Multicasting Video in Dense 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. End users are enticed with claims of hundreds of megabits per second data rates in emerging 4G systems only to find adequate support in limited geographic areas. 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. We have conducted a simulation-based performance analysis of video streaming. In order to scale, the streaming application relies on multicast along with application-level forward error correction (APFEC) to mitigate packet loss. Our results suggests that APFEC offers limited benefits in a network where the loss processes are effectively uncorrelated and when the mean loss rate grows quickly as users move out of range of the AP. However, we conjecture that loss processes in crowd spots are better modeled by a Gilbert Elliot loss model because synchronized movements of spectators, user's mobility, and handoff effects will likely contribute to the level of correlated loss. For well engineered crowd spot deployments, the network can provide a targeted maximum average loss rate to devices which would allow the system to operate in a region that is in APFEC's 'sweet spot'. Our results confirm that in certain conditions, APFEC is highly effective. However, we also show that the effectiveness of APFEC in crowd spots is a result of a complicated interaction between the stochastic structure of the loss process and underlying channel effects, the 802.11 MAC protocol, the sending rate associated with the stream, the behavior of users, and the choice (N,k) parameters.**

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.

Our motivating example is a sports and entertainment venue. While 802.11 has been deployed in stadiums and arenas for many years, WiFi connectivity was generally limited to facility support operations or to provide hotspots at specific locations. Smartphones are expected to make up over 50% of all cellular phones by 2014 [1]. 3G/4G operators view WiFi networks as a way to offload data traffic from their broadband network. 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 [2-8]. 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 [9]. 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). The fact that potentially many of these

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).

incidents can occur within the course of a game, along with the fact that the intensity and the frequency of these incidents will continue to increase over the next decade, motivates our research in wireless systems that support crowd spots. An objective of the research presented in this paper is to explore how 802.11 scales in crowd spot situations such as at athletic events in stadiums. We leave for future work a study exploring the benefits of collaborative wireless networks involving WiFi and 3G/4G broadband systems.

In sports and entertainment crowd spots, it is likely that the applications of interest consist of streamed video (either replays or multiple video camera feeds), web browsing to view game stats, or Internet browsing by fans to access similar content from other events occurring in different locations. 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' different channels to watch. 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.

Distributing broadcast video to wireless devices has been successfully accomplished in crowd spots using other technologies. Satellite is an established broadcast technology although it likely does not offer the best form factor for the type of system we are studying. A more relevant example involves NASCAR racing events where spectators can view video and audio streams from each car in the race. We are aware of at least one company, Kangaroo TV, that provides handheld devices to race fans willing to pay the service fee. In this example, Sprint provides controlled access to licensed spectrum. It is clear that the direction is to use smart phones rather than custom devices.

The research presented in this paper explores the efficacy of WiFi to support crowd spots where video streaming is the dominant application. We limit the study to 802.11g as this is the standard of WiFi is universally supported by the current generation of smart phones. In spite of the increase power requirements of 802.11n, we expect that future smartphones will support 802.11n. We defer to future work the potential advantages 802.11n contribute to supporting crowd spots. In order to scale, we assume video content is distributed using multicast. It is likely that the video streams will be highly compressed making them sensitive to packet loss. We use application forward error correction (APFEC) to mitigate the effects of packet loss on the video stream. However, a fundamental tradeoff exists: as the strength of FEC increases so will the bandwidth consumed by the stream. We present results of a simulation study that address the following questions:

- For sets of simulation scenarios involving realistic traffic workloads, how well can the system scale? What are the limiting factors?
- What are the tradeoffs surrounding the key parameters including the APFEC strength, the video encoding rate, the playback buffer, and the 802.11 frames size, and choice of modulation and coding settings?
- How can the system be optimized?

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. Section V describes related work and relates prior results to our results and conclusions. Section VI and VII describe the methodology of experiments and their results. The paper is concluded in Section VIII.

II. BACKGROUND

A tremendous amount of research has explored 802.11. The majority of this prior work has focused on 802.11b systems. Prior work has identified the challenges and limitations of large scale 802.11 networks [2-8]. The work in [2] 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 [3] 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 [4,8]. 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. The only work that we could find involving anything other than 802.11b was [3] 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 AP reached 54.7%. The authors conclude that 'current handoff mechanisms' are inadequate. Aside from a small number of studies (including our own [9]), a relatively small amount of measurement-based studies of large-scale 802.11g networks has been done.

IEEE 802.11g Summary

As specified in [10], the main differences between 802.11g and 802.11b are:

- 802.11g supports an Extended Rate Phy (ERP) Orthogonal Frequency Division Multiplexing (ERP-OFDM) mode at 2.4 GHz. This mode supports modulation and coding combinations that offer data rates up to 54 Mbps.
- The set of basic rates (i.e., the set of rates that all clients associated with a particular AP must support)

is expanded in 802.11g. The 802.11g basic rate set includes all basic rates supported by 802.11b and also rates ranging from 6 Mbps to 54 Mbps.

- 802.11g has smaller slot times. The default DIFS, and CWmin values are reduced.

However, the basic mechanisms for sharing channel bandwidth have not changed. As illustrated in Figure 1, stations rely on carrier sense 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.

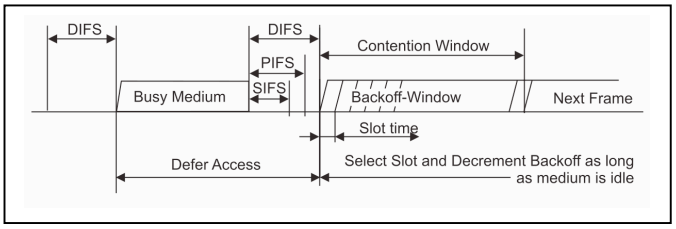


Figure 1. 802.11 Timing

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 as illustrated in Figure 1 still applies. As with broadcast transmissions, a multicast transmission must use the basic rate. For our study we assume a basic rate of 6 Mbps.

In spite of the tremendous amount of academic research in the topic, multicast over wireless is still quite problematic (we refer to [11-13] 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 could observe drastically different conditions than 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. Under certain condition APFEC can eliminate packet loss. In realistic scenarios however, packet loss events are correlated over time. FEC assumes loss events are uniformly distributed and consequently will lose its effectiveness as the duration of loss bursts increase. 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.

Forward Error Correction

Shannon theorized that error correction codes could support error free transmissions that approach the 'Shannon Capacity' of the channel [14]. The Reed-Solomon (RS) block coder, which is a Maximum Distance Separable (MDS) erasure codes, ensures that up to N-k lost symbols can be regenerated as long as k out of N symbols arrive intact. This code has been widely implemented in network communications equipment however its application to the higher layers of a networking stack has been limited due to computational complexity[15].

Much of the early work in coding theory was driven by telecommunications requirements and operated on bit streams. Since the late 1990s the Internet community has advocated for the use of FEC at the higher layers of the networking stack [16-18]. Advances in computation capabilities of general purpose computers (including handheld devices) has recently renewed interest in application level FEC. Two recent codes that have further enabled application-layer coding are Low Density Parity Check (LDPC) codes and Fountain codes. LDPC codes are low complexity and can operate on very large block sizes. However, they do not have the same 'ideal code' property that RS provides. Fountain codes have received much attention recently as they are deemed appropriate for the coding required by streaming video applications [19]. One necessary property of application-level FEC is that it operate on 'symbols' that are packets. Raptor codes, which are a type of fountain code, are considered more versatile for streaming than RS or LDPC because of reduced constraints on the choice of N and k [20]. Raptor also offers constant encoding and linear decoding costs making them appropriate for live streaming or Video-on-Demand applications. A Raptor code collects a set k source packets and produces a block of N packets, typically laid out as the k source packets followed by (N-k) packets containing error correction data. For a fixed k, N can be arbitrarily large. If the receiver receives k or more packets from the block, it can correctly recreate any source packets that were dropped. The *coding rate* (also referred to as the coding efficiency or simply the rate) is defined as k / N. The *redundancy* is defined as (N-k)/N or (1-rate). In our modeling, we ignore the computation requirements associated with APFEC that become evident as N gets large. In some

codes, it is possible to find a compromise between complexity and FEC efficiency by introducing a 'receiver overhead' parameter (referred to as ε) [21]. When ε is non-zero, on average $(1+\varepsilon)k$ symbols are required to recover the message. Our work assumes ε is 0.

In our study, we consider an 'ideal' code meaning a systematic erasure code that is MDS. This code, which we refer to as APFEC, operates using symbols that are packets. The (N,k) parameters are specified in units of packets. 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 selects a new stream to view, the 'channel zapping' delay is the time to fill the playback buffer before the video stream is displayed.

III. Experimental Methodology

The simulations were done using the Network Simulator - 2 (ns2) version 2.33 [22]. Figure 2 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 [23].

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 [10]. The simulation model supports a single modulation and coding setting, not adaptive modulation and coding (AMC). 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 receiver capture model is defined by a receive threshold which is equivalent to a radio's receiver sensitivity. Packets are received if the signal strength observed when the last bit of the frame arrives is within the radio's receive sensitivity threshold. If not, the packet is dropped.

A simulation experiment is in part defined by the modulation and coding that is assigned to the channel. The default settings assume a data rate of 54 Mbps and a basic rate of 6 Mbps. The receiver threshold is set to a value of -87dBm which is the receiver sensitivity for a Cisco 1250 AP operating at BPSK ½ [24]. As this AP's receiver sensitivity for 64QAM is -76dBm, the receiver capture behavior correctly models the sensitivity of broadcast traffic.

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. As shown in Figure 2, multicast sessions are actually treated as unicast over the wired network (prior to the WiFi network), although packets are marked with the multicast session (this field is '-1' if the packet is not a part of a multicast stream). When these packets arrive at the AP, the wireless channel and the 802.11 MAC ensure that multicast frames are received only by wireless stations that are members of the group. This restricts the multicast traffic flow to be from the AP to wireless stations. This also requires the wireless sinks to be located at the wireless devices. Both restrictions are sufficient for our study.

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. For the results presented in this paper, we assume the receiver overhead is negligible. Therefore, once k packets are received, it passes those packets up the stack and ignores any additional packets that arrive for that block. If the server does not have sufficient source data to fill a block, after a timeout amount of time waiting for additional source data, the server pads the block with filler data and completes the transmission of the block.

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 with a sending 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 and a one-way latency of 33 milliseconds. If we use a single multicast UDP flow, we observe an average UDP throughput of 5.8 Mbps. We confirmed in both cases that loss occurs at the queue at the AP that buffers packets waiting for a transmission opportunity over the downstream channel.

Traffic Models

The configured traffic workloads can consist of a set of performance monitor flows, several types of background traffic, and a set of either unicast or multicast flows that represent the video broadcast traffic. The number of video

streaming sessions viewed by each station as well as the number of stations that sink one or more streams are experimental parameters. A video stream is modeled as a constant bit rate stream, parameterized by a packet size and interpacket departure time. Unless indicated otherwise, all video streams are transmitted at a constant rate of 768Kbps implemented by sending fixed size packets (1400 bytes by default) every 0.0146 seconds.

Experiments involve up to N wireless stations. A number of stations are configured to generate background traffic while other stations will participate in one or more multicast groups. In addition to the desired level of multicast traffic, the work load will consist of background TCP traffic (downstream back logged TCP flows). The choice of propagation models and the placement of the wireless stations with respect to the AP are additional experimental parameters.

Performance Metrics

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. Assuming no APFEC, no channel errors, and no congestion, the minimum one-way latency between the Monitor Server Node and the Monitor Station wireless node is 4.6 milliseconds.

We make use of the maximum burst length (MBL) and mean interloss distance (MILD) metrics as defined in [25]. 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.

Given the vector m_i (where *i=1,2, ...n)* that denotes the number of loss bursts having length *i,* the MBL is defined as follows:

$$MBL = \frac{\left(\sum_{i=1}^{n-1} i m_i\right)}{\sum_{i=1}^{n-1} m_i}$$

Given the vector d_i (where *i=1,2, ...n)* that denotes the number of runs with no loss having length *i*, the MILD is defined as follows:

$$MILD = \frac{\left(\sum_{i=1}^{n-1} i d_i\right)}{\sum_{i=1}^{n-1} d_i}$$

We create a separate monitor flow that emulates a unicast video streaming session. APFEC is NOT applied to this stream. The objective is to assess the loss correlation structure over the channel that is observed by the Monitor Station. At the end of the simulation, the receiving side of this session computes the MBL and MILD metrics based on the observed packet arrivals. We refer to this as the MBL Monitor. As shown in Figure 2, we position an MBL Monitor at the wired monitor server node and at the monitor wireless station. The MBL Monitor traffic generator sends a unicast stream of UDP packets of a desired size and at a desired frequency to the stream sink. As with the multicast video streams, the default behavior of the traffic generator sends 1400 byte packets every 0.0146 seconds. At the end of the simulation the MBL metric is computed.

We also compute the MBL that is observed by one of the video flows (we always used the flow that is observed by the first streaming node identified as Node 1 in Figure 2). Depending on the simulation, this flow is either unicast or multicast. We trace packet arrivals at the receiving 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.

In addition to the MBL metrics we also assess the effectiveness of APFEC in each simulation. For a given APFEC process 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 loss rate that is observed by the video stream receiver after FEC and P_{raw} is the loss rate over the channel observed by the station before FEC is applied.

The final set of performance metrics provide a relative assessment of the quality of the rendered video as observed by a station. Assessing the perceived performance of video streaming is challenging. Reference-oriented assessment techniques require the original source video content to be compared with the content that is decoded at the receiver [26]. The receiving side of the assessment would typically compute a peak signal-to-noise ratio (PSNR) measure to quantify the level of distortion caused by the network. However, it has been shown that PSNR might not accurately capture user perceived quality [27,28]. A second class of video quality assessment maps network measurable information to occurrences of 'artifacts'. An artifact is a general term that implies any imperfection to the final visualized video stream. The work in [28] shows that the most distracting artifacts to the end user are blurring, blocking, and color distortion. For Internet streaming, a common artifact is the pause in viewing as the playback buffer is refilled.

The difficulty in programmatically assessing video streaming quality artifacts is twofold: first, mapping network performance statistics to a perceived quality assessment

depends on many factors, including the details of the video encoding/decoding process, the interaction between the playback buffer and the streaming server, and the sensitivity of the video content to loss and latency dynamics that are in effect over the path; second, the perceived quality assessed by human users is highly subjective.

In our study, we define the following generic metrics that serve to provide us with a relative measure of video quality (relative with respect to baseline simulation results). The metrics are aligned with traditional broadcast communities guidelines such as that the rate of artifacts should not exceed 1 per 4 hours [21]. The metrics are based on the stream of packets observed by the same stream that is maintaining the MBL Video Statistic. However, the perspective of the artifacts is the stream after APFEC has been performed. The trace is divided into time intervals (we use 20 seconds). We define the *indicator of an artifact* in interval *i* as:

$A_i = \{0,1\}$ where *i* specifies the interval number in the range {0,N}. A value of 0 implies the artifact is not expected to happen during interval *i*, a value of 1 predicts the artifact would occur during the *i'th* interval.

- Loss Tolerance Artifact (LTA): The video player is assigned a static loss rate threshold. As most video codecs provide resiliency to loss, the assumption is that dropped packets do not significantly affect the perceived quality until it exceeds some threshold. By default we assign this threshold to be 0.02 (2%). Therefore, the artifact indicator is defined as:

  $A_i = 1$; if $l_i > l_{threshold}$; else $A_i = 0$; where l_i is the observed loss rate during interval *i* and $l_{threshold}$ is the configured loss tolerance threshold of the codec. We assume that the probability of an artifact occurring in any particular interval is independent of what happens in any other event. The probability of an artifact occurring in an interval is given by:

  $p_{lta} : \frac{\sum_{i=1}^{N} A_i}{N}$ where $N = \frac{T_{totalTime}}{T_{\text{int}Time}}$ is the number of intervals. The expected number of artifacts in N intervals is: $E[x] : \sum_{i=1}^{N} i p_{lta}$. The LTA is defined as the expected number of artifacts in 1 hour (i.e., where N is the number of intervals in the total time of the simulation).

- Minimum Throughput Artifact (MTA): In a manner similar to the LTA, the MTA defines an artifact incident in interval i is if the average arrival rate is less than a minimum required throughput threshold. We define the minimum throughput as:

  $T_{\min} = c * r_{encode}$ where *c* is a course measure of the level of compression of the stream (ranging between 0 and 1, with a value of 1 implying no compression) and r_{encode} is the average encoding rate of the video. Our results reflect a *c* of 0.75. The artifact indicator is defined as:

  $A_i = 1$; if $r_{pa} < c * r_{encode}$; else $A_i = 0$; where r_{encode} is the encoder rate used to create the stream and r_{pa} is the observed arrival rate at the playback buffer. The expected number of artifacts in 1 hour is:

  $E[x] : \sum_{i=1}^{N} i p_{mta}$. The: MTA is defined as the expected number of artifacts in 1 hour.

- Playback Buffer Depletion Artifact (PBDA): If the station experiences a long fade or if congestion occurs in the network, the arrival of packets is interrupted forcing the player to drain the playback buffer. The drain rate is defined as:

  $r_d = c r_{encode} - r_{pa}$ where c is the compression factor, r_{encode} is the video encoding rate, and r_{pa} is the observed arrival rate at the playback buffer.

  The PBDA analyzes the trace of packet arrivals to the playback buffer and computes the drain rate observed in each interval. The metric computes the number of bytes drained per interval, maintaining the count across intervals. If the drain rate is less than or equal to 0 in any interval, the count is reset. A PBDA incident occurs if the number of bytes drained exceeds the size of the playback buffer. In each simulation, we assume the playback buffer (in units of packets) is set to twice the APFEC block size. Once the buffer is depleted, the actual artifact is likely to be a pause in the rendering of the video to give the playback buffer time to refill. Our model assumes that the refill rate is the average rate of packet arrivals that has been observed in the recent history of the current artifact. The final PBDA metric is the expected number of times the playback buffer depletion happen per hour.

- Channel Zapping Time: This estimates the time required to fill the playback buffer once it is depleted. This artifact supplements the PBDA artifact by estimating the impact of playback buffer depletion incidents.

Collectively these artifact metrics provide additional information that helps quantify system performance. Each has one or more tunable parameters that allow the assessment to provide a meaningful assessment that can be customized to the details of the streaming implementation.

IV. ANALYSIS

We divide the analysis into two components. First we use an artificial loss process in an uncongested network to evaluate the sensitivity that controlled levels of correlated loss has on APFEC effectiveness. We position the stations so that none will ever experience channel impairment. Second, we examine the impacts of loss driven by channel impairment rather than by analytic loss models. We extend the latter component to additionally show the impacts of loss that is based on a combination of channel impairment and network congestion.

.

Artificial Loss Analysis:

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 [29-31]. 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.

Experiments 1 – 5 that are identified in Table 1 represent the baseline analysis. Experiments 1 and 2 involved unicast video flows while the remaining experiments involved multicast video. We fix the code rate to 0.80 for all experiments. The experimental parameters are the FEC parameters (i.e., the *N* and *k*) and the loss model settings. In all experiments, we varied the loss rate between 0.10 and 0.24. The first experiment uses a Bernoulli loss model and the remaining experiments use a GE loss model. Experiments 1 - 4 hold the level of correlation constant and vary the average loss rate. The fifth experiment uses a GE loss model, holds the average loss rate constant and varies the intensity of correlated loss.

Figure 3 illustrates the results of experiments 1 and 2. Figures 3a and 3c visualize the performance when individual packet loss events are independent. Figures 3b and 3d illustrate when packet loss events are correlated. Each of the eight curves plotted in each figure involves a different average loss rate. Each curve plots ten simulation results. For each curve, the average loss rate was fixed and APFEC parameters (i.e., the N,k) were varied as follows: (5,4), (10,8), (20,16), (40,32), (80,64), (160,128), (320,256), (640,512), (1280,1024), and (2560,2048). The results visualize APFEC effectiveness as the block size (N) is varied.

Uncorrelated loss scenario

Assuming uncorrelated packet loss in the network we expect the following results [32]:

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

When subject to a Bernoulli packet loss process, the MBL can be modeled as a geometric random variable. The expected MBL is simply *1/(1-p)* where *p* is the average loss rate. This corresponds to an expected MBL ranging from 1.11 to 1.32 for the loss rates considered. Figure 3c illustrates that the MBL observed in the simulator by the MBL Monitor ranged from 1.16 to 1.36. The simulation results are about 4% higher than the theoretical results. We attribute the difference to subtle perturbation to the system caused by the performance monitor traffic and also by 802.11 collisions.

Figure 3a confirms the analytic results suggested above. APFEC is able to correct all errors as long as the redundancy is less than the average loss rate and as long as N is large enough. If the redundancy equals the average loss rate, the effective loss rate can be P_{raw}/2 (again, as long as N is large enough). The final case, when the redundancy is less than the average loss rate, results in no benefit.

Correlated loss scenario

For each curve in Figure 3b, the average loss rate was fixed and the 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. Figure 3d shows the MBL metric estimates the average loss run length at about 4 packets. The reason why this is not 25 is because there are five multicast flows (in addition to the performance metric flows) sharing the channel which effectively spreads the effects of loss bursts across all flows. 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.

Experiment 3 is identical to experiment 2 except that the five unicast flows are replaced with five multicast flows each with five stations in the group. As expected, the results visualized in Figure 4a are very similar to the unicast results involving correlated loss illustrated in Figures 3b. Since each wireless station does not consume or generate any background traffic, the results are independent of the group size. Figure 4b illustrates the MBL that is observed by the monitored video streaming flow. Figure 4c illustrates the MBL that is observed by the unicast MBL monitor flow. Because the loss process is outside the wireless network, it impacts unicast and multicast flows that send the same packet rate in a similar manner. The minor differences observed between Figures 4b and 4c are due to the different methods used to obtain the MBL statistic.

We repeat the experiment but limit the traffic to a single multicast flow. Figures 4d shows that the single multicast flow is subject to higher level of correlated loss than the multiple flow experiment. Figure 4e indicates the monitored multicast video flow observes an MBL of about 10 packets. Figure 4f indicates that the unicast video flow experiences an MBL of about 9 packets. Because the loss process as well as the MBL analysis is in units of packets, the performance monitor results suggest that performance is dependent on the number of flows. In a real system, correlated loss is a function of time. If we were to modify our methodology to use units of time rather than packets, we would expect the results for the single multicast flow case to be similar to the results for the five flow case. The simulated loss process operates in units of packets. There are two flows in experiment 4 (the single multicast flow and the MBL monitor flow) that share the effects of the correlated loss process.

Figure 5 illustrates additional results from experiment 4. Figure 5a illustrates the average loss rate observed by all multicast video stream receivers after APFEC has been applied. These results directly reflect the APFEC effectiveness measure shown in Figure 4a. The loss rate observed for several data points exceeds the configured loss rate of the loss process. The additional loss is due to packet loss that occurs over the 802.11 channel caused by collisions.

Figures 5b through 5d are the results based on the video artifact metrics. The Loss Tolerance Artifact results in Figure 5b show that when the average loss rate is 0.14 or less, block sizes in the range of 640 to 1024 are required to ensure the rate of LTA is less than 10 per hour. Figure 5c indicates that the rate of MTTA is low for the lower loss rates and moderate block sizes. Once the block size exceeds 256 the higher loss rates experience an MTTA rate larger than 50 per hour. As the block size increases, the departure process after APFEC becomes bursty, especially at the higher loss rates. This burstiness leads to more frequent MTTA incidents. The PBDA rate is high only for the high loss rate scenarios when the block size is 16 or less. Once the block size exceeds 16, the gaps in the arrival rate of video traffic to the decoder are not long enough to fully deplete the playback buffer.

While it is apparent that large block sizes can be required to maximize APFEC effectiveness, Figure 6 illustrates the negative impacts. Figure 6a suggests that it might take up to 3 minutes to refill the playback buffer. The zapping time is a function of not just the block size, but also the arrival rate of packets to the playback buffer. Therefore, as channel conditions or network congestion distorts the video stream, the channel zapping time increases. The model assumes the buffer will be refilled at the observed arrival rate of video data. Figure 6b 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 side buffer for at least k out of the N packets in a block to arrive. All it takes is one packet in the block to be lost to cause all subsequent packets to have to wait in the queue for the completion of the block. As long as the playback buffer is as large as the APFEC block size, the playback buffer will smooth out the burstiness induced by the APFEC process. Our assumption that the playback buffer is twice the block size will tend to overestimate the channel zapping time however we thought it to be a reasonable design choice. The viewer can chose to begin playback once ½ the buffer has been filled. We anticipate that the optimal size should be the smallest size that minimizes the playback buffer depletion artifact metric rate.

In experiment 4 the average loss rate is varied but the level of correlated loss was held constant. In experiment 5, the loss rate is held constant at a rate of 0.14 while the GE loss model parameters are varied from (153.5, 25) to (2,304, 375). The FEC *(N,k)* parameters were adjusted from (20, 16) to (10,240, 8,192). The code rate remained at 0.80 as in the previous experiments. As shown in Figure 7a, the minimum APFEC block size required to correct all loss was 5,120 packets. For the highest level of correlated loss, the highest block size was not able to correct all loss. Figure 7b illustrates the raw loss rate for each simulation. As seen in Figure 5a, the average loss rate statistic is higher than the configured loss rate due to additional loss of the 802.11 channel. Figure 7c shows that the average packet latency gets very large as the block size reaches 10,240 packets.

The objective of experiment 6 is to find the minimum level of redundancy that is necessary to correct all errors when the block size is fixed at 640 packets. Figures 7d shows the

redundancy to correct errors in all loss cases is at least 50% (which corresponds to a coding rate of 0.50). Figures 7e and 7f show the challenges surrounding finding the best level of redundancy. Once the redundancy exceeds 40%, the additional overhead consumes all available (basic rate) bandwidth. Packet loss and queue delay increases as a result of the congestion.

Channel Impairment Analysis

The channel impairment analysis assesses the impact of realistic loss processes based on established channel models. We use two propagation models: a shadowing model that is provided by the ns2 simulator and a Ricean fading model that is available as a separate package for ns2 [30]. It has been established that signal strength samples between a transmitter-receiver pair at fixed distance is random and distributed log-normally about the large scale propagation value [15]. This behavior is captured by a log-normal shadowing model which is parameterized by the standard deviation of the Gaussian distribution (specified in units of dB). Fading models are used to characterize the propagation effects when the signal that arrives at a receiver suffers from rapid fluctuation due to multipath effects. When a dominant non-fading signal component is present (such as in a well engineered WiFi crowd spot), the small-scale fading effects can be modeled by a Ricean distribution. The Ricean distribution degenerates to a Rayleigh distribution when the dominant signal fades away. A Ricean channel model is parameterized by the K-factor which specifies the ratio between the dominant signal power and the variance of the multipath signals in units of dB. A K-factor of 0 dB causes the channel to experience Rayleigh channel effects.

Figure 8 illustrates the results of an experiment that explores the impacts of both propagation models on APFEC effectiveness. Multicast streams are configured as in the previous experiments, however just three streams are established rather than five with 10 stations in each multicast group. Figures 8a-8c illustrate the impact the shadowing model's Gaussian distribution standard deviation parameter has on APFEC effectiveness. Figure 8a suggests that the average effectiveness experienced by all 30 wireless stations that sink a multicast stream is independent of the APFEC block size. This is some what misleading as each station experiences different levels of APFEC effectiveness. Figure 8b illustrates the APFEC effectiveness experienced by one station located 18 meters from the AP. Figure 8c illustrates the effectiveness of APFEC experienced by a client located 48 meters from the AP.. For stations within a distance of 18 meters from the AP, the effectiveness of APFEC follows the expected results for an uncorrelated loss process. APFEC corrects all loss as long as the loss rate is less than the redundancy (0.20 in this case). The raw loss rates ranges from 0.05 to 0.25 as the Gaussian standard deviation increases from 4.0 to 10.5. It turns out that the raw loss for when the standard deviation parameter is 7.9 is 0.20 which is the level of redundancy. For this parameter setting, the APFEC effectiveness becomes unstable as the block size gets large. Multiple runs shows the trend is for the effectiveness to decrease. Further analysis is required to explain why the effectiveness does not converge to the expected value of 0.50.

Figures 8d-8f show the results for an identical experiment using a Ricean channel model. The two channel models exhibit differences in their respective large scale path loss assumptions. The shadowing model results indicates that stations must be within 18 meters of the AP. The Ricean model results suggests that with APFEC, stations can be located beyond 48 meters. The station at 18 meters observed loss in the range of 0.01 to 0.07, which is correctable by APFEC. At 48 meters, the loss rate ranged from 0.0 to 0.18, which is also correctable by APFEC. The main observation from the results in Figure 8 is that both propagation models do not produce correlated loss. In all simulations, the MBL statistic never exceeded a value of 1.5 packets which indicates the loss process is uncorrelated.

The final experiment adds varying levels of competing backlogged TCP traffic to the prior experiment using a K-factor of 6 dB. Figure 9a illustrates that one or more competing TCP flows reduces the average APFEC effectiveness observed by all stations from about 0.85 to less than 0.35. Further, the results suggest that the effectiveness tends to decrease as the block size increases. Figure 10 illustrates the raw and effective loss (i.e., after APFEC) that was observed by the stations located at 18 and 48 meters respectively. At 18 meters, the raw loss rate ranged from 0.20 to 0.30 for the simulations with at least one competing TCP flow. Figure 9b shows that the effectiveness drops as the block size grows from 20 to 80 packets. The raw loss rate is fairly constant in this range of block sizes (at about 0.30). We conjecture that an interaction between APFEC and 802.11 is causing the drop in APFEC effectiveness.

At 48 meters, the raw loss rate was between 0.40 and 0.48 for the simulations with a least one TCP flow. The APFEC results illustrated in Figure 9c show that this loss rate is beyond APFEC's ability to correct dropped packets. For the simulations with 0 background TCP flows, the raw loss rate is about 0.20. As in the previous experiment, we see somewhat unexpected behavior as the block size increases (i.e., the effectiveness drops from 0.50 to 0 as the block size approaches 5,120 packets).

The main conclusion that we draw from the experiment visualized in Figures 9-11 is competing unicast traffic has a significant impact on multicast traffic. Backlogged unicast traffic will consume a disproportionate amount of bandwidth due to TCP's congestion control algorithms. The 802.11 MAC protocol's use of acknowledgements for unicast traffic is the root cause of the unfairness. An area of future work is to define an 802.11e MAC parameter combination that allows a network operator a control knob to specify the allocation

fairness between unicast and multicast traffic.

Figure 11 shows the artifact results as observed by the station at 18 meters for this experiment. Figure 11a shows that the LTA metric drops to a reasonable number for very large block sizes. As the block size increases, the departure rate after the receiver's APFEC process becomes bursty. This causes fewer intervals that exceed the loss threshold but these intervals greatly exceed the threshold. A similar behavior occurs with the MTA metric. However Figure 11b does show that the metric is more sensitive to the number of competing TCP flows than the LTA. For the simulations with 3 or more competing TCP flows, the rate of MTA peaks when the block size is 640 packets. Smaller block sizes tend to lead to smooth departures from the APFEC receiver. The departure process becomes bursty as the block size grows the number of intervals that exhibit a low throughput also increases. As the block size grows, the metric indicates extreme burstiness as the majority of intervals reflect a throughput of 0 and a small number of intervals reflect a throughput that is in the range of mbps. The Channel zapping time reflects the increase in the average time to refill the playback buffer as the block size increases.

It is clear that channel effects largely dictate performance of APFEC. We saw a significant difference using two widely use propagation models. The quantitative results are artifacts of our model. Further, the channel impacts were urealistically static. In a crowd spot, we anticipate that channel conditions will vary over time due to synchronized movement of spectators , user mobility, and bursty competing traffic. In ongoing work, we are developing more accurate channel models of crowd spot WiFi networks.

V. Conclusion

Much of the prior research that has studied large scale 802.11 deployments has focused on characterizing the network subject to user behaviors. The research presented in this paper shares this goal but focuses on systems that must deliver broadcast video in dense 802.11 deployment situations which we refer to as crowd spots. The analysis that we have presented focuses on the impact of different types of loss processes video streaming. The assessment was based primarily on an APFEC effectiveness measure and on a artifact metrics that better assess the subsequent impact on the application. We showed that APFEC offers limited benefits in a network where the loss processes are effectively uncorrelated and when the mean loss rate grows quickly as users move out of range of the AP. We conjecture that loss processes in crowd spots are better modeled by a Gilbert Elliot loss model because synchronized movements of spectators, user's mobility, and handoff effects will likely contribute to the level of correlated loss. For well engineered crowd spot deployments, the network can provide a targeted maximum average loss rate to devices which would allow the system to operate in a region that is in APFEC's 'sweet spot'. Based on the results from experiment 6 as an example, we consider a moderately correlated loss scenario with an average MBL of 5 packets and with an average long-term loss rate of 0.10. The majority of loss can be corrected by using a coding rate of 0.25 and a block size of 640 packets.

The effectiveness of APFEC in crowd spots is a result of a complicated interaction between the stochastic structure of the loss process and underlying channel effects, the 802.11 MAC protocol, the sending rate associated with the stream, the behavior of users, and the choice (N,k) parameters. We have observed a variety of behaviors, with some unexpected results when realistic channel models were considered. However the core challenges are readily apparent:

1. The redundancy of APFEC must be less than the average loss rate. The challenge is the average loss rate is highly variable across the set of stations in the multicast group.
2. As the level of correlated loss increases, larger APFEC block sizes are required. As the block size increases, the playback buffer size must grow. This however is problematic as the channel zapping time can get large.

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 can be used to help identify realistic simulation scenarios to further study in simulation. Our future work will also determine the necessary 802.11e MAC parameter settings that can be used to control the bias that 802.11 applies towards unicast traffic (at the expense of multicast traffic). 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] P. Ross, "Top 11 Technologies of the Decade", IEEE Spectrum, January 2011.

[2] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, J. Zahorjan. "Measurement-based Characterization of 802.11 in a Hotspot Setting." http://portal.acm.org/citation.cfm?id=1080150

[3] R. Raghavendra, E. Belding, K. Papagiannaki, K. Almeroth. "Understanding Handoffs in Large IEEE 802.11 Wireless Networks." http://www.imconf.net/imc-2007/papers/imc192.pdf

[4] Jardosh, K. Ramachandran, K. Almeroth, E. Belding-Royer. "Understanding Congestion in IEEE 802.11 Wireless Networks." http://www.usenix.org/events/imc05/tech/full_papers/jardosh/jardosh_new.pdf

[5] M. Balazinska, P. Castro. "Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network." http://portal.acm.org/citation.cfm?id=1066127

[6] Balachandran, G. Voelker, P. Bahl, P. Rangan. "Characterizing User Behavior and Network Performance in a Public Wireless LAN." http://sysnet.ucsd.edu/pawn/papers/wireless_sig.pdf

[7] T. Henderson, D. Kotz, I. Abyzov. "The Changing Usage of a Mature Campus-wide Wireless Network." http://portal.acm.org/citation.cfm?id=1023720.1023739

[8] Jardosh, K. Ramachandran, K. Almeroth, E. Belding-Royer. "Understanding Link-Layer Behavior in Highly Congested IEEE 802.11 Wireless Networks." http://portal.acm.org/citation.cfm?id=1080151

[9] M. Juang, K. Wang J.Martin, "A Measurement Study on Link Capacity of a High Stress IEEE 802.11b/g Network", Proceedings of the 2008 IEEE International Conference on Computer Communications and Networks (ICCCN 2008), (St. Thomas, Virgin Islands, August 2008).
[10] 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.
[11] 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, andM. Fujise, Eds., CRC Press, Boca Raton, Fla, USA, 2006.
[12] M.-T. Sun, L. Huang, S. Wang, A. Arora, 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.
[13] S. Kasera, J. Kuri, "Reliable Multicast in Multi-access Wireless LANS", Proceedings of INFOCOM99, March, 1999.
[14] C. Shannon, "A Mathematical Theory of Communication", Bell System Technical Journal, Vol. 27, pp. 379-423, 1948.
[15] T. Rapport, Wireless Communications: Principles and Practice, Second Edition, Prentice Hall.
[16] Rizzo, L., "Effective Erasure Codes for Reliable Computer Communications Protocols", ACM Computer Communications Review, Vol. 27, no., 2, April 1997, pp. 24-36.
[17] Luby, M., Vicisano, L., Gemmell, J., Rizzo, L., Handley, M., and J. Crowcroft, "The Use of ForwardError Correction (FEC) in Reliable Multicast",RFC 3453, December 2002.
[18] B. Adamson, C. Borman, M. Handley, J. Macker, "NACK-Oriented Reliable Multicast (NORM) Transport Protocol, IETF Request for Comments 5740, November 2009.
[19] D. MacKay, "Fountain Codes", Proceedings of IEEE Communications, vol, 152, no 6, December 2005, pp 1062-1078.
[20] A. Shokrollahi, "Raptor Codes", IEEE Transactions InformationTheory, vol. 52,no. 6, June 2006, pp. 2551-2567.
[21] Luby, M., Stockhammer, T., Watson, M., "Application Layer FEC in IPTV Services", IEEE Communications Magazine, May 2008.
[22] Networking Simulator 2 (ns2), available at http://www.isi.edu/nsnam/ns/
[23] R. Punnoose, P. Nikitin, D. Stancil, "Efficient Simulation of Ricean Fading within a Packet Simulator", Proceedings of the IEEE Vehicular Technology Conference, 2000.
[24] Cisco Aironet 1250 Series Access Point Data Sheet, available at http://www.cisco.com/en/US/prod/collateral/wireless/ps5678/ps6973/ps8382/product_data_sheet0900aecd806b7c5c.pdf
[25] Nafaa A., Taleb T., Murphy L., "Forward Error Correction Strategies for media Streaming Over Wireless Networks", IEEE Communications Magazine, December 2007.
[26] S. Winkler, P. Mohandas, "The Evolution of Video Quality Measurement: From PSNR to Hybrid Metrics", IEEE Transactions on Broadcasting, vol 54, no. 3, September 2008.
[27] J Xia, Y. Shi, K. Teunissen, I. Heynderickx, "Perceivable Artifacts in Compressed Video and their Relation to Video Quality", Signal Processing: Image Communications, Vol 24, No. 7, August 2009, pp. 548-556.
[28] N. Degrande, K. Laevens, D. Vleeschauwer, R. Sharpe, "Increasing the User Perceived Quality for IPTV Services", IEEE Communications Magazine, February 2008.
[29] Gilbert, E.N., "Capacity of a Burst-Noise Channel", Bell System Technical Journal 39(1960), pp. 1253-1265.
[30] Elliot, E.O., "Estimates of Error Rates for Codes on Burst-Noise Channels", Bell System Technical Journal 42 (1963), pp. 1977- 1997.
[31] Hablinger, G, Hohlfeld O, "The Gilbert-Elliott Model for Packet Loss in Real Time Services on the Internet", International Workshop on the Quality of Service, IWQoS2008, June 2008, pp. 239-248.
[32] 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 1

Experiment	FEC parameters	Flows under observation	Background traffic	Loss process
1	Variable block size (#1 below), fixed redundancy 0.20	5 downstream CBR flows (768Kbps, packet size 1400)	none	Bernoulli, vary loss (#2 below)
2	Variable block size (#1 below), fixed redundancy 0.20	5 downstream CBR flows (768Kbps, packet size 1400)	none	Gilbert-Elliot, vary loss (#2 below) by varying state duration (#3 below)
3	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)
4	Variable block size (#1 below), fixed redundancy 0.20	1 multicast stream (1 station) (768Kbps, packet size 1400)	none	Gilbert-Elliot, vary loss (#2 below) by varying state duration (#3 below)
5	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)
6	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

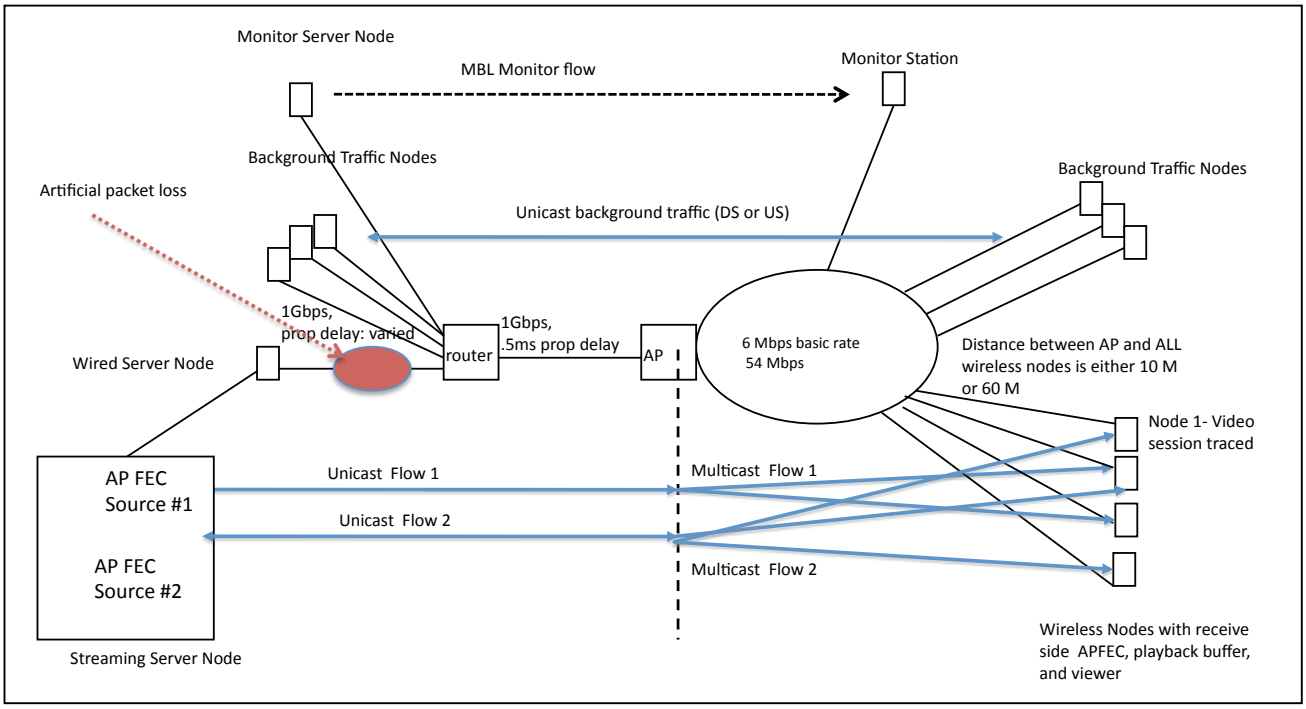


Figure 2. Simulation Model

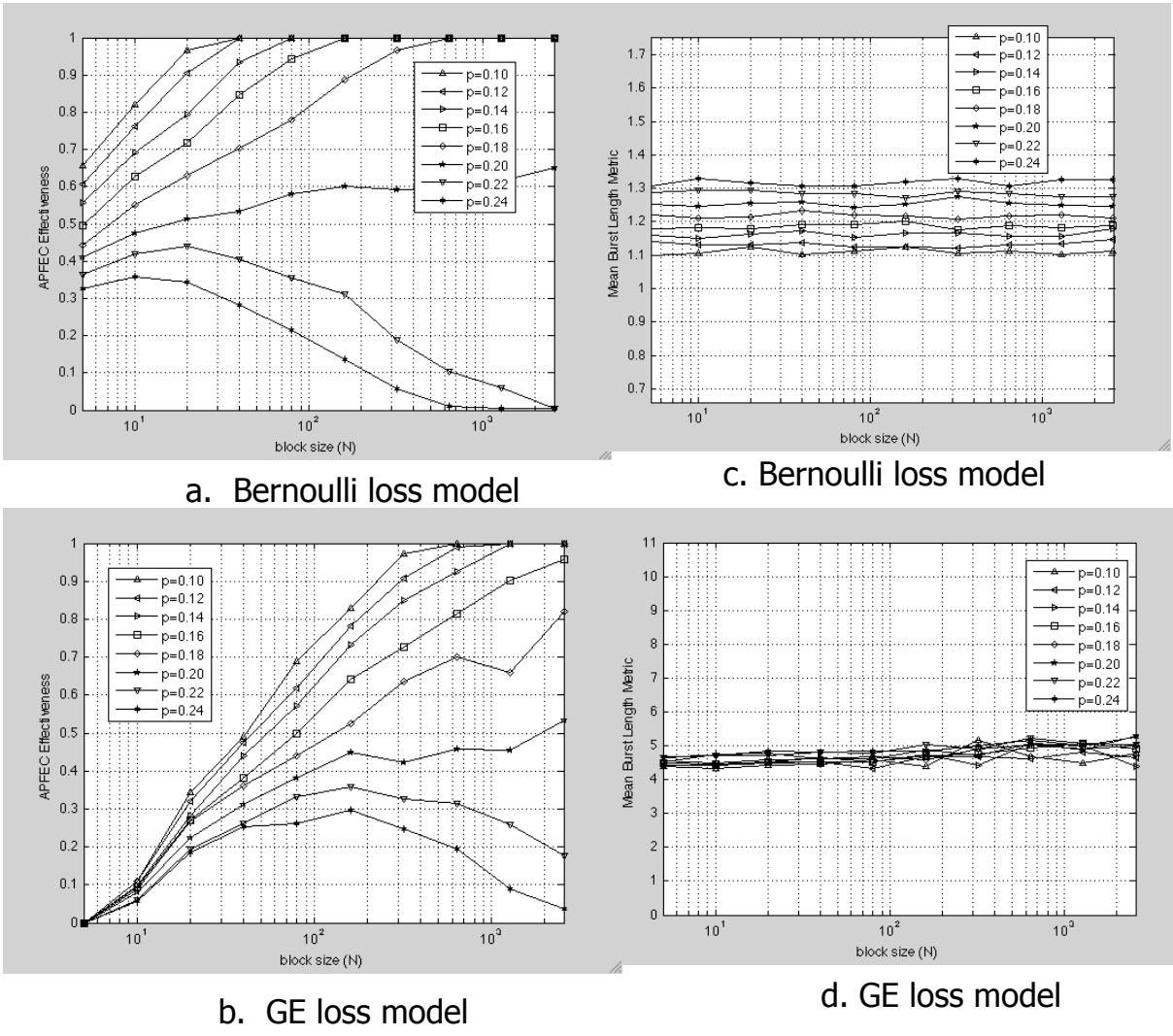
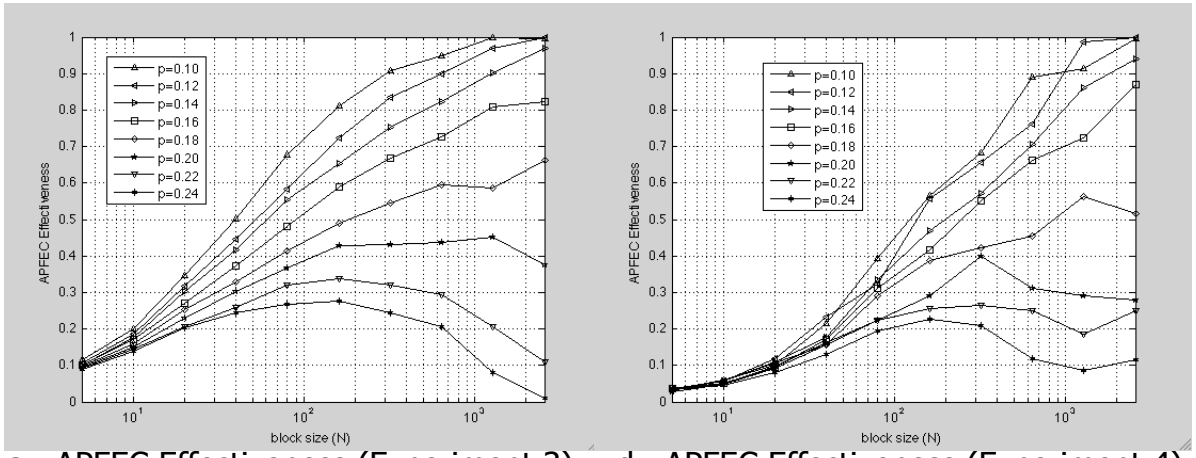
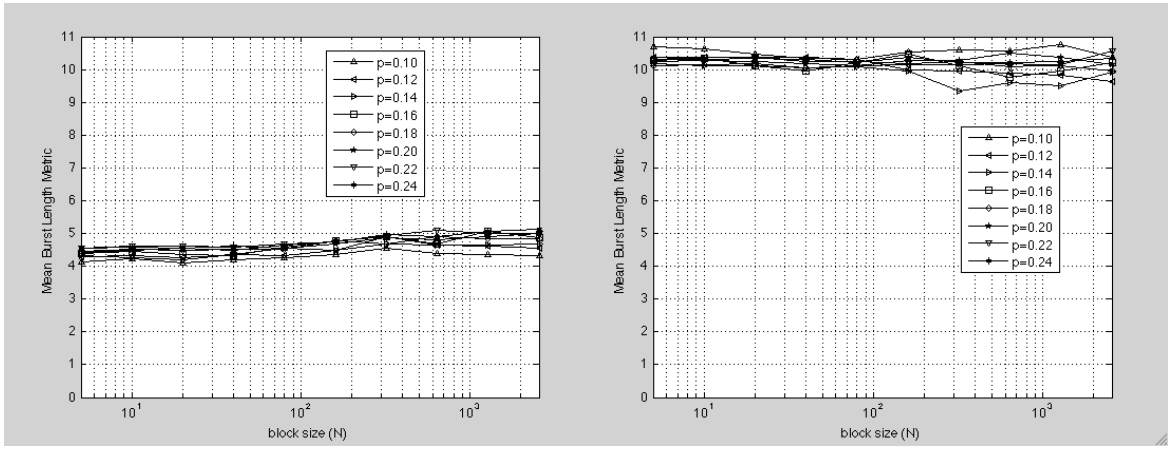


Figure 3: APFEC Effectiveness and MBL Metric for Experiments 1 and 2 (Redundancy Fixed at 0.20, EXP311, 321)



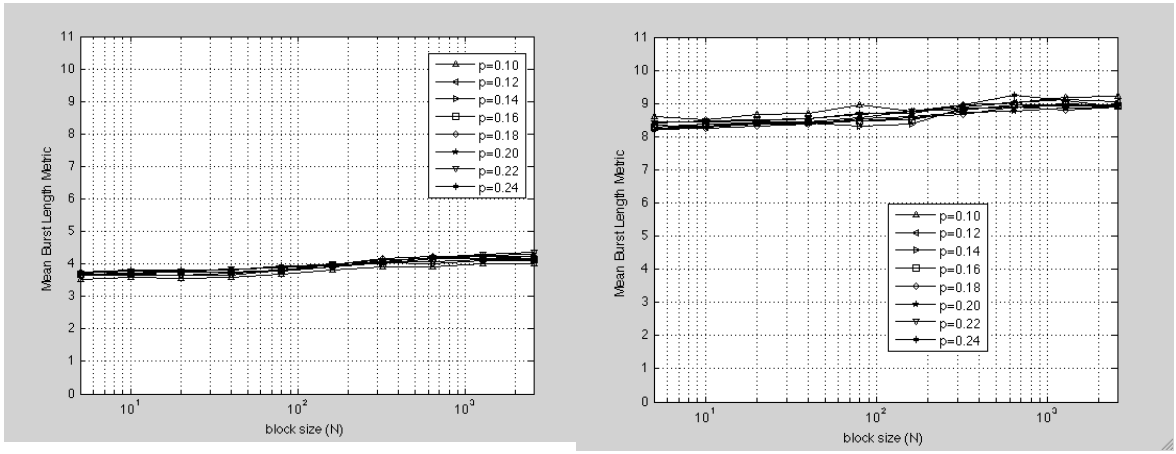
a. APFEC Effectiveness (Experiment 3)

d. APFEC Effectiveness (Experiment 4)



b. Multicast MBL Metric (Experiment 3)

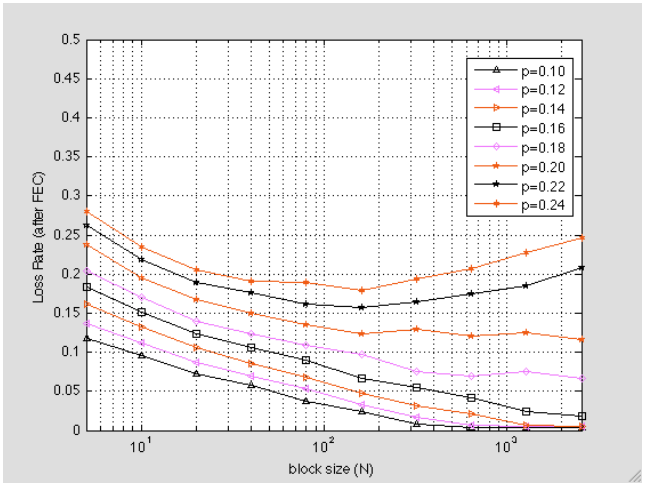
e. Multicast MBL Metric (Experiment 4)



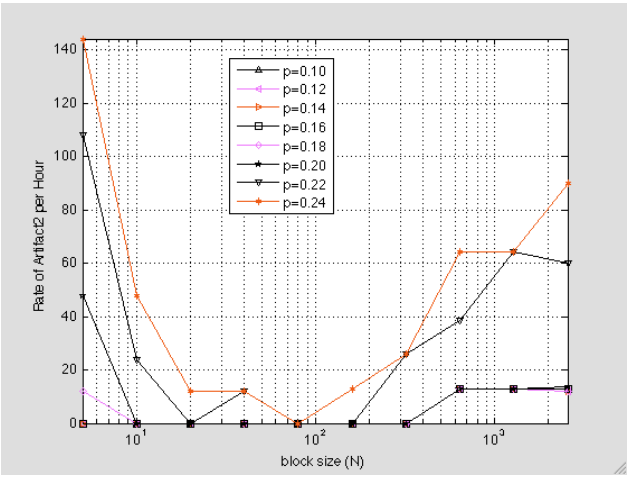
c. Unicast MBL Metric (Experiment 3)

e. Unicast MBL Metric (Experiment 4)

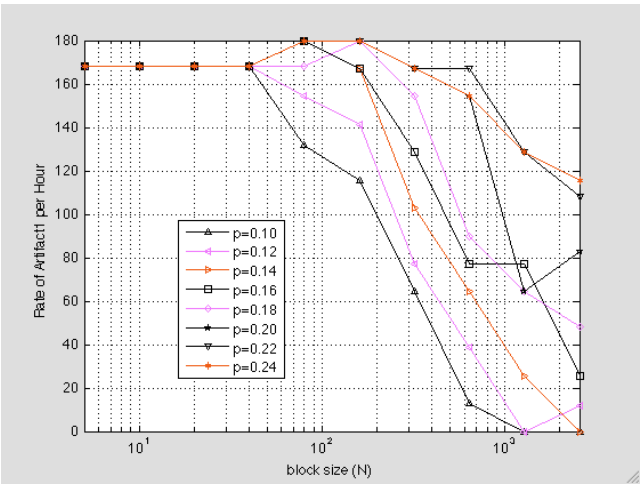
Figure 4: APFEC Effectiveness and MBL Metric Observed by One Multicast and One Unicast flow for Experiments 3 and 4 (EXP351 unmodified and modified)



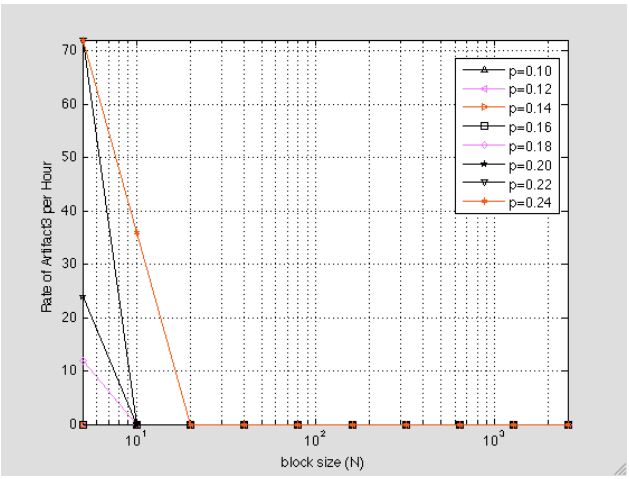
a. Minimum Throughput Threshold Artifact



c. Minimum Throughput Threshold Artifact

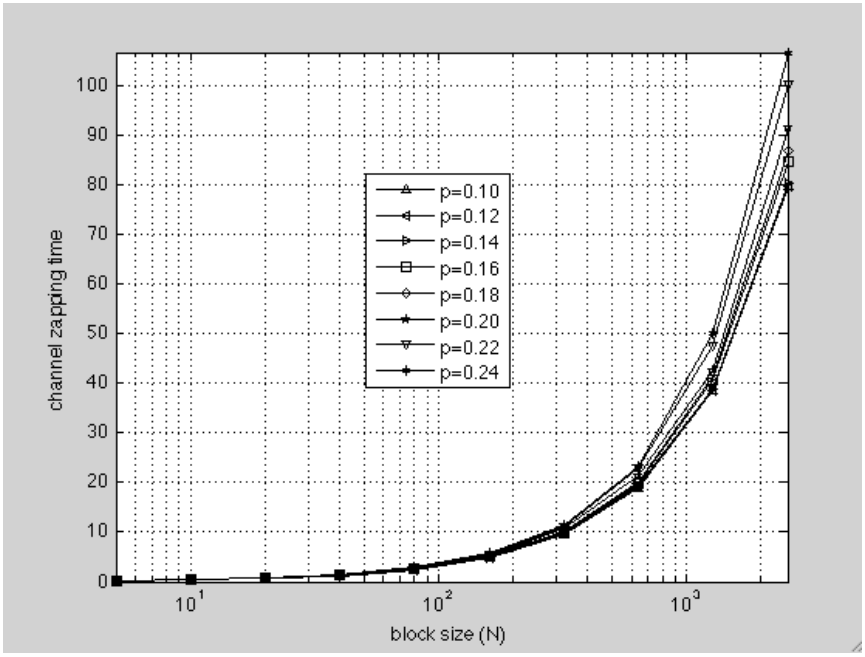


b. Loss Tolerance Artifact

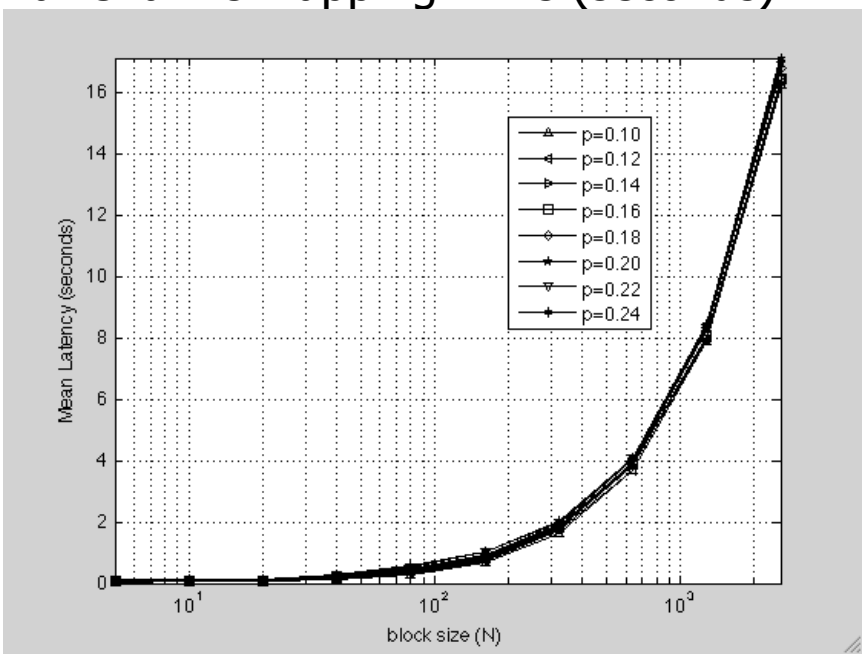


d. Playback Buffer Depletion Artifact

Figure 5: Artifact Metrics for Experiment 3 (EXP351)



a. Channel Zapping Time (seconds)



b. Average Packet Latency (seconds)

Figure 6: Playback Buffer Fill Time (Channel Zapping Time) and Average Packet Latency for Experiment 3 (EXP351)

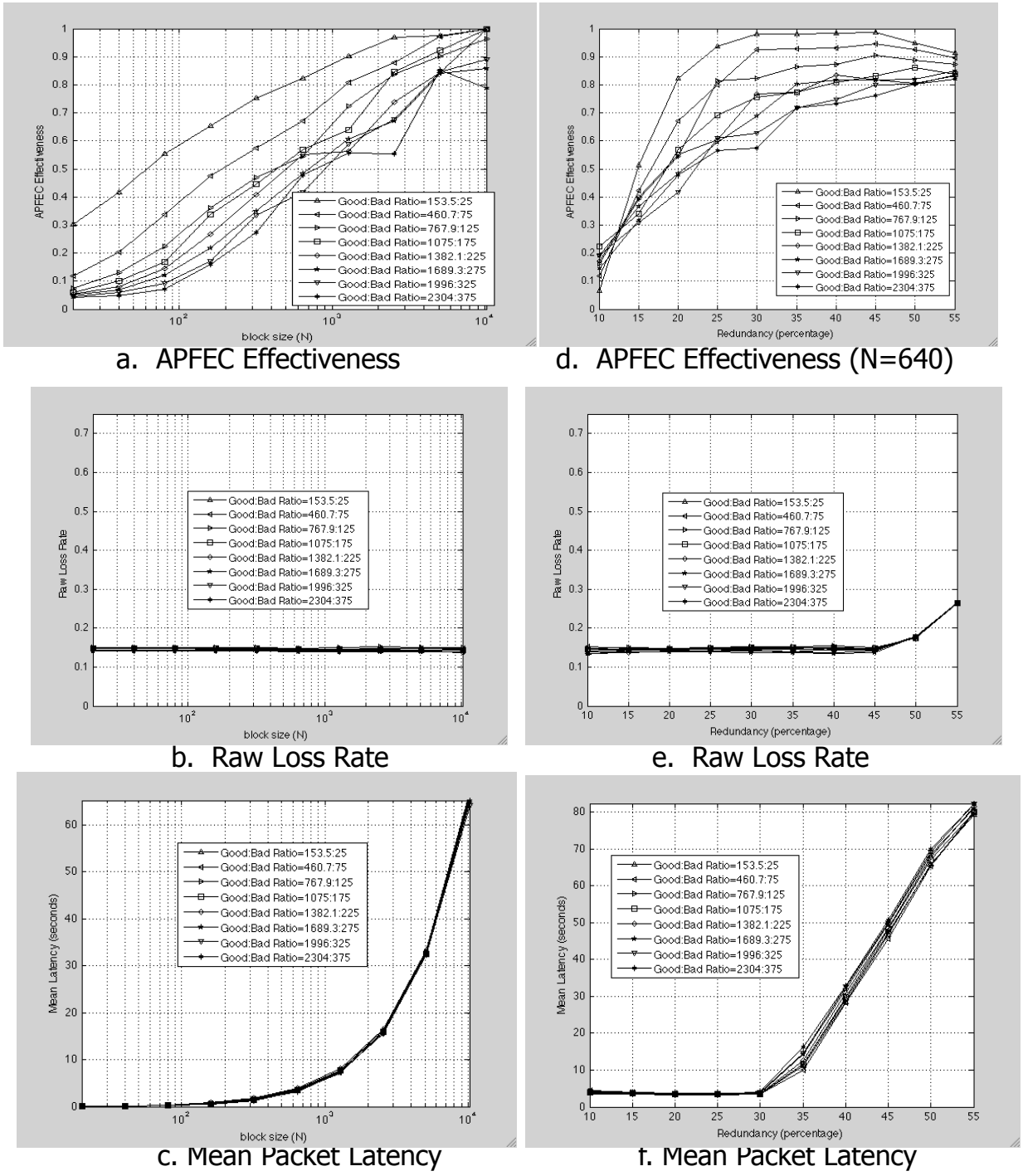
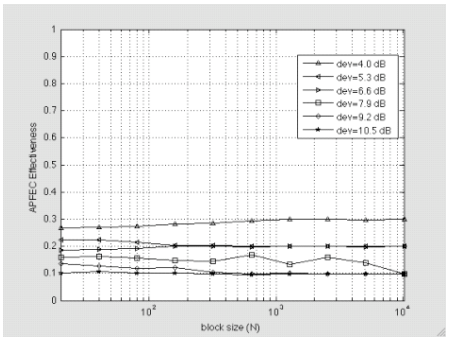
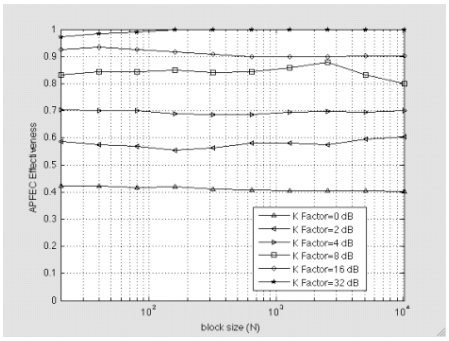


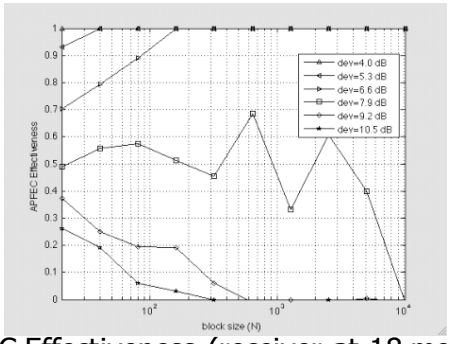
Figure 7: APFEC Effectiveness and Effect Loss Rate for Experiments 5 and 6 (EXP371 and EXP391)



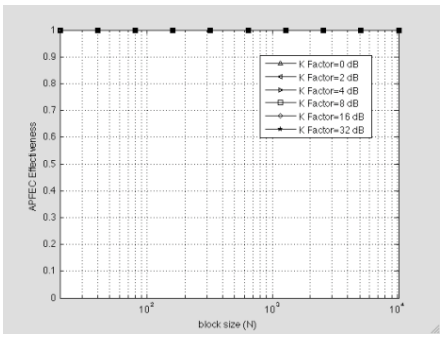
a. APFEC Effectiveness (all receivers)



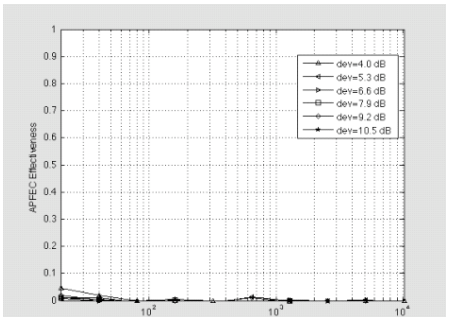
d. APFEC Effectiveness (all receivers)



b. APFEC Effectiveness (receiver at 18 meters)



e. APFEC Effectiveness (receiver at 18 meters)



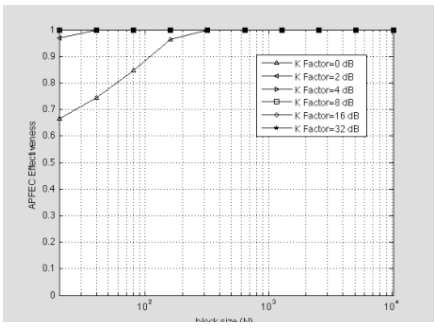
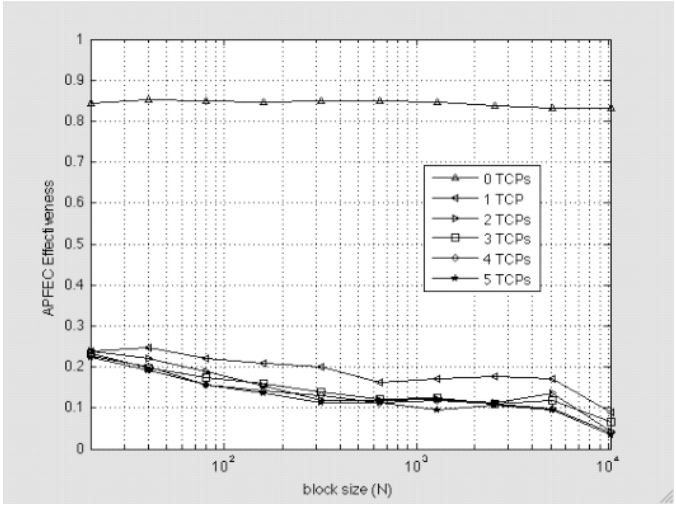
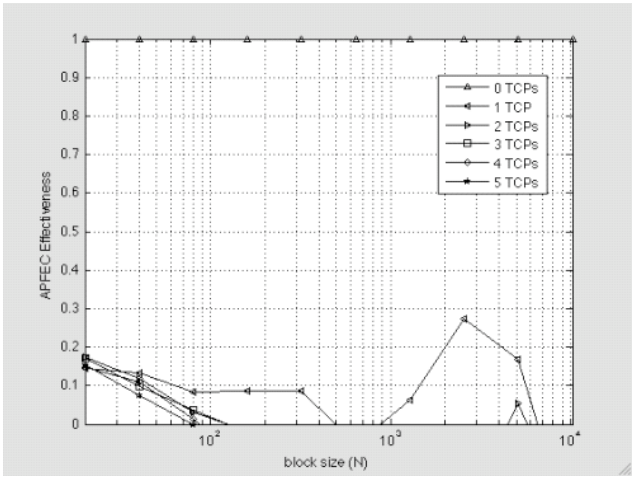


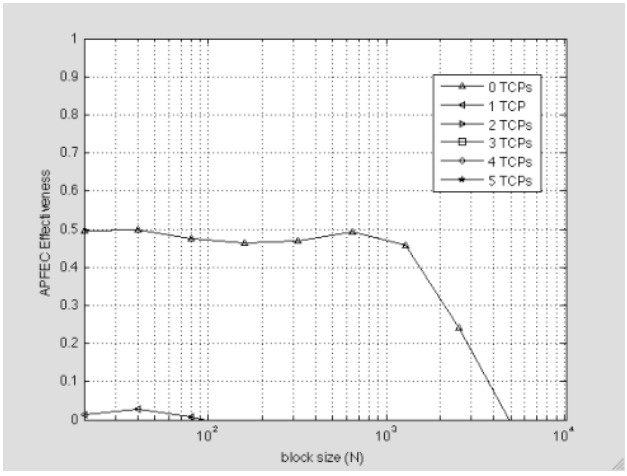
Figure 8. APFEC Subject to Shadowing and Ricean Propagation Models as Observed at 18 and 48 Meters (EXP411 and EXP451)



a. APFEC Effectiveness (all receivers)



b. APFEC Effectiveness (receiver at 18 meters)



c. APFEC Effectiveness (receiver at 48 meters)

Figure 9. APFEC Subject to Competing TCP Traffic with a Ricean Channel Model as Observed at 18 and 48 Meters (K-factor Set to 6)

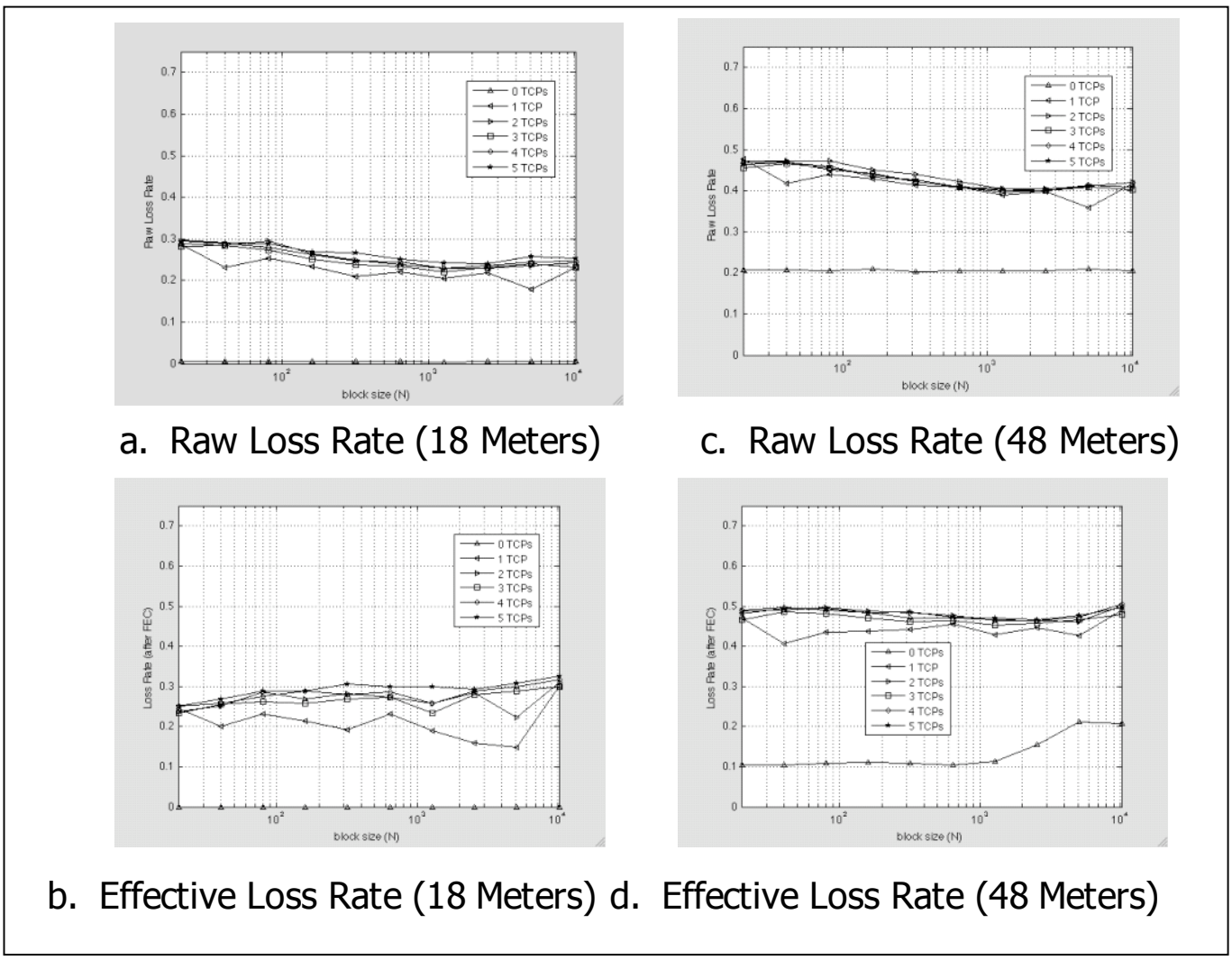


Figure 10. Observed Loss at 18 and 48 Meters (K-factor 6)

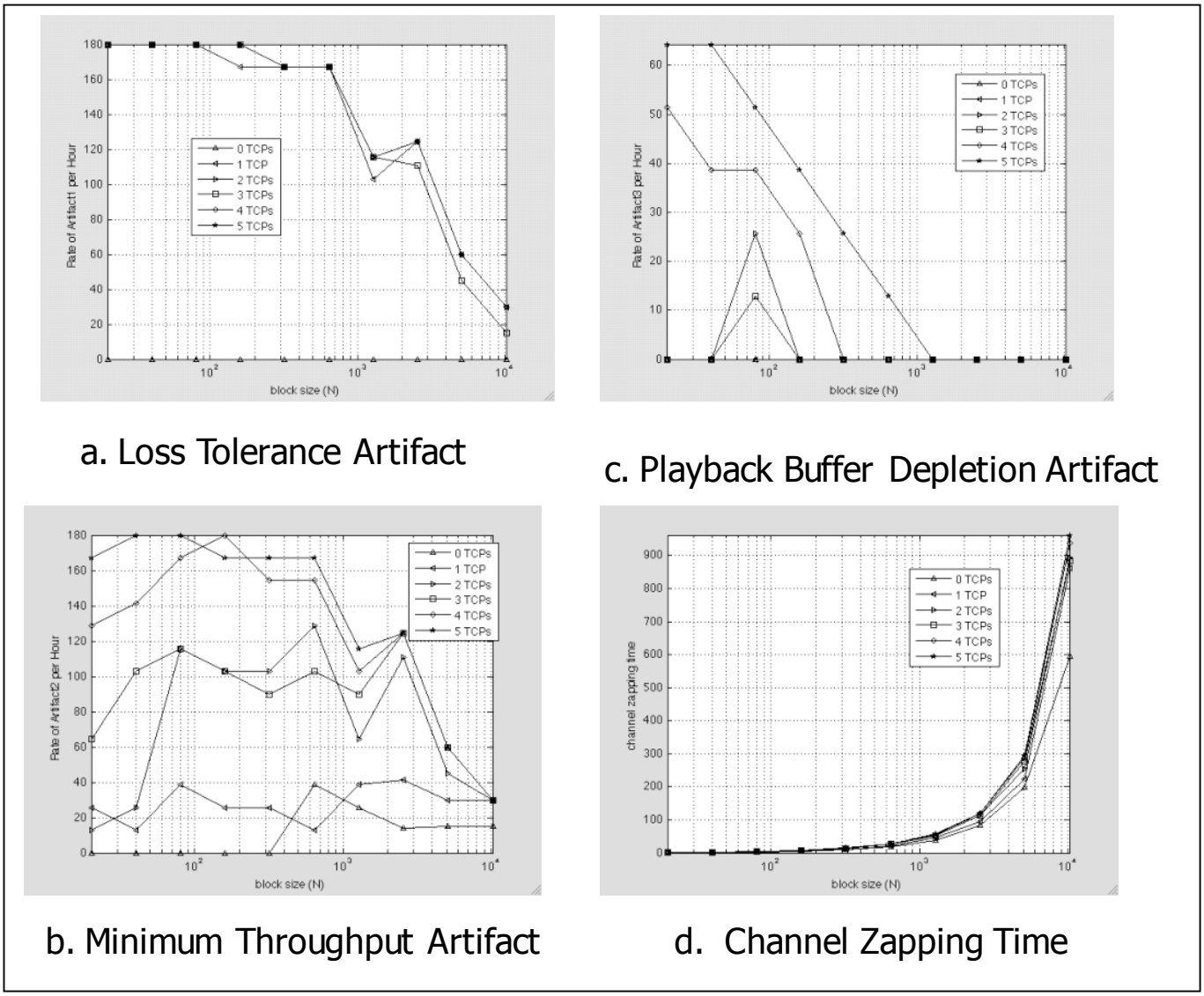


Figure 11. Artifacts when Subject to Competing TCP Traffic in Ricean Conditions (K-factor=6, station at 18 meters)