

Using RTT Variability for Adaptive Cross-Layer Approach to Multimedia Delivery in Heterogeneous Networks

Baek-Young Choi, *Member, IEEE*, Sejun Song, Yue Wang, and Eun Kyo Park

Abstract—A holistic approach should be made for a wider adoption of a cross-layer approach. A cross-layer design on a wireless network assumed with a certain network condition, for instance, can have a limited usage in heterogeneous environments with diverse access network technologies and time varying network performance. The first step toward a cross-layer approach is an automatic detection of the underlying access network type, so that appropriate schemes can be applied without manual configurations. To address the issue, we investigate the characteristics of round-trip time (RTT) on wireless and wired networks. We conduct extensive experiments from diverse network environments and perform quantitative analyses on RTT variability. We show that RTT variability on a wireless network exhibits greatly larger mean, standard deviation, and min-to-high percentiles at least 10 ms bigger than those of wired networks due to the MAC layer retransmissions. We also find that the impact of packet size on wireless channel is particularly significant. Thus through a simple set of testing, one can accurately classify whether or not there has been a wireless network involved. We then propose effective adaptive cross-layer schemes for multimedia delivery over error-prone links. They include limiting the MAC layer retransmissions, controlling the application layer forward error correction (FEC) level, and selecting an optimal packet size. We conduct an analysis on the interplay of those adaptive parameters given a network condition. It enables us to find optimal cross-layer adaptive parameters when they are used concurrently.

Index Terms—Cross-layer approach, heterogeneous networks, round-trip time (RTT) variability.

I. INTRODUCTION

THE popularity of wireless networks, particularly 802.11 wireless LAN (WLAN) has grown rapidly in recent years, beyond just an access network extension to the wired infrastructure. Delay-sensitive streaming multimedia transfer is becoming

an increasingly important application over wireless networks. A simple individual scheme, such as receiver prebuffering, is not suitable as it takes several or tens of seconds. It has recently led to many cross-layer proposals over wireless networks. However, those proposals typically assume that the access network type is *known a priori*.

While an individual suggestion for a cross-layer design in isolation may appear appealing in a specific case, its adoption could be limited in *heterogeneous environments* with diverse access network technologies and time varying network performance. Thus, a more holistic approach should be made for a wider adoption of cross-layer approach. The first and foremost integral step is an *automatic detection of underlying access network*, so that appropriate schemes can be applied without manual configurations. Based upon the varying link characteristics, then, *flexible and adaptive* schemes should be exercised. These are the two main issues we address in this paper.

Understanding the characteristics of network connection types and identifying them automatically using end-to-end approach is very useful for many scenarios, not only for adaptive cross-layer protocol designs. The use of network type detection includes TCP congestion control, bandwidth measurements, overlay network construction, and rogue access point detection.

The congestion control algorithms including many TCP variants and TCP friendly multimedia protocols have been originally designed for wired networks. Those protocols interpret all data loss as congestion in the network, and in case of data loss, the rate control algorithms slow down the transmission rate. For example, TCP Vegas uses the estimate of min round-trip time (RTT) in its congestion avoidance algorithm [1], TCP Reno [2] assumes packet loss is caused by network congestion, and TCP-friendly rate control (TFRC) [3], [4] is designed to provide rate control for unicast multimedia flow operating on the wired network based on TCP Reno's throughput equation. Along with the recent increment of the wireless deployment to access the Internet services, several studies show the performance degradation in the wireless networks [5] in relation with the congestion control algorithms in TCP. In a wireless network, it is no longer appropriate to assume that most losses are caused by congestion. Data loss is often caused by the relatively low quality of the wireless link as well as by the handover events. If data get lost for some other reasons than congestion, then performance is unnecessarily degraded as the rate control algorithm reduces its transmission rate in response to the loss. Although there have been several TCP protocol studies [6]–[9] and TFRC protocol works [10]–[13] to improve performance on the wireless net-

Manuscript received October 07, 2008; revised May 27, 2009. Current version published September 16, 2009. This work was supported in part by the U.S. National Science Foundation under Grant No. 0729197. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. National Science Foundation. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Aggelos K. Katsaggelos.

B.-Y. Choi and E. K. Park are with the University of Missouri, Kansas City, MO 64110 USA (e-mail: choiby@umkc.edu; ekpark@umkc.edu).

S. Song is with Texas A&M University, College Station, TX 77843 USA (e-mail: song@entc.tamu.edu).

Y. Wang was with the University of Missouri, Kansas City, MO 64110 USA. He is currently an information systems consultant in Beijing, China (e-mail: wangx82@gmail.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TMM.2009.2026103

work, the practical deployment has been known to be very difficult due to the heterogeneous nature of the network.

In many bandwidth measurement tools [14]–[16], delay is used as an important parameter reflecting data transmission time. However, in wireless networks, link layer retransmission is performed for reliability to compensate error-prone channel, which results in longer delay regardless of data size. In peer-to-peer applications or application overlay constructions, a node may prefer neighbors with high bandwidth wired network connection over ones with wireless connection which might be lossy and/or unreliable.

Classifying connection type can also be used for identifying rogue access points. A monitor node within an access network determines whether a host is connected via wired or wireless connection. If a host with a wireless connection is not authorized when compared with the authorized WLAN host list, the AP attached by that host is detected as a rogue AP. This software-based method is more scalable and cost effective than using RF scanner devices [17], [18].

One may argue access network type can be reported explicitly by an end system. However, accurate identification is often not trivial. A compromised system may indicate its connection type inaccurately for malicious purposes, and some nodes may not want to reveal their connection type.

In this paper, we first present our study on the RTT of different network environments using end-to-end approach. Our primary intent is to point out the quantitative characteristic differences of RTT variability (i.e., constant portion is removed with minimum delay) between wired and wireless access network environments. To begin with, we have conducted extensive experiments of the RTT variability from various network environments by analyzing various statistics of RTT on both wired and wireless networks. The study quantitatively verifies the user expectation that wireless is inherently not as reliable and wireless will under perform than wired networks.

We analyze the RTT patterns, such as RTT probability distribution, timeout percentage, hour-of-day drift, and packet size effects, in order to seek definitive links between wireless versus wired infrastructure. Such quantitative characterizations can lead to wireless protocol design enhancement so that RTT is not as volatile. In addition, the knowledge (our hypothesis) of wired versus wireless infrastructure can aid network monitoring/tuning tools to make better decisions, or more accurate diagnosis: the observation of RTT loss in the wired network suggests congestion, whereas such loss in the wireless environment is common. We are able to come up with a simple set of testing rules to classify whether or not there has been wireless network involved, which can be used in adaptive protocol design as well as realistic simulation scenarios for other studies. Our work is unique in the extensive experiments on RTT comparison between wired and wireless environments, the quantitative analysis on RTT variability using *end-to-end approach* and enabling method for a *simple and effective* scheme to classify connection type.

The variability of RTT in wireless network is mainly caused by the MAC layer retransmissions and the probability of interference by noise increases with the packet size due to the packet time in the air. Thus, it is desirable to optimally decide the max-

imum number of MAC layer retransmissions and the packet size, given a network condition and other possible adaptation parameters. Once the receiver is identified from a server as accessing multimedia streams from a wireless network, several effective cross-layer schemes for wireless network can be applied. We propose *adaptive schemes* for multimedia transfer particularly in the context of video delivery over a wireless network given a channel condition. The adaptive parameters include the MAC layer retransmission limit, the level of application layer FEC, and the packet size. We provide an analysis on the interplay of those adaptive parameters and the performance trade-offs, given a network channel condition. The analysis provides insights on how the proposed adaptive schemes can be best utilized concurrently.

The remainder of this paper is structured as follows. In Section II, the background and related work are discussed. We describe our measurement methodology for various access networks in Section III. Then we describe experimental results for automatic access network technology detection in detail in Section IV. In Section V, we discuss cross-layer schemes and analysis for multimedia delivery. Finally, we conclude our paper in Section VI.

II. WLAN BACKGROUND AND RELATED WORK

The 802.11 WLAN standard has two different medium access control mechanisms, the distributed coordination function (DCF) and the point coordination function (PCF). The DCF is the basic mechanism which uses carrier sense multiple access with collision avoidance (CSMA/CA). In this mode, a station senses the medium when it wants to start a communication. The station uses a persistence strategy with back-off until it finds the channel idle. After it finds the channel idle, it waits for a period of time called distributed interframe space (DIFS) and then sends a request to send (RTS) signal. RTS and clear to send (CTS) signals are used for a channel reservation with a small time gap of short interframe space (SIFS). Once the channel is reserved, a data frame is sent and acknowledgement will be followed with SIFS, if successful. Otherwise, retransmission will be made after exponential backoff.

There are multiple sources of performance degradation in an 802.11 network. In the physical layer, 802.11 shares the unlicensed 2.4-GHz ISM band with many other devices such as a microwave oven and cordless phone. A sender detects RF energy and delays transmission until the channel is quiet (CSMA/CA). An 802.11 packet in the air may be corrupted due to the interference from other devices or packets, or overpowered at the receiver. In the link layer, multiple levels of delays are involved including a queueing delay at AP, exponential backoff on loss, and packet transmission time. The packet transmission time is a function of the packet size and the encoded frame rate. 802.11 drivers often encode data at a lower rate after the packet loss for the less error-prone packet delivery. It should be noticed that most of the delay components are not exposed directly via upper layer protocol features.

Many wireless performance studies tackle the capacity issues [19], such as how a single slow node brings down the all nodes' throughput [20]. Reference [21] identifies the pitfalls of wireless monitoring and provides solutions, such as merging mul-

multiple sniffer traces and their placement in relation to AP positions. Reference [22] explores characteristics of the campus WLAN environment that provide opportunities for caching, prefetching, coverage planning, and resource reservation. One previous study on wireless performance [23] shows that various interferences, hand-off, and AP queueing degrade the throughput. The research in [24] and [25] identifies 802.11 traffic by using passive measurements on a gateway within the access network.

There are several previous studies that measure and report TCP RTTs in relation to network performance measurement, congestion, and monitoring. Bryan Veal *et al.* compare two passive methods to measure TCP RTT [26]. Another large-scale study shows great RTT variability within TCP connections among multiple Internet destinations [27]. The recent TCP RTT study focuses on a specific university web server and external hosts [28]. The work in [18] is close to ours, as it aims to identify access network type with probes. However, they use distributional statistics of absolute delay rather than delay variability.

To the best of our knowledge, our work is the first to provide an extensive and quantitative analysis on RTT variability using an end-to-end approach to find simple and effective classification characteristics between wired and wireless environments.

There are many solutions proposed for video transmission over wireless networks. In [29], the authors propose an error protection scheme for video transmission over WLAN using priority queueing at the network layer and retransmission limit adaptation at the link layer. A packet coding scheme and a protocol called “complete UDP” are proposed in [30], which use a physical layer and a radio link layer for packet level error recovery. Hybrid ARQ schemes, in which the rate of the associated FEC is changed adaptively based on the channel conditions, have been discussed in [31] and [32]. The author in [33] addressed the issue of error-control mechanisms under delay constraints. Interleaved FEC codes and Go-Back-N ARQ schemes were investigated for a burst-error channel. Reference [34] presents a cross-layer adaptation framework and a prototype implementation, considering an end system environment such as CPU frequency, CPU allocation, and application quality in mobile systems. While our proposed schemes on a cross-layer approach can be used with some of the above techniques complementarily, our main focus is to find the optimal parameters of the adaptive schemes under time varying channel conditions and application constraints.

III. RTT MEASUREMENT METHODOLOGY

We consider various network environments for data collection. For the wireless environments, there could be main differences between an isolated home and an enterprise. The former has a single wireless access point (AP) as a simple range extender for a wired network. However, the latter may have tens or hundreds of distinct APs, which are carefully located and designed to cope with the radio-frequency (RF) survey and thoroughly managed to minimize contentions, maximize throughput, and provide an illusion of seamless coverage.

For the comparison between wireless versus wired network environments, we chose three different test settings, (“home” as Data Sets I, “enterprise” as Data Sets II, and “university

campus” as Data Sets III), which are typical WLAN user environments to have consistency and performance questions comparing with wired networks. These environments differ in access network technologies (i.e., Ethernet, 802.11a/b/g), ISP bandwidths, geographic locations, network equipments, and the ability to control the noise of the collection environments.

We describe in detail the three different test setting scenarios as follows.

A. Data Sets I (Home Environment)

At home, a Linksys WLAN AP Model WRT54G [35] configured for 802.11b is used to have four computers (three wireless, one wired) to access the Internet. The ISP is Road Runner cable modem with 786-kbps bandwidth.

1) *Scenario A:* We collect data from a quiescent environment by eliminating other device usage including the Internet, microwave oven, and cordless phone.

2) *Scenario B:* We collect data from a noisy environment by turning on a microwave oven to inject RF noise, which shares the unlicensed spectrum of 2.4 GHz with 802.11.

B. Data Sets II (Enterprise Environment)

At a business corporation building with about 900 employees, the WLAN APs are the Cisco 1130 AG configured for 802.11a. WPA2 is enabled on the AP. The ISP is AT&T business class DS3 with 45-mbps bandwidth to the Internet. The WLAN environment has only one good radio signal AP in range (little AP overlapping/interfering). This is the result of a careful WLAN site-survey prior to deployment at work to avoid overlap. For wired network connectivity, 100-Mbps Ethernet is used.

C. Data Sets III (University Campus Environment)

At UMKC campus, WPA2 is enabled on the Cisco AIR-AP1131AG-A-K9 AP configured for 802.11g. The ISP is MORENET, with 110-mbps link speed to the Internet.

Although there are several passive RTT measurement methods available, we choose to use an active method to collect RTT data. Since our goal is to analyze the RTT data and identify the common RTT variation patterns between WLAN and wired infrastructure, the active method can serve the purpose with flexibility of facilitating an end-to-end approach without using a dedicated packet capture equipment. We have also verified our active RTT measurement results with the RTTs from the preliminary passive measurements of TCP SYN and SYN/ACK time analysis and confirmed that the results are similar.

True Ping [36] is used for data collection, which is an open source tool that provides us more accurate timestamp information than others. To keep the scope manageable in a three-month study, we chose a well-known web server as the common destination for all data sets. We maintain each data set to keep the same path during the data collection and verify it by using periodic traceroute. On each data collection, we use a script to launch 14 continuous probing series, varying packet size from 100 bytes to 1400 bytes. The result file contains the date, time, and RTT results data. All data collections are over 4 h of duration with 5-s interval for both wired and wireless, except the Scenario WLAN.B in Data Sets I where a microwave oven is turned on to increase signal interference for 10 min with 500-ms

TABLE I
COMPARISON OF PACKET LOSS RATES

Data Sets I			Data Sets II		Data Sets III	
wired	wlan.A	wlan.B	wired	wlan	wired	wlan
0.0%	4.5%	19~90.0%	0.0%	0.5%	0.0%	0.1%

interval. The same machine is used for the entire data set collection. The collected data sets are carefully processed to filter out any results that have a higher or lower hop count in order to minimize the results from alternating paths.

IV. ANALYSIS OF RTT EXPERIMENTAL RESULTS

In this section, we present an extensive analysis of the data sets including packet loss, empirical cumulative density function, mean, standard deviation, and percentiles according to the different packet sizes. We also investigate the impact of time of the day. Table I shows the result of the packet loss test. As commonly expected, WLAN has consistent packet loss on all the test settings. Particularly, the situation is significantly exacerbated in the noisy scenarios, WLAN.B, the packet loss rate is as much as 19 ~ 90%. This indicates wireless channels are very susceptible to noises and the performance can be drastically degraded, *especially depending on the packet time in the air.*

For the rest of the analysis, we use the variable portion of delay after subtracting the minimum delay from the measured RTT values. Due to the space limitation, we only present one or two data set test results for each analysis. Since we observe the common characteristics over the different data sets, the similar result discussion should apply to the other data sets as well, unless mentioned otherwise.

Fig. 1 shows the empirical cumulative density functions of variable delays in comparison between wired and wireless (WLAN.A and WLAN.B in data set I) network delays. All WLAN scenarios have greater disperse than wired scenarios, which implies a higher probability of long delay on the WLAN as discussed in [23]. In Fig. 2, we investigate the impact of packet size on the RTT delay. The mean RTT of WLAN increases linearly according to the increment of the packet size. The overall RTT delay of WLAN is greatly higher than the normal transmission delay on the large packets. It results from the *high probability of interference by noise along the increment of the packet size*, which then causes *packet retransmission in the link layer*. On the other hand, the mean RTT of the wired network remains steady regardless of packet size. There is no significant increment of the transmission delay. As illustrated in Fig. 3, we investigate the variability of RTT in terms of standard deviation. It presents that the wired RTT standard deviation has ranges between 2.6 ms and 3.3 ms and the WLAN RTT standard deviation has ranges between 13.5 ms and 16 ms. WLAN has *at least 10 ms* higher RTT standard deviation than a wired network. The prominent differences in mean and standard deviation between WLANs and wired networks imply that the simple test (as little as 50 to 100 samples) with the large packet size can differentiate WLAN from wired network. This means that adaptation of tools or protocols can utilize such a short online measurement.

Figs. 4 and 5 show various percentiles of RTTs according to the packet sizes. As illustrated in Fig. 4, the wired network

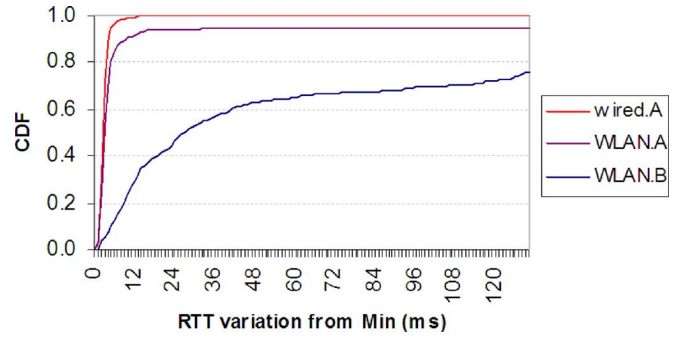


Fig. 1. Empirical CDF of delay variability (Data Sets I, packet size = 500 B).

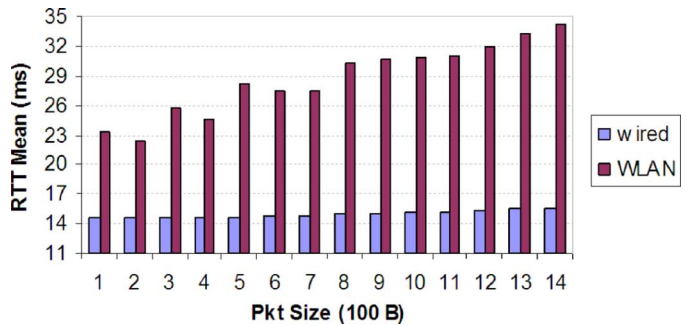


Fig. 2. Mean RTT per packet size (Data Sets III).

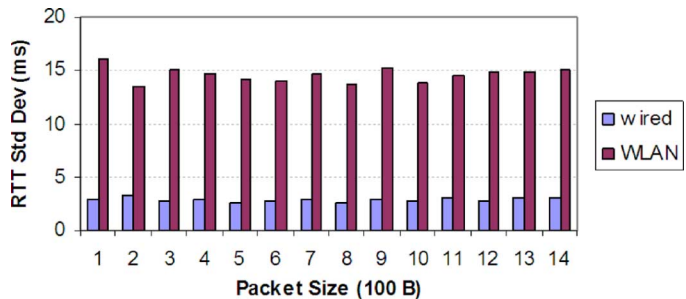


Fig. 3. Standard deviation per packet sizes (Data Sets III).

has less than 1-ms RTT delay changes in percentiles for the different packet sizes and only a 4-ms RTT delay difference between the minimum and 95 percentile. Meanwhile, as shown in Fig. 5, WLAN has about 10-ms RTT delay changes in percentiles for the different packet sizes and about a 40-ms RTT delay difference between the minimum and 95 percentile.

To check the possibility of delay correlations among consecutive test probes, we show the scatter plots of RTTs in Figs. 6 and 7. In both wired and wireless networks, the figures show low correlation coefficient values—close to zero. It is consistent with [37] where point-to-point backbone delay is measured, the duration of congestions is typically very small about a few milliseconds, and thus the measurement probes that are bigger than the time scale can be considered independent. It is noticed that the correlation coefficient value in WLAN is negative. It implies that extremely high RTTs are often followed by small RTTs.

Next, in order to confirm and eliminate the impact of time of the day specifically, we analyze the CDFs of RTT variabilities

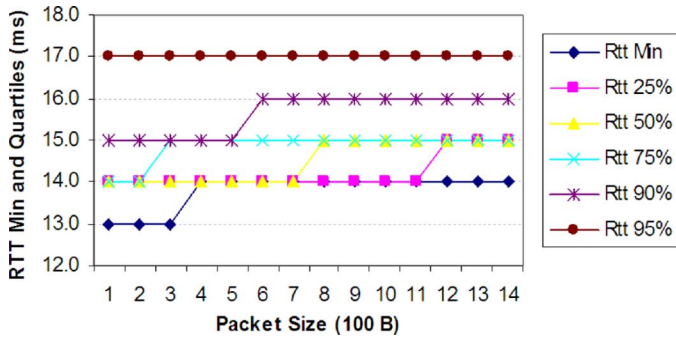


Fig. 4. Percentiles per packet size (Data Sets III-Wired).

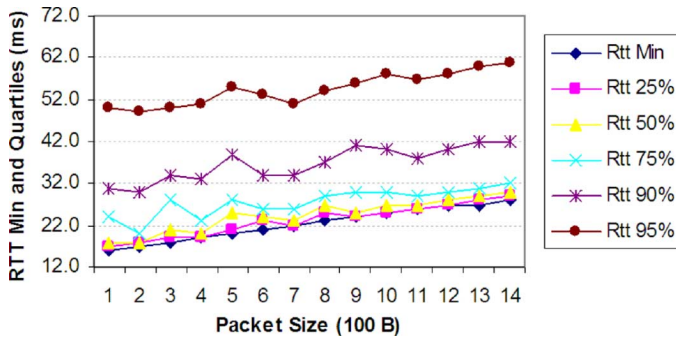


Fig. 5. Percentiles per packet size (Data Sets III-WLAN).

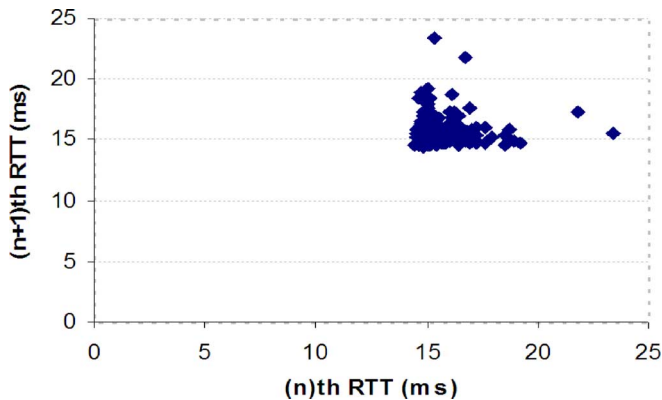


Fig. 6. Scatter plot of wired network (Data Sets III, packet size 500 B). Correlation coefficient: 0.1244.

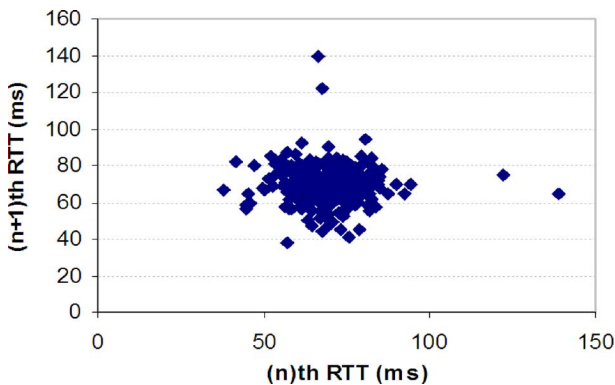


Fig. 7. Scatter plot of WLAN network (Data Sets III, packet size 500 B). Correlation coefficient = -0.0325.

per hour. For the home environments (Data Sets I) and the office environments (Data Sets II), we cannot find any particular

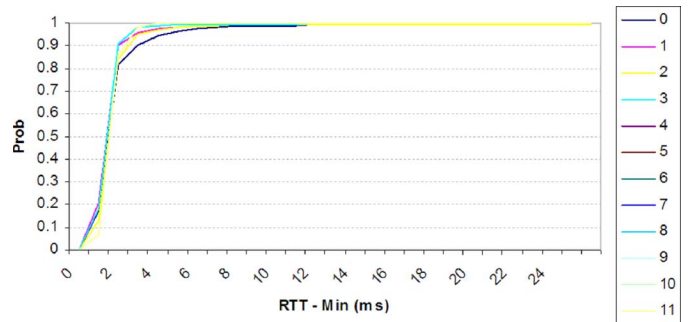


Fig. 8. CDF by hour of the day (Data Sets III-Wired. Packet size = 100 B).

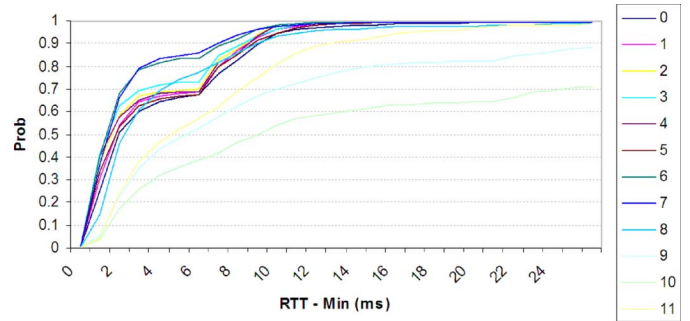


Fig. 9. CDF by hour of the day (Data Sets III-WLAN. Packet size = 100 B).

usage changes over different times on both the WLAN and wired network. However, as shown in Figs. 8 and 9, the campus environment (Data Sets III) presents an impact in WLAN that shows slightly more WLAN usage between 9 and 10 am. In general, the wired network does not experience noticeable performance degradation over time of the day, because of the high provisioning as well as immunity to channel dynamics. In the WLAN, the usage of home environments is negligible to notice the changes and the usage changes of office environments are small because of the high provisioning as well as the high utilization of the wired network. However, the test on the campus WLAN environments implies the dynamic mobility in the morning in the campus over under-provisioned WLAN. Even in such a case, the impact of the time of day in RTT variability is below 4 ms for the 90th percentile. Therefore, we find that using RTT variability is a stable method in differentiating network types regardless of the time of the day.

Our observations on the RTT variability can be summarized as follows.

- 1) The RTT variability (i.e., measured RTT–minimum RTT) exhibits consistent characteristics of network types rather than the absolute value of RTT over different networks.
- 2) The variability of RTTs is more prominent for a large packet size with wireless networks.
- 3) As for testing criteria, standard deviation or min-to-90th percentile (≥ 10 ms) of large packets (1500 B) can be effectively used to find the existence of a wireless network.
- 4) The scheme provides an accurate decision of underlying access network types not sensitive to the correlation of probe times or the time of the day.

As stated earlier, as little as 50 ~ 100 packets are sufficient to make the decision. In all the network environments we tested, the decision was 100% accurate in our experiments. In the next

section, we provide schemes to optimally select the MAC layer retransmission limit and the packet size given a network condition toward a cross-layer approach to multimedia delivery.

V. CROSS-LAYER DESIGN FOR ADAPTIVE VIDEO DELIVERY

Once the receiver is identified as accessing multimedia streams from a wireless network, we, in this section, discuss adaptive multimedia transfer schemes particularly addressing the issue of the retransmissions and packet size on error-prone links. We conduct the analysis of the interplay of adaptive parameters and show the performance tradeoffs. It provides insights on how the proposed adaptive schemes can be optimally used.

A. Adaptive Video Delivery Schemes

Toward the adaptive video delivery, we propose the following building blocks for channel and application adaptive strategies.

1) *Adaptive MAC Layer Retransmission Limiting*: We have shown excessively variable delays in the wireless networks that are particularly caused by MAC layer retransmission. However, video streaming applications are delay sensitive with stringent timing constraints. As packets delivered late are of no use, the number of retransmissions should be limited, and handing over erroneous packets to the application layer would be more effective. UDP-Lite [38] is available for such a purpose. Unlike UDP, where either all or none of a packet is protected by a checksum, UDP-Lite allows for partial checksums that only cover part of a datagram, and will therefore deliver packets that have been partially corrupted. Support for UDP-Lite is added in the Linux kernel version 2.6.20. It is particularly useful for multimedia protocols, where receiving a packet with a partly damaged payload is better than receiving no packet at all. Application layer error correction and concealment techniques can mitigate the effect. In the next subsection, we discuss the issue of limiting MAC layer retransmissions theoretically in detail.

2) *Adaptive Application Layer FEC*: Erroneous packets sent to the application layer can be recovered by the application layer FEC. Block-based Reed–Solomon (RS) codes [39] can be employed for this purpose. With RS coding, a series of k packets are sent with additional $(m - k)$ parity packets. The RS decoder at the application layer can correct up to $(m - k)$ packet losses from the 802.11 MAC layer out of m RS coded packets. Each byte of parity packets corresponds to the same i th byte of k video packets. Then the m packets are buffered at the receiver for decoding and error correction.

Note that FEC causes a constant overhead regardless of packet error/loss, while ARQ overhead only occurs in case of error. Thus, in general, a good channel condition encourages ARQ rather than FEC, and a bad channel condition would prefer FEC. However, too poor a channel condition would not be able to recover errors with FEC. Therefore, MAC layer ARQ with limited retransmissions should be chosen in consideration of the channel condition as well as the delay constraints of the application.

3) *Adaptive Packet Size Decision*: We showed earlier that a packet size is tightly related to packet delay (or RTT) and channel condition. The packet size of video packets should be

chosen adaptively to allow the maximal goodput (the amount of useful data over time unit), given the channel condition (or channel error rate), delay constraint of the application (i.e., retransmission threshold), and the application layer FEC. The theoretical analysis and the results are discussed in the next subsection.

B. Analysis of Adaptive Cross-Layer Strategies

Now we show the analysis of the above strategies and provide the tradeoffs of the related parameters and performance.

First, we consider the channel condition with a channel error model. Suppose p_{ber} is a bit error rate and assume random errors, then a packet error rate of length l byte, $p_{per}(l)$, is computed as follows:

$$p_{per}(l) = 1 - (1 - p_{ber})^{8l}. \quad (1)$$

$p_{succ}^{data,ack}(l)$, the probability of a successful transmission of a packet size l with an acknowledgement packet size a , is obtained as shown in the following:

$$p_{succ}^{data,ack}(l) = (1 - p_{per}(l))(1 - p_{per}(a)). \quad (2)$$

We assume RTS and CTS packets are always successful as their packet sizes are very small and the signals are sent with a high priority.

Next, $p_{succ}^{mac}(i|l)$, the probability that a packet with size l is successfully transmitted in the MAC layer after i retransmissions, is given as follows:

$$p_{succ}^{mac}(i|l) = (1 - p_{succ}^{data,ack}(l))^i p_{succ}^{data,ack}(l). \quad (3)$$

Then the probability that a packet successfully transmitted within r retransmissions in the MAC layer is obtained as follows:

$$p_{succ}^{mac}(l, r) = 1 - (1 - p_{succ}^{data,ack}(l))^{r+1}. \quad (4)$$

Equation (4) enables us to compute the expected delay of a packet transmission:

$$E(d(l, r)) = \sum_{i=0}^r p_{succ}^{mac}(i|l) [(i * d_{rto} + d_{rtt}(l))] + (r + 1)(1 - p_{succ}^{mac}(l, r))d_{rto} \quad (5)$$

where d_{rto} and d_{rtt} are a retransmission timeout and a round-trip time of a packet, respectively. The first term is for a successful transmission within r attempts, and the latter is for an unsuccessful transmission after r attempts.

The above expected delay given by (5) gives a tradeoff of delay and the number of retransmissions. Thus, given a delay constraint of an application and a channel condition, we can compute the limit of the number of retransmissions in the MAC layer.

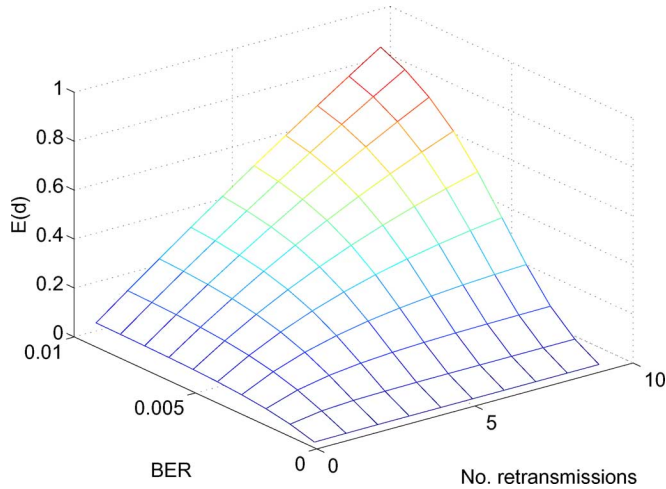


Fig. 10. Expected delay (s) with varying channel conditions and number of retransmissions.

Fig. 10 illustrates the expected delay with varying channel conditions and number of retransmissions. As expected, it shows that higher channel error and retransmissions increase the delay.

Next, let us consider the performance of an application layer error correction. Using (m, k) RS code at the application layer, the packets can be recovered with $(m - k)$ or less number of packet errors out of m packets. Thus, the probability of a successful transfer of m packets including parities after RS decoding is

$$p_{succ}^{app}(l, r, m, k) = \sum_{i=0}^{m-k} \binom{m}{i} (1 - p_{succ}^{mac}(l, r))^i p_{succ}^{mac}(l, r)^{m-i}. \quad (6)$$

When there are less than k , say $m - i (< k)$ correctly received packets, including both video and parity packets possibly, those will not be fully recovered into k video packets with RS coding. Among them, video packets can be still used however, and there will be $(m - i)(k/m)$ video packets out of $(m - i)$ packets on average. Therefore, the goodput with application layer error correction, $G(l, r, m, k)$, can be calculated by the fraction of the average number of recovered or correctly transmitted video packets over the expected time to transfer m packets:

$$G(l, r, m, k) = \frac{8lk p_{succ}^{app}(l, r, m, k)}{m E(d(l, r))} \cdot \frac{k}{m} \frac{\sum_{i=m-k+1}^m (m-i) \binom{m}{i} (1 - p_{succ}^{mac}(l, r))^i p_{succ}^{mac}(l, r)^{m-i}}{m E(d(l, r))}. \quad (7)$$

Now the optimal packet size l^* which gives the most goodput can be obtained as follows:

$$l^* = \arg_l \max G(l, r, m, k). \quad (8)$$

Fig. 11 shows the goodput with varying packet sizes and channel conditions. In general, a lower bit error rate channel

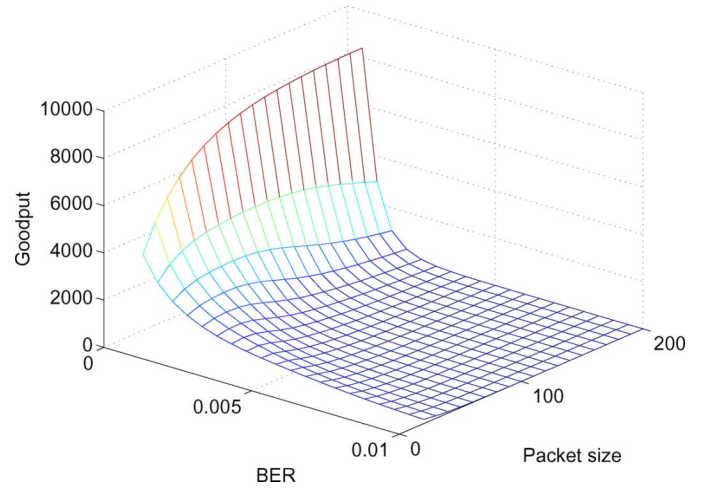


Fig. 11. Goodput (bps) with varying packet size (B) and channel condition (BER) (no. retransmissions = 3).

TABLE II
OPTIMAL GOODPUT, FEC LEVEL, RETRANSMISSION LIMIT,
AND PACKET SIZE ON VARIOUS CHANNEL CONDITIONS

BER	Goodput	FEC level	Rtr. limit	Pkt size
0.0001	34131	0	0	200
0.0011	4133	1	1	70
0.0051	679	1	2	20
0.0096	221	5	7	20

gives better goodput. With a large packet size, however, the goodput decreases drastically as the channel condition gets worse.

We evaluate the performance of goodput by varying the optimization parameters from (7). For illustration purpose, since there are more than two independent variables involved, we first show the goodput by varying two parameters while setting the others as constants. For a concise presentation, we only depict the prominent results rather than showing the performance for many combinations of parameters. The optimal parameters and goodputs for various channel conditions are summarized in Table II.

First, we vary FEC levels and the maximum number of retransmissions for good and bad channel conditions, and small and large packet sizes. Figs. 12–15 show the goodput with two different levels of BER of 0.0001 and 0.0096 and packet sizes of 20 and 200 bytes. We observe that even though the impact of FEC levels is not as significant as the MAC layer retransmission limit in general, the FEC level can effectively contribute to the goodput, particularly when the channel condition is bad and the packet size is small.

We further investigate the performance of the goodput with varying the number of retransmissions and the packet sizes. Figs. 16–19 depict the goodput with BER set to be 0.0001, 0.0011, 0.0051, and 0.0096, respectively. The optimal values corresponding to Figs. 16–19 are summarized in Table II. Notice that when the channel condition is good, bigger packet size and small number of retransmissions lead to the most goodput (see Fig. 16). Meanwhile, when the channel condition is bad, the goodput becomes lower in general, and a small packet size and a large number of MAC layer retransmissions are better off

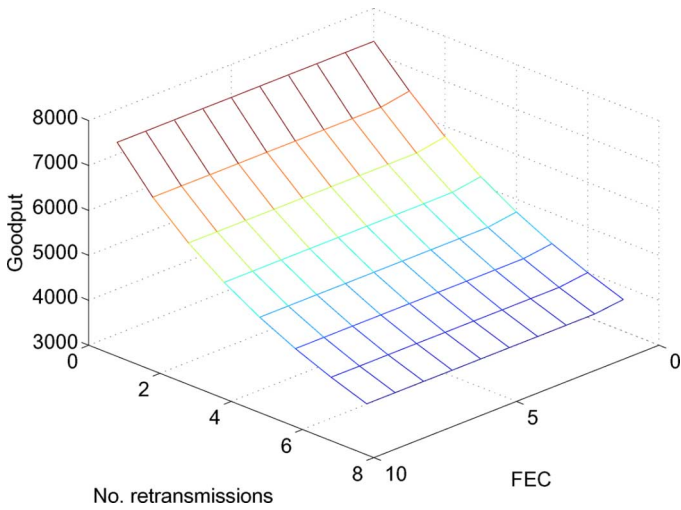


Fig. 12. Goodput (bps) with varying application layer FEC levels and MAC layer retransmission limits (BER = 0.0001, B = 20).

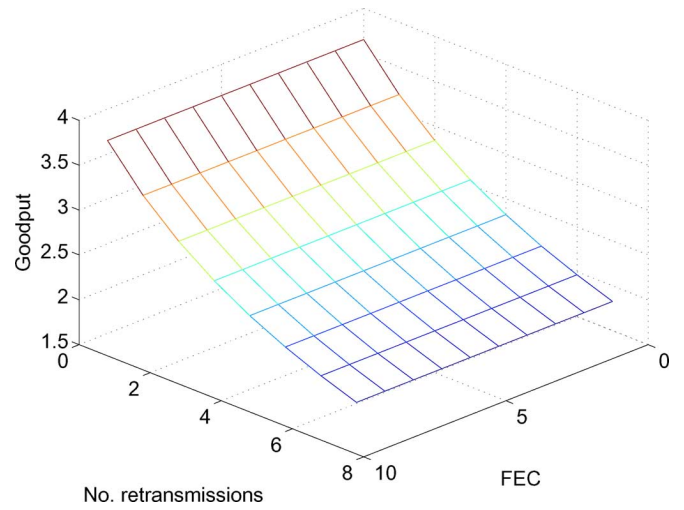


Fig. 15. Goodput (bps) with varying application layer FEC levels and MAC layer retransmission limits (BER = 0.0096, B = 200).

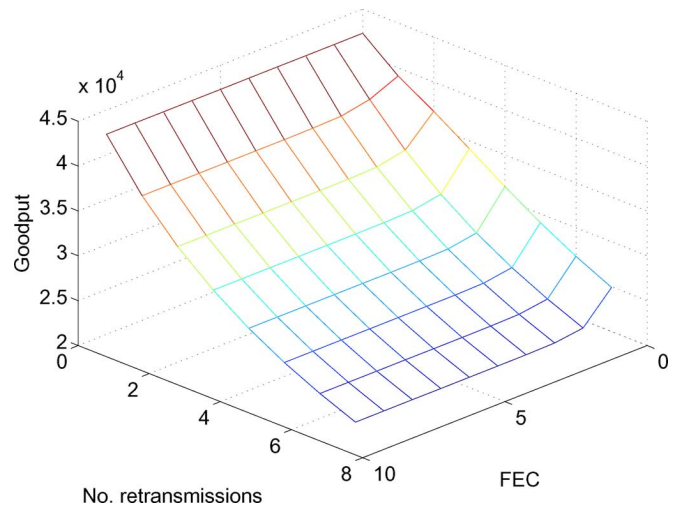


Fig. 13. Goodput (bps) with varying application layer FEC levels and MAC layer retransmission limits (BER = 0.0001, B = 200).

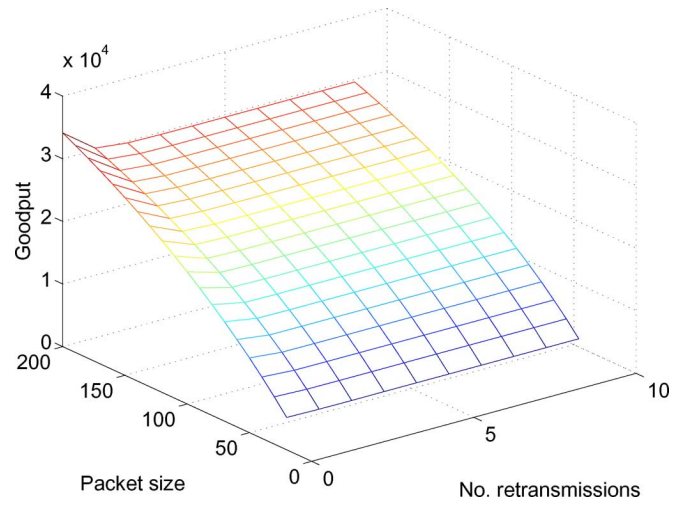


Fig. 16. Goodput (bps) with varying packet sizes (B) and MAC layer retransmission limits (BER = 0.0001).

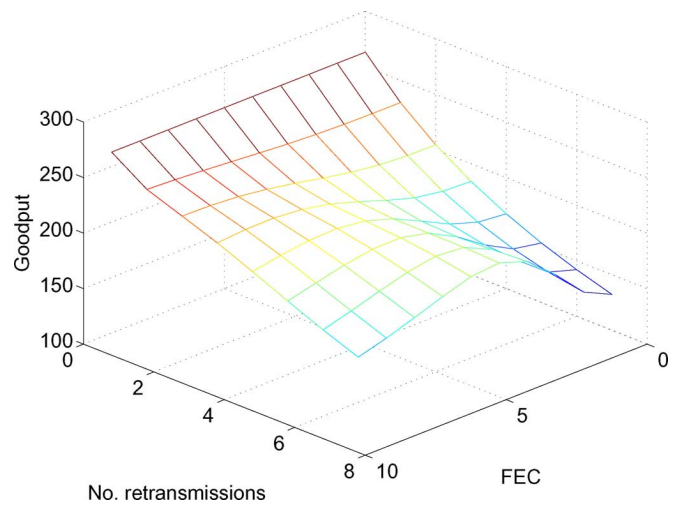


Fig. 14. Goodput (bps) with varying application layer FEC levels and MAC layer retransmission limits (BER = 0.0096, B = 20).

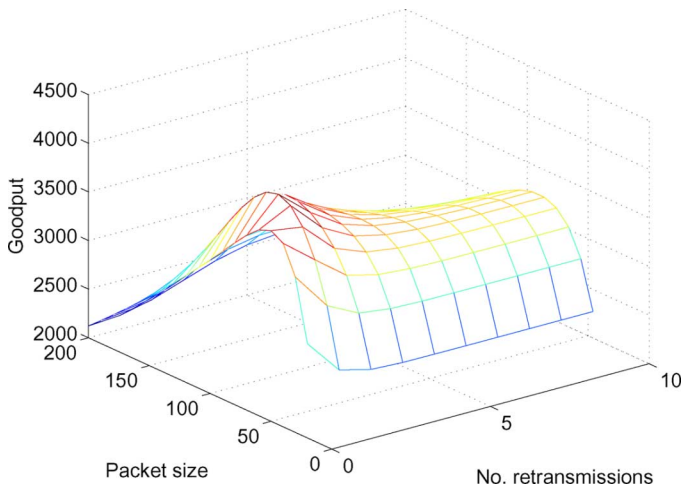


Fig. 17. Goodput (bps) with varying packet sizes (B) and MAC layer retransmission limits (BER = 0.0011).

(see Fig. 19). It is because too many erroneous packets cannot be recovered in the application layer. As illustrated in Figs. 17

and 18, in mild to medium channel error conditions, the optimal packet size and number of retransmissions that produce the maximum goodput should be chosen carefully, which can be found by (8). The figures reveal interesting tradeoffs among

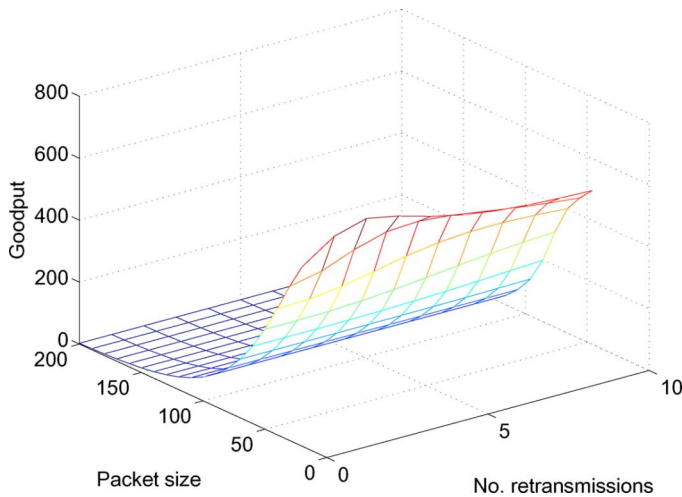


Fig. 18. Goodput (bps) with varying packet sizes (B) and MAC layer retransmission limits (BER = 0.0051).

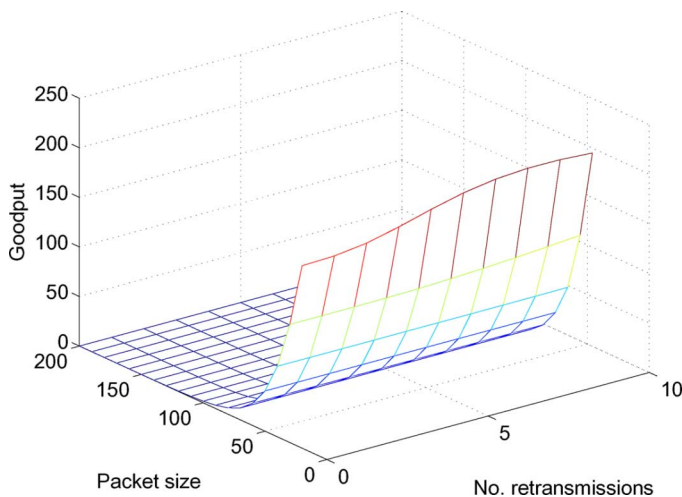


Fig. 19. Goodput (bps) with varying packet sizes (B) and MAC layer retransmission limits (BER = 0.0096).

packet sizes, the number of retransmissions, and channel conditions in regards to the goodput.

VI. CONCLUSIONS

We have proposed a holistic approach for a cross-layer solution over heterogeneous network environments. We have shown that the variability of RTT can be effectively used for an automatic online distinction between wireless and wired networks. We presented a quantitative analysis on RTT variability in diverse wireless and wired environments. We have used various metrics including mean, standard deviation, and various percentiles of variabilities for the comparison. The experimental results show that the variable portion of RTTs on a wireless network has a significantly larger standard deviation and min-to-high percentiles due to the MAC layer retransmissions, and are significantly impacted by packet size than those of a wired network, regardless of the specific vendor technologies. We have shown that the characteristics can be identified even with very

small number of samples. Thus, the results lead to easy and accurate classification of a wireless network with a simple set of testing. Based on the observation of RTT variability, we then proposed effective adaptive cross-layer schemes for multimedia delivery over the time varying link characteristics, including the MAC layer retransmission limit, the application layer FEC level control, and selecting a packet size. We have analyzed the performance tradeoffs to find how the proposed adaptive schemes can be optimally utilized.

We believe that our holistic approach would ease a wider adoption of cross-layer schemes over networks with diverse access network technologies, and adapt important system parameters to varying network environment in order to maximize the achievable multimedia performance. In the future, we plan to investigate other application layer mitigation techniques and the efficacy, addressing jitter performance in addition to goodput and delay.

REFERENCES

- [1] S. Low, L. Peterson, and L. Wang, *Understanding TCP Vegas: Theory and practice*, 2000. [Online]. Available: <http://www.ee.mu.oz.au/staff/slow/research>.
- [2] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Reno performance: A simple model and its empirical validation," *IEEE/ACM Trans. Netw.*, vol. 8, no. 2, pp. 133–145, Apr. 2000.
- [3] S. Floyd, M. Handley, and J. Padhye, "Equation-based congestion control for unicast applications," in *Proc. ACM SIGCOMM*, 2000.
- [4] M. Handley, S. Floyd, J. Padhye, and J. Widmer, *TCP Friendly Rate Control (TFRC): Protocol Specification*, 2003. [Online]. Available: <ftp://ftp.rfc-editor.org/in-notes/rfc3448.txt>.
- [5] Y. Tian, K. Xu, and N. Ansari, "TCP in wireless environments: Problems and solutions," *IEEE (Radio) Commun. Mag.*, vol. 43, no. 3, pp. S27–S32, Mar. 2005.
- [6] C. Casetti, M. Gerla, S. Lee, M. Sanadidi, and R. Wang, "Enhancing TCP Congestion Control Via Connection Bandwidth Estimation, A Performance Study," UCLA CSD, 2002, Tech. Rep. 020024.
- [7] W. Zeng, H. Zhang, J. Liu, N. Cackov, S. Vujicic, B. Vujicic, V. Vukadinovic, and L. Trajkovic, "Improvement of TCP over wireless links," in *Proc. ASI Exchange*, Vancouver, BC, Canada, Mar. 2003.
- [8] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, "TCP westwood: Bandwidth estimation for enhanced transport over wireless links," in *Proc. 7th Annu. Int. Conf. Mobile Computing and Networking (MobiCom '01)*, New York, Jul. 2001, pp. 287–297.
- [9] K. Xu, Y. Tian, and N. Ansari, "TCP-Jersey for Wireless IP Communications," *IEEE J. Select. Areas Commun.*, vol. 22, no. 4, pp. 747–756, May 2004.
- [10] B. Zhou, C. P. Fu, and V. O. K. Li, "TFRC Ven0: An enhancement of TCP friendly rate control over wired/wireless networks," in *Proc. IEEE Int. Conf. Network Protocols (ICNP)*, 2007.
- [11] R. Chaudhary and L. Jacob, "ECN based TCP-friendly rate control for wireless multimedia streaming," in *Proc. IEEE ICCCN*, 2003.
- [12] L. Huang, U. Horn, and F. Hartung, "Proxy-based TCP-friendly streaming over mobile networks," in *Proc. ACM WoWMoM*, 2002.
- [13] J. Y. Pyun, Y. Kim, and K. H. Jang, "Wireless measurement based resource allocation for QoS provisioning over IEEE 802.11 wireless LAN," *IEEE Trans. Consum. Electron.*, vol. 49, no. 3, pp. 614–620, Aug. 2003.
- [14] *Iperf*. [Online]. Available: <http://sourceforge.net/projects/iperf/>.
- [15] *netperf*. [Online]. Available: <http://www.netperf.org/netperf/>.
- [16] *Test tcp (ttcp)*. [Online]. Available: <http://www.pcausa.com/Utilities/pcattcp.htm>.
- [17] C. D. Mano, A. Blaich, Q. Liao, Y. Jiang, D. A. Cieslak, D. C. Salyers, and A. Striegel, "RIPPS: Rogue identifying packet payload slicer detecting unauthorized wireless hosts through network traffic conditioning," *ACM Trans. Inf. Syst. Secur. (TISSEC)*, vol. 11, no. 2, pp. 1–23, 2008.
- [18] W. Wei, B. Wang, C. Zhang, J. Kurose, and D. Towsley, "Classification of access network types: Ethernet, Wireless LAN, ADSL, cable modem or dialup?," in *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005.

- [19] J. Gretarsson, "Performance analysis of the intertwined effects between network layers for 802.11g transmissions," in *Proc. ACM Workshop Wireless Multimedia Networking*, 2005, pp. 123–130.
- [20] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11b," in *Proc. IEEE INFOCOM*, San Francisco, CA, 2003.
- [21] J. Yeo, M. Youssef, and A. Agrawala, "A framework for Wireless LAN monitoring and its applications," in *Proc. WiSE*, Oct. 2004.
- [22] F. Chinchilla, M. Lindsey, and M. Papadopouli, "Analysis of wireless information locality and association patterns in a campus," in *Proc. IEEE INFOCOM*, Hong Kong, Mar. 2004.
- [23] Y.-C. Cheng, M. Afanasyev, P. Verkaik, P. Benko, J. Chiang, A. C. Snoeren, S. Savage, and G. M. Voelker, "Automating cross-layer diagnosis of enterprise wireless networks," in *Proc. ACM SIGCOMM*, Aug. 2007.
- [24] W. Wei, K. Suh, B. Wang, Y. Gu, J. Kurose, and D. Towsley, "Passive online rogue access point detection using sequential hypothesis testing with TCP ACK-pairs," in *Proc. 7th ACM SIGCOMM Conf. Internet Measurement (IMC '07)*, New York, Oct. 2007, pp. 365–378.
- [25] W. Wei, S. Jaiswal, J. Kurose, and D. Towsley, "Identifying 802.11 traffic from passive measurements using iterative Bayesian inference," in *Proc. IEEE INFOCOM*, Barcelona, Spain, Apr. 2006.
- [26] B. Veal, K. Li, and D. Lowenthal, "New methods for passive estimation of TCP round-trip times," in *Proc. Passive and Active Measurement Conf.*, 2005.
- [27] J. Aikat, J. Kaur, F. D. Smith, and K. Jeffay, "Variability in TCP roundtrip times," in *Proc. ACM Internet Measurement Conf.*, Oct. 2003, pp. 279–284.
- [28] P. Sessini and A. Mahanti, *Observations on Round-Trip Times of TCP Connections*, ZD Net, 2006.
- [29] Q. Li and M. van der Schaar, "Providing adaptive QoS to layered video over wireless local area networks through real-time retry limit adaptation," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 278–290, Apr. 2004.
- [30] H. Zheng and J. Boyce, "Packet coding schemes for MPEG video over Internet and wireless networks," in *Proc. IEEE Wireless Communications and Networking Conf. (WCNC)*, Chicago, IL, Sep. 2000, pp. 191–195.
- [31] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communications: A review," *Proc. IEEE*, vol. 86, no. 5, pp. 974–997, May 1998.
- [32] H. Ma and M. E. Zarki, "MPEG-2 video transmission over wireless access networks using type-I hybrid ARQ schemes for VoD services," in *Proc. Packet Video Workshop*, Apr. 1999.
- [33] M. Zorzi, "Some results on error control for burst-error channels under delay constraints," *IEEE Trans. Veh. Technol.*, vol. 50, no. 3, pp. 12–24, May 2001.
- [34] W. Yuan, K. Nahrstedt, S. Adve, D. L. Jones, and R. H. Kravets, "Design and evaluation of a cross-layer adaptation framework for mobile multimedia systems," in *Proc. SPIE/ACM Multimedia Computing and Networking Conf. (MMCN)*, 2003.
- [35] I. C. Systems, *Linksys 802.11g AP Model WRT54G Description and Firmware Versions History*, 2008.
- [36] A. Planet, True Ping, An Open Source Program for Network Administrators, 2006. [Online]. Available: <http://www.andrea-planet.com/tping/>.
- [37] B.-Y. Choi, S. Moon, R. Cruz, Z.-L. Zhang, and C. Diot, "Quantile sampling for practical delay monitoring in Internet backbone networks," *Elsevier J. Comput. Netw.*, vol. 51, no. 10, pp. 2701–2716, Jul. 2007.
- [38] L.-A. Larzon, M. Degermark, S. Pink, L.-E. Jonsson, and G. Fairhurst, The Lightweight User Datagram Protocol (UDP-Lite), 2004. [Online]. Available: <http://tools.ietf.org/html/rfc3828/>.
- [39] S. B. Wicker, Biargava, and V. K. Bhargava, *Reed-Solomon Codes and Their Applications*. New York: Wiley, 1999.



Baek-Young Choi (M'02) received the B.S. degree from Pusan National University, Pusan, Korea, in 1993, the M.S. degree from Pohang University of Science and Technology, Pohang, Korea, in 1995, and the Ph.D. degree from the University of Minnesota, Twin Cities, in 2003, all in computer science and engineering.

She held positions at Sprint Advanced Technology Labs and the University of Minnesota, Duluth, as a Postdoctoral Researcher and 3M McKnight Distinguished Visiting Assistant Professor, respectively.

She has been an Assistant Professor at the University of Missouri, Kansas City, since 2005. Her research interests include network measurement and resource management of diverse network types.



Sejun Song received the B.S. degree in computer science from Pusan National University, Pusan, Korea, and the M.S. and Ph.D. degrees in computer science and engineering from the University of Minnesota, Twin Cities, in 1999 and 2001, respectively.

He is an Assistant Professor in the Department of Engineering Technology and Industrial Distribution at Texas A&M University, College Station. Prior to joining academia, he had several years of industry experience from Cisco Systems, Honeywell Research Lab, and Positive Networks. His research interests lie

in broad areas of networked systems including measurement, high availability, data storage, real-time, and embedded systems.



Yue Wang received the B.S. degree in computer science and mathematics from Ottawa University, Ottawa, KS, in 1990 and the M.S. degree in computer science from the University of Missouri, Kansas City, in 2008.

He is an information systems consultant in Beijing, China. Prior to that, he held positions in the area of network, security, and systems integration as Enterprise Security Manager, Chief Information Security Officer, and Senior Network Consultant at Lab One, K&Y, and DST systems, respectively.



Eun Kyo Park received the Ph.D. degree in computer science from the Northwestern University, Chicago, IL.

He is a Professor of computer science at the University of Missouri, Kansas City. His research interests include computer communications and networks, distributed systems, data mining and bioinformatics, optimizations, information and knowledge management, software architectures, and object-oriented methodology. Currently, he is on an assignment serving as a Program Director, Division

of Computing and Communications Foundations, at the U.S. National Science Foundation.