Hindawi Publishing Corporation
Advances in Multimedia
Volume 2007, Article ID 64695, 10 pages
doi:10.1155/2007/64695

Research Article

Survival of the Fittest: An Active Queue Management Technique for Noisy Packet Flows

Shirish S. Karande, Kiran Misra, and Hayder Radha

Department of Electrical and Computer Engineering (ECE), Michigan State University, East Lansing, MI 48824, USA

Received 7 November 2006; Accepted 21 December 2006

Recommended by Jianfei Cai

We present a novel active queue management (AQM) technique to demonstrate the efficacy of practically harnessing the predictive utility of SSR indications for improved video communication. We consider a network within which corrupted packets are relayed over multiple hops, but a certain percentage of packets needs to be dropped at an intermediate node due to congestion. We propose an AQM technique, *survival of the fittest* (SOTF), to be employed at the relay node, within which we use packet state information, available from SSR indications and checksums, to drop packets with the highest corruption levels. On the basis of actual 802.11b measurements we show that such a side information (SI) aware processing within the network can provide significant performance benefits over an SI-unaware scheme, *random queue management* (RQM), which is forced to randomly discard packets. With trace-based simulations, we show the utility of the proposed AQM technique in improving the error recovery performance of cross-layer FEC schemes. Finally, with the help of H.264-based video simulations these improvements are shown to translate into a significant improvement in video quality.

1. INTRODUCTION

The traditional version of the 802.11b MAC [1, 2] drops all the packets that contain any bit corruptions. While such protocols definitely help in modularizing the system architecture, under certain channel conditions, the packet drops due to bit corruptions can be prohibitively high, especially for bandwidth hungry multimedia applications. Consequently, many recent studies (e.g., [3–17]) have recommended the development of protocols that do not drop corrupted packets and recover information even from partially damaged packets. In this paper, we collectively refer to such (and only such) protocols as cross-layer protocols. Prior investigations into cross-layer protocols have reported significant utility for delivery of video over wireless channels.

In a traditional protocol, a packet is either erased (i.e., dropped) or received without any distortions (i.e., errors). Thus, a receiver has complete channel state information (CSI). As against this, in cross-layer protocols, where the channel conditions are defined by the bit error rates (BERs) in the corrupted packets, a receiver does not inherently have CSI about the corruption levels in individual packets. Consequently, even though it is well known that receiver side CSI leads to a capacity gain (see [18, 19]). Most of the prior work on the discussed cross-layer protocols completely ignores the need and utility of such CSI. In [14] to motivate the utility of monitoring the corruption levels in packets we consider the use of checksums to differentiate between corrupted and un-corrupted packets. Even, such a simplistic binary CSI proved to have significant utility in improving error recovery. However to gain further performance gains it is necessary to identify practical mechanism that could facilitate further differentiation between corrupted packets. Motivated by the above discussion, in [16, 17, 20] we observed that typical 802.11b radio devices can associate signal-to-silence ratio (SSR) indications with individual packets. Experimental studies in [20] suggested that these SSR indications could be used to infer the BER in each individual packet with sufficient accuracy. In [16] it was shown that such improved CSI obtained from SSR indications can be used to realize performance gains in error recovery and video quality.

The predictive utility of SSR indications facilitates the design of novel cross-layer protocol optimizations, which would otherwise be infeasible. In this work, to exhibit the practical utility of this prediction tool, we consider a network scenario which has been largely ignored by studies that recommend relay of corrupted packets. We consider the relay of corrupted packets over multiple hops in presence of congestion. Congestion control [21] is a component essential to maintain the operability of any network. Hence, to scale cross-layer protocols to larger networks it is essential to develop resource allocation strategies that are optimized to work in presence of corrupted packets. We believe that prior works have not investigated such strategies due to the lack of a practical predictive tool. In this work, we exhibit the utility of SSR indications in optimizing the resource allocations within the network. We propose an active queue management scheme, *survival of the fittest* (SOTF), under which the relay node ranks the received packets in accordance with the BERs predicted by the side information. Thus under SOTF, rather than randomly discarding the packets, we propose to discard the packets that are suspected to have the highest corruption levels. If the BER estimates are robust, then such a mechanism naturally reduces the total bit corruptions seen by the eventual receiver leading to an improvement in the capacity. To the best of our knowledge, this is a first attempt to realize such a queue management mechanism.

To demonstrate the utility of SOTF we compare its performance with a random queue management (RQM) scheme that randomly discards packets at the relay node. Our comparisons are based on experimentation with actual 802.11b error and SSR traces. We limit our analysis to a simple yet representative two-hop topology. We firstly compare the performance of the above schemes in terms of the statistical analysis of the channel conditions that are presented to the eventual receiver. Subsequently, at the receiver, for both queue management schemes, we employ an error recovery mechanism that utilizes SSR and checksums as side information. Such an error recovery mechanism currently provides the state-of-the-art performance. We employ low density parity check (LDPC) code-based FEC simulations to show that side information at the relay node can improve the performance of cross-layer protocols beyond which was the previously best reported performance [16]. Finally we utilize the proposed FEC scheme to deliver H.264 [22] compressed video. Our simulations show that SOTF can improve the video quality by peak signal-to-noise ratio (PSNR) gain of several dB.

The organization of the paper is as follows: in Section 2 we review the related works. In Section 3 we briefly review our trace collection methodology and some essential observations from [16, 20]. In Section 4, we describe the network model and the proposed SOTF scheme. Section 5 focuses on statistical analysis of the BER observed by the eventual receiver under both queue management schemes. In Section 6 we present the analysis based on FEC and video simulations. Finally, in Section 7 we summarize the key conclusions of this work.

2. RELATED WORK

2.1. Cross-layer protocols that do not drop corrupted packets

Utility of relaying corrupted packets was first explored by Larzon et al. in their UDP-lite protocol [3]. This work was subsequently extended by Singh et al. for cellular video [4]. The complete UDP protocol [5], which used the frame error rate from the radio link physical layer for improved performance, was presented by Zheng and Boyce These cross-layer protocols were further extended, especially to 802.11b WLANs. The following studies have investigated various aspects of relaying corrupted packets: Servetti and De Martin have investigated design of unequal error protection [6], Masala et al. investigated performance of standard compatible multipacket combining [7], Riemann and Winstein investigated the utility of cross-layer protocols in extending the range of 802.11b [8], Pauchet et al. proposed techniques for robust source coding in presence of bit corruptions in [9], Jiang proposed a technique to recover packets from header corruptions without any additional redundancy [10]. The above studies have largely concentrated on single-hop networks, however [11, 12] exhibit the utility of cross-layer protocols in multihop networks.

The authors have been involved in the design of cross-layer protocols and the prior contributions that we build upon are the following.

(i) In [13] we investigated the utility of relaying corrupted packets in 802.11b WLANs on the basis of actual error traces. This study was the first attempt at exploring the utility of corrupted packets in 802.11b WLANs and also the first study to base their analysis on actual 802.11b error traces. In the works mentioned in the above paragraph, with the exception of [8, 9], all studies on 802.11b networks are based on some model-based simulations and not actual error traces. Additionally the work presented in [9] bases their analysis on the error traces used in [13].

(ii) In [14] we considered an abstraction of the cross-layer protocols and conducted theoretical analysis to identify important design guidelines for cross-layer protocols. Prior to this work the importance of side information in cross-layer protocols was completely over looked. In [14] we showed that realizing cross-layer protocols by merely turning off the checksums could have adverse consequences. In particular, it was shown that in the absence of side information the performance of cross-layer protocols can actually be worse than conventional protocols. Nevertheless, in [14], we exhibited that, with side information (SI) cross-layer protocols always perform better than conventional protocols, and also better than cross-layer protocols without SI.

(iii) In [14] it was shown that the improvement of cross-layer protocols could be severely diminished if a large number of packets are dropped due to corruptions in the header. Introduction of parity bits in the header,

for error correction, is not always architecturally feasible. Hence, in [15] we presented a scheme, which can detect the packet header and identity in presence of bit corruptions. Based on experiments with actual networks, it was shown that under realistic traffic load, the header detection scheme could drastically reduce the number of packet drops. A similar scheme has since been presented in [10].

(iv) In [16, 17, 20] we extend the work presented in [14]. In particular, we explore the feasibility of improving the side information mechanism. In [20] we observed that the BER in a corrupted packet has a largely environment invariant relationship with SSR indications. Hence, in [16] we used SSR indications as side information for the error recovery algorithms. We observed that utilization of SSR indications leads to significant performance gains on top of the performance achievable by just employing the recommendations in [14, 15].

The work presented in this paper improves upon the recommendations in [14–17, 20] by utilizing side information at the relay node. Our work differs significantly from prior multihop investigations, since in [11, 12, 17] no side information is used for protocol optimizations, while in [14, 16] the side information is used only at the eventual receiver.

2.2. Other related works

Performance evaluation of protocols on the basis of actual 802.11b measurements is essential to verify their practical efficacy. A number of prior works (e.g., [23, 24]) have presented performance evaluations of actual deployment of 802.11b networks, but these studies do not include analysis of corrupted frames. The authors' prior work in [13–17, 20, 25–27] is among the few studies that actually utilize bit level measurements. Additionally [16, 17, 20] are the only studies, we are aware of, that relate the bit corruptions and thus the capacity under cross-layer protocols to SSR indications. As explained previously, the presented work advances the analysis presented in [16]. It is also important to note that link quality-based real-time protocol optimizations have been investigated in many prior efforts (e.g., [28–30]). However, studies such as these mostly concentrate on packet drops and/or base their analysis on theoretical channel models. The work in [16] and the presented work are the only studies we are aware of that utilize actual SSR measurements for real-time protocol optimizations.

3. TRACE COLLECTION METHODOLOGY

The error traces used in this work are drawn primarily from the channel modeling study presented in [20] and a few similar additional measurements. The traces we use are categorized into "Home" and "Office" traces to represent a low interference and relatively high interference environment. (However, note that in both cases the amount of traffic on an identical channel is low compared to the amount of data we collected.) We base our analysis on a total of 3 million packets, where each packet is of size 1024 bytes. For all the traces used in this paper, the 802.11b PHY data rate was always maintained at 11 Mbps. The packet payload size was maintained at 1024 bytes. The trace collection methodology we employ is such that we are able to associate an SSR indication with each packet. In [20] we empirically evaluated the relationship between SSR and (a) $1-\delta$, the probability of getting a corruption-free packet (b) ε, the probability of bit error are conditioned on the fact that the packet is corrupted. To facilitate an independent reading of this document we have replicated the results presented in [20], in Figure 1. Figure 1(a) shows the relationship of SSR to $1-\delta$ and Figure 1(b) shows the relationship of SSR to ε. In [20] it was shown that SSR indications have an environment invariant relationship with $1-\delta$ and ε. In Figure 1(a) it can also be observed that the probability of packet corruption is negligible for SSR > 14 dB and the bit error rate for SSR < 8 dB is very high. Thus the

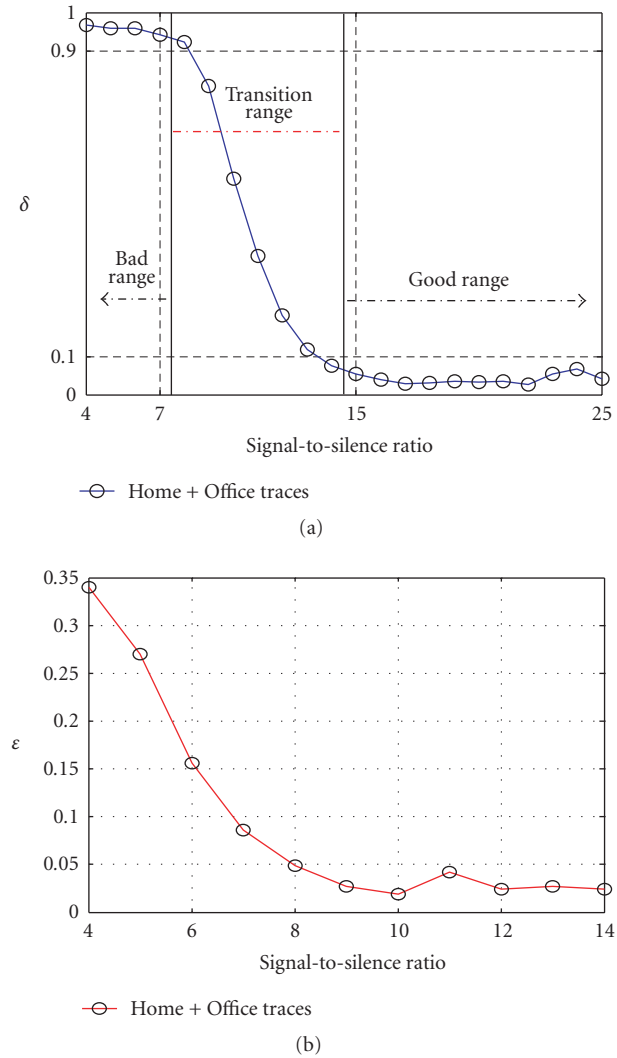


Figure 1: (a) δ(SSR), probability of receiving an error-free packet (b) ε(SSR), probability of bit error in a corrupted packet as a function of SSR [16].

utility of cross-layer protocol is primarily exhibited when a significant proportion of the packets is received with SSR values in the "transition range" (8–14 dB) [16, 17, 20]. Consequently, we focus our analysis on the error traces primarily in this range.

4. NETWORK MODEL

The network topology we consider in this paper is motivated by the illustration in Figure 2(a). Figure 2(a) shows a media server connected to a wireless access point (AP), A. Through A it serves video to clients C, D, and E. The content meant for client C has to be routed through an intermediate wireless router B. However, router B is also involved in cross-traffic from X to Y. Here X and Y could represent a single node or a group of nodes. Since at router B the access to the medium has to be shared, the bandwidth available to the video flow on link B-C is less than that available on link A-B. Additionally as the transmissions at A are meant for multiple receivers, it is not always feasible to optimize the rate allocations, at the media server, purely for client C. In such a scenario router B has to drop a certain number of packets due to congestion. The packets at router B can be dropped actively to avoid congestion or get dropped due to buffer overflow. The network topology illustrated by Figure 2(a) could be considered to represent a multihop 802.11b WLAN or could be considered to be a part of larger ad hoc/mesh network. Consequently, even though we focus on a simple network topology, the work presented in this paper can form an important component that can be used to improve communication in larger networks.

In this work we will concentrate on the flow to client C. Additionally, to avoid unnecessarily obscuring the presented discussion, we will assume that the link B-C is corruption-free. Thus the topology illustrated in Figure 2(a) can be represented by Figure 2(b). The packet drops at node B can be represented by a pseudolink B-$\underline{B}$. We assume that an AQM methodology is employed at the node B and packets are discarded actively to avoid overflow. The processing at router B can be described by the flow chart provided in Figure 3. The AQM techniques we consider employ (a) a collection buffer (CB) where all received packets, meant for client C, are input (b) a transmission buffer (TB) from which we opportunistically transmit packets (c) a *packet discarding mechanism* (PDM). The PDM removes blocks of u packets from the CB to always maintain the length of the CB to less than u packets. The PDM then discards v packets from the block of u packets before inputting these packets into the TB for transmission. The parameter v is controlled to ensure that no overflows occur in the TB. Under a steady-state assumption, the length of the TB can be maintained to a value slightly greater than u packets. The buffers CB and TB always operate in a first-in first-out (FIFO) mode. In all the simulations in this work, we assume constant flow loads and thus value of v is maintained constant. The controllability of v under time-varying loads is not a focus of this work and hence detailed investigation from a control perspective is part of future work. The network and queue management model we consider here is

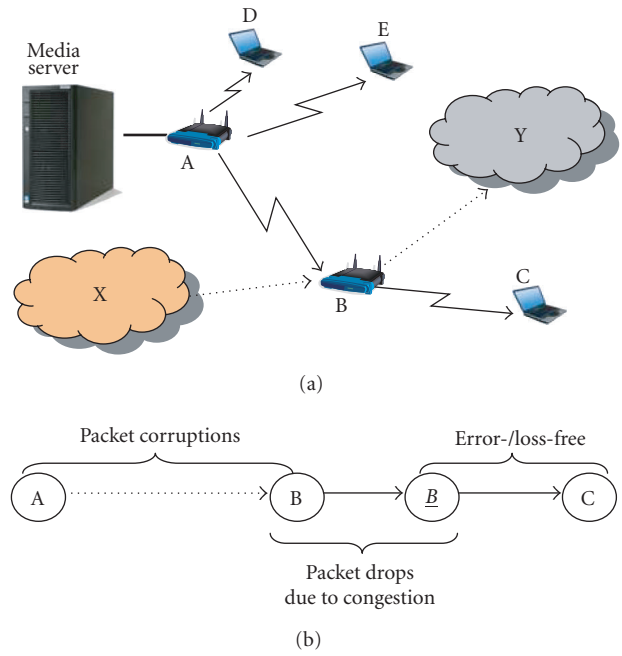


Figure 2: (a) Network topology used for the current work on AQM. (b) The representative equivalent topology actually used for simulations.

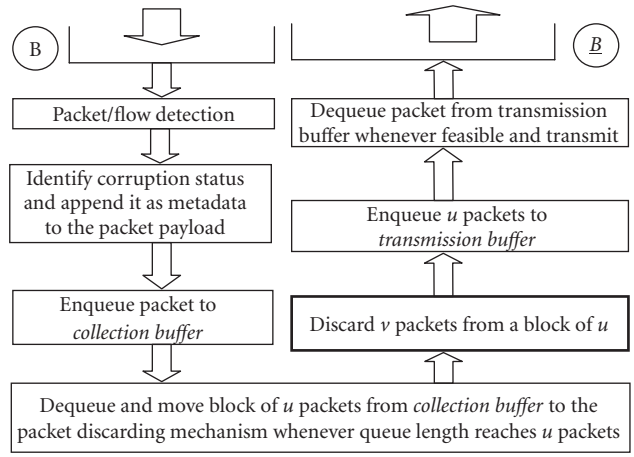


Figure 3: Processing and active queue management at router B.

simplistic, but nevertheless representative. Additionally such a simplistic model allows us to avoid unnecessarily obscuring the presentation of our contributions.

The processing at node B consists of two additional components.

(a) Packet/flow detection

The buffering mechanism we utilize here is specific to a particular flow and thus it is essential to establish the identity of a packet in presence of bit corruptions. A possible algorithm for flow detection is the header detection scheme

presented in our prior work in [15]. The header detection algorithm utilizes flow statistics and baye's rule [18] to detect the critical header fields (CHFs) in presence of bit corruptions. These CHFs are sufficient to establish the flow ID of the packet. In this work, we assume that flow from the media server to the client C forms the major portion of the traffic flowing through router B. In such circumstances, the header detection mechanism works particularly well for the corrupted packets being received from A. For additional details of header detection, we refer the reader to [15]. For the purposes of our simulation, we assume that the header detection works perfectly and thus the cross-layer protocols do not drop any packets due to corruptions.

(b) Identification of the corruption status

For efficient operation of the PDM and to facilitate efficient error recovery at the eventual receiver, it is essential to estimate the BER in each packet. We assume that a checksum has been employed on the entire packet body and thus we are able to differentiate between corrupted and uncorrupted packets. Additionally, we assume that the relay node employs an 802.11b radio device that can provide SSR indications. More specifically, in this paper, we assume that the device is identical to the one used in [16, 20] and hence the corruption status can be determined by utilizing the relationships provided in Figure 1(b). The corruption status of a packet i, with slight abuse of notation, is denoted by ε_i. If the checksums are successful we set $\varepsilon_i := 0$, else we set $\varepsilon_i := \varepsilon(\mathrm{SSR}_i)$ where SSR_i is the SSR indication associated with the packet i. To permit side information-based processing at the eventual receiver, the corruption status of the packet has to be recorded and appended to the packet body as metadata. To record the corruption status we can either record the quantized version of ε_i or record the checksum result along with the SSR indication. In either case we assume that the cost of accurately relaying such metadata over link B-C is negligible.

4.1. Efficient packet discarding

We consider two packet discarding mechanisms in this work.

(a) Random queue management

In this scheme the PDM completely disregards the corruption status of individual packets and discards packets randomly.

(b) Survival of the fittest

In this scheme the PDM sorts the packets in a block of u packets in ascending order of corruption status. In such a sorted list, v packets with the highest corruption status are discarded.

In this paper, as described before, we focus on exhibiting the performance improvement that SOTF can provide over RQM, especially in terms of video quality. To the best of our knowledge this paper is the only study that investigates AQM in presence of bit corruptions. Hence we have to limit our performance comparisons to RQM and SOTF.

5. STATISTICAL ANALYSIS

In this section, we compare the performance of SOTF and RQM in terms of the BER that is observed by the eventual receiver in the relayed packets. SOTF seeks to discard the most corrupted packets and hence in principle should be able to facilitate lower BER. However, performance improvements due to SOTF are critically dependent on the predictive utility of SSR indications. Hence the first statistic we measure is $p(\mathrm{BER}_{\mathrm{RQM}} > \mathrm{BER}_{\mathrm{SOTF}})$, which represents the proportion of packet blocks for which the BER observed by the eventual receiver under SOTF is less than that observed under RQM. Prior to presenting the results of such measurements, it is important to note that existence of corrupted packets is essential to differentiate between RQM and SOTF. Hence, we base our analysis only on those blocks of packets that have at least one corrupted packet. The above restriction is roughly equivalent to continuously operate under 20 dB.

In accordance with the above discussion, we evaluated $p(\mathrm{BER}_{\mathrm{RQM}} > \mathrm{BER}_{\mathrm{SOTF}})$ on our entire data set. The results of our empirical measurements, for various block lengths u and congestion-based packet drops v, are presented in Table 1. It can be observed in Table 1 that 90% of the times, even for short block lengths, SOTF performs better than RQM. Such an observation remains true irrespective of the number of packets being dropped due to congestion. Additionally as the block length is increased, the guarantee with which SOTF provides improved performance also increases.

The results in Table 1 are sufficient to indicate that with a high likelihood, SOTF should perform better than RQM. However to quantify the magnitude of average improvement we evaluated $E[\Delta\mathrm{BER}] = E[\mathrm{BER}_{\mathrm{RQM}} - \mathrm{BER}_{\mathrm{SOTF}}]$, where $\mathrm{BER}_{\mathrm{RQM}}$ and $\mathrm{BER}_{\mathrm{SOTF}}$ are random variables that represent the BERs observed in each u-v packet block received by the eventual receiver. Table 2 presents the results of our evaluations. It can be seen that SOTF can reduce the average BER by 0.3 to 1%. The average BER we observed in RQM was 4.5%, hence the reduction in BER due to SOTF can be considered significant.

Finally, it is important to note that the link quality experienced by each block of packets can vary significantly. Hence, the BER in each block and the performance improvement due to SOTF can also vary. To illustrate the performance improvement provided by SOTF in presence of such variations, in Figure 4 we present scatter plots for Home and Office setup. We arbitrarily chose $u = 64$ and $v = 6$ for the measurements presented in Figure 4, but similar results can be obtained for other parameter values. From Figure 4 it can be observed that even for identical values of average SSR the performance improvements due to SOTF can vary significantly. There indeed exist a few cases when SOTF performs worse than RQM. However despite the variation in performance improvement, for most of the blocks, irrespective of the average SSR, SOTF provides reductions in BER. Thus results in this section provided substantial statistical evidence of the

Table 1: Empirical evaluation of $p(\text{BER}_{\text{RQM}} > \text{BER}_{\text{SOTF}})$.

v/u	Packet block length u			
	32	64	128	256
0.03	0.88	0.92	0.95	0.97
0.06	0.89	0.93	0.95	0.97
0.09	0.85	0.91	0.95	0.97
0.12	0.90	0.93	0.95	0.97
0.15	0.87	0.92	0.95	0.97

Table 2: Evaluation of $E[\Delta\text{BER}] = E[\text{BER}_{\text{RQM}} - \text{BER}_{\text{SOTF}}]$ in 10^{-2}.

v/u	Packet block length u			
	32	64	128	256
0.03	0.3030	0.3417	0.3649	0.3747
0.06	0.5716	0.6029	0.5950	0.5779
0.09	0.7165	0.7480	0.7269	0.6842
0.12	0.9639	0.9223	0.8514	0.7892
0.15	1.0211	0.9896	0.9006	0.8361

capacity gain that can be facilitated to the eventual receiver by employing SOTF at the relay node.

6. SIMULATIONS

6.1. Error recovery performance of LDPC -based FEC

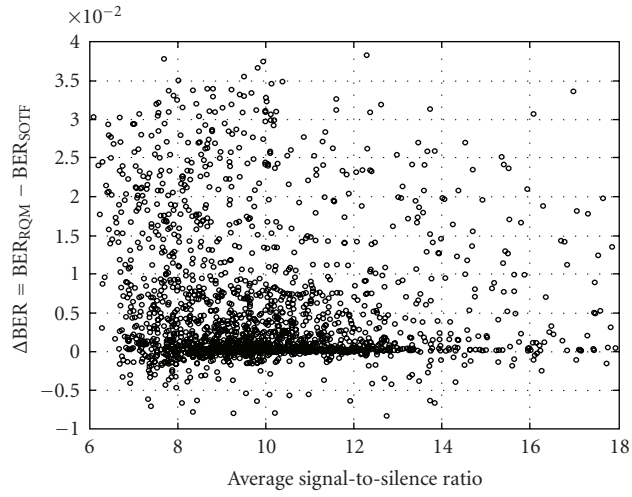
In this section we exhibit the utility of SOTF in improving the performance of a cross-layer FEC and hence in improving the video quality. Without loss of generality for the remainder of the document we will focus our analysis on parameter values $u = 64$ and $v = 6$. For our simulations we assume that the media server employs an LDPC-based FEC scheme. The FEC scheme is block-based and each block consists of 64 packets of size 1024 bytes. Each FEC block can be broken down into multiple code words. The length of the individual code words and the total code words in each FEC block vary in accordance with the chosen channel coding rate. However, we maintain the number of message bits in each code word to 8192 bits. We realize FEC schemes with varying degree of robustness by increasing the length and hence the redundancy in each code word. We consider code length varying from 9216 bits to 16384-bits. Thus, we consider channel-coding rates varying from 0.89 to 0.5. The code words are mapped to the FEC block by interleaving them equally across all the packets in the block. For example, a 16384- bit code word is interleaved across 64 packets in the FEC block by mapping 256 bits from the code word to each packet. Thus in this case the FEC block consists of 32 code words. Additionally note that the LDPC codes we utilize are regular codes and are girth optimized using progressive edge growth (PEG) techniques [31].

At the eventual receiver, the LDPC-based FEC scheme is decoded using log-likelihood ratio (LLR) domain sum-product algorithm. For background information about this algorithm and LDPC codes please refer to [31]. The LLR initialization $L(c_i)$ associated with code bit c_i is dependent on the received bit y_i and an prior assumption of the bit error probability. As suggested in [16], we employ the following initialization.

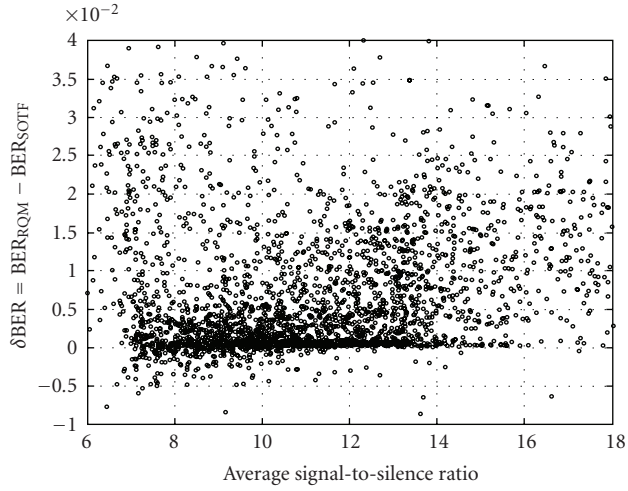
(i) If bit belongs to a dropped/discarded packet, then set $L(c_i) = 0$.

(ii) If bit belongs to an uncorrupted packet, then set the LLR to ∞ if the received bit is 0 or to $-\infty$ if the received bit is 1.

(iii) If the bit belongs to a corrupted packet with SSR indication SSR, then set the LLR as $L(c_i) = (-1)^{y_i} \log((1 - \varepsilon(\text{SSR}))/\varepsilon(\text{SSR}))$ where the relationship $\varepsilon(\text{SSR})$ is obtained by utilizing the corruption status provided by the metadata and the relationship provided in Figure 1.



(a) Home traces



(b) Office traces

Figure 4: The above figures present an XY-plot (Average SSR, ΔBER) for various packet blocks. The block length is fixed at $u = 64$ and the number of packets discarded per block due to congestion is fixed at $v = 6$.

Note that as per the above discussion we assume that at the receiver, the error recovery mechanism can utilize SSR indications as side information irrespective of the AQM technique employed at the relay node. Finally, it should be mentioned that the LDPC decoding algorithm we employ stops upon converging to a code word or when the number of iterations exceeds 25.

Prior to discussing the results, it is essential to take note of the fact that the capacity of wireless channel can vary significantly with variation in SSR. Even in the transition region the capacity of the wireless channel can vary from almost 1 to well below 0.5. The traces we utilize for our simulations have been collected under a variety of channel conditions and hence it is impossible to guarantee a successful decoding for all code words without choosing an impractically low channel coding rate (<0.5). Hence in our analysis presented here we do not make an explicit effort to identify a channel coding rate that recovers almost all the code words. Instead we compare the performance of RQM and SOTF in terms of the chosen set of rates and interpret the results from an outage perspective [18]. In practice we expect our investigations to be combined with other error recovery optimizations (e.g., [19, 32]) however to maintain the clarity of our presentation we avoid these optimizations for now.

Figure 5 shows the result of our LDPC simulations. In Figure 5(a) the performance improvement due to SOTF can be clearly observed. Even at low channel coding rates (0.5 to 0.7) SOTF can improve error recovery performance by over 10% for the Office data and by over 5% for Home data. The performance improvement due to SOTF can be appreciated further by observing the results presented in Figure 5(b). Figure 5(b) shows the difference in the average number of iterations required to process a received word. It can be seen that even at low coding rates SOTF can reduce the number of iterations by over 10%. Such reduction in complexity can be important for power-constrained wireless receivers. Thus the results in this section are consistent with our observation in Figure 4 and provide further evidence of the utility of SOTF. In the subsequent section we will demonstrate the impact of improved error recovery on the quality of video.

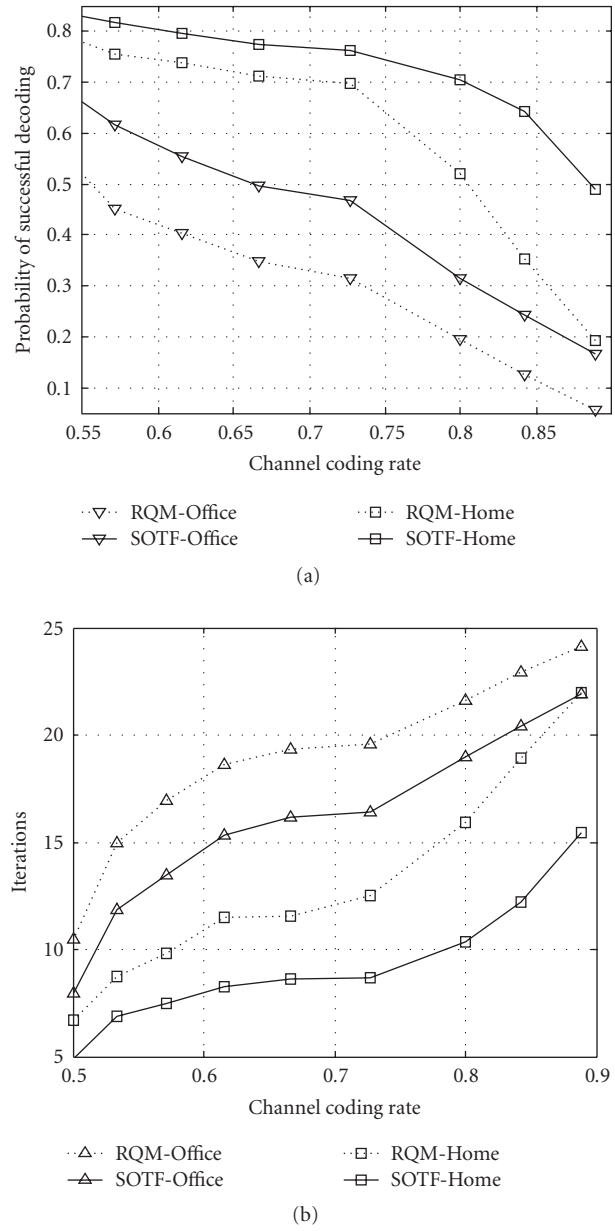
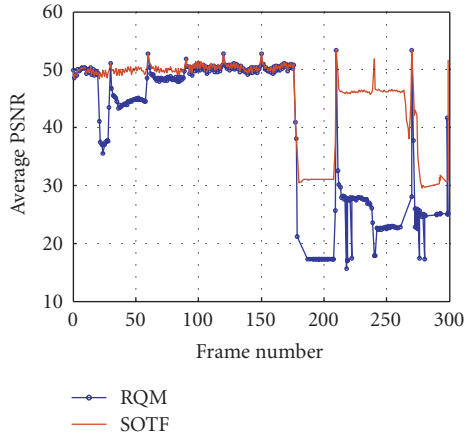


Figure 5: Comparison of SOTF and RQM in terms of the error recovery performance of LDPC based FEC. (a) Probability of successful decoding. (b) Average number of iterations per code word. The block-length is fixed at $u = 64$ and the number of packets discarded per block due to congestion is fixed at $v = 6$.

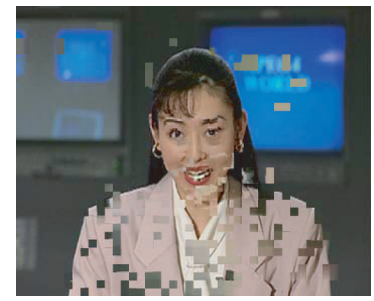
6.2. Comparison based on video quality

In this section we compare the performance of SOTF and RQM on the basis of video quality. We assume that the media server employs an H.264-based source encoder. The compressed video bit stream can be broken into independently decodable video packets of size 8192 bits. The server maps each of these video packets to a single code word in the LDPC-based FEC scheme. At the receiver the compressed video data is recovered from the output of the FEC decoder. If a code word is decoded successfully, then the video packet is sent to the H.264 video decoder, however if the channel decoding is unsuccessful, then the video packet is dropped. For our simulations we employ version 10.1 of the reference software for H.264. The source decoder employs the error concealment feature of "*frame copying*." Our analysis in this section is based on the standard video test sequences Akiyo, Foreman, and Stefan. We use the frame resolution "CIF" and frame frequency of 30 frames/sec. The GOP structure we employ is of the form IBPBP with a GOP size of 30 frames. The source coding employs constant quantization, however the quantization parameter is chosen so as to achieve a bit rate of 1.2 Mbps for the compressed video stream. Additionally, it should be highlighted that the 802.11b error traces we employ for our video simulations are chosen arbitrarily from the larger set of traces. The actual performance metrics can vary on the basis of the chosen traces, nevertheless, the results we

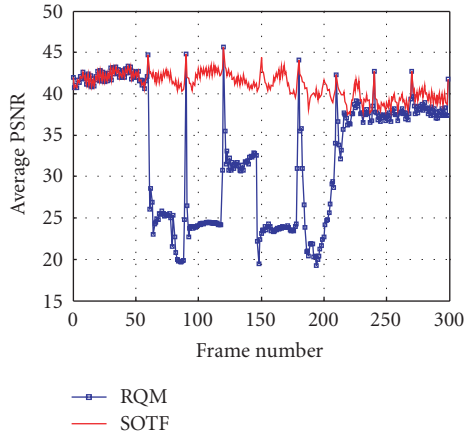


(a)

(b) SOTF, frame 250



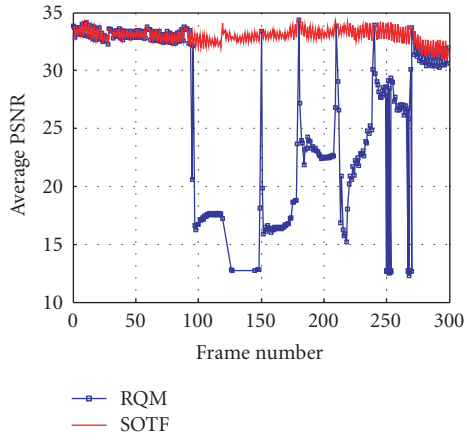
(c) RQM, frame 250



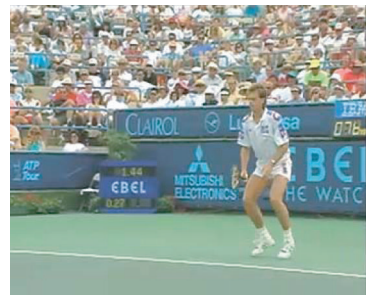
(d)

(e) SOTF, frame 100

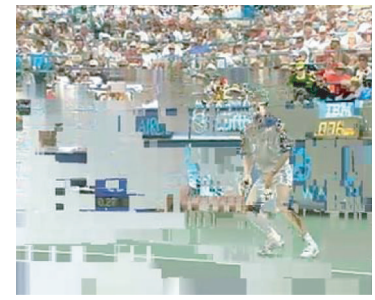
(f) RQM, frame 100



(g)



(h) SOTF, frame 140



(i) RQM, frame 140

Figure 6: Video quality comparison of RQM and SOTF. Above results are based on trace samples obtained from the Home setup and channel coding rate of 0.66. (a), (d), (g) provide a temporal snapshot of PSNR performance. Picture frame-based subjective comparison is presented in (b)–(h) for Akiyo, (e)–(f) for Foreman, (h)–(i) for Stefan. Also note that (b), (e), (h) represent the performance of SOTF and (c), (f), (i) represent the performance of RQM.

Table 3: Video quality: average gain in PSNR (in dB).

Sequence	Akiyo		Foreman		Stefan	
environment	Home	Office	Home	Office	Home	Office
Coding rate						
0.57	0.2	7.19	0.65	3.99	1.48	2.09
0.61	0.2	9.11	2.55	3.99	1.48	2.03
0.66	9.74	11.18	6.16	8.27	7.23	8.43

present in this section are representative of the comparative performance trends observed on all the error traces.

Table 3 presents the results of the video simulations. The performance gain in PSNR due to SOTF has been recorded in Table 3. It can be observed that even at a coding rate of 0.57 the performance gain due to SOTF in the chosen Office traces can be in excess of 2 dB. A similar gain is not seen for all sequences for the chosen Home traces. Nevertheless as the coding rate is reduced the performance gains due to SOTF are evident under all the channel conditions. In particular it was observed that the capacity under RQM drops below 0.66 frequently, while remaining above 0.66 more often for SOTF. Hence for a coding rate of 0.66 it can be observed that video quality degradation under RQM is significantly more drastic than the degradation under SOTF. From Table 1 it can be observed that SOTF can provide PSNR gain in excess of 8 dB for all the considered traces and video sequences. The exact amount of performance improvement is dependent on the choice of error trace, coding rate, and source bit rate. Hence the exact amount of performance gains due to SOTF in actual deployments can vary, but, nevertheless the presented simulations provided sufficient evidence of its significant utility under a variety of channel conditions.

To further illustrate the performance improvement in video quality we present the results for the Home traces for the coding rate 0.66 in additional detail. Figure 6 compares the performance of SOTF and PSNR in terms of temporal snapshots in Figures 6(a), 6(d), and 6(g) and visual samples in Figures 6(b), 6(c), 6(e), 6(f), 6(h), and 6(i). From the temporal snapshots it is clearly evident that SOTF can provide drastic performance gains over muliple GOPs. Especially for the traces used for Foreman and Stefan it can be seen that, while the video quality under RQM deteriorates frequently, SOTF is able to provide quality equivalent to the lossless stream. In Figures 6(b), 6(c), 6(e), 6(f), 6(h), and 6(i) we have chosen a specific picture frame from the corresponding temporal snapshot to facilitate a visual comparison between SOTF and RQM. In the presented examples it can be seen that while the video quality under SOTF is excellent, the video under RQM is incomprehensible. Thus from the presented simulations it is clearly evident that SOTF can provide substantial improvement in video quality compared to RQM.

7. CONCLUSION

In this work we explored the feasibility of harnessing the predictive utility of SSR indications at a relay node. We considered the use of SSR indications in improving the efficiency of AQM at a relay node. We proposed a novel AQM scheme, SOTF, which utilizes side information at the relay node to identify the most corrupted packets and discard them. SOTF was compared with a scheme, RQM, which randomly discards packets without any side information. Experiments with actual 802.11b traces reveal that SOTF can significantly reduce the BER observed at the eventual receiver. Such reductions in BER are responsible for improving the capacity offered to the eventual receiver. The improvement in capacity is exhibited on the basis of LDPC-based FEC simulations and H.264-based video simulations. Our experiments clearly exhibit that SOTF can lead to significant improvement in the error recovery performance and a PSNR gain of several dB in the video quality. Thus in this work we have clearly exhibited the utility and feasibility of improving the efficiency of video communication by utilizing side information at the relay node. In future we hope to extend the work in this paper to multiple relays and to time-varying traffic patterns.

ACKNOWLEDGMENTS

The authors gratefully appreciate the support for this work under NSF Awards CCF-0515253, CNS-0430436, and MEDC Grant GR-296. The authors also appreciate Utpal Parrikar's contributions to an initial version of this work.

REFERENCES

[1] ISO/IEC 8802-11:1999(E), Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specications, August 1999.

[2] IEEE Std 802.11b-1999, Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specications: Higher-Speed Physical Layer Extension in the 2.4GHz band, September 1999.

[3] L. Larzon, M. Degermark, and S. Pink, "The UDP lite protocol," IETF Internet Draft, February 2001.

[4] A. Singh, A. Konrad, and A. D. Joseph, "Performance evaluation of UDP lite for cellular video," in *Proceedings of the 11th IEEE International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '01)*, pp. 117–124, Port Jefferson, NY, USA, June 2001.

[5] H. Zheng and J. Boyce, "An improved UDP protocol for video transmission over Internet-to-wireless networks," *IEEE Transactions on Multimedia*, vol. 3, no. 3, pp. 356–365, 2001.

[6] A. Servetti and J. C. De Martin, "Link-level unequal error detection for speech transmission over 802.11b networks," in *Proceedings of Special Workshop in Maui (SWIM '04)*, Maui, Hawaii, USA, January 2004.

[7] E. Masala, A. Servetti, and J. C. De Martin, "Standard compatible error correction for multimedia transmissions over 802.11 WLAN," in *Proceedings of IEEE International Conference on*

Multimedia and Expo (ICME '05), pp. 880–883, Amsterdam, The Netherlands, July 2005.

[8] R. Riemann and K. Winstein, "Improving 802.11 Range with Forward Error Correction," MIT-CSAIL-TR-2005-011.

[9] F. Pauchet, C. Guillemot, M. Kerdranvat, S. Pateux, and P. Siohan, "Conditions for SVC bit error resilience testing," in *The 17th Meeting of Joint Video Team (JVT '05)*, Nice, France, October 2005.

[10] W. Jiang, "A bit error correction without redundant data: a Mac layer technique for 802.11 networks," in *Proceedings of the 2nd International Workshop on Wireless Network Measurement (WiNMee '06)*, Boston, Mass, USA, April 2006.

[11] E. Masala, M. Bottero, and J. C. De Martin, "MAC-level partial checksum for H.264 video transmission over 802.11 ad hoc wireless networks," in *Proceedings of the 61st IEEE Vehicular Technology Conference (VTC '05)*, vol. 5, pp. 2864–2868, Stockholm, Sweden, May-June 2005.

[12] S. Yi, Y. Shan, S. Kalyanaraman, and B. Azimi-Sadjadi, "Header error protection for multimedia data transmission in wireless ad hoc networks," in *Proceedings of the 13th IEEE International Conference on Image Processing (ICIP '06)*, Atlanta, Ga, USA, October 2006.

[13] S. A. Khayam, S. S. Karande, H. Radha, and D. Loguinov, "Performance analysis and modeling of errors and losses over 802.11b LANs for high-bit-rate real-time multimedia," *Signal Processing: Image Communication*, vol. 18, no. 7, pp. 575–595, 2003.

[14] S. S. Karande and H. Radha, "Hybrid erasure-error protocols for wireless video," *IEEE Transactions on Multimedia*, vol. 9, no. 2, pp. 307–319, 2007.

[15] S. A. Khayam, S. S. Karande, M. U. Ilyas, and H. Radha, "Header detection to improve multimedia quality over wireless networks," *IEEE Transactions on Multimedia*, vol. 9, no. 2, pp. 377–385, 2007.

[16] S. S. Karande, U. Parrikar, K. Misra, and H. Radha, "Utilizing SSR indications for improved video communication in presence of 802.11b residue errors," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME '06)*, pp. 1973–1976, Toronto, ON, Canada, July 2006.

[17] S. S. Karande, K. Misra, and H. Radha, "CLIX: network coding and cross-layer information exchange of wireless video," in *Proceedings of the 13th IEEE International Conference on Image Processing (ICIP '06)*, Atlanta, Ga, USA, October 2006.

[18] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley Series in Telecommunications and Signal Processing, John Wiley & Sons, New York, NY, USA, 2006.

[19] S. C. Draper, B. Frey, and F. R. Kschischang, "Rateless coding for non-ergodic channels with decoder channel state information," http://www.eecs.berkeley.edu/~sdraper/researchDir/pdf/ratelessTimeVary.pdf.

[20] S. S. Karande, U. Parrikar, K. Misra, and H. Radha, "On modeling of 802.11b residue errors," in *Proceedings of the 40th Annual Conference on Information Sciences and Systems (CISS '06)*, pp. 283–288, Princeton, NJ, USA, March 2006.

[21] R. Srikant, *The Mathematics of Internet Congestion Control*, Birkhäuser, Boston, Mass, USA, 2003.

[22] ISO/IEC JTC 1/SC29/WG11 and ITU-T SG16 Q.6, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 — ISO/IEC 14496-10 AVC)," Doc. JVT.

[23] D. B. Faria and D. R. Cheriton, "No long-term secrets: location-based security in overprovisioned wireless LAN," in *Proceedings of the 3rd ACM Workshop on Hot Topics in Networks (HotNets-III '04)*, San Diego, Calif, USA, November 2004.

[24] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan, "Measurement-based characterization of 802.11 in a hotspot setting," in *Proceedings of the ACM Workshop on Experimental Approaches to Wireless Network Design and Analysis (SIGCOMM '05)*, pp. 5–10, Philadelphia, Pa, USA, August 2005.

[25] S. A. Khayam and H. Radha, "Linear-complexity models for wireless MAC-to-MAC channels," *Wireless Networks*, vol. 11, no. 5, pp. 543–555, 2005.

[26] S. A. Khayam and H. Radha, "Constant-complexity models for wireless channels," in *Proceedings of the 25th Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '06)*, Barcelona, Catalonia, Spain, April 2006.

[27] S. A. Khayam, H. Radha, S. Aviyente, and J. R. Deller Jr., "Markov and multifractal wavelet models for wireless MAC-to-MAC channels," *Elsevier Performance Evaluation Journal*, vol. 64, no. 4, pp. 298–314, May 2007.

[28] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, "Cross-layer design for wireless networks," *IEEE Communications Magazine*, vol. 41, no. 10, pp. 74–80, 2003.

[29] M. van der Schaar, S. Krishnamachari, S. Choi, and X. Xu, "Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 WLANs," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 10, pp. 1752–1763, 2003.

[30] M. van der Schaar and S. Shankar, "Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms," *IEEE Wireless Communications*, vol. 12, no. 4, pp. 50–58, 2005.

[31] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 386–398, 2005.

[32] http://www.icsi.berkeley.edu/PET/.