channel capacity. Since this module is for power savings, it is enabled only when the video sender is battery-powered.

B. Traffic Monitoring

The monitoring module maintains a table for each multicast group as shown in Fig. 6. When a video packet destined for a multicast group arrives at the MAC layer of the video sender, the monitoring module inserts an entry to the maintenance table with the identification field in the IP header of the packet stored into Packet ID field. Meanwhile, it stores the current time into the time stamp field. If multiple packets are aggregated via frame aggregation scheme in 802.11n, each of packets is mapped to a single entry.

There are four cases for a video packet generated from the application layer of the video sender to fail to be delivered to the receivers. First, the packet can be dropped at the sender's interface queue due to network congestion. Second, the packet can be lost during unicast transmission from the sender to the AP due to a collision or channel error. Third, the packet can be dropped at the AP's queue due to network congestion. Finally, the packet can be lost during multicast transmission from the AP to the multicast receivers due to a collision or channel error. For efficient rate control, packet losses resulting from network congestion (due to queue overflow and collisions) and those from channel errors should be differentiated [18].

Since the monitoring module runs at the MAC layer of the video sender, it is easy to detect the first case, i.e., the loss of a packet due to sender's queue overflow. In such a case, the Result field is set to 'fail'. If a packet is lost after trying its transmission to the AP, (i.e., second case), it may be due to either collision or the channel error. In this case, the monitoring module chooses a random real number *r* in (0, 1]. Then, if *r* is smaller than *p*/*q*, where *p* is the collision probability of arbitrary packet and *q* is the packet loss probability due to collision and channel error, the Result field is set to 'fail'. Otherwise, the entry for the packet is removed from the multicast maintenance table so that the lost packet due to the channel error does not affect the rate adaption.

The packet loss rate *q* is simply obtained by the monitoring module. If the video sender knows the number of contending nodes in the network, it can calculate the collision probability *p* based on the well-known equations [19]. However, since EMS does not receive any information from the AP and other stations, it cannot employ this method. Instead, we measure the loss rate of beacon frames and use it as the collision probability *p*. Since the frame length of a beacon is very small and its transmission rate at the PHY layer is the basic rate, the error probability of a beacon frame is negligible. Therefore, we can assume that if a beacon frame was lost, then it collided with other frames. Usually there is a beacon timer at the MAC layer for detecting missed beacons. Using this timer, the monitoring module calculates the beacon loss rate for the last 10 beacon transmissions and regards it as *p*.

Let's consider the third and the fourth cases. The monitoring module overhears every transmission over the wireless channel in order to check whether the AP successfully sends multicast packets to receivers. Even if a multicast packet is detected at the sender, it could be lost at multicast receivers due to the channel error since the channel condition varies. However, since the AP sends multicast packet with the basic rate at the PHY layer, the channel error probability is relatively low. So we can assume that, if a multicast packet is detected at the monitoring module, then receivers will succeed in receiving the packet. In such a case, the monitoring module checks if the multicast address is found in its table. If yes, it retrieves the entry by the packet ID field and sets the Results field to 'success'. Meanwhile, the Delay field is set to the difference between the current time and the time stored in the Timestamp field.

The monitoring module periodically reports the packet-loss rate and average delay to the rate-adaptation module based on a report timer.[1] There is a dedicated report timer for each of multicast group. When a report timer for multicast group *G* expiries, the monitoring module finds the last entry l_G which has the newest Timestamp value among the entries with 'success' Result field for *G*. Again, we assume the packets not detected by the sender are lost due to either queue overflow or collisions, not the wireless channel error. So, for all previous entries with Timestamp older than l_G and unspecified Result field, their Result fields are set to 'fail'.

The packet-loss rate of a multicast group *G* (R_G) is calculated as

$$R_G = \frac{N_G^f}{N_G^s + N_G^f} \qquad (1)$$

where N_G^s and N_G^f are the number of entries with 'success' Result field and 'fail' Result field for the multicast group *G*, respectively. The average delay of a multicast group G (D_G) is the average of Delay field of the entries with 'success' Result field for *G*. After reporting R_G and D_G, the monitoring module deletes all entries whose Results field is 'success' or 'fail', and restarts the report timer. The value of the timer is set to R_G.

As stated above, traffic monitoring in EMS is based on two basic assumptions; all multicast packets use a low PHY rate and beacon frame loss can be used to approximate data packet collision. When these assumptions are not met, the monitoring results may result in inefficient sleep scheduling and rate adaptation. According to the IEEE 802.11 standard, multicast transmissions need to be performed with one of the 'Basic Service Set' rates. Even though some WiFi APs allow users to select PHY rate for multicast, most users do not change the PHY rate in practice. The effectiveness of collision rate estimation with the monitoring beacon frame is discussed in the literature [31, 32]. The beacon-based estimation is simple and practical for estimating collision rate, especially when it is not possible to utilize any feedback from APs and receivers.

C. Rate Adaptation

There have been numerous video adaptation mechanisms proposed for video streaming. Of them, TCP Friendly Rate Control (TFRC) is a widely used rate-control mechanism. TFRC is designed to fully utilize network capacities while maintaining fair resource sharing with TCP flows and achieve small rate fluctuations. The rate-adaptation module in EMS employs the TCP-friendly approach of TFRC.[2] In TFRC, the receiver periodically sends a feedback report indicating the rate of recent loss events to the sender. Upon receiving the feedback, the sender updates the round trip time (RTT). The rate-adaptation module in EMS exploits R_G and D_G obtained from the monitoring module for calculation of a proper sending rate.

There are two different phases in the video rate adaptation of EMS. In the initial slow start phase, the rate-adaptation module starts with one packet per second and tries to double its sending rate *K* upon reception of feedback from the monitoring module. Upon detection of the first packet loss (i.e., $R_G > 0$), the rate- adaptation module enters the congestion avoidance phase.

During the congestion avoidance phase, the next sending rate K is determined to be the minimum of twice the previous receiving rate J and the TCP rate K_{TCP} as calculated with the TCP throughput equation:

$$K = \min(\, 2 \times J, K_{TCP}). \qquad (2)$$

[1] Initially, the timeout value of the report timer is set to 1 second.

[2] We propose a general architecture for enabling rate-adaptive multicasting in mobile hotspots and it can cooperate with not only TFRC but also other existing rate adaptation schemes. For example, TFRC variants designed for wireless networks such as [32] could be used for performance enhancement.

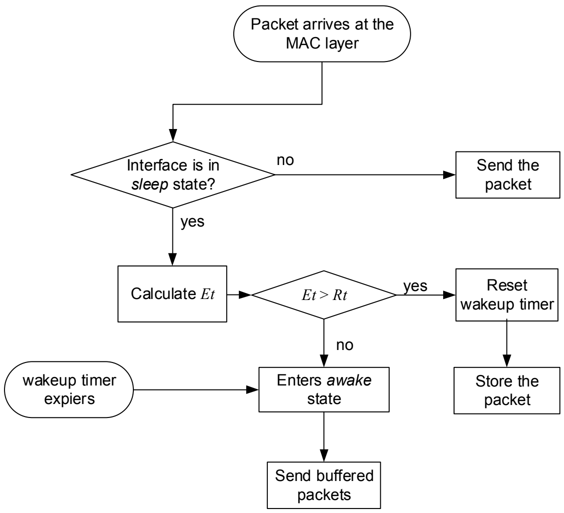


Fig. 8. Flowchart of the sleep-scheduling module

The receiving rate J is obtained by $K'(1 - R_G)$ where K' is the previous sending rate. To calculate K_{TCP}, we use a simplified version of the throughput equation for Reno TCP:

$$K_{TCP} = \frac{s}{RTT\sqrt{\frac{2p}{3}} + t_{RTO}\left(3\sqrt{\frac{3p}{8}}\right)p(1 + 32p^2)} \tag{3}$$

where s is the packet size in bytes and RTT is two times of D_G and t_{RTO} is four times of RTT. By limiting K to K_{TCP} we expect that a multicast flow and a unicast TCP flow fairly share the wireless channel. See [13] for details of (2) and (3).

It is assumed that, as shown in Fig. 5, the video streaming application can adjust the video sending rate according to the feedback from the rate-adaptation module. However, if the video streaming application does not have the adaptation capability or the minimum video rate supported by the application is larger than the currently requested rate K, then the rate-adaptation module can selectively drop video packets according to their importance. For example, if the rate-adaptation module drops video packets of B-frame before dropping the video packets of I-frame or P-frame, video quality can be improved. The rate-adaptation module in EMS can utilize the various packet dropping polices proposed in the literature.

D. Sleep Scheduling

The basic idea of power saving in EMS works as follows. The video sender applies the buffering mechanism just as the AP does. When a video packet arrives at the MAC layer of the video sender, the sleep-scheduling module stores the packet to a buffer while keeping the interface in *sleep* state. This packet stays buffered for the remaining time until the next DTIM beacon reception which becomes the expected transmission time of all buffered packets. The sleep-scheduling module then puts the interface in *awake* state and the video sender starts sending the buffered video packets. If all packets are delivered to the AP before the AP transmits the DTIM beacon, delayed transmissions at the video sender do not increase the e2e delay between the video sender and multicast receivers.

Fig. 7 shows the transmission and reception timing of a packet and the change of the video sender's state with the sleep-scheduling module. One can see that the video sender does not enter *sleep* state right after receiving the DTIM beacon but stay in *awake* state to receive multicast packets from the AP. This is because if the video sender's interface is in *sleep* state right after receiving the DTIM beacon, the monitoring module which requires channel overhearing would not work properly. After receiving all multicast packets, the sender enters *sleep* state and the monitoring module stops working

Fig. 8 describes the detailed operation of the sleep-scheduling module. When a packet arrives at the MAC layer, the sleep-scheduling module first checks the state of interface. If it is in *awake* state, the packet is directly sent according to the legacy IEEE 802.11 procedures. Otherwise, the sleep-scheduling module updates the expected transmission time E_t of buffered packets:

$$E_t = E_t' + (T_o + T_{data} + tSIFS + T_{ack}) \tag{4}$$

where $E_t{}'$ is the previous value of E_t, T_o is the time overhead for packet transmission, T_{data} and T_{ack} are transmission times for the data and ACK frame, respectively, and *tSIFS* is the length of SIFS. T_o is the time for binary backoff and retransmissions. Initially, T_o is set to

$$T_o = \frac{CW_{min} \times SlotTime}{2} \tag{5}$$

and updated after sending buffered video packets. (How to update T_o will be discussed later.) T_{data} can be expressed as:

$$T_{data} = tPLCP + \frac{(lMACOverhead + l) \cdot 8}{S} \tag{6}$$

where $tPLCP$ is the transmission times for PLCP preamble and header, $lMACOverhead$ and l are the lengths of the overhead of MAC frame (i.e., the MAC header and FCS) and the payload length, respectively, and S is the current data rate at the PHY layer. The sleep-scheduling module then checks if E_t is less than the remaining time to the next DTIM beacon reception (R_t). If yes, the sleep-scheduling module puts the interface in *awake* state and consecutively sends buffered packets. Otherwise, there is still enough time to send buffered packets. In such a case, the sleep-scheduling module stores the packet and resets the wakeup timer. A timeout value of the wakeup timer is set to $(E_t - R_t)$. When the wakeup timer expires, the interface wakes up and the buffered packets are transmitted.

When the video sender sends the buffered packets, the sleep-scheduling module measures the total transmission time of the packets. Let M_t be the measured total transmission time. The sleep-scheduling module updates T_o by

$$T_o = T_o{}' + (M_t - E_t)/n \tag{7}$$

where $T_o{}'$ is the previous value of T_o, n is the number of transmitted buffered packets.

Table I. Power consumption rates of WLAN card in different PHY state

PHY state	Tx	Rx	listen	sleep
Consumption rate	2W	0.9W	0.8W	40mW

IV. IMPLEMENTATION AND EVALUATION

A. Implementation

We have implemented a prototype of EMS to evaluate its effectiveness. Given below is a brief overview of this implementation. We used a laptop running on the Ubuntu Linux 12.04. It has an internal mini-PCI type card which embeds Atheros 5007 chipsets. The MadWifi, an open source Linux driver for Linux, is used as an interface driver.

In order to implement the monitoring module, we have modified the MadWifi driver since it runs at the MAC layer. This module captures the multicast packets generated from the application layer based on the destination address field of IP header. In addition, it captures all multicast packets transmitted from the AP and finds packets whose source address field of IP header is the same as its IP address.

The rate-adaptation module is implemented as a user-level application. An internal socket API is used to send feedback from the monitoring module to the rate-adaptation module. For experimentation, we have also developed a video streaming application which uses the scalable video codec (SVC) [20]. A video sequence is encoded in a base layer and several enhancement layers with SVC. According to the video rate information provided by the rate-adaptation module, the streaming application adjusts the number of layers to be transmitted. Forward error correction (FEC) schemes for video streaming such as [33] are not used.

The sleep scheduling module is implemented by modifying the MadWifi driver. One practical challenge in implementing the sleep scheduling module is to put the interface in low-power state. Unfortunately, the MadWifi driver does not provide an API to control the power state, so we cannot directly measure how much energy is consumed by the WLAN interface. In our experiments, we measure Tx time, Rx time, listen time (time for listening to the idle channel), and sleep time, and then translate the measured time distribution to the energy-efficiency measure using the WLAN cards' power consumption rates. Table I shows the power consumption rates in different PHY states that we obtained from [21]. We also accounted for the cost of state transitions. We assume the wakeup operation takes 1600 μs and the sleep operation takes 400 μs. Thus, the total overhead of a wakeup and sleep is about 2ms [22]. We also assume that the power consumption rate during the transition is the same as that during listening (0.8W).

B. Experimental results

Our testbed consists of an AP, a video sender, four multicast receivers, an FTP server and two FTP clients. During experiments, all multicast receivers belong to the same multicast group and the video sender streams a video to the multicast group. Each of FTP clients uploads a large file from the FTP server to generate background traffic. All entities run on the Linux operating system (with 3.3.0 kernel). Each of entities is equipped with an 802.11a/b/g card and uses the MadWifi driver. PHY modes of interfaces are set to 802.11g and the beacon interval is set to 100ms. The RTS/CTS exchange is disabled.

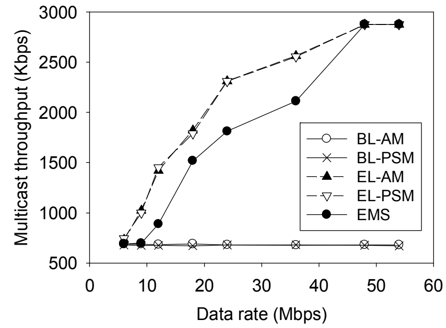


Fig. 9. Multicast throughput comparison with varying PHY data rate *D*

A test video sequence is the well-known Soccer sequence at 4CIF resolution and 30fps, which is repeated for 50 seconds. By using the Joint Scalable Video Model (JSVM) software [23], we encoded the 50s-long video with three quality-layers: one base-layer (BL) and two enhancement layers (EL1 and EL2). The bit rates of BL, EL1, and EL2 are 689Kbps, 829Kbps, and 1358Kbps, respectively. The GOP length is 15. For error concealment of decoded video, we use the Scalable Video coding streaming Evaluation Framework (SVEF) [24]. The length of play-out buffer is set to 3 seconds.

EMS is compared to the following four cases. 1) The video sender sends only the base-layer and uses the AM. 2) The video sender sends only the base-layer and uses the PSM. 3) The video sender sends all layers and uses the AM. 4) The video sender sends all base-layer and uses the PSM. These four cases are denoted by BL-AM, BL-PSM, EL-AM, and EL-PSM, respectively. The multicast receivers, FTP server, and FTP clients always operate in AM.

According to the IEEE 802.11 standard, the AP sends multicast packets with 6Mbps, which is one of base data rates. The PHY data rate for unicast transmissions is fixed at 6Mbps except for the video sender's transmission. The PHY data rate, *D*, of the video sender, changes during the experiments. Since all entities are located close to each other, packet loss due to error over the wireless channel occurs rarely.

Fig. 9 shows the multicast throughput which is given by the average throughput of multicast receivers. We can see that the multicast throughput is almost constant in the case of BL-AM and BL-PSM. This is because the available bandwidth between the sender and receivers is larger than the requested bandwidth for streaming the base layer, regardless of *D*. The multicast throughputs of EL-AM, EL-PSM, and EMS are increased with the increase in *D*. EMS achieves a lower throughput than EL-AM and EL-PSM due to its video rate-adaptation.

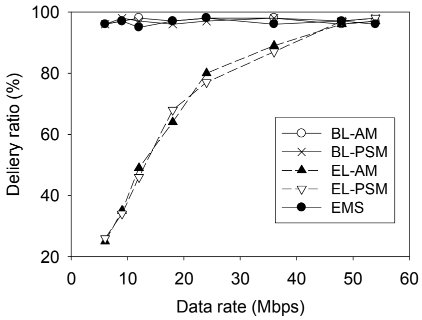


Fig. 10. Delivery ratio comparison with varying PHY data rate *D*

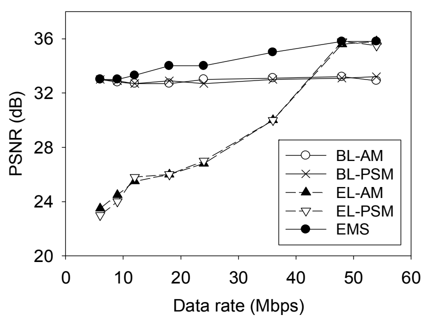


Fig. 11. PSNR comparison with varying PHY data rate *D*

Fig. 10 shows the average delivery ratio measured at the multicast receivers. The average delivery ratio is defined as the ratio of the number of video packets received by the receiver to that of video packets transmitted by the video sender. In BL-AM and BL-PSM, more than 95% packets are successfully delivered to receivers in all cases. Since the video sender only send the base-layer in BL-AM and BL-PSM, the network is not congested. A small fraction of video packet are lost due to collisions at the MAC layer. In contrast, EL-AM and EL-PSM show a low delivery ratio especially when *D* is low. If *D* is lower than 48Mbps, the available bandwidth between the video server and the multicast receivers is not enough to send all enhancement layers. EMS, with the help of its rate adaptation, achieves almost similar throughput as EL-AM and EL-PSM. Note that packet losses due to collisions cannot be avoided with the rate adaptation. To solve this problem, we need a retransmission scheme for multicast transmission such as [3-5]. Since EMS is a sender-based solution, it can cooperate with the retransmission schemes to increase the delivery ratio up to 100%.

Now, we present a quantitative assessment of video quality perceived by a user using Peak Signal-to-Noise Ratio (PSNR). PSNR is commonly used as a quality measurement between the original and reconstructed video, where higher PSNR means better video quality. Fig. 11 shows the average PSNR measured at the multicast receivers. We can see that the PSNR is closely related to the delivery ratio, not to multicast throughput. When *D* is lower than 48Mbps, PSNR of EL-AM and EL-PSM is much lower than that of BL-AM and BL-PSM because many video packets containing base-layer frames are lost in of EL-AM and EL-PSM. It means that if the video sender does not know a proper video rate, reducing its video rate may provide receivers better video quality. EMS shows the

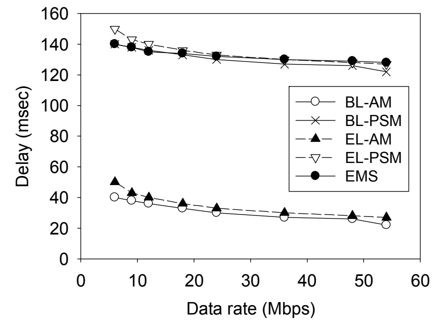


Fig. 12. Delay comparison with varying PHY data rate *D*

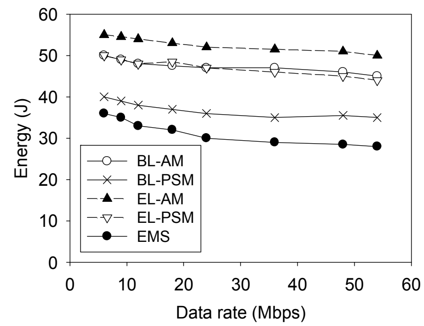


Fig. 13. Consumed energy comparison with varying PHY data rate *D*

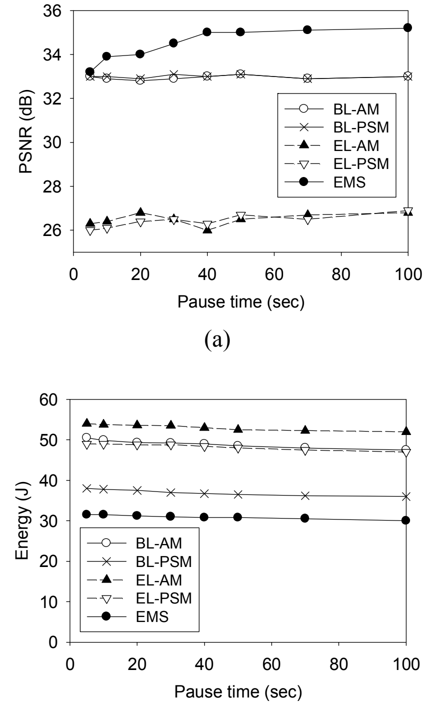
best performance in terms of PSNR. Since EMS improves the multicast throughput without decreasing the delivery ratio, it outperforms BL-AM and BL-PSM when $D > 9$Mbps.

In order to evaluate the effectiveness of sleep scheduling in EMS, we measured the average e2e delay and the consumed energy. As shown in Fig. 12, the delay is significantly reduced when the video server is operating in AM. In cases of BL-AM and EL-AM, the AP does not buffer multicast video packets since there is no PSM device. When PSM is applied to the video sender, (i.e., BL-PSM and EL-PSM), the delay is significantly increased due to an additional delay caused by the buffering mechanism. EMS achieves similar delays as BL-PSM and EL-PSM so EMS does not have negative impact on delays.

Fig. 13 shows energy consumed by the video sender's WLAN interface until transmitting the last video packet. Of BL-AM, BL-PSM, EL-AM, and EL-PSM, BL-PSM consumes the least energy since the video sender transmits only the base-layer and the video sender's WLAN interface is allowed to enter *sleep* state. EMS consumes less energy than BL-PSM although the video sender sends more packets according to the video rate-adaptation. EMS saves power up to 45% over EL-AM, and up to 20% over BL-PSM. From Figs. 11 and 13, we can conclude that EMS can improve the quality of video at multicast receivers while saving the video sender's power.

C. Simulation results

In order to evaluate the performance of EMS under dynamic channel conditions, we simulated it using the ns-2 simulator. The set of MAC parameters and the network topology are the same as the experimental settings in the previous subsection, except for the mobility of the video sender.



(a)

(b)

Fig. 14. Simulation results: Performance comparisons with varying pause time. (a) PSNR. (b) Consumed energy.

In particular, the video sender moves within a circle whose radius is 400m according to the random waypoint model while the AP is located at the center of the circle. We fixed the speed of station used by the random waypoint model to 10m/s. Other entities except the video sender do not move during the simulation. For the PHY modeling, we use the two-ray ground propagation model. The transmission power and receive sensitivity for each transmission rate are obtained from the data sheet of 3COM 11G USB adaptor [25]. We also use the Ricean fading model with Ricean *K* factor of 3 dB to reflect the short-term multi-path fading effect. The video sender adjusts the PHY data rate *D* based on the auto-rate fallback (ARF) mechanism [26].

As shown in Fig. 14(a), PSNR of BL-AM, BL-PSM, EL-AM, and EL-PSM is not affected much by the mobility of the video sender. This is because the video sending rate is fixed regardless of fluctuation of PHY data rate. In contrast, PSNR of EMS is affected by the mobility of the sender. When the pause time decreases (i.e., the sender's mobility increases), the channel condition changes quickly and thus the bandwidth estimation by the rate-adaptation module becomes inaccurate. However, except when the pause time is 5 seconds, EMS shows the best PSNR compared to others.

Fig. 14(b) shows consumed energy by the video sender's WLAN interface. In all schemes, the energy consumption increases as the pause time decreases. When the sender's mobility is high and thus the channel condition changes dynamically, retransmissions at the MAC layer occur frequently due to the wireless channel error. Such retransmissions waste the video sender's power. However, the effect of mobility on the power consumption is not high.

V. Related Work

Enhanced multicast protocol for WLANs. There has been extensive research into improving the performance of multicast transmission over IEEE 802.11 WLANs. In IEEE 802.11 WLANs, multicast frames are simply broadcast without ARQ mechanisms. Therefore, multicast frame losses frequently occur due to channel errors or frame collisions. To overcome this limitation, various ARQ mechanisms have been proposed. Tang *et al.* proposed Broadcast Medium Window (BMW) that treats a multicast transmission as multiple unicast transmissions to each receiver [3]. In the Batch Mode Multicast MAC protocol (BMMM) [4], an AP sequentially exchanges RTS/CTS with receivers and then transmits a multicast data frame. The NAK-based ARQ mechanism is proposed in the Leader-Based Protocol (LBP) [5]. The main idea of NAK-based ARQ mechanism is reducing the control frame overhead by allowing an AP to receive multiple feedbacks from receivers in a single time slot. Another main problem of WLAN multicast is that a low and fixed transmission rate is used for multicast transmissions. Existing rate-adaptation mechanisms for unicast transmissions cannot be directly applied in multicast scenarios due to the absence of MAC-level feedbacks (CTS or ACK) in multicast. In order to make the AP obtain feedbacks from multicast receivers, supplementary MAC-level signaling methods are needed. In [6], Choi *et al.* proposed the Leader-based Multicast with the Auto Rate Fallback (LM-ARF) protocol that combines the NAK-based ARQ mechanism of LBP with the rate-adaptation mechanism of ARF. Rate Adaptive Multicast (RAM) [7] and Auto-Rate Selection mechanism for Multicast (ARSM) [8] adopt the closed-loop approach, i.e., the AP determines the transmission rate based on the explicit channel feedback from the receivers. In contrast to the existing multicast protocols for WLANs [3-8] which aim to optimize transmissions from the AP to multicast receivers, EMS is designed for efficient transmissions from the multicast source to the AP. EMS and the existing multicast protocols can be used together without creating any conflict.

Sleep during data transfer. In PSM-throttling [27], a client reshapes the TCP traffic into periodic bursts, and then it sleeps and wakes up at the right time based on the predicted packet arrival time in order to minimize the energy consumption. Traffic reshaping is done in two steps, at the TCP and the MAC layers. Catnap [28] exploits the bandwidth discrepancy between the broadband and the Wi-Fi network to save energy on mobile devices by combining small gaps between packets into meaningful sleep intervals. It requires modification not only on the AP and the client but also on the server. Catnap targets data-oriented applications, such as web and file transfers, which are not delay-sensitive. Both PSM-throttling and Catnap are designed to save power for downlink transmissions, not for uplink transmissions. Therefore, they cannot be applied to a video sender in the MIH scenario.

Cross layer approach for MAC layer adaptation. Sen *et al.* proposed a system, called Medusa, for efficient delivery of high quality media to one or more WiFi clients [29]. In Medusa, packet transmission parameters such as retransmissions, PHY rate selection, and packet transmission order are decided in a content-dependent manner. In contrast to EMS, Medusa requires client-side modifications. In addition, it does not affect power consumption of clients. Dyson [30] proposed a software architecture in which a centralized controller gives pertinent information such as radio channel conditions to APs and clients. Dyson is a good solution for optimizing client associations, handling VoIP clients, and reserving airtime for specific users but it cannot support the MIH scenario efficiently.

VI. CONCLUSION

In this paper, we proposed a new solution for efficient multicast streaming in the 'Multicast Within Hotspot' (MIH) scenario, called EMS, which utilizes the unique features of IEEE 802.11 multicast transmission. In EMS, the multicast sender dynamically adjusts its sending rate without any feedback from the AP or multicast receivers. In addition, it puts its WLAN interface in *sleep* state for a longer time by buffering packets at the multicast sender. Via experimentation in a real IEEE 802.11 WLAN testbed and simulations, we have demonstrated the effectiveness and feasibility of EMS. In future, we plan to extend EMS for supporting multicast over other types of wireless networks. We will also study the effectiveness of EMS in large-scale WLANs such as enterprise WLANs.

Acknowledgements

The work reported in this paper was supported in part by the NSF under Grants 1160775 and 1317411.

REFERENCES

[1] IEEE 802.11-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 1999.

[2] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012–2017 http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf

[3] K. Tang and M. Gerla, "MAC reliable broadcast in ad hoc networks," In Proceedings of IEEE MILCOM'01, Oct. 2001.

[4] M.T. Sun, L. Huang, A. Arora, and T.H. Lai, "MAC layer multicast in IEEE 802.11 wireless networks," In Proceedings of the ICPP'02, Aug. 2002.

[5] J. Kuri and S. K. Kasera, "Reliable Multicast in Multi-access Wireless LANs," ACM Wireless Networks, vol. 7, no. 4, 2001.

[6] S. Choi, N. Choi, Y. Seok, T. Kwon, and Y. Choi, "Leader-based Rate Adaptive Multicasting for Wireless LANs" In Proceedings of IEEE GLOBECOM'07, Nov. 2007.

[7] A. Basalamah, H. Sugimoto, and T. Sato, "A Rate Adaptive Multicast Protocol for Providing MAC Layer Reliability inWLANs," IEICE Trans. Commun., vol. E89-B, no. 10, 2006.

[8] J. Villalón, P. Cuenca, L. Barbosa, Y. Seok, and T. Turletti, "Cross-Layer Architecture for Adaptive Video Multicast Streaming Over Multirate Wireless LANs", IEEE J. Select. Areas Commun., vol. 25, no. 4, 2007.

[9] M. Chen, and A. Zakhor, "Rate Control for Streaming Video over Wireless", IEEE Wirel. Commun., Aug 2005.

[10] Y. Fu, R. Hu, G. Tian, and Z. Wang, "TCP-Friendly Rate Control for Streaming Service over 3G Network", In Proceedings of WiCOM 2006, Sept. 2006.

[11] B. Görkemli, M. O. Sunay, and A. M. Tekalp, "Video Streaming over Wireless DCCP", In Proceedings of IEEE ICIP 2008, San Diego, USA., Oct. 2008.

[12] K. Tappayuthpijarn, G. Liebl, T. Stockhammer, and E. Steinbach, "Adaptive Video Streaming over a Mobile Network with TCP-Friendly Rate Control", In Proceedings of IWCMC, Leipzig, Germany, June 2009.

[13] M. Handley, S. Floyd, J. Padhye, and J. Widmer, TCP Friendly Rate Control (TFRC): Protocol Specification. IETF, 2003, RFC 3448.

[14] G. Bianchi, A. Detti, P. Loreti, C. Pisa, "Cross-layer H.264 Scalable Video Downstream Delivery Over WLANs", In Proceedings of IEEE WoWMoM, Motreal, Canada, June 2010.

[15] L. Xiaofeng, R. Tonghua, and X. Jin, "A cross-layer design for transmission of scalable H.264 video over IEEE 802.11e networks", In Proceedings of ICCP, Cluj-Napoca, Romania 2010.

[16] H.-L Chen, P.-C Lee, and S.-H Hu, "Improving Scalable Video Transmission over IEEE 802.11e through a Cross-layer Architecture", In Proceedings of IEEE ICWMC, Athens, Greece, July 2008.

[17] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen and J.-P. Sheu "The broadcast storm problem in a mobile ad hoc network", In Proceedings of IEEE/ACM MOBICOM, 1999.

[18] S. Pack, X. Shen, J. W. Mark, and L. Cai, "A Two-Phase Loss Differentiation Algorithm for Improving TFRC Performance in IEEE 802.11 WLANs", IEEE Transactions on Wireless Communications, vol. 6, no. 11, pp. 4164-4175, 2007.

[19] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," IEEE J. Sel. Areas Commun., vol. 18, no. 3, pp. 535–547, Mar. 2000.

[20] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard", IEEE Trans. Circuits Syst. Video Technol. 17, Sep. 2007.

[21] Power Consumption and Energy Efficiency Comparisons of WLAN Products. http://www.atheros.com/media/resource/resource_15_file2.pdf.

[22] Ki-Young Jang , Shuai Hao, Anmol Sheth, and Ramesh Govindan, "Snooze: energy management in 802.11n WLANs", In Proceedings of CONEXT, p.1-12, Tokyo, Japan, Dec. 2011.

[23] Joint Scalable Video Model: http://ip.hhi.de/imagecom G1/savce/downloads/SVC-Reference-Software.htm

[24] SVEF: Scalable Video-streaming Evaluation Framework: http://svef.netgroup.uniroma2.it/

[25] 3Com OfficeConnect Wireless 54 Mbps 11g Compact USB Adapter: http://pdfs.icecat.biz/pdf/304374-6833.pdf

[26] A. Kamerman and L. Monteban, "WaveLAN II: A high-performance wireless LAN for the unlicensed band," Bell Labs Technol. J., vol. 2, no. 2, pp. 118–133, 1997.

[27] E. Tan, L. Guo, S. Chen, and X. Zhang, "PSM-throttling: Minimizing energy consumption for bulk data communications in WLANs", In Proceedings of ICNP, Beijing, China, Oct. 2007.

[28] F. Dogar, P. Steenkiste, and K. Papagiannaki, "Catnap: exploiting high bandwidth wireless interfaces to save energy for mobile devices", In Proceedings of MobiSys, San Francisco, USA, June 2010.

[29] Sayandeep Sen, Neel Kamal Madabhushi, and Suman Banerjee, "Scalable WiFi media delivery through adaptive broadcasts", In Proceedings of NSDI, San Jose, USA, Apr. 2010.

[30] R. Murty, A. Wolman, J. Padhye, and M. Welsh, "Dyson: An Architecture for Extensible Wireless LANs", In Proceedings of USENIX ATC, Boston, USA, June 2010.

[31] Z. Dou, Z. Zhao, Q. Jin, L. Zhang Y. Shu and O. Yang, "Energy-Efficient Rate Adaptation for Outdoor Long Distance WiFi Links", In Proceedings of IEEE INFOCOM Workshop on GCN, Sanghai, China 2011.

[32] S. Pack, X. Shen, J. W. Mark, and L. Cai, "A Two-Phase Loss Differentiation Algorithm for Improving TFRC Performance in IEEE 802.11 WLANs", IEEE Trans. on Wireless Commun., vol. 6, no. 11, Nov. 2007.

[33] S. Won, "Graceful Degradation FEC Layer for Multimedia Broadcast/Multicast Service in LTE Mobile Systems", ETRI Journal, vol. 35, no. 6, Dec. 2013