TCP PERFORMANCE IMPROVEMENT OVER WIRELESS ATM NETWORKS THROUGH A NEW AAL PROTOCOL

Ian F. Akyildiz and Inwhee Joe

Broadband and Wireless Networking Lab.
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332
E-Mail: ian@ee.gatech.edu; inwhee@ee.gatech.edu

ABSTRACT

This paper describes the design and simulation of a new AAL protocol (AAL-X) for improving TCP performance over wireless ATM networks. The wireless links are characterized by higher error rates and burstier error patterns in comparison with the fiber-based links for which ATM was designed. Since the poor performance of TCP over wireless ATM networks is mainly due to the fact that TCP always responds to all packet losses by congestion control, the key idea in the design is to push the error control portion of TCP to the AAL layer so that TCP is only responsible for congestion control. The AAL-X is based on a novel and reliable ARQ mechanism to support quality-critical TCP traffic over wireless ATM networks. The proposed AAL protocol has been validated using the OPNET tool on the simulated wireless ATM network. The simulation results show that the AAL-X provides higher throughput for TCP over wireless ATM networks compared to the existing approach of TCP with AAL 5.

1. INTRODUCTION

Modern TCP implementations contain several new algorithms to improve TCP performance on packet loss. For example, TCP-Tahoe employs the *fast retransmit* algorithm [5] and TCP-Reno adds the *fast recovery* algorithm [6], while TCP-Vegas attempts to provide earlier detection of packet loss and accurate round-trip delay estimation [3]. However, all of these implementations are optimized for the case when a single packet is dropped from a single window. If multiple packets are dropped from a single window, their performance suffers severely, because the sender is forced to recover by means of a retransmission timeout instead of fast recovery. For TCP over ATM networks, cell losses in ATM switches may span TCP packet boundaries, resulting in the loss of two TCP packets. Moreover, when ATM networks include wireless links, multiple packet losses can happen due to higher and burstier error patterns of wireless links.

Recently, SACK and New-Reno have been proposed to improve TCP performance when multiple packet losses occur within a single window [7, 4]. However, TCP does not have to shrink its congestion window at all in response to the packet losses due to link errors or handoffs, because such losses have nothing to do with network congestion. Obviously, this unnecessary window shrinking causes significant performance degradation. The fundamental solution to this problem is to distinguish between link errors and network congestion. Therefore, our approach is to push the error control portion of TCP down to the ATM Adaptation Layer (AAL) so that TCP is only responsible for congestion control. As a result, TCP does not invoke congestion control mechanism under any circumstances except for real network congestion.

In this paper, we propose a new AAL protocol, *AAL-X*, to improve TCP performance over wireless ATM networks. To support quality-critical TCP traffic over wireless ATM networks, the AAL-X protocol is based on a new and very efficient ARQ (Automatic Repeat Request) scheme, which does not have timers and retransmits when the packet is indeed lost or in error [1]. In the next section, we describe a detailed design of AAL-X, while in Section 3 we present our simulation models and performance evaluation results from OPNET simulation. Finally, we conclude the paper by highlighting our contribution.

2. THE AAL-X DESIGN

Currently, AAL 5 is used for TCP over ATM networks. It is well-known that TCP introduces significant performance degradation over wireless ATM networks. Therefore, we need to use more appropriate AAL protocol rather than AAL 5 to solve the problems with TCP over wireless ATM networks. We propose the AAL-X to support quality critical traffic (e.g., data, imagery) over high bit-error-rate (BER) wireless links by using an ARQ-based retransmission scheme. Unlike the existing AAL protocols, the AAL-X consists of three sublayers: Common Part Convergence Sublayer (CPCS), Common Part ARQ (CP-ARQ) sublayer, and Segmentation and Reassembly Sublayer (SAR). Since the CP-ARQ sublayer takes care of end-to-end error control, the CPCS sublayer can pass error-free data up to the TCP layer.

Since the retransmission timer is not used in TCP over AAL-X, instead TCP needs another mechanism to invoke the congestion control method during the congestion pe-

0-7803-4984-9/98/$10.00

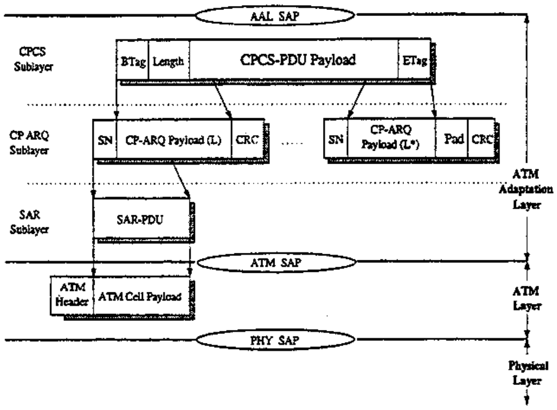

Figure 1: Packet Formats of the AAL-X

riod. The ATM network management can give a congestion notification to the TCP layer, because it can discover network congestion by monitoring RM (Resource Management) cells.

2.1. The Common Part Convergence Sublayer (CPCS)

The CPCS-PDU (Protocol Data Unit) format is shown in Figure 1. The CPCS-PDU includes a header of three octets which consists of two fields to denote the beginning of the CPCS-PDU by *BTag* (1 byte) and the number of octets of the CPCS-PDU by *Length* (2 bytes). The length field is used as the final checking means to ensure that the CPCS sublayer passes error-free and orderly data up to the TCP layer. The CPCS-PDU also includes a trailer to indicate the end of the CPCS-PDU by the *ETag* field (1 byte).

When the CPCS sublayer receives the last CP-ARQ PDU, it tells the CP-ARQ sublayer to send the control packet immediately without waiting for the next period of control packet transmission. This allows a quick response when a small amount of data is being sent.

2.2. The Common Part ARQ (CP-ARQ) Sublayer

The CP-ARQ sublayer provides a reliable end-to-end transmission capability for wireless ATM networks. The ARQ scheme will deal with all kinds of errors by using the retransmission. There are two types of errors to be considered here: link errors induced by the wireless channel and cell losses due to handoff or network congestion resulting from the statistical multiplexing of ATM.

The CP-ARQ PDU format is shown in Figure 1. The *Sequence Number* (SN) field is used to deliver the CP-ARQ PDUs in order. If a CP-ARQ PDU is lost or in error, the CP-ARQ layer will attempt to retransmit the PDU with the same SN in the buffer. The size of SN field is 2 octets, which is large enough to support the window size even for satellite networks. The Cyclic Redundancy Check (CRC-16) is used for error detection.

Except for the last CP-ARQ PDU, the payload length of the CP-ARQ PDU is denoted by L, where L is a step-function of the end-to-end path conditions, i.e., the end-

BTag	MSN	LI	BMAP	Pad	ETag	CRC
1 byte	2 bytes	2 bytes	variable		1 byte	2 bytes

Figure 2: Control Packet Format

to-end path BER calculated at the CP-ARQ sublayer from the error statistics of the most recently received packets by comparing error packets and retransmitted correct packets. The end-to-end path condition is dominated by the worst quality link in the end-to-end path such as wireless links. For the last CP-ARQ packet, the payload length is denoted by L^* which covers the rest of the CPCS-PDU. The *Pad* field ensures that the last CP-ARQ PDU is also a multiple of SAR-PDU (48 bytes). The Pad field is 0-43 octets which can be found by considering the length of header and trailer of the CP-ARQ PDU.

The payload length of the CP-ARQ PDU, L, is dynamically updated according to the look-up table which gives the payload length as a function of the end-to-end path BER, where $L = 48 * i - 4$ for $i = 1, 2, 3, \ldots$ and 4 bytes represent the sum of header and trailer of the CP-ARQ PDU. Initially, $i = 1$ will be used for the worst case, and the transmitter updates the payload length L every time it receives the control packets from the receiver. Based on the new payload length, the receiver updates the generation rate of control packets. The transmitter does not take any action for the new control packet generation rate. This well-defined procedure can easily solve the efficiency problem during the period when the conditions of end-to-end path become worse.

The Survivable ARQ (S-ARQ) is used at the CP-ARQ sublayer, which offers moderate complexity with reasonable efficiency without dealing with timers [1]. The S-ARQ will be briefly explained below.

A. The Survivable ARQ (S-ARQ):

Unlike classical ARQ schemes, the Survivable ARQ scheme does not have any retransmission timer in order to prevent redundant retransmissions and longer recovery periods from a timeout mechanism. The key ideas of S-ARQ are "variable packet size" and "periodic status message". The packet size varies with the end-to-end path conditions so as to maximize the throughput efficiency. The receiver sends its status to the transmitter on a periodic basis by control packets, thereby eliminating the timeout mechanism.

The control packet structure is shown in Figure 2. The control packet consists of seven fields: $BTag$ and $ETag$ have the same bit patterns just as in the CPCS-PDU, MSN is the maximum sequence number of the CP-ARQ PDU below which every PDU has been received correctly, $BMAP$ is a bit map indicating outstanding CP-ARQ PDUs between MSN and the last received PDU, LI is the length indicator for $BMAP$ in bits, *Pad* makes the control packet a multiple of SAR-PDU (48 bytes), and CRC-16 is used for error detection in the control packet.

Another feature of S-ARQ is to provide negative and positive feedbacks by utilizing the $BMAP$. Upon receipt of a control packet, the transmitter marks all CP-ARQ PDUs preceding MSN as successfully transmitted. The PDUs which are set to one in the $BMAP$ are also marked as successfully transmitted; this is the positive acknowledgment aspect of S-ARQ. For the PDUs which are set to zero in the $BMAP$ and have been transmitted, the m value is decreased by 1, where m indicates the frequency of control packets per round trip delay. If this m value goes to zero, the PDU is retransmitted, because sufficient time (round trip delay) has elapsed; this is the negative acknowledgment aspect of the protocol. An estimate of the round trip delay can be made as a smoothed moving average of round trip delay, srt, given by [8],

$$srt_i = \alpha \cdot srt_{i-1} + (1-\alpha) \cdot rt_i \qquad (1)$$

where α is a smoothing constant ranging from 0 to 1 (e.g., usually 0.9 in case of TCP) and rt_i is the round trip delay measured for the $i-th$ CP-ARQ packet.

In addition, S-ARQ provides multiple and selective acknowledgments by turning on bits in the $BMAP$ for packets received correctly at the receiver. Since TCP can acknowledge packets received only in order, TCP may experience poor performance when multiple packets are lost from one window of data. In fact, TCP takes one round trip time to find out about each lost packet, while S-ARQ takes care of multiple packet losses in one round trip time by using the $BMAP$.

The performance of ARQ schemes is very sensitive to the packet size. It is apparent that if the packet size is too small, the protocol is operating inefficiently due to the overhead required per packet. On the contrary, if the packet size is made too large, the packet is more likely to be received in error, resulting in more retransmissions and further reduced throughput. Therefore, there exists an optimal packet size in the sense of maximizing the throughput efficiency of the protocol. The optimal choice is found to depend on the error statistics of the channel and on the number of overhead bits used for control. Since our protocol retransmits only packets in error based on the SR-ARQ scheme, the optimal information payload length is given by [9],

$$L_{opt} = \frac{-h\ln(1-P_b) - \sqrt{-4h\ln(1-P_b) + h^2\ln(1-P_b)^2}}{2\ln(1-P_b)} \qquad (2)$$

where P_b is the end-to-end path BER and h is the number of overhead bits per CP-ARQ packet. In S-ARQ, the payload length is updated dynamically according to the end-to-end path conditions in order to maximize the throughput efficiency. Since the overhead size h is fixed to 4 bytes in the CP-ARQ packet, the optimal payload length L_{opt} is determined entirely based on the end-to-end path BER at that time. Then, the actual payload length of the CP-ARQ packet in Figure 1, L, is chosen as the closest value to L_{opt}, making the CP-ARQ PDU a multiple of SAR-PDU (48 bytes).

2.3. The Segmentation and Reassembly (SAR) Sublayer

The SAR sublayer function is adopted from AAL 5. The SAR-PDU consists simply of 48 octets of payload, carrying a portion of the CP-ARQ PDU. The ATM-user-to-ATM-user (AAU) bit, last bit in the payload-type field (3 bits) of the ATM cell header, is used to indicate which portion of the CP-ARQ PDU is contained in a SAR-PDU. That is, the AAU bit is set to 1 for the last cell of a CS-ARQ PDU, and set to 0 for all other cells. The process is the same as in AAL 5, except that the AAU bit is used to delineate the CP-ARQ PDU in AAL-X instead of the CPCS-PDU.

2.4. Robustness of AAL X

In this section, we investigate some scenarios to check if AAL X layer is robust. We evaluate the following cases:

Scenario 1: Suppose that the first cell or any middle cell (AAU=0) of the CP-ARQ PDU is lost or damaged: CRC-16 detects the error after receiving the last cell of the CP-ARQ PDU, and the receiver will ask for retransmission of this CP-ARQ PDU.

Scenario 2: Suppose that the last cell (AAU=1) of the CP-ARQ PDU is damaged: CRC-16 detects the error after receiving the last cell of the CP-ARQ PDU, and the receiver will ask for retransmission of this CP-ARQ PDU.

Scenario 3: Suppose that the last cell (AAU=1) of the CP-ARQ PDU is lost: The CP-ARQ sublayer cannot notice that a cell is missing until the receiver gets the last cell of the next CP-ARQ PDU. After that, CRC-16 detects the error and the receiver will ask for retransmission of the first CP-ARQ PDU. For the second CP-ARQ PDU, it will ask for retransmission by the missing sequence number after receiving another subsequent CP-ARQ PDU.

Scenario 4: Suppose that all cells of a CP-ARQ PDU are lost: The sequence number in the CP-ARQ PDU header catches the missing CP-ARQ PDU, and the receiver will ask for retransmission of this CP-ARQ PDU.

Scenario 5: Suppose that any octet in the CPCS-PDU is matched with BTag or ETag field by chance: This problem can be solved by the length field in the CPCS-PDU header.

Scenario 6: CRC Failure Case

First, consider the probability of CRC failure. For CRC-16, the undetected error probability of burst errors of length larger than 17 is 0.0015%, which is very low. However, it is true that the CRC could fail to detect errors although the probability is very small. One solution to that is to add a length field to the CP-ARQ PDU. Since the length field needs about 2 bytes, the overhead size becomes 6 bytes for each CP-ARQ PDU, which is too much, when considering the entire overhead including a 20 byte TCP header, a 20 byte IP header, and a variable padding length (0-43 bytes).

Instead, the length field in the CPCS-PDU header is used as the final checking means for error detection, so as to minimize the number of overhead size. For example, the total overhead size in AAL X is 8 bytes (i.e., 4 byte overhead from the CPCS sublayer plus 4 byte overhead from the CP-ARQ sublayer) for the best case, which is the same as in the AAL 5. When the CPCS sublayer detects an error by the length field, it will send a retransmission request for the entire CPCS-PDU to the transmitter. Of course, the transmitter needs to hold CPCS-PDUs in the buffer until the acknowledgment for the CPCS-PDU arrives from the receiver. Since the buffer size is a multiple of the window size in the S-ARQ scheme, there is no need to worry about

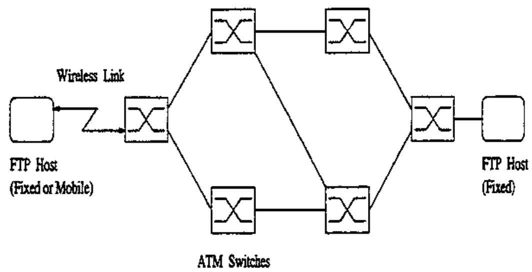

Figure 3: Simulation Model

the buffer problem. In order to acknowledge each CPCS-PDU, tag fields (BTag and ETag) can be used to identify CPCS-PDUs. For each CPCS-PDU, a unique bit sequence will be assigned.

Based on these scenarios, we can conclude that the AAL X protocol is reliable. It can easily detect the lost cells and cells in error, and ask for retransmission.

3. SIMULATION

The objective of our simulation is to evaluate the performance of AAL-X by demonstrating the performance improvements at the TCP level compared to the existing TCP approach with AAL 5 over wireless ATM networks.

3.1. Simulation Model

We develop our network simulation models using the MIL3's OPNET simulation package. As shown in Figure 3, the network model consists of 6 ATM switches. Each ATM switch supports 8 cross connections and the link speed is OC-3 (155 Mbps). Since the TCP/IP traffic is connectionless data, the service class is classified as class D where no time relation exists between the source and the destination, and the bit rate is variable. In each ATM switch, the buffer size of 1000 cells is reserved for class D traffic. Switching delays in the ATM switch and propagation delays are assumed to be negligible. The wireless channel is modeled as a Rayleigh fading channel to be combined with AWGN (Additive White Gaussian Noise) generator.

In order to evaluate the TCP/AAL 5 performance versus AAL-X, we conduct file transfers using FTP, which is a typical TCP application. The receiver buffer size in TCP is 64 Kbytes. For AAL 5, the initial retransmission timer value is set to 1 sec. When the retransmission timer expires, TCP always triggers slow start and retransmission at the same time as in the standard approach of AAL 5.

We develop two simulation models as follows:

- *Configuration 1*: TCP/IP and AAL 5 (as the standard in OPNET)
- *Configuration 2*: We keep the congestion control in TCP/IP and shift the error control function of TCP to AAL-X.

Since the AAL-X takes care of error control in *Configuration 2*, TCP does not have to request retransmissions.

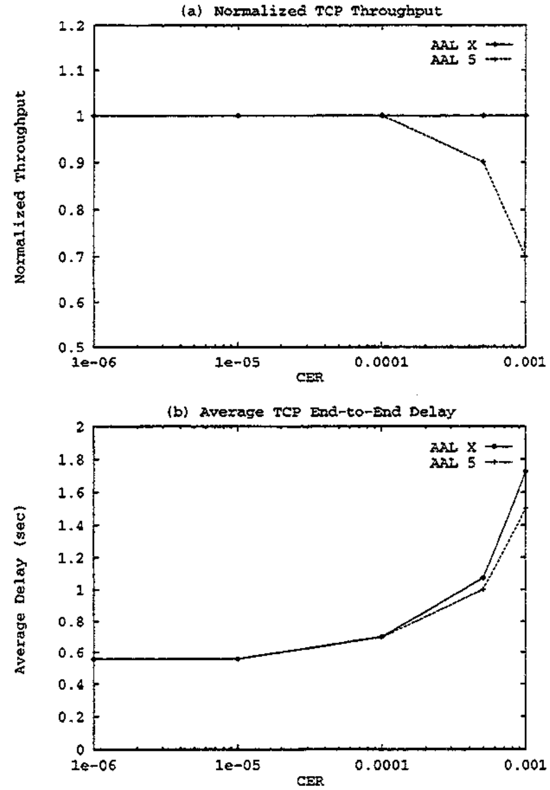

Figure 4: Comparison of Normalized Throughput and Average End-to-End Delay at the TCP level

We can easily eliminate the retransmission part from the TCP protocol by setting the timer value to infinity. As a result, the TCP retransmission function cannot be invoked, because timeout events never occur due to the infinite timer.

3.2. Simulation Results

We consider a simulation setup for wireless ATM. For simulation parameters, the carrier frequency is 2.4 GHz ISM band and the data rate is DS1 (1.544 Mbits/s). We measure the following performance parameters as a function of CER (Cell Error Rate) on a simulated wireless ATM network, where CER is defined as the percentage of errored or lost cells to total cells transmitted.

- *TCP offered load (bits/sec)*: The average rate of traffic offered to the TCP layer by the applications at the source. It is calculated by dividing the total bits submitted by the simulation time.
- *TCP throughput (bits/sec)*: The total number of bits forwarded to the application layer by the TCP layer at the destination.
- *TCP end-to-end delay time (sec)*: The end-to-end delay of packets received by the TCP layer. It is measured from the time an application data packet is sent

to the source TCP layer to the time it is received by the TCP layer at the destination.

Figure 4(a) presents the normalized TCP throughput as a function of CER for AAL 5 versus AAL-X. The normalized throughput is defined as the ratio of throughput to offered load. After some transient period in the early simulation time, the AAL-X keeps a perfect normalized throughput 1.0 until CER of 10^{-3} which is the range of our interest, while AAL 5 provides lower normalized throughput of 1.0, 0.9, and 0.7 for CER of 10^{-4}, 5×10^{-4}, and 10^{-3} each, as shown in Figure 4(a).

On the other hand, even if the AAL-X pays the price of slightly more delay due to the overhead from the CP-ARQ sublayer in return for higher throughput as shown in Figure 4(b), this causes no problem with quality-critical, but delay-insensitive traffic such as TCP/IP traffic. For example, the average TCP end-to-end delay in AAL-X is about 1.7 seconds at the CER of 10^{-3}, a little longer than 1.5 seconds in AAL 5, but it is acceptable in case of delay-insensitive traffic. In summary, since the AAL-X provides higher throughput at the TCP level for higher CER ranges like in a wireless ATM path, it can be a good solution for improving TCP performance over wireless ATM networks.

4. CONCLUSIONS

Since the poor performance of TCP over wireless ATM networks is mainly attributed to the fact that TCP always responds to all packet losses by congestion control, the error control portion of TCP is pushed down to the AAL layer so that TCP is only responsible for congestion control. In this paper, we propose a new AAL protocol (AAL-X) to take care of error control, which leverages off the current standard AAL 5 protocol. The AAL-X is based on a reliable ARQ-based mechanism to support quality-critical TCP traffic over wireless ATM networks. Moreover, the AAL-X has the ability to increase the throughput efficiency by using the variable length of CP-ARQ PDU, depending on the end-to-end path conditions.

The proposed AAL protocol has been validated using the OPNET simulation tool. We conducted file transfer experiments for different CER values, and compared with the standard approach of TCP/IP with AAL 5, in order to evaluate the TCP performance over AAL-X. In summary, since the AAL-X provides higher throughput at the TCP level for higher CER values as compared to the traditional TCP approach with AAL 5, it can be justified as a good solution for improving TCP performance over wireless ATM networks.

5. REFERENCES

[1] I. F. Akyildiz and I. Joe, "A New ARQ Protocol for Wireless ATM Networks," *Proceedings of IEEE International Conference on Communications ICC '98*, pp. 1109-1113, June 1998.

[2] ATM Forum Wireless ATM Working Group, *ATM Forum Report*, December 1997.

[3] L. S. Brakmo *et al*, "TCP Vegas: New Techniques for Congestion Detection and Avoidance," *ACM SIGCOMM*, pp. 24-35, August 1994.

[4] K. Fall and S. Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP," *ACM SIGCOMM*, Vol. 26, No. 3, pp. 5-21, July 1996.

[5] V. Jacobson, "Congestion Avoidance and Control," *ACM SIGCOMM*, pp. 314-329, 1988.

[6] V. Jacobson, "Modified TCP Congestion Avoidance Algorithm," *Technical Report*, April 1990.

[7] M. Mathis *et al*, "TCP Selective Acknowledgment (SACK) Options", *RFC-2018*, October 1996.

[8] J. Postel, "Transmission Control Protocol," *RFC 793*, September 1981.

[9] M. Schwartz, "Telecommunication Networks: Protocols, Modeling and Analysis," *Addison-Wesley*, 1987.