CROSS-LAYER PROTOCOL DESIGN FOR REAL-TIME MULTIMEDIA APPLICATIONS OVER 802.11b NETWORKS

Syed A. Khayam, Shirish Karande, Michael Krappel, and Hayder Radha
Department of Electrical & Computer Engineering
Michigan State University
East Lansing, MI 48824
+1-517-432-9958
{khayamsy, karandes, krappelm, radha}@egr.msu.edu

ABSTRACT

Inherent vulnerability of the wireless medium renders it more susceptible to errors and losses than classical wired media. In this paper, we evaluate the suitability of protocols and strategies across different layers of the stack to provide real-time services over 802.11b wireless LANs. More specifically, within the context of cross-layer design, we compare the performance of UDP with UDP Lite - a proposed framework, which improves bandwidth utilization by delivering partially damaged packets to the real-time application. First, we study the high-level end-to-end throughput improvement achieved by making cross-layer modifications to support a UDP Lite framework. We compare the quality of perceived media rendered by UDP (dropped packets only) and UDP Lite (dropped and corrupted packets). This formulates one of the key findings of this study, that is, although UDP Lite improves the overall high-level throughput by relaying corrupted packets to the real-time application, it fails to provide significant enhancement in perceived media quality. This can, in part, be attributed to the bursty nature of errors and losses that we observed at the application layer regardless of the selected transport protocol. Finally, we compare the error-recovery/concealment overhead required by UDP and UDP Lite in order to deliver lossless multimedia. We conclude that the overhead required by UDP Lite is considerably lower than UDP, since the received corrupted packets that are delivered by UDP Lite (but not by UDP) facilitate error-recovery.

1. INTRODUCTION

Pragmatism of the UDP/IP protocol suite has provided designers of multimedia applications with the flexibility to render real-time multimedia over Ethernet-based networks. With the advent of wireless networks the concept of real-time communication entered a new realm. The wireless medium due to its vulnerability is more error-prone than its wired counterparts. In addition, deployment issues of wireless networks impede their support of reliable high-bitrates in realistic home/business/classroom settings. Nevertheless, rapidly growing ubiquity of wireless LANs, especially 802.11 networks, has necessitated migration of existing wired-Ethernet technologies to the wireless domain, i.e., to tailor them for bandwidth efficiency and error tolerance. In order to cater for the increased error-rate, enhanced robustness at the physical layer was introduced in 802.11 networks. Naturally, these robust MAC/physical layer protocols were deployed in conjunction with the time-tested UDP/IP protocol stack.

Despite the robustness of the 802.11 physical layer, some of the errors propagate to the link layer. These errors are detected using *frame check sequence* (FCS) and the link layer discards such frames with no regard to the number and location of the errors. A scheme, suggested in [1], [2], tried to address this issue by making adjustments to the protocol stack at the transport and link layers. This variant, known as UDP Lite, exploits the inherent error-tolerance characteristics of multimedia content to improve bandwidth utilization. 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 by the partial checksum. The “Length Field” in UDP header is utilized to specify the partial checksum length. Consequently, this scheme requires the 802.11b MAC-layer to abandon retransmissions and pass partially corrupted packets to higher layers. We refer to this strategy as *MAC Lite*[1].

In this paper we address key questions that determine the suitability of protocols across different layers of the stack in order to support real-time multimedia over 802.11b networks. In particular, we investigate cross-layer schemes that require no modifications to the protocol

---

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

structure, for example, header length, field organization etc. First, we evaluate the “throughput” improvement attained by deploying a UDP Lite-type framework. As mentioned previously, real-time content can inherently tolerate isolated errors and losses. However, error/loss bursts (beyond a certain threshold) adversely affect the quality of perceived media. Therefore, throughput improvement will be more visible when it causes a reduction in packet drop burst length. We observe that packet drops are bursty and, hence, can be modeled as a two-state Markov chain. Model parameters for the stack variants provide insight into the respective packet drop burst lengths. This leads us to the next question, and that is, how much enhancement in perceived media quality is rendered by the high-throughput (corrupted) packets relayed by UDP Lite? Lastly, as a consequence of the preceding discussion, we evaluate the level of application layer loss-protection, error-concealment and/or error-correction required to achieve acceptable quality under the two protocol-stack variants.

The remainder of this paper is organized as follows. Section 2 describes the experimental setup to generate error traces at 2, 5.5 and 11 Mbps. Section 3 studies the throughput achieved by using different variants of the protocol stack. Section 4 proposes a two-state Markov model for the scenarios under consideration. Section 5 compares the multimedia quality rendered by these protocol variants. Section 6 investigates the amount of redundancy required by both frameworks under these lossy conditions. Section 7 summarizes some key conclusions of this paper.

2. 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 Prism2 chipset device drivers (linux-wlan-ng-0.1.14-pre3). Source code of the device driver at the clients was modified to capture screenshots of MAC data frames.

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 bitrate at different stationary client positions with and without LoS. It was observed that with clear LoS, the error rate, at all bitrates, 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. A total of 3 experiments were conducted for each bitrate. Each experiment involved the transmission of 100,000 packets, and 10 error traces per bitrate were generated as a result. These experiments were performed at different times of day to nullify effects of the environment and unrelated traffic.

3. THROUGHPUT ANALYSIS

As suggested by our preceding discussion, UDP Lite (in conjunction with MAC Lite) increases overall high-level throughput by providing the flexibility of partial checksum. The performance of UDP Lite for video over cellular networks was evaluated in [3]. 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 [6]. 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 UDP without retransmissions). Non-bursty (random) errors were induced at a fixed rate (20%) for each corrupted video frame.

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 the corrupted data.

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.

Bitrate (Mbps)	**UDP with MAC (%)**	**UDP with MAC Lite(%)**	**UDP Lite with MAC Lite(%)**
2	99.9825	99.9825	99.9994
5.5	63.8613	64.4225	85.61
11	14.2361	14.702	39.57

Table 1. Throughput of Protocol Stack Variants

Table 1 outlines the throughput of all the protocol stack variants. Clearly, the throughput decreases drastically with the increase in bitrate. Approximately 100% throughput at 2 Mbps construed further investigation at this bitrate inconsequential. In order to improve transmission robustness, most APs drop to a lower bitrate when link quality deterioration is detected. Nevertheless, analysis at higher bitrates is essential to evaluate the feasibility of high-bitrate multimedia applications.

Throughput of traditional UDP in conjunction with MAC Lite is similar to the UDP with 802.11b MAC case, since, the corrupted frames relayed by MAC Lite are being dropped by checksums operating at IP and UDP layers. The last column of Table 1 depicts a positive improvement in throughput rendered by the UDP Lite-based framework.

Here, it is noteworthy that real-time applications are inherently tolerant of isolated errors and packet drops. However, bursty bit-errors/packet-drops can cause significant degradation in multimedia quality. Hence, it is important to study the efficacy of UDP Lite in decreasing the length of packet drop bursts.

4. PACKET LOSS MODELING

The discussion provided in the previous section implies that improvement in media quality is inversely proportional to packet drop burst lengths. We observed that in all protocol stack scenarios, packet drops exhibit bursty behavior. Therefore, we employed a simple two-state Markov model to approximate this bursty packet drop behavior. Table 2, Table 3 and Table 4 summarize the transition probabilities of the packet-level two-state Markov model for all the stack variants. P_{packet} represents the overall packet drop probability.

These transition probabilities reflect the behavior of the channel at 2, 5.5 and 11 Mbps. Robustness of the 802.11 physical layer reduces with the increase in bitrate. Hence, at higher bitrates, the physical layer is more susceptible to propagate errors to the MAC layer. Consequently, P_{bb} and P_{packet} exhibit directly proportional relationship with the bitrate.

Bitrate	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

Table 2. Transition Probabilities of Packet-Level Markov Model for Traditional 802.11b Protocol Stack

Bitrate	P_{gg}	P_{gb}	P_{bg}	P_{bb}	P_{packet}
2	0.999	0.0001	0.6471	0.3529	0.0002
5.5	0.821	0.179	0.3248	0.6752	0.3558
11	0.378	0.6217	0.107	0.8929	0.853

Table 3. Transition Probabilities of Packet-Level Markov Model for UDP (with MAC Lite)

Bitrate	P_{gg}	P_{gb}	P_{bg}	P_{bb}	P_{packet}
2	0.999	0.00006	1	0	0.00006
5.5	0.902	0.0982	0.584	0.416	0.1439
11	0.569	0.4311	0.282	0.718	0.6043

Table 4. Transition Probabilities of Packet-Level Markov Model For UDP Lite

P_{bb} provides a direct measure of packet drop burst length. It can be observed that the P_{bb} decreases significantly in the UDP Lite case (see Table 4). This observation is also intuitively convincing since UDP Lite relays partially corrupted packets as opposed to the other two stack variants. It can be observed that transition probabilities in Table 2 and Table 3 are comparable. This substantiates that the corrupted frames relayed by MAC Lite are being dropped by checksums operating at IP and UDP layers. A more detailed analysis at bit- and byte-level is presented in [7].

At this point a focal question arises; Can the additional “high-throughput” (corrupted) packets provided by UDP Lite enhance the quality of transmitted real-time media? The following section addresses this question by comparing the video quality rendered by the protocol stack variants under consideration.

5. MULTIMEDIA ANALYSIS

In order to evaluate an example of a multimedia application, several MPEG-4 video sequences were compressed and a corresponding set of video bitstreams were generated to simulate transmissions at 2, 5.5, and 11 Mbps. Visual evaluation of these simulated transmissions was conducted and is provided below. It is important to note that MPEG-4 provides error resilience features such as reversible variable length coding, data partitioning, and other techniques [4]. These error resilience techniques were enabled in our simulations.

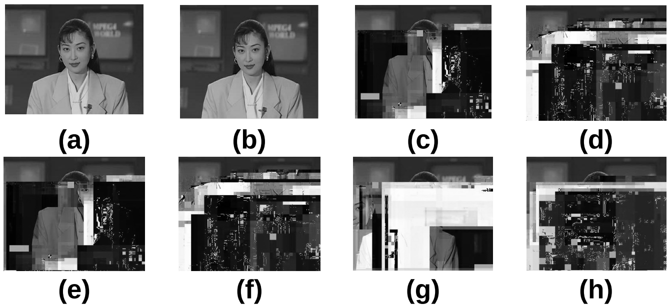

Figure 1. (a) Original sequence; Transmission using 802.11b protocol stack at (b) 2 Mbps, (c) 5.5 Mbps, and (d) 11 Mbps; using UDP (with MAC Lite) at (e) 5.5 Mbps, and (f) 11 Mbps; using UDP Lite (with MAC Lite) at (g) 5.5 Mbps, (h) 11 Mbps

Figure 1. (b) is consistent with our previous observations (i.e., the error-rate at 2 Mbps is negligible). Some modest artifacts can be observed in the corrupted frames. However, the overall transmission quality remains largely unaltered. Henceforth, all visual results are provided for the 5.5 and 11 Mbps cases only. Figure 1. (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 bitrates.

Furthermore, comparison of Figure 1. (e)(f) with Figure 1. (g)(h) outlines that the high throughput (corrupted) packets provided by UDP Lite fail to improve the perceived video quality. Thus, it can be concluded that irrespective of the transport layer protocol, an application layer FEC is required to deliver high-quality multimedia at bitrates higher than 2 Mbps.

6. FEC ANALYSIS

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

For UDP-based transport layer schemes, we employ a Generalized Reed-Solomon-based erasure block code (Berklamp algorithm [5]) 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 the UDP Lite should be able to decode errors and erasures simultaneously. We use a variant of the abovementioned FEC algorithm that is capable of error and erasure recovery [5]. When using the FEC algorithm for either UDP or UDP Lite, we record a "*decoding error*" if the redundancy (r) in a block is not sufficient to recover and correct all the erasures and/or errors. Therefore, the *decoding error probability* (P_e) quantifies the level of failure (to recover corrupted or lost bytes) encountered by the FEC scheme.

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.

It is clear from Table 5 that higher levels of effective (error-free) throughput are achievable with UDP Lite when compared with the traditional UDP scenario. For our 5.5 Mbps experiments, completely reliable high-bitrate multimedia could be delivered while utilizing more than 3.5 Mbps of the total 5.5 Mbps bitrate (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 bitrates 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 an appropriate 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 retransmission–based traditional MAC or retransmission-less MAC Lite).

	***r* (%)**	**P_e**
UDP with 802.11b MAC	60	0.02
UDP with MAC Lite	60	0.03
UDP Lite with MAC Lite	35	0

Table 5. FEC Analysis of Protocol Stack Variants

7. CONCLUSIONS

In this paper, we evaluated two options for the delivery of multimedia content over 802.11b wireless LANs. We conclude that UDP Lite renders a significant throughput improvement by relaying corrupted packets to the application layer. However, the improvement in throughput can only be effective if the packet-loss burst length is reduced. We present a two-state Markov model for packet-drop bursts. This model shows that UDP Lite decreases the packet-drop length significantly. We evaluate the improvement in the quality of perceived media rendered by the increased (corrupted) UDP Lite throughput. Our results show that UDP Lite fails to improve the media quality substantially because of the nature of errors induced by the 802.11b channel. Therefore, FEC protection is required at the application layer regardless of the underlying transport layer protocol in order to deliver high-bitrate multimedia. We investigate the FEC overhead required by UDP and UDP Lite. This formulates 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. This illustrates the suitability of contemporary cross-layer protocol strategies for supporting application-specific multimedia at rates supported by 802.11b networks under realistic conditions.

8. REFERENCES

[1] L. Larzon, M. Degermark, and S. Pink, " Efficient Use of Wireless Bandwidth for Multimedia Applications," IEEE International Workshop on Mobile Multimedia Communications (MoMUC), November 1999.

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

[3] A. Singh, A. Konrad, and A. D. Joseph, "Performance Evaluation of UDP Lite for Cellular Video," Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), June 2001.

[4] 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.

[5] R. E. Blahut, "Theory and Practice of Error Control Codes," Addison-Wesley, May 1984.

[6] R. Ludwig, A. Konrad, and A. Joseph, "Optimizing the End-to-End Performance of Reliable Flows over Wireless Links", ACM Mobicom, August 1999.

[7] S. Khayam, S. Karande, H. Radha, and D. Loguinov, "Performance Analysis of Errors and Losses over 802.11b LANs for High-Bitrate Real-Time Multimedia," to appear in Signal Processing: Image Communication Journal, 2003.