# Dynamic Rate and FEC Adaptation for Video Multicast in Multi-rate Wireless Networks

Özgü Alay†, Caleb Li†, Anurag Rai‡, Thanasis Korakis†, Yao Wang†, and Shivendra Panwar†

†Dept. of Electrical and Computer Engineering, Polytechnic Institute of NYU, Brooklyn, NY
‡Department of Computer Science and Engineering, Indian Institute of Technology Kanpur, India

E-mails: ozgu@vision.poly.edu, cli20@students.poly.edu, anurai@iitk.ac.in, {korakis,yao}@poly.edu, panwar@catt.poly.edu

*Abstract*— **Video multicast over Wireless Local Area Networks (WLANs) faces many challenges due to varying channel conditions and limited bandwidth. A promising solution to this problem is the use of packet level Forward Error Correction (FEC) mechanisms. However, the adjustment of the FEC rate is not a trivial issue due to the dynamic wireless environment. This decision becomes more complicated if we consider the multi-rate capability of the existing wireless LAN technology. In this paper, we propose a novel method which dynamically adapts the transmission rate and FEC for video multicast over multirate wireless networks. In order to evaluate the system experimentally, we implemented a prototype using open source drivers and socket programming. Our experimental results show that the proposed system significantly improves the multicast system performance.**

**Index Terms**: forward error correction, rate adaptation, wireless networks, IEEE 802.11b, 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. Multicasting is an effective solution for simultaneous transmission of data to a group of users, since it saves network resources by spreading the same data stream across multiple receivers. However, the high packet loss ratio and bandwidth variations of wireless channels make video multicast over wireless networks a challenging problem.

Wireless networks possess several distinct characteristics that render the analysis, design and actual implementation methodologies developed for wired networks either invalid or inefficient. Specifically, wireless channels are error-prone and link quality in terms of error probability and available bandwidth is time-varying. Several methods have been proposed to cope with the above issues in a wireless environment. Two well known ones are Forward Error Correction and the Dynamic Rate Adaptation.

[1]This 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 the Polytechnic Institute of NYU.

In a wireless network, as the external environment changes, the channel error rate varies, resulting in deleterious effects on multimedia transmission. In order to cope with errors and hence have robust video transmission, we need accurate channel-condition estimation and an effective error control mechanism. Furthermore, due to bursty and location dependent errors, 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 number of retransmissions. A promising solution for error control in video multicast over wireless networks is the use of forward error correction (FEC) to handle losses [1]-[4]. In such a system, block erasure codes are used to correct errors using redundant information in the data stream. For example in an $(n, k)$ block erasure code, there are a total of $n$ packets where $k$ of them are source packets and $(n-k)$ of them are redundant parity packets. The parity packets are generated in such a way that any $k$ of the $n$ encoded packets are sufficient to reconstruct the $k$ source packets. The advantage of using block erasure codes for multicasting is that a single parity packet can be used to correct independent single-packet losses among different receivers.

IEEE 802.11 wireless LANs (WLANs) are one of the fastest growing network technologies in the wireless communications field. The IEEE 802.11b Media Access Control (MAC) protocol provides a multirate-capable physical-layer [5]. The dynamic rate adaptation feature of IEEE 802.11 lets the transmitter adapt the transmission rate based on the wireless channel operating conditions which may vary with time. Some proposals for unicast communications have been reported in the literature to adapt the PHY data rate according to the channel conditions, such as ARF (Auto Rate Fallback) [6] or RBAR (Receiver Based Auto Rate) mechanisms [7]. However, multicast/broadcast packets are always transmitted using the base transmission rate of the system (e.g., 1Mbps for IEEE 802.11b). The intention of such a conservative approach is to minimize losses at the stations that are located far away from the transmitter, so that they are able to successfully receive and decode the packet.

The above rate adaptation mechanisms rely on the estimation of individual channel states, and therefore they can not be directly applied to multicast service. The difficulty comes from the fact that the channel conditions between the Access Point (AP) and each receiver in the multicast group may differ, and in the absence of feedback the AP does not have any means to get to know the operating conditions of each individual receiver. In [8], a rate adaptive multicast scheme has been proposed where the transmitter must first send an RTS frame to indicate the beginning of a multicast transmission. This RTS frame is used by all the multicast receivers to measure the Receiver Signal Strength (RSS). Then, each multicast receiver has to send a variable length dummy CTS frame with a length that depends on the selected PHY transmission mode. Finally, the transmitter senses the channel to measure the collision duration and can adapt the PHY rate transmission of the multicast data frame accordingly. In [9], SNR-based Auto Rate for Multicast (SARM) is proposed for multimedia streaming applications. In SARM, multicast receivers measure the SNR of periodically broadcasted beacon frames and transmit this information back to the AP. To minimize feedback collision, the backoff time for the transmission of this feedback from each station, increases linearly with the received SNR value. Then, the AP selects the lowest received SNR to adapt the PHY rate transmission. Recently, Villalon et al. [10] studied a cross-layer approach for adaptive video multicast considering the multi-rate capabilities of wireless networks. Their architecture requires knowing the operating conditions of the channel as perceived by the multicast members. The PHY data rate to be used for the multicast traffic is determined based on the feedback received by the AP. They also proposed to use an H.264 hierarchical video coder in order to adapt the video transmission to the channel conditions perceived by the multicast receivers. Although the above algorithms seem to improve the performance of multicast in wireless networks by taking advantage of multi-rate capability, they do not make use of any error recovery mechanism. An exception is the mechanism in [10] where the authors only employ data retransmission. However, such a mechanism is not a good fit for multicast. Furthermore, although these studies have shown the efficiency of rate adaptive multicast schemes via simulations, they do not provide insights on the way rate adaptation should be applied for multicast in a real wireless network.

In this paper, we propose a method which dynamically adapts the transmission rate and FEC in a multicast wireless network. The basic idea behind this method is to use the highest sustainable transmission rate together with sufficient amount of FEC in order to maximize the video quality of multicast receivers. In order to validate the proposed scheme in a real environment, we implemented a prototype and carried out extensive experiments in a mid-size testbed. The proposed system was implemented in the MAC layer as well as in the application layer, using open source drivers and socket programming, along with a packet level FEC implementation. The experimental results show that the new mechanism works efficiently in a real environment and significantly improves the performance of the network.

The paper is organized as follows. In Section II, packet level FEC is discussed along with the rate adaptation mechanism. The implementation effort is elaborated in Section III. Section IV reports and analyzes the obtained results. We conclude the paper in Section V.

## II. Joint Transmission Rate and FEC Adaptation

### A. Motivation

Although rate adaptation is a standard feature in today's wireless networks, multicast/broadcast packets are transmitted using the base transmission rate of the system (e.g., 1Mbps for IEEE 802.11b). The motivation for using the lowest transmission rate is to enable far away receivers to successfully receive and decode the transmitted packets. In multicast, one cannot rely on retransmission to correct lost packets. Allowing the multicast server to retransmit lost packets to all receivers would dramatically increase the overhead on the network, since each receiver may ask for retransmission of different packets (due to the independent errors at different receivers). Without any other error control mechanism, the server has to transmit at the lowest possible rate in order to accommodate users with poor channel conditions.

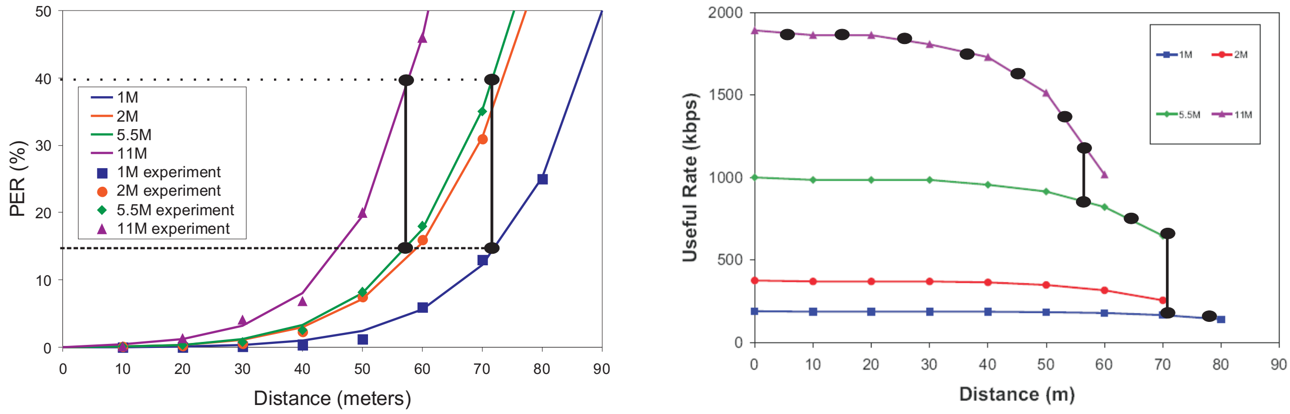
Forward error correction (FEC) at the application layer is a promising alternative for handling losses in multicast services. The basic idea of FEC is that redundant information is sent a-priori by the source station, in order to be used by the receivers to correct errors/losses without contacting the source. Since CRC-based error detection at the link layer results in the removal of the corrupted packets, many FEC-based protocols try to recover these packets [13]. However, such a scheme introduces overhead since extra parity packets are now transmitted by the source station. The overhead introduced is the number of parity packets to be sent for $k$ source packets. The number of parity packets, $m$, can be determined as follows:

$$m = kP_E/(1 - P_E) \tag{1}$$

where $P_E$ is the Packet Error Rate (PER). Note that, the level of the overhead depends on the packet error rate, $P_E$, in the network. Thus, the higher is the packet error rate, the more the parity packets that must be transmitted by the server, and therefore the more the overhead and the less the FEC rate, $\gamma_{FEC}$, which is the ratio of source packets to the total number of packets, $\frac{k}{k+m} = 1 - P_E$.

From the above discussion, we conclude that it is important to have an accurate estimation of the packet error rate, in order to apply no more FEC parity packets than needed. In wireless networks, we know that different transmission rates, $R_{PHY}$, give different PER. Furthermore, due to path loss and fading, for a fixed transmission rate, we observe different PER values at different locations, $l$. Hence, for a fixed location and physical transmission rate, the FEC rate, $\gamma_{FEC}$, can be formulated as follows

$$\gamma_{FEC}(l, R_{PHY}) = 1 - P_E(l, R_{PHY}) \tag{2}$$




(a) Rate Transition Thresholds

(b) Useful Rate Transitions

Fig. 1. Packet Error Rate and Useful Rate vs Distance

Note that the FEC overhead is not the only overhead in the system. In order to consider the other overheads (e.g., headers, etc.), we define the effective data ratio, $\beta$, as the ratio of the time spent to transmit the actual payload data to the total transmission time.

Based on the discussion above, the useful rate, $R_{useful}$, can be computed as follows,

$$R_{useful} = \gamma_{FEC}(l, R_{PHY})\ \beta(R_{PHY})\ R_{PHY} \tag{3}$$

In the above formulation, it is not clear how someone should define the combination of transmission rate and FEC rate in order to increase the efficiency of the network. On one hand, the higher the transmission rate is, the higher the PER and therefore the more FEC parity packets should be transmitted. On the other hand, as the transmission rate increases, the more efficient the use of the medium becomes, allowing more room for extra FEC parity packets. Therefore, while designing a multicast system, we should consider the adjustment of the transmission rate and the FEC rate jointly.

### B. Proposed System

In this paper, we design a joint transmission rate and FEC adaptation algorithm to maximize the multicast performance. The basic idea behind the proposed system is that all the multicast receivers periodically send PER information to the AP. Based on the received PER information, the AP identifies the worst channel and adjusts the transmission rate and the FEC based on this PER. For a fixed transmission rate, the adjustment of the FEC rate is not a trivial issue due to the dynamic wireless environment. This decision becomes more complicated if we consider the multi-rate capability of existing wireless LAN technology. Note that for a fixed distance, the PER is different for different transmission rates. Hence, while switching from one transmission rate to another, we should adjust the amount of FEC to be applied. However, when we consider the joint design of transmission rate and FEC, we should define the PER thresholds that will lead to a change of transmission rate and the corresponding FEC that is needed, as well as the FEC to be applied while the system stays at the same transmission rate for different PER values.

In order to determine these parameters, we used the experimental data in [15]. In Figure 1(a), we present the PER curves for an IEEE 802.11b system at different distances in an outdoor environment for different transmission rates. Figure 1(b) illustrates the corresponding useful rates. Note that using a higher transmission rate together with stronger FEC is more efficient than using a lower transmission rate with weaker FEC for video multicast. Following this rule, in the new scheme we focus on the highest transmission rate balancing the packet losses by applying the needed FEC. Note that 2Mbps and 5.5Mbps have similar PER curves, hence instead of using 2Mbps, we directly use 5.5Mbps in our rate adaptation algorithm. In these figures, at 70m, we can sustain transmission at 5.5Mbps with 40% PER or at 1Mpbs with a PER rate of 15%. In this case, instead of transmitting at 1Mbps (which has a useful rate of 167kbps), we choose to transmit at 5.5Mbps applying additional FEC (with a useful rate of 647kbps). Similarly, at 60m, we can sustain a transmission at 11Mbps, at 5.5Mbps and at 1Mbps, but we choose to transmit at 11Mbps. Note that by doing these transitions, for a given location, rather than operating at the direct transmission rate of 1Mbps, we operate at the highest sustainable transmission rate. With this design, we are able to operate at the highest possible useful rates at all distances, as indicated in Figure 1(b).

There are several rate adaptation approaches in the literature for the unicast transmission. Although these approaches can be applied to our proposed system, in this paper we choose to search through different rates, starting from the base rate, to find the optimal solution. In our system, in order to guarantee that we cover all the multicast clients, we start with the 1Mbps base rate of IEEE 802.11b, and based on the PER information received from the clients, we adapt the transmission rate along

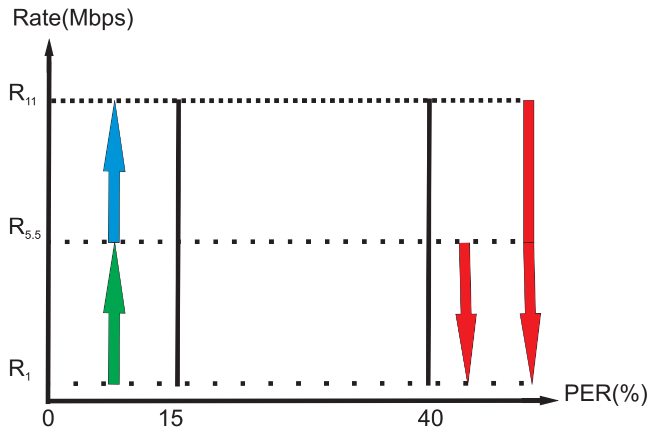



Fig. 2. Transmission Rate and FEC adaptation Algorithm

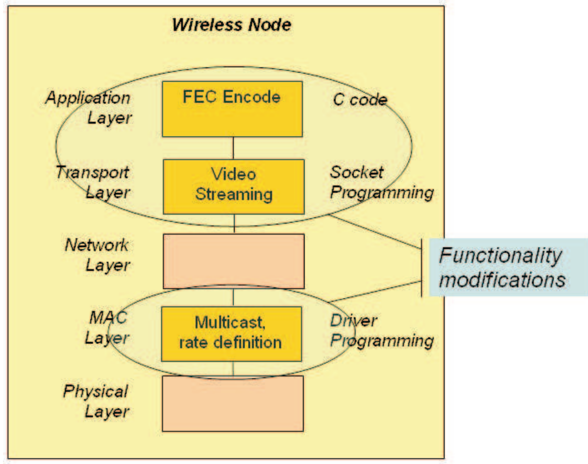



Fig. 3. Node Architecture

with the FEC. The rate and FEC adaptation algorithm has two components:

1) Switch the Transmission Rate: In Figure 2, we illustrate the transmission rate adaptation algorithm used in our system. In the figure the arrows indicate the change in the transmission rate. When the PER is between 0 and 15%, we switch to a higher transmission rate. Note that when we switch to a higher transmission rate, since the PER in the previous transmission rate is between 0 and 15%, the PER in the current transmission rate will be less than 40% (see Figure 1(a)). Hence, while switching to a higher transmission rate, we consider the worst case scenario and assume the PER is 40%. At any transmission rate, if the PER received is higher than 40%, we assume that we can not sustain the current transmission rate and we directly switch to the base rate where we assume the PER is 40%. Here, in order to handle small fluctuations in the channel, we sent 20% more FEC than that dictated by (1). Therefore, during rate transmission switch, we send 48% FEC. For example, if we currently transmit at 1Mbps and the PER is between 0 and 15%, we jump to a higher transmission rate, 5.5Mbps and send 48% FEC. If at 5.5Mbps, the PER is still small (e.g. 0-15%), we switch to a higher rate and start to transmit at 11Mbps with 48% FEC. Finally, if the PER received is higher than 40%, we directly switch to the base rate, 1Mbps with 48% FEC. For 11Mbps, if the PER is slightly higher than 40%, we can switch to the next lower transmission rate of 5.5Mbps. However, in this paper, we choose to switch to the base rate since PER may be much higher than 40%. Note that for this case we can not sustain 5.5Mbps.
2) Keep the Same Transmission Rate: Note that in the figure, there are regions where we do not switch the transmission rate. We determine these regions using our experimental results illustrated in Figure 1(a). In this figure, note that if the PER at 1Mbps is more than 15%, it is not worth to jump to a higher transmission rate since we will observe more than 40% PER at 5.5Mbps.

Note that for different transmission rates and FEC, we have different useful rates. Therefore, while switching between the transmission rates, we also switch to the corresponding video rate. Furthermore, when we keep the same transmission rate, we still adjust the FEC rate based on the PER which leads to different video rates. We will discuss the different video rates used in our experiments in detail in Section IV-A.

## III. Implementation Efforts

In order to gain more insights into the system performance of the proposed system in a real environment, we implement the joint rate and FEC adaptation algorithm on our experimental testbed [16]. In our implementation, we changed the basic functionality of a wireless node in three different layers of the protocol stack: the MAC, the transport and the application layer. The block diagram of the system we designed and implemented is depicted in Figure 3. The main features of the system are as follows:

### A. MAC layer

For the implementation of the MAC layer we used open source drivers in a Linux platform. In particular we used the MadWifi driver [17] for the Atheros chip-set [18]. We used the driver in broadcast mode (i.e. no acknowledgment, no retransmissions). Additionally, we added a new feature in the driver that allowed us to choose the transmission rate that we would use. In our experiments, we set up the wireless cards to work in the 802.11b mode, and therefore we had to choose between four different rates: 1Mbps, 2Mbps, 5.5Mbps, 11Mbps.

### B. Transport layer

In order to implement the video streaming service we built a video client/server application using UDP/IP socket programming. The server is a program that can read a FEC encoded video file, packetize accordingly and transmit. Similarly, the program in the client side receives packets, does the FEC decoding and stores the resulting video into a file.

### C. Application layer

In the application layer we have two main components: Packet Level FEC encoding and decoding, and feedback generation.

*1) Packet Level FEC:* In the application layer we implement a packet level FEC mechanism. We utilize Reed Solomon (RS) codes since it is one of the well-known block codes with good error correction properties and is widely used in FEC schemes. In general an $(n, k)$ RS code contains $k$ source packets and $(n - k)$ parity packets. Altogether, they form a group of $n$ packets, in a way that any $k$ of the $n$ packets can be used to reconstruct the $k$ source packets [14]. In this work, we use systematic $(n, k)$ codes, where the first $k$ of the $n$ encoded packets are identical to the $k$ source packets.

In the current implementation, the generation of the parity packets is done offline. We first encode the video with a GOP (Group of Picture) size of 16 frames. Note that a GOP always starts with an Intra frame, hence, each GOP is independently decodable. In our experiments we utilize RS(n,k) codes where we generate parity packets based on the number of the source packets in a GOP. In other words, the number of source packets in each block depends on the number of packets in each GOP. While generating the parity packets, we make sure that we generate enough parity packets for the worst channel conditions. We store these files in the hard drive of the node in order to use them as inputs on the video streaming server. Note that it would be a waste of bandwidth to send all the parity packets generated since the channel condition is not always bad. Therefore, the number of parity packets to be sent, $m$, is chosen based on the current channel conditions (i.e. packet error rate $P_E$) as formulated in (1).

Upon reception of the packets, the receiver decodes the FEC encoded packets, generates the video file and stores it in the node. As long as the number of lost packets is less than $m$, all original video packets can be decoded successfully. When the loss exceeds the FEC correction limit, only the received video packets are saved into the decoded video files. Finally the quality of the video is obtained using the Peak Signal to Noise Ratio (PSNR) of the received video.

*2) Feedback Generation:* Upon reception of the blocks, the clients are responsible for computing the PER and sending this information to the access point. The PER is computed and sent once for every two blocks. In each block, we extract how many source packets there are in a block using the parity packets. The clients count the number of source packets received and compute the PER, which is the ratio of missing packets to all source packets. The access point receives the PER from multiple clients and finds the maximum PER. Based on this maximum PER, the access point adjusts its transmission rate and FEC rate. In the current implementation, we assume that the channel conditions do not vary too fast. Hence, the number of parity packets to be sent is calculated using the latest maximum PER. Note that accurate PER estimation is very important, as it dictates the number of parity packets to be sent.

| *Transmission Rate* | *PER Range* (%) | *Video Rate* (*Mbps*) |
|---|---|---|
| 1Mbps | 0-25 | 0.13 |
| | 25-40 | 0.10 |
| 5.5Mbps | 0-25 | 0.70 |
| | 25-40 | 0.52 |
| 11Mbps | 0-20 | 1.44 |
| | 20-40 | 0.98 |

TABLE I

VIDEO RATE ADAPTATION BASED ON THE PER FOR DIFFERENT TRANSMISSION RATES

## IV. RESULTS

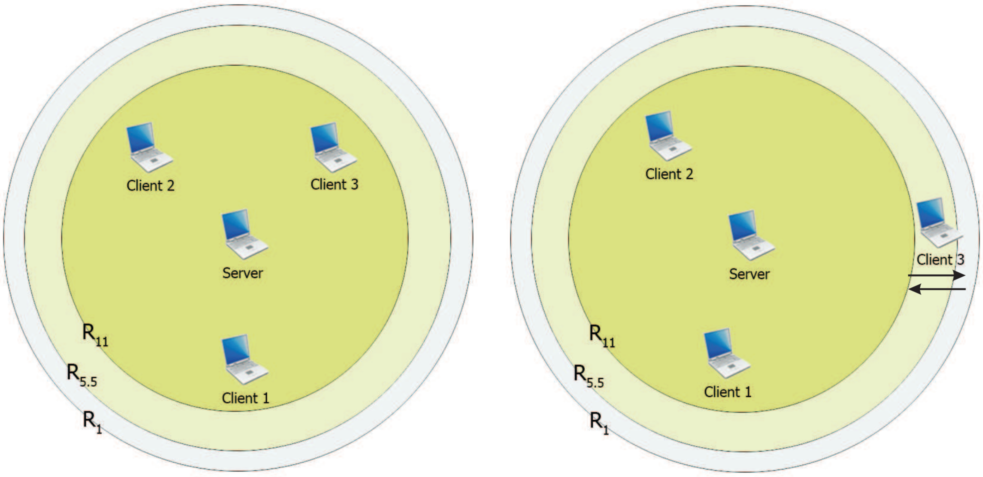
In our experimental study we use an IEEE 802.11b based WLAN. In order to understand the behavior of such a network we conducted experiments using broadcast modes in an outdoor environment. The experiments were conducted in Columbus Park, Brooklyn. During our experiments, we are mainly interested in the packet losses due to channel conditions rather than the traffic contention in the channel. Hence, in our experiments, we transmit for only 20% of the time (airtime), in order to keep the traffic level low. For instance, at 1Mbps transmission rate, we transmit 200kbits in one second. This includes the transmission of the source packets along with the parity packets with their corresponding headers.

### A. Video Rate Adaptation

The proposed rate and FEC adaptation scheme is developed assuming the video rate can be adapted. Note that since we have different transmission rates, we need videos at different rates. Furthermore, for each transmission rate, due to different PER, we need videos at different rates. In practice, this can be implemented in two ways. The first, more desirable approach is to encode a video into a scalable stream, which can be truncated to match the useful rates corresponding to different transmission rates and FEC rates. The second option is to pre-encode the same video into multiple versions with different bit rates, corresponding to the achievable useful rates under different transmission rates and FEC rates. When the system switches the transmission rate and the FEC rate, it also switches to a different bit stream, whose rate is below the available useful rate.

Although one can use the recently developed H.264/SVC standard [19] and its reference software to create a scalable bit stream, lack of a real-time SVC decoder that can handle packet losses, prevented us from using the first option. Instead, our current implementation adopted the second approach. To enable bit-stream switching, we code the video into independent GOPs, and switch only at the beginning of each GOP. Recall that a GOP always starts with an Intra frame, hence, each GOP is independently decodable.

Instead of creating many versions of the same video corresponding to all possible transmission rates and FEC rates, to simplify the system implementation, we create only two video streams for each transmission rate, targeted for two different regions of the PER. Note that we only use an airtime of 20%




(a) Scenario-I (b) Scenario-II

Fig. 4. Different Experimental Scenarios

of the transmission rate which includes packet headers, video and FEC parity packets. For example, for 11Mbps transmission rate, considering the airtime and packet headers, the effective data ratio is $\beta$=0.17 (see [15]). This corresponds to a rate of 11x0.17=1.89 Mbps for video and FEC transmission. For the high PER region, between 20-40%, we choose the video rate based on the maximum PER, which is 40%. Recall that we apply 20% more FEC protection than the predicted PER, so that the total FEC overhead is 48%, or the FEC rate is 52%. This leads to a useful video rate of 1.89x0.52=0.98 Mbps. For the low PER region, between 0-20%, we choose the video rate based on 20% PER. This leads to a video rate of 1.89x(1-0.2x1.2)= 1.44 Mbps. Table I summarizes the actual video rates adopted for different transmission rates and PER regions.

The packet video streams in our experiments are created by encoding a video clip (Soccer, 352x288, 30fps, 1200 frames) using the H.264/AVC [20] Main Profile with a GOP size of 16 in IBPBP mode. We used the slice mode to create slices of size 1470 Bytes or less and packetized each slice into an RTP packet. The packets in a GOP forms a block and since the PER feedback is generated for every 2 blocks, the feedback information is transmitted approximately once every second. 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 decoder uses *frame copy* as the error concealment method to recover areas affected by lost packets.

### B. Testbed Configuration

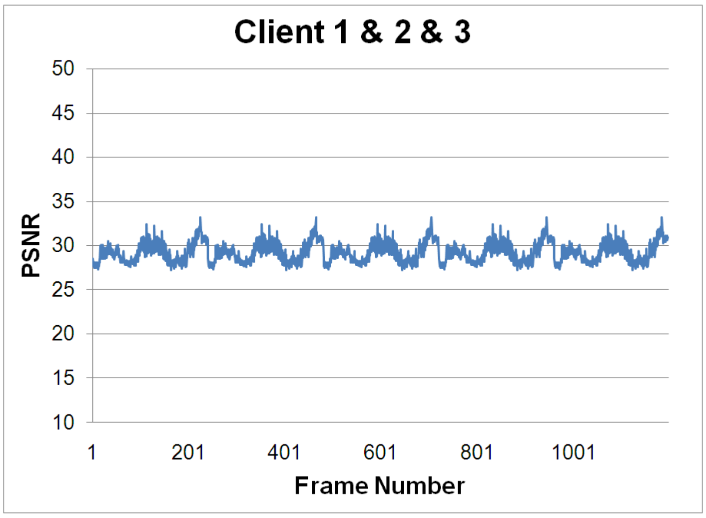
The testbed used in the experiments consists of 4 Linux laptops with 802.11 wireless cards based on the Atheros chipset. The stations share channel 11 of the 2.4 GHz band. In this experimental study, one station is used as a server and the other three stations are the multicast receivers (clients). The server and the receivers are within line of sight.

In order to show the effectiveness of the proposed algorithm, we consider two different scenarios.

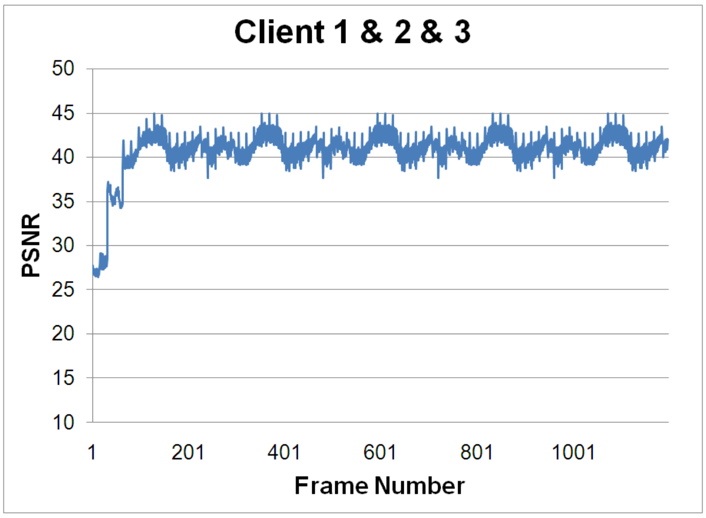
1) Scenario-I: In the first scenario, we assume the server and the clients are in close proximity, as illustrated in Figure 4(a).
2) Scenario-II: Server, Client 1 and Client 2 are in close proximity with each other while Client 3 is a mobile user moving around, as illustrated in Figure 4(b).

We first ran the experiment in Scenario-I and compare the video quality of the proposed system with this in direct transmission, as illustrated in Figure 5. In direct transmission, we assume that the direct transmission operates in the low PER region of 1Mbps (i.e. video rate= 0.13Mbps). Even though all clients are close to the access point, since the physical transmission rate is 1Mbps, the video quality is low, as illustrated in Figure 5(a). When the proposed algorithm is used, the PER is small since the clients are close to the server. Hence, the video server starts transmitting using 1Mbps and gradually increases the transmission rate, reaching 11Mbps. Compared to direct transmission, the proposed algorithm significantly improves the video quality for all the users, as illustrated in Figure 5(b). Note that this scenario shows the maximum achievable quality improvement, compared to direct transmission. The proposed rate and FEC adaptation system improves the average PSNR to 40.68 dB for all users, compared to 29.23 dB for all users with direct transmission.

In the second experiment we consider Scenario-II and compare the video quality of the proposed system with direct transmission, as illustrated in Figure 6-7. With direct transmission, we assume that the direct transmission operates in the high PER region of 1Mbps (i.e. video rate= 0.10Mbps). The video quality in the case of direct transmission is illustrated in Figure 6. Note that, in this scenario the FEC rate is fixed and chosen based on Client 3. Hence, Client 1 and Client 2 receive more parity packets than necessary and are able to decode all the blocks correctly. However, since the transmission rate is 1Mbps, the video quality is still low as illustrated in Figure

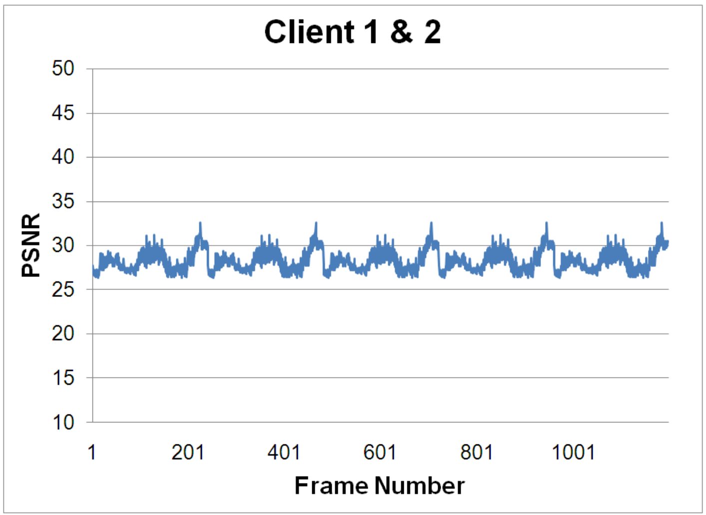



(a) Direct Transmission




(b) Proposed Rate-FEC Adaptation

Fig. 5. PSNR vs Frame Number for Experimental Scenario-I (Average PSNR for direct transmission and the proposed algorithm are 29.23dB and 40.68dB, respectively)




(a) Client 1 and 2

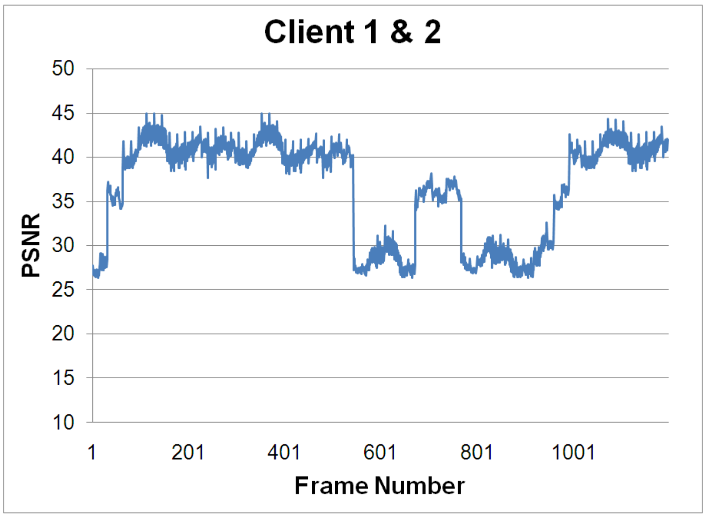

(b) Client 3

Fig. 6. PSNR vs Frame Number with Direct Transmission for Experimental Scenario-II (Avereage PSNR of Client 1 & 2 and Client 3 are 28.31dB and 27.88dB, respectively)

6(a). On the other hand, due to the fluctuations in the channel, the applied FEC is occasionally insufficient for Client 3. Hence, there are some blocks which can not be decoded as it is depicted in 6(b). The proposed rate-FEC adaptation system adjusts its parameters based on the worst PER. Whenever the PER is small, the algorithm switches to a higher transmission rate and whenever Client 3 moves away from the server and experiences a higher PER, the proposed system switches to a lower transmission rate (Figure 7). Client 1 and Client 2 are able to decode all the blocks as presented in Figure 7(a). Since Client 3 is mobile, it occasionally experiences higher PER than the system predicts. This leads to unrecoverable packets and a sharp drop in the PSNR of Client 3 as depicted in 7(b). Compared to direct transmission, the proposed algorithm provides equal or better quality to all users. The proposed algorithm improves the average PSNR to 36.59 dB compared to 28.31 dB with direct transmission for Client 1 & 2. For Client 3, the proposed algorithm improves the average PSNR to 34.97 dB compared to 27.88 dB with direct transmission.

## V. Conclusion

In this paper, we propose a novel method which dynamically adapts the transmission rate and FEC for video multicast over multirate wireless networks. In order to evaluate the system experimentally, we implemented a prototype using open source drivers and socket programming. Our experimental results show that the proposed system significantly improves the multicast system performance. The promising experimental




(a) Client 1 and 2


(b) Client 3

Fig. 7. PSNR vs Frame Number with Proposed Rate-FEC Adaptation for Experimental Scenario-II (Average PSNR of Client 1 & 2 and Client 3 are 36.59dB and 34.97dB, respectively)

results show that the joint consideration of transmission rate adaptation and FEC adjustment may consist a new paradigm for real time multicast transmission over wireless networks.

In our experiments, we use a testbed of medium size (4 nodes). Ongoing research considers testing the proposed system in a testbed with more nodes.

In this paper, we only use the PER information for the previous block in order to predict the PER for the upcoming block. A future direction is to predict the PER more accurately by incorporating the history of reported PERs. Finally, in our future work we are planning to adapt the proposed mechanism in the environment of IEEE 802.11a/g where a larger set of transmission rates is available and therefore a more dynamic scheme of joint rate and FEC adaptation is required.

## References


[1] A. Basalamah , H. Sugimoto and T. Sato, "Rate adaptive reliable multicast MAC protocol for WLANs," in *Proceedings of IEEE VTC*, 2006.

[2] 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.

[3] I. Bajic, "Efficient Error Control for Wireless Video Multicast," in *Proceedings of IEEE MMSP*, 2006.

[4] 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.

[5] LAN MAN Standards Committee of the IEEE Computer Society, ANSI/IEEE Std 802.11, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-speed Physical Layer Extension in the 2.4 GHz Band," IEEE 802.11 Standard, 1999.

[6] A. Kamerman and L. Monteban, "WaveLAN II: A high-performance wireless LAN for the unlicensed band," Bell Labs Technical Journal, pp. 118133, 1997.

[7] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," in *Proceedings of ACM MOBICOM*, July 2001.

[8] A. Basalamah, H. Sugimoto, and T. Sato, "Rate adaptive reliable multicast MAC protocol for WLANs," in *Proceedings of IEEE VTC*, May 2006.

[9] Y. Park, Y. Seok, N. Choi, Y. Choi, and J. M. Bonnin, "Rate-adaptive multimedia multicasting over IEEE 802.11 wireless LANs," in *Proceedings of IEEE CCNC*, Jan. 2006.

[10] 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.

[11] P. K. McKinley and A. P. Mani, "An experimental study of adaptive forward error correction for wireless collaborative computing", in *Proceedings of IEEE SAINT*, 2001.

[12] P. K. McKinley and A. P. Mani, "Experimental Evaluation of Error Control for Video Multicast", in *Proceedings of IEEE DCSW*, 2001.

[13] L. Rizzo, "Effective erasure codes for reliable computer communication protocols", in *ACM Computer Communication Review*, April 1997.

[14] A. J. McAuley, "Reliable broadband communications using burst erasure correcting code", in *Proceedings of ACM SIGCOMM*, September 1990.

[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] Joint Scalable Video Model (JSVM), JSVM Software, Joint Video Team, Doc. JVT-X203, Geneva, Switzerland, June 2007

[20] ITU-T Recommendation H.264, "Advanced video coding for generic audiovisual services," 2003