

Implementing a Cooperative MAC Protocol for Wireless Video Multicast

Özgül Alay, Zhe Xu, Thanasis Korakis, Yao Wang and Shivendra Panwar

Department of Electrical and Computer Engineering, Polytechnic Institute of NYU, Brooklyn, New York

E-mails: ozgu@vision.poly.edu, zxu01@students.poly.edu, {korakis,yao}@poly.edu, panwar@catt.poly.edu

Abstract—Wireless video multicast enables delivery of popular events to many wireless users in a bandwidth efficient manner. However, providing good and stable video quality to a large number of users with varying channel conditions remains elusive. In our previous work, we integrated layered video coding with cooperative communication to enable efficient and robust video multicast in infrastructure-based wireless networks [1]. Through simulation and analysis, we showed that cooperative multicast improves the multicast system performance and the coverage area. In this work, we integrate the proposed system with packet level Forward Error Correction (FEC) and evaluate the viability of the system in a realistic environment. We implement the system at the MAC layer and report the experimental results in a medium size (i.e., 8 stations) testbed. The experimental results confirm that the new cooperative MAC protocol for multicast, delivers superior performance.

Index Terms: layered video coding, MAC, user cooperation, FEC, wireless video multicast

I. INTRODUCTION

In recent years, the demand for video applications over wireless networks has risen with the increase in both the bandwidth of wireless channels and the computational power of mobile devices. To provide efficient delivery among a group of users simultaneously, multicast has been used as an effective solution, as it saves network resources by sharing the data streams across receivers. However, providing good and stable video quality to a large number of users with varying channel conditions remains elusive due to the high packet loss ratio and bandwidth variations of wireless channels.

Wireless channels can be characterized by their bursty and location dependent errors. Hence, each user in a multicast system will most likely lose different packets. Therefore, a simple ARQ (Automatic Repeat reQuest) based scheme is not appropriate for video multicast over wireless channels since it can cause a large volume of retransmissions. There are several studies discussing error control in video multicast over wireless networks [2],[4]. In a multicast scenario, heterogeneity among clients is another issue since each receiver has a different connection quality and power limitation. Scalable (layered) video coding is one approach to solve the heterogeneity problem. Several researchers have studied layered video multicast in infrastructure-based wireless networks [5]-[6]. Moreover, video multicast over ad-hoc networks have been considered in [7],[8], where the use of multiple description video is proposed to overcome the unreliability of wireless links. However, none of these papers consider the use of cooperation.

Generally, multicast receivers may have very different channel quality, with ones closer to the sender having better quality on average and far away receivers having poor quality. In a conventional multicast system, the sender adjusts its transmission rate to the user with the worst channel condition. Hence, the system is severely affected by path loss and multipath fading. User cooperation is one effective technique to combat path loss and fading where terminals process and forward the overheard signal transmitted by other nodes to their intended destination [9]. The initial attempts for developing cooperative communications focused on physical (PHY) layer schemes where the destination can improve its ability to decode the original packet, combining different copies of the same signal transmitted by the source and different relay stations. However, cooperation at the physical layer encounters several formidable obstacles, when system realization is concerned. First and foremost, joint decoding at the receiver is plausible, only if an accurate synchronization can be maintained among all the stations involved in the communication, which is difficult to cope with in reality. Secondly, the cooperative coding scheme is fundamentally different from the conventional ones implemented in commercial wireless products (e.g., IEEE 802.11) and therefore it demands a total overhaul for the existing design of physical layer hardware, which is yet another daunting undertaking.

In our previous work, we designed and implemented a cooperative MAC-layer protocol, called CoopMAC, for point-to-point transmission [10],[11]. The CoopMAC protocol is based on a scheme under which a source station that experiences a poor quality channel with a specific destination, chooses to use a helper station as a relay, sending its data in a two-hop manner instead of transmitting directly to the destination at a low rate. The cooperative scheme takes advantage of the fact that the helper is selected in a way such that the two links (source to helper and helper to destination) have high transmission rates. We showed that the protocol performs better than the standard single hop 802.11 MAC protocol [12]. We argue that it is more advantageous to consider user cooperation in multicast, since the relays are part of the intended recipients of the sender transmission and hence are free from the incentive and security concerns that have hindered the practical deployment of cooperation for point-to-point communications.

Recently, we proposed a cooperative multi-hop transmission scheme to enable efficient and robust video multicast in infrastructure-based wireless networks [1]. The basic idea behind the system is that we divide all the receivers into two groups such that receivers in Group 1 have better average channel quality than those in Group 2. For the multicast transmission, we let the sender choose its transmission rate based on the worst channel quality of Group 1. Then, selected receivers in Group 1 will relay the received information

¹This work is supported in part by National Science Foundation (NSF) under award 0430885, and the New York State Center for Advanced Technology in Telecommunications (CATT). The work is also supported by Wireless Internet Center for Advanced Technology (WICAT), an NSF Industry/University Research Center at Polytechnic Institute of NYU.

seconds for their transmissions, respectively, such that $T = T_1 + NT_2$ where N stands for the number of relays. In other words, the sender transmits over $t_1 = T_1/T$ proportion of each transmission interval, and each relay sends over $t_2 = T_2/T$ proportion and they satisfy the constraint $t_1 + Nt_2 = 1$. We choose the relay positions and a minimal N such that we can cover an area of a radius r_{ext} such that $r_{ext} > r_d$.

Since each relay transmits at a different time slot, the received video rates observed by the receivers will be different from the physical layer transmission rate. We can express the received video rate for Group 1 and Group 2, R_{v_1} and R_{v_2} , as

$$R_{v_1} = t_1 \gamma_{FEC}(r_1, R_1) \beta R_1 \quad (1)$$

$$R_{v_2} = t_2 \gamma_{FEC}(r_2, R_2) \beta R_2 \quad (2)$$

where β is defined as the effective data ratio which is the ratio of the time spent to transmit the actual payload data to the total transmission time. Note that β depends on the overhead due to packet headers as well as the actual data transmission time over the air time.

Note that with the general system set up described so far, the video rate delivered to Group1 and Group2 users may differ. This can be realized by coding a video into a layered stream, so that relays only forward a subset of layers. This way, we can make users in Group 1 get much better quality than that offered by direct transmission, whereas users in Group 2 get video quality better than or similar to direct transmission.

III. IMPLEMENTATION EFFORTS

We implement cooperative multicast system along with FEC on our experimental testbed [16]. In our earlier implementation of a cooperative MAC layer (known as CoopMAC [10]) for unicast, we identified relays as 'helpers'. We choose to follow this nomenclature while describing the implementation of the cooperative multicast scheme.

Before going into the details of our implementation, we list below the assumptions we made during our experiments:

- We assume that the number of helpers and their MAC addresses are already known at the transmitter or the Access Point (AP). Hence, in the implementation we do not consider optimum helper selection.
- Due to our inability to access the MAC layer firmware in a real system (the higher MAC layer functionality is implemented in the driver while the lower level time sensitive functions are implemented in the wireless card), it was impossible for us to suppress the contention at the MAC layer and implement a fully operational TDMA scheme where the transmission of the source is followed by a sequential transmission of the helpers, one after the other. Therefore, we emulated a TDMA system by adding different delays before the transmission of each helper. This way, we give enough time to each helper to transmit its block of packets without contending with other stations. Therefore, although the MAC layer of the implementation is based on CSMA, in practice the helpers transmit in a sequential TDMA fashion, after the initial transmission of the AP.

A. Driver Implementation

We implemented the MAC layer using the open source driver Madwifi 0.9.2 [17] that works with Atheros chipsets [18]. The implementation details of each module are summarized as follows:

- 1) In each packet, we added a new header between the 802.11 header and the payload that we call CoopHeader. The CoopHeader consists of the following fields: *Destination Address*, *Source Address*, *Helpers Addresses* and *number of helpers*. Since we are broadcasting the data, the broadcast MAC address is the destination address. The AP defines the helpers for a particular broadcast, and adds their MAC addresses in the *Helpers Addresses* field.
- 2) Each station that receives a packet, checks whether it is selected by the AP as a potential helper. In order to do so, it checks the *Helpers Addresses* field in the CoopHeader. If one of the MAC addresses indicated in this field is equal to its address, it realizes that it is a helper and forwards the packet to the FEC module in the application layer which will be discussed further in Section III-B.
- 3) A Group 1 receiver only receives packets from the AP.
- 4) A Group 2 receiver only receives packets from its dedicated helper and discards all other packets from the source or other helpers.

B. Socket Programming

In order to implement video streaming, we built a video client/server application using UDP/IP socket programming along with FEC encoding/decoding.

- 1) In the transmitter, we run a server program that reads a FEC encoded, RTP packetized video file, and transmits the packets accordingly. Here, we use a block of 64 source packets and transmit a sufficient amount of FEC packets based on the channel quality of the transmitter-to-Group1 receivers link.
- 2) At the helpers, we run a program which receives packets and stores them in a file. Furthermore, we implemented a FEC module which buffers all the packets of the same block. If the total number of source packets and parity packets in a block is greater than 64, it uses the FEC packets and recovers the 64 source packets. For the second-hop transmission, we generate new parity packets based on the channel quality of the helper-to-Group2 receivers link and transmit them along with the source packets. If the total number of received source packets and parity packets in a block is smaller than 64, the module generates new parity packets only for the correctly received source packets and transmits them along with the source packets.

As mentioned in the beginning of this section, the forwarding of the block of packets by the helpers is done sequentially. Specifically, we implement the scheduling among the helpers by adding different delays before the transmission of each helper. The AP sends guard packets after the transmission of each block in order to inform the helpers that the transmission of one block is completed. Upon reception of the guard packets the first

helper, Helper 1, starts to transmit the block of packets immediately. On the other hand, Helper 2 waits for a fixed period of time which is equal to the time needed to transmit 64 source packets and m parity packets for a particular transmission rate. After this period, Helper 2 transmits its block of packets. Rest of the helpers continue the transmission in the same manner.

The helpers have the ability to forward all the received packets or to filter the transmission in the second hop by transmitting only a particular video layer. This can be defined before the experiment based on a GUI designed for this purpose. Using this GUI, we are able to choose packet transmission in one of the two modes:

- *Non-Layered Cooperative Multicast*: The helpers forward all the received packets.
- *Layered Cooperative Multicast*: The helpers check a video packet to see whether it belongs to the base layer or the enhancement layer. A packet is forwarded only if it belongs to the base layer. This mode results in differentiated video qualities among Group 1 and Group 2 receivers.

For the current implementation, we only consider two layers: base layer and enhancement layer. However, this procedure can be extended to support a higher number of layers.

- 3) At the receivers, we run a client program which receives packets and stores them in a file.

The stored files at the helpers and receivers are first FEC decoded to recover the video file. Then, we decode these files using a video decoder. In our experimental study, the applied parity packets are sometimes insufficient to correct all packet losses. In that case, only the correctly received video packets are fed into the video decoder. The decoder uses *frame copy* as the error concealment method to recover areas affected by lost packets. The video quality at each receiver for a particular experiment is determined by the average PSNRs of all the decoded frames of the video.

C. Layered Video Architecture

Our cooperative multicast framework can in principle work with other layered coding methods, but we choose to employ H.264/AVC [19] for our testbed implementation since H.264/SVC currently has no slice structure support which makes error handling difficult. In our experiments, we generate temporal scalable video bit streams with slice mode in order to create slices which are packetized into a packet. Specifically, we use H.264 Main Profile to encode a video sequence with the coding structure shown in Figure 3. The base layer (BL) consists of the slices of IDR (Instantaneous Decoder Refresh) type, P type and reference B type (Bs), and the enhancement layer (EL) consists of the slices of non-reference B types (B). The arrows in the figure indicate the reference dependencies during encoding, which forms a hierarchical motion prediction structure. From the figure, BL is independently decodable, while EL relies on BL for correct decoding. Note that a lost I or P picture from BL can affect all the following pictures in both BL and EL until the next IDR picture. A lost Bs picture in BL, although not affecting other pictures in the BL, affects decoding of the EL. On the other hand, the loss of any

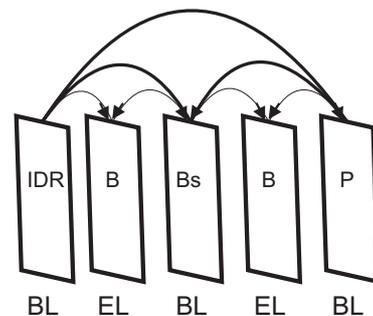


Fig. 3. The H.264 temporal scalability coding structure.

picture in the EL does not affect decoding of any other picture. For the encoding of the videos, we use the H.264 reference software JM11 and, for the decoding of the received streams, we modified the JM11 decoder so that it can support slice level errors.

The packet video streams in our experiments are created by encoding a video clip (Soccer, 352x288, 30fps, 240 frames). We used the slice mode to create slices of size 1470 Bytes or less and packetized each slice into an RTP packet.

IV. RESULTS

A. Experimental Setup

In our experiments, we use one transmitter, three helpers that are also Group 1 receivers, one Group 1 receiver that is not a helper and three Group 2 receivers. The experimental setup is depicted in Figure 4. All stations share channel 11 (2.462GHz) under IEEE 802.11b ad-hoc mode.

In [1], we determined the optimal operating system parameters in terms of r_1 , r_2 , R_1 , R_2 , N , t_1 , and t_2 , that maximizes different multicast performance criteria, while guaranteeing that the cooperative system can cover all users with the same range as the direct transmission. Our PER measurement study in [15] shows that direct transmission can cover up to 80 m with transmission rate of 1Mbps. To cover 80 m with two hop transmission, the best configuration is to set $r_1 = r_2 = 50m$, and $R_1 = R_2 = 11Mbps$, and $N = 5$. Based on this result, for the experiment, we place a Group 1 non-helper receiver and all three helpers at a distance of 50m from the access point, and the Group 2 receivers are 50m apart from the helpers. Although we need five helpers to cover the 80 meter radius, we only used three helpers in the testbed to illustrate the basic idea. Note that increasing the number of helpers will lower the video quality at each receiver as the transmission time available for the AP and each helper will be reduced.

We first ran experiments with the conventional multicast system. Then, we conducted two sets of experiments: non-layered cooperative multicast and layered cooperative multicast. In order to remove any random effect and short-term fluctuations, we ran each experiment with the same setting 10 times and averaged the results. The Group 2 results of cooperative multicast presented below are obtained by averaging the quality of all Group 2 receivers. Similarly, the Group 1 results are obtained by averaging all Group 1 receivers (including the three helpers). For the direct transmission case, we averaged the results of Group 2 receivers. Note that with this set up, the reported quality for each group in the cooperative system indicates the achievable average quality at farthest receiver in

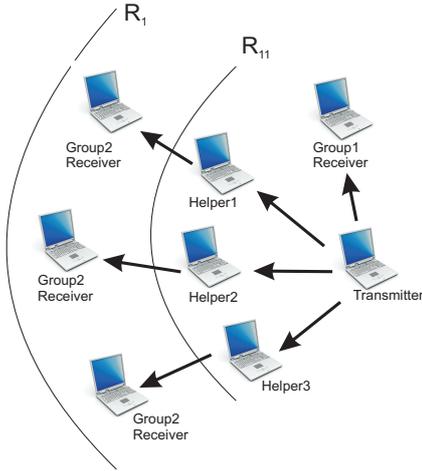


Fig. 4. Experimental setup.

each group, and like wise, the reported quality for the direct transmission system represents the achievable quality at the farthest receiver in its coverage area.

For a given transmission rate, we determine the FEC rate based on our PER measurement study (see Table II in [15]). Specifically, for transmitting at 11 Mbps to 50 m, the PER is around 20% and the added FEC overhead is 16 packets for 64 source packet, yielding $\gamma(R_1, r_1) = \gamma(R_2, r_2) = 0.8$. When transmitting at 1 Mbps to 80 m, the PER is around 25%, and the added FEC overhead is 22 packets for 64 source packets, yielding $\gamma(R_d, r_d) = 0.74$. Recall that β is defined as the effective data ratio which is the ratio of the time spent to transmit the actual payload data to the total transmission time which depends on the overhead due to packet headers as well as the actual data transmission time over the air time. For a fixed transmission rate, the overhead due to packet headers is fixed, however the actual data transmission time over the air time depends on the video rate. Note that as we increase the video rate, β also increases and β takes its maximum value when the overhead is only due to packet headers. In our experiments, we transmit a video at different video rates in order to observe both the channel effect and the congestion that is generated when the video rate is beyond the sustainable rate of the system.

B. Non-Layered Cooperative Multicast

Figure 5 compares the results obtained for direct transmission and non-layered cooperative multicast, both with and without FEC. For non-layered cooperative multicast, we want all users receiving the same video rate. This is achieved by assigning equal time slot for each transmission (sender-to-Group1 and each of the three relay transmission), so that $t_1 = t_2 = 1/4$. At a low video rate, most receivers can recover all lost packets. Therefore, the video quality initially increases as the video rate increases. However, as we increase the rate beyond a certain point, due to the contention in the channel, there is not enough time for the transmission of all the packets, hence, the decoded video quality starts to drop. The results show that the use of FEC significantly improves the video quality for both direct transmission and cooperative multicast. Note that, even at 1Mbps, there is significant packet loss at Group 2 receivers, and therefore direct transmission without FEC yields poor average quality.

Non-layered cooperative multicast with FEC can sustain up to 1.2Mbps video rate ($\gamma_{FEC} = 0.8, \beta = 0.545$). Above this rate, due to congestion, the video quality drops sharply. With direct transmission, we can only sustain a video rate of 0.5Mbps with FEC ($\gamma_{FEC} = 0.74, \beta = 0.67$). In theory, if we only consider the packet headers, β takes its maximum value which is approximately equal to 0.8. From the results above we observe that the values of β that we got in the experiments are close to the theoretical value in both the cooperative as well as the direct transmission scenarios. For the cooperative case, the value of β is lower than the direct transmission due to the fact that in the implementation we add some guard packets between sequential transmissions of different stations and therefore we slightly increase the overhead in the system.

Compared to 33.76 dB for Group 2 users with direct transmission, cooperative multicast system improves the maximum achievable average PSNR to 38.89dB and 37.31dB for Group 1 and Group 2 users, respectively. We observe that these PSNR numbers are very close to the encoding PSNRs at the respective video rates, suggesting that the applied FEC parity packets are able to correct almost all the lost packets at these video rates. Although these PSNRs are averaged over the farthest receivers in each transmission range, closer-by receivers would see similar quality as well. We also observe that Group 1 receivers have better quality than Group 2 receivers. This is due to the fact that if a helper does not receive all the packets in a block in the first hop, it cannot relay all the packets to its Group 2 receivers. Moreover, there may be some additional losses in the second hop transmission.

C. Layered Cooperative Multicast

For the layered cooperative multicast experiment, the sender transmits the base and the enhancement layer, and the helpers only forward the base layer. Therefore, while Group 1 receivers experience a full frame rate of 30fps, Group 2 receivers experience a frame rate of 15fps. In Figure 6, we present the layered cooperative multicast results and compare with direct transmission and non-layered cooperative multicast, all with FEC. The reported video rate for the layered cooperative multicast is the sum of base layer and enhancement layer rates and as the video rate increases, both the base and enhancement layer rate also increase. In our videos, the BL rate to EL rate ratio is approximately 4 (i.e. 80% of the overall video rate is BL and 20% of the overall rate is EL). When we use layered video, since the helpers do not need to forward all the packets, the sender has more time to transmit. Therefore, the video rate at the sender can be increased to yield a higher video quality for Group 1 receivers. Specifically, since the video rate of Group1 (BL+EL) is 5/4 of the rate of Group2 (BL only), the transmission time partition must satisfy $t_1 = 5t_2/4$. Given the constraint $t_1 + 3t_2 = 1$, we have $t_1 = 5/17$, and $t_2 = 4/17$.

The reported PSNR is the average over the PSNRs of decoded frames only. Note that, even though the PSNR of Group 2 receivers is only slightly lower than the PSNR of Group 1 receivers, Group 2 receivers experience a frame rate of 15fps rather than 30fps. Compared to non-layered cooperative multicast, layered system was able to increase the sustainable video rate from 1.2Mbps to 1.5Mbps with a corresponding gain of PSNR from 38.89 dB to 40.21 dB

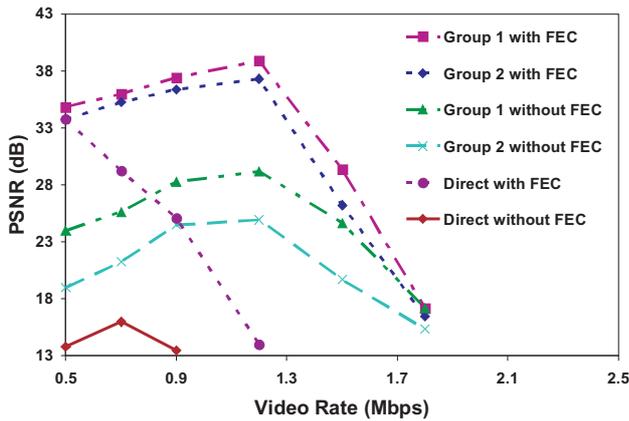


Fig. 5. Comparison of Non-Layered Cooperative Multicast with Direct Transmission with and without FEC.

at Group 1, while keeping the PSNR at Group 2 about the same, but at half of the frame rate (from 37.31dB at 30 fps to 36.80dB at 15 fps).

Based on the above results, we can conclude that:

- The use of FEC significantly improves the video quality for both direct transmission and cooperative multicast.
- The cooperative multicast protocol can provide significant performance improvement over the direct transmission system in realistic settings.
- Video layering can further improve the quality at Group 1 users at the cost of quality at Group 2 users. With temporal scalability adopted in our testbed, the perceptual quality of Group 2 is only slightly worse than Group 1.
- The proposed scheme is backward compatible with the IEEE802.11 technology and can be implemented at the MAC layer with no modification in the hardware.

V. CONCLUSION

In this paper, we implement a cooperative layered video multicast system at MAC layer and demonstrate the viability of the system in a real environment. The experimental measurements in a medium size (i.e., 8 stations) testbed are reported, which not only helps to develop a deeper understanding of the protocol behavior but also confirms that the cooperative MAC protocol for multicast delivers superior performance compared to the conventional approach. Moreover, using video layering can improve the quality at Group 1 users at the cost of quality at Group 2 users. The proposed scheme is backward compatible with 802.11 and easy to be implemented since it does not need any hardware modification. It can be implemented in software by changing the MAC and the layer above it.

As for possible future work, investigation of the performance in a larger testbed would be attempted. Moreover, we would investigate and develop algorithms that can dynamically select the helpers in a real environment. Another research direction is to study the cross layer MAC-PHY schemes where receivers will be able to combine the copies of the signal that they receive from multiple helpers.

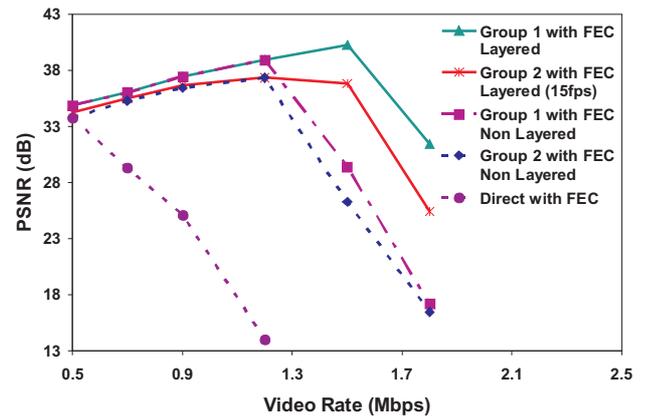


Fig. 6. Comparison of Layered Cooperative Multicast with Non-Layered Cooperative Multicast and Direct Transmission with FEC.

REFERENCES

- [1] O. Alay, T. Korakis, Y. Wang, E. Erkip, S. Panwar, "Layered Wireless Video Multicast using Omni-directional Relays", in *Proceedings of IEEE ICASSP*, 2008.
- [2] I. Bajic, "Efficient Error Control for Wireless Video Multicast," in *Proceedings of IEEE MMSP*, 2006.
- [3] J. Villalon, P. Cuenca, L. O. Barbosa, Y. Seok, and T. Turletti, "Cross-Layer Architecture for Adaptive Video Multicast Streaming Over Multirate Wireless LANs," *IEEE Transactions on Selected Areas in Communications*, vol. 25, no. 4, May 2007. pp 699-711.
- [4] C. Huang, J. H. David and C. Chang, "Congestion and Error Control for Layered Scalable Video Multicast over WiMAX," in *Proceedings of IEEE Mobile WiMAX Symposium*, 2007.
- [5] L. Lao, J. Cui, M. Y. Sanadidi and M. Gerla, "Scalable and Adaptive Multicast Video Streaming for Heterogeneous and Mobile Users," in *Proceedings of IEEE ISWCS*, 2005.
- [6] Z. Liu, Z. Wu, H. Liu, M. Wu and A. Stein, "A Layered Hybrid-ARQ Scheme for Scalable Video Multicast Over Wireless Networks," in *Proceedings of IEEE Asilomar*, 2007
- [7] S. Mao, X. Cheng, Y. T. Hou, and H. Sherali, "Multiple Tree Video Multicast Over Wireless Ad-Hoc Networks," in *Proceedings of IEEE BROADNETS*, 2004.
- [8] W. Wei and A. Zakhori, "Multiple Tree Video Multicast Over Wireless Ad-Hoc Networks," *IEEE Transactions on Circuits and Systems for Video Technology*, 17(1):215, January 2007.
- [9] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity-Part I and Part II," *IEEE Trans. Commun.*, vol. 51, pp. 1927-48, November 2003.
- [10] P. Liu, Z. Tao, Z. Lin, E. Erkip, and S. Panwar, "Cooperative wireless communications: A cross-layer approach," *IEEE Wireless Communications*, 13(4):8492, August 2006
- [11] T. Korakis, Z. Tao, S. Makda, B. Gitelman, and S. Panwar, "To Serve is to Receive Implications of Cooperation in a Real Environment," *Proceedings of Networking 2007*, 2007
- [12] IEEE 802.11, "Wireless LAN Medium Access Control (MAC) and Physical layer (PHY) Specifications," Standard, IEEE, Aug 1999
- [13] H. Liu, M. Wu, D. Li, S. Mathur, K. Ramaswamy, L. Han and D. Raychaudhuri, "A Staggered FEC System for Seamless Handoff in Wireless LANs: Implementation Experience and Experimental Study," in *Proceedings of IEEE International Symposium on Multimedia*, 2007.
- [14] T. A. Lee, S. G. Chan, Q. Zhang, W. Zhu, and Y. Zhang, "Allocation of Layer Bandwidths and FECs for Video Multicast Over Wired and Wireless Networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 12, December 2002. pp 1059-1070.
- [15] O. Alay, T. Korakis, Y. Wang, S. Panwar, "An Experimental Study of Packet Loss and Forward Error Correction in Video Multicast over IEEE 802.11b Network", in *Proceedings of IEEE CCNC*, 2009.
- [16] "Wireless Implementation Testbed Laboratory," <http://witestlab.poly.edu/wiki/>
- [17] "MadWifi: Linux kernel drivers for Wireless LAN devices," <http://madwifi.org/>
- [18] "Atheros: Chipsets for Wireless LAN devices," <http://www.atheros.com/>
- [19] ITU-T Recommendation H.264, "Advanced video coding for generic audiovisual services," 2003