Available online at www.sciencedirect.com
SCIENCE DIRECT®
Signal Processing: *Image Communication* 18 (2003) 575–595
SIGNAL PROCESSING: IMAGE COMMUNICATION
www.elsevier.com/locate/image

Performance analysis and modeling of errors and losses over 802.11b LANs for high-bit-rate real-time multimedia

Syed A. Khayam[a,*], Shirish Karande[a], Hayder Radha[a], Dmitri Loguinov[b]

[a] *Department of Electrical & Computer Engineering, EB 2120, Michigan State University, East Lansing, MI 48824, USA*
[b] *Computer Science Department, HRBB 515C, Texas A&M University, College Station, TX 77843, USA*

Received 29 November 2002; accepted 2 April 2003

Abstract

Inherent error-resilient nature of multimedia content renders two high-level options for wireless multimedia application design. One option is to employ (semi-) reliable wireless Medium Access Control (MAC) functions in conjunction with the traditional User Datagram Protocol (UDP). The other option is to employ a less-reliable MAC and transport layer protocol stack that passes corrupted packets to the application layer, which consequently achieves a "higher throughput". This "higher throughput" traffic, however, could contain many "useless" corrupted packets. In this paper, we address key questions regarding the viability of the above two options for the support of high-bit-rate wireless multimedia applications over 802.11b LANs. First, we study the level of throughput improvements realized by the less-reliable protocol stack at 2, 5.5 and 11 Mbps data rates using actual measurements that mimic realistic home or business settings. Second, we analyze and model the error patterns within the "higher throughput" corrupted packets to evaluate their potential impact on multimedia applications. Third, we compare the amount of overhead that is needed at the application layer to achieve different levels of lost- and corrupted-packet recovery for the two (reliable and less-reliable) protocol stack scenarios. Major conclusions of our study include: (1) Either protocol-stack is viable at 2 Mbps while neither of them is viable at 11 Mbps under realistic settings; and (2) Some benefits of the "higher throughput" corrupted packets can be realized at 5.5 Mbps when combined with a joint erasure-error protection algorithm at the application layer.

Keywords: High-bit-rate multimedia; Wireless channel modeling; 802.11b networks; Protocol design; UDP Lite

1. Introduction

The advent of wireless networks has advanced real-time multimedia communication into a novel realm. The wireless medium, due to its inherent vulnerability and physical characteristics, is more error prone than most of the contemporary (wired) media. Nevertheless, migration of technologies from the wired to wireless domain is currently underway. One of the key challenges in this context is the delivery of real-time content

*Corresponding author. Fax: +1-517-353-1980.
E-mail addresses: khayamsy@egr.msu.edu (S.A. Khayam), karandes@egr.msu.edu (S. Karande), radha@egr.msu.edu (H. Radha), dmitri@loguinov.com (D. Loguinov).

0923-5965/03/$ - see front matter
doi:10.1016/S0923-5965(03)00053-5

over the wireless medium. The promise of real-time multimedia over the wired Ethernet is often attributed to the simplicity and pragmatism of User Datagram Protocol (UDP)/IP protocol suite. However, this protocol suite was designed for considerably low error-rate environments as opposed to wireless networks. In this regard, a positive aspect of real-time applications is their inherent tolerance to a certain level of corrupted and/or dropped packets. This characteristic of real-time applications motivated the development of new methods and protocols for wireless networks. For example, a new transport layer protocol which is tailored for real-time applications over error-prone networks has been proposed recently [1–3]. This protocol, known as UDP Lite, relies on the premise that multimedia applications can tolerate, and therefore should, receive corrupted packets. Naturally, this UDP Lite-based framework makes it necessary for the Medium Access Control (MAC) layer to forward corrupted packets to the higher layers. Consequently, developers of wireless multimedia applications are faced with two (high-level) options: (1) Employing current wireless MAC functions, which include dropping of corrupted packets and attempting to recover these packets through retransmissions, in conjunction with the traditional UDP protocol; or (2) Allowing the MAC layer to pass corrupted packets to a UDP Lite-type transport layer which in turn relies on the application layer to handle the corrupted data. A key advantage of the second option is the increase in the, potentially corrupted, throughput of the end-to-end transport layer packets. However, these "high-throughput" packets contain corrupted data.

Therefore, and based on the above two options, three key questions can be raised: (1) How much improvement in the "throughput" is actually being gained by using the new UDP Lite-based framework? (2) How badly corrupted these "high-throughput" packets are? In other words, how much and what type of error patterns are observed in these packets? (3) As a consequence of the first two questions, what level of application layer loss-protection, error-concealment and/or error-correction would be required to achieve acceptable quality under the two protocol-stack variants? In this paper, we address the above questions in the context of high-bit-rate[1] multimedia applications[2] over 802.11b wireless LANs.

At this point, it is important to outline how we intend to answer these three questions. The first question, which has been addressed partially by previous studies (see, for example, [3,4]), can be answered through a set of experiments and related analysis that is presented in this paper. Our analysis of packet throughput is conducted at the 2, 5.5 and 11 Mbps data rates using actual measurements that mimic realistic home or business settings.[3] Answering the second question requires a thorough analysis and some modeling of the error patterns at the MAC layer (i.e., errors that are not corrected by the physical layer). In addition to presenting our experimental results and analysis for byte-level[4] error patterns at the MAC layer, we propose a new hierarchical Markov-based model for representing these error patterns. As shown later in the paper, the proposed model captures the observed errors more accurately than the traditional two-state Markov chain.

Addressing the third question is naturally more challenging than the first two questions, as it depends on a wide spectrum of objective and subjective evaluations of a variety of multimedia applications and corresponding error protection and concealment algorithms. For example, some applications targeted for wireless networks may take advantage of better error resilience features, such as Reversible Variable-Length Coding (RVLC) that is supported in the MPEG-4 video standard [5]. Other examples include a range of (video) error concealment algorithms and/or Forward-Error-Correction (FEC) methods [6,7]. Due to this wide spectrum of methods and algorithms, which can influence the answer to the above third

[1] In this work, we conducted our study and analysis at the full-rates provided by the different modes at the physical layer. Consequently, our definition of "high-bit-rate" follows the 2, 5.5, and 11 Mbps bit-rates supported by 802.11b.

[2] An example of real-time multimedia is MPEG-4 video [5], which we use occasionally in this paper to illustrate some visual simulation results in support of our findings.

[3] We believe that a minimum criterion for a "realistic" setting is a wireless transmission between two rooms. This is explained further later in the paper.

[4] In order to develop a thorough understanding of the channel, we performed analysis of error traces at two levels of granularity, i.e., bytes and packets. Respectively, it is referred to as byte-level and packet-level analysis.

question, we had no option but to focus on one important (sub-) question of this spectrum: What is the amount of overhead that is needed at the application layer to achieve different levels of lost- and corrupted-packet recovery for the two protocol-stack scenarios? The answer to this question provides a worst-case measure of the level of overhead needed to correct/recover a desired level of corrupted/dropped packets for both UDP and UDP Lite. As explained later, this leads to an important conclusion about the viability of multimedia applications at rates higher than 2 Mbps under realistic settings.

Our decision to focus on high-bit-rate multimedia has been mainly influenced by the fact that emerging wireless LAN standards are designed to support unprecedented high-bit-rates. In particular, 802.11 LANs are finding their way into homes and businesses. Consequently, it is quite feasible that in the near future these LANs could utilize unicast and/or multicast frameworks to provide real-time television-like services to large numbers of users. As explained above, evaluating this feasibility requires a methodical understanding of the error and loss performance at different layers of the protocol stack. We perform all analysis and modeling above the physical layer. Here, we rely on the assumption that, in addition to its adherence to the constraints of the underlying standard, the physical layer is providing the best possible performance at a given desired bit-rate and under given channel conditions. Hence, our channel definition encompasses the wireless medium and the 802.11b physical layer.

The remainder of this paper is organized as follows. Section 2 provides a brief overview of related work in this area. Section 3 describes the experimental setup to generate error traces at 2, 5.5 and 11 Mbps. Section 4, first, studies the throughput of the MAC layer under different loss conditions and uses this insight to conclude an appropriate link layer strategy for multimedia applications. This section, then, investigates the throughput of transport layer protocols, i.e., UDP and UDP Lite in conjunction with the new link layer strategy. A two-state Markov model for packet drop bursts is also presented in this section. Section 5 outlines a heavy-tail-based byte-level burst behavior and proposes two Markov-based models to approximate the burstiness. Section 6 compares the amount of redundancy required by UDP and UDP Lite under the previously mentioned loss conditions. Section 7 summarizes some key conclusions of this paper.

2. Related work

Many studies have attempted to characterize errors in wireless networks. For example, Ludwig et al. [8] analyze block erasure traces in cellular networks. Improvement of TCP performance over wireless links has also been under study by several researchers (e.g., [9,10]).

Recent years have seen a tremendous increase in the demand for streaming real-time applications. In order to provide television-like services, efficient multimedia coding on reliable multicast networks has been an area of active research [11–13]. With wireless networks fast becoming ubiquitous, an imperative direction is to tailor future applications and protocols to adapt accordingly. Some recent works have explored the area of real-time communication over 802.11 networks [14,15]. However, we believe that the impact of such networks on high-bit-rate multimedia needs further investigation.

Despite the robustness at the 802.11 physical layer, some errors propagate to the link layer. These errors are detected using the frame check sequence (FCS) and the link layer discards such (corrupted) packets with no regard to the number and location of the errors [16,17]. A scheme, suggested in [1,3], tried to address this issue by making adjustments to the protocol stack at the transport and link layers. This variant, which is known as UDP Lite, is a lightweight version of the traditional UDP for real-time applications over wireless networks. The protocol allows for delivery of partially damaged packets to the application layer by ignoring errors in the application layer payload. UDP Lite relies on the premise that real-time applications often prefer partially damaged packets over lost packets. It, therefore, allows for checksum on the transport and application layer headers while ignoring checksum on the actual payload. Each packet is divided into a ''sensitive'' and ''insensitive'' part. The sensitive part, starting from the transport layer header, is covered

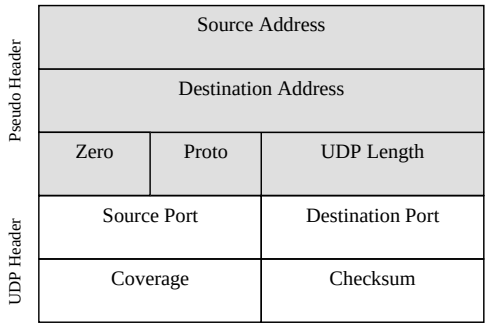

Fig. 1. The UDP Lite headers [2].

by the partial checksum. The "Length Field" in UDP header is utilized to specify the partial checksum length (referred to as "Coverage" in the UDP Lite header).[5] Fig. 1 outlines the UDP Lite headers.

One fundamental problem with this approach is that the link layer drops frames with errors before they reach the transport layer. To cater for this problem, Larzon et al. [1] suggested that the device driver at the receiver be modified to ignore checksum errors for packets carrying UDP Lite (real-time) traffic. This modified wireless MAC layer is referred to as *MAC Lite*.[6] One of the key questions that we address in this study is the efficacy of MAC Lite/UDP Lite protocol stack when compared with a UDP-based stack for high-bit-rate applications.

3. Experimental setup

Our simulation setup employed an 802.11b Access Point (AP) operating in Distributed Coordination Function mode and three wireless stations communicating in the infrastructure network configuration. One of the stations was operating as the server and the remaining two as multicast clients. All wireless stations were Linux boxes using DLink DWL-650 PCMCIA wireless cards with Prism2 chipset device drivers (linux-wlan-ng-0.1.14-pre3) [19]. Prism2 device drivers support a "monitor mode" that allows delivery of MAC frames with failed FCS. Source code of the device driver at the clients was modified to capture screenshots of all (corrupted/uncorrupted) MAC data frames in the kernel buffer space. The clients periodically concatenated the screenshots in user space while operating in monitor mode.

Initially, the server was placed in clear line of sight (LoS) of the AP. The AP was forced to transmit at 2, 5.5 and 11 Mbps for each observation. The server was stationary and transmitted a continuous stream of predetermined patterns to the multicast clients. Traces were generated for each bit-rate at different stationary client positions with and without LoS. It was observed that with clear LoS, the error-rate (at all bit-rates) was extremely low. Such excellent performance deemed further LoS study inconsequential. Hence, both clients were positioned in a separate room across the hallway to simulate a more realistic business/classroom/home-network wireless setup, as shown in Fig. 2. A total of 3 experiments were conducted for each bit-rate. Each experiment involved the transmission of 100,000 packets, and 10 error traces per bit-rate were generated as a result. These experiments were performed at different times of day to nullify effects of the environment and unrelated traffic.

[5] Traditionally, the "Length Field" specifies the length of the UDP packet (i.e., UDP header + payload) [18]. A potential problem arises when UDP Lite uses this field to specify "Coverage" since the receiver uses UDP packet length in transport layer checksum verification. To address this issue, UDP Lite calculates the UDP packet length from the IP header.

[6] In [1] this link layer strategy was referred to as PPP Lite since Point-to-Point Protocol was being employed for their observations.

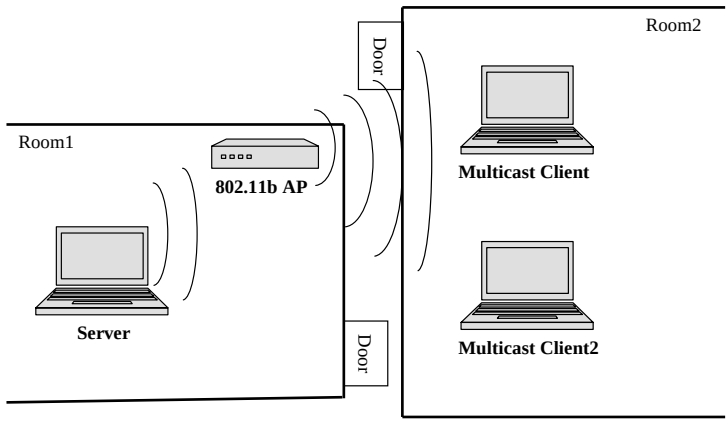

Fig. 2. Simulation setup for error trace collection.

4. Packet-level throughput analysis

The inherent error/loss tolerance of real-time multimedia applications provides a foundation for UDP Lite-like protocols. This section answers the first question raised in Section 1, i.e., how much improvement in the "throughput" is actually being gained by using the new UDP Lite-based framework? Throughout this discussion, the term "throughput" refers to the number of MAC/UDP packets (error-free or corrupted) that are passed by the MAC-/transport layer to the higher layers of the protocol stack. Before addressing the question at hand, we briefly discuss related 802.11b MAC layer functionality and its scope and application in this study.

The 802.11b MAC layer drops corrupted MAC frames regardless of the number and location of the errors. Due to the inherent lossy nature of the wireless medium, real-time applications might prefer corrupted packets as opposed to dropped packets. As mentioned previously, real-time applications can tolerate corruptions to some degree and error resilience features of some modern-day real-time applications further boost performance in such an error-prone scenario. Furthermore, retransmission-based recovery of corrupted packets is not a viable option in some key emerging multimedia applications (e.g., multicast video). Therefore, we performed all analysis while disabling the MAC layer retransmission functionality at the server.

We evaluated different combinations of MAC- and transport layer variants i.e., traditional 802.11b protocol stack (using 802.11b MAC with UDP), MAC Lite with UDP, and MAC Lite with UDP Lite. The fourth combination (i.e., 802.11b MAC with UDP Lite) provides the same results as 802.11b MAC with UDP, since all frames with errors are dropped at the MAC layer, and therefore, corrupted packets never reach the UDP Lite-based transport layer. For all UDP Lite examples, we performed checksum only on the UDP headers without any assumption about the real-time application headers. Thus, the application is solely responsible for handling (i.e., decoding or discarding) the corrupted data.

4.1. MAC layer throughput analysis

As mentioned before, MAC layer "throughput" corresponds to the number of packets that are getting relayed to the network layer i.e., ratio of the number of packets received by the MAC layer to the number of packets received by the network layer. Since the 802.11b MAC drops all corrupted packets, each such packet adversely impacts the throughput. This reduction in throughput propagates to the application layer and, in a retransmission-less scenario, could drastically degrade the quality of perceived media. In this

section we evaluate the 802.11b MAC layer throughput at 2, 5.5 and 11 Mbps, and, suggest remedies to mitigate the throughput reduction experienced due to channel-induced errors.

The maximum packet loss burst at 2 Mbps is two packets long and the throughput is approximately 100%. Such high-quality performance construed further investigation at this bit-rate insignificant. Nevertheless, analysis at higher bit-rates is essential to evaluate the feasibility of high-bit-rate applications. Initial analysis of the MAC layer traces revealed that mostly the errors occurred as bursts of corrupted bits. Often, these bit-error bursts extended across byte (and sometimes packet) boundaries thereby resulting in bursts of corrupted bytes and packets. The cumulative and probability density functions (CDF and PDF) of packet loss bursts for 5.5 and 11 Mbps are outlined in Fig. 3. At 5.5 Mbps (see Fig. 3(a)), more than 60% of packet bursts had a length of one dropped packet. Also, more than 90% of the times, the length of the packet drop burst is smaller than 6 packets. This depicts that at 5.5 Mbps the error-rate of the 802.11b LAN is relatively low when compared with 11 Mbps. In other words, at 5.5 Mbps the byte-level bursts are small and, therefore, they are not frequently spread across packet boundaries. The longest burst in this case is of 17 consecutive packets. It is clear from Fig. 3(b) that at 11 Mbps more than 90% of the times the packet drop bursts are 31 packets long, incurring bursts as long as 45 packets. In addition, the probability distribution has a very high standard deviation, which complicates the design of an application layer error resilience scheme. This bursty behavior at the 11 Mbps rate is further substantiated by our byte-level analysis that is presented in the next section.

Table 1 outlines the packet drop mean, standard deviation and throughput at 2, 5.5 and 11 Mbps, respectively. The average length of packet bursts is 0.0003, 1.68 and 16.35 for 2, 5.5 and 11 Mbps respectively. Clearly, at 2 and 5.5 Mbps the channel bursts are relatively small, whereas at 11 Mbps bursts are on average 16 packets long. Furthermore, at 11 Mbps the standard deviation of packet burst length is very high, i.e., approximately 12 consecutive packets drops. The resulting throughput drops to 14.23%, which can have a profound adverse affect on the quality of perceived media.

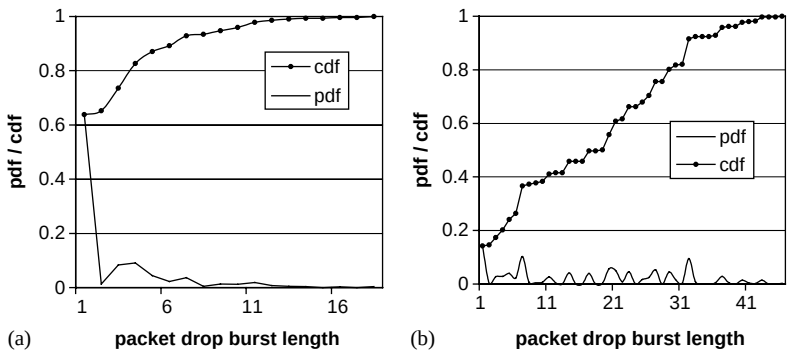

Fig. 3. CDF and PDF of 802.11b MAC packet drop bursts at: (a) 5.5 Mbps and (b) 11 Mbps.

Table 1
Packet drop statistics of traditional MAC layer for an 802.11b network

Bit-rate[a]	Mean	Standard deviation	Throughput (%)
2	0.00029	0.02329	99.9825
5.5	1.67930	2.94070	63.8613
11	16.3493	12.3343	14.2361

[a] Bit-rate is expressed in Mbps throughout this paper.

4.2. Packet-level two-state Markov model

The discussion provided in the previous section implies that mostly the packet drops bursts occur as a consecutive sequence of dropped packets. Therefore, we employed a simple two-state Markov Model to approximate the bursty packet drop behavior. Table 2 summarizes the transition probabilities of the packet-level two-state Markov model for the traditional 802.11b protocol stack (i.e., UDP operating on top of 802.11b MAC) and the overall probability that a packet is dropped (P_{packet}).

These transition probabilities reflect the behavior of the channel at 2, 5.5 and 11 Mbps. Since the robustness of the 802.11 physical layer reduces with the increase in bit-rate [16], it is more susceptible to propagate errors to the MAC layer at higher bit-rates. Consequently, P_{bb} and P_{packet} in Table 2 exhibit directly proportional relationship with the bit-rate. Other transition probabilities are also intuitively convincing.

Packet-drop burst-length substantiates the need for an alternate methodology to improve reliability at the MAC layer in the absence of retransmissions. In [1] it was suggested that a promising short-term approach is to modify the device driver at the receiver in order to turn the link layer FCS off. As mentioned before, we refer to this strategy as MAC Lite.

Evidently, for applications that cannot employ retransmission at the MAC layer, improvement in throughput will only be visible if the strategy of disabling MAC layer FCS is used in conjunction with some appropriate transport layer scheme. Otherwise, all corrupted frames relayed by the MAC layer will be dropped by the checksums operating at network and transport layers.

4.3. Transport layer throughput analysis

In the light of our previous discussion, MAC Lite can be used to improve throughput at the MAC layer. At this point an imperative direction is to study strategies at layers above MAC which can be employed to maximize the throughput of packets with zero-errors or with errors that are tolerable by the multimedia decoders. Network layer amendments are impractical since any modification at this layer must be reflected in the routers. Also, IP layer computes the checksum over the IP header only and we believe that packets with errors in the IP header can be potentially damaging and, thus, should be dropped. The transport layer, however, performs checksum over the entire packet (transport layer header and payload) and is, therefore, the major source of packet drops in this scenario. Hence, further analysis of the transport layer is necessary to improve the eventual ''throughput'' reflected at the application layer.

As mentioned previously, UDP Lite addresses this problem by providing the flexibility of partial checksum. The performance of UDP Lite for video over cellular networks was evaluated in [4]. It shows significant improvement over UDP in terms of throughput and PSNR. The simulations were performed on a custom wireless simulator (*WSim*) utilizing error traces provided in [8]. Traces used for this study listed a sequence of corrupt and correct frames without providing error distributions within a corrupt frame. Traces with error rate of 1.58% were employed, which generated a packet drop of approximately 2.1% (for

Table 2
Transition probabilities of packet-level Markov model for traditional 802.11b protocol stack

Bit-rate	P_{gg}	P_{gb}	P_{bg}	P_{bb}	P_{packet}
2	0.999	0.0001	0.6470	0.3529	0.0002
5.5	0.820	0.1794	0.3163	0.6836	0.3614
11	0.376	0.6235	0.1034	0.8966	0.8576

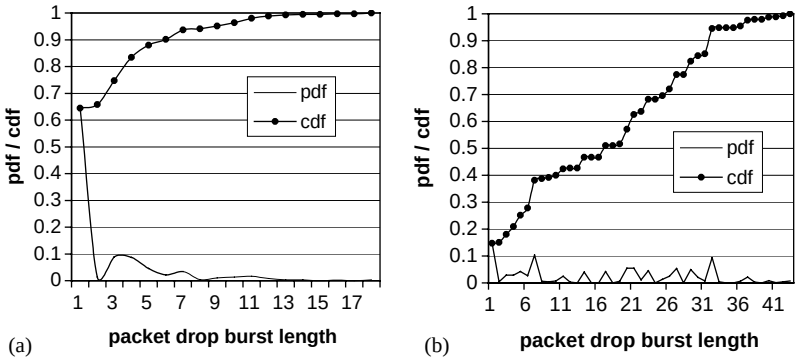

Fig. 4. CDF and PDF of UDP (in conjunction with MAC Lite) packet drop bursts at: (a) 5.5 Mbps and (b) 11 Mbps.

Table 3
Packet drop statistics for UDP (in conjunction with MAC Lite)

Bit-rate	Mean	Standard deviation	Throughput (%)
2	0.00029	0.02329	99.9825
5.5	1.59579	2.82139	64.4225
11	15.6719	11.8852	14.702

UDP without retransmissions). Non-bursty (random) errors were induced at a fixed rate (20%) for each corrupted video frame.[7]

A custom simulation testbed was developed for our simulations. This simulator generated the appropriate UDP, IP and MAC header fields and appended them to the application layer payload. At the client side, the error traces were used to corrupt 802.11b frames before subjecting them to MAC Lite, IP and UDP layer parsing.

4.4. Throughput of UDP with MAC Lite

Fig. 4 shows the CDF and PDF for packet drops of UDP in conjunction with MAC Lite at 5.5 and 11 Mbps. These plots support the comment that most of the corrupted frames being passed by MAC Lite are dropped at the IP and UDP layers. The distributions largely resemble traditional 802.11b MAC statistics (see Fig. 3). The longest packet drop burst for 5.5 and 11 Mbps is 17 and 42 packets, respectively.

Tables 3 and 4 outline the statistics and transition probabilities of packet drop bursts of UDP (with MAC Lite) at 2, 5.5 and 11 Mbps. Table 3 suggests that MAC Lite improves the overall throughput by 0.47%. This improvement is quite insignificant and will be further reduced for longer traces. Also, the Markov model transition probabilities for this case are largely unvaried.

4.5. Throughput of UDP Lite with MAC Lite

The probability distributions in Fig. 5 show a positive improvement in the mean and standard deviation of the packet-drop bursts. For 5.5 Mbps, approximately 95% of the packet-drop bursts are smaller than 3

[7] Our analysis indicates a much higher packet drop in 802.11b networks (at 5.5 and 11 Mbps).

Table 4
Transition probabilities of packet-level Markov model for UDP (in conjunction with MAC Lite)

Bit-rate	P_{gg}	P_{gb}	P_{bg}	P_{bb}
2	0.99989	0.0001	0.64706	0.35294
5.5	0.82096	0.179	0.32484	0.67516
11	0.37831	0.6217	0.10705	0.89294

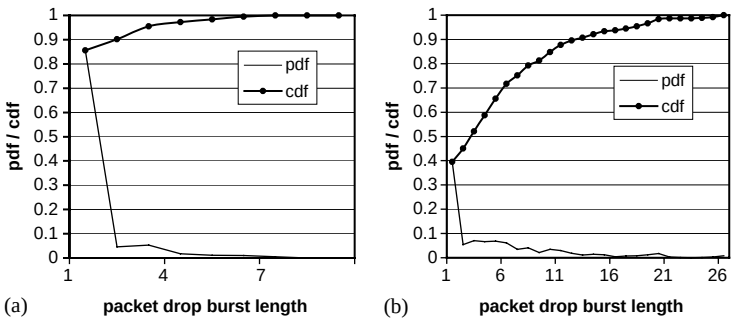

Fig. 5. CDF and PDF of UDP Lite packet loss bursts at: (a) 5.5 Mbps and (b) 11 Mbps.

Table 5
Packet drop statistics for UDP Lite

Bit-rate	Mean	Standard deviation	Throughput (%)
2	0.00006	0.00783	99.9994
5.5	0.33651	0.97775	85.61
11	4.20488	5.44947	39.57

packets. This is an encouraging result since it depicts a clear improvement in the throughput by the UDP Lite-based framework. The longest packet drop burst length is 8 and 25 for 5.5 and 11 Mbps, respectively.

Table 5 shows that the average burst length is much smaller in case of UDP Lite as compared to UDP (4.2 versus 15.67 for 11 Mbps). Also, standard deviation of the UDP Lite distribution is approximately 50% less than UDP. These observations could facilitate the design of an application layer FEC that is capable of combating such impairments. Moreover, the throughput increases by 21.2% and 24.86% for 5.5 and 11 Mbps, respectively (compare Table 5 and Table 3). Transition probabilities of the UDP Lite-based packet-level model are given in Table 6.

4.6. Discussion

The analysis provided in this section clearly depicts that UDP Lite enhances the end-to-end throughput significantly. In addition, it reduces the variance of the packet drop distribution, which in turn mitigates the complexity of designing an application layer error control scheme. However, increase in throughput is not a direct metric of improvement in the quality of real-time media. The number and distribution of errors, in the partially corrupted packets provided by UDP Lite, can severely degrade the quality of perceived media.

Table 6
Transition probabilities of packet-level Markov model for UDP Lite

Bit-rate	P_{gg}	P_{gb}	P_{bg}	P_{bb}
2	0.99994	0.00006	1	0
5.5	0.90178	0.09822	0.5844	0.4156
11	0.5689	0.4311	0.2823	0.7177

Consequently, a focal question needs to be addressed in this context: Can the additional "high-throughput" (corrupted) packets provided by UDP Lite enhance the quality of transmitted real-time media? The answer to this question depends on the number and distribution of errors in a corrupted packet. The following section provides insight into the byte-level error patterns observed in the high-throughput (corrupted) packets.

5. Byte-level error pattern analysis

In this section, we investigate the byte-level error patterns induced by the channel. As mentioned before, our channel comprises of the wireless medium and the 802.11b Physical layer. We, then, utilize this insight to generate an appropriate model for the error phenomenon.

5.1. Byte-level two-state Markov model

Discussions in the previous section have depicted that the packet drops are bursty. The packet-level Markov model motivated the application of a similar model at the byte-level. The error traces were employed to calculate byte-level state transition probabilities for each bit-rate. We focus our discussion to byte-level rather than bit-level analysis since most error or loss protection techniques at the application layer operate on byte boundaries.

Table 7 outlines the state transition probabilities and the overall probability that a byte is corrupted. Note that, P_{gb} and P_{byte} are directly proportional to the rate of transmission, i.e., as the bit-rate is increased, the overall probability of error also increases, thus, channel transitions to the bad state occur more frequently. Again, this observation can be inferred intuitively since the robustness level at the 802.11b physical layer is reduced with the increase in bit-rate.

A potentially anomaly can be observed in Table 7, and that is, P_{bb} for 5.5 Mbps is less than the corresponding P_{bb} for 2 Mbps. Contrarily, probability of byte error (P_{byte}) is increasing with the bit-rate. This necessitates further analysis of the byte-level-error probability distribution in order to decide the suitability of the respective two-state model. Thorough examination of the error traces revealed two observations that contradict the original assumption (i.e., byte-level corruptions are always bursty): (a) In addition to the consecutive byte-level bursts, there existed some small isolated bad bytes; and (b) Bad states had isolated small instances of good bytes amongst a number of corrupted bytes.

The two-state Markov model is incapable of handling the above-mentioned anomalies. Consequently, the following sections address these anomalies and propose an improved model for the "channel" based on our observations of the of the byte-level error patterns. Before proceeding, we present some of the key statistics of the byte-level burst lengths. Fig. 6 illustrates the PDF for byte-level burst-lengths at 2, 5.5 and 11 Mbps. Mean and standard-deviation of these bursts are tabulated in Table 8. It can be observed again that the byte-level bursts at the 2 Mbps are longer than 5.5 Mbps.

Table 7
Byte-level Markov model transition probabilities

Bit-rate	P_{gg}	P_{gb}	P_{bg}	P_{bb}	P_{byte}
2	0.999	0.00002	0.327	0.673	0.00006
5.5	0.991	0.00909	0.368	0.632	0.024
11	0.946	0.05419	0.304	0.696	0.15

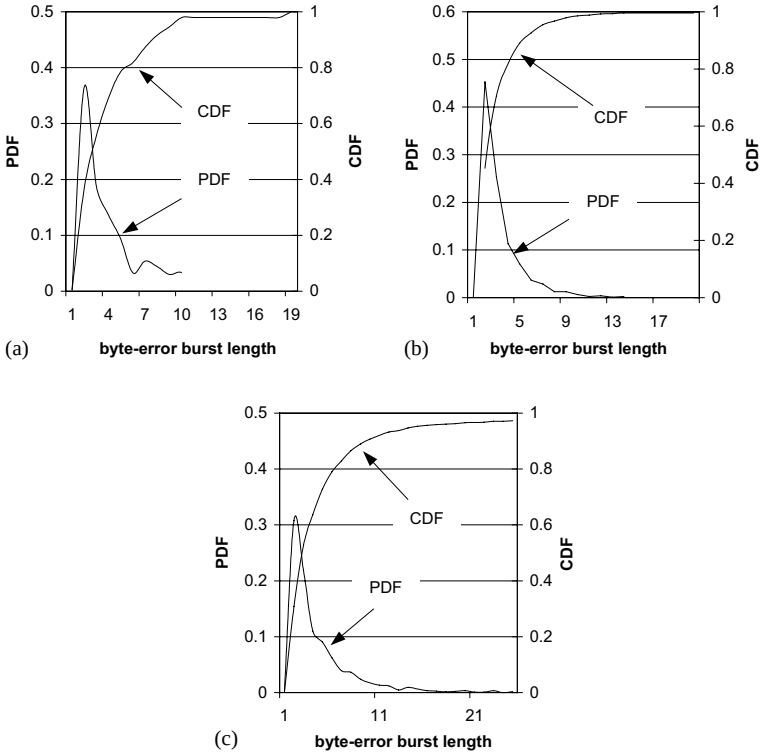

Fig. 6. CDF and PDF of byte-level bursts at: (a) 2 Mbps, (b) 5.5 Mbps and (c) 11 Mbps.

5.2. Modeling of byte-level probability distribution

Due to the diversity of the Gamma distribution in capturing the shape and scale of a wide range of statistical data, we employed non-linear regression analysis to generate a best-fit Gamma PDF for our byte-level burst length:

$$f(x) = \frac{\lambda e^{-\lambda x}(\lambda x)^{\alpha-1}}{\Gamma(\alpha)}$$

for $x \geqslant 0$, $\lambda > 0$, $\alpha > 0$. The resulting Gamma PDFs are shown in Fig. 7. The shape and scale parameters for the fits are given in Table 9. Overall, the Gamma PDF fits the general shape of the collected-data histograms, but is unable to fit the tails. Many error control techniques are not designed for the very worst-case scenario. Hence, distribution of the tail is important for determining an efficient error control scheme.

Table 8
Statistics for byte-level analysis

Bit-rate	Mean	Standard deviation
2	4.232	3.108
5.5	3.3948	2.384
11	5.458	6.722

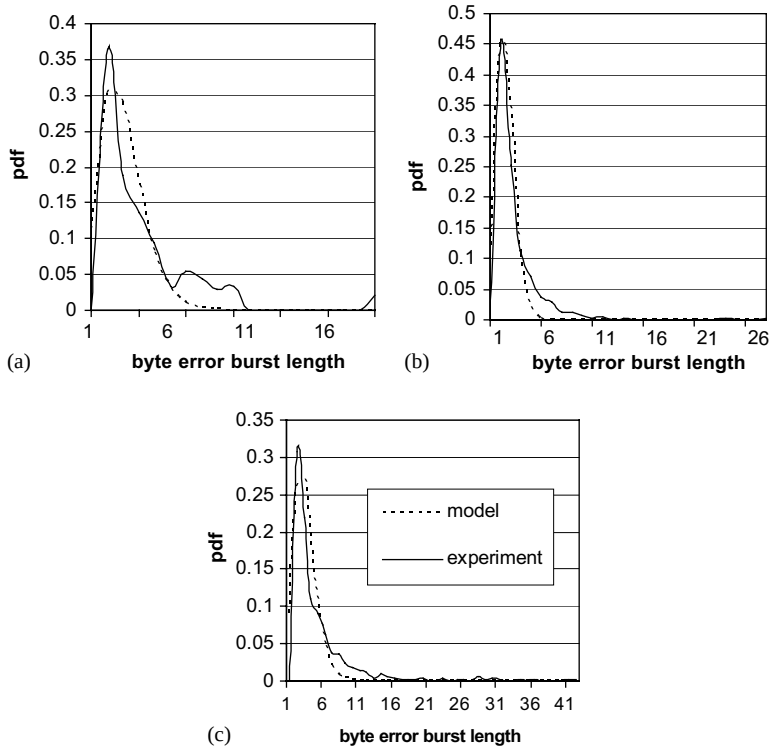

Fig. 7. Gamma approximation for byte-level burst distribution at: (a) 2 Mbps, (b) 5.5 Mbps and (c) 11 Mbps.

Exponential[8] and Pareto distributions were applied to provide a suitable fit for the tail. These fits are shown in Fig. 8. The Pareto distribution provided a good fit as the correlation was greater than 90% for all data rates. The fit provided by the exponential distribution had a lower correlation than the Pareto distribution for all three data rates, but for 5.5 Mbps the fit given by exponential was comparable to the Pareto fit. Thus, it can be said that even though the overall probability distribution looks like Gamma, the tail indeed exhibits a heavy-tail behavior. This characteristic can be attributed to the occurrences of long byte-level errors. Clearly, the channel has a hybrid overall behavior, i.e., a combination of random and burst errors. Therefore, the two-state Markov Model is rendered inadequate with the increase in bit-rate.

[8] Although the exponential distribution is a special case of the Gamma distribution, one can model the tail with Gamma parameters that are different from the Gamma parameters used for the main distribution. In this case, the tail will have the parameter $\alpha = 1$, which represents an exponential random variable, while the main PDF will have $\alpha \neq 1$.

Table 9
Parameters of gamma distribution for 2, 5.5 and 11 Mbps

Bit-rate	α (shape)	λ (scale)
2	4.96682	0.60563
5.5	8.07703	0.31632
11	4.48498	0.726767

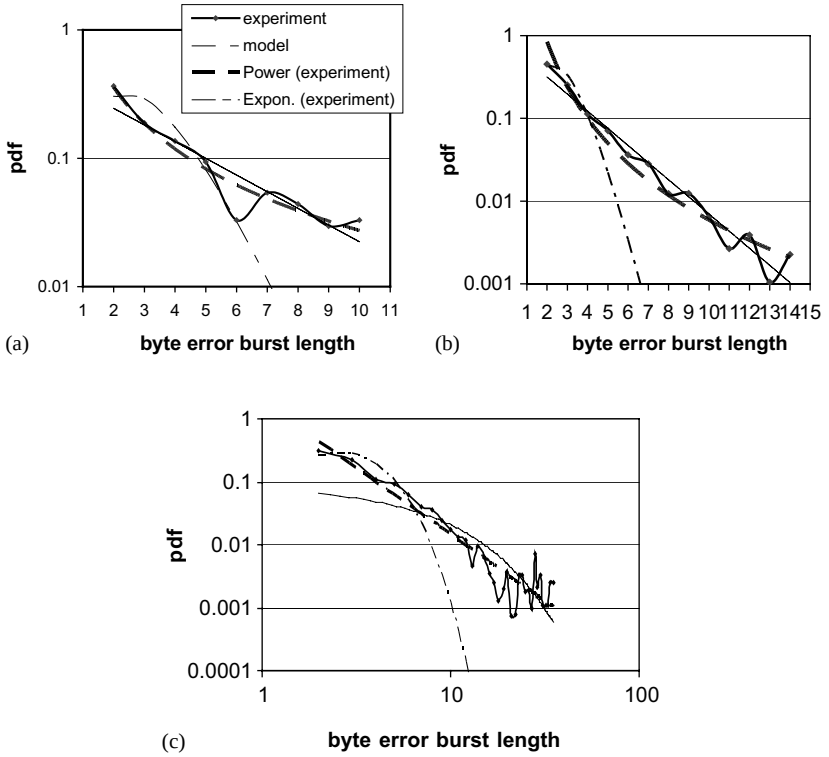

Fig. 8. Tail Modeling of byte-level PDF for (a) 2 Mbps, (b) 5.5 Mbps and (c) 11 Mbps.

The errors exhibit non-stationary behavior and, consequently, in Section 5.3 we attempt to provide a better approximate for the channel by proposing a two-state hierarchical Markov Model (hMM).[9]

5.3. Hierarchical Markov model

The two anomalies discussed in the preceding sections substantiate the need for a better model. Since the small isolated corrupted bytes (noted in Anomaly (a) above) do not deteriorate the overall quality of multimedia, hence, (a) should not be characterized as a bad state. Conversely, the isolated good bytes in Anomaly (b) should be considered as part of a (high-level) bad state rather than a criterion for transition to the good state. In order to incorporate these observations, we developed a simple two-state hMM. It is

[9] This model is abbreviated as hMM to differentiate it from the well-known Hidden Markov Model (HMM).

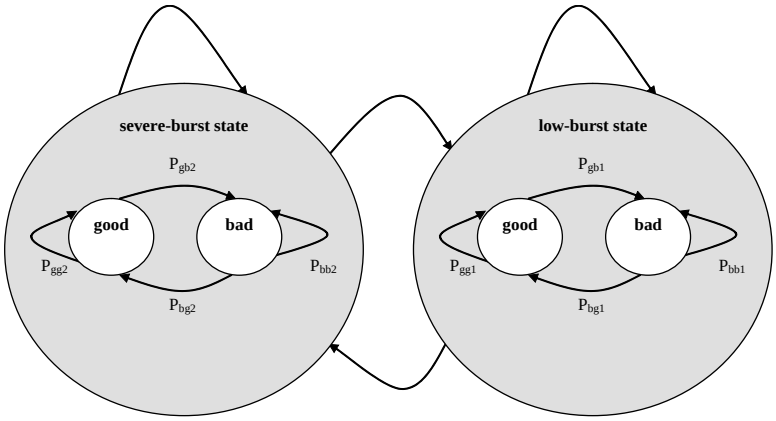

Fig. 9. The hierarchical Markov model.

important to note that this discussion is not applicable to the packet-level model since byte-level bursts (even though interleaved with good bytes) are still spread over multiple consecutive packets. This explains why the two-state Markov model provides a good approximate at the packet-level.

Under the proposed hMM model, "severe-burst" and "low-burst" states were identified in the error traces. Furthermore, each of the aforementioned states had a two-state Markov model as depicted in Fig. 9. One of the challenges of the proposed hMM model is the delineation of the high-level "severe-burst" and "low-burst" states. Ideally, this delineation should be based on the level of impact that these high-level states have on the application layer. This task, however, becomes very difficult due to the wide range of possible degradations that can result from different patterns of burst and random errors. Here, we propose (actually resort to) a simple heuristic for this delineation process. (More elaborate techniques are certainly feasible and are a subject of further work.)

5.3.1. State demarcation heuristic

The flowchart in Fig. 10 outlines the heuristic employed to identify the boundaries of severe-burst and low-burst states. All the bursts in this heuristic correspond to byte-level bursts. We assume that the real-time application layer exhibits some tolerance against small isolated bursts. This caters for Anomaly (a) and the maximum length of such small error bursts is defined as *threshold_1* i.e., if the error burst is smaller than *threshold_1*, we classify it as a low-burst. Severe-burst state is exited on receiving *threshold_2* number of consecutive good bytes. Two distinct two-state Markov chains are employed to model severe-burst and low-burst parts of the error trace. Selection of these two threshold values can be based on the application under consideration. For example in the case of real-time video, the size of an "average" macroblock or a group-of-macroblocks (e.g., a "slice") can be a good threshold.

The probabilities in Table 10 were calculated using the above-mentioned heuristic with *threshold_1* = *threshold_2* = 50 (as an example). This model addresses the anomalies mentioned before since the P_{bb} is increasing with the increase in bit-rate. Also, and as expected, the transition probabilities for the embedded severe- and low-burst states indicate that in a severe-burst state the probability of getting a burst of bad bytes is quite high.

This model and the corresponding curve fits substantiate the two key observations: (1) Error bursts contain small isolated good byte bursts and long "burst" of good bytes include isolated (random) error bytes; and (2) PDF of the burst length can be characterized as a heavy-tail distribution.

At this stage we use this (high-level) channel insight to observe the quality of multimedia rendered by the network in question. In [4] Singh et al., claim that UDP Lite improves throughput, which, in turn enhances

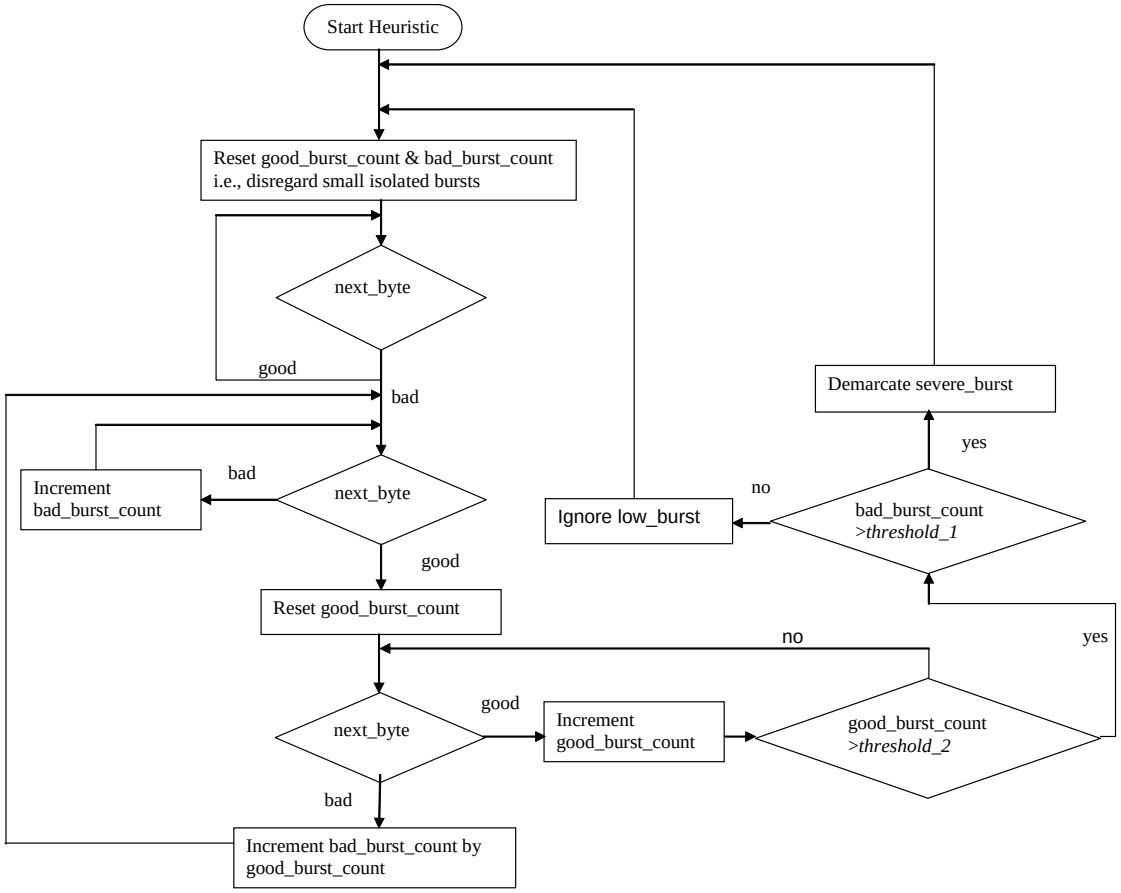

Fig. 10. The state demarcation heuristic.

Table 10
hMM Transition probabilities for 2, 5.5 and 11 Mbps

	2 Mbps				5.5 Mbps				11 Mbps			
	P_{gg}	P_{gb}	P_{bg}	P_{bb}	P_{gg}	P_{gb}	P_{bg}	P_{bb}	P_{gg}	P_{gb}	P_{bg}	P_{bb}
Higher states	0.999	0.00001	0.777	0.223	0.999	0.00008	0.224	0.776	0.999	0.0009	0.013	0.986
Low-burst	0.999	0.0005	0.426	0.574	0.995	0.00489	0.407	0.593	0.983	0.0168	0.371	0.629
Severe-burst	0.918	0.8168	0.341	0.659	0.900	0.09958	0.335	0.665	0.895	0.1049	0.292	0.707

the quality of the perceived media. It can be inferred from the analysis provided in this section that behavior of errors in our 802.11 channel is an amalgamation of randomness and burstiness. Therefore, we believe that a good number of the high-throughput UDP Lite packets will be severely corrupted, and hence, cannot improve the perceived media quality substantially. This necessitates analysis and comparison of the FEC overhead required by UDP and UDP Lite for the delivery of high-bit-rate multimedia. However, a conclusive statement about the amount of redundancy associated with UDP (erasure recovery) and UDP Lite (erasure and error recovery) cannot be made without thorough investigation.

6. Application layer analysis

Discussions in the previous sections have primarily focused on throughput improvements and analysis at layers below the application layer. At this point it is essential to study the impact of the "high-throughput" (corrupted) packets on the application. We wanted to invoke maximum redundancy and error-resilience in order to evaluate the video quality in such favorable conditions. Therefore, we encoded an MPEG-4 video sequence, which can be coded at low-bit-rate, at a relatively higher bit-rate. In addition, we enabled the MPEG-4 error-resilience features such as RVLC and data partitioning. Packetization utilized the "video packet feature", thus, preventing the propagation affect of a single packet loss across multiple packet frames. The video sequence was employed to simulate transmissions at 2, 5.5 and 11 Mbps. Visual evaluation of the simulated transmissions (provided in the following discussion) will clearly outline that despite this form of redundant and resilient compression, due to the channel error patterns, the provided redundancy and error-resilience fail to improve the video quality.

Fig. 11 presents examples of our video experimental results. Fig. 11(b) is consistent with our previous observations, that is, the error-rate at 2 Mbps is negligible. Some modest artifacts can be observed in the corrupted frames but the overall quality of the sequence remains largely unaltered. Therefore, the following visual results are provided for the 5.5 and 11 Mbps cases only. Fig. 11(c) and (d) clearly depict that, even with some video-specific error-resilience, and in the absence of any error- or loss-recovery, the quality of perceived image degrades drastically at higher bit-rates. In addition, comparison of Fig. 11(e) and (f) with Fig. 11(g) and (h) outlines that the high throughput (corrupted) packets provided by UDP Lite fail to improve the perceived video quality. Our conclusions are in agreement with [20], which concludes that current video-specific error resilience techniques cannot deliver high-quality video under bursty (high error-rate) channel conditions. Thus it can be concluded that, irrespective of the transport layer protocol, an application layer FEC is required to deliver high-quality multimedia at bit-rates higher than the 2 Mbps rate. Hence, the remainder of this section addresses the last question raised in Section 1: What level of FEC would be required to support transmission of high-quality and high-bit-rate multimedia?

Analysis in preceding sections concludes that no FEC is required for the 2 Mbps case and the error probability at the 11 Mbps is too high for an efficient FEC scheme to provide significant improvement in media quality. Henceforth, further analysis in this section focuses on the 5.5 Mbps case.

Before proceeding, it is worth mentioning that we need to address both packet losses and errors, where a packet loss is considered as an *erasure*.[10] For error correction, if a codeword has r number of redundant symbols then a maximum of $\lfloor r/2 \rfloor$ transmission errors in that block can be corrected. No error can be corrected if the number of transmission errors in a block is greater than $\lfloor r/2 \rfloor$. Meanwhile, many contemporary FEC schemes have been designed for and applied to erasure recovery (e.g., [6]). Erasures naturally imply knowledge of error locations. This knowledge of the error locations can help the FEC algorithm in error recovery. Hence, if a codeword has a redundancy of r then a maximum of r erasures can be recovered. Here, we focus on block-based FEC schemes due to their popularity and simplicity for recovering and correcting losses and errors.

For UDP-based transport layer schemes, we employ a Generalized Reed-Solomon-based erasure block code (Berklamp algorithm [7]) to recover dropped packets. In the UDP-Lite case, some of the packets reaching the application layer have errors, while some other packets are still completely dropped. Thus, an FEC scheme for UDP Lite should be able to decode errors and erasures simultaneously. Therefore, we use a variant of the above-mentioned FEC algorithm that is capable of joint error and erasure recovery [7]. When using the FEC algorithm for either UDP or UDP Lite, we record a "*decoding error*" if the redundancy in a block is not sufficient to recover and/or correct all the erasures and/or errors. Therefore,

[10] A channel "erasure" is an error, for which the position of the error is known but the magnitude is not. Therefore, a dropped packet can be classified as an erasure.

Fig. 11. (a) MPEG-4 encoded original sequence; Decoded video after simulated transmission using traditional 802.11b protocol stack at (b) 2 Mbps, (c) 5.5 Mbps and (d) 11 Mbps; using UDP (with MAC Lite) at (e) 5.5 Mbps and (f) 11 Mbps; using UDP Lite (with MAC Lite) at (g) 5.5 Mbps, (h) 11 Mbps.

the *decoding error probability* (P_e) quantifies the level of failure (to recover corrupted or lost bytes) that is encountered by the FEC scheme. Other important measures that we considered include "*message packet throughput*" (T_M) and "*bandwidth utilization*" (BW_U), where

$$T_M = 100\left(\frac{\text{Number of message packets received}}{\text{Number of message packets transmitted}}\right),$$

$$BW_U = 100\left(\frac{\text{Number of message packets received}}{\text{Total number of packets (message + redundancy) transmitted}}\right).$$

Table 11
FEC Analysis for UDP with traditional 802.11b MAC and MAC Lite at 5.5 Mbps

UDP with 802.11b MAC				UDP with MAC Lite			
r (%)[a]	P_e	T_M	BW_U	r (%)	P_e	T_M	BW_U
0	1	64.42	64.42	0	1	63.86	63.86
10	0.96	64.39	57.95	10	0.97	63.81	57.93
20	0.75	68.05	54.44	20	0.76	67.41	53.93
30	0.57	72.31	50.62	30	0.58	71.85	50.29
40	0.41	78.05	46.83	40	0.42	77.34	46.40
50	0.28	83.62	41.8	50	0.29	82.87	41.43
60	0.02	98.55	39.42	60	0.03	98.10	39.24

[a] r (%): percentage redundancy.

BW_U quantifies the efficiency of the FEC scheme and is inversely proportional to the redundancy in a codeword. In accordance with previous discussions, we evaluate the performance of FEC for the three combinations of MAC/UDP protocol scenarios: traditional 802.11b MAC with UDP, MAC Lite with UDP, and MAC Lite with UDP Lite. We use an FEC codeword length of $n = 100$ bytes. Each FEC codeword is composed of one byte from a different packet, where each packet consists of 512 bytes. Therefore, each packet contributes to 512 separate FEC codewords, and each codeword spans over 100 packets. The results for the two UDP scenarios are shown in Table 11. The first row of the table shows the performance numbers when no FEC is employed. Note that although the throughput is high when no (or small) redundancy is used, the decoding error probability is very high (100% for the zero-redundancy case). Therefore, in essence Table 11 outlines the tradeoff between efficiency and reliability. Also from the tables, it can be observed that the message packet throughput without any redundancy is slightly greater than the 10% redundancy case. In such a scenario, the redundant packets will not cause a noteworthy reduction in the decoding error probability. Hence, this redundancy can be replaced by message packets, which in turn improve the overall bandwidth utilization.

For the UDP Lite case, we need to consider other parameters since both errors and erasures are involved. Here, it is important to identify the condition that causes a decoding error to occur in this case. Let e_1 be the number of erasures and e_2 be the number of errors in an FEC codeword. Complete recovery is possible iff $e_1 + 2 \times e_2 \leqslant r$. Therefore, if $e_1 + 2 \times e_2 > r$, we have a *decoding error* and no (error-free) recovery is possible. Consequently, insufficient redundancy may cause some packets to have corruptions even after the FEC decoding. Hence, we measure two additional parameters: "corrupted part of message packet throughput (T_{MC})" and "corrupted part of message bandwidth (BW_{UC})", where

$$T_{MC} = 100\left(\frac{\text{Number of corrupted message packets received after FEC}}{\text{Number of message packets received after FEC}}\right),$$

$$BW_{UC} = 100\left(\frac{\text{Number of corrupted message packets received after FEC}}{\text{Number of message packets transmitted}}\right).$$

We also measure the packet *corruption level* (c_l), which is a direct measure of the level of errors in the corrupted packets, where

$$c_l = 100\left(\frac{\text{Number of corrupted bytes in packets with corruptions}}{\text{Total number of bytes in a packet}}\right).$$

Table 12
FEC analysis for UDP Lite at 5.5 Mbps

r (%)	P_e	T_M	BW_U	T_{MC}	BW_{UC}	c_l
0	1	85.25	85.2	23.9	20.4	40.7
10	0.88	86.39	77.8	20.3	15.8	37.6
20	0.38	89.53	71.6	12.6	9	50.4
30	0.12	95.89	67.1	3.17	2.13	4.36
35	0	100	65	0	0	0

Table 12 summarizes the FEC performance numbers based on the above parameters for the UDP Lite scenario. It is clear from the table that higher levels of effective (error-free) throughput are achievable in this case when compared with the traditional UDP scenario. For example, based on our traces at 5.5 Mbps, we were able to achieve 65% effective bandwidth utilization while having zero decoding errors (which implies $T_{MC} = 0$ and $BW_{UC} = 0$). Also, at relatively low decoding error probabilities, we achieved around 70% bandwidth utilization.

A key reason for these improvements in effective throughput when compared with the UDP scenarios can be explained by taking the corruption level into consideration. In the MAC Lite/UDP Lite case, even when a decoding error occurs, the corruption level in already received corrupted packets could be reduced, and consequently, at least some part of a dropped packet could be recovered. Indeed, not all 512 codewords in a block fail to recover any bytes in the case of a decoding error. Such a scenario was not possible in the UDP case since all the 512 codewords had equal number of erasures to correct. Furthermore, when the number of lost packets (erasures) were higher than the number of redundant packets, no recovery was possible in the UDP case. However, in the case of errors and erasures, codewords can have different number of errors, and hence, the number of recoveries will change, accordingly. In effect, if the number of errors in corrupted packets is relatively low (i.e., substantially lower than 50%), then the above property enables complete recovery with comparatively lesser redundancy. It can be observed that there is more than 60% improvement in effective throughput (while achieving complete recovery) in the UDP Lite case when compared with UDP. Hence, for our 5.5 Mbps experiments, completely reliable high-bit-rate multimedia could be delivered while utilizing more than 3.5 Mbps of the total 5.5 Mbps bit-rate (i.e., around 65% utilization). This gives some measure for a lower bound on maximum bandwidth utilization with 100% reliability on an 802.11b LAN for multimedia applications. Naturally, higher bit-rates can be achieved if some errors that can be concealed by the application are acceptable. Hence, we conclude that the MAC-Lite/UDP Lite-based framework in conjunction with application layer FEC exhibits clear throughput improvements over traditional UDP. It is important to note that this conclusion is true regardless of the MAC layer strategy used in conjunction with UDP (i.e., in conjunction with either traditional MAC-based retransmission or MAC Lite-based).

7. Conclusions and summary

In this paper, we studied the feasibility of two high-level options for the delivery of multimedia content over 802.11b wireless LANs. First, we presented our measurements of MAC layer error and loss traces over an 802.11b network at 2, 5.5 and 11 Mbps rates. We utilized these traces to address our first concern (i.e., how much improvement in the "throughput" is actually being gained by using the new UDP Lite-based framework?) and evaluated the transport layer throughput with different combinations of transport- and link layer variants. More specifically, we compared overall (error-free and corrupted) end-to-end

throughput provided by UDP and UDP Lite. In addition, we employed a simple two-state Markov model to approximate the packet drop bursts. We showed that this model provided promising results at all three bit-rates.

Furthermore, we conclude that UDP Lite renders a significant throughput improvement by relaying corrupted packets to the application layer. This led us to the next question, i.e., how much and what type of error patterns are observed in these packets? Since the errors in the traces are largely bursty in nature, we again used the two-state Markov models and derived the respective transitional probabilities. Our analysis revealed that traditional Markov models were not always adequate for capturing 802.11b error and loss patterns since the collected traces showed a variety of burst and random error patterns. We also evaluated the byte-error-burst probability density function and showed that its tail can be characterized using a Pareto distribution. This provides further insight into the type and number of errors observed at the aforementioned bit-rates. Therefore, we proposed a hMM with high-level states, each of which includes (i.e., embedded within it) traditional Markov chains. The high-level states were identified based on the "level of burstiness" within different segments of the observed traces. Our analysis revealed that the hMM framework provided a better approximate of byte-level bursts.

Lastly, we observed that, UDP Lite improves the overall end-to-end throughput. However, due to the non-stationary and high-bursty nature of errors, the quality of perceived media was largely unaltered when compared with a standard UDP-based protocol stack. Hence, the nature of such bursty errors demands FEC protection at the application layer regardless of the underlying transport layer protocol in order to deliver high-bit-rate multimedia. Therefore, we focused on comparing the FEC overhead required by UDP and UDP Lite. This led to one of the key findings of our study, and that is, the amount of FEC overhead required by UDP Lite is considerably less than traditional UDP. Hence UDP Lite provides improvement in bandwidth utilization (i.e., the amount of redundancy required) in order to deliver lossless multimedia. For example, UDP Lite at 5.5 Mbps uses an FEC overhead of (approximately) 2 Mbps to recover packet drops and corruptions. Therefore, an effective bandwidth utilization of more than 3.5 Mbps was achieved. This illustrates the viability of supporting high-bit-rate and high-quality multimedia applications at rates higher than 2 Mbps under realistic conditions.

References

[1] L. Larzon, M. Degermark, S. Pink, UDP Lite for Real Time Multimedia Applications, in: IEEE International Conference of Communications (ICC), Vancouver, British Columbia, Canada, June 1999.
[2] L. Larzon, M. Degermark, S. Pink, The UDP lite protocol, IETF Internet Draft, February 2001.
[3] L. Larzon, M. Degermark, S. Pink, Efficient use of wireless bandwidth for multimedia applications, IEEE International Workshop on Mobile Multimedia Communications (MoMUC), San Diego, California, USA, November 1999.
[4] A. Singh, A. Konrad, A.D. Joseph, Performance evaluation of UDP lite for cellular video, International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), Port Jefferson, New York, USA, June 2001.
[5] ISO/IEC JTC 1/SC 29/WG 11, Text of ISO/IEC 14496-2:2001, Unifying N2502, N3301, N3056, and N3664, Doc. N4350, July 2001.
[6] L. Rizzo, Effective erasure codes for reliable computer communication protocols, ACM Comput. Commun. Rev. 27 (2) (1997) 24–36.
[7] R.E. Blahut, Theory and Practice of Error Control Codes, Addison-Wesley, Reading, MA, May 1984.
[8] R. Ludwig, A. Konrad, A. Joseph, Optimizing the End-to-End Performance of Reliable Wireless Links, ACM Mobicom, Seattle, Washington, USA, August 1999.
[9] C.E. Koksal, H.I. Kassab, H. Balakrishnan, An Analysis of Short-Term Fairness in Wireless Media-Access Protocols, ACM SIGMETRICS, Santa Clara, CA, USA, June 2000.
[10] H. Balakrishnan, V.N. Padmanabhan, S. Seshan, R.H. Katz, A Comparison of Mechanisms for Improving TCP Performance over Wireless Links, IEEE/ACM Transactions on Networking 5 (6) (1997) 756–769.
[11] S.D. Servetto, R. Puri, J. Wagner, P. Scholtes, M. Vetterli, Video Multicast in (Large) Local Area Networks, IEEE Infocom, New York, USA, June 2002.

[12] S. McCanne, V. Jacobson, M. Vetterli, Receiver-Driven Layered Multicast, ACM SIGCOMM, Stanford, CA, USA, August 1996.
[13] H. Ma, M. El Zarki, Broadcast/Multicast MPEG-2 Video over Broadband Fixed Wireless Access Networks, IEEE Network 12 (6) (1998) 80–93.
[14] M. Veeraraghavan, N. Cocker, T. Moors, Support of voice services in IEEE 802.11 wireless LANs, IEEE INFOCOM, New York, USA, June 2002.
[15] J.L. Sobrinho, A.S. Krishnakumar, Real-time traffic over the IEEE 802.11 Medium Access Control Layer, Bell Labs Technical Journal 1 (2) (1996) 172–187.
[16] ISO/IEC 8802-11:1999(E), Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, August 1999.
[17] IEEE Std 802.11b-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz band, September 1999.
[18] J. Postel, User Datagram Protocol, RFC 768, August 1980.
[19] Linux-wlan Home Page. http://www.linux-wlan.org/.
[20] S. Gringeri, R. Egorov, K. Shuaib, A. Lewis, Bert Basch, Robust Compression and Transmission of MPEG-4 Video, ACM Multimedia (MM), Orlando, FL, November 1999.