Effect of Packet Size on Loss Rate and Delay in Wireless Links

Jari Korhonen, and Ye Wang
School of Computing
National University of Singapore

***Abstract*—Transmitting large packets over wireless networks helps to reduce header overhead, but may have adverse effect on loss rate due to corruptions in a radio link. Packet loss in lower layers, however, is typically hidden from the upper protocol layers by link or MAC layer protocols. For this reason, errors in the physical layer are observed by the application as higher variance in end-to-end delay rather than increased packet loss rate. In this paper, we study the effect of packet size on loss rate and delay characteristics in a wireless real-time application. We derive analytical model for the dependency between packet length and delay characteristics. We validate our theoretical analysis through experiments in an ad hoc network using WLAN technologies. We show that careful design of packetization schemes in the application layer may significantly improve radio link resource utilization in delay sensitive media streaming under difficult wireless network conditions.**

Keywords - Wireless multimedia streaming, Packet-switched wireless networking, Digital wireless channel, Packet error model

I. INTRODUCTION

Wireless networking is an essential part of modern telecommunications, including both low bit rate cellular systems (GPRS, 3G) and high bit rate wireless local area networks (WLANs). Unfortunately, digital radio links are typically prone to bit errors. Depending on the protocols and technologies used in the physical and link layers, bit errors can cause quality degradation that is observable even at the transport and application layers in the form of increased packet loss rate or higher end-to-end transport delay. Both effects are harmful, especially for applications requiring timely delivery of data, for example, multimedia streaming and Internet telephony.

Usually, link or transport layer protocols rely on checksums to detect errors in packets, and discard any packet containing one or more erroneous bits. Large packets are more likely than small packets to be discarded due to bit errors. On the other hand, small packet size leads to higher proportional protocol header overhead. Therefore packet size optimization is an essential research problem in wireless telecommunications.

Initial studies in this field have sought to find optimum predefined packet lengths for certain network designs and channel conditions [1, 2]. To address the variation in network conditions, solutions for adaptive packet size adjustment have been proposed in the literature consequently [3, 4, 5, 6], sometimes in conjunction with other adaptive techniques for radio communications [7].

Existing techniques utilizing adaptive packet length in wireless communications were mainly designed to reside in the medium access control (MAC) or link layer. From the network designer's point of view, it is natural to operate in the MAC layer when issues related to the physical transport channel are considered. However, in many cases, better performance could be achieved if the method for fragmenting data units were optimized for the specific application. Packet fragmentation in the link or MAC layer tends to increase bandwidth usage and it is also likely to cause delay due to segmentation and packet re-assembly operations [8].

A closely related well-known problem is the differentiation between packet losses caused by congestion and bit errors in a wireless channel. This information would be useful for streaming, because wireless losses do not require the transport rate to be reduced but congestion losses do. In the literature, several methods for packet loss differentiation have been proposed, based on either an analysis of end-to-end delay characteristics or bit error detection requiring changes to the link or transport layer protocols [9, 10, 11, 12].

When the wireless MAC layer employs retransmissions, it is reasonable to reduce the MAC layer retransmissions by decreasing the packet size. In streaming applications there is usually a strict deadline for each packet, and therefore retransmission delay is undesired. Another motivation is to improve the efficiency in radio link utilization. Retransmissions waste the channel capacity and decrease the overall performance of the network.

The standard application programming interfaces (APIs) for socket communications, such as Winsock for Windows and Berkeley socket API for UNIX, do not include mechanisms for application to get information about the wireless link state directly from the MAC layer. In application developers' point of view, the actual network beyond the socket API is a "black box". But is it possible to get information about the state of the wireless link using the standard APIs for UDP communications only and analyzing the relative transport times for packets with different sizes? And is it possible to improve the performance of a wireless link using an appropriate application layer framing scheme? We seek to address these questions in this paper.

The rest of this paper is organized as follows. In Section II, we analyze theoretically the relationship between packet length and packet loss rate in the wireless link and the delay characteristics experienced in the application layer. In Section III, we evaluate the theory by means of practical experiments in real wireless system using IEEE 802.11b WLAN. In Section IV, we discuss the results and possible applications as well as fragmentation of encoded multimedia data and protocol issues related to adaptive streaming in general. We conclude in Section V.

II. THEORETICAL ANALYSIS OF A WIRELESS CHANNEL

There are several well-known approaches for modeling error and delay characteristics in wireless channels. Stochastic bit error models are designed to describe how the varying physical characteristics of wireless transport channels can be turned into bit errors that the protocols above the physical layer have to cope with.

The simplest way to model bit errors is to assume that bit error probability is the same for every bit. Independent Bernoulli experiments can then be performed for each bit to decide if the bit is correct or not. In practice, this model is not realistic because bit errors in a real radio channel tend to occur clustered as bursts.

To take the burstiness of errors into consideration, advanced bit error models typically use two-state Markov models. In the 'good' state, all the bits are correct or the bit error probability is very low. In the 'bad' state, bits are incorrect at high probability. Naturally, the transition between the good and bad states has to be modeled as well. Usually, the major differences between statistical bit error models lay just in the probability distribution for the state transitions. The best-known two-state bit error model is the Gilbert-Elliot model [13]. Other somewhat similar but more realistic models have been developed recently [14].

It is rational to use state transition models in network simulators to generate complete bit error patterns. These models, however, are not practical for the analysis of packet loss characteristics above the layer where the damaged packets are discarded.

For the hypothetical basis of our analysis, we make the following assumptions: 1) Bit errors appear as burst. Error burst occurrences follow Poisson process. 2) All damaged packets are discarded. Possibility of bit error detector to fail is omitted. 3) When link layer retransmissions are used, each retransmission attempt for a lost packet causes an extra retransmission delay on top of the normal transport delay. This assumption has been made to derive the theoretical mean end-to-end transport time of the Stop-and-Wait ARQ protocol [15].

Ruling out the congestion-based packet losses, we can assume that there are two cases in which packets are dropped due to bit errors: when packet transmission starts during an error burst or when a new error burst appears during packet transmission. As we suppose error burst occurrences to follow the Poisson process, the probability $p(l)$ of at least one error burst to appear during the transmission of a packet containing l bits can be resolved using Equation (1):

$$p(l) = 1 - \exp(-\lambda l) \qquad (1)$$

Here, $1/\lambda$ is the average number of bits between two error bursts. Equation (1) does not take into account the possibility that packet transmission may begin during an error burst. If the probability that a bit is transmitted within an error burst is denoted as b, the packet loss rate can be solved by combining the two different cases discussed above as in Equation (2):

$$p(l) = b + (1 - b) \cdot (1 - \exp(-\lambda l)) \qquad (2)$$

If we define c=1-b, Equation (2) can be simplified as (3):

$$p(l) = 1 - c \cdot \exp(-\lambda l) \qquad (3)$$

Now it is possible to describe the channel conditions with two parameters, c and λ. Inversely, if the packet loss rate is known with two different packet lengths l_1 and l_2, the two parameters can be solved with (4) and (5), respectively:

$$\lambda = \ln\left(\frac{1 - p(l_1)}{1 - p(l_2)}\right) \Big/ (l_2 - l_1) \qquad (4)$$

$$c = \frac{1 - p(l_1)}{\exp(-\lambda l_1)} = \frac{1 - p(l_2)}{\exp(-\lambda l_2)} \qquad (5)$$

However, most practical wireless communications systems utilize mechanisms at low protocol layers to increase the reliability of packet delivery over a radio channel. Whenever link layer retransmissions are used, the packet loss rate observed at upper layers does not follow the model presented above. In general, a high packet loss rate increases average end-to-end transport time due to the extra delay caused by retransmissions on top of the normal transport delay.

Assuming a constant packet loss rate and an infinite number of possible retransmission attempts, the average number of retransmissions needed to deliver each packet reliably is given in Equation (6). This is also the proportional retransmission overhead.

$$r = \sum_{n=1}^{\infty} p^n = \frac{p}{1 - p} \qquad (6)$$

To approximate the extra delay caused by retransmissions, we can assume each retransmission attempt to cause a constant extra delay, t_r. The theoretical mean end-to-end transport time t can be derived using Equation (7) [14]:

$$t = t_0 + t_r \cdot \frac{p}{1 - p} \qquad (7)$$

Here, t_0 is the average end-to-end transport delay without retransmissions. Because both t_r and t_0 are system-dependent constants, packet loss rate p is the only parameter we can influence to minimize t.

In practice, retransmission delay is not the only factor causing variation in end-to-end transport times. Considering the random variation caused by various factors, such as medium access control algorithms, processor task scheduling and local buffering, the probability distribution curve becomes smoother as shown in Figure 1.

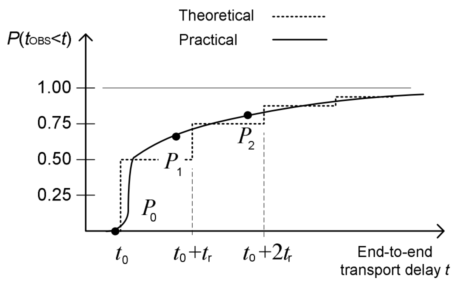

Figure 1. Theoretical and practical distributions for packet transport delays $P(t_{OBS}<t)$ in a wireless network using retransmissions. In this example, the packet loss rate $p = 0.5$.

When the proportional amount of packets that have arrived is known at each point of time, it is possible to estimate the values of t_0 and t_r. Because t_0 is the theoretical minimum transport time, we may assume that t_0 is the shortest detected transport time. We may assume that the probability for the observed transport time t_{OBS} to be smaller than the given t (assuming $t > t_0$) follows roughly Equation (8):

$$P(t_{OBS} \leq t) = 1 - p^{(t - t_0 + t_r)/t_r} \tag{8}$$

Now it is possible to select two known end-to-end delays t_1 and t_2 with the associated packet arrival probabilities P_1 and P_2, and calculate the approximations for the packet loss rate p and retransmission delay t_r with (9) and (10):

$$p = \exp\left(\frac{(t_1 - t_0)\ln(1 - P_2) - (t_2 - t_0)\ln(1 - P_1)}{t_1 - t_2}\right) \tag{9}$$

$$t_r = \frac{(t_2 - t_0)\ln(1 - P_1) - (t_1 - t_0)\ln(1 - P_2)}{\ln((P_2 - 1)/(P_1 - 1))} \tag{10}$$

We have now specified how the distribution of relative packet transport delays can be expressed as a function of packet loss rate, assuming that the number of retransmission attempts is unlimited. Together with the packet loss model, it is possible to solve the parameters c and λ describing network conditions, if the transport delay distribution is known for at least two packet streams using different packet sizes.

This theory can be exploited for two separate purposes. First, time-critical conversational applications can adjust packet sizes to satisfy the desired trade-off between header overhead and packet delay. For example, if it is required that a certain proportion of packets must arrive within a given time limit, it is possible to set the packet size to meet this requirement exactly. The formulas discussed above can be used in the adaptation.

Second, wireless channel utilization can be optimized by minimizing the sum of the header overhead and the retransmission overhead. Let packet loss rate be a function of packet length, $p(l)$, as in Equation (1). Header overhead is h/l, where h is the constant header size and l is the size of the packet payload. Relative retransmission overhead follows the number of retransmissions needed per packet as given in Equation (6). The total overhead can then be expressed as Equation (11):

$$R(l) = \frac{h}{l} + \frac{p(l)}{1 - p(l)} \tag{11}$$

Now we can find the minimum of $R(l)$ either by using trial packets of different lengths, or if $p(l)$ is known, by solving analytically the local minimum from the differential equation:

$$\frac{dR(l)}{dl} = 0 \Leftrightarrow \frac{h}{l^2} = \frac{\frac{d}{dl}p(l)}{1 - p(l)} + \frac{p(l)\left(\frac{d}{dl}p(l)\right)}{(1 - p(l))^2} \tag{12}$$

There is usually no trivial solution for Equation (12) even if $p(l)$ is known. This is why it is usually reasonable to estimate the differential function by approximation. A sample algorithm for searching the optimal packet size is given in the steps below:

1) Transmit continuous flows of packets with different predetermined lengths (l_1, l_2, …).

2) Compute the packet loss rates for each packet flow respectively using Equation (9).

3) Approximate numerically the value of the derivative of function $R(l)$ for each packet size interval using Equation (13):

$$\begin{aligned} R'(l_a, l_b) \approx & \frac{4h}{(l_a + l_b)^2} + \\ & \frac{(p(l_a) + p(l_b))}{(l_b - l_a)(1 - (p(l_a) + p(l_b))/2)} + \\ & \frac{(p(l_a) + p(l_b))(p(l_a) + p(l_b))}{2(l_b - l_a)(1 - (p(l_a) + p(l_b))/2)^2} \end{aligned} \tag{13}$$

4) The optimal packet size l_{opt} is in the range where $R'(l_a, l_b)$ is closest to zero ($l_a < l_{opt} < l_b$).

5) Optionally, select new predefined packet sizes to find more accurate approximation and start this algorithm from the beginning.

6) Repeat until the result is satisfactory.

Of course, packet loss rate and delay characteristics are dynamic by nature. This is why the system needs to continuously monitor their behavior to re-compute the optimal packet size where necessary. The detailed design of the required protocols and algorithms are out of the scope of this paper.

III. PRACTICAL EXPERIMENTS

In practice, the theoretical analysis presented above applies to a highly generic wireless link only. The accuracy and the applicability of the model depend on the actual telecommunications system involved because in modern packet radio systems, different advanced schemes in the hardware and driver software could radically change the behavior of a generic packet radio system. For example, MAC layer retransmissions are widely deployed for the recovery of corrupted frames or frame fragments, and error correcting codes and robust modulation schemes are used to combat bit errors. In addition, hardware compression and adaptive frame fragmentation or aggregation may be used to optimize throughput in varying radio link conditions.

Nevertheless, it should still be possible to estimate link layer characteristics from the application layer. This assumption has motivated us to conduct practical measurements to validate the theoretical results in Section II.

Our test system consisted of two parts: sender and receiver applications running on different laptop computers, both using conventional Winsock API for communications via UDP. We implemented a simple application for the tests. Laptops were connected through an ad hoc network connection using the IEEE 802.11b WLAN standard. An ad hoc topology was chosen to eliminate the influence of the backbone network on the test results.

In each test case, continuous real-time flows of packets were transmitted. Each packet contained a sequence number, timestamp and dummy payload data. The payload was in compressed format to prevent possible hardware level compression schemes influencing actual packet size in the radio channel. The sender application alternated between three different payload sizes: 30, 700 and 1400 bytes, resulting in an average payload size of 710 bytes. The receiver application measured the relative one-way trip times (ROTT) and stored the information in a log file. The test arrangement is illustrated in Figure 2.

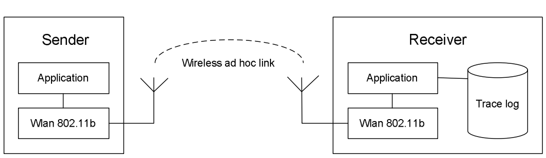

Figure 2. Test arrangement for streaming tests.

The WLAN hardware we used did not allow any changes to the IEEE 802.11b parameters; the MAC layer retransmissions, in particular, could not be turned off. For this reason, we omitted packet loss measurements. However, Arranz et al. have measured frame errors [16], and their results fit reasonably well with our hypothesis for the relationship between packet loss rate and packet size. The main focus of our test was to analyze the cumulative distribution function (cdf) for the relative transport delays.

The test was repeated several times with different physical arrangements, such as changing the distance between the laptops and walking around with one or both of the laptops. The outliers of test cases were identified intuitively from the measurement results and removed from the final analysis.

Figure 3 shows a typical sample trace of the relative transport delays for a sequence of packets. In this example, the transmission rate for the payload is 1.2 Mbit/s. As in Figure 3, packet delays alternate over time. Some individual packets suffer especially high delays, which appear as spikes. In some areas the variation is notably lower, for example between 15 and 35 seconds. From these results, we can infer that the period with low average delay corresponds to low transmission error rate. In other periods we may assume that MAC layer packet losses lead to higher average transport delay due to MAC layer retransmissions.

A shorter extract of the trace is shown in Figure 4, in which the difference between delays for packets with different sizes is clearly visible. This extract is taken from the period of anticipated transmission errors. More accurate information about the delay distribution can be obtained by comparing the cdfs of the relative delays for packets with different payload lengths.

Figure 5 shows the cdf corresponding to the time span from 28 to 35 seconds in Figure 3. The shapes of these curves seem to follow nicely the hypothesis made in Section II (Figure 1); however, there appears to be very little difference between the curves for each payload size, except from the transitions caused by different propagation delays.

The commercial WLAN products have provided us with efficient MAC layer error protection and recovery mechanisms that allow generally stable performance, independent of packet size. However, if the presumably problematic periods are considered, the difference between cdfs is more visible. In Figure 6, the local cdfs corresponding to the trace between 84 sec and 88 sec (roughly corresponding to the period shown in Figure 4) are plotted.

Figure 6 shows the theoretical curves fitted to the observed curves through solving the packet loss rate p and retransmission delay t_r from Equations (9) and (10). The results seem to support our hypothesis discussed in Section II, as the curves fit well together. The resulting values for the parameters are summarized in Table 1.

TABLE 1. PACKET LOSS RATES AND RETRANSMISSION DELAYS RESOLVED

Payload size	***P***	t_r
30 bytes	0.1054	4.89 ms
700 bytes	0.3203	5.03 ms
1400 bytes	0.4860	3.63 ms

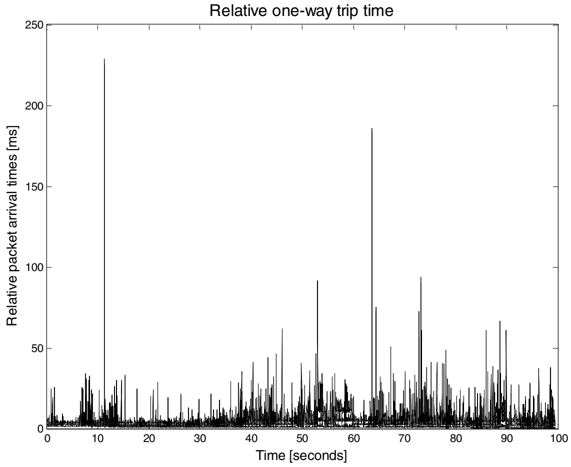

Figure 3. Trace of the relative one-way packet trip times (1.2 Mbit/s)

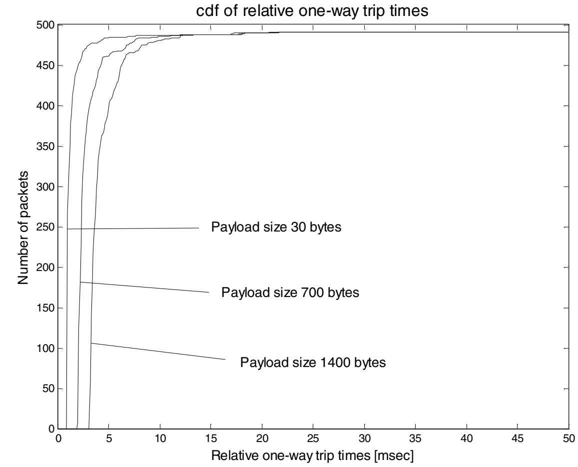

Figure 5. Cumulative distribution function of the relative one-way packet trip times related to the relatively error-free area from 28 sec to 35 sec.

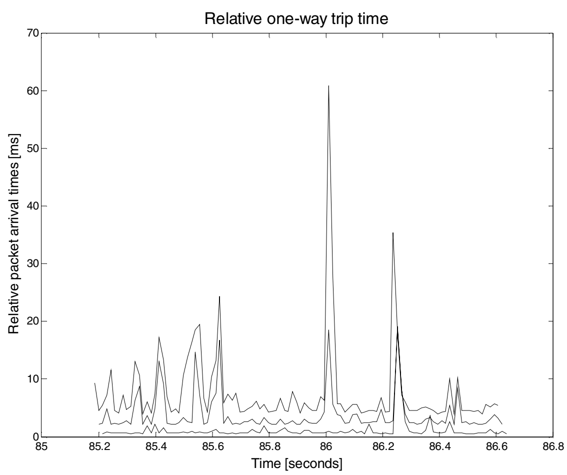

Figure 4. More detailed clip of the relative one-way packet trip times. The lowest plot shows the trace for payload size 30 bytes, the middle plot for 700 bytes, and the highest plot for 1400 bytes.

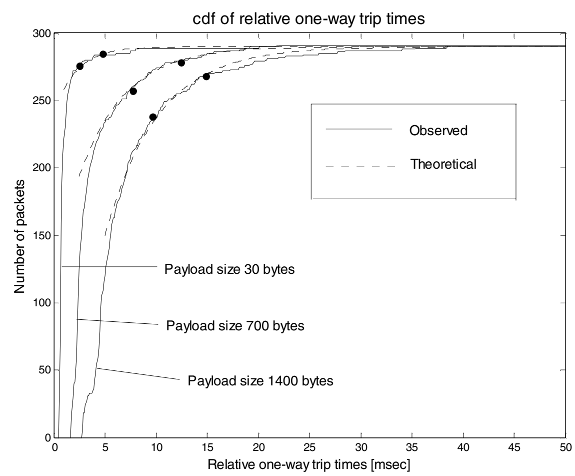

Figure 6. Cumulative distribution function of the relative one-way packet trip times from 84 sec to 88 sec. Dotted line shows the theoretical curve fitted to the observed curves.

By applying the payload sizes and resulting packet loss rates to Equations (4) and (5), we can estimate the bit error rate and burstiness from the parameters c and λ. Because three different predefined packet payload lengths have been used in our experiment, there are three different packet length pairs as shown in Table 2, to use for computations. Using any of them should give relatively close approximations to c and λ. Table 2 shows that in our example, the results are indeed close to each other in every case, and we conclude that the resolved packet loss rates conform quite well to our theoretical analysis.

TABLE 2. BIT ERROR RATES AND BURSTINESS RESOLVED

l_1	l_2	p_1	p_2	c	Λ
30	700	0.1054	0.3203	0.906	4.04e-4
700	1400	0.3203	0.4860	0.899	3.99e-4
30	1400	0.1054	0.4860	0.906	4.10e-4

IV. DISCUSSION

Our observations show that IEEE 802.11b usually provides very good performance even under difficult radio link conditions. This is in line with the research results by others, such as in [16]. In our test system, the WLAN hardware was able to adapt its functionality to the decreased link quality rapidly, providing reliable connection for the end user. Applications with requirements for quite stable bandwidth do not seem to suffer from a significant number of packet losses due to physical channel conditions.

In general, the variation in transport delay in our test results is low. However, temporarily increased jitter seems to be a clear indicator of bit errors in the physical layer. Even in this case, delay spikes do not usually exceed the acceptable level for a conversational real-time application. Clear differences in transport delay and jitter for packets of different size during the bad state have been observed, but the difference is not significant for practical applications. We therefore conclude that application level packet size optimization is not necessary in advanced high bit rate wireless environments, such as IEEE 802.11b WLAN in most cases.

The case is different in lower bit rate, wide-range radio systems. According to the measurements with GPRS and HSCSD in [17, 18], absolute transport delays and delay variances in cellular networks are significantly higher than in WLAN. Also, modern cellular radio networks employ link layer retransmissions, and packet losses observed in the upper layers are relatively rare [18]. In these environments, we can expect transport delay behavior to be similar to that in WLAN, but with much higher absolute values for the delay and jitter. In these networks, packet size optimization in the application layer should be beneficial during the bad state periods.

Another aspect to consider is the avoidance of packet losses in the wireless medium. Several methods for end-to-end differentiation between congestion and wireless losses have been presented in the literature. Typically, these methods are based on analyzing relative transport delays or packet interarrival times [9, 11, 12]. However, they do not consider the impact of possible ACK-based retransmission schemes residing in the MAC layer of the wireless access technology. When a contention-based allocation of wireless resources is used, too large packet sizes may not appear immediately as packet losses, but cause undesirable link layer retransmissions. Under heavy traffic conditions, this may lead to congestion-like behavior in wireless links. Application level packet size optimization could facilitate efficient usage of wireless network resources, improving the service provided to all end users sharing the network.

Typically, it is undesirable to fragment multimedia frames arbitrarily to avoid error propagation. However, there are several ways to implement packet size optimization for multimedia streaming. For short frames, it is possible to simply change the number of frames per packet. There are also more advanced packetization methods relying on the information about the internal structure of media frames, such as slice interleaving in video coding [19] and interframe shuffling of spectral samples in audio coding [20]. Details of such packetization schemes for streaming multimedia are out of the scope of this paper.

V. CONCLUSIONS

In this paper, we have observed packet delay patterns in the application layer of a wireless packet-switched streaming system, which we have studied via both theoretical analysis and practical measurements. The measurement results show that although MAC layer mechanisms of high-speed WLANs generally perform very well even under physical transmission errors, there is a detectable difference in relative transport delay distribution for packets of different lengths.

Our analysis shows that there is a straightforward connection between bit error characteristics and observed delay characteristics. An intelligent streaming application could derive useful information about the underlying network by analyzing the delay pattern for packets with different lengths. This information can be useful in adjusting application level framing under different network conditions to optimize both end-to-end delay and wireless resource utilization.

REFERENCES

[1] C. K. Siew, and D. J. Goodman, "Packet Data Transmission over Mobile Radio Channels", IEEE Transactions on Vehicular Technology, vol. 38, no. 2, pp. 95-101, May 1989.

[2] M. Sarraf, "Effect of Slot Size on TDMA Performance in Presence of Per Slot Overhead", Proc. of GLOBECOM, Dallas, Texas, vol. 1, pp. 604-610, November 1989.

[3] P. Lettieri, and M. B. Srivastava, "Adaptive Frame Length Control for Improving Wireless Link Throughput, Range, and Energy Efficiency", Proc. of INFOCOM '98, San Francisco, California, vol. 2, pp. 564-571, March 1998.

[4] S-T. Sheu, Y-H. Lee, M-H. Chen, Y-C. Yu, and Y-C. Huang, "PLFC: The Packet Length Fuzzy Controller to Improve the Performance of WLAN under the Interference of Microwave Oven", Proc. of GLOBECOM, San Francisco, California, vol. 3, pp. 1427-1431, November 2000.

[5] J. Yin, X. Wang, and D. P. Agrawal: "Optimal Packet Size in Error-Prone Channel for IEEE 802.11 Distributed Coordination Function". Proc. of WCNC, Atlanta, Georgia, vol. 3, pp. 1654-1659, March 2004.

[6] E. Modiano: "An Adaptive Algorithm for Optimizing the Packet Size Used in Wireless ARQ Protocols". Wireless Networks, vol. 5, no. 4, pp. 279-286, August 1999.

[7] C. Chien, M. B. Srivastava, R. Jain, P. Lettieri, V. Aggarwal, and R. Sternowski, "Adaptive Radio for Multimedia Wireless Links", IEEE Journal on Sel. Areas in Comm., vol. 17, no. 5, pp. 793-813, May 1999.

[8] --: "Transparent end-to-end packet switched streaming service (PSS); RTP usage model". 3GPP TR 26.937 V6.0.0, 2004.

[9] S. Cen, P. C. Cosman, and G. M. Voelker: "End-to-End Differentation of Congestion and Wireless Losses". IEEE/ACM Transactions on Networking, vol. 11, no. 5, pp. 703-717. October 2003.

[10] J. Garcia, and A. Brunstrom: "Checksum-based Loss Differentiation". Proc. of MWCN, Stockholm, Sweden, pp. 244- 248, September 2002.

[11] S. Biaz, and N. H. Vaidya: "Discriminating Congestion Losses from Wireless Losses using Inter-Arrival Times at the Receiver". Proc. of ASSET, Richardson, Texas, pp. 10-17, March 1999.

[12] Y. Tobe, Y. Tamura, A. Molano, S. Ghosh, and H. Tokuda: "Achieving Moderate Fairness for UDP Flows by Path-Status Classification". Proc. of LCN, Tampa, Florida, pp. 252-261, November 2000.

[13] E. Elliot, "Estimates of Error Rates for Codes on Burst-Noise Channels". The Bell Syst. Tech. Journal, vol. 42, pp. 1977-1997, September 1963.

[14] A. Köpke, A. Willig, and H. Karl, "Chaotic Maps as Parsimonious Bit Error Models of Wireless Channels", Proc. of IEEE INFOCOM, San Francisco, California, March 2003.

[15] A. Leon-Garcia, and I. Widjaja, *Communication Networks*, McGraw-Hill Co - Singapore, 2000, pp. 294-296.

[16] M. G. Arranz, R. Aguero, L. Munoz, and P. Mähönen: "Behavior of UDP-Based Applications over IEEE 802.11 Wireless Networks". Proc. of PIMRS, vol. 2, pp. F-72 – F-77, October 2001.

[17] J. Korhonen, O. Aalto, A. Gurtov, and H. Laamanen, "Measured Performance of GSM HSCSD and GPRS". Proc. of ICC, pp. 1330-1334, 2001.

[18] R. Chakravorty, and I. Pratt: "Performance Issues with general Packet Radio Service". Journal of Communication and Networks, vol. 4, no. 2, December 2002.

[19] T. Stockhammer, T. Viegand, T. Oelbaum, and F. Obermeier: "Video Coding and Transport Layer Techniques for H.264/AVC-based Transmission over Packet-Lossy Networks". Proc. of ICIP, Barcelona, Spain, vol. 3., pp. 481-484, September 2003.

[20] J. Korhonen: "Error Robustness Scheme for Perceptually Coded Audio Based on Interframe Shuffling of Samples". Proc. of ICASSP, Orlando, Florida, pp. 2053-2056, May 2002.