Broadcasting Video Over 802.11g Networks Using Application FEC

Jim Martin, James Westall, Rahul Amin
Clemson University
Clemson, SC, 29634, USA
Corresponding author: jim.martin@cs.clemson.edu

***Abstract*- As broadband wireless networks become further adopted by society, scalability issues are sure to arise. One type of scenario that is particularly challenging is an event that involves large groups of people located in a relatively small area. We refer to these environments as *'crowd spots'*. Our motivating example is a sports and entertainment venue where video streaming is likely to be the dominant application. The objective of the research presented in this paper is to explore the performance of multicast video over 802.11 using APFEC when subject to potentially high levels of correlated loss. Our results confirm that in certain conditions, APFEC is highly effective. As an example, assume a wireless device that receives a video stream observes an average loss rate of 0.14 and that APFEC is configured with a code rate of 0.80 and a block size of 640 packets. Our simulation-based results suggest that APFEC can eliminate between 40% to 82% of loss. If the block size is increased to 10,240 packets, the effectiveness of APFEC increases to between 0.80 and 1.0. However, increasing the block size from 640 to 10,240 packets causes the packet latency to increase from 4 to 60 seconds and the channel zapping times to increase from 2 to 400 seconds.**

I. INTRODUCTION

As broadband wireless networks become further adopted by society, scalability issues are sure to arise. Today's 3G and emerging 4G cellular-based systems are engineered to provide sufficient service levels that meet the needs of subscribers while conforming to economic and spectral constraints. As demand for resources grows, cellular providers must either increase spectrum efficiency, increase raw spectrum, or modify how users share existing spectrum. Venues involving large gatherings of people, such as conference or entertainment events, are particularly challenging. We refer to these environments as *'crowd spots'*. WiFi 'hot spots' are small areas of 802.11 wireless coverage that can provide Internet access. A crowd spot involves hundreds, thousands or tens of thousands of people temporarily grouped together in dense formations. A rapidly growing percentage of participants in crowd spots will have multimodal smartphone devices and will expect wireless connectivity. To meet broadband wireless subscriber's insatiable appetite for bandwidth, 3G/4G operators will need to offload traffic from their broadband spectrum to open WiFi networks. This is motivating the deployment of WiFi infrastructure at sports and entertainment locations. WiFi offers stadium and event operators a medium to further engage spectators, one that can potentially lead to new revenue generating services.

The challenges surrounding dense 802.11 deployments that serve large public gatherings have been identified in the literature [1-7]. Recent incidents highlight the fact that WiFi crowd spots can be unreliable[1]. In prior work we have assessed the impacts of 'crowd spots' on 802.11 performance in a stadium environment [8]. We found that when significant events occur in the game that cause crowd reactions (e.g., crowds standing up and cheering after a play) the signal strength of a handheld device can drop by 25 dB for time periods that extend beyond 20 seconds. During these episodes we observed a drop in system throughput (over one 802.11g channel) of over 20 Mbps. We conjecture that the performance drop is due to propagation impairment caused by the synchronized movement of a large number of people and also by a spike in demand for bandwidth as spectators attempt to utilize wireless systems (e.g., to watch video replays provided by the venue). It is apparent that one attribute of crowd spots will be potentially high levels of correlated packet loss.

In sports and entertainment crowd spots, it is likely that the applications of interest involve streamed video (either replays or multiple video camera feeds). We envision that multiple video 'channels' will be available, each carrying different views of the event from different cameras. In a manner similar to how some sporting events are broadcast on cable or satellite networks, a spectator can 'tune' to different channels to watch venue-specific content. The streaming model might be on-demand where video replays are made available or linear where the content is a continuous video stream from a particular camera. Our research focuses on linear streaming

This work was funded in part by Cisco Research and by NSF grant ECCS-0948132.

[1] A recent example is the failed demo performed by Steve Jobs at the 2010 Apple Worldwide Developers Conference (WWDC2010).

where multicast is required for scalability and APFEC is required to mitigate packet loss.

The objective of the research presented in this paper is to explore the performance of multicast video over 802.11 using APFEC when subject to controlled levels of correlated loss. Of particular interest is to obtain a set of baseline results that provides guidance as to the required level of APFEC redundancy and the required block size that are necessary to mitigate a realistic loss process over an 802.11 network. The remainder of the paper is organized as follows. Section II provides background information on APFEC and 802.11 multicast. Section III describes the experimental methodology that is used in our study. Section IV presents and analyzes the results of our study. The paper is concluded in Section V.

II. Background

Prior work has identified the challenges and limitations of large-scale 802.11 networks [1-6]. The work in [1] suggests that at an academic conference, retransmissions accounted for 46% of all data transmissions, the client spent much of its time switching between modulation and coding modes, and the 802.11 overhead was exceedingly high (only 40% of all transmission time was spent sending original data packets). The work in [2] suggests unnecessary handoffs are common, even when users are not mobile. Several studies suggest that higher transmission rates are better than lower, more robust modulation and coding levels, simply because the network wide advantages are significant if one compares a particular wireless transmission at 1Mbps with one at 54 Mbps [3]. The gain of BPSK is experienced only by the transmitter, and the remaining users suffer as the channel is busy for a very long time as compared to a transmission based on 64QAM. The majority of this previous work was based on 802.11b systems. One exception is the work in [2] which studied the performance of a dense 802.11b/a network. The authors found that 72% of all handoffs were performed between APs on the same channel and the number of handoffs from and to the same wireless access point (AP) reached 54.7%. The authors conclude that 'current handoff mechanisms' are inadequate. The work in [7] is the most similar to our study. The authors study and correlate the impacts of physical layer impairment to APFEC performance in a stadium-based venue.

The basic mechanisms for sharing channel bandwidth in an 802.11g network are specified in [9]. 802.11 stations rely on carrier sense, random delays, and exponential backoffs to minimize collisions and to provide stability. Our study assumes an 802.11g network (i.e., 802.11b devices are not supported) and that CTS/RTS is not enabled. Multicast operates the same in all versions of 802.11. When an AP receives a multicast packet that is destined for one or more stations on the channel it broadcasts the packet over the channel. The AP might 'snoop' traffic arriving from wireless stations to keep track of which multicast group traffic must be forwarded. All stations receive the frame (provided the information is correctly received from the RF transmission), however only those stations that have joined the multicast group will accept the frame. Stations that accept the frame are not to issue an ACK back to the AP. The transmission timing used for unicast transmissions still applies. However, as with broadcast transmissions, multicast transmissions must use the basic rate.

In spite of the tremendous amount of academic research in the topic, multicast over wireless is still quite problematic (we refer to [10-12] however there are a large number of papers that focus on the problem). The obvious difficulty is that each wireless station that has joined the multicast group may experience conditions that are drastically different than those of other stations. Both large-scale propagation and multipath fading effects can cause stations that are located similar distances from an AP to experience widely different channel properties. This difficult is especially true in crowd spots. A second difficulty surrounding multicast over wireless networks is that modern video encoding techniques that are optimized for wireless are typically highly compressed making them sensitive to packet jitter and especially to loss.

Two well established techniques to counter these challenges are application level forward error correction (APFEC) and packet interleaving. APFEC can be used to mitigate the effects of packet loss by dividing the packet stream into blocks and adding a desired level of error correction data to the source data. Taking advantage of time diversity, packet interleaving reduces the impact of correlated packet loss by spreading loss events out across large timescales (larger than the timescale associated with loss correlation). Interleaving can increase the effectiveness of FEC as well as the effectiveness of error resiliency that is a part of the video encoding process. For real-time multicast streaming, end-to-end latency should be low. Therefore, in the research reported in this paper, we focus on APFEC and defer the study of packet interleaving for future work.

As described in [13-15], Raptor codes are well suited for video streaming over the Internet. We assume an 'ideal' code, which we refer to as APFEC, that offers the properties of Raptor codes including constant encoding and linear decoding costs and relatively low computation requirements. APFEC operates using symbols that are packets. The (N,k) parameters are specified in units of packets. We ignore the computation requirements associated with APFEC that become evident as N gets large.

While APFEC can increase the reliability of video broadcasts, it does so with a cost. As the strength of the error correction increases, the stream consumes more bandwidth and the average per packet latency increases. Both of these effects can directly impact the perceived quality of video streaming. As the overhead increases, it might require lower quality encoding to ensure the stream does not exceed a budgeted bandwidth allocation. The average packet latency increases as the block size grows as all packets following a gap in an arrival stream must wait until enough error correction data arrives to restore the lost packets. While increasing the size of the playback buffer can smooth the jitter in the stream of packets delivered to the video decoder, large playback buffers lead to large channel zapping times. When a user changes video streams, the channel zapping time is the time it takes to fill the playback buffer.

III. EXPERIMENTAL METHODOLOGY

The simulations were done using the Network Simulator -2 (ns2) version 2.33 [16]. Diagram 1 (located in the Appendix) illustrates the simulated network model that is considered in our study. A set of wireless stations interact with a set of wired nodes. The placement of each station with respect to the AP is an experimental parameter. In all experiments, the AP is considered the origin of the X-Y grid. We used a Ricean fading model that is available as a separately installable package for ns2 [17]. However for the results presented in this paper, the stations are positioned within 10 meters of the AP so that channel effects do not have a significant impact on our results[2].

We limit the study to a single 802.11g channel. We modified the ns2 802.11 simulation model to ensure that the MAC frame timing and state machine conforms to the IEEE 802.11g specification [9]. The physical layer is defined by a basic rate and by the rate at which application data is transferred (we refer to this as the data rate). The default settings assume a data rate of 54 Mbps and a basic rate of 6 Mbps.

We extended ns2's existing 802.11 model with multicast capability. Wireless stations can join a multicast flow statically through the initial configuration. As required by the 802.11 standard, the AP broadcasts multicast packets. Each station will receive all packets but will accept only IP packets associated with multicast sessions it has joined. APFEC is applied at the streaming server. For a code configured by N and k, the server will send a total of N packets per block of which k are data and N-k are redundant packets. The video streaming clients located at each wireless station that has joined the multicast group will receive the packets in a block. Once k packets are received, it passes those packets up the stack and ignores any additional packets that arrive for that block. If k packets do not arrive at the receiver within a block timeout amount of time, the receiver assumes the current block is complete and passes up whatever data is queued. If packets from the next block arrive, the receiver assumes the current block is complete and passes up packets from the block that are in the buffer.

All experiments are designed so that the wireless link is the bottleneck. To validate the configuration, we run a simple simulation with one unicast UDP flow configured with a CBR traffic generator that sends data at rate that exceeds the channel data rate. The flow is active between the monitor server node to the monitor station. The monitor is located close to the AP and consequently does not suffer channel impairment effects. The flow experiences an application throughput of 24 Mbps. If we use a single multicast UDP flow, we observe an average UDP throughput of 5.8 Mbps. Both results are similar to what we have measured in our lab and on our campus 802.11g network [8]. We confirmed in both cases that loss occurs at the AP's queue that holds packets waiting for a transmission opportunity to a wireless station.

The configured traffic workloads consist of a set of performance monitor flows and a set of multicast flows that represent the video broadcast traffic. The number of video streaming sessions that are active in a simulation is an experimental parameter. A video stream is modeled as a constant bit rate stream, parameterized by a packet size and interpacket departure time. Each multicast stream sends data at a rate of 768Kbps (1400 byte packets sent every 0.0146 seconds). Table 1 (located in the Appendix) summarizes the three experiments that are presented in this paper.

Performance of the system is assessed using a combination of network and application oriented performance metrics. The average one-way, end-to-end latency for all UDP packets of a given stream is monitored. We make use of the Maximum Burst Length (MBL) metric as defined in [18]. The MBL gives the sampled mean of the average loss run length based on the empirically derived probability distribution of the different loss run lengths. The MBL can be defined in units of time or in units of packets. We use units of packets.

We monitor the multicast flow that is received by Node 1 (refer to Diagram 1). We trace packet arrivals at this node (i.e., before APFEC is applied by the receiver). At the end of the simulation, a script processes the trace file and computes the MBL statistic. We refer to this result as the MBL Video Statistic. For a given APFEC setting, defined by a block size, N, and N-k redundant packets, we define the APFEC effectiveness as $Fec_{eff} = 1 - \frac{P_e}{P_{raw}}$ where P_e is the effective

[2] An extended version of this paper explores the impacts of channel impairments. Please refer to http://www.cs.clemson.edu/~jmarty /projects/WiFi/APFECPaperV7.pdf

loss rate that is observed by the video stream receiver after FEC and P_{raw} is the raw loss rate over the channel observed by the station before FEC is applied.

The final performance metric is the channel zapping time. For Internet streaming, a common artifact is the pause in viewing as the playback buffer is refilled [19]. To assess the possible impact of this artifact, we monitor the 'channel zapping' time which is the time required to fill the playback buffer when it is depleted. This time also reflects the cost (in time) to change video channels. We rely on the average channel zapping time as a measure of the negative impacts caused by large playback buffers. In ongoing work we are extending the simulation model to include the playback process which will provide further assessment details such as how frequently a stream might experience channel zaps due to network conditions.

An artificial loss process is applied at a link connecting two wired nodes. All results reported in this paper use the simple two parameter variant of the two-state Gilbert-Elliot (GE) model [20-21]. In the 'good' state packet loss occurs with a probability of *p*. A loss event triggers a transition to the 'bad' state where losses occur with probability *1-r*. A successful transmission in the bad state causes the model to return to the good state. We refer to the loss model parameters as the average good state duration and average loss run length. The loss process operates on units of packets and in particular on the aggregate stream of packets that flow over the link. Therefore, the loss correlation effects observed by a single flow depends on the percentage of the link capacity that the flow is using compared to the link capacity used by competing flows.

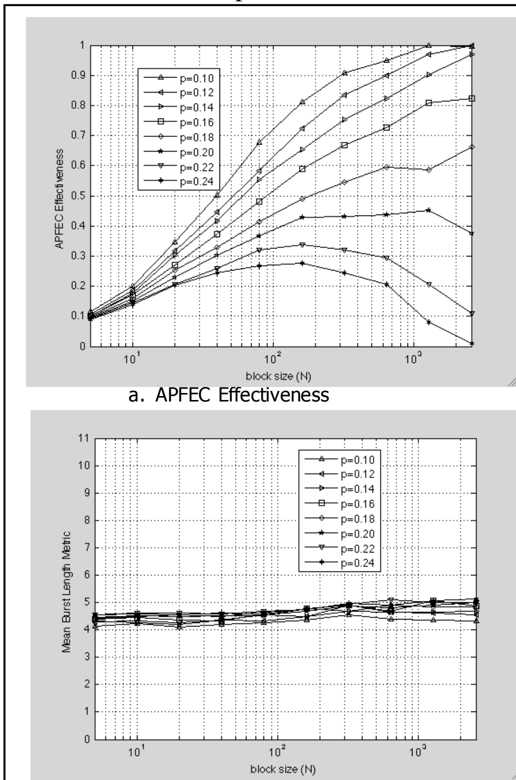
IV. ANALYSIS

Figure 1 illustrates the results of Experiment 1. For each curve in Figure 1a, the average loss rate was fixed and APFEC (N,k) parameters were varied however the level of redundancy was fixed at a value of 0.20. The GE loss process average good state duration was varied and the average loss run length was fixed at 25 packets. As shown in the legend, the average loss rate ranged from 0.10 to 0.24. In all simulations the actual loss rate produced by the ns2 GE loss generator was larger than the expected value by 1-5%. Our analysis is not affected by this as the APFEC efficiency assessment is based on the measured raw loss rate. Each curve plots 10 simulation results. Figure 1a illustrates the relationship between APFEC effectiveness, the block size, and the average loss rate. The results are consistent with the expected results of an ideal code as N goes to infinity [22]:

- If redundancy > P_{raw}, then $P_e \rightarrow 0$ as N goes to infinity
- If redundancy = P_{raw}, then $P_e \rightarrow P_{raw}/2$ as N goes to infinity
- If redundancy < P_{raw}, then $P_e \rightarrow P_{raw}$ as N goes to infinity

Each of the three cases above corresponds to an APFEC effectiveness of 1, 0.5, and 0.0 respectively. The results suggest that the block size required for convergence to an APFEC effectiveness of 1 increases exponentially as the loss rate approaches the redundancy. In this example, when the average loss rate exceeds the redundancy, the effectiveness peaks at a block size of approximately 160 packets before converging to a value of 0. Further analysis is required to generally characterize peak effectiveness as a function of redundancy, loss rate, and mean loss burst length. We also observe that very large simulation runs lengths are required for the statistics to converge when we use the largest block size. In our analysis, the statistics are based on simulation run times of 300 seconds which leads to sufficient statistical accuracy for the purposes of this paper. In ongoing work we are using appropriate statistical methods to better characterize the long term behaviors when large block sizes are involved.

Figure 1b graphs the MBL Video Statistic that is observed in the simulations. Each of the five multicast flows experiences roughly 1/5 of the intensity of correlated loss and explains why Figure 1b shows the multicast streams observe an MBL of about 4.5 packets.

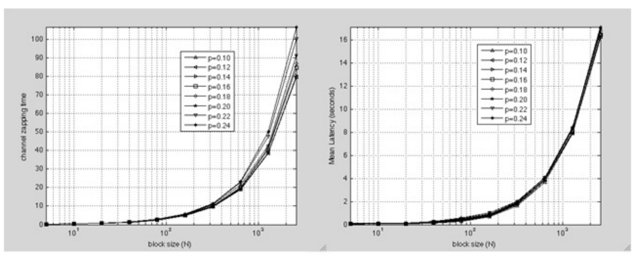


a. APFEC Effectiveness

b. Multicast MBL Metric

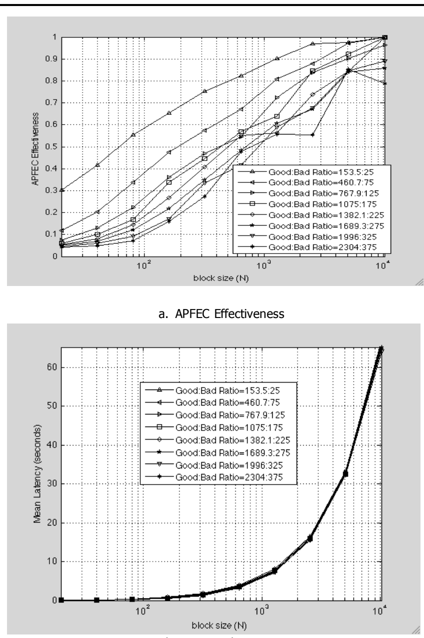
Figure 1. APFEC Performance Observed in Experiment 1

While it is apparent that large block sizes might be required to maximize APFEC effectiveness, Figure 2 illustrates the impact of large block sizes. Our model assumes the playback buffer at the streaming receiver is twice the size of the APFEC block size. Figure 2a suggests that it might take up to 100 seconds to refill the playback buffer. The model assumes the buffer will be refilled at the most recent arrival rate of video data. Therefore, as channel conditions or network congestion distorts the video stream, the channel zapping time increases. Figure 2b shows that the average end-to-end packet latency for the simulations grows to over 16 seconds at the highest block size. The majority of this latency is incurred as packets wait in the APFEC receive buffer for at least k out of the N packets in a block to arrive. Once the first packet drop occurs in the block, all subsequent packets must wait in the queue for the completion of the block. As long as the playback buffer capacity is greater than the APFEC block size, the playback buffer will smooth out the burstiness induced by the APFEC process.



a. Channel Zapping Time (seconds) b. Average Packet Latency (seconds)

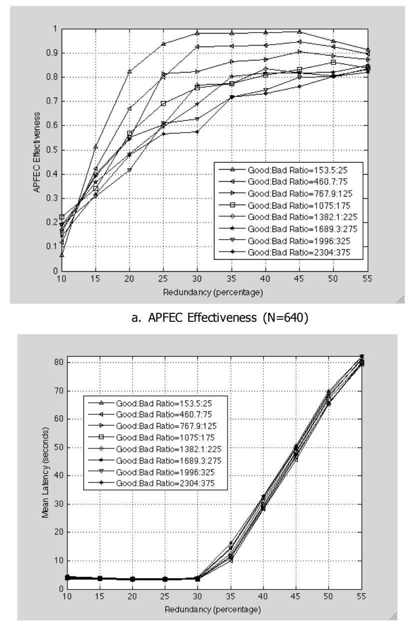
Figure 2. Delays Observed in Experiment 1



b. Mean Packet Latency

Figure 3. Experiment 2 Results

In Experiment 1 the average loss rate is varied but the level of correlated loss was held constant. In Experiment 2, the loss rate is held constant at a rate of 0.14 while the loss model's average loss run length is increased from a value of 25 packets to 375 packets. The FEC *(N,k)* parameters ranged from (20, 16) to (10,240, 8,192). The level of redundancy remained at 0.20 as in the previous experiments. As shown in Figure 3a, an APFEC effectiveness larger than 0.90 is observed only in the simulations in which the average loss run length did not exceed 175 packets. The APFEC effectiveness is based on the measured raw loss rate, the discrepancy does not impact our results. Figure 3b shows that the average packet latency exceeds 60 seconds as the block size reaches 10,240 packets. The channel zapping time (these results are not displayed) grows tremendously large (400 seconds) as the block size reaches the highest setting.



b. Mean Packet Latency

Figure 4. Experiment 3 Results

The objective of Experiment 3 was to find the minimum level of redundancy that is necessary to correct all errors when the block size is fixed at 640 packets. Figure 4a shows that loss cannot be fully eliminated. The after-FEC loss rates corresponding to the simulations with the least level of correlated loss never dropped below 1%. The minimum level of redundancy required to maximize effectiveness ranges from 30% to 45%. However beyond 45%, the FEC overhead increases the bandwidth consumption of each stream causing congestion in the 802.11 network and consequently the APFEC efficiency drops in some cases. A redundancy setting in the range of 0.20 to 0.30 corresponds to the optimal operating range for APFEC. Figure 4b shows that once the redundancy exceeds 30%, packet latency increases due to queue delay at the AP. The latency gets very large because the output buffer capacity at the AP is set to 60,000 packets. Once the redundancy reaches 45%, packet loss increases as the AP's output buffer reaches capacity. The channel zapping time (which is not shown) is less than 2 seconds in all simulations from Experiment 3. This is because the playback buffer is relatively small (640 packets).

V. Conclusion

The research presented in this paper has explored the use of APFEC to deliver broadcast video subject to loss processes that are likely to be similar to channel effects that end user's handheld devices might encounter in WiFi crowd spots. An objective of our research was to obtain a set of baseline results that provides guidance as to the required level of APFEC redundancy and the required block size that are necessary to mitigate loss in anticipated WiFi crowd spot conditions. Results from Experiment 3 suggest that if the average loss rate is 0.14, a redundancy level of 0.20 with a block size of 640 packets can eliminate between 40% and 82% of loss events. Experiment 2 suggests that if the block size is increased to 10,240 packets, the effectiveness of APFEC increases to between 0.80 to 1.0. However, increasing the block size from 640 to 10,240 packets causes the packet latency to increase from 4 to 60 seconds and the channel zapping times to increase from 2 seconds to 400 seconds. Although the results from [7] are not directly comparable, they did observe a similar favorable operating region for APFEC when code rates are between 0.70 and 0.75 and when block sizes between 200 and 500 packets are used (the raw packet loss process details were not provided by the authors).

In future work we plan on assessing and modeling the loss process of the WiFi network available to fans in our campus football stadium during home games. These results will allow us to further calibrate our models. The goal of the next phase of our research is to develop an efficient online algorithm that allows the system to adapt the APFEC and network system parameters to optimize performance.

References

[1] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, J. Zahorjan. "Measurement-based Characterization of 802.11 in a Hotspot Setting," in Proceedings of ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis, 2005.

[2] R. Raghavendra, E.M. Belding, K. Papagiannaki, and K.C. Almeroth, "Understanding Handoffs in Large IEEE 802.11 Wireless Networks," in *Proceedings of* Internet Measurement Conf. (IMC), *Oct. 2007.*

[3] A.P. Jardosh, K.N. Ramachandran, K.C. Almeroth, and E.M. Belding-Royer, "Understanding Congestion in ieee 802.11b Wireless Networks," in Proceedings of Internet Measurement Conf. *(IMC),* Oct. 2005.

[4] M**.** Balazinska and **P.** Castro**, "**Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network," in Proceedings of MobiSys, May 2003.

[5] B A. Balachandran, G.M. Voelker, P. Bahl, and P.V. Rangan, "Characterizing User Behavior and Network Performance in a Public Wireless LAN," *in* Proceedings of ACM SIGMETRICS*,* pp. 195-205, June 2002**.**

[6] T**.** Henderson**,** D**.** Kotz**,** and I. Abyzov**, "**The Changing Usage of a Mature Campus-Wide Wireless Network**,"** in Proceedings of ACM MobiCom, 2004**.**

[7] V. Sgardoni, M. Sarafianou, P. Ferre, A. Nix, D. Bull, "Robust Video Broadcasting over 802.11a/g in Time-Correlated Fading Channels", IEEE Transactions on Consumer Electronics, Vol. 55, No. 1, February 2009.

[8] M. Juang, K. Wang J.Martin, "A Measurement Study on Link Capacity of a High Stress IEEE 802.11b/g Network", in Proceedings of the 2008 IEEE International Conference on Computer Communications and Networks (ICCCN 2008), St. Thomas, Virgin Islands, August 2008.

[9] IEEE 802.11-2007 IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, June 2007.

[10] S. Choi and K. Choi, "Reliable multicast for wireless LAN,", in Resource, Mobility, and Security Management in Wireless Networks and Mobile Communications, Y. Zhang, H. Hu, and M. Fujise, Eds., CRC Press, Boca Raton, Fla, USA, 2006.

[11] M.-T. Sun, L. Huang, S. Wang, and T.-H. Lai, "Reliable MAC layer multicast in IEEE 802.11 wireless networks," Wireless Communications and Mobile Computing, vol. 3, no. 4, pp. 439–453, 2003.

[12] S. Kasera, J. Kuri, "Reliable Multicast in Multi-access Wireless LANS", in Proceedings of INFOCOM99, March, 1999.

[13] D. MacKay, "Fountain Codes", Proceedings of IEEE Communications, vol, 152, no 6, December 2005, pp 1062-1078.

[14] A. Shokrollahi, "Raptor Codes", IEEE Transactions InformationTheory, vol. 52,no. 6, June 2006, pp. 2551-2567.

[15] Luby, M., Stockhammer, T., Watson, M., "Application Layer FEC in IPTV Services", IEEE Communications Magazine, May 2008.

[16] Networking Simulator 2 (ns2), available at http://www.isi.edu/nsnam/ns/

[17] R. Punnoose, P. Nikitin, D. Stancil, "Efficient Simulation of Ricean Fading within a Packet Simulator", in Proceedings of the IEEE Vehicular Technology Conference, 2000.

[18] Nafaa A., Taleb T., Murphy L., "Forward Error Correction Strategies for media Streaming Over Wireless Networks", IEEE Communications Magazine, December 2007.

[19] N. Degrande, K. Laevens, D. Vleeschauwer, R. Sharpe, "Increasing the User Perceived Quality for IPTV Services", IEEE Communications Magazine, February 2008.

[20] Gilbert, E.N., "Capacity of a Burst-Noise Channel", Bell System Technical Journal 39(1960), pp. 1253-1265.

[21] Elliot, E.O., "Estimates of Error Rates for Codes on Burst-Noise Channels", Bell System Technical Journal 42 (1963), pp. 1977- 1997.

[22] J. Westall, J. Martin, "Modeling Application Based Forward Error Correction", technical report available at http://www.cs.clemson.edu/~jmarty/projects/WiFi/fecmodel.pdf, September, 2010.

Appendix

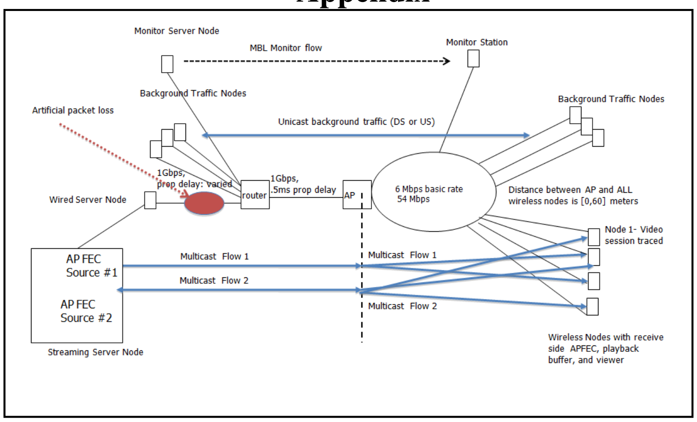


Diagram 1. Simulation Model

Experiment	FEC parameters	Flows under observation	Background traffic	Loss process
1	Variable block size (#1 below), fixed redundancy 0.20	5 multicast streams (5 stations/group) (768Kbps, packet size 1400)	none	Gilbert-Elliot, vary loss (#2 below) by varying state duration (#3 below)
2	Variable block size (#5 below), fixed redundancy 0.20	5 multicast CBR streams (768Kbps, packet size 1400)	none	Gilbert-Elliot, fix loss, vary level of correlation (#4 below)
3	Fixed block size 640 packets, vary redundancy	5 multicast CBR streams (768Kbps, packet size 1400	none	Gilbert-Elliot, fix loss, vary level of correlation (#4 below)

1. BlockSizes (N,k) (fixed code rate 0.20): (5,4), (10,8), (20,16), (40,32), (80,64), (160,128), (320,256), (640,512), (1280,1024), (2560,2048)
2. Long term loss rates: 0.10, 0.12, 0.14, 0.16, 0.18, 0.20, 0.22, 0.24
3. Gilbert-Elliot (varying rates) avg good run length : avg bad run length: (225:25), (182.5:25), (153.5:25), (131.5:25), (113:25), (100:25) (87,25), (79,25)
4. Gilbert-Elliot (fixed loss 0.14, vary run lengths for highly correlated loss: (153.5,25), (460.7,75), (767.9,125), (1075,175), (1382.1,225), (1689.3,275), (1996, 325), (2304,375)
5. BlockSizes (N,k) (fixed code rate 0.20): (20,16), (40,32),(80,64), (160,128), (320,356), (640,512), (1280,1024), (2560,2048), (5120,4096), (10240, 8192)

Table 1: Baseline Analysis Experimental Definition