A Statistical Receiver-based Approach for Improved Throughput of Multimedia Communications over Wireless LANs

Syed A. Khayam[†], Muhammad U. Ilyas[†,‡], Klaus Pörsch[†], Shirish Karande[†] and Hayder Radha[†]

[†]Dept. of ECE / 2120 Engineering Bldg.
Michigan State University
East Lansing, MI 48824 USA
{khayamsy, ilyasmuh, poersch, karandes, radha}@egr.msu.edu

[‡]Dept. of CS / School of Arts & Sciences
Lahore University of Management Sciences
Lahore, Punjab 54792 Pakistan
muilyas@lums.edu.pk

Abstract— **Delay-sensitivity of real-time applications stipulates resilience against errors and losses in the multimedia content. Such resilience is particularly important for bandwidth-constrained and error-prone wireless networks. Recent wireless studies have highlighted that significant improvements in bandwidth utilization and multimedia quality can be achieved if the decision to retain or drop "corrupted" packets is made at the application layer instead of medium access, network or transport layers. Such a strategy however necessitates that maximum number of (good and bad) packets are relayed to the application layer while minimizing any modifications (if any) to the widely-deployed UDP/IP protocol stack. To this end, previous studies have proposed that, while ignoring corruptions in packet payload, only packets with errors in the headers should be dropped. In this paper, we introduce a receiver-based scheme which uses the history of a multimedia session to correct packet header errors thereby improving the throughput of real-time applications over wireless local area networks (LANs). The proposed scheme is truly receiver-based and, therefore, does not require any modifications to the source or any intermediate network node. Only minor modifications are required to the protocol stack of the multimedia receiver. Specifically, the proposed scheme generates statistics based on the history of "critical" protocol header fields for active multimedia sessions. These statistics are in turn employed to (a) correct errors in the critical fields and (b) determine if the relevant packets belong to the receiver's session of interest.**

I. Introduction

Most of the contemporary communication channels incur unpredictable and time-varying losses, such as congestion-induced losses over the Internet and bit-error/mobility-induced losses over wireless channels. This data loss is particularly detrimental for real-time communications since their delay constraints generally do not allow retransmission-based recovery of lost packets. In particular, bandwidth-constrained and error-prone wireless channels require multimedia applications to exhibit enhanced resilience against errors and losses. In order to provide such resilience, emerging multimedia (and in particular video) standards (such as MPEG-4 and JVT) have adopted a number of error-resilience and concealment functions. These functions ensure that below a certain loss threshold the distortion in the received multimedia is a continuous/graceful and strictly monotonically decreasing function of the number of losses. Hence, the distortion in multimedia quality at the receiver can be substantially decreased by reducing the amount of data loss.

Due to the error-prone nature of contemporary wireless media, enhanced robustness is provided at the physical layer of wireless protocol stacks. The physical layer cannot correct some of the errors which in turn propagate to higher layers of the protocol stack. These residue errors cause checksum failures at higher layers and consequently lead to a significant number of packet drops on a link employing a conventional protocol stack, such as a UDP/IP/802.11-MAC protocol stack. Previous studies [1]-[7] proposed to reduce packet drops by extending partial protection which covered the headers, while no protection was provided for the data payload. It was demonstrated in these prior studies that relaying corrupted data to the application (in conjunction with error correction) can yield significant improvements in the multimedia quality. However, supporting partial protection necessitates protocol changes at the multimedia transmitter and/or intermediate network nodes. In many realistic scenarios, modifications to multimedia servers cannot be dictated by end-receivers. For instance, public domain video web-casting is oblivious of the end-link communication media and often identical multicast sessions serve heterogeneous users with wired as well as wireless end-links. Thus, transmitter modifications hampered wide-spread deployment of previously suggested solutions.

In this paper, we propose a *receiver-based* approach to relay corrupted packets to the multimedia applications over 802.11b wireless LANs. To the best of the authors' knowledge, this work formulates the first attempt to render a methodology which does not require any modifications to the multimedia source and/or intermediate network nodes. In addition to being truly receiver-based, our proposed methodology retains the single most desirable property of previously proposed methods, and that is, to relay packets with payload errors to the application layer. Our technique uses statistics of correctly received packets to estimate critical header information and hence provides further improvement in the overall application layer throughput. The receiver-based nature of the proposed solution does introduce the notion of false positives, i.e., some packets that do not belong to the multimedia application are forwarded to it. However, we demonstrate that our methodology can be tuned to significantly reduce or completely remove the false positives.

We identify *critical header fields* (CHF) that are required by a multimedia receiver to uniquely identify different multimedia sessions. Our multimedia receiver maintains a CHF histogram from previously received error-free packets i.e., the error-free CHF history is used to compute the a priori probabilities of CHF corresponding to different multimedia sessions. Thus, on receiving a packet our multimedia receiver first checks it for errors using the standard checksums operating at the UDP and the 802.11b MAC layers. If the packet is error-free then it is used to update the CHF histogram. If the packet has errors then we apply the Bayes rule [8] on the CHF histogram in order to ascertain the *most-likely transmitted CHF* for the received packet. If the probability of the most-likely CHF is greater than a certain threshold, then the packet is forwarded to the application that corresponds to the most-likely CHF. We evaluate a range of thresholds to adequately characterize the performance of the proposed methodology.

We evaluate three variants of the above technique. Two of the variants employ only *local* statistics, i.e., CHF statistics of packets destined for the local receiver only. The local statistics variants differ only in the level of attempted recovery of corrupted packets (as explained later in the paper). The third variant exploits the broadcast nature of wireless LANs and maintains *global* statistics of all transmissions on the wireless medium. We again emphasize that our proposed methodology can introduce false positives in all three variants. However, our results illustrate that the false positives can be significantly reduced or completely eliminated by fine tuning the threshold.

The rest of the paper is organized as follows. Section II elaborates our proposed technique for increased throughput in detail. Section III explains the simulation setup that we have employed to generate our results. Section IV shows and interprets the results of all simulation runs. Section V summarizes key conclusions of this paper.

II. METHODOLOGY

Due to the real-time nature of many multimedia applications, throughout this work we assume that UDP is used as the transport layer protocol. We start by identifying the fields in 802.11-MAC, IP and UDP headers that are critical to our technique. We focus on fields that are not liable to change throughout the duration of the transmission of a data stream belonging to a particular session. From the 802.11-MAC header it contains the *destination MAC address*, from the IP header the *source and destination IP addresses* and from the UDP header the *source and destination ports*. We collectively refer to these fields as the *critical header fields* (CHF).

At any time during the data transmission, the CHF for all packets in a datagram stream will be unique and different from those of all other datagram streams in the wireless network. A multimedia receiver maintains a CHF histogram based on the last N *error-free* packets that it receives. (Here, the term error-free refers to packets that pass the frame check sequence (FCS) at the 802.11 MAC layer.) Packets that fail the FCS are called *corrupted* packets. The generation and maintenance of the histogram can be based either on only the error-free packets that were meant for the present receiver or, due to the broadcast nature of wireless LANs, on the error-free packets received for any node in the basic service set. In the former case the histogram will be generated based on *local packet statistics* only while in the latter case the histogram will be based on *global packet statistics*. We propose and evaluate two local statistics variants (LSVs), namely LSV-1 and LSV-2, and one global statistics variant (GSV).

A. Description of the Proposed Methodology

Figure 1 outlines a generalized flowchart of all three variants of the proposed technique. On receiving a packet, the receiver checks the FCS. If the packet passes the FCS and if the packet was destined for this receiver, it passes the packet up to the application. Then it decides whether or not the histogram should be updated with the CHF of the received packet. This decision is dependent on the variant under consideration. If the decision is affirmative, the histogram is updated and the receiver returns to its initial state, else the receiver returns directly to the initial state without updating the histogram.

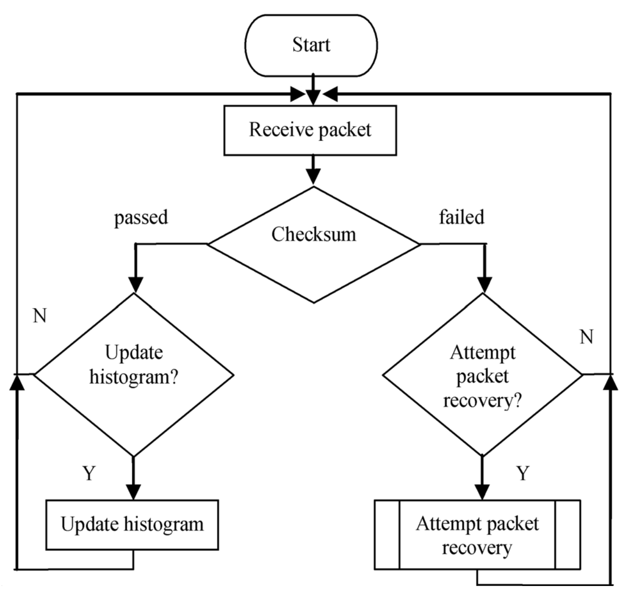

Figure 1. Flowchart of the general methodology.

If the received packet fails the FCS then the receiver has to decide whether it will attempt packet recovery or not. This decision will again vary with respect to the different variants. If the decision is affirmative, the receiver attempts packet recovery by applying the Bayes rule [8] as follows.

Let W be the CHF in the received packet, and the set $\Im = \{X_1, X_2, ..., X_n\}$ be the set of all previously received CHF of the last N packets used for updating the histogram; each $X_i \in \Im$ for $1 \leq i \leq n$ corresponds to a CHF. Therefore, $\bigcup_{i=1}^{n} X_i$ forms the set of all bins of the histogram. The Bayes rule [8] is mathematically expressed as

$$P\left(X_i|W\right) = \frac{P\left(W|X_i\right)P\left(X_i\right)}{\sum_{j=1}^{n} P\left(W|X_j\right)P\left(X_j\right)}. \quad (1)$$

The above expression gives us the probability that, given that the received CHF is W, the actually transmitted CHF was X_i. The a priori probabilities $P(X_i)$ in (1) are derived directly from the CHF histogram.

In order to compute the conditional probability terms, $P\left(W|X_i\right)$'s, on the right hand side of (1), we assume a memory-less channel with a fixed probability of bit-error. The authors agree that based on previous wireless error and loss modeling studies [9]-[14], the assumption of a memory-less channel is somewhat unrealistic. However, a model with memory can be employed only in conjunction with real-time channel prediction and characterization. We are unaware of any current scheme capable of providing such side-information. In the absence of good channel estimates, our memory-less approach is the only viable option. Although our proposed methodology relies on the memory-less premise, the error traces that will be used for simulations later in this paper were collected over an actual 802.11b network under realistic settings. The memory-less assumption, hence, provides a lower bound on the achievable performance. Moreover, we assume that we are making a blind estimation and an a priori value of instantaneous bit error probability is not available. Any knowledge of a more accurate statistic would obviously improve the performance. Thus, the performance curves presented in this paper provide the minimum performance improvements that can be achieved by the proposed methods.

Let the bit-error probability be represented by p. To calculate the conditional probability, $P(W|X_i)$, for a particular X_i, we find out the Hamming distance between W and X_i. The hamming distance will indicate the total number of bits that are different between W and X_i. Now if we assume that the different bits are in fact the bit-errors introduced by the memory-less channel, then $P(W|X_i)$ can be written as

$$P(W|X_i) = p^{hd(W,X_i)}(1-p)^{L-hd(W,X_i)}, \quad (2)$$

where L is the length of the CHF in bits and the function $hd\left(a,b\right)$ returns the Hamming distance between bit sequences a and b. The expression given in (2) renders the probability that X_i was the transmitted CHF which, due to channel bit-errors, was received as W.

The X_i that gives the greatest conditional probability $P\left(X_i|W\right)$represents the most-likely CHF of a received (corrupted) packet. We refer to $P\left(X_i|W\right)$as the *confidence level* that is associated with the most-likely choice of X_i. Let $C_{threshold}$ be the minimum confidence level that is required to establish that W is a distorted version of X_i. That is, if $P\left(X_i|W\right) \geq C_{threshold}$ then the received (corrupted) CHF is replaced with the most-likely CHF and the resulting packet is passed up to the application.

Now that we have defined the generic methodology of our proposed technique, we define the three variants that are evaluated in this paper.

B. LSV-1: Correct All Packets Based on Local Packet Statistics

In the first variant, called LSV-1, the histogram is generated from local packet statistics only, i.e., only error-free packets destined for the "local" receiver are used for updating the histogram. When a corrupted packet is received, LSV-1 attempts recovery regardless of any other considerations.

C. LSV-2 - Correct Selected Packets Based on Local Packet Statistics

The second variant, called LSV-2, is similar to LSV-1 since it also generates the histogram from local packet statistics only. LSV-2 differs from LSV-1 in that it is more selective about its choice of packets for which it attempts packet recovery. When a corrupted packet is received, LSV-2 first reads the destination MAC and destination IP addresses. If either or both of them turn out to be the receiver's own, it proceeds with packet recovery. If both the destination MAC and destination IP addresses are not of the local receiver then the packet is discarded.

D. GSV - Correct All Packets Based on Global Packet Statistics

Unlike the first two variants, the GSV variant exploits the broadcast nature of wireless LANs and the histogram is generated from global packet statistics, i.e. all error-free packets that are picked up by the receiver whether or not they are destined for this particular receiver. (For unencrypted channels, Global statistics can be easily achieved by operating a wireless card in promiscuous mode.) When a corrupted packet is received, the receiver attempts recovery irrespective of any other considerations.

The three variants described above differ in the number of corrupted frames that they attempt to partially correct and the size of the packet history that is maintained by them. These factors directly impact the memory requirements and computational load of these variants.

III. Simulation Setup

To simulate the wireless channel we used the 802.11b wireless bit-error traces of [6] and [7]. For our simulation setup, we assumed the 80.11b network [15], [16] outlined in Figure 2. Our setup consisted of six video servers with IP addresses of well-known multimedia web servers. We simulated three wireless receivers that are associated with the same wireless access point. Each server streamed a unicast video stream to one of the wireless receivers. Hence, there were a total of six video streams that were divided equally among the receivers (two streams per receiver). The video streams were compressed using the H.264/JVT standard [17] such that the streams had the same source bitrate. Distinct transport layer ports were used for each stream at the receivers. Each transmitted packet is corrupted using the bit-error traces thereby simulating traffic at 2, 5.5 and 11 Mbps.

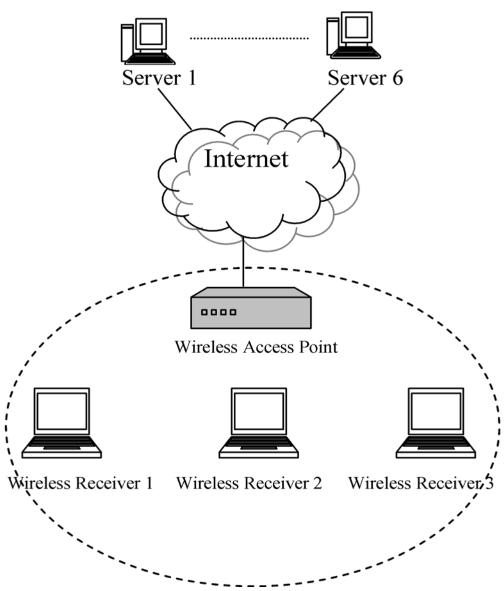

Figure 2. Simulation setup for evaluation of LSV-1, LSV-2 and GSV.

We performed simulations for LSV-1, LSV-2 and GSV with $C_{threshold}$ ranging between $0.5 \leq C_{threshold} \leq 0.99$ at a resolution of 0.01. All three variants were tested for each of the three data rates supported by the 802.11b standard [15], [16], i.e. 2, 5.5 and 11 Mbps. We assumed a fixed bit-error-rate of $p = 0.1$, and a history size of $N = 300$ was employed for all three variants. A packet size of 512 bytes (excluding UDP, IP and MAC headers) wass used for all the simulations. All results reported in this paper were observed at wireless receiver 1. Since the bit-error traces were actually generated at a single receiver [6], similar results are expected at the remaining receivers.

IV. Results

In this section we evaluate the performances of LSV-1, LSV-2 and GSV at data rates of 2, 5.5 and 11 Mbps.

A. *Performance at 2 Mbps*

Figure 3 depicts the performances of LSV-1, LSV-2 and GSV for a range of thresholds at the 2 Mbps data rate. It is clear that the error-rate at 2 Mbps is quite low. This is shown by the "error-free packets received" plot in Figure 3. Since more than 90% of the packets are received without any errors, the performance evaluation reduces to the remaining 10% corrupted packets. It can be easily observed from the "correct application layer throughput"[1] plot that all three variants (LSV-1, LSV-2 and GSV) recover most of the corrupted packets. Note, however, that the false positive rate of LSV-1 is relatively higher than LSV-2 and GSV. This is not surprising since LSV-1 attempts recovery without checking if the packet is actually transmitted for the local receiver. The false positive rate of LSV-2 and GSV is almost zero at all thresholds. Again, this result is rather unsurprising since both LSV-2 and GSV attempt recovery only for packet that were meant to be received at wireless receiver 1. We deduce that, while keeping the overall (correct throughput and false positives) performance under consideration, LSV-1 does not provide satisfactory performance at low thresholds. So in order to effectively deploy LSV-1 in a practical setting, it is important to use a high threshold value. However, LSV-2 and GSV render excellent performance at all thresholds.

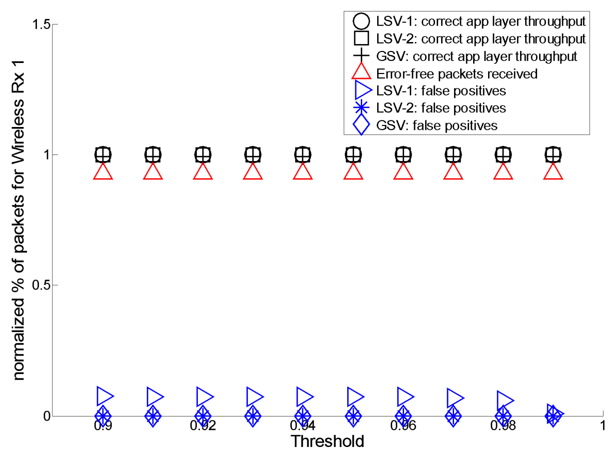

Figure 3. Performances of LSV-1, LSV-2 and GSV at 2 Mbps.

B. *Performance at 5.5 Mbps*

The performances of LSV-1, LSV-2 and GSV at 5.5 Mbps are outlined in Figure 4. It can be seen that the percentage of error-free packets drops significantly in this case (approximately 70%) as opposed to 90% for the 2 Mbps case. The "correct application throughput" is almost 100% for all three variants. The false positives for LSV-1 are very high at low thresholds while LSV-2 and GSV provide negligible false positives at all thresholds. Hence, the deductions from the 2 Mbps case hold true at 5.5 Mbps as well.

[1]Here, *correct application layer throughput* implies packets that were sent for wireless receiver 1 and received by the application layer of wireless receiver 1. While this "correct" throughput may include packets with corrupted data, the decision to retain or drop these packets will be taken by the respective application.

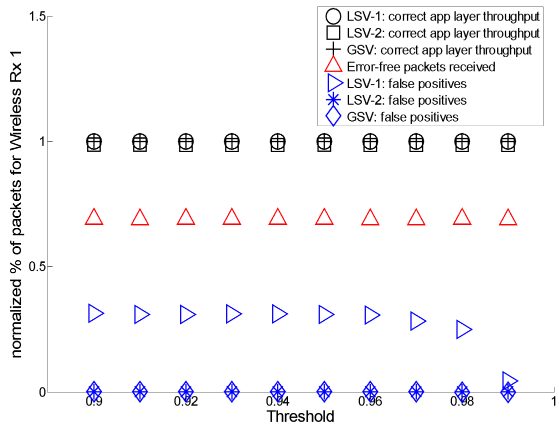

Figure 4. Performances of LSV-1, LSV-2 and GSV at 5.5 Mbps.

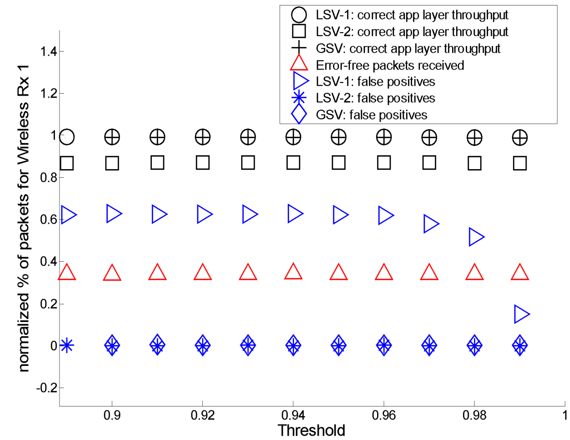

Figure 5. Performances of LSV-1, LSV-2 and GSV at 11 Mbps.

C. Performance at 11 Mbps

Finally, in Figure 5 we evaluate the performance of the three proposed variants at 11 Mbps. As expected, due to the high error-rate at 11 Mbps, the percentage of error-free packets ($\approx 35\%$) is even lower than the 5.5 Mbps case. One important result that should be highlighted in Figure 5 is that the "correct application layer throughput" for LSV-2 is slightly lower than LSV-1 and GSV. This is easily explained by the inherent characteristic that distinguishes LSV-2 from LSV-1 and GSV, and that is, LSV-2 only attempts recovery when either the destination MAC or the destination IP address is not corrupted. On the extremely error-prone 11 Mbps channel, both MAC and IP address fields are simultaneously corrupted by errors at many occasions and that is why LSV-2 fails to perform as well as LSV-1 and GSV. The false positive rate for LSV-1 is very high at 11 Mbps and renders it virtually ineffective at low thresholds. The GSV provides very good performance by providing almost 100% correct throughput in conjunction with very low false positives.

Based on the above simulation results, it can be concluded that the GSV approach provides the best overall performance at all data rates by providing almost 100% correct recovery, while maintaining a very low false positive rate over a wide range of threshold values.

V. Conclusions

In this paper, we proposed a receiver-based methodology to improve throughput of multimedia applications over 802.11b wireless LANs. We evaluated three variants of the proposed technique, namely LSV-1, LSV-2 and GSV. We concluded that LSV-1 has a very high false alarm rate and can only be employed in conjunction with carefully selected receiver thresholds. We also demonstrated that LSV-2 provides very good overall performance at low data rates but renders lower throughput than LSV-1 and GSV at high data rates. While keeping both false positives and throughput under consideration, GSV provides the best performance at all data rates.

References

[1] L. Larzon, M. Degermark, and S. Pink, "UDP Lite for Real Time Multimedia Applications," *IEEE ICC*, June 1999.
[2] L. Larzon, M. Degermark, and S. Pink, "Efficient Use of Wireless Bandwidth for Multimedia Applications," *IEEE MoMUC*, November 1999.
[3] A. Singh, A. Konrad, and A. D. Joseph, "Performance Evaluation of UDP Lite for Cellular Video," *ACM NOSSDAV*, June 2001.
[4] H. Zheng and J. Boyce, "An Improved UDP Protocol for Video Transmission Over Internet-to-Wireless Networks," *IEEE Trans. on Multimedia*, (3)3, pp. 356–365, September 2001.
[5] H. Zheng, "Optimizing Wireless Multimedia Transmissions through Cross Layer Design," *IEEE ICME*, July 2003.
[6] S. A. Khayam, S. Karande, H. Radha, and D. Loguinov, "Performance Analysis and Modeling of Errors and Losses over 802.11b LANs for High-Bitrate Real-Time Multimedia," *Signal Processing: Image Communication*, (18)7, pp. 575–595, 2003.
[7] S. A. Khayam, S. Karande, M. Krappel, and H. Radha, "Cross-Layer Protocol Design for Real-Time Multimedia Applications over 802.11b Networks," *IEEE ICME*, July 2003.
[8] Harry L. Van Trees, "Detection, Estimation, and Modulation Theory, Part I," Wiley: New York, 2001.
[9] G.T. Nguyen, R. Katz, and B. Noble, "A Trace-based Approach for Modeling Wireless Channel Behavior," *Winter Simulation Conference*, December 1996.
[10] M. Zorzi and R. R. Rao, "On Channel Modeling for Delay Analysis of Packet Communications over Wireless Links," *Allerton Conference on Communications, Control and Computing*, September 1998.
[11] H. Balakrishnan and R. Katz, "Explicit Loss Notification and Wireless Web Performance," *IEEE Globecom*, November 1998.
[12] A. Konrad, B. Y. Zhao, A. D. Joseph, and R. Ludwig, "A Markov-based Channel Model Algorithm for Wireless Networks," *ACM WINET*, (9)3, pp. 189–199, 2003.
[13] P. Ji, B. Liu, D. Towsley, Z. Ge, and J. Kurose, "Modeling Frame-level Errors in GSM Wireless Channels," *Performance Evaluation Journal*, (55)1-2, pp. 165–181, 2004.
[14] S. Khayam and H. Radha, "Markov-based Modeling of Wireless Local Area Networks," *ACM MSWiM*, September 2003.
[15] ISO/IEC 8802-11:1999(E), "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," August 1999.
[16] 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.
[17] 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.