An Adaptive FEC Algorithm for Mobile Wireless Networks

Jong-Suk Ahn† and John Heidmann‡
†Computer Engineering Dep. DongGuk University Jung-Gu Pil-Dong 3-Ga 26 Seoul, Korea
‡USC/ISI 4676 Admiralty Way Marina del Rey, CA 90292, USA
USC/ISI-TR-555 3/25/02

***Abstract*-- Wireless mobile networks tend to drop a large portion of packets due to propagation errors. To improve reliability over noisy wireless channels, wireless networks can employ forward error correction (FEC) techniques. Static FEC algorithms, however, can degrade the performance by poorly matching their overhead to the degree of the underlying channel error, especially when the channel path loss rate fluctuates widely. This paper investigates the benefits of an adaptable FEC mechanism for wireless networks with severe packet loss. We show that our adaptive FEC technique improves the performance by dynamically tuning FEC strength to the current amount of wireless channel loss. We quantify these benefits through a hybrid simulation integrating packet-level simulation with bit-level details and validate the simulation model through experimentation.**

Index terms—Adaptive FEC, Wireless Mobile Networks

I. INTRODUCTION

In spite of the recent wide deployment of wireless mobile networks, wireless networks suffer the poor performance comparing to wired networks due to the heavy propagation error. Their average bit error rate (BER) is known to vary from 10^{-6} to 10^{-1}, implying that in the worst case, most packets in a wireless network tend to be corrupted. At short time-scale (less than 100ms), the instant BER also fluctuates widely and rapidly, making it difficult for a fixed FEC algorithm to match the wireless network's needs. This problem is particularly acute in sensor networks with their low-power radios. Recent live experiments over sensor networks [1] reported that the absence of proper preparations against this severe and variable propagation error incurred an enormous packet loss rate more than 50% due to corruption.

The cause of wireless error typically is explained by two effects, large-scale and small-scale fading effects each of which roughly corresponds to long-time and short-time scales effects in terms of time [2]. The large-scale fading model predicts that the average signal strength attenuates as a function of the distance between a transmitter and its receiver (T-R) powered by an exponent ranging between 2 and 6. Note that BER is inversely proportional to the signal-to noise ratio (SNR). While this model forecasts the average signal power at a certain T-R distance, the small-scale fading describes wide deviations of the signal power from the average value by a slight change of T-R distance.

The short-term fading effect is mainly attributed to two main physical phenomena such as multipath interference and Doppler Effect. The multipath interference indicates that signal waves arrive in phase or out of phase at the receiver due to their different travel distances even though they radiate at the same time. Doppler Effect means that the frequency of signal waves changes at the receiver differently from that at the transmitter due to either the mobility of the transmitter or the receiver. The slight change on the communication environments differently superimposes signal waves, leading to a large difference on the received signal power. Together, these two models explain why the signal power will widely fluctuate from time to time while the average smoothly degrades as T-R distance increases.

Medium access control (MAC) protocols use techniques to either avoid or correct propagation errors. For avoidance, the physical layer adopts modulation and multiplexing techniques while the link layer employs interleaving schemes. Modulation enhances the error resistance by separating physical signal representations mapped to each logical symbol as apart as possible. Two multiplexing techniques, direct sequence spread spectrum (DS-SS) and frequency hopping spread spectrums (FH-SS) prevent long-lasting bursty noise signals from tainting a whole bit or a entire frame by broadening the signal's frequency spectrum or switching the carrier frequency in a random sequence. Interleaving also keeps multiple consecutive bit-errors from incurably concentrating at a single frame by alternatively transmitting segments split from different frames. For error correction, the link layer uses Automatic ReQuest (ARQ) and FEC to

---

This work was partially supported by NSF as part of the CONSER project, contract number ANI-9986208. This work is supported by the research program of Dongguk University.

remedy bit-errors in a reactive and proactive way respectively.

Most wireless networks use a combination of these techniques at different level of the protocol stack. For channels with heavy losses, FEC is often employed even though its redundant coding reduces efficiency. To allow trade-off between efficiency and reliability, they furnish a fixed set of FEC levels to choose from depending on the underlying channel quality and various applications' demands. GSM [3], for example, provides four channel codes each of which occupies 0, 25%, 33% and 50% of the total frame with their parity check bits respectively.

To improve the static FEC performance over constantly changing wireless channel status, this paper proposes an FEC-level Adaptation (FECA) mechanism which can adjust the amount of parity check bits based on the channel status. FECA takes type-I hybrid ARQ approach [4] which retransmits data with check bits rather than type-II hybrid ARQ approach which incrementally resends only check-bits. Type-II hybrid ARQ assumes that the previous data packet is successfully buffered at the receiver. The evaluation of our wireless networks with low-power radios shows that receivers, however, can't frequently receive even tainted packets due to the corruption of their preamble.

FECA adapts to the channel BER fluctuations without explicit feedback information from the receivers. To properly adjust the FEC strength to the channel state transitions, FECA expedites upward and downward transitions to the higher and lower level of FEC strength. Each transition is activated by either a packet loss or the timeout of an exponential tunable back-off timer. The live measurements over our wireless networks exhibit enough long-lasting positive correlation of BER for FECA to be stabilized even by this incremental adaptation. Through a hybrid simulation integrating the packet level abstraction with bit-level details, finally we confirm that FECA performs better than static FEC algorithms over wireless mobile channels with full of relative short-term variations. The contribution of this paper is to show the benefits of an adaptable FEC algorithm and verify the model through experimentation.

This paper is organized as follows. Section 2 lists some related works and Section 3 examines the adaptive algorithm's applicability over wireless mobile networks. Section 4 describes the design issues of FECA. Section 5 elaborates two possible approaches integrating bit-level details with a packet simulator to evaluate FECA throughput. Section 6 reveals FECA behaviors under various error distributions by hybrid simulation. Finally Section 7 summarizes experiment results and presents our future research list.

II. Related Work

FECA combines two techniques: FEC and adaptive link-layer algorithms. We review these areas next.

FEC algorithms are employed over either the link layer of wireless networks or the application layer of the Internet to resist against packet losses due to congestion and bit corruption respectively. The real-time applications tend to employ packet-level FEC algorithms which attach some of previously sent data at the current packet to compensate previously lost packets. Recently Bolot[5] evolves the FEC adaptation dynamically by coupling FEC with a rate control algorithm. He tries to appropriately determine the amount of redundancy based on the average loss rate measured at the receiver.

Bit-level FEC algorithms for the link layer are also extensively investigated to overcome the heavy propagation error. Based on what is retransmitted, the link-level FEC is grouped into two: type-I and type-II hybrid ARQ [4]. While type-II retransmits only the parity-check bits incrementally until the corrupted data packet is corrected, type-I resends FEC code along with the data. Type-II becomes more efficient under the low channel error rate since type-II only retransmits the check bits under the assumption that the receiver holds the previous data packet even if it is corrupted. When wireless networks are heavily lossy so that packets can't be recognized due to the corrupted preamble, type-I hybrid ARQ tends to be effective. It is because the type-II hybrid ARQ algorithms can't recover the contaminated packet if any of all the retransmitted parity packets including the data packet is missed. Furthermore, the implementation of type-II algorithms would be more complex than type-I.

In contrast to the application-level FEC, link-level dynamic FEC algorithms need to deal with two additional problems; packet size and the duration of wireless channel status. The packet size significantly affects the corruption rate while it rarely influences the packet loss rate due to congestion. The link-level FEC should not increase the packet size when strengthening the resistance to the propagation error. Further, the wireless channel behavior changes rapidly comparing to the long lasting congestion over the Internet like few hours. To quickly adapt to the fast changing channel, the link-level FEC needs to quickly adapt to the currently measured channel status rather than some average value computed over a long time interval.

For dynamic adaptability, several researchers [6][7][8]propose to dynamically change the link-level algorithm's parameters such as transmission speed of ARQ, Maximum Transmission Unit (MTU), and modulation schemes based on the average measured packet loss rate or SNR evaluated at the receiver side. In detail, Holland [8] proposes that senders select a suitable modulation method among the four possible ones according to SNR reported from the receiver side by acknowledgement packets. He also evaluated the channel's dynamicity, namely correlation behaviors by analytically computing how long a given measured status of the wireless channel will persist.

FECA is a dynamic type-I hybrid ARQ algorithm appropriately adapting the FEC strength to the constantly changing wireless channel status. Differently from the above dynamic related works, FECA works without explicit channel information and hardware support. Holland needs the SNR feedback and the various modulations supported from the transmission hardware while the other two schemes need to measure the average packet loss rate. FECA, finally, is cooperative with the above all algorithms in that it can be employed together with the others. For example, wireless node changes the modulation scheme while adopting the different strength of FEC. Note that wireless networks already have redundantly spread several error correction and avoidance algorithms over their protocol stacks. Based on the layer that they are located at and their techniques, they altogether try to recover different time-scale errors and different degree of errors without much interference among them.

III. Adaptation Applicability Over Wireless Mobile Networks

The design of a link-level adaptive algorithm like FECA depends on the underlying BER distribution, precisely its amplitude and duration statistics of BER fluctuations over wireless mobile networks. If the BER distribution rarely exhibits a positive correlation, it would be difficult to deploy adaptive algorithms. Otherwise, one must evaluate the time scale of correlations to determine how to adapt the algorithm's tunable parameters appropriately. This section estimates the time scale of correlations by both theoretical analysis and live experiments over wireless networks.

A. Theoretical Analysis

In this section, we approximate theoretically how fast BER changes as a function of T-R distance due to either the large-scale fading or the small-scale fading. For the BER change due to the large-scale fading, we need to consider two theoretical relations which relate two metrics such as (BER, SNR) and (SNR, T-R distance) respectively. At first, according to the analytical equation [9], the probability of BER is inversely proportional to a parameter called a ratio of signal energy per bit to noise power density per Hertz (E_b/N_o) which is basically SNR divided by the date rate. When the signal power becomes weaker by 10 times, the BER with no line-of-sight (LOS) path roughly increases approximately by 10 times while the BER of LOS grows about 10 times faster than the BER with no LOS path.

Figure 1 shows the relation between the average signal power and T-R distance based on the log-normal shadowing model, one of the most popular large-scale fading models. For Figure 1, we assume 914 Mhz Lucent LAN environments with the 1-meter close-in reference distance, 0.2818 Watt transmitting power and $1.559 * 10^{-11}$ Watt receiving threshold, which are represented by two horizontal lines in Figure 1. The receiving threshold is a threshold to tell the signal from the noise. Radio transceivers [10] can determine this threshold appropriately according to the communication environment before operation. The reference distance is the distance before which transmitting power degrades as predicted in the free space propagation model, meaning that the power decays as the square of the T-R distance.

In Figure 1, the four lines with exponents from 3 to 6 represent wireless networks with no LOS path such as office environments with a lot of obstacles whereas the other one represents indoor wireless communications with LOS path or free space. The four lines steeply attenuate around 10 times over 10-meter movement while the other line with exponent 2 depicts slower slope degrading the same degree of SNR changes over 100 meters.

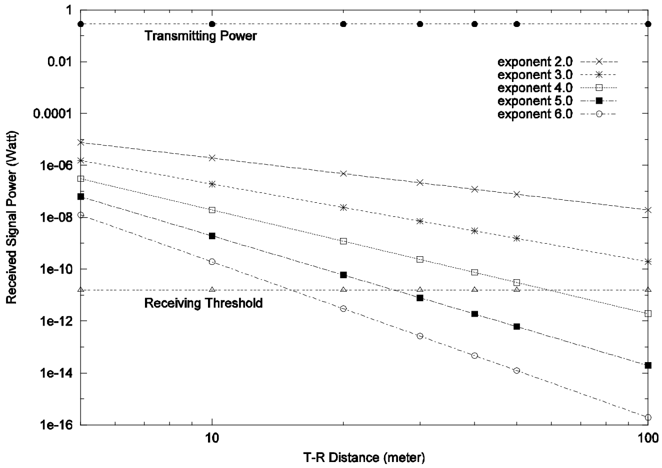

Figure 1 Average Power Degradation Due to Large-Scale Fading

Based on these two analytical analyses, we can summarize that SNR attenuates more rapidly in environments with no LOS path than those with LOS while BER deteriorates more quickly in environments with LOS path than those with no LOS path. From these two observations, we roughly say that the average BER rarely changes within a few seconds in most wireless environments such as office and free space. For example, when a receiver walks away at the speed of less than 0.5 m/s (meter per second) in office with a lot of obstacles, the signal power degrades 10 times every 10 meters and accordingly the BER increases 10 times. Considering that the average BER varies 10 times during 20 seconds and the packet propagation delay is quite short (a few milliseconds), we argue that an adaptive FEC algorithm can adjust its strength to the optimal value over many round trip times (RTT).

To see the BER variation by the small-scale fading, we plot the average small-scale Rayleigh fading duration in Figure 2 when a receiver node moves away at two different speeds, 0.1 and 10 m/s at three different locations, 5, 10, and 20 meters apart from the transmitter in the wireless LAN described for Figure 1. For the maximum power of the six lines from which the signal power falls, we compute the average power by adopting the log-normal shadowing model with the exponent 3. The horizontal and vertical lines of Figure 2 correspond to the receiving threshold and 1 ms RTT of wireless LANs that we exemplify as an applicable network.

Figure 2 indicates that the spectrum of the average duration spans from a few microseconds to a few seconds depending on the speed and the distance from the sender. The comparison of the six lines shows that as a receiver becomes apart from the transmitter, the duration of heavy BER lasts longer. Finally the comparison also indicates that the average duration decreases in proportion to the mobility speed.

When considering feedback delays for monitoring the channel status and the correlation duration shown in Figure 2, the usefulness of FECA would be limited in that it may not adapt to a whole range of short-term fluctuations by the small-scale fading. For the wireless LAN with 1 ms RTT, the shortest fading duration that FECA would adapt to would be a few tens of milliseconds. In summary, from these theoretical analyses we believe that FECA improves the performance in most wireless networks by tracking the average BER behaviors while it accomplishes better in some networks whose average short-term fluctuation's width would be at least larger than a few tens of milliseconds.

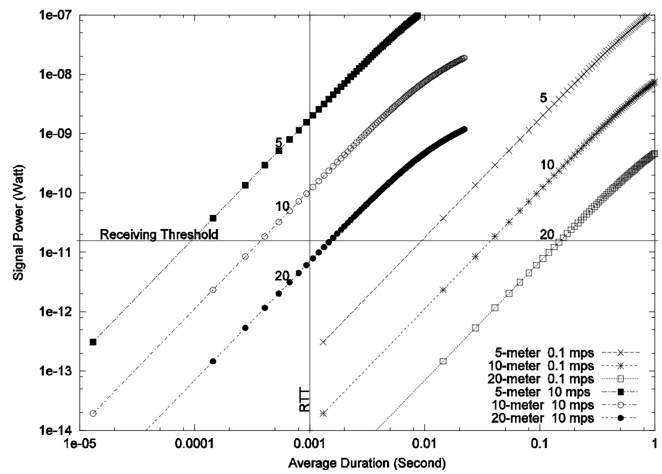

Figure 2 Average Fading Duration Due to Small-Scale Fading

One example wireless network for FECA would be one in which mobile nodes move slowly among a lot of obstacles, causing the short-term BER fluctuation ranging from tens or hundred milliseconds. In detail, it would be the remote area exploration in which several robots moves slowly like less than 1 meter-per-second while constantly communicating through various geographical obstacles such as rocks, valleys, and etc. Another applicable network of FECA would be the one at the other extreme in which mobile nodes moves very fast like 100 km/h in which FECA will follow the large-scale fading continuing longer than a few minutes. In this case, the small-scale fading would cause microsecond variances contained in a single frame so that the large-scale fading would be the only one changing BER over a stream of frames. Finally when the average duration of BER is in the middle between a few hundred microseconds and a few milliseconds, FECA only suffers the overhead of adaptation since it can't monitor the channel status fast enough to follow this degree of fluctuations.

B. Measurements of wireless channel behaviors in a sensor network

Section III.A suggested that FECA would be appropriate for some wireless networks based on analysis of common propagation models. However, these models are quite general and built for relatively high-power radios such as cell phones and 802.11 data networks. Since we are interested in designing an adaptive FEC algorithm for wireless sensor networks with very low-power radios, this section presents experimental measurements that evaluate radio interference over Motes, 8-bit sensor nodes with low-power radios operating at around 900 MHz[1].

To see the adaptability of FECA to the sensor network, we measured two metrics: the degree of correlation variation and the corruption rate change as the packet size. The correlation implies the likelihood that when a packet is corrupted, a certain number of following packets are consecutively contaminated at the same degree as that packet. The corruption rate means the ratio of the number of corrupted bytes to the number of correctible bytes by the added check-byte. These two metrics indicate how long the channel state measured at a given time persists and whether the addition of check bytes is worth since the increase of packet size also incurs more corruption.

Figure 3 plots the correlation function based on packet traces collected over a real sensor network where one Mote constantly transmits 35-byte (4-byte header and 31-byte payload) packets to a receiver at the speed of 5.6kps without any acknowledgement. We made these measurements in the hallway of ISI where during the working hours more than 90% of received packets are severely corrupted due to the profound interference by its cordless phone system sharing the same 850-950MHz ISM (Industrial, Scientific and Medical) band and people's movements. Also more than 10% packets are not recognized due to the corrupted preamble. For three lines on the graph, we collected data at three different T-R distances which are 1, 3, and 5 meters approximately and each point on each line averages three 1-hour measurements conducted over different days.

For each number n at x-axis of Figure 3, we divide the sequence of m packets into n-consecutive groups where m is the total number of packets we collected in the experiment. The total number of n-consecutive packet groups out of m packets is $m - n + 1$. Figure 3 plots the ratio of the number of n-consecutive packet groups belonging to the same FEC level to $m - n + 1$. To see how fast FEC changes for curing all the corrupted packets, we mapped the number of contaminated bytes in each packet to a certain FEC level. For this translation, we assume five FEC levels each of which corrects 5, 10, 15, 20, and 25 corrupted bytes based on the observation that the maximum number of corrupted bytes in this experiment is up to 25 bytes. The reason for counting corrupted bytes rather than computing BER is that in most FEC algorithms [11], the basic data unit to recover is a symbol whose size is appropriately determined based on the channel characteristics. Note that when the symbol size is x, the maximum size of packets that we can correct with attached check words is 2^x. In this experiment, we assume the symbol size as 8-bit and fix the packet size to 35-byte.

Figure 3 confirms a strong positive correlation of BER among back-to-back packets. Figure 3, for example, predicts that the next packet suffers the same degree of corruption with probability 0.6 at the 1 meter T-R distance. These experimental data in Figure 3 agree with the Holland's observation [8] that the transmitter achieves better performance by selecting the appropriate modulation method for the next packet based on the explicit feedback about the previous packet's degree of corruption.

The comparison of the three lines, however, indicates that the correlation becomes weaker as the T-R separation grows. It is because that as the T-R separation becomes larger, the start-of-packet symbol becomes more corrupted so that the receiver receives fewer packets. In other words, at the larger T-R distance, the next arrived packet at the receiver is not likely to be the next packet sent at the transmitter. The measured average inter-packet gap is 134ms, 200ms, and 462ms in the increasing order of T-R distance. Based on the probability distribution of Figure 3 and these inter-packet gaps, the average correlation duration times of the three locations are 902ms, 950ms, and 1,573ms respectively. This observation matches with that of Figure 2 in that the duration of the same contamination becomes larger as a function of T-R distance. In our wireless sensor network, when a packet is corrupted, the same degree of corruption lasts longer than 900ms in terms of time.

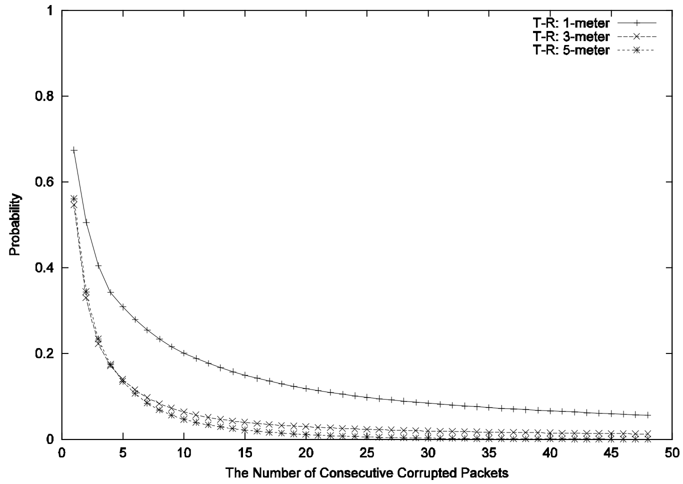

Figure 3 Correlation Function over Sensor Wireless Network in ISI

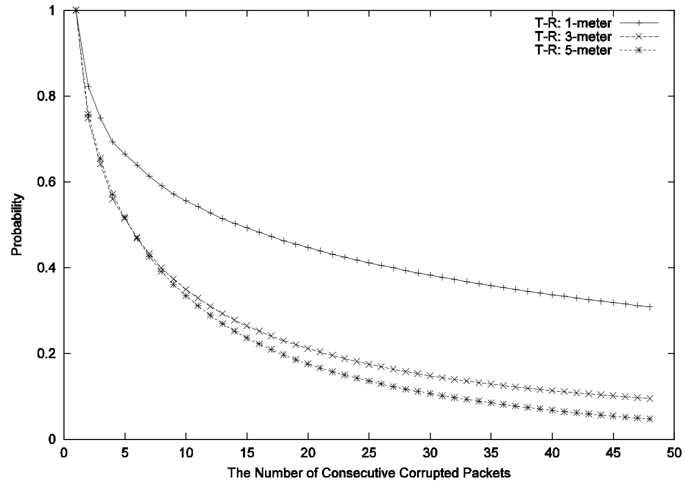

Figure 4 Smoothed Correlation Function over Sensor Wireless Network in ISI

FECA can't follow very rapid changes in time scale, so in Figure 4, we analyze measured data to see how effective FECA will if it adapts only relatively slowly. In detail, we want to confirm whether FECA has a chance to match to the channel status by not responding to every monitored BER change. As a way to find the existence of the relatively low-frequency behavior of wireless channels, we ignore one different FEC level between two same FEC levels when we count the number of n-consecutive packet groups belonging to the same FEC level. For example, when the sequence of computed FEC levels is 2, 2, 3, 2, 1, 1, the smoothed probability of observing four consecutive packets with FEC level 2 is 1/3 by ignoring 1 between two 2's.

Figure 4 says that the wireless channel in ISI exhibits the strong positive correlation characteristics in this relative long-term time scale up to a few seconds. The probability that five consecutive packets suffer the same degree of propagation errors is higher than 50%. The average correlation time expands from a few hundred milliseconds to a few ten seconds like 2.1 seconds, 2.0 seconds and 4.4 seconds at these three places. This observation strongly suggests that we can still improve the performance even by slowly following these low-frequency fluctuations even though we don't adapt to every BER change due to the delay of feedback and the lack of the detailed information about the wireless channel.

Figure 5, finally evaluate whether the addition of the check-byte is worth in this wireless network since the addition also increases the number of corrupted bytes due to the increased packet size. Figure 5 plots the number of corrupted bytes increased due to the added check-byte when the T-R distance is 3-meter. The number of added check-byte is the number n at x-axis of Figure 5 minus with 20-byte data at the origin. The three lower lines in Figure 5 represent three different days' measurements and each point on the three lines averages around one-hour trace. And the uppermost solid line without any mark indicates the number of correctible bytes as a function of added check bytes. The steepest slope of the three lower lines in Figure 5 indicates that the addition of 10-byte incurs 3-byte of corruption in the worst case. Since the addition of two check bytes cures one corrupted byte in our FEC algorithm that we plan to implement in Motes, FECA can still correct two-byte more than the number of bytes corrupted due to the addition of FEC code.

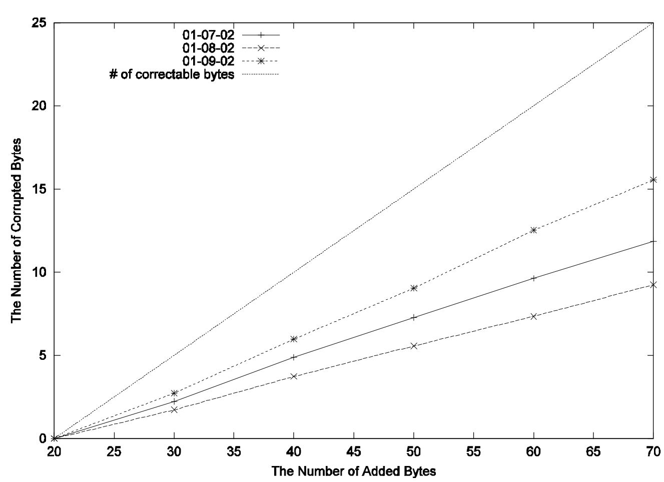

Figure 5 The Number of Corrupted Bytes as a Function of Packet Size

IV. FECA Algorithm

In this section, we design FECA algorithm that can gradually adapt up to the average BER fluctuations and even short-term fluctuations whose average duration lasts at least longer than a few tens of RTTs. This adaptation problem would be a search problem to pick a suitable FEC level from a number of available FEC levels with the implicit feedback such as timeout about each corrupted packet loss.

The search algorithm closely depends on the feedback type such as explicit and implicit. When a transmitter gets the previous BER information like Holland [8], the selection of FEC level is easily decided for the next transmitting packet even though this scheme requires existing protocols to be modified for acknowledgement packets to carry the BER information on their header.

When the positive acknowledgement for the packet arrival is only available, the search problem would be equivalent to the one that the receiver-driven layered multicast (RLM) [12] tried to solve. RLM looks for the optimal number of multicast groups whose combined sending rate matches to each receiver's dynamically available bandwidth. The transmission rate summing different combinations of multicast groups would be analogous to the strength of each FEC level.

These two problems, however, differ in the time scale of fluctuations that the two algorithms endeavor to trace down. RLM tries to follow relatively slowly varying fluctuations while FECA needs to deal with relatively short-term but wide oscillations by using fast link-by-link feedback. RLM is also more complicated since it needs to tell other nodes' adjustment from its own one. The learning of other nodes' behavior, the late feedback and the uncertainties due to statistical multiplexing only allow RLM to follow low-frequency variations based on the average packet loss. RLM, for example, adopts three time windows (100 sec, 10 sec, 1 sec) for evaluating the degree of the average network congestion to decide the layer drop. In contrast, FECA quickly tracks down variances whose duration is longer than the fast feedback without much uncertainty on the effect by its behavior.

Based on these differences, FECA rapidly joins at the next higher FEC level at each packet loss. Most flow control algorithms adopt a weighted average to filter out high-frequency changes. Since FECA tries to catch up with the short-term variance longer than ten times of RTT, it is equivalent to that FECA sets the smoothing factor with 1 for rapid response. This prompt reaction to each packet loss can cause spurious predictions for short-term changes lasting less than one RTT, leading to heavier check bits. To minimize the effect of false predictions, FECA needs to quickly drop from the newly joined level to the lower level when packet loss is no longer reported.

To determine the appropriate drop time, FECA has a drop timer in each FEC level whose timeout value is adjusted by an exponential back-off algorithm. Whenever FECA joins the higher FEC level due to the packet loss, it increases the drop timer of this new FEC level up to T_{max} by multiplying with the multiplication factor, α greater than 1. This increase operation implies that the more FECA visits a FEC level, the larger its drop timeout becomes so that FECA stays at this FEC level for longer time. Notice that α and T_{max} determines the polling frequency to see whether the channel status is enough improved to lower the FEC level, thus affecting the stabilization overhead of FECA. The smaller α, for example, the more often FECA would aggressively poll the channel status by frequently going down to the lower FEC level.

FECA, furthermore, retains a global polling timer to decay the drop timers of other FEC levels except the drop timer of the current FEC level. In contrast to learning the current channel status, FECA also needs to forget the old status by gradually resetting the drop timer's value. Whenever the global polling timer is expired every T_p, FEC shrinks drop timeouts of the all other levels up to T_{min} by multiplying with a decay factor, β less than 1. The longer FECA has not dropped any packet by adopting a certain level of FEC, the shorter it will stay at the other levels of FEC when it adopts one of the other levels next time.

Figure 6 depicts an FECA's representative behavior adjusting the current level of FEC, L_{cnt} to the channel BER fluctuations. At first FECA grows L_{cnt} from 0 to 3 as it suffers three consecutive packet losses during the beginning $3 * T_r$ where T_r is a timeout taken before resending the packet by the link layer. On the successful delivery, FECA keeps L_{cnt} to level 3 until its drop timeout expires. On the drop timer's expiration at time S_3, FECA lowers L_{cnt} to level 2 to see if the channel is improved. When FECA experiences another packet loss, FECA restores L_{cnt} to level 3 and lengthens its drop timeout S_3 by multiplying with α. In parallel with this adjustment of L_{cnt}, the global polling timer constantly shortens the drop timeout of all other levels by β whenever polling timeout, T_p. is expired.

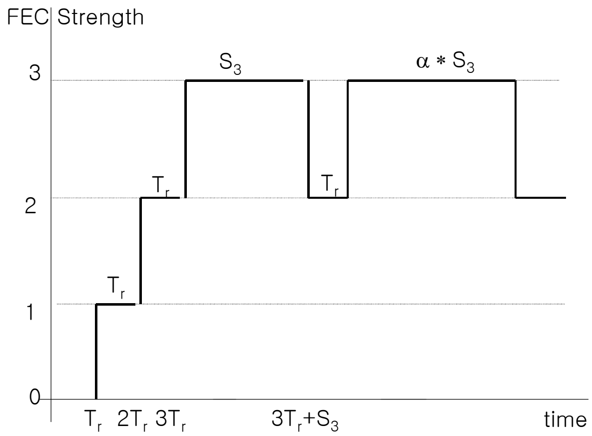

Figure 6 FECA Behavior Example

V. SIMULATION METHODOLOGY FOR PATH LOSS IN WIRELESS CHANNELS

Recently, as wireless networks are integrated into the Internet, the impact of bursty error in wireless channels has begun to be investigated. Packet-level network simulators [13][14] now model both wired and wireless networks. Since one analytical model seldom represents all various wireless channels, simulators tend to include several models to select based on the simulation purpose.

The current ns-2, version 2.1b8a[14], for example, has implemented three large-scale fading models (free space model, two-ray ground reflection model, and shadowing model) and one small-scale fading model (Ricean distribution) [15]. These models allow ns-2 to compute the average signal power of every packet perceived at the receiver and based on the remained power determine the packet drop either probabilistically or deterministically. This approach typically assumes a constant signal during the transmission duration of each packet, and so doesn't model errors which grow larger in proportional to the packet size. To overcome this problem, Holland [8] re-computes the signal power whenever the small-scale fading duration is expired when the packet transmission duration is longer than the current computed small-scale fading duration.

Packet simulators can also model wireless channels with table-driven approach [16], based on detailed channel simulations or experiments over real wireless networks. This model has a weakness that the channel model only applies to specific networks not general ones. By contrast, channel simulators [17][18] provide measurements of bit-level errors although at considerable computational cost.

Although packet-level error modeling is appropriate for many wireless models, it is insufficient to study FEC performance which is sensitive to bit-level error patterns. (FEC effectiveness depends on both the number and location of damaged bits.) We therefore added a bit-level error model into ns. We model bit-level errors with a simple two-state Markov chain, the Gilbert channel model [19]. This approach is similar to that used by prior researchers [6][20] in studying link-level ARQ and FEC. The two-state Markov chain abstracts bursty error distribution with one state representing a heavy error rate with a short interval, and the other representing a longer interval of light error. Intuitively, these two states would represent the average behavior of large-scale and small-scale fading effects, respectively. Although this approach does not model a specific physical environment, it can be tuned to approximate levels of error.

In ns-2, FEC and errors are models as part of the wireless stack. As a packet passes down the stack on the sender side, FEC increases the packets size according to the number of check bits and the error model calculates how many bits are corrupted. At the receiver side, the FEC module evaluates if the incoming packet is corrected based on the FEC quality and the number of corrupted bits. Valid packets are passed up to the MAC, while invalid packets are dropped.

VI. EVALUATION

We next evaluate FECA performance through simulation. First, we seek to evaluate FECA in a stable scenario to determine if it finds an appropriate level of FEC to match various error conditions. We model error with a several levels of light, long-term error but occasional bursts of heavy error. This corresponds to a two-state model with light-error rates of 0-3% for a fixed 1 second interval, and a 5% heavy error rate for 10 ms.

Figure 7 compares FECA to four state algorithms including ARQ and three FEC schemes (1/4 BCH, 1/3 BCH and 1/2 BCH [21]) while varying the off-state BER. Note that BCH (Bose-Chaudhuri-Hocquenghem) is a variant of Hamming codes for correcting multiple errors. We normalize each algorithm's throughput over erroneous channels to the ideal maximum throughput achieved over the error-free channel. We compute throughput as the total data bits delivered over a simple wireless network in which a transmitter continuously transmits 1023-bit packets over a 512 kbps wireless 802.11b-like MAC channel to the receiver. Each point in Figure 7 represents the average performance over five simulation runs each of which is 100 seconds long. Since the variance of these measurements is less than 1%, we have not plotted their confidence intervals.

For ARQ in Figure 7, we use 802.11b retransmission mechanism characterized by a stop-and-wait and an exponential back-off algorithm. For this experiment, we only corrupt data packets not control packets such as RTS, CTS and ACK packets to avoid dependence on MAC protocol details. This approach would overestimate ARQ performance. For 1/4, 1/3, and 1/2 FEC codes, the simulation adopts BCH(1023, 768, 26), BCH(1023, 708, 34), and BCH(1023, 523, 55) in which three fields in the parenthesis indicate the number of total bits, data bits, and the maximum number of corrupted bits to be recovered. For this experiment, FECA selects one of the above three FEC levels with its five tunable parameters, α, β, T_{max}, T_{min}, T_p assigned to 2, 0.9, 1 seconds, 6ms, and 6ms respectively where 6ms is an approximate minimum timeout taken in this 802.11b network topology for retransmission. Each packet such as data, RTS, and CTS packets takes around 2ms for their transmission over this network.

Figure 7 shows that ARQ's performance quickly degrades as the light-error state BER increases. At a BER of 0.5% all packets are corrupted and ARQ is ineffective. A 0.1% BER, for example, implies that a one-bit error is likely to occur in each 1023-bit frame. In contrast, the three FEC codes provide poor performance when the light-error state's BER is less than 0.1% due to the overhead of check bits, but their throughputs remain constant before their cutoff points, 1.25% and 2% BER for 1/4 and 1/3 BCH. Finally 1/2 BCH provides poor but constant performance throughout the whole range due to its strong correction capability.

Finally, the comparison of FECA performance to the other ones in Figure 7 indicates that FECA succeeds in adapting its strength to the underlying long-term channel state. It achieves nearly the best performance among the four other algorithms over the whole range of error rates by following ARQ, 1/4 BCH, 1/3 BCH or 1/2 BCH in each four intervals, [0, 0.5%], [0.5% 1.5%], [1.5% 2.275%], [2.275%, 3%] respectively even though there is negligible performance discrepancy. The performance gap would be due to the overhead of FECA's dynamic adjustments such as blind drops from a higher layer driven by drop timers' expiration and rapid increase of FEC responding to short-term BER spikes.

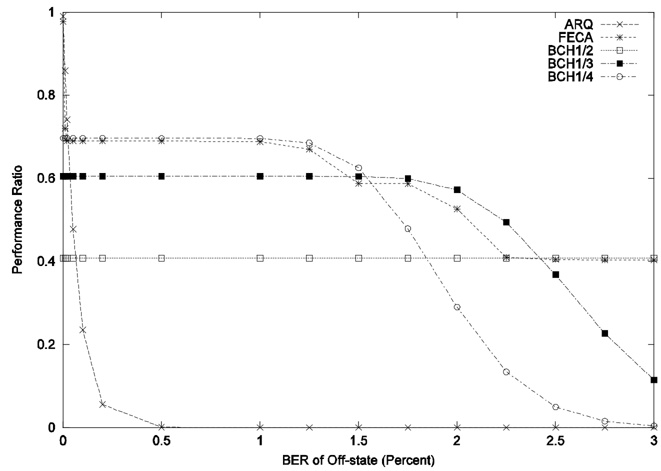

Figure 7 Five Error Correction Algorithms Performance over a Wireless Channel

We next evaluate FECA behavior, the overhead of FECA adaptation as a function of the duration of error states when a node moves back and forth between two places. We abstract this wireless channel with two two-state Markov chains each of which is determined as (5% BER, 5 msec), (0.3% BER, t_v), and (5% BER, 5 msec), (3% BER, t_v) respectively where t_v indicates the variable time period varying from 10ms to 1 second plotted as x-axis. And each state's transition probability to the next one is set to 0.7 with the stay probability to 0.3. We set T_{max} to 100ms to enhance the responsiveness of FECA for short-term fluctuations.

From Figure 8, we can see that the four static algorithms (all but FECA) provide basically the constant performance regardless of the fluctuation duration. In detail, the performance of 1/3 and 1/4 BCH are indistinguishable since they suffer the same degree of packet losses in the 3% BER state. On the other hand, ARQ performs poorly as the duration of the two light states becomes shorter since the MAC layer's timeouts due to packet losses makes ARQ more frequently miss the chance to resend packets during the next adjacent light error state.

As the variable duration, t_v becomes larger, FECA begins to perform better than the other algorithms, by up to 12% at 1 second comparing to 1/2 BCH. It is because that as the variable BER state becomes stable over longer times, the adjustment overhead of FECA is negligible compared to the performance gain from the dynamic adjustment. As the BER oscillates more rapidly like when the variable duration, t_v is smaller than 100ms, the adjustment of FECA achieves less improvement than 1/2 BCH even though Figure 8 shows that FECA still achieves better than the other three algorithms, ARQ, 1/3, and 1/4 BCH. Notice that 100ms is the time taken for sending around 17 packets since each packet takes 6ms to arrive at the receiver. Finally, we observe that when BER oscillates quickly, it is impossible for FECA to stabilize.

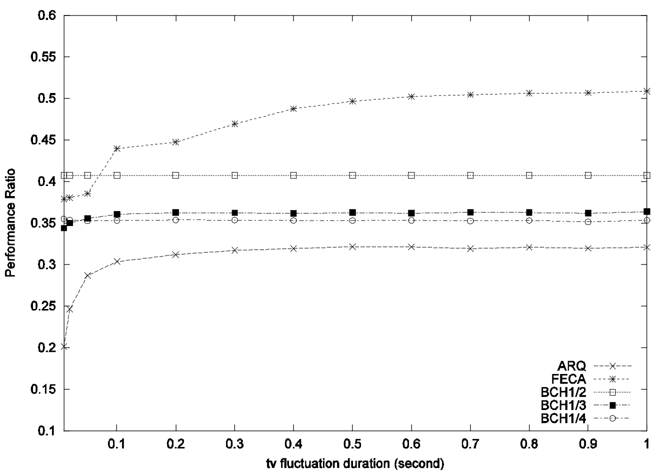

Figure 8 Five Error Correction Algorithms Performance over a Mobile Wireless Channel

VII. Conclusions

This paper proposes an adaptive FEC algorithm, FECA that dynamically trades-off between reliability and efficiency for better performance. We use experiments taken from short-range radios to characterize the range of bit-level errors, and then compare FECA to several alternatives through simulation. Simulation experiments show that FECA can outperform static FEC or ARQ algorithms over a range of error conditions, provided that error rates do not oscillate two rapidly. We are currently implementing FECA over the UCB Mote hardware [22]. Finally, we plan to evaluate FECA performance over a network of motes.

References

[1] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan. *Building Efficient Wireless Sensor Networks with Low-Level Naming*, www.cs.ucsd.edu/sosp01/papers/heidemann.pdf SOSP01, October 2001

[2] T. S. Rappaport. *Wireless Communications: Principles and Practice*, Prentice Hall, 2nd edition 2002

[3] GSM 05.03 Channel coding

[4] S. Lin and D. J. Costello, *Error Control Coding*, Prentic Hall, 1983

[5] J. C. Bolot, S. Fosse-Parisis, and D. Towsley, *Adaptive FEC-Based Error Control for Internet Telephony*, Infocom'99, pp. 1453-1460, April 1999

[6] P. Lettieri and M. B. Srivastava. *Adaptive Frame Length Control for Improving Wireless Link Throughput, Range, and Energy Efficiency*, Proceedings of Infocom'98, pp. 564-571, April 1998

[7] G. Wu, C-W. Chu, K Wine, J. Evans, and R. Frenkiel. *WINMAC: A Novel Transmission Protocol for Infostations* 49th IEEE Vehicular Conference Proceeding, pp. 1340-1344, May 1999

[8] G. Holland, N. Vaidya, and P. Bahl, *A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks*, ACM SigMobile, pp 236-250, July 2001

[9] Sklar, B. *Digital Communications:Fundamentals and Applications*, Prentice Hall 2001

[10] TR100 Chip manual, http://www.rfm.com

[11] http://www.csl.sony.co.jp/person/morelos/ecc/codes.html

[12] S. McCanne, V. Jacobson, and M. Vetterli. *Receiver-driven Layered Multicast*", Proceedings of the SIGCOMM vol 26, 4 pp.117-130, 1996

[13] M. Takai, R. Bagrodia, A. Lee, M. Gerla, *Impact of Channel Models on Simulation of Large Scale Wireless Networks*, Proceedings of MSWiM'99, pp. 7-14, August 1999

[14] Wireless and Mobility Extensions to NS-2, http://www.monarch.cs.cmu.edu/cmu-ns.html

[15] R. J. Punnoose, P. V. Nikitin, and D. D. Stancil, *Efficient Simulation of Ricean Fading within a Packet Simulator* Proceedings of VTC'00, pp. 764-767, September 2000

[16] A. Kumar and R. Gupta, *Capacity Evaluation of Frequency Hopping Based Ad-hoc Systems*, Proceedings of SigMetrics01, pp. 133-142, June 2001

[17] C. M. Keller, *Generic Channel Simulator*, MIT Lincoln Laboratory Project Report AST-46, August 1997

[18] Rappaport, T. S., S. Y. Seidel, and K. Takamizawa., *Statistical Channel Impulse Response Models for Factory and Open Plan Building Radio Communication System Design*, IEEE Transactions on Communications, vol. COM-39 No. 5, pp794-806, May 1991

[19] E. N. Gilbert, *Capacity of a Burst-Noise Channel*, Bell Syst. Tech. J., vol. 39, pp 1253-1266, Sept. 1960

[20] M. Zorzi. *Performance of FEC and ARQ Error Control in Bursty Channels under Delay Constraints*, Proceedings of VTC'98, pp. 1390-1394, May 1998

[21] W. Peterson and E. Weldon, Jr., *Error-Correcting Codes*, 2nd Edition, The Massachusetts Institute of Technology

[22] J. Hill, R. Szewczyk, A. Woo, S. Hollar, A. Culler, and K. Pister, *System Architecture Directions for Network Sensors*, Proceedings of the 9th international conference on Architectural Support for Programming Languages and Operating Systems, pp. 93-104, Nov. 2000