Performance Improvement of TCP in Wireless Cellular Network Based on Acknowledgement Control

MASAHIRO Miyoshi†, MASASHI Sugano‡, MASAYUKI Murata†

†Department of Infomatics and Mathematical Science, Graduate School of Engineering Science, Osaka University, 1-3 Machikaneyama, Toyanaka, Osaka 560-8531, Japan

‡Osaka Prefecture College of Health Sciences, 3-7-30 Habikino, Habikino, Osaka 583-8555, Japan

SUMMARY We present a simple method for improving TCP performance in a wireless cellular network. In our proposed method, the TCP receiver does not send one ACK, but multiple ACKs when the packet loss rate exceeds the predefined threshold. Since the major application of the current Internet is to download the Web documents from the fixed servers to the wireless terminal, it only requires changes at the wireless terminal side for performance improvement. Then, TCP becomes robust against radio link errors. We present an analytical method, and show that two ACKs for each TCP packet is sufficient to improve the performance.

Keywords: Wireless cellular network, TCP, Throughput, Transmission errors, ACK

I. Introduction

A mobile Internet technology on wireless cellular networks has been developed rapidly in these several years. In the Internet, TCP is used as a transport layer protocol. If packet loss is detected, TCP recognizes the congestion occurrence of the network and performs congestion control by throttling the congestion window. Therefore, it is well known that the packet losses due to the transmission errors cause unexpected degradation of TCP throughput in a wireless cellular network environment. Especially when ACK is lost, in spite of having sent the data segment correctly, the performance of TCP deteriorates terribly.

Many researches have been presented to improve TCP throughput in the wireless cellular network. For example, ELN (Explicit Loss Notification) [1] controls the window size appropriately by observing packet losses on the radio link. WTCP [2] changes window based congestion control to rate based congestion control. However, those solutions have not been realized because those require major changes to network infrastructures. ELN needs to change both the BSs (Base Stations) and wireless terminals to observe packet losses on the radio link. WTCP needs to change both clients and servers for the new TCP congestion control algorithm.

In this paper, we propose a method for improving the performance of TCP by a minor change in treating the ACK packets. In our proposed method, the TCP receiver does not send one ACK, but multiple ACKs when the packet loss rate exceeds the predefined threshold. Since the major application of the current Internet is to download the Web documents from the fixed servers to the wireless terminal, it only requires changes at the wireless terminal side for performance improvement. Wireless terminals monitor a packet loss rate, and estimate the error rate of the wireless channel. Then, TCP becomes robust against radio link errors. We present an analytical method, and show that two ACKs for each TCP packet is sufficient to improve the performance.

The rest of this paper is organized as follows. In Section II, we explain the model of the wireless cellular network under consideration. Then, we explain our proposed method in Section III. We derive the appropriate number of ACKs analytically. In Section IV, we evaluate TCP performance by means of simulation. Finally, Section V is devoted to concluding remarks.

II. Network model

In this paper, we investigate the performance of TCP for the network configuration shown in Figure 1 [3], with IMT-2000 support [4]. In this model, it is assumed that a TCP segment is transmitted towards the wireless terminal from the wired terminals (or servers). We mainly consider packet losses due to buffer overflow at the wireless terminals and transmission errors on the radio link. We assume the TCP Reno version, which is a major implementation in the current TCP code.

III. Performance improvement by multiple acknowledgments

In order to make TCP robust against ACK losses caused by transmission errors, the wireless terminal send multiple ACKs when the packet error probability of the wireless network p_{err} exceeds the threshold value. See Figure 2. When the wired terminal sends a TCP segment and the wireless terminal returns multiple ACKs, the following three cases are considered.

(1) The wired terminal receives one ACK, and it continues the normal operation.
(2) The wired terminal receives more than two ACKs and it recognizes the second and proceeding ACKs

as duplicate ACKs. It invokes the fast recovery in this case.

(3) Only if all of multiple ACKs are lost, the timeout occurs at the wired terminal.

While transmitting two or more ACKs decreases the loss of ACK due to the transmission error on the wireless link, duplicate ACK may be received at the wired terminal as in the case (2) above. Thus, it is effective in transmitting two or more ACKs in the range of the high transmission error rate. However, if the error rate is low, the TCP performance might deteriorate by the fast recovery. Therefore, it is necessary to determine the appropriate number of ACKs according to the quality of wireless channel. In the proceeding subsections, we explain how to estimate the rate of packet loss in a radio link, and the method of deriving the appropriate number of ACKs analytically. We last note that even when the wired network actually falls into congestion, the multiple ACKs does not affect the proper congestion control operation.

III.1 Estimation on packet loss rate on wireless link

Packet loss on the radio link and the buffer overflows at the bottleneck buffer take place independently. Then the total packet loss rate observed at wireless terminal, p, is expressed as follows:

$$p = 1-(1-p_{err\_uplink})(1-p_{buff})(1-p_{err\_downlink}) \qquad (1)$$

Here, p_{err_uplink}, $p_{err\_downlink}$, and p_{buff} represent the packet error probability of the uplink, the packet error probability of the downlink, and the packet loss probability at the bottleneck buffer, respectively. When the error rate of uplink and downlink is assumed to be almost equal to p_{err}, we have

$$p_{err\_uplink} \cong p_{err\_downlink} = p_{err} \qquad (2)$$

Then Eq. (1) can be simply rewritten as;

$$p_{err} = 1-\sqrt{\frac{1-p}{1-p_{buff}}} \qquad (3)$$

Thus, we can recognize p_{err} by monitoring p at the wireless terminal according to Eq. (3). Here, p_{buff} can be observed at the wireless terminal because the wireless terminal is bottleneck in the current network model. Otherwise, p_{buff} can be given as a fixed parameter if the router employs RED [5].

In monitoring the rate of packet rejection, p, at the wireless terminal, its change would become too large if the time unit of measurement is small. Conversely, when the time unit of measurement is large, more memories at the wireless terminal are necessary. We therefore introduce a method of calculating p by the moving average method, following an estimation method of RTT (Round Trip Time) values in TCP [6].

We show the above-mentioned in Figure 3 by ns-2 simulator [7]. Here, network model is shown in Figure 1 and parameter setting is summarized in Table 1. In these conditions, we change TCP connection from one to five like in Figure 3 and calculating p per one second. In Figure 3, the curve of moving average is in a good agreement with each p.

$$p[k] = \frac{7}{8} p[k] + \frac{1}{8} p[k-1]$$
$$p[0] = 0 \qquad k = 0,1,2\ldots \qquad (4)$$

III.2 Derivation of appropriate number of ACKs

Let p (n) denote the packet loss rate when transmitting n ACKs from the wireless terminal. p(n) is given by;

$$p(n) = p_{buff}(n) + p_{err}(n) - p_{buff}(n) p_{err}(n)$$
$$p(1) = p, \quad p_{buff}(1) = p_{buff}, \quad p_{err}(1) = p_{err}$$
$$n = 1,2,3\ldots \qquad (5)$$

where $p_{buff}(n)$ and $p_{err}(n)$ are the packet error rate of bottleneck buffer and wireless link when transmitting n ACKs, respectively. $p_{err}(n)$ can be determined as:

$$p_{err}(n) = p_{err}{}^{n} \qquad (6)$$

and $p_{buff}(n)$ can be determined by the following equation:

$$p_{buff}(n) = P_{dACK0} P_{rcv3}(n) + P_{dACK1} P_{rcv2}(n) + P_{dACK2} P_{rcv1}(n) + P_{dACK3} \qquad (7)$$

where P_{dACK0}, P_{dACK1}, P_{dACK2} and P_{dACK3} are probabilities that receive zero, one, two, and three ACKs. These are represented by the following expressions;

- Probability of receiving no duplicate ACK:
  $$P_{dACK0} = 1 - \sqrt[3]{p_{buff}} \qquad (8)$$
- Probability of receiving one duplicate ACK:
  $$P_{dACK1} = \sqrt[3]{p_{buff}} - (\sqrt[3]{p_{buff}})^2 \qquad (9)$$
- Probability of receiving two duplicate ACKs:
  $$P_{dACK2} = (\sqrt[3]{p_{buff}})^2 - p_{buff} \qquad (10)$$
- Probability of receiving three duplicate ACKs:
  $$P_{dACK3} = p_{buff} \qquad (11)$$

Furthermore, $P_{rcv1}(n)$, $P_{rcv2}(n)$ and $P_{rcv3}(n)$ are probabilities of receiving ACKs out of n ACKs. These are calculated from the binomial distribuion with a generating probability p_{err}.

- Probability of receiving one or more ACKs out of n ACKs:
  $$P_{rcv1}(n) = \sum_{i=1}^{n} {}_{n}\mathrm{C}_{i} (1 - p_{err})^{i} p_{err}^{n-i} \qquad (12)$$
- Probability of receiving two or more ACKs out of n ACKs:
  $$P_{rcv2}(n) = \sum_{i=2}^{n} {}_{n}\mathrm{C}_{i} (1 - p_{err})^{i} p_{err}^{n-i} \qquad (13)$$
- Probability of receiving three or more ACKs out of n ACKs:

$$P_{rcv3}(n) = \sum_{i=3}^{n} {}_nC_i (1-p_{err})^i p_{err}^{n-i} \tag{14}$$

Thus, we can determine p(n) by using Eqs. (5) through (14) by parameters n, p_{err} and p_{buff}.

Next, we derive TCP throughput for each number of ACKs analytically using the formula shown in [8]. In our analysis, it is characterized by three parameters RTT (Round Trip Time), To (Time Out Time) and p as follows;

$$S_{TCP} = \frac{1}{RTT\sqrt{\frac{2bp}{3}} + To\min(1, 3\sqrt{\frac{3bp}{8}})p(1+32p^2)} \tag{15}$$

where b is a delayed ACK parameter. Normally, b = 2.

Figures 4 and 5 plot the packet loss rate and TCP throughput when changing the packet error rate on the radio link, p_{err}. We set $p_{buff} = 0.01$, RTT = 100 ms and To = 400 ms. As shown in Figure 5, transmission of two ACKs can achieve the best performance in the range of $0.1 < p_{err} < 0.47$. The reason is that as shown in Figure 4, transmission of two ACKs results in smallest p in the above-mentioned range.

IV. Simulation results

In this section, we evaluate our proposed method by ns-2 simulator [7]. Parameter setting is summarized in Table 1.

Simulation results are shown in Table 2 (the number of wireless terminal is one). As can be observed in the table, transmission of two ACKs can improve the throughput with p_{err} larger than 0.05.

Then we show the window size appearance of TCP in the case of our method being effective ($p_{err} = 0.1$) in Figure 6. As shown in the figure, our proposed method keeps larger window sizes than the original method. On the other hand, as shown in Figure 7, the window size is not larger than the original one with $p_{err} = 0.01$. It is because transmissions of two ACKs per each data packet causes much duplicate ACKs and the TCP sender recognizes it as packet loss to invoke the fast recovery.

Moreover, we show the results in the case of increasing wireless terminal in Table 3 (the number of wireless terminal is five). As can be observed in the table, transmission of two ACKs can improve the throughput with p_{err} larger than 0.1. In this case, the effective range is narrower than Tabel 2 (the number of wireless terminal is one). It is because transmissions of two ACKs per each data packet causes much more duplicate ACKs due to increase wireless terminal.

Thus, it becomes important to choose the number of ACKs according to the value of p_{err}. As shown in the analysis, our method can easily incorporate such a control method by monitoring p and observing p_{buff} at the wireless terminal in the current network model.

V. Conclusion

We have presented a method for improving TCP throughput in a wireless cellular network, which needs changes of TCP layer only by the side of a wireless terminal. In our method, the wireless terminal sends multiple ACKs. By introducing it, it is expected that TCP becomes robust against radio link errors. We have introduced the estimation method of packet error rate on the radio link, and have determined the appropriate number of ACKs by analytical method. By means of simulation, we have revealed that TCP throughput can be improved in the range with the high error rate of the radio link as we have expected.

References

[1] H. Balakrishnan and R. H. Katz, "Explicit Loss Notification and Wireless Web Performance," Proc. IEEE GLOBECOM '98 Internet Mini-Conference, Nov. 1998.

[2] P. Sinha, N. Venkitaraman, R. Sivakumar and V. Bharghavan, "WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks," Proc. ACM MOBICOM '99, pp.231-241, Aug. 1999.

[3] H. Inamura and T. Ishikawa, “A TCP for W-CDMA:3G wireless packet service.draft-inamura-docomo-00”, July 2000.

[4] http://www.itu.int/imt/

[5] S. Floyd and V. Jacobson, “Random Early Detection Gateways for Congestion Avoidance,” IEEE/ACM Transaction on Networking, Aug. 1993.

[6] W. R. Stevens, TCP/IP Illustrated Volume 1: The Protocols. Addison-Wesley, pp.297-322, 1994.

[7] http://www.isi.edu/nsnam/ns/

[8] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation," Proc. ACM SIGCOMM '98, Sep. 1998.

Author biography

Masahiro Miyoshi

was born in Hyogo, Japan, on December 26, 1964. He received the B.E. degrees in Faculty of Engineering Department of Electrical and Electronics Engineering from Kobe University, Japan, in 1988. In 1988, he joined Denso Corporation Co. Ltd. as a Researcher. His research is in ITS network.

Masashi Sugano

received the B.E., M.E., and D.E. degrees in Information and Computer Sciences from Osaka University, Japan, in 1986, 1988, and 1993, respectively. In 1988, he joined Mita Industrial Co. Ltd. (currently, Kyocera Mita Corporation) as a Researcher. From September 1996, he has been an Associate Professor of Osaka Prefecture College of Health Sciences. His research interests include design and performance evaluation of computer communication networks, network reliability, and wireless network systems. He is a member of IEEE and ACM.

Masayuki Murata

received the M.E. and D.E. degrees in Information and Computer Sciences from Osaka University, Japan, in 1984 and 1988, respectively. In April 1984, he joined Tokyo Research Laboratory, IBM Japan, as a Researcher. From September 1987 to January 1989, he was an Assistant Professor with Computation Center, Osaka University. In February 1989, he moved to the Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University. From 1992 to 1999, he was an Associate Professor in the Graduate School of Engineering Science, Osaka University, and from April 1999, he has been a Professor of Osaka University. He moved to Advanced Networked Environment Division, Cybermedia Center, Osaka University in April 2000. He has more than two hundred papers of international and domestic journals and conferences. His research interests include computer communication networks, performance modeling and evaluation. He is a member of IEEE, ACM, The Internet Society, and IPSJ.

Captions for all figures, photographs, and tables

Figure 1: Network model

Figure 2: Control of number of acknowledgment

Figure 3: Calculating p by the moving average method

Figure 4: Packet loss rate (analysis)

Figure 5: TCP throughput (analysis)

Figure 6: TCP window size (perr= 0.1)

Figure 7: TCP window size (perr= 0.01)

Table 1: Parameter sets

Table 2: TCP throughput (1 node)

Table 3: TCP throughput (5 node)

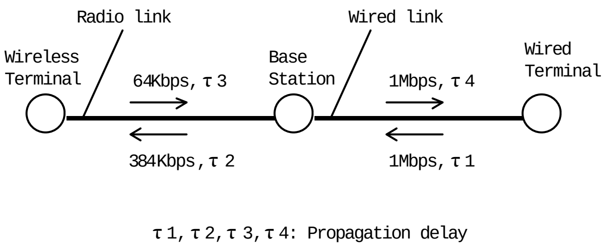

Figure 1: Network model

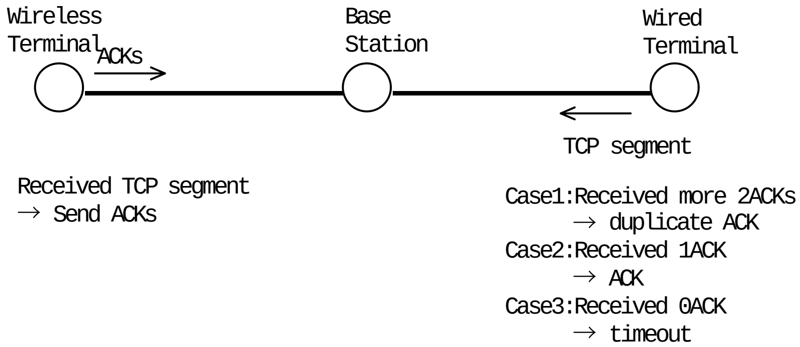

Figure 2: Control of number of acknowledgment

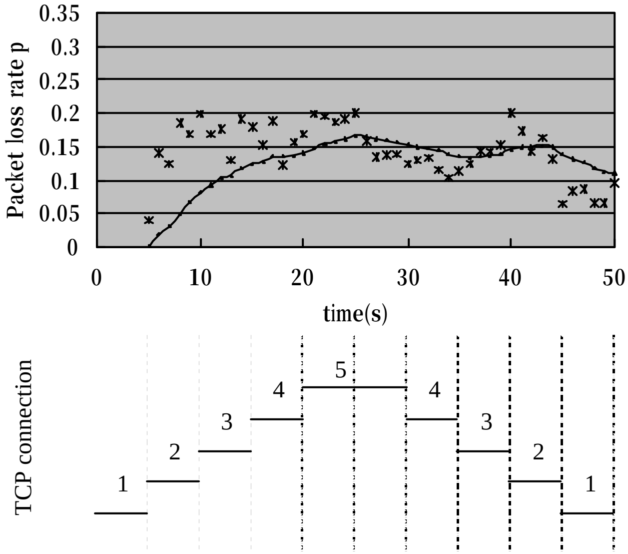

Figure 3: Calculating p by the moving average method

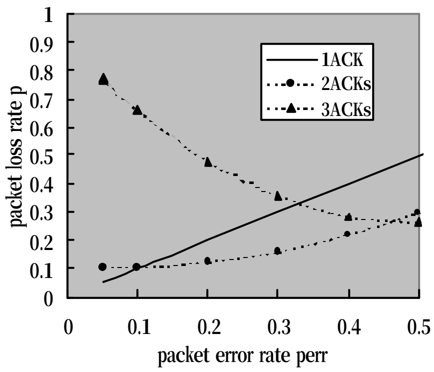

Figure 4: Packet loss rate (analysis)

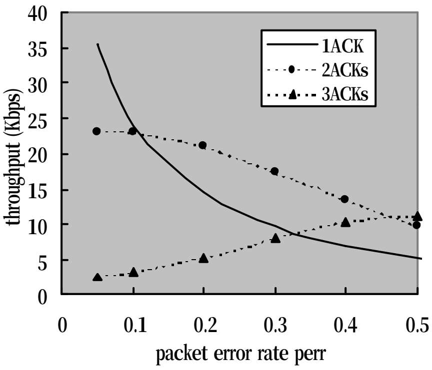

Figure 5: TCP throughput (analysis)

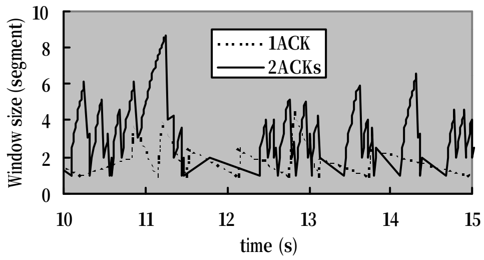

Figure 6: TCP window size (p_{err}= 0.1)

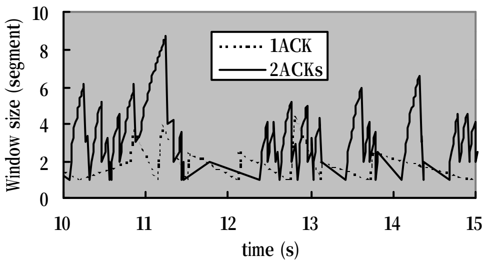

Figure 7: TCP window size (p_{err} = 0.01)

Table 1: Parameter sets

TCP segment size	100 byte
ACK size	40 byte
Buffer size (Wireless Terminal, BS, Wired Terminal)	50 Kbyte
Propagation delay (ô1,ô2,ô3,ô4)	1 ms

Table 2: TCP throughput (1 node)

perr	1ACK (Kbps)	2ACK (Kbps)
0	214.1	211.7
0.01	202.9	200.2
0.05	68.55	74.55
0.1	20.42	20.78
0.2	2.058	2.92
0.3	0.388	0.55
0.4	0.114	0.152
0.5	0.044	0.073

Table 3: TCP throughput (5 node)

perr	1ACK (Kbps)	2ACK (Kbps)
0	292.8	286.2
0.01	259	247.6
0.05	202.2	192.3
0.1	89.25	105.45
0.2	18.86	20.8
0.3	1.845	2.93
0.4	0.486	0.56
0.5	0.11	0.126