UTILIZING SSR INDICATIONS FOR IMPROVED VIDEO COMMUNICATION IN PRESENCE OF 802.11B RESIDUE ERRORS $^{\perp}$

Shirish Karande, Utpal Parrikar, Kiran Misra, Hayder Radha
Michigan State University, East Lansing, MI 48824, USA
Email:{karandes, prabhupa, misrakir, radha}@egr.msu.edu

ABSTRACT

Radio hardware used for the reception of 802.11b frames is capable of associating a Signal to Silence Ratio (SSR) with each received frame. If a received frame is corrupted, then these SSR indications can be used to provide robust apriori estimate of the bit error rate in the packet. In many recently proposed cross-layer protocols, for transmission of video over wireless networks, recovery of information from partially corrupted packets has shown significant utility. In this paper, based on experiments with actual 802.11b error traces, we show that the Channel State Information (CSI) provided by the SSR indications can be used to improve the error recovery performance of an FEC scheme employed in conjunction with a cross-layer protocol. H.264 based simulation are used to establish the efficacy of the proposed work for video applications; specifically for video over 802.11b WLAN.

1. INTRODUCTION

The utility (especially for multimedia transmission) of cross-layer protocols, that recover information from corrupted packets, has been noticed by many recent research efforts (e.g. [1], [2], [5] and references within). The pivotal concept behind these works is to avoid completely dropping the content of a partially corrupted packet. The responsibility of recovering information from the corrupted packet can be given to the link-level or the application level. In this paper we consider a system model in which the responsibility of error recovery is given to the application level Forward Error Correction (FEC) scheme.

In traditional packet based communication a packet is either erased or is completely error free. Thus the receiver has complete Channel State Information (CSI). On the contrary, when corrupted packets are relayed to the application layer, the received packets at the application level are not necessarily error free. Information about the corruption level or the bit error rate in a packet can help in enhancing the error recovery performance. In [1] with the help of two abstract schemes it was shown that even CSI that can meagerly identify uncorrupted packets from corrupted ones is sufficient to provide substantial performance gains in terms of the eventual video quality. However none of the previous work (including [1]) has analyzed either theoretically or experimentally, the utility of CSI that could provide a robust estimate of the bit error rate in a corrupted packet, on a per-packet basis. A primary reason for this could be the inability to identify methodologies in existing network hardware/software implementations that could provide such a CSI without any additional cost/modifications.

Radio hardware used for reception of 802.11b frames is capable of recording and associating a Signal to Silence Ratio (SSR) to each received frame. In [2] it has been shown that the relationship of the SSR indication to the bit error rate ε in the corrupted packet does not vary significantly across different environments. Therefore, the SSR indication associated with each packet can be used to provide a robust CSI to the cross-layer error recovery mechanism. Thus the current radio-link hardware used for 802.11b already has provisions to facilitate additional CSI about the bit error rate in a corrupted packet[1].

Based on the above motivation, in this paper, we conduct experiments with actual 802.11b residue error traces to establish the utility of the SSR indications in cross-layer systems. Video applications are typically extremely bandwidth hungry and hence have served as the prime stimulus for design of the considered cross-layer schemes. Thus as an example application we consider video (multicast) over an 802.11b WLAN and evaluate the efficacy of the additional CSI in improving in video throughput/quality.

The paper is organized as: Section 2 provides description on the considered cross-layer schemes, the residue error traces used for the experimentation and the FEC scheme employed for video. Section 3 presents the results and observations. The results in this section explicitly establish the utility of the proposed work (especially for transmission of video). Section 3 also presents some capacity deductions that provide essential insight in understanding why and when the SSR indications are useful in improving the performance of the cross-layer scheme. Finally in section 4 we summarize the key conclusions of this work.

2. EXPERIMENTAL METHODOLOGY

2.1. System Model

Figure 1 describes the network model considered in this paper. A remote server is responsible for FEC encoding of the video data and these FEC encoded packets are multicast over a WLAN by the Access Point (A.P), which is turn is connected to a gateway. The multicast packets are received by various clients in the WLAN. We only consider clients that are employing a cross-layer methodology to receive these packets[2]. The 802.11 MAC Frame Check Sequence (FCS) can detect whether a received packet is error free and can provide this CSI to the application layer (just as it is done in a conventional protocol stack).

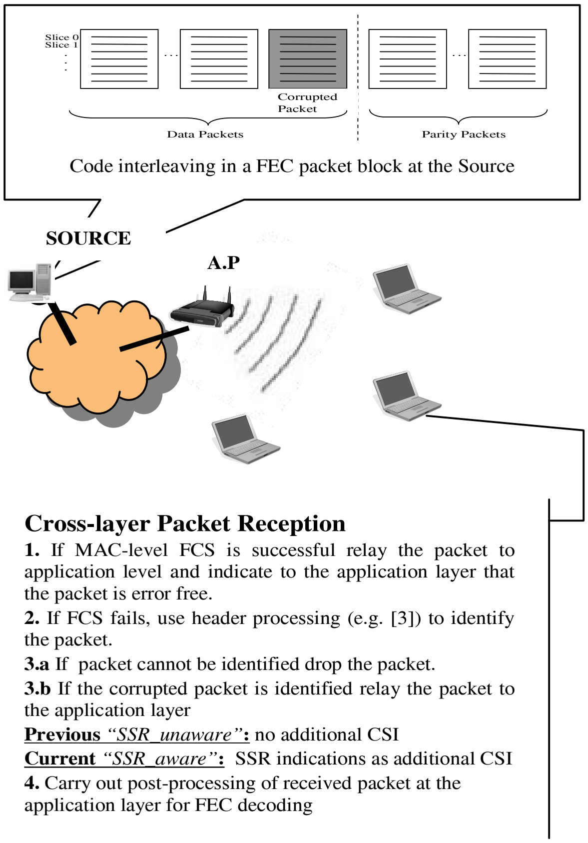
The difference between the proposed cross-layer "*SSR aware*" scheme and the previous "*SSR_unaware*" scheme is articulated by step 3.b of "Cross-layer Packet Reception" in Figure 1. The considered *SSR_unaware* scheme is a combination of the proposals in [1], [2] and [5], which have shown significant improvements for wireless video when compared with traditional (non cross-layer) protocols. It is also very well established that information recovery from corrupted packets is indeed very useful and comparison with

---

$^{\perp}$ This work has been supported by the National Science Foundation under NSF Theoretical Foundation Award CCF-0515253.

[1] Similar functionality may be available for other standards also.

[2] [5] a cross-layer methodology can be completely receiver driven and thus it is feasible for clients that do not employ a cross-layer strategy to co-exist with the ones that do.



Cross-layer Packet Reception

1. If MAC-level FCS is successful relay the packet to application level and indicate to the application layer that the packet is error free.

2. If FCS fails, use header processing (e.g. [3]) to identify the packet.

3.a If packet cannot be identified drop the packet.

3.b If the corrupted packet is identified relay the packet to the application layer

Previous ***"SSR_unaware"*****:** no additional CSI

Current ***"SSR_aware"*****:** SSR indications as additional CSI

4. Carry out post-processing of received packet at the application layer for FEC decoding

Figure 1 Description of the Cross-layer System

traditional protocols have consistently shown significant performance improvements. Thus in this paper we restrict ourselves to comparing the proposed *SSR aware* scheme with one of the best available cross-layer schemes that does not depend on the SSR information.

2.2. 802.11b Residue Error Traces

The error traces used for our experiments are drawn from the channel modeling study presented in [2], [3], [4]. In particular we consider the traces in [4]. The traces in [4] were categorized into "Home" and "Office" traces to represent a low interference and relatively high interference environment. In [4] it was observed that the probability of packet corruption is negligible for SSR >14dB "Good Range" and the bit error rate for SSR < 8dB is very high "Bad Range". Thus the utility of cross-layer protocol is primarily exhibited when a significant proportion of the packets are received with SSR values in the "Transition Range" (8-14dB) [2]. Consequently we focus our analysis on residue error traces that were collected in channel conditions where atleast 20% of the packets were received with SSR indication in the Transition Range. Hence for the proposed work we chose 6 traces (3 from each environment) of 40,000 packets each. For details about the trace collection methodology please refer to [2], [3], [4]. However it should be mentioned that for the traces used in this paper, the 802.11b PHY data rate was always maintained at 11Mbps and the packet transmission rate was adjusted so as to facilitate a payload bit-rate of approximately 3Mbps in the "Home Data" and maximum possible bitrate lesser than 11Mbps achieved on the basis of flooding in the "Office Data". The packet payload size was maintained at 1024 bytes

2.3. Forward Error Correction Scheme

The FEC scheme used in this paper uses binary Low Density Parity Check (LDPC) codes. Our FEC block as shown in Figure 1, could be broken down into different codewords that were interleaved across packets. Each packet contributed 64 bits to a codeword. Thus as our packet size is 1024 bytes, the total codewords in a single FEC block were 128. Each LDPC codeword consisted of $k = 8192$ bits, thus each FEC packet block consisted of 64 Message packets. The number of parity packets are increased in steps of 4 packets, thus the number of parity bits in each codeword are increased in steps of 512 bits, so as to provide FEC blocks with packet-block-length varying from 68 packets to 128 packets. Thus we studied the performance of codes from rate 0.94 to rate 0.5. (Due to brevity we report results only for the maximal rates for which the eventual probability of decoding failure for the LDPC decoding was under ~15%). The Progressive-Edge-Growth [7] (PEG) algorithm based on large-girth tanner graphs was used to determine check matrices for all the LDPC codes. For background material on LDPC refer to [6] and references within.

2.4. Video Encoding

For the purpose of our video simulations we used streams compressed using H.264 [8]. To emulate the conditions where the wireless traces were collected we use a video stream with a bit-rate of 2.1 Mbps for the simulations on Home Data, and bit-rate of 5.4 Mbps for simulations on Office Data[3]. We report results for test sequences "mobile" and "stefan". The resolution of these sequences was "CIF". We use a GOP (IBPBP..) size of 15 frames, and a frame rate of 30 fps. Test sequences are repeated to provide a playout sequence of 900 picture frames. The compressed stream was mapped to the FEC scheme by embedding a video slice/ video packet in each codeword. LDPC codes can detect a decoding failure, thus if any codeword could not correct all the errors we drop that slice. All standard error concealment features were turned on. In the event of data/frame loss, the previous correctly received frame was copied and also used as a reference for the motion vectors.

2.5. Utilizing CSI in LDPC decoding

A numerically efficient version of the LDPC decoding algorithm uses the log-likelihood ratios (LLR) in estimating the transmitted code-word. In the LDPC decoding algorithm the LLR $L(c_i)$ associated with code bit c_i is initialized at the start of the decoding on the basis of the received bit y_i and an apriori assumption of the bit error probability of the i^{th} bit being P_e as

$$L(c_i) = (-1)^{y_i} \log\left(\frac{1-P_e}{P_e}\right) \qquad (1)$$

The message-passing algorithm (log-domain sum-product algorithm) is then iteratively used to update $L(c_i)$. A bit is

---

[3] The source-coding rate used here is chosen meagerly to represent the total amount of video content we would expect in a typical application that is using 3Mbps or the maximum possible bandwidth. Naturally in actual office application a source-coding rate of 5.4Mbps for "CIF" resolution may be too high. However multiple video sequences can indeed provide a combined content of 5.4Mbps. A single video sequence of 5.4Mbps could be considered to be an approximation for such total video content. Such approximations allow us to focus on issues more core to the focus of the proposed work.

decoded to 1 if $L(c_i) \le 0$ and 0 if $L(c_i) > 0$. For details and background on LDPC decoding please refer to [6].

An accurate initialization of LLR plays a key role in the performance of LDPC codes and thus improved CSI can help in improving the LDPC decoding. Thus, if a packet is received without any errors then $L(c_i)$ corresponding to the bits in that packet is obtained by setting $P_e = 0$; and similarly if a packet is dropped $L(c_i)$ is obtained by setting $P_e = 0.5$. In [2] for all the corrupted packets P_e was obtained from the average bit error rate. Thus in our experiments, for the *SSR_unaware* scheme, for all corrupted packets we set $P_e = \overline{\varepsilon(trace)}$, where $\overline{\varepsilon(trace)}$ represents the average probability of bit error in a corrupted packet for a given residue error trace. With extensive experimentation with 802.11b wireless traces [2], [3], [4] an empirical relationship between the bit error probability in a corrupted packet and the associated SSR indication has been established (especially see traces [2]). In an *SSR_aware* scheme this empirical relationship was used to obtain $P_e = \varepsilon(SSR)$.

3. EXPERIMENTAL ANALYSIS

Prior to presenting the results that shall exhibit the utility of the *SSR aware* scheme, to better understand the observations, it is important to develop some theoretical deductions and insight in the considered problem. Thus for this purpose consider the following notation (similar to [1]):

- For a packet received with a particular SSR indication let (i) $\delta(SSR)$ represent the probability of a packet being corrupted and (ii) $\varepsilon(SSR)$ represent the probability of a bit error in the corrupted packet.
- Let $f(SSR)$ denote the probability of receiving a packet with a particular SSR indication.
- Let Z be an indicator variable that indicates whether a packet is corrupted (i.e. Z corresponds to a checksum on the data payload) and SSR is a variable that indicates the Signal to Silence Ratio. Thus for a *SSR_unaware* scheme Z is made available as side-information to the receiver, while for an *SSR_aware* scheme, both Z and SSR are available as side-information.

Now let's assume a simplistic theoretical model, where we do not have any packet drops due to header corruption, the packet corruption process is memory-less and the bit-error process in a corrupted packet is also memory-less. For such a model, the application layer channel capacity of the *SSR_unaware* scheme and the *SSR_aware* scheme are given by equations (2) and (3). (proofs have been omitted due to brevity)

$$C_{SSR\_unaware} = (1-\delta) + \delta\left(1 - h_b(\varepsilon)\right) \quad (2)$$

$$C_{SSR\_aware} = (1-\delta) + \sum_{SSR} f(SSR) \cdot \delta(SSR)\left(1 - h_b\left(\varepsilon(SSR)\right)\right) \quad (3)$$

Note: $\delta = \sum_{ssr} f(SSR)\delta(SSR)$ & $\varepsilon = \sum_{ssr} f(SSR)\delta(SSR)\varepsilon(SSR)$ represent the overall probability of a packet being corrupted and the overall probability of bit error in a corrupted packet.

Using Jensen's Inequality/Convexity [9] it can be easily shown that $C_{SSR\_aware} \ge C_{SSR\_unaware}$. Thus the *SSR_aware* scheme should always perform better. However an *SSR_aware* scheme provides negligible improvement in capacity if (i) δ is very small, which is indeed the case for "Good" SSR values or (ii) $\varepsilon(SSR)$ is close to 0.5, which is indeed the case for "Bad" SSR values. (iii) $f(SSR) \cdot \delta(SSR)$ is such that most of the corrupted packets have the identical SSR indication.

Our experiments have been conducted on actual 802.11b wireless traces. In these traces neither the packet corruption process nor the bit error process is observed to be memory-less. However the above theoretical insight is still helpful in understanding the result of our experimentation tabulated in Table I. In Table I, similar to our deductions (i) and (ii), it can be observed that it is feasible in the Good and Bad SSR ranges for the *SSR_aware* scheme to provide negligible benefits in the error recovery performance. Nevertheless, in the transition region, the *SSR_aware* scheme consistently provides an improved error recovery performance. The % code failures are reduced by a minimum of 1.2% to a maximum of 11%. It is also important to note, that though it cannot be guaranteed that an *SSR_aware* scheme will be helpful in the Good Range, there do exist many instances when it does help. Similarly if the channel coding rate is reduced significantly (less than 0.33) then some performance benefit can be observed in the Bad Range too.

An extremely essential observation to make in Table I. is the SSR standard deviation (std. dev.) for the corrupted packets in each trace. The std. dev. was found to be atleast 3dB in all the traces presented here[4]. In the light of deduction (iii) the above stated observation is crucial to emphasize the practical utility of SSR indications. Furthermore, it should be stated that even when the SSR value averaged over a large number of (100 to 1000) packets is constant; the SSR indication on per packet basis can vary significantly. In such an event an *SSR_aware* scheme will provide performance benefits. The significance of the SSR std. dev. can be especially appreciated by observing the error recovery performance for trace 6. In trace 6 though the average channel conditions are such that the SSR values are "Good", the variation in SSR value is significant enough for an *SSR_aware* scheme to exhibit improved error recovery performance.

Finally it is important to verify that the improvement in error recovery performance does translate into improvement in video quality. Table I shows that the average PSNR quality over 900 frames typically improves by 1-2db. However, in many instances for large periods in the traces the SSR is entirely in the Good Range. Thus the average PSNR does not tell the complete story. Hence for a better evaluation, Figure 2 (a) and Figure 3 (a) provide temporal snap shots for some experiments on trace 1 and 4. In the temporal snapshots it can be clearly observed that there can be multiple GOPS over which the PSNR of an *SSR_aware* scheme is better than an *SSR_unaware* by 5-15dB. Such a big difference in video quality is perceptually visible and a typical difference in block-distortion in the two schemes is exhibited by the frame captures in Figure 2 (b) and Figure 3 (b). In addition to the block distortion, the motion discontinuity in an *SSR_unaware* scheme was also significantly higher. It should be also noted that by

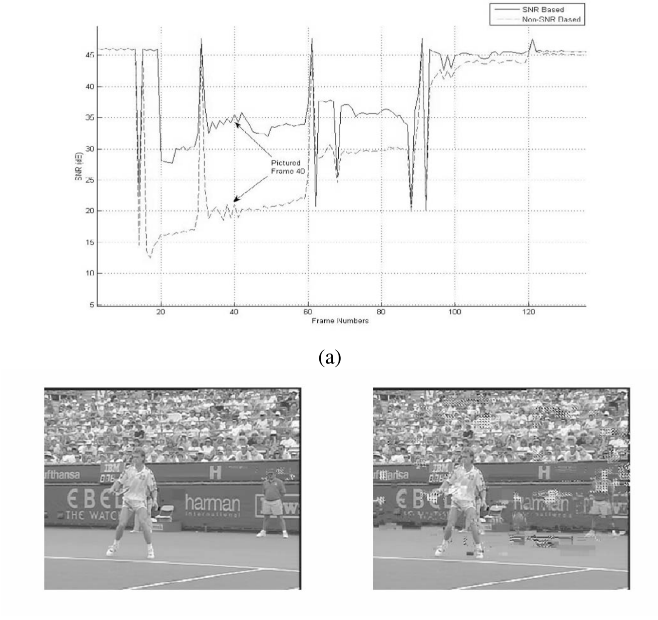
---

[4] In a wider study, in our extensive experimentation with 802.11b over millions of packets we have observed that SSR deviation is always significant. This deviation can be observed to increase due to presence of walls or mobility. Detailed presentation of statistical analysis of a larger set of residue error traces is outside the scope of this paper,

Table 1 Improvement in Cross-layer Performance: Error Recovery and Video Quality

Tr. No.	Env.	SSR Std. Dev.	Coding Rate		% Code Failures$^\perp$								PSNR Gain in dB	
			Src.	Chnl.	Good		Trans.		Bad		Total		Stefan	Mobile
1	Home	3.2 dB	2.1 Mbps	0.8	-	-	0.5	2.5	-	-	0.5	2.5	5.5	6.78
2		3.1 dB		0.66	3.6	5	4.0	7.8	-	-	3.9	7.5	0.93	0.75
3		7.7 dB		0.94	5.4	7.0	14	16	-	-	11	13	1.19	0
4	Off	3.8 dB	5.4 Mbps	0.8	0.0	0.0	7	18	100	100	6	16	1.6	2.3
5		4.1 dB		0.66	0.0	0.0	1.9	3.1	100	100	1.7	2.8	0.78	2.07
6		7 dB		0.5	3.8	6.0	-	-	-	-	3.8	6.0	0.82	1.37

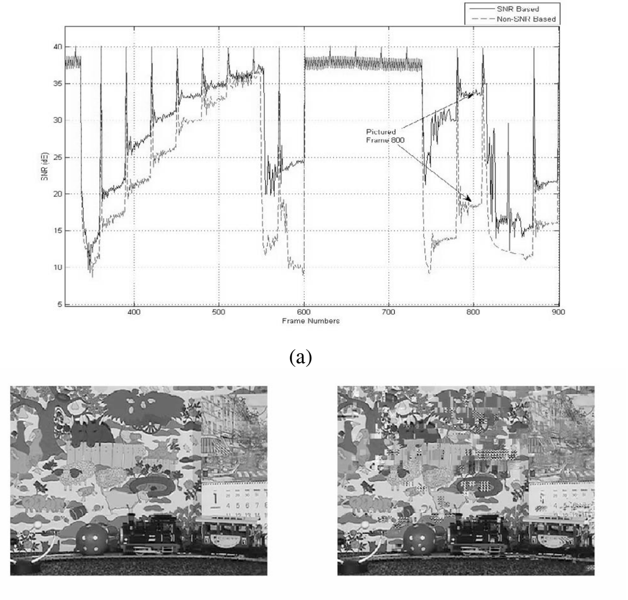
$^\perp$The results in the sub-columns Good, Transition and Bad are obtained by classifying an FEC packet block based on the SSR averaged over the block. Also note that the highlighted sub-columns present the results corresponding to the "Previous *SSR Unaware* Scheme".



(a)

(b)

Figure 2 Stefan 5.4Mbps over trace 4. (a) temporal snapshot (b) frame 40. **Left:** with SSR CSI, **Right**: without SSR CSI.



(a)

(b)

Figure 3 Mobile 2.1Mbps over trace 1. (a) temporal snapshot (b) frame 800 **Left:** with SSR CSI, **Right**: without SSR CSI.

observing the performance of trace 1 in Table I, we can conclude that in the transition region we can get an average PSNR improvement of 5-6db.

4. CONCLUSION

In this paper it has been shown that the (per packet) SSR indications are capable of providing robust CSI. This robust CSI is useful in improving the error recovery performance of a cross-layer FEC scheme and video throughput over cross-layer. In the good and bad SSR range the utility of the proposed scheme can be diminished but in the transition region the performance of an *SSR_aware* scheme consistently provides improvement. If the channel conditions are such that the SSR is in the transition region for large periods, 5-6 dB gain in terms of average PSNR quality of video can be obtained.

5. REFERENCES

[1] S. Karande and H. Radha, "The Utility of Hybrid Error Erasure LDPC (HEEL) Codes for Wireless Multimedia," IEEE ICC, May 2005.

[2] 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-Bitrate Real-Time Multimedia," Signal Processing: Image Communication, vol.18, no.7, pp. 575-595, August 2003.

[3] S.A. Khayam and H.Radha, "Linear-Complexity Models for Wireless MAC-to-MAC Channels," ACM/Kluwer Wireless Networks (WINET) Journal - Special Issue on Selected Papers from MSWiM'03, vol. 11, no. 5, pp. 543-555, September 2005.

[4] S. Karande, U. Parrikar, K. M. Misra and H.Radha, "On Modeling of 802.11 Residue Errors," CISS, March 2006.

[5] S. A. Khayam, M.U.Ilyas, K.Pörsch, S. Karande, and H. Radha, "A Statistical Receiver-based Approach for Improved Throughput of Multimedia Communications over Wireless LANs," IEEE ICC, May 2005.

[6] W. E. Ryan, "An Introduction to LDPC Codes," in CRC Handbook for Coding and Signal Processing for Recording Systems (B. Vasic, ed.) CRC Press.

[7] X-Yu Hu, E.Eleftheriou, D.M. Arnold, "Regular and Irregular Progressive Edge-Growth Tanner Graphs", IEEE Transactions on Information Theory, vol. 51. no. 1, pp. 386-398, 2003.

[8] 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-G050, March 2003.

[9] T. M. Cover and J. A Thomas, "Elements of Information Theory," Wiley Series in Telecommunications.