Utility Of Signal To Silence Ratio As A Predictive Tool For Enhancing Cross-layer Multimedia Protocols[†ϒ]

Shirish S. Karande, Kiran Misra, Utpal Parrikar and Hayder Radha
Department of Electrical & Computer Engineering / 2120 Engineering Building
Michigan State University, East Lansing, MI 48824 USA
{karandes, misrakir, prabhupa, radha}@egr.msu.edu

Abstract – Wireless multimedia applications can benefit significantly by employing cross-layer protocols that recover information from corrupted packets. The performance of such protocols can be improved with the help of Receiver-side Channel State Information (CSI). However, in practice, the bits observed at the link-level do not have an inherent soft-information associated with them. Hence standard physical-layer link quality indications (e.g. the signal strength of each individual bit) cannot be used as CSI. Due to the lack of a practical predictive tool, many standard optimizations for capacity planning, soft-decoding, rate control etc., cannot be practically employed with the considered cross-layer protocols. In this paper we address this important engineering problem. In practice, 802.11b radio devices are capable of measuring the signal power for the first few microseconds (μs) of each packet reception. This measurement can be used to associate a coarse Signal-to-Silence (SSR) ratio with each individual packet. This work, investigates the practical predictive utility of such coarse measurements. We show that SSRs can be used to provide meaningful CSI with a reasonably link/infrastructure invariant model training. Depending upon the specific application, the empirically determined correlation between the SSR indications and the Bit Error Rate (BER) can be utilized in a realtime or non-realtime fashion. We consider two sample applications to illustrate the practical predictive utility of SSRs: (a) We show that SSRs can be used for *cross-layer network planning*. (b) As the primary contribution of this work, we show that SSRs can be used to provide realtime CSI to improve the error-recovery performance of cross-layer Forward Error Correction (FEC) schemes, especially under dynamic channel conditions. H.264 based simulations are used to exhibit the implied improvements for multimedia applications.

Index Terms- **channel capacity, channel coding, video coding, wireless networks, channel modeling**

[†] Part of this work has been presented as [21]-[22].

[ϒ] This work was supported in part by NSF Award CCF-0515253, NSF Award CNS-0430436, MEDC Grant GR-296, and unrestricted gift from Microsoft Research.

I. INTRODUCTION

The traditional link-level abstraction dictates that all corrupted packets should be discarded. Unlike the wired Internet based communication, the number of packet drops due to bit errors can be substantial in wireless networks. Video applications are usually extremely bandwidth hungry and get adversely affected by excessive reduction in bandwidth due to packet drops. In typical multi-room home/office settings, there exist many clients that do not have a Line of Sight (LoS) communication path with the wireless Access Point (AP). Such clients often experience significant attenuation in the received signal. In addition, the fading effects and, the variation in channel quality there of, experienced by such clients is non–trivial. The video quality available to such clients can be severely limited on account of excessive packet drops.

The impact of bit corruptions, on the effective bandwidth and thus the video quality, can be substantially reduced by abandoning the traditional link-level abstraction. Many recent research studies (e.g. [10]-[20]) have proposed cross-layer multimedia protocols that relay corrupted packets to higher layers. The utility of such protocols has been observed in a variety of multimedia communication systems: closed (e.g.[19]) as well as open loop (e.g. [10],[12]) architectures, single cell networks (e.g. [11],[14]) as well as multi-hop ad-hoc networks (e.g. [15],[18]) and voice (e.g.[13]) as well as video (e.g.[14]-[16]) applications In addition, experiments with actual 802.11b WLAN deployments have consistently shown that information recovery from corrupted packets can lead to significant improvement in connectivity [20] and multimedia throughput ([14]-[16]). Therefore prior studies have generated substantial motivation to design future multimedia protocols on the basis of a paradigm where corrupted packets do not get dropped.

Despite their early promise, the above described cross-layer protocols, suffer from some important drawbacks. More specifically, a number of standard multimedia optimizations for rate-control, error-resilience, resource allocation, etc. have not been extended to work with the cross-layer framework. We believe that the lack of such an initiative can be primarily attributed to the absence of a predictive tool, which can be used for link-quality estimation. The work presented in this paper is motivated by the above observation. Hence, it is critical to understand the importance of link-quality estimation and the difficulties associated with practically estimating

the same when cross-layer protocols are employed. In the following paragraphs we elaborate upon this issue to clearly motivate the presented work.

In wireless networks the capacity of a link can vary with space and time. The link-capacity and quality plays an important role in determining the resource allocation in wireless networks. For example, consider the work by Majumdar et. al. in [1] related to design of unequal error protection (UEP) for video multicast. The design algorithm presented in [1] requires knowledge about the link-capacities of the users. Thus knowledge about link-capacity is useful for network planning. Additionally, since the link-quality can vary with time, channel state information (CSI) available on a more realtime basis can be utilized for network adaptation. See the contributions in [2]-[9] for examples where link-quality is utilized for rate-control, adaptive modulation, server allocation, routing, joint source channel coding etc.

The above discussion highlights the importance of estimating link-quality. Under the traditional link-abstraction, the channel impairments observed at the link-level consist only of packet erasures. Thus the link quality is completely determined by the Packet Erasure Rate (PER), δ. In practice, it is easy to identify the packets that are dropped and hence the PER also can be easily determined. Consequently, a number of link-quality based optimizations can be readily employed with the traditional protocols. However, when cross-layer protocols are employed, the link-level channel is impaired by bit-corruptions. Thus the channel capacity is determined by the PER as well as the Bit Error Rate (BER) in the corrupted packets. The difficulty of determining the BER emerges from the fact that unlike a packet drop, a bit corruption is not "*observable*", i.e. a decoder is not aware of the exact location or count of bit corruptions. Additionally, the bits at the link-level are simple ones and zeros; these bits do not have any inherent soft-information, such as the signal-strength, associated with them. Hence, it is essential to identify a practical mechanism to estimate the BER in the corrupted packets. Addressing this important problem is the primary focus of this paper.

As has been elaborated above, in practice, the signal-strength of each bit or the BER is not observable. However, there may exist other variables such as the traffic load, the PER, checksum etc. which are observable. The correlation between these observable variables and the BER can thus be exploited for channel estimation.

Hence, identification of suitable observable variables is one of the most important aspects of realizing a practical mechanism for link-quality estimation. The work in this paper is motivated by the observation that commercial 802.11b devices are capable of associating a coarse measure, *Signal-to-Silence Ratio* (SSR), with each received packet. The SSR value is evaluated on the basis of a *Received Signal Strength Indication* (RSSI) which measures the signal strength during the preamble stage of the packet reception, and the *Silence Value* which measures the signal strength just prior to the packet reception. The objective of the presented work is to determine the efficacy of the CSI provided by these coarse SSR indications. The key steps in the contributions of this work are:

- **Correlation Modeling:** As the first step, we seek to establish a relationship between SSR indications and BER. In principle, we can analytically deduce such a relationship by assuming some channel model and device specific detector model (i.e. a model for the physical layer processing done at an 802.11b wireless receiver). Such an analytical approach is critically dependent on the accuracy of the device model. Accurate device models are typically hard to determine and additionally cumbersome to analyze. Hence, in practice it is significantly easier to empirically identify the behavior of a device. Therefore we utilize actual 802.11b measurements to establish a device specific empirical relationship between SSR and BER. In general, the determination of such an empirical relationship has a receiver-specific training overhead associated with it. However our analysis will show that there exist scenarios where the empirical relationship can be determined in a reasonably link/infrastructure independent fashion. Therefore, based on our observations, we believe that, under a number of practical scenarios, the training overhead can be kept within modest limits

- **Cross-layer Network Planning:** The above determined relationship can be used to model the link-level error behavior with a Compound Binary Symmetric Channel (CBSC). The capacity of the CBSC can thus be used to estimate the cross-layer link capacity. These capacity estimates can in turn be used for cross-layer network planning. In general cross-layer network planning can include optimization of unequal error protection, retransmission limits, bandwidth and other resource allocations. Due to brevity, we do not consider such optimizations in detail, but limit ourselves to a simplistic yet representative illustration of the predictive utility of SSRs in network planning. With the example of a Reed-Solomon (RS) based FEC scheme, we show that SSR indica-

tions can be used in a non-realtime fashion to predict (a) the suitability of a channel coding rate for an individual or group of receivers, (b) the utility of cross-layer protocols for various receivers

- **Decode-side CSI:** In a WLAN, the radio-link quality can vary significantly on account of multi-path reflections, fading, receiver mobility, interference and variations in the surroundings. In such environments the BER in each packet can vary significantly. Information about the BER in a packet can help in enhancing the error recovery performance. In [15] we exhibited that even simplistic CSI, available from checksums, can significantly improve the efficiency of cross-layer protocols. In this work, we show that SSR indications can be used to provide further performance gains. We demonstrate this utility by considering an architecture where Low Density Parity Check (LDPC) code based FEC is used for transport of H.264 compressed video. On the basis of trace based simulations, we show that the CSI available due to SSR indications can significantly improve the error recovery performance of LDPC decoding algorithms. This improved error recovery performance is shown to provide substantial gains in multimedia quality, especially when the variations in link-quality are significant.

The organization of the remainder of this paper is as follows: Section II develops important theoretical preliminaries to facilitate an easy understanding of the discussions in the following sections. In this section we define the notion of a CBSC. We utilize some of the classical work by Stambler [25] to develop capacity expressions for a CBSC, with or without decoder-side CSI. In Section III of this paper we provide details about the methodology employed to collect the wireless error traces. It is important to highlight that all claims in this paper have been validated on the basis of simulations with actual 802.11b wireless error traces and actual measurements of the SSR. In Section IV we deduce an empirical relationship between SSR indications and the probability of bit error in a corrupted packet. This relationship is used to deduce SSR dependent capacity measures. We provide experimental and theoretical analysis exhibiting the efficacy of using these capacity measures for network planning. In Section V we explore the utility of SSR as a realtime predictive tool. We show that CSI available from SSRs can improve capacity. Additionally, on the basis of trace based simulations, we also exhibit the utility of SSRs in improving the error recovery performance and the multimedia quality. Finally in Section V, we provide a summary of some of the key conclusions of this work and identify important future goals.

II. PRELIMINARIES AND OVERVIEW

A. Compound Binary Symmetric Channel Model for MAC-to-MAC Channels

In this work we model the channel from the Medium Access (MAC) layer of the transmitter to that of the receiver as a CBSC. This model is used in the error recovery algorithms (as outlined in section II.A and V.B.2) and also for estimating the link-capacity (as outlined in section II.B, section IV.B and section V.A.2). Thus, in this work, we define a CBSC as follows:

Definition 1: A Compound Binary Symmetric Channel (CBSC) is a discrete memory less *symmetric* channel defined over a binary input ($X = [0,1]$) and binary output ($Y = [0,1]$) alphabet. The channel behavior is described by a probability mass function (p.m.f) $f(s)$ on a set of states S, where each state $s \in S$ corresponds to a BSC with cross-over probability ε_s.

In the subsequent sections, it will be shown that we can associate a *SSR* indication and a checksum indication *Z* with each received packet. Here *Z* is just a binary indication such that *Z* = 0 when the packet is corrupted and *Z* = 1 when the packet does not contain any bit-corruptions. We show that the (Z, SSR) tuple can be used to define the states of a CBSC channel. Furthermore, in section IV.A, on the basis of actual measurements, we train each of the (Z, SSR)-BSC states, i.e. we associate a probability of bit error $\varepsilon_{(Z,SSR)}$ with each state. These trained models are then used to describe the channel behavior observed by each receiver in terms of an equivalent CBSC. In section IV.A we argue that the BSC parameter $\varepsilon_{(Z,SSR)}$ associated with each state (Z, SSR) can be assumed to be an (link) invariant. Consequently, the state definitions of the equivalent CBSC models remain identical for all the receivers. However the p.m.f governing the frequency of the states can vary. In practice, we can observe the frequency of the (Z, SSR) indications to determine the p.m.f $f(\cdot)$ associated with each receiver. As shall be shown in the following sub-section, this p.m.f can be used along with the cross-over probabilities of each state to determine the link-capacity. In section IV we also show that such capacity estimates can be used for network planning. Additionally, since each packet has an (Z, SSR) indication associated with it, our

trained models can also be used in realtime for estimating the channel state in each packet. In section V we use such realtime CSI to associate soft-confidence values with the received bits. These soft-confidence values are then used to improve the error recovery performance of LDPC decoding algorithms.

B. Capacity of CBSC and the role of side-information

In the absence of any channel state information, the CBSC reduces to a BSC with cross-over probability $p = \sum_{s \in S} f(s) \cdot \varepsilon_s$, which represents the cross-over probability averaged over all the states. Thus the capacity of a CBSC is given by

$$C = 1 - h_b(p) = 1 - h_b\left(\sum_{s \in S} f(s)\varepsilon_s\right) \tag{1}$$

However, it can be shown that in presence of decoder-side channel state information the capacity increases. We refer the reader to the work in [24]-[26] for detailed discussion on capacity in presence of side-information. As discussed in [24]- [26], the core insight behind evaluating the capacity of a CBSC in presence of decoder-side CSI is that its capacity is equivalent to a channel whose input is given by $x \in X$ while the output is given by $(s, y) \in S \times Y$. Thus, the capacity of CBSC with decoder-side channel state information is given by

$$C_{CSI} = \sum_{s \in S} f(s)\left(1 - h_b(\varepsilon_s)\right) \tag{2}$$

Furthermore, in practice, under certain protocol designs, the CSI may be available to a receiver in a *lumped* form. For example, in our prior work in [15], we utilize only the checksum as side-information. This is equivalent to the information about states $(Z = 0, SSR)$ being lumped into a single a state (Z = 0) and that of states $(Z = 1, SSR)$ being lumped into a state (Z = 1). The lumped states can be represented by a decomposition of the set S into mutually disjoint and collectively exhaustive subsets $\{S_i\}$, i.e. $S_i \subseteq S$, $\bigcup_i S_i = S$, $S_i \cap S_j = \phi$. Thus, when CSI is available to a receiver in a lumped form the decoder does not have exact state information, but is aware of the subset S_i to which the current channel state belongs. The probability of occurrence of a lumped

state can be expressed as $f(S_i) = \sum_{s \in S_i \subseteq S} f(s)$. Similarly the average probability of bit-error in each lumped state can be expressed as $\varepsilon_{S_i} = \sum_{s \in S_i \subseteq S} f(s)\varepsilon_s$. The capacity of CBSC with such lumped side-information is equivalent to the capacity of a CBSC channel with states $\{S_i\}$ and perfect decoder-side CSI. Hence the capacity can be expressed as

$$C_{CSI-lumped} = \sum_i \left(\left(\sum_{s \in S_i} f(s) \right) \left(1 - h_b \left(\sum_{s \in S_i} f(s)\varepsilon_s \right) \right) \right) = \sum_i \left(f(S_i) \left(1 - h_b \left(\varepsilon_{S_i} \right) \right) \right) \quad (3)$$

At this point, it should be noted that the capacity function is a convex function. Jensen's Inequality [23] states that for any convex function $g(\cdot)$, $E[g(\cdot)] \geq g(E[\cdot])$ with equality occurring if and only if $g(\cdot)$ is constant over the entire *domain*. Thus with the help of Jensen's Inequality we can easily show that capacity of CBSC increases with decoder side CSI. The capacity gain is reduced if the state information is lumped. Therefore we can write:

$$C_{CSI} \geq C_{CSI-lumped} \geq C \quad (4)$$

where $C_{CSI} = C_{CSI-lumped}$ if and only if all the states in every lumped state are identical, i.e. $\forall\, i, a, b$ such that $a, b \in S_i$ we have $\varepsilon_a = \varepsilon_b = \varepsilon_{S_i}$. Similarly, $C_{CSI} = C_{CSI-lumped} = C$ if and only if all the states of a CBSC have an identically description. Consequently, the gains provided by side-information increase with variations in channel behavior, i.e. with variations in the cross-over probability.

C. Utilizing CSI is decoding of Low Density Parity Check codes

In the past decade LDPC codes have exhibited record breaking performances on a variety of channels (see [27]-[28] and references within). Hence, the FEC schemes employed in section V of this work are also based on LDPC codes. The exact decoding of LDPC codes is an NP complete problem. However, belief propagation based algorithms used for LDPC decoding [27] have exhibited efficient performance. In this work we utilize a numerically efficient version of the sum-product decoding algorithm that uses the log-likelihood ratios (LLR) in

estimating the transmitted code-word. The input to this algorithm consists of an initial belief in the value of a code bit. This initial belief is derived on the basis of the channel output and the bit-error probability. Therefore, for the considered algorithm, the initial belief $L(c)$ in a bit c can be expressed by the following LLR:

$$L(c) = \log\left(\frac{p(c=0|y)}{p(c=1|y)}\right) = (-1)^{y} \log\left(\frac{1-p_e}{p_e}\right) \tag{5}$$

where y represents the received bit value and p_e represents the probability of bit corruption. The sum-product algorithm utilizes these initial beliefs to iteratively improve the estimate of the codeword. Due to brevity we cannot include the details of the sum-product algorithm and refer the reader to [27] for an introduction to LDPC codes. However, it should be highlighted that the error recovery performance of the sum-product algorithm is critically dependent on the accuracy of the initial beliefs. Thus CSI information can be utilized to improve the accuracy of the initial beliefs and hence the error recovery performance. When LDPC codes are employed on a CBSC channel, the probability p_e of a bit being flipped is dependent on each state. In the absence of any side-information the LLRs for all the bits are determined on the basis of the average probability of error (seen over all possible states). However, with side-information we can use a more accurate value of p_e for each bit; more specifically the value of p_e can be set to either ε_s or ε_{S_i} depending upon whether CSI is available directly or in a lumped form. In section V we utilize the CSI information available from the *SSR* and *Z* indications to de-duce accurate initial beliefs.

III. TRACE COLLECTION METHODOLOGY

In order to demonstrate the practical efficacy of the schemes proposed in this work, we base our analysis on actual wireless error traces. As shown in Figure 1 our experimental setup consisted of an 802.11b Access Point (AP) operating in Distributed Coordination Function (DCF) mode with RF output power set at 18dBm. A station is connected to the AP using 100Mbps Ethernet and acts like a server, while another wireless station serves as a Line of Sight (LoS) client. The LoS client is placed in close proximity to the AP to avoid any packet drops and hence any packet retransmissions. We utilize a third wireless station to collect the required traces. This station

acts as "sniffer" and overhears all the packet transmissions to the LoS client. At the sniffer we used a DWL 122 wireless card based on Prism 2.5 chipset with linux-wlan-ng-0.2.1pre21 device driver [29]-[30]. The Prism based card enables operation in "monitor" mode which enables delivery of all the MAC frames without any filtering, including those with failed Frame Check Sequence (FCS). The source code of the "sniffer's" device driver was further modified to dump all the corrupted/uncorrupted frames from the kernel buffer space. The traces collected at the sniffer and the LoS clients were compared offline to obtain the error trace. We embedded specific bit patterns in each of the transmitted packet to enable the identification of all the corrupted packets. Packet dissectors were implemented to ensure that only packets pertinent to our wireless experiment are processed.

The Prism2.5 device measures *Received Signal Strength Indication* (RSSI) value and "*Silence*" value at the antenna of the radio hardware. The RSSI is measured for 10μs[1] during the preamble stage of MAC frame reception. The *Silence* value measures power before the start of the frame. In the radio device that we used, both these values are recorded in a single byte representing the signal strength in dbm. We define the SSR to be the difference between the RSSI value and the *Silence* value. Thus we associate an SSR measure in dB with each of the received packets. In addition, we also associate a binary indication Z with each packet. The value of Z is determined on the basis of a checksum failure. As mentioned previously, we set $Z = 0$ if the packet is corrupted and $Z = 1$ if the packet is uncorrupted.

The work environments where traces were collected have been classified into "*Home"* and "*Office"* setup. This nomenclature is primarily determined by location of the author's home and office. Both these locations were separated by a distance greater than 3 miles and thus there wireless environments are assumed to be completely uncorrelated. To assist further visualization, Figure 2 provides a rough sketch of both the wireless environments:

Home Setup: This setup consisted of an almost interference free home environment with one or two neighboring APs on (widely separated) channels. The specific house in which we collected the traces is a two-storey town

[1] The duration for which the signal strength is measured is determined by the radio-hardware and cannot be altered.

house with a basement. The house is located at a corner and thus the interfering APs (if any) were at a significant distance from the sniffer. The walls of the house were wooden and hence as shown in Figure 2 the sniffer machine and the AP were separated by 0 to 3 wood walls.

Office Setup: Our office setup is located in the Engineering Building at Michigan State University. Thus our collection setup was surrounded by 3 to 4 neighboring APs. During our trace collection, we ensured that none of these APs were operating on a channel identical to our experiment. However, due to the high density of the APs, it was infeasible to completely avoid co-channel interference.

The traces utilized in this paper are a part of a larger effort to collect wireless measurements. However for the purpose of this draft we consider about 2 million packets collected in each of the setups. These packets were transmitted using the 802.11b PHY data rate of 11Mbps. The packet transmission rate was maintained at 384 packets/sec in the Home Setup and at nearly 1000 packets/sec in the Office Setup. Thus we collected traces for a total time of ~87 minutes at home and ~34 minutes at the office. These traces were not collected in a contiguous manner. The total trace collection time was broken into contiguous collection times varying between 3 to12 mins. In each of the contiguous periods, the traces were collected by a human subject holding the laptop at waist height. The human subject had a meandering mobility (less than 2-10feet/minute).

In the subsequent sections, we exhibit that the utility of cross-layer protocols is the most pronounced when the SSR values are in the range of 8-14dB. We deduced this fact from some preliminary analysis and thus a deliberate effort was taken to concentrate most of our trace collection effort in this SSR range.

IV. CROSS-LAYER NETWORK PLANNING

In this section we evaluate the empirical relationship between SSR indications and the link-level channel impairments. This relationship is utilized to associate a protocol dependent capacity measure with each SSR value. We then show that such capacity measures can be used to provide predictive utility for network planning.

A. Relationship of SSR with Link-level Impairments:

As mentioned in section II, we model the link-level channel as a CBSC. The states of the CBSC are defined on the basis of the (Z, SSR) tuple. Thus in this section we are primarily interested in associating a BSC parameter $\varepsilon_{(Z,SSR)}$ with each of the states. In addition, we shall also be interested in evaluating the conditional probability $\delta_{(SSR)}$ of a packet being corrupted given an SSR value. This conditional measure will be utilized in the following sub-section, to establish a relationship between capacity and SSR. Note that $\delta_{(SSR)}$ represents the probability of a packet being dropped by a traditional protocol.

Our trace collection methodology associates a (Z, SSR) -tuple with each received packet. Hence the packets in our traces can be classified on the basis of the (Z, SSR) label. Thus $\varepsilon_{(Z,SSR)}$ can be obtained by averaging the observed BERs over all the packets with a given (Z, SSR) label. Additionally, we evaluate the frequency $f(\cdot)$ of each state (Z, SSR). These frequencies can then be used to determine the $\delta_{(SSR)}$ in a straight forward manner using the following expression:

$$\delta_{(SSR)} = \frac{f(Z=0, SSR)}{f(Z=0, SSR) + f(Z=1, SSR)} \tag{6}$$

***Loss Rate*:** Figure 3 shows the result of empirically evaluating $\delta_{(SSR)}$. It can be observed that above 14dB the probability of packet being corrupted is less than 10%, while below 8dB the probability of a packet being corrupted is almost 100%. Thus we refer to SSR values below 8dB as the *bad range*, the values above 14dB as the *good range* and the values from 8 to 14dB as the *transition range*. Cross-layer protocols are required only when a significant number of packets get dropped due to bit corruptions. Hence from here on we focus our analysis primarily on channel condition where the SSR value is less than 15dB.

***Bit-Error Rate*:** A checksum indication of Z = 1implies that a packet is error free. Hence, the BSC parameter associated with each state $(Z=1,SSR)$ is set to zero. Consequently, for notational convenience we shall often represent the BSC parameter $\varepsilon_{(Z=0,SSR)}$ by $\varepsilon_{(SSR)}$. Figure 4 shows the result of empirically evaluating $\varepsilon_{(SSR)}$.

From Figure 3 and Figure 4 it can be observed that the relationships $\delta_{(SSR)}$ and $\varepsilon_{(SSR)}$ are reasonably infrastructure invariant. Such infrastructure invariance may not be valid for all possible classes of home and office environments. However, we would like to highlight that the Home and Office setups chosen for this paper have crucial differences:

- The topology of these setups, determined by the size of the rooms, walls and other objects are completely distinct. (For example, in the office environment, there is a significant number of metal objects, such as desks, file cabinets, etc, which cause significantly higher number of reflections and multi-path effects when compared to the "Home" environment.)
- The packet transmission rates (used to collect the error trace) are also completely distinct.
- The number of interfering APs surrounding our setups are very different
- A distance greater than 3 miles separates our home and office setups. Hence we believe that the wireless environments of these infrastructures are completely uncorrelated.

Therefore, on the basis of empirical evaluations, we believe that in practice it should still be feasible to determine a common single model for a large variety of links/infrastructures. Whenever such a situation is not feasible, we recommend the use of an infrastructure specific channel model.

B. Capacity gain of cross-layer protocols:

The above derived empirical relationships can be used to estimate the *capacity gain* provided by cross-layer protocols at various SSR values. As discussed previously, we determine the inherent link-level capacity on the basis of an equivalent CBSC. Thus, if we assume that the SSR indications observed on a link remain constant then the frequency of states $(Z=0,SSR)$ and $(Z=1,SSR)$ in an equivalent CBSC are given by $\delta_{(SSR)}$ and $1-\delta_{(SSR)}$ respectively. Therefore, in accordance with equation (1), the inherent link-level capacity is given by

$$C_{inherent}\left(ssr\right)=C_{cross-layer}\left(ssr\right)=1-h_b\left(\sum_{(Z,ssr)} f\left(Z,ssr\right)\varepsilon_{(Z,ssr)}\right)=1-h_b\left(\delta_{(SSR)}\varepsilon_{(SSR)}\right) \quad (7)$$

In general cross-layer protocols cannot recover information from all the corrupted packets. Under certain circumstance the identity of packet cannot be established and hence the packet has to be dropped. Hence in general the capacity of cross-layer protocols is lesser than the inherent capacity. However, techniques such *Header Detection* [16] can be used to reduce the proportion of such packet drops to negligible levels. Therefore in this paper we assume that the capacity of cross-layer protocols is equivalent to the inherent capacity. Furthermore, note that the capacity under a traditional protocol (i.e. a traditional link-layer abstraction) is completely determined by the loss rate. Therefore, as a function of SSR, the capacity of a traditional protocol is given by:

$$C_{traditional}\left(ssr\right)=\left(1-\delta_{(SSR)}\right) \quad (8)$$

Figure 5 shows the capacities evaluated on the basis of the above equations. It can be observed that the cross-layer protocols provide negligible capacity gain in the good range and for SSR values lesser than 5dB. However, for the transition range and parts of the bad range (i.e. from 5-15dB) the increase in capacity can vary from a minimum of 0.05 to a maximum in excess of 0.5. Thus it is evident that the utility of employing a cross-layer protocol can vary significantly with change in the average SSR. Such variations in utility have to be taken into consideration while planning a network. In section V we illustrate the utility of the relationships established in Figure 5 for *cross-layer network planning*.

C. Cross-layer network planning

In this section, we demonstrate the utility of SSR as a predictive tool for network planning. To do so we emulate the communication setup described by Figure 7. In Figure 7 a server is transmitting (video) application packets, protected by a *systematic* Reed-Solomon (RS) code based FEC, to receivers placed in diverse locations in a network. The channel between the transmitter and various receivers is simulated on the basis of 4 trace samples each, of 20, 000 packets, collected in the Home and Office *setup*. The traces from the Home Setup are represented by (H1, H2, H3, H4), while those from the Office Setup are represented by (O1, O2, O3, O4). Our

objective is to show that the SSR statistics of each trace are sufficient to predict (a) the suitability of the chosen channel coding rate (b) the necessity for employing cross-layer protocols. The necessary statistics associated with each trace have been recorded in Table II.

The FEC schemes used in this paper are realized by interleaving codewords across packets. These FEC schemes can be completely characterized by, N the packet block length, the code length n, code rate R, the packet size s in bits and the symbol size b (in bits) on which the code has been implemented. Thus each FEC block consists of $(N.R)$ message packets and $(N \cdot s)/(b.n)$ codewords. The parameters chosen for the RS based FEC used in this section are $N = 100$, $n = 200$, $R = 0.8$, $s = 8192$ and $b = 8$. We evaluate the performance of such a FEC scheme, in conjunction with cross-layer as well as conventional protocols. The decoding algorithm used in conjunction with a traditional protocol is an erasure decoding algorithm, while that employed with cross-layer protocols is a hybrid erasure-error decoding algorithm. Neither of these algorithms utilize any realtime CSI. Table I shows the performance of the FEC scheme in terms of $goodput = 1 - P_d$ where P_d is the probability of error in decoding a codeword.

From Table I it can be clearly seen that the utility of a cross-layer approach can vary greatly. Observations on traces:

(a) H1 and O2 exhibit significant utility for cross-layer protocols.

(b) H4 and O1 would lead to a claim of moderate improvement.

(c) H3 and O4 also show moderate improvement, but the performance of both protocols is poor.

(d) H2 and O3 would lead us to believe that there is no need for cross-layer protocols.

Studies prior to this work (e.g. [14], [16]) have completely ignored the role of SSR or SNR in influencing the utility of cross-layer protocols. Cross-layer schemes such as the ones considered in this paper always add additional architectural and processing complexity. A system administrator or network designer may choose to employ cross-layer protocols only when necessary, or may choose to place a workstation with higher computational capability in locations where cross-layer protocols are essential. Thus variations in performance, exhibited by

the RS based FEC, if not explained/predicted appropriately may make the task of cross-layer network planning very cumbersome.

The relationship established in the previous subsection can now be used to predict/explain the variation in the utility of cross-layer protocols. On the basis of Figure 5, we should not expect any advantage in employing cross-layer protocols if a large percentage of the packets are in the good and bad range. In other words, for cross-layer protocols to be of any practical utility, they should be employed only when the channel conditions are primarily in the transition range. Moreover, even though cross-layer protocols provide capacity gains in a part of the bad range, this performance gain will not be realized with a coding rate of 0.8. Consequently from Table II it can be clearly seen that cross-layer protocols when employed over traces:

(a) H1 and O2 show significant benefit in performance because almost 2/3 of the packet blocks are in the transition range

(b) H4 and O1 show moderate benefit in performance because the number of packets in the transition range is lesser (about 1/3)

(c) H2 and O3 show negligible improvement in performance because almost all packets are in the good range.

(d) H3 and O4 provide a performance better than traditional. However the goodput is still very poor because a large percentage of the packets are in the bad range.

Thus with an appropriate site-survey a network designer can clearly identify the utility of employing cross-layer protocols in different locations in the network.

V. UTILITY OF SSR AS A REALTIME PREDICTVE TOOL

The channel conditions observed on a wireless link can frequently deviate from the average behavior. Consequently the channel impairments endured even by temporally adjacent packets can vary significantly. Our discussion in section II implies that the decoder-side CSI can lead to significant capacity gains under such varying channel conditions. In our prior work [15] we have considered a cross-layer scheme which utilizes only the checksum indication Z as side-information. We refer to such a scheme as an "*SSR-unaware*" cross-layer scheme. In this section we show that CSI available from SSR indications can provide additional capacity gains.

We employ LDPC code based FEC simulations on our wireless traces to exhibit the improvement in error recovery. Finally, H.264 based video simulations are used to show the benefits for a video application.

A. Cross-layer capacity gain due to side-information:

An *SSR-aware* scheme, utilizes the (Z, SSR) tuple as side-information. From equation (2) of section II we can express the capacity of an *SSR-aware* protocol as:

$$C_{SSR-Aware} = \sum_{(Z,SSR)} f(Z,SSR)\left(1-h_b\left(\varepsilon_{(Z,SSR)}\right)\right) = \sum_{SSR} f(Z=1,SSR) + \sum_{SSR} f(Z=0,SSR)\left(1-h_b\left(\varepsilon_{(SSR)}\right)\right) \quad (9)$$

Meanwhile, an SSR-unaware scheme only utilizes the indication Z as side-information. This is equivalent to utilizing the (Z, SSR) information in a lumped form, where the lumped states are represented by $Z_0 = \{(Z,SSR)|Z=0\}$ and $Z_1 = \{(Z,SSR)|Z=1\}$. Thus with the help of equation (3) from section II we can express the capacity of an SSR-unaware protocol as:

$$C_{SSR-unware} = \sum_{i=0}^{1} f(Z_i)\left(1-h_b\left(\sum_{(Z,SSR)\in Z_i} f(Z,SSR)\varepsilon_{(SSR,Z)}\right)\right) = f(Z_1) + \left(1-h_b\left(\sum_{SSR} f(Z=0,SSR)\varepsilon_{(SSR)}\right)\right) \quad (10)$$

Inequality (4) in section II directly implies that $C_{SSR-aware} \geq C_{SSR-unaware}$. Additionally, comparing (9) and (10) it can be deduced that the difference in *capacity gain* due to SSR will increase when (a) $f(SSR, Z=0)$ are not negligible (i.e. a significant number of the packets are corrupted) and (b) when there is a substantial variation in SSR indications. To clearly illustrate this fact we consider an example:

Example 1: Let us assume that a link has an average SSR value of 10dB and the variations in link-quality can be expressed by the following truncated Gaussian distribution on the SSR indications:

$$if \quad 4 \leq SSR \leq 16 \quad f(SSR) = \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{(SSR-10)}{2\sigma^2}\right)}\right) \Bigg/ \sum_{SSR=4}^{16}\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{(SSR-10)}{2\sigma^2}\right)}\right), \; else \; f(SSR) = 0$$

Now note that $f(Z=0,SSR) = \delta_{(SSR)} f(SSR)$ and $f(Z=1,SSR) = \left(1-\delta_{(SSR)}\right) f(SSR)$. Thus we can evaluate the capacity of the SSR-aware and SSR-unaware scheme with the help of (9), (10) and the em-

pirical relationships established in the previous section. Figure 6 plots the capacities for the considered example. It can be clearly seen that the capacity gain increases as the standard deviation σ increases. This conclusion can be further formalized with the help of the *Holder's Defect Formula.*

Holder's Defect Formula ([36]):

If $g:[a,b]\longrightarrow \mathbf{R}$ is twice differentiable and if we have the bounds

$$0 \le m \le g^{"}(x) \le M \quad for\ all\ x \in [a,b]$$

then for any real values $a \le x_1 \le x_2 \le \cdots \le x_n \le b$ and any non-negative reals p_k, $k = 1,2,\ldots,n$ with $\sum_{i=1}^{n} p_i = 1$, there exists a real value $\mu \in [m,M]$ for which one has the formula

$$\sum_{k=1}^{n} p_k g(x_k) - g\left(\sum_{k=1}^{n} p_k x_k\right) = \frac{1}{4}\mu \sum_{j=1}^{n}\sum_{k=1}^{n} p_j p_k (x_j - x_k)^2$$

Now let's choose $p_k = f(SSR = k | Z = 0) = \dfrac{f(SSR = k)\delta_{(SSR=k)}}{f(Z_0)}$ where $f(Z_0) = \sum_{k=1}^{14} f(Z = 0, SSR = k)$, $x_k = \varepsilon_{(Z=0,SSR=k)} = \varepsilon_{(SSR=k)}$, $g(x_k) = 1 - h_b(x_k)$ and $[a,b] \equiv [0,0.5]$. For such a choice we have $m = 4$ and hence from the above formula we can conclude that

$$C_{SSR-aware} - C_{SSR-unaware} \ge f(Z_0)\sum_{j=1}^{14}\sum_{k=1}^{14}\left(f(SSR_j | Z = 0) f(SSR_k | Z = 0)\left(\varepsilon_{(SSR=j)} - \varepsilon_{(SSR=k)}\right)^2\right)$$

$$\Rightarrow C_{SSR-aware} - C_{SSR-unaware} \ge f(Z_0) \cdot \mathrm{var}\left(\varepsilon_{(SSR)} | Z = 0\right)$$

where, $f(Z_0)$ is the probability of a packet being corrupted and $\mathrm{var}\left(\varepsilon_{(SSR)} | Z = 0\right)$ is the conditional variance of the cross-over probability $\varepsilon_{(SSR)}$. At this point, we assume that the relationship between $\varepsilon_{(SSR)}$ and SSR is convex, then by re-applying the Holder's formula we get

$$\Rightarrow C_{SSR-aware} - C_{SSR-unaware} \ge f(Z_0) \cdot \mu^{"} \cdot \mathrm{var}\left(SSR | Z = 0\right)$$

where $\mu^{"} \approx \min_k\left(0.25 \cdot \left(\varepsilon_{(k+2)} + \varepsilon_{(k-2)} - 2\varepsilon_{(k)}\right)\right)$ (numerical approximation of the second derivative).

From the above discussion we should expect the SSR aware scheme to provide significant performance gains when the SSR values lie primarily in the transition range and exhibit substantial variations. Our simulations in the subsequent subsections further validate this claim.

B. Trace-based simulations

B.1. *High-level System Description and Experimental Methodology:*

To illustrate the practical utility of an *SSR_aware* scheme we consider a *Video Communication System* described by Figure 8. In the system described in Figure 8 a remote-server multicasts H.264 compressed video using an LDPC code based FEC scheme. A receiver can employ either an *SSR_aware* or an *SSR_unaware* cross-layer scheme. We compare the performance of these schemes by employing trace based simulations. In accordance to the discussion in the previous sub-section, we focus our analysis on traces that contain at least 20% packets in the transition range. We chose 6 traces (3 from each environment) of 20,000 packets each to emulate the wireless channel. In addition, we use the following FEC and video specifications:

FEC: The Progressive-Edge-Growth (PEG) algorithm based on large-girth tanner graphs was used to determine check matrices for all the LDPC codes [28] used in our FEC scheme. The channel codes we employed consisted of 8192 message bits and the redundancy was increased in steps of 512 bits. Thus we studied the performance of codes from rate R= 0.94 to 0.5, packet block length N = 136 to 256, code length n = 8704 to 16384. We consider only binary codes, thus b = 2. As our trace consists of a packet payload of 1024 bytes, s = 8192 bits. Thus the total LDPC codewords in a single FEC block were 128.

Video Encoding: For the purpose of our video simulations we used streams compressed using H.264 (version 9.1 of the reference software) [33]. To emulate conditions where the wireless traces were collected we use video streams with a bit-rate of 2.1 Mbps for the simulations on Home Data, and bit-rate of 5.4 Mbps[2] for simulations on Office Data. We report results for test sequences "Mobile" and "Stefan". The resolution of these sequences was "CIF". We use a GOP (IBPBP) size of 30 frames, frame frequency of 30 fps. Test se-

[2] The source bitrates we have chosen do not necessarily represent practical choices. Our choice of bitrates was purely based on a motive to provide a single contiguous source ("compressed video + FEC" source) that could occupy almost all the available bandwidth.

quences are repeated thrice to provide a long enough playout sequences. The compressed stream was mapped to the FEC scheme by embedding a Network Adaptation layer Unit (NALU) in each interleaved codeword as shown in Figure 8. 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.

B.2. *Utilizing CSI for LDPC decoding*

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. As described in section II, we can utilize the CSI to improve the accuracy of the LLR initializations. In this work, we assume that state of the channel remains identical throughout the packet. Thus (Z, SSR) tuple indication can be used to associate a channel state with each bit in the packet. Depending upon the cross-layer methodology employed at the receiver these channel states can be used for LLR initialization in the following manner:

- For the SSR_aware scheme if a bit is received in state $(Z=0, SSR)$ set $L(c) := (-1)^y \log\left(\frac{1-\varepsilon_{SSR}}{\varepsilon_{SSR}}\right)$
- For the SSR_unaware scheme if a bit is received in state $(Z=0, SSR)$ set $L(c) := (-1)^y \log\left(\frac{1-\varepsilon_{Z_0}}{\varepsilon_{Z_0}}\right)$

  where $\varepsilon_{Z_0} = \sum_{SSR} f(SSR, Z=0)\varepsilon_{(SSR)}$. Note that, the value of ε_{Z_0} depends upon the frequencies $f(SSR, Z=0)$. These frequencies vary with each link and trace. Hence for the purpose of our simulations we empirically deduce these frequencies for each trace.
- For both the schemes, if the bit is received in state $(Z=1, SSR)$ set $L(c) := (-1)^y \infty$. In practice we replace the infinity with a large finite value.
- Similarly, for both the schemes, if the bit belongs to an erased packet, set $L(c) := 0$

B.3. *Results and Analysis*:

The results of our experiments are tabulated in Table III. The observations in Table III are similar to the capacity predictions made in the sub-section A. It can be observed that, the performance improvements provided by the *SSR_aware* scheme are negligible in the good and bad SSR ranges. Never the less, in the transition range, 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 the performance improvement provided by the *SSR_aware* scheme increases with variations in the SSR values. The standard deviation σ of the SSR values was found to be at least 3dB in all the traces presented here[3]. The significance of σ can be especially appreciated by observing the error recovery performance for trace 6. In trace 6, even though the average channel conditions are "good", the variation in SSR value is significant enough for an *SSR_aware* scheme to exhibit improved error recovery performance.

Table III also shows the improvement of video quality in terms of PSNR. It was observed 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 9 (a), Figure 10 (a), Figure 11 (a) and Figure 12 (a**)** provide temporal snapshots for 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* scheme 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 9 (b), Figure 10 (b), Figure 11 (b) and Figure 12 (b). The *SSR_aware* scheme provides an improvement of 23dB in Figure 9 (b), 14dB in Figure 10 (b), 18dB in Figure 11 (b) and 6dB in Figure 12 (b). In addition to the block distortion, the motion discontinuity in an *SSR_unaware* scheme was also significantly higher. Thus it can be clearly observed that under dynamic channel

[3] 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 (even limited) mobility.

conditions, SSR can be used as a realtime predictive tool to significantly improve the performance of cross-layer multimedia protocols.

VI. CONCLUSIONS

In this paper, with the help of analysis and experimentation with actual 802.11b traces, we have exhibited the predictive utility of SSR values in the design of cross-layer protocols. SSR values can be used on the basis of site-survey techniques to facilitate a *Cross-layer Network Plan*. Moreover, it has also been shown that SSR values can be used on realtime basis, to provide vital CSI for individual packets. Such realtime CSI can be used to improve the error recovery under dynamic channel conditions. Under certain realistic wireless channel conditions where the link-quality has significant variations, use of SSR can lead to an improvement in video quality in excess of 5-10 dB. Having established the predictive utility of SSRs, in future, we intend to use this tool to help design cross-layer equivalents of a variety of standard multimedia communication protocols. For example in [34] we present preliminary results of utilizing SSR information for planning Network Coding in presence of errors, while in [35] we utilize SSR information in realtime for an Adaptive Network Coding scheme.

REFERENCES

[1] A. Majumdar, D. G. Sachs, I. V. Kozinetsev, K. Ramchandran, “Multicast and unicast realtime video streaming over wireless LANs”, IEEE Transaction on Circuits Systems and Video Technology. Vol. 6, 2002.

[2] S. Shakkottai, T. S. Rappaport and P. C. Karlsson ``Cross-layer Design for Wireless Networks,", IEEE Communications magazine, October, 2003.

[3] M. van der Schaar, S. Krishnamachari S. Choi, X. Xu, “Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 WLANs”, IEEE Journal on Selected Areas of Communication, vol. 21, Dec. 2003.

[4] I. Kozinetsev and J. McVeigh, “Improving last-hop multicast streaming video over 802.11”, BroadWiM 2004.

[5] Y. Wu, P. A. Chou, Q. Zhang, K. Jain, W. Zhu, and S.-Y. Kung, “Network planning in wireless ad hoc networks: a cross-layer approach,” IEEE Journal on Selected Areas in Communications, special issue on wireless ad hoc networks, vol. 23, no. 1, pp. 136-150, Jan. 2005.

[6] M. van der Schaar and S. Shankar, "Cross-Layer Wireless Multimedia Transmission: Challenges, principles, and new paradigms," IEEE Wireless Communications Magazine, vol. 12, no. 4, pp. 50-58, August 2005.

[7] I. Haratcherev, J. Taal, K. Langendoen, R. Lagendijk and H. Sips, “Optimized Video Streaming over 802.11 by Cross-layer Signaling,” IEEE Communication Magazine, Special Issue on Cross-Layer Design, Jan. 2006.

[8] D. B. Faria, D. R. Cheriton, “No Long-term Secrets: Location-based Security in Overprovisioned Wireless LAN, ACM Workshop HoT-NeTs, Nov. 2004.

[9] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan, “Measurement-based Characterization of 802.11 in a Hotspot Setting,” SIGCOMM 2005 Workshop, E-WIND, Aug. 2005.

[10] L. Larzon, M. Degermark, and S. Pink, “UDP Lite for Real Time Multimedia Applications,” IEEE International Conference of Communications (ICC), Vancouver, June 1999.

[11] A. Singh, A. Konrad, A. D. Joseph, “Performance Evaluation of UDP Lite for Cellular Video,” NOSSDAV, 2001.

[12] H. Zheng and J. Boyce, “An Improved UDP Protocol for Video Transmission Over Internet-to-Wireless Networks,” IEEE Trans. on Multimedia, vol. 3, no. 3, pp. 356−365, September 2001.

[13] G. Ding, H. Ghafoor and B. Bhargava, “Resilient Video Transmission over Wireless Networks”, 6th IEEE International Conf. on Object-oriented Realtime Distributed Computing, Japan, May 2003.

[14] 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 Realtime Multimedia,” Signal Processing: Image Communication, vol. 18, Aug 2003.

[15] 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.

[16] 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.

[17] A. Servetti, J.C. De Martin, "Link-Level Unequal Error Detection for Speech Transmission over 802.11b Networks," Proc. Special Workshop in Maui (SWIM), Maui, Hawaii, January 2004.

[18] E. Masala, M. Bottero, J.C. De Martin, “MAC-Level Partial Checksum for H.264 Video Transmission over 802.11 Ad Hoc Wireless Networks” Proceedings of IEEE 61st Semiannual Vehicular Technology Conference (VTC), May 2005.

[19] E. Masala, A. Servetti, J.C. De Martin, “Standard Compatible Error Correction for Multimedia Transmissions over 802.11 WLAN”, Proc. of IEEE International Conference on Multimedia & Expo (ICME), July 2005.

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

[21] S. Karande, U. Parrikar, K. Misra and H. Radha, “On Modeling of 802.11b Residue Errors,” Conference on Information Sciences & Systems (CISS), March 2006.

[22] S. Karande, U. Parrikar, K. Misra and H. Radha, "Utilizing Signal to Silence Ratio indications for improved Video Communication in presence of 802.11b Residue Errors ", IEEE International Conference on Multimedia & Expo (ICME), July 2006.

[23] 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.

[24] A. Lapidoth and P. Narayan, “Reliable communication under channel uncertainty”, IEEE Trans. Inform. Theory, 44:2148–2177, October 1998.

[25] S. Z. Stambler, “Shannon’s theorems for a complete class of discrete channels whose state is known at the output”, Prob. Peredachi Informatsii, 11(4):3–12, 1975.

[26] S. C. Draper, B. Frey, and F. R. Kschischang, "Rateless Coding for Non-Ergodic Channels with Decoder Channel State Information", www.eecs.berkeley.edu/~sdraper/researchDir/pdf/ratelessTimeVary.pdf

[27] W.E. Ryan, “An introduction to LDPC codes,” in CRC Handbook for Coding and Signal Processing for Recoding Systems (B. Vasic, ed.), CRC Press, 2004.

[28] X. 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.

[29] ISL3873: Wireless LAN Integrated Medium Access Controller with Baseband Processor Intersil Corporation, 2000.Application Note FN4868.

[30] http://www.linux-wlan.org

[31] http://www.wi-fiplanet.com/tutorials/article.php/1116311

[32] http://www.awe-communications.com/Network/WLAN/index.html

[33] 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.

[34] S. Karande, K. Misra and H. Radha, "CLIX: Network Coding and Cross-layer Information Exchange of Wireless Video", IEEE ICIP 2006.

[35] S. Karande, K. Misra, U. Ilyas, H. Radha, "Adaptive Network Coding With Noisy Packets", Conference on Information Sciences & Systems (CISS), March 2006.

[36] J. Michael Steele, "An Introduction To The Art Of Mathematical Inequalities: The Cauchy-Schwarz Master Class", Cambridge University Press, 2004.

Table I Protocol dependent goodput of Reed-Solomon based FEC for various 802.11b traces

	H1	H2	H3	H4	O1	O2	O3	O4
Traditional	52%	99%	11%	74%	74%	40%	92%	34%
Cross-layer	75%	99%	24%	81%	83%	62%	94%	47%
Improvement	28%	0%	13%	7%	9%	22%	2%	13%

Table II Distribution of packets in different SSR ranges for various 802.11b traces

	H1	H2	H3	H4	O1	O2	O3	O4
bad	1.8%	0.2%	32.6%	6.8%	2.8%	12.2%	1.4%	13%
good	37.4%	97.5%	28%	66.9%	66.7%	16.5%	91.3%	43%
transition	60.8%	2.3%	39.4%	26.3%	30.5%	71.3%	7.3%	44%

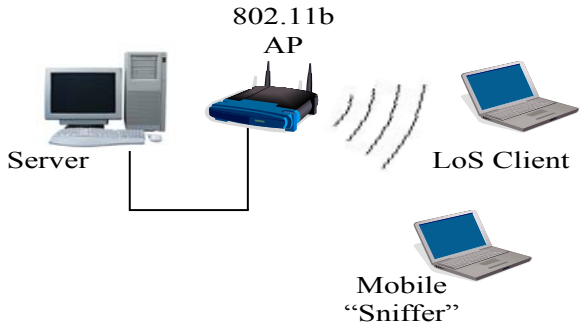
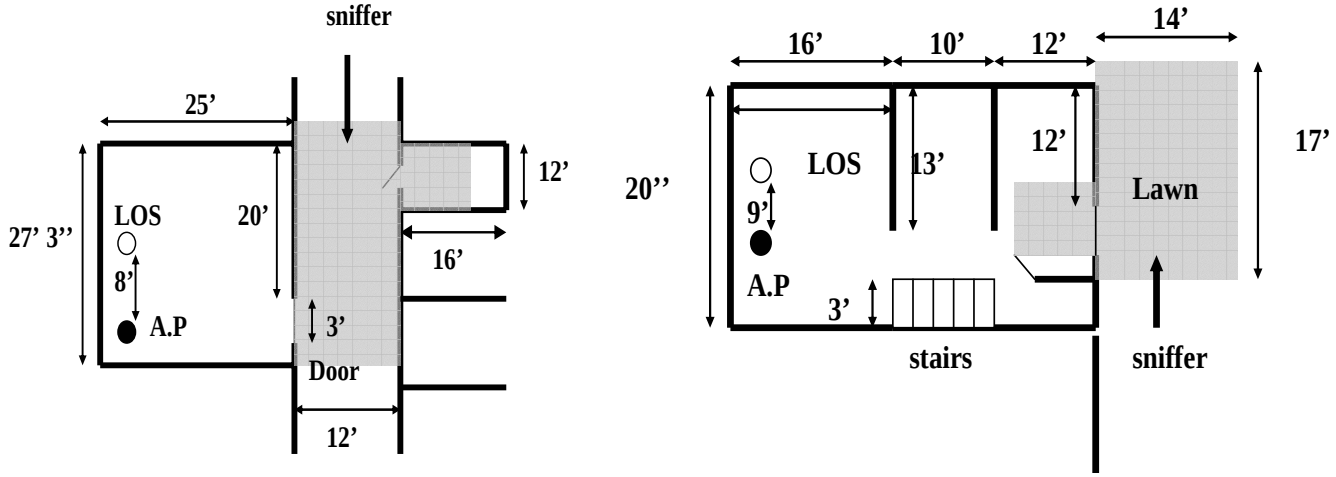


Figure 1 Trace collection setup



(a) Office Setup (b) Home Setup

Figure 2 Trace collection environments

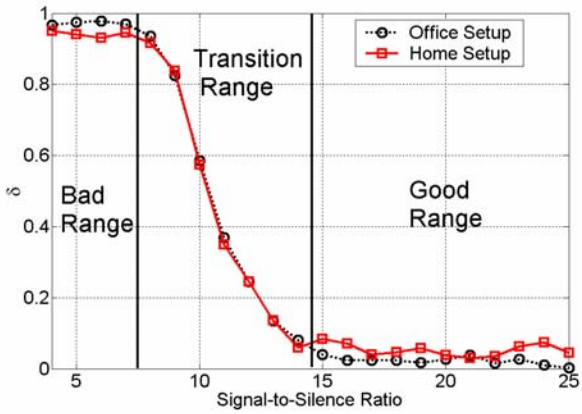


Figure 3 Empirically determined relationship $\delta_{(SSR)}$

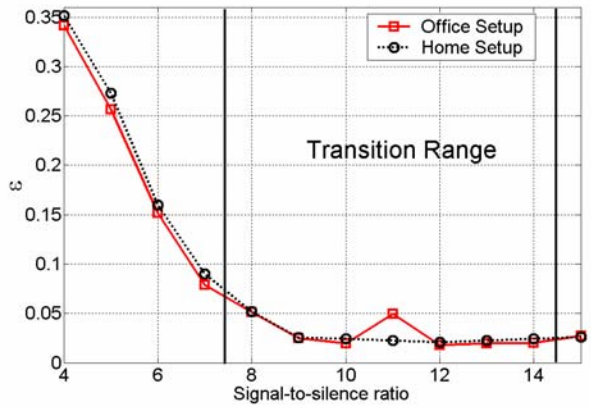


Figure 4 Empirically determined relationship $\varepsilon_{(SSR)}$

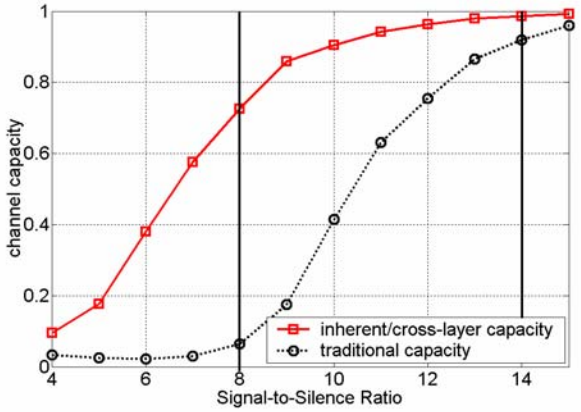


Figure 5 Capacity gain due to cross-layer protocols

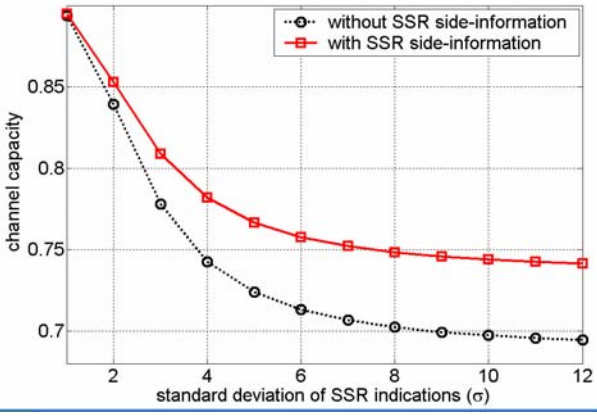


Figure 6 Capacity gain due to side-information

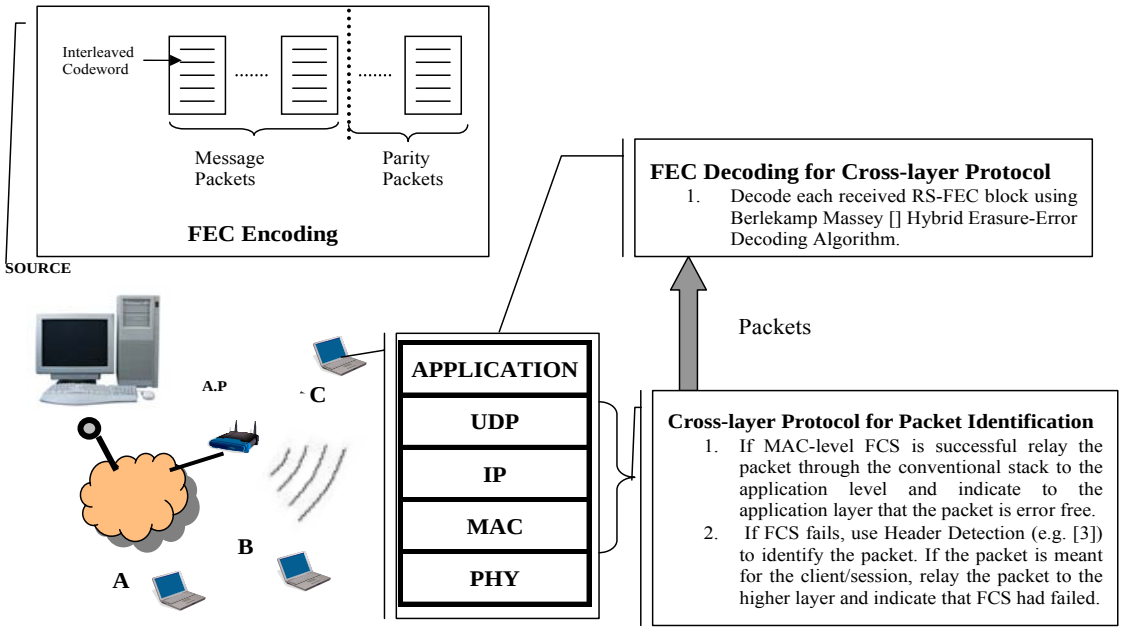


Figure 7 System Model for Cross-layer Packet recovery

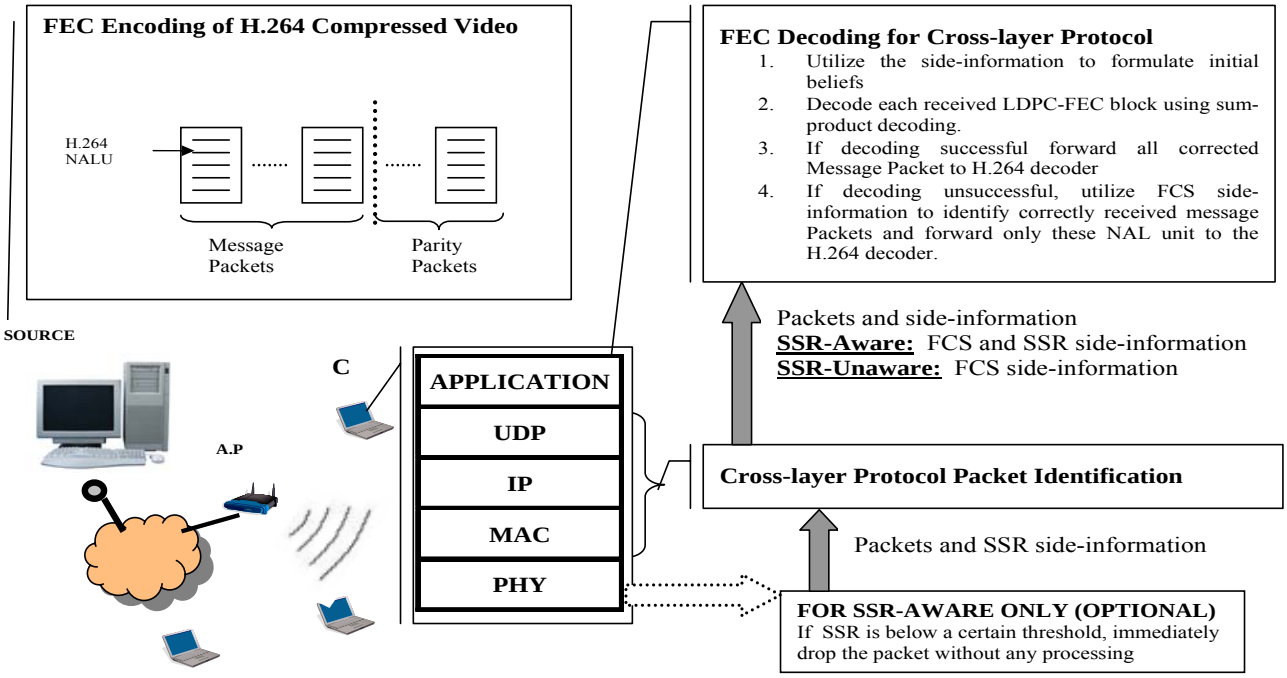
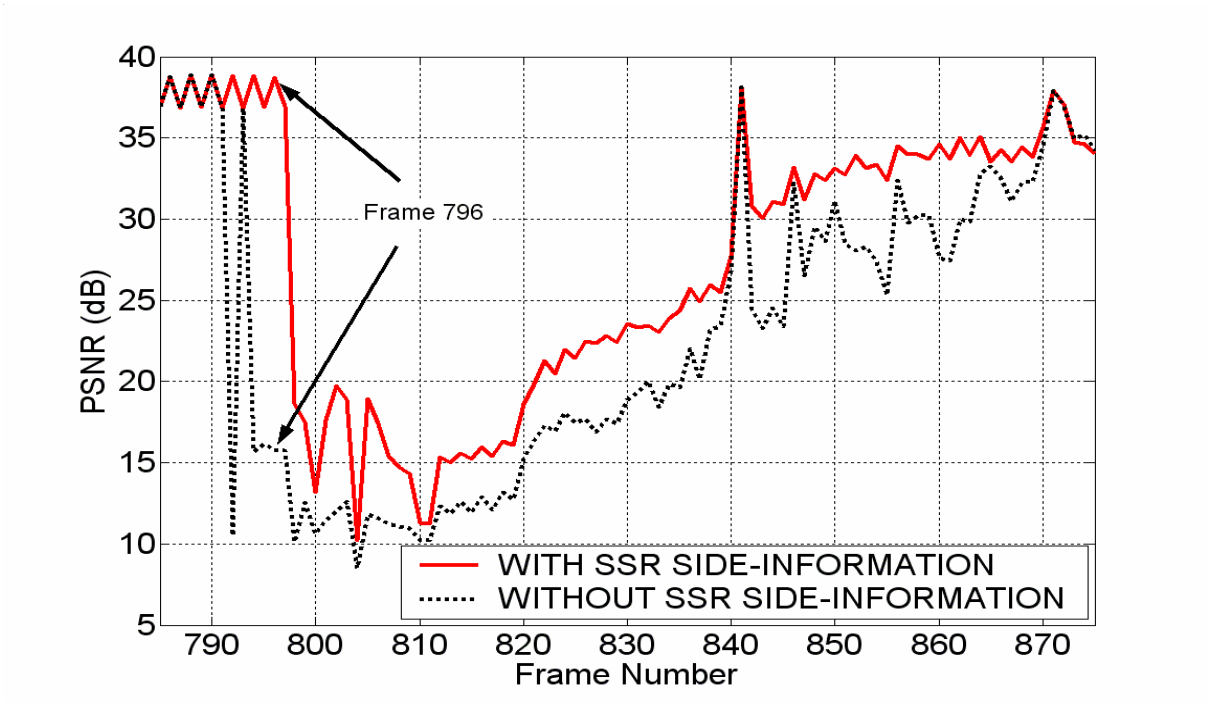
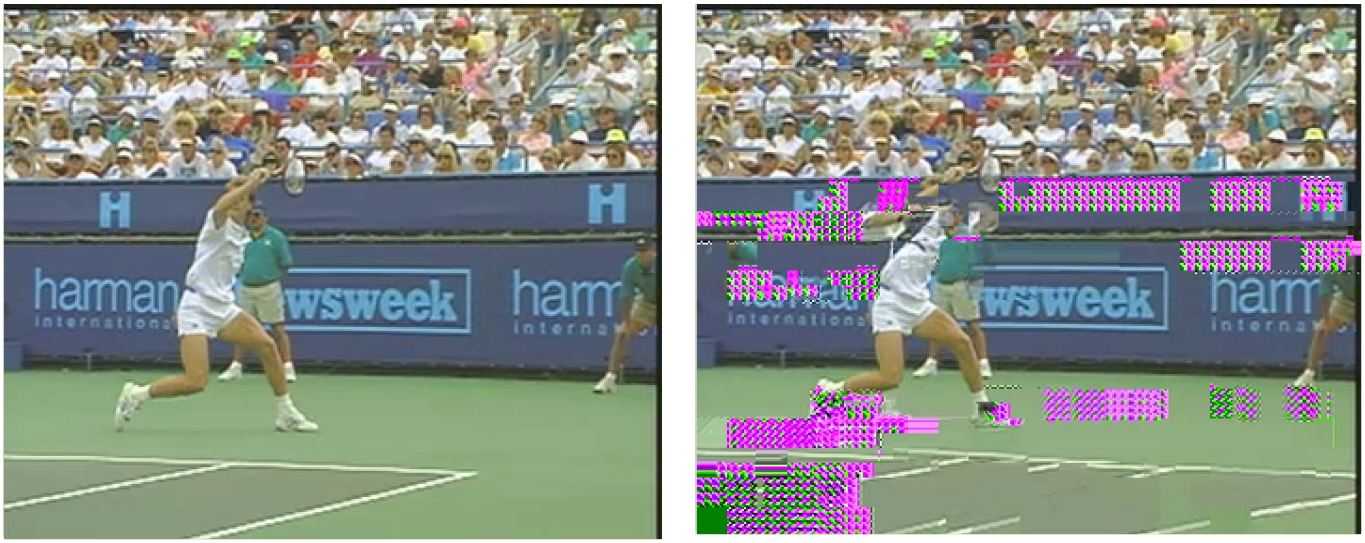


Figure 8 System Model for Cross-layer Packet Recovery with Side-Information



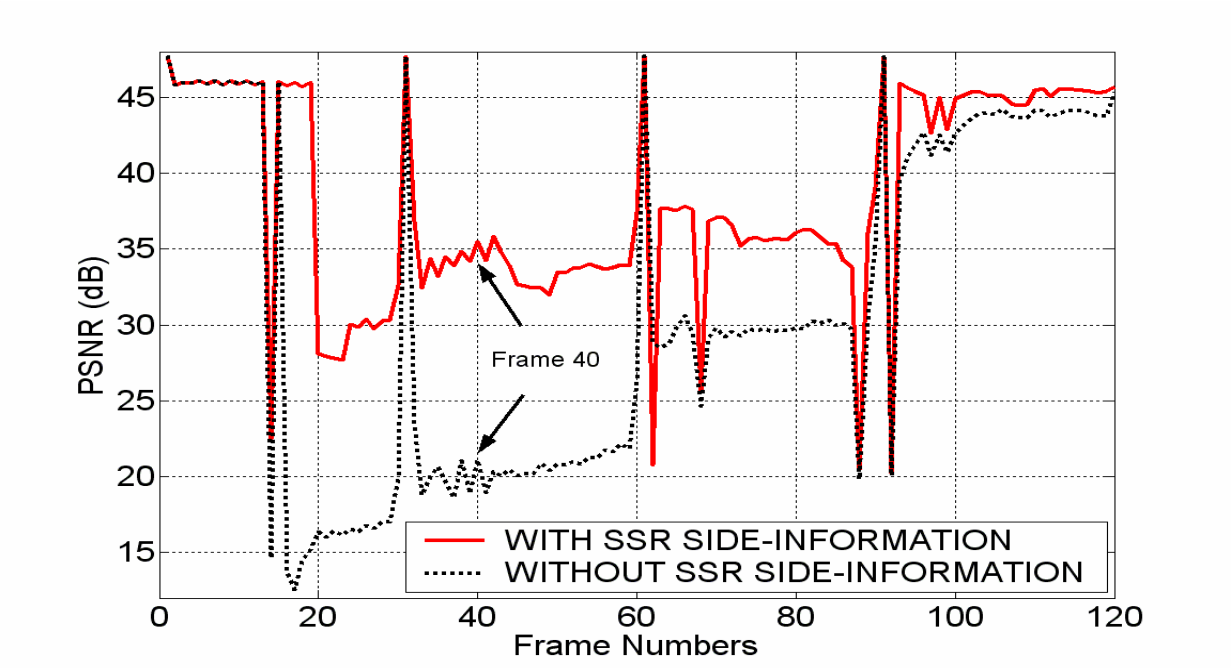
(a)



38.72 dB **15.78 dB**

(b)

Figure 9 Results of transmitting Stefan 2.1Mbps video over trace 4. (a) temporal snapshot (b) frame 796 corresponding to the temporal snapshot. Left: Current Scheme with SSR based CSI, Right: Previous Scheme.



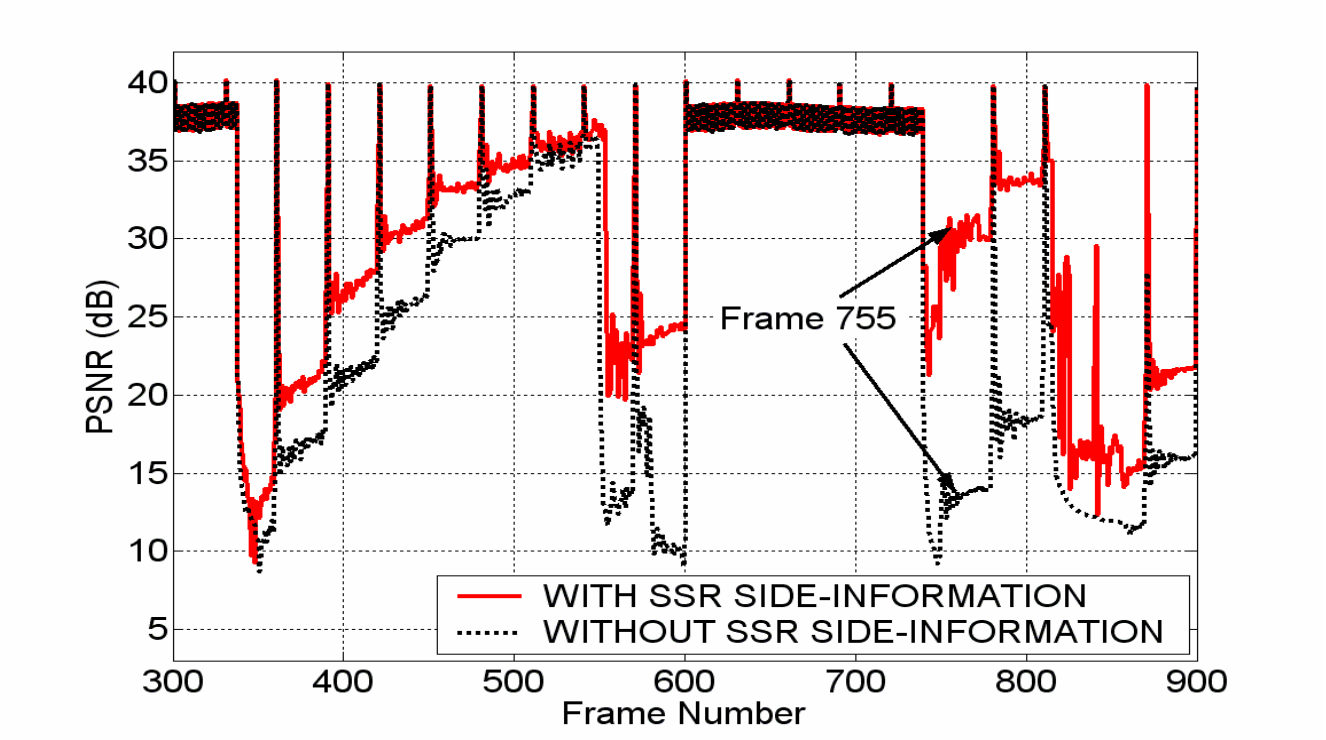
(a)



35.47 dB **21.09 dB**

(b)

Figure 10 Results of transmitting Stefan 5.4Mbps video over trace 4. (a) temporal snapshot (b) frame 40 corresponding to the temporal snapshot. Left: Current Scheme with SSR based CSI, Right: Previous Scheme.



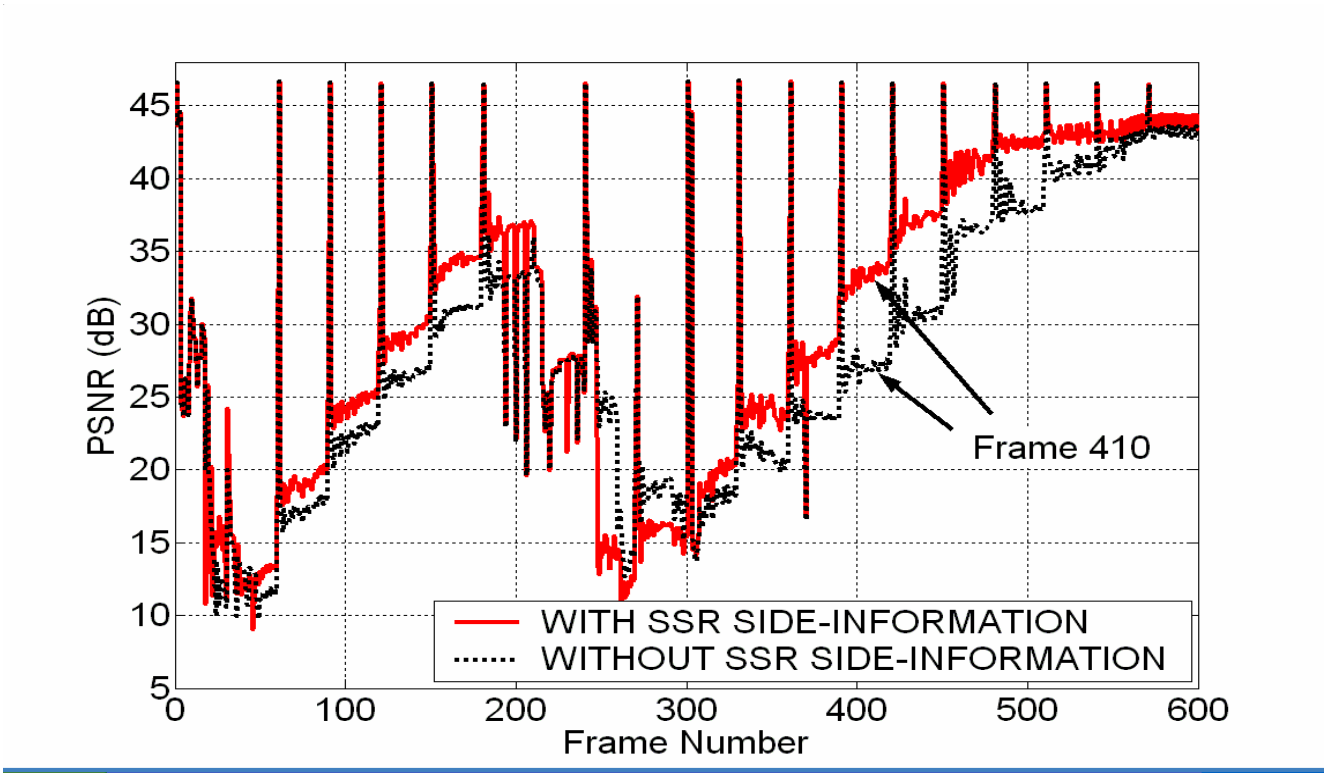
(a)



31.38 dB **13.30 dB**

(b)

Figure 11 Results of transmitting Mobile 2.1Mbps video over trace 1. (a) temporal snapshot (b) frame 800 corresponding to the temporal snapshot. Left: Current Scheme with SSR based CSI, Right: Previous Scheme.



(a)



33.93 dB **27.57 dB**

(b)

Figure 12 Results of transmitting Mobile 5.4Mbps video over trace 1. (a) temporal snapshot (b) frame 410 corresponding to the temporal snapshot. Left: Current Scheme with SSR based CSI, Right: Previous Scheme.

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

Trace		1	2	3	4	5	6
WLAN Infrastructure		Home Setup			Office Setup		
Standard Deviation in the observed SSR values (dB)		3.2	3.2	7.7	3.8	4.1	7
Channel Coding Rate		0.8	0.66	0.94	0.8	0.66	0.95
Source Bit-rate		2.1 Mbps			5.4 Mbps		
Percentage Code Failures[⊥][ſ]	**good**	-	3.6	5.4	0.0	0.0	3.8
	SSR Range	-	5	7.0	0.0	0.0	6.0
	transition	0.5	4.0	14	7	1.9	-
	SSR Range	2.5	7.8	16	18	3.1	-
	bad	-	-	-	100	100	-
	SSR Range	-	-	-	100	100	-
	Total	0.5	3.9	11	6	1.7	3.8
	SSR Range	2.5	7.5	13	16	2.8	6.0
Ave. PSNR GAIN(dB)[ϒ]	**Stefan**	5.5	0.93	1.19	1.6	0.78	0.82
	Mobile	6.78	0.75	0	2.3	2.07	1.37

[⊥]In the above table the column entry of *percentage code failures* has been divided into sub-columns good, transition and bad. The results in these sub-columns are obtained by classifying an FEC packet block based on the SSR averaged over the block.

[ſ] The highlighted sub-columns present the results corresponding to the "*SSR_Unaware*" scheme.

[ϒ] The PSNR gain describe the average improvement in video quality provided by the "*SSR_aware*" scheme. The PSNR has been averaged over 900 frames. Thus in certain cases, despite the existence of specific GOPs where the gain was in excess of 10dB, the average gain is negligible.