A TCP Acceleration Algorithm for a Wireless Link using Rate Adaptation Based on Round-Trip-Time and Virtual Receiver Window Information

Sungwon Lee, *Member, IEEE*, Seongsoo Park, Jaehwang Yu, and Choongseon Hong, *Member, IEEE*

***Abstract*—FAST Transmission Control Protocol (TCP) was previously proposed for high capacity network environments with long delay, and "FAST TCP with Snoop" performs better than conventional TCP enhancements in mobile wireless network environments. However, FAST TCP has limitations when used over a dynamic mobile wireless link with a high frame error ratio (FER) and frequent delay changes due to the variable rate. We propose an enhanced TCP acceleration algorithm at the TCP sender side which efficiently adapts to the maximum transmission rate of a mobile wireless link using the round trip time (RTT) and virtual receiver window (RWND) information. The proposed algorithm provides superior performance over mobile wireless network environments.**

***Index Terms*—TCP acceleration, FAST TCP, rate adaptation.**

I. Introducion

TCP performance improvements are widely studied and many algorithms have been proposed [1]. If we consider mobile wireless network environments, "FAST TCP with Snoop", which is based on TCP Vegas with high capacity in long delay network environments, shows the best performance [2][3].

FAST TCP is susceptible to both queuing delays and packet loss. Under normal network conditions, FAST TCP periodically updates the congestion window (w) based on the average RTT according to Equation (1), where $\gamma \in (0,1]$, baseRTT is the minimum observed RTT, and α is a positive protocol parameter that determines a router's total number of queued packets in equilibrium with the flow's path. Thus, as described in Equation (1), an accurate estimation of RTT is the key performance factor for FAST TCP [3].

$$w = min\{2w, (1-\gamma)w + \gamma(\frac{baseRTT}{RTT}w + \alpha)\} \quad (1)$$

Although FAST TCP provides improved performance, it has several issues in mobile wireless network environments. In mobile wireless networks, the retransmission of the data-link layer protocol (e.g., radio link control or Snoop) can cause a large packet transmission delay. Also, the dynamic bandwidth in a mobile wireless link can cause variable packet transmission delays. However, if packet loss or a long transmission delay occurs within a mobile wireless link for any of the above reasons, FAST TCP will operate in the same manner as conventional TCP. Thus poor performance will occur due to an incorrect estimation of the TCP transmission rate [3][4]. FAST TCP also shows performance degradation with the occurrence of the dynamic addition or deletion of TCP sessions within a shared mobile wireless link [5]. Asymmetric links may also cause performance degradation [6]. As a result, the main reason for a loss of performance in FAST TCP is its assumption of a stable network in which packet loss is uncommon. Also, FAST TCP considers long delay times to be queuing buffers within the network. Thus, it is limited in adapting to a dynamic mobile wireless link.

An enhanced TCP acceleration algorithm for a mobile wireless link based on a sender side transmission rate adaptation using the virtual RWND size value and RTT information is proposed here. When integrated with Snoop, the proposed algorithm performs better than conventional "FAST TCP with Snoop". The proposed algorithm hides the un-stable and dynamic characteristics of the mobile wireless link from the TCP sender using Snoop and virtual RWND. Using RTT with virtual RWND, the proposed scheme quickly adapts to the maximum transmission rate of a dynamic wireless link.

The proposed algorithm is described in Section II, and its performance is evaluated in Section III. Section IV concludes the paper and presents suggestions for future work.

II. TCP Acceleration using Rate Adaptation (TARA) Algorithm

The goal of the proposed algorithm is to calculate the near optimal transmission rate of a last mile mobile wireless down-link. The calculation of the near optimal transmission rate allows for the quick adaptation of the sender side's TCP transmission rate to a dynamically changing mobile wireless link. Two performance improvements are used in the proposed algorithm. One is the virtualized TCP RWND value at the base-station (BS), and the other is the adapted transmission rate calculation scheme for TCP senders, such as FTP/HTTP servers. The proposed algorithm is described in Figure 1. A BS supports the conventional Snoop algorithm, which supports local error recovery over a mobile wireless link between an MS and a BS. This paper does not propose an algorithm to replace Snoop. Instead, we use conventional Snoop and its enhanced algorithms. Thus, the proposed TCP Acceleration using a Rate Adaptation (TARA) algorithm can be integrated with the original Snoop and its enhanced variants.

Additionally, a BS supports a virtual RWND feature by replacing the original RWND value of a received TCP packet

Manuscript received March 4, 2009. The associate editor coordinating the review of this letter and approving it for publication was F. Theoleyre.

This research was supported by SK Telecom.

S. Lee and C. Hong are with the Department of Computer Engineering, College of Electronics and Information, Kyung Hee University, Korea (e-mail: drsungwon@khu.ac.kr).

S. Park and J. Yu are with the Institute of Future Technology, SK Telecom C&I Business.

Digital Object Identifier 10.1109/LCOMM.2009.090520

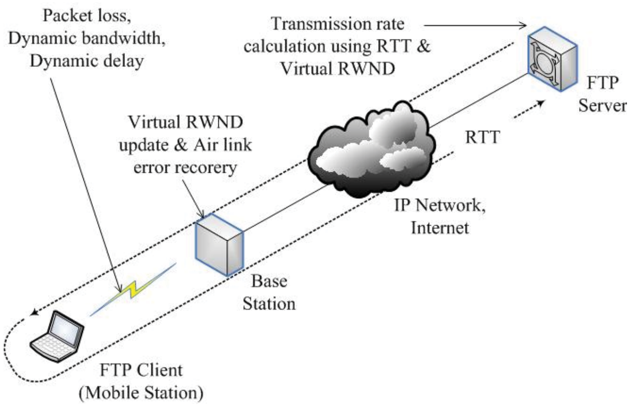

Fig. 1. Proposed algorithm operation scenario.

from an MS with the BS's own packet buffer size. Snoop and virtual RWND hide the unstable and dynamic characteristics of the mobile wireless link to the TCP sender. Thus, a BS can accumulate enough packets from the TCP sender, in spite of the TCP sender's transmission window fluctuation, which is commonly caused by packet loss or dynamic delay. The BS can also fulfill a mobile wireless link independently of the conditions of the link. According to the near optimal transmission rate calculation of the TCP sender, the proposed algorithm calculates a transmission rate for an MS based on the RTT of the MS and the virtual RWND from the BS. To measure the RTT for the TCP sender, the sender stores a transmission time for each TCP packet, and calculates the RTT for each packet when the acknowledgement arrives from the MS.

In the proposed algorithm, the TCP sender calculates the new transmission rate (R_{new}) based on Equation (2) for each TCP session, where $TE\_sensitivity(\theta)$ is a weighting factor used to limit the maximum transmission rate of a TCP session.

$$R_{new}(b/s) = \frac{VirtualRWND(bits)}{RTT(ms)} \times \theta \qquad (2)$$

The $TE\_sensitivity$ value can be used to control the peak rate of the TCP sender. The limitation can be set to the maximum transmission rate of the mobile wireless link, or to a target data rate which is less than the maximum transmission rate and is based on the service level agreement (SLA) between the operator and the mobile station subscriber. However, if we assign a large $TE\_sensitivity$ value, which can exceed the maximum transmission rate of the mobile wireless link, TARA automatically adapts its transmission rate using a virtual RWND as well as the RTT information. Based on Equation (2), a TCP sender can increase the TCP transmission rate to the maximum rate in a very short time. When the first round-trip packet arrives back at the TCP sender, it will calculate the RTT and determine the virtual RWND. Thus, a TCP sender can increase the TCP transmission rate up to a specific peak rate as soon as the calculated RTT value for the first round-trip packet is available. Equation (2) also shows that the proposed algorithm can efficiently adapt to the dynamic mobile wireless link status. For example, if a mobile wireless link experiences a large FER or reduced bandwidth due to a bad air link condition, Equation (2) will reduce the TCP transmission rate. In turn, it will reduce the accumulated packet buffer size at the BS during poor mobile wireless link conditions, such as handoff or high error level.

III. Performance Evaluation

We evaluated the performance of the proposed algorithm and compared it to a conventional algorithm with an NS2 simulator. We considered "FAST TCP with Snoop" as our conventional algorithm, and used the FAST TCP and Snoop modules in NS2. An FTP service was considered. The FTP client was located at an MS, and the FTP server (TCP sender) was located at the network side. Snoop was deployed at the BS, and FAST TCP was deployed at the FTP server. For the proposed algorithm, Snoop and the virtual RWND feature were used at the BS, and the TARA feature was deployed at the FTP server. The transmission rate of the mobile wireless link between the MS and the BS was assumed to follow two models. One was the "Planned Network" model that represents a good air link. In this case, the peak rate was 10 Mb/s, the minimum rate was 8 Mb/s, and the mean rate was 9 Mb/s with a uniform distribution. The other model was the "Unplanned Network" model that represents poor air link environments with abrupt air link quality changes, with a peak rate of 10 Mb/s, a minimum rate of 1 Mb/s, and a mean rate of 5.5 Mb/s with a uniform distribution. Several FER cases were considered at 0, 1, and 5% with uniform distributions. The transmission rate of the wire-line between the BS and the FTP server was assumed to be a 100 Mb/s fixed rate without FER. Also, single user and multiple user cases were considered. The BS's buffer size (= maximum virtual RWND) for each TCP session was configured to be 900 TCP packets.

For the performance comparison factor, we used the effective data rate ($R_{effective}$), which is the rate that the user experiences from the communication link. The effective data rate was calculated using Equation (3), where S_{ftp_file} is the FTP file size (1/10 MB), $T_{complete}$ is the FTP completion time, and $T_{request}$ is the FTP request time.

$$R_{effective}(b/s) = \frac{S_{ftp\_file}(bits)}{T_{complete} - T_{request}(sec)} \qquad (3)$$

Figure 2 and Figure 3 show an effective data rate comparison for the "Planned Network" model and the "Unplanned Network" model, respectively. We configured the $TE\_sensitivity$ to be 0.03 to allow the calculated maximum rate to reach the peak data rate (in this case, 10 Mb/s) of the mobile wireless link. This value can be derived from the assumed RTT (20 ms) and the BS's TCP session buffer size (900 packets).

The simulation results showed that the proposed TARA algorithm provided improved performance over "FAST TCP with Snoop". In the planned network model, performance improvement gains for the single user case reached 20%, 20%, and 28% for FERs of 0%, 1%, and 5%, respectively. For the multiple user case, performance improvement gains reached 3%, 11%, and 17% for FERs of 0%, 1%, and 5%, respectively. The simulation results for the unplanned network model showed slightly better performance improvement gains, which reached 34%, 49%, and 40% for FERs of 0%, 1%, and 5%, respectively, in the single user case. For the multiple user

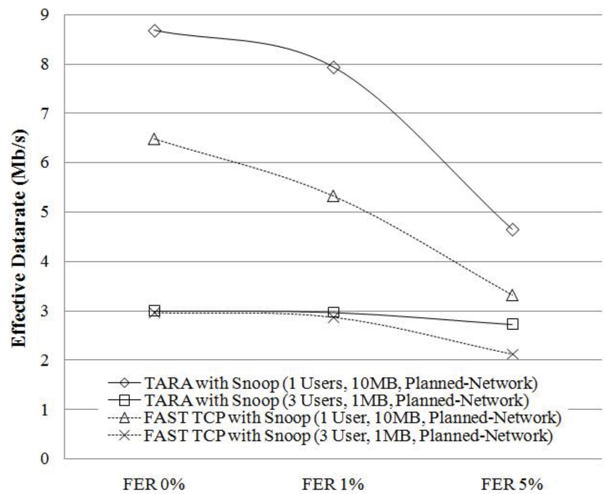

Fig. 2. Effective data rate for planned network model.

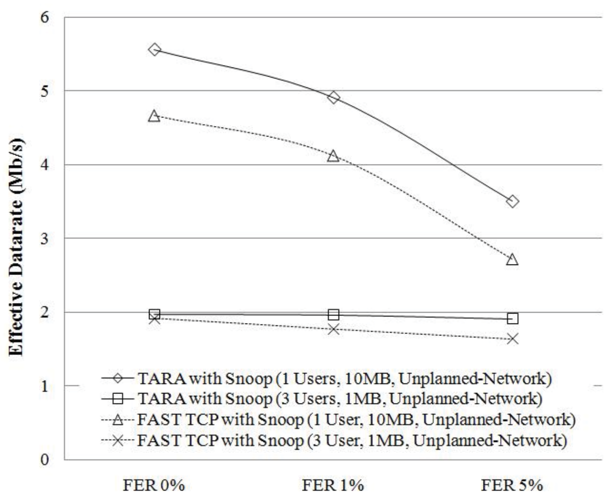

Fig. 3. Effective data rate for unplanned network model.

case, performance improvement gains reached 2%, 4%, and 29% for FERs of 0%, 1%, and 5%, respectively.

Performance improvements occurred for three reasons. The first reason for this is the fast transmission rate adaptation of the TCP sender shown in Equation (2). Figure 4 shows the cumulative distribution function (CDF) of the measured air link rate at the BS, and the CDF of the calculated TCP transmission rate at the sender side using TARA where the $TE\_sensitivity$ was varied by 0.03 and 0.02. If $TE\_sensitivity$ was assumed to be 0.03, which means TARA supports the maximum air link rate (10 Mb/s), the CDFs of the measured air link and the calculated transmission rate were similar. Thus, we can see that the proposed TARA adapted efficiently to dynamically changing air link environments as explained in Section II. In the graph in which the $TE\_sensitivity$ was 0.02, we can see that TARA, using the $TE\_sensitivity$ factor, limited the maximum transmission rate to 7.2 Mb/s. The second reason for performance improvement was the minimization of the impact of the mobile wireless link on the TCP sender using Snoop and virtual RWND. A low FER could be recovered by the BS, leaving enough packets to fulfill the mobile wireless link. The third reason was the instant rate reduction of the TCP sender

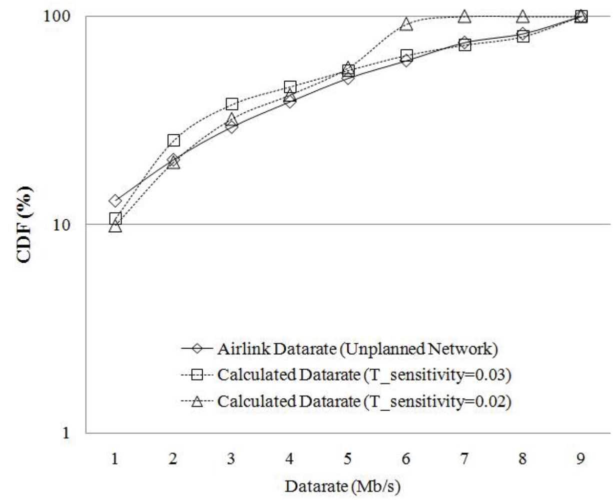

Fig. 4. CDF for measured airlink data rate and calculated data rates.

using RTT in the case of a high FER, or a reduced data rate due to a bad air link status, possibly caused by a handover. If RTT increased and virtual RWND decreased due to a poor air link status, the proposed algorithm reduced its transmission rate. Thus, the proposed algorithm can reduce the accumulated buffer size at a BS. However, as RTT decreased and virtual RWND increased, indicating that the mobile wireless link had a stable status, the proposed algorithm quickly recovered the transmission rate.

IV. Conclusion

The error recovery operation used in FAST TCP is the same as that in conventional TCP, and it considers a large RTT as a large network buffer. Thus, "FAST TCP with Snoop" causes performance degradation in mobile wireless network environments. Using virtual RWND and RTT, the proposed TARA algorithm quickly adapts to a maximum transmission rate and avoids buffer overflow at the BS for temporal error recovery. When integrated with Snoop, the proposed algorithm provides enhanced performance compared to "FAST TCP with Snoop" for both good and poor link quality wireless network environments.

References

[1] K.-C. Leung and V. O. K. Li, "Transmission control protocol (TCP) in wireless networks: issues, approaches, and challenges," *IEEE Commun. Surveys & Tutorials*, pp. 64-79, 4th Quarter 2006.

[2] S. Vangala and M. A. Labrador, "The TCP SACK-aware snoop protocol for TCP over wireless networks," in *Proc. Vehicular Technology Conference - Fall (VTC-Fall 2003)*, pp. 2624-2628, Oct. 2003.

[3] C. Jin, D. X. Wei, and S. H. Low, "FAST TCP: motivation, architecture, algorithm, performance," *IEEE/ACM Trans. Networking*, vol. 14, no. 6, pp. 1246-1259, Dec. 2006.

[4] F. Ge and L. Tan, "A partial super fast recovery algorithm for FAST TCP," in *Proc. 2nd International Conference on Wireless Broadband and Ultra Wideband Communications*, pp. 57-61, Aug. 2007.

[5] S. Belhaj and M. Tagina, "VFAST TCP: an improvement of FAST TCP," in *Proc. Tenth International Conference on Computer Modeling and Simulation (UKSIM 2008)*, pp. 88-93, Apr. 2008.

[6] F. Ge, L. Tan, and M. Zukerman, "Throughput of FAST TCP in asymmetric network," *IEEE Commun. Lett.*, vol. 12, no. 2, pp. 158-160, Feb. 2008.