Recommended Paper

Improving TCP Performance over Wireless with Data Link Layer ARQ

Satoshi Utsumi,† Salahuddin Muhammad Salim Zabir,††
Gen Kitagata†† and Norio Shiratori††

We propose a new TCP we named as "TCP Identification & Revivable Window (TCP-I&RW)" to improve TCP performance in wireless networks supporting data link layer automatic repeat request (ARQ) for link errors on wireless links. The TCP sender in TCP-I&RW places an identification tag decided by using random number for every data segment. Like conventional TCP protocols, if a segment loss is detected, it first infers a congestion, lowers the sending rate and retransmits with a different identification tag. The TCP sender figures out the actual cause of segment loss depending on identification reply in the acknowledgement corresponding to the retransmitted data. If the segment loss is not due to congestion, the TCP sender revives its transmission rate to the value prior to the retransmission. As such, erroneous detection of congestion is avoided. This ensures an improved throughput for TCP over wireless links supporting data link layer ARQ. Experiments show our proposed new scheme can achieve better performance than existing well established schemes in the cellular network and also in the satellite network. Also, TCP-I&RW incorporates a mechanism to nullify any attempt by aggressive TCP receivers hoping to occupy unfairly high shares of link bandwidth.

1. Introduction

In recent years, mobile and wireless devices are being incorporated to the Internet at a rapid pace. This initiates the need for optimizing major applications to perform satisfactorily over wireless links. Almost eighty percent of the Internet applications run on TCP [1]. Therefore, while accommodating mobile wireless devices in the Internet, TCP concerns deserve special attention [2)~4),8),13)~20)].

Wireless links are generally prone to a higher link error rate than their wired counterparts. Since conventional TCP protocols are designed to be used in wired networks with low link error rates, their designs do not take link errors into account. That is, they assume segment losses occur solely due to congestion in networks [6]. Whenever a segment loss occurs, conventional TCP infers congestion and the TCP sender reduces its sending rate as a remedial measure. Therefore, when conventional TCP are used over wireless networks, they interpret all the losses to be originating from congestion and frequently lower their transmission rates unnecessarily. As such, deployment of conventional TCP over wireless links results in a decreased throughput [2),4)].

The problem has so far been attempted to be solved using different approaches. Among them, solutions supported by data link layer are the most promising ones. Solving problems related to link errors primarily at data link layer, is natural as well as efficient from deployment point of view. Commercial standards and products for wireless connection also guarantee such provisions at data link layer. As such, in order to ensure high performance for TCP over wireless, TCP enhancement proposals that count on such data link layer support have received widespread popularity.

In this paper, we therefore, focus on this type of approach to improve TCP performance over wireless networks. The essential idea is that when a segment loss occurs, our TCP first behaves like conventional TCP by lowering the transmission rate. It then figures out the actual cause of each segment loss using some tricky identification tags in the corresponding acknowledgement. If tags suggest that the loss is not due to congestion, it revives the transmission rate to that before the detected loss. Our proposed TCP thus alleviates unnecssary decrease in TCP transmission rate and corresponding throughput degradation. We name this new TCP as "TCP Identification & Re-

† Advantest Corporation
†† Tohoku University

The initial version of this paper was presented at the DICOMO2002 held on Jul. 2002, which was sponsored by SIGMBL. This paper was recommended to be submitted to the IPSJ Journal by the Chairman of SIGMBL.

vivable Window (TCP-I&RW)". TCP-I&RW is designed for cases where wireless links support data link layer ARQ (Automatic Repeat reQuest) to recover from link errors. We propose modifications to the TCP sender only. The TCP receiver is kept unchanged if it supports commonly implemented TCP timestamp option. Our scheme does not pose implementation problems like the other well established existing wireless TCP solutions to be outlined in Section 4. In addition, experiments show this scheme yields better performance than existing ones in the cellular network and in the satellite network.

We organize the rest of this paper as follows. Section 2 outlines TCP characteristics over wireless links. In Section 3 we present our scheme. In Section 4 we describe other works to improve TCP perfromance over wireless supported by data link layer. In Section 5 we evaluate the performance implications of our scheme. In Section 6 we discuss an unique feature of our scheme in the light of aggressive TCP receivers. Finally we conclude in Section 7.

2. TCP Issue over Wireless Links

Conventional TCP senders detect segment losses using either of the following ways[5].

1. Retransmission Timeout: if the TCP sender does not receive the corresponding acknowledgement within a certain time RTO (Retransmission Timeout) after the TCP sender had sent the segment, it decides that the segment was lost in the network. RTO is determined based on RTT(Round Trip Time) of the connection dynamically.
2. Fast Retransmit: When the TCP receiver receives any segment which is not the expected next in-sequence, it sends a duplicate acknowledgement (dupack) containing the same acknowledgement number as the previous acknowlegement. The TCP sender decides that the segment was lost in network, if it receives three dupacks consecutively.

Conventional TCP can perform well in wired networks, where the segment losses due to link errors can be ignored[6]. But wireless links are prone to a high link error rate. Hence in wireless networks conventional TCP senders end up shrinking congestion window unnecessarily for segment losses due to link errors. As such, conventional TCP protocols cannot perform well in wireless networks.

Data link layer Automatic Repeat reQuest (ARQ) protocols can be used to hide the influence of link errors on wireless links from the end hosts. Data link layer ARQ protocols alone cannot always succeed in this effort for TCP because of the following two reasons.

1. If it takes too much time to retransmit on wireless links, timeout (i.e., spurious timeout) of TCP may occur[7].
2. When retransmissions on data link layer make the order of segment arrivals at the TCP receiver different from the order of segment transmissions at the TCP sender, the TCP receiver sends dupacks[8]. As such, fast retransmit (i.e., spurious fast retransmit) occurs.

3. Our Scheme

In this section, we propose a new scheme to avoid TCP performance degradation in the wireless network. This scheme has the following features.

1. Wireless links support data link layer ARQ.
2. Our proposed new TCP, i.e., TCP Identification & Revivable Window is deployed instead of conventional TCP.

3.1 New TCP: TCP Identification & Revivable Window

In our scheme, we use new TCP as the transport layer protocol. The values of congestion window and slow start threshold are stored for a certain time after timeout or receiving three dupacks. If data link layer ARQ on the wireless link recovers segment losses due to link errors, our proposed new TCP sender revives the congestion window and the slow start threshold to its pre-loss state by restoring the saved values. The new TCP sender accommodates the following new functions.

1. To place an identification tag on each data segment.
2. To memorize the values of the congestion window and the slow start threshold just before retransmitting the data.
3. To restore the values of congestion window and slow start threshold depending on the identification tag on the acknowledgement corresponding to the retransmitted data. Such restoration is named as *window revival* in this paper.

It is not needed to modify the TCP receiver implementing TCP timestamp option[9].

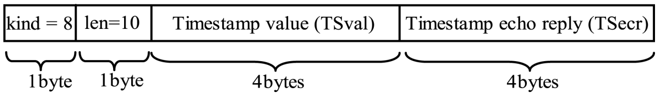

Fig. 2 TCP timestamp option field.

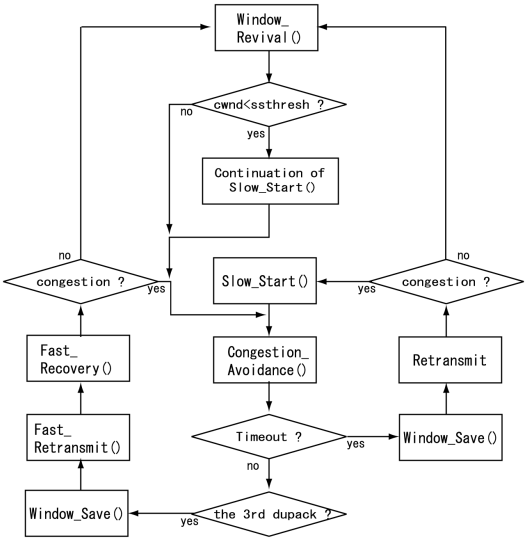

Fig. 1 Flowchart of TCP-I&RW sender operation.

We call this new TCP as TCP Identification & Revivable Window (TCP-I&RW). The TCP-I&RW sender operates on the following algorithms: *Slow_Start()*, *Congestion_Avoidance()*, *Fast_Retransmit()*, *Fast_Recovery()*, *Window_Save()* and *Window_Revival()*. First four algorithms have already been implemented in TCP-Reno[6]. The remaining two algorithms are proposed in this paper. **Figure 1** shows the flowchart of TCP-I&RW.

3.2 TCP Identification

While sending a data segment, the TCP-I&RW sender places a new identification tag (TCP identification) on the timestamp value (TSval) field on TCP header in each data segment. This tag distinguishes the data segment from other ones. The TCP receiver with TCP timestamp function support copies the TCP identification to timestamp echo reply (TSecr) field on the TCP header in the acknowledgement corresponding to the data segment as TCP identification reply. **Figure 2** shows the form of TCP timestamp option field on TCP header.

The TCP identification is composed of the data identification (DID) and the segment identification (SID) as shown in **Fig. 3**. Each block of data has a unique DID. However, the values of SID are different in each original transmitted data segment or retransmitted data segment.

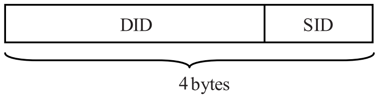

Fig. 3 TCP identification.

Since TCP identification is implemented on TSval fields, the value of the identification should support the PAWS (Protection Against Wrapped Sequence number) algorithm[9], which is implemented by TCP timestamp option. Otherwise, the TCP receiver may drop valid data segments. To support PAWS algorithm, we propose the following methods to decide the value of the identifications as follows.

1. DID is put before SID as more significant bits on the TCP TSval field (as figreffig:id).
2. DID takes on values from a monotone increasing function. For example, DID of two successive data blocks i and $i{+}1$ will maintain the following relation $DID_{i+1} = DID_i + 1$.

If the value of TCP identification on the unreceived data segment is gussable, *aggressive TCP receivers* may manipulate TCP identification reply on the acknowledgement to occupy an unfairly high share of the link bandwidth (described in Section 6). The TCP-I&RW sender therefore should decide SID randomly (ideally, using a random number generator).

Again, by maintaining appropriate data structure at the TCP-I&RW sender the basic TCP timestamp functionality is not affected.

3.3 Window Revival

Our proposed TCP-I&RW incorporates a window revival mechanism to overcome the reduced data transmission rate in case of segment loss due to link error. This functionality is achieved through the following data structure and algorithms.

1. The TCP sender adds TCP identification on each data segment. When the TCP sender sends a new data segment, it stores the value corresponding to the SID of TCP identification on the segment to the element of the array *seg_id[X]*, where X is the value corresponding to the DID of TCP identification on the data segment.

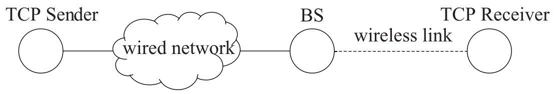

Fig. 4 System model.

2. Just before the TCP sender retransmits data first time being triggered by timeout or three dupacks, it stores the values of congestion window and slow start threshold in the variable *cwnd_prev* and *ssthresh_prev* respectively. This constitutes *Window_Save()* algorithm.
3. Upon receiveing the first acknowledgement for the retransmitted segment, the TCP sender considers TCP identification reply to be consisting of two parts, X and Y. The values of X and Y correspond to DID and SID in TCP identification reply respectively. If X equals DID of the retransmitted data segment and Y equals *seg_id[X]*, it is inferred that the originally transmitted TCP data segment has not been lost due to congestion. That is, its retransmission was spurious. Therefore, the TCP sender restores the values of congestion window and slow start threshold to the values of the variable *cwnd_prev* and *ssthresh_prev* respectively. this constitutes *Window_Revival()* algorithm.

4. Related Works

Snoop scheme [8] and Delayed Duplicate Acknowledgements scheme [3] are the most prominent existing solutions to improve TCP performance in wireless networks using retransmissions on wireless links. We provide a brief outline of these schemes using the network model as in **Fig. 4**. In this system, there is a base station BS between a TCP sender host and a TCP receiver host. The TCP sender host is connected to BS by the wired network and the host with the TCP receiver host is connected to BS by the wireless link.

The Eifel algorithm [19),20] has been proposed to overcome TCP performance problems related to spurious retransmissions. Therefore it improves TCP performance over wireless networks using data link layer ARQ. The mechanism has some structural similarity with the one in TCP-I&RW.

4.1 Snoop Scheme

Snoop is a TCP-aware retransmission mechanism on the wireless link. Snoop is implemented at the base station BS, where it observes TCP headers of segments passing through BS. Snoop behaves as follows.

1. Snoop buffers TCP data segments sent by the TCP sender until the corresponding acknowledgements pass through BS.
2. If Snoop receives a dupack from the TCP receiver and the corresponding lost segment is in the buffer, it drops the dupack and retransmits the corresponding segment.
3. If Snoop has not received the acknowlegement after sending a data segment until timeout determined by Snoop occurs, then it retransmits the data segment.

Snoop has the following problems on the implementation.

1. Per-flow management is needed on BS. So a large amount of resources on BS is consumed.
2. If IP encryption is used, this scheme is not usable.

Also TCP performance is affected owing to the following reasons.

1. Link error detections based on TCP-dupacks is not efficient.
2. If the round trip time between BS and TCP receiver are large, spurious timeout may occur frequently.

4.2 Delayed Duplicate Acknowlegements Scheme

Delayed Duplicate Acknowlegements (DDA) scheme uses data link layer ARQ on the wireless link. If the order of the segment arrival at the TCP receiver is different from the order of the segment transmission at the TCP sender, then the TCP receiver delays to generate the third dupack (and later dupacks) for a certain optimal time d. If the TCP receiver receives the segment retransmitted by data link layer ARQ during the time d, then the TCP receiver doesn't send the third dupack.

The implementation problems of DDA scheme lie with difficulties to estimate the optimal time d both statically and dynamically.

Also TCP performance is affected owing to the following reasons.

1. If the time d is too large, throughput may decrease due to spurious timeout.
2. If segment losses occur due to congestion, the error recovery is unnecessarily delayed for the time d. As such, the benefits of using dupacks vanish.

4.3 Eifel Algorithm

TCP implementing the Eifel algorithm detects spurious retransmissions resulting from spurious timeout or spurious fast retransmit in a manner similar to TCP-I&RW. TCP implementing the Eifel algorithm realizes optimal TCP performance in wireless networks with data link layer ARQ. The Eifel algorithm has similarity with the mechanism in TCP-I&RW at the following points.

1. The TCP sender distinguishes an original segment from retransmitted segments using TCP timestamp option.(An approach using 2 bits from the 4 remaining reserved bits in TCP header is also proposed [19),20)].)
2. The TCP sender stores the values of congestion window and slow start threshold just before retransmitting data. The TCP sender restores these values upon detecting a spurious retransmission.

TCP implementing the Eifel algorithm can get the same performance as TCP-I&RW for the TCP receiver with TCP timestamp option in certain scenarios.

However, the Eifel algorithm is not foolproof. An aggressive TCP receiver can manipulate TSecr field (or, the flags on the reserved bits) in an acknowledgement. Therefore, the aggressive TCP receiver can deceive the TCP sender in interpreting a non-spurious retransmission to be a spurious one. Thus the aggressive TCP receiver can enjoy an unfairly higher share of link bandwidth than it deserves.To the contrary, our proposal TCP-I&RW is robust against such ill-motivated attempts. This is because TCP-I&RW decides the value of SID randomly and detects spurious retransmissions depending on the random value. Hence, even aggressive TCP receivers cannot deceive the TCP sender in interpreting a non-spurious retransmission to be a spurious retransmission.

5. Evaluation

In this section, we evaluate TCP-I&RW with data link layer ARQ and other related flow control schemes using the ns-2 simulator [10)]. We consider a cellular network and a satellite network for the evaluation. Our performance metrics are as follows [12)].

1. Throughput (for the cellular network).
2. Bandwidth utilization (for the satellite network).

The other flow control schemes that we consider are (1) TCP-Reno without data link layer ARQ, (2) TCP-Reno with data link layer ARQ, (3) Snoop scheme (TCP-Reno based) and (4) DDA scheme (TCP-Reno based).

The TCP sender is assumed to be performing a bulk data transfer. Each TCP data segment contains 1,000 bytes, while each TCP acknowledgement contains 40 bytes. We use NAK-based selective repeat [11)] as data link layer ARQ on the wireless link. Each link level NAK contains 16 bytes. The wireless link has a priority scheduling queue holding at most 50 segments. That is, each TCP data segment retransmitted by the wireless link and each data link layer NAK have priority to be sent before other segments in the queue. TCP acknowledgements and data link layer NAKs are assumed not to be lost due to link error, as these are sufficiently small. The time d of DDA scheme is statically set twice the propagation delay (i.e., round trip time) on the wireless link.

5.1 Cellular Networks

We measure throughput of a flow on the network with the wireless link which is available exclusively for the flow. We evaluate on the system as the same as shown in figreffig:system. Generally a cellular network has this form. Segment losses due to congestion in the wired network are assumed. The propagation delay on the path in the wired network is set 10 ms (both up and down). The bandwidth and the propagation delay on the wireless link are set 2 Mbps and 50 ms respectively (both up and down). We measure the average throughput of a flow which is generated during 120 seconds (about 1,000 round trip times) by the TCP sender in this system. The data segments get lost due to link errors with a probability $P_{link-error}$ varying from 0% to 10% at 1% intervals. We observe cases when the data segments are lost due to congestion in the wired network with a probability $P_{congestion}$.

Figures 5, **6** and **7** show the results when the probability of the segment loss due to congestion in the wired network is 0%, 1% and 2% respectively. We can observe TCP-I&RW with data link layer ARQ realizes the best throughput independently of the probability of the segment loss due to congestion and due to link errors.

5.2 Satellite Networks

We measure bandwidth utilization on the wireless link shared by many flows. We evaluate on the system shown in **Fig. 8**. In this system, there is an earth station between 20 TCP sender hosts and one TCP receiver host (20 TCP receivers are in this host). Each host with each

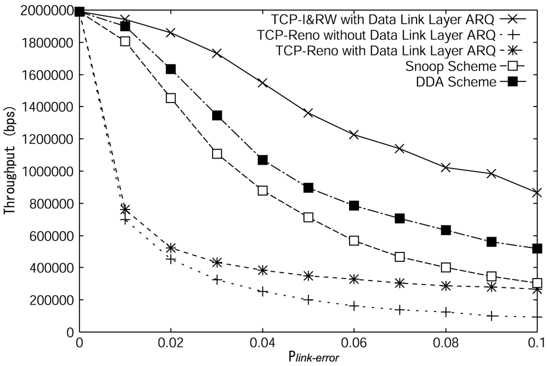

Fig. 5 TCP throughput in the cellular network ($P_{congestion} = 0$).

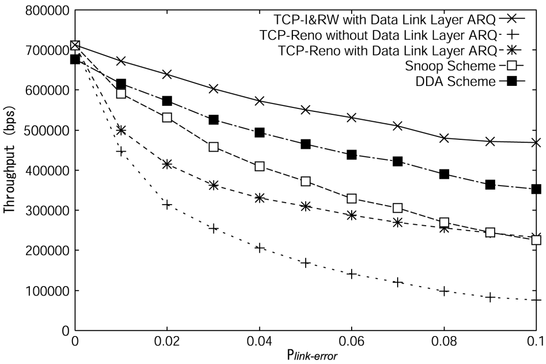

Fig. 6 TCP throughput in the cellular network ($P_{congestion} = 0.01$).

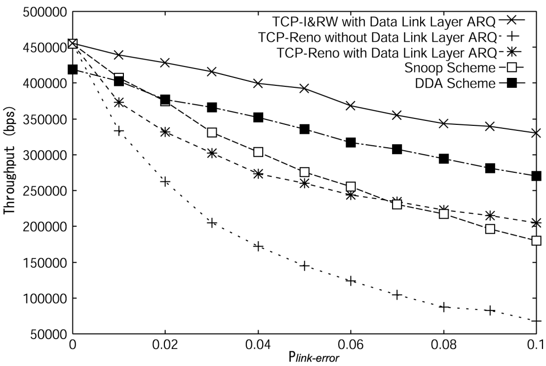

Fig. 7 TCP throughput in the cellular network ($P_{congestion} = 0.02$).

TCP sender is connected to the earth station by wired links. The earth station and the host with the TCP receivers form the wireless link with the satellite. We assume this satellite is a Geosynchronous Earth Orbit (GEO) Satellite. The bandwidth and the propagation delay of the wired link are 50 Mbps and 10 ms respectively (both up and down). The bandwidth and the propagation delay of the satellite link are

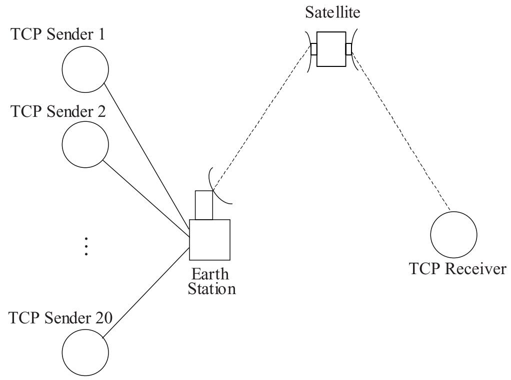

Fig. 8 Simulation scenario for the satellite network.

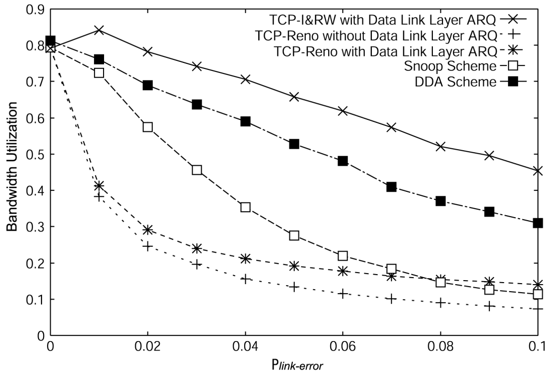

Fig. 9 Bandwidth utilization on the satellite link.

10 Mbps and 225 ms [17] respectively (both up and down). We measure bandwidth utilization of 20 flows generated during 570 seconds (about 1,000 round trip times) in this system where the data segments get lost due to link error with a probability $P_{link-error}$ varied from 0% to 10% at 1% intervals.

Bandwidth utilization is defined as the ratio of the total throughput of all flows sharing the link to the link bandwidth. Simulation results obtained for bandwidth utilization on the satellite link is shown in **Fig. 9**. This shows that TCP-I&RW with data link layer ARQ can realize the best bandwidth utilization.

6. Aggressive TCP Receiver

In this section we discuss the inherent disasterous situation that might occur had TCP-I&RW not supported its unique TCP identification mechanism.

After a retransmission in response to a segment loss, TCP-I&RW congestion window and slow start threshold depend on the identification reply in the corresponding acknowledge-

Table 1 Bandwidth utilization on the satellite link with one aggressive flow.

	Bandwidth Utilization
total (one aggressive flow and 19 normal flows)	0.819
one aggressive flow	0.103
one normal flow (average)	0.038

ment. When this value represents the identification of the originally transmitted data segment, rather than a retransmitted one, congestion window and slow start threshold revive back to the values they had before the first time retransmission.

If the conditions for *window revival* (i.e., the values of TCP identifications on the original data segments) could be guessed at the TCP receiver host, it would be possible to develop an aggressive TCP receiver with ability to fool the TCP sender during segment losses due to congestion. In such a case, while sending the acknowledgement of a segment which had been lost due to congestion, the TCP receiver might attach the value of TCP identification on the original data segment which was not actually received by the TCP receiver. Thus, an aggressive TCP receiver could make the TCP sender restore the values of congestion window and slow start threshold even after a true congestion to occupy an unfairly high share.

As an example, we show a case where an aggressive TCP receiver is allowed to occupy an unfairly high share. It is assumed that the values of the identifications are guessable (for example, the case where the values of SIDs are following a rule and the TCP receivers know the rule). We mesured (1) the total bandwidth utilization, (2) the bandwidth utilization occupied by the TCP-I&RW flow whose receiver is the aggressive one and (3) the average bandwidth utilization occupied by one normal (non-aggressive) TCP-I&RW on the satellite link in the satellite network of Section 5. Here one TCP receiver (among 20) is the aggressive one. We set 0% as the probability of a segment loss due to link errors on the satellite link and 64k bytes to the value of advertised window of the aggressive TCP flow. A 0% link error implies all segment losses are due to congestion. **Table 1** shows the result.

In this simulation one aggressive TCP flow occupies about one eighth part of the total bandwidth utilization, though 20 flows share the satellite link. That is, the aggressive TCP flow is yielding an unfairly high throughput.

For the TCP sender implementing the Eifel algorithm, it is possible to develop aggressive TCP receivers manipulating the value of the TSecr field (or, the flags on the reserved bits) to occupy an unfairly high share of link bandwidth.

7. Conclusion

TCP performance over wireless links suffer severely because of misinterpretation of segment losses due to link errors. Proposals that count on retransmission support over wireless links are promising to overcome this problem. These solutions, e.g., Snoop scheme or DDA scheme, however suffer either from implementation problems or from performance drawbacks or both. In this paper we therefore propose a new variant of TCP we name as TCP-I&RW. This variant of TCP with data link layer ARQ effectively differentiates between segment losses due to congestion and those due to link error. That is, our scheme can detect spurious retransmission on wireless links. In addition to providing improved performance than these existing most effective schemes, TCP-I&RW is free from implementation problems. Experiments show that TCP-I&RW with data link layer ARQ can get higher performance than the existing schemes in the cellular network and in the satellite networks. Again, TCP implementing the Eifel algorithm that somewhat resembles TCP-I&RW is vulnerable to aggressive TCP receiver malpractices. To the contrary, our proposal incorporates a mechanism to nullify any such attempt by an aggressive TCP receiver.

References

1) Bagal, P., Kalyanaraman, S. and Packer, B.: Comparative Study of RED, ECN and TCP Rate Control, Technical Report, Department of ECSE, Rensselaer Polytechnic Institute, Troy NY 12180-3590, USA (1999).
2) Balakrishnan, H., Padmanabhan, V., Seshan, S. and Katz, R.: A comparison of mechanisms for improving TCP performance over wireless links, *IEEE/ACM Trans. Networking*, Vol.5, No.6, pp.756–769 (1997).
3) Vaidya, N., Mehta, M., Perkins, C. and Mon-

tenegro, G.: Delayed Duplicate Acknowledgements: A TCP-Unaware Approach to Improve Performance of TCP over Wireless, *Wireless Communications and Mobile computing*, Vol.2, pp.59–70 (2002).
4) Bakre, A. and Badrinath, B.: I-TCP: Indirect TCP for Mobile Hosts, *Proc. 15th International Conference on Distributed Computing Systems* (*ICDCS*) (1995).
5) Stevens, W.R.: *TCP/IP Illustrated*, Addison-Wesley (1994).
6) Jacobson, V.: Congestion avoidance and control, *Proc. ACM Special Interest Group on Data Communications* (*SIGCOMM*) (1988).
7) DeSunibem, A., Chuah, N. and Yue, O.: Throughput Performance of Transport-Layer Protocols over Wireless LANs, *Proc. IEEE Global Telecommunications Conference '93* (*Globecom '93*) (1993).
8) Balakrishnan, H., Seshan, S. and Katz, R.: Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks, *ACM Wireless Networks*, Vol.1, No.4, pp.469–482 (1995).
9) Jacobson, V., Braden, R. and Borman, D.: TCP Extensions for High Performance, *IETF RFC 1323* (1992).
10) Fall, K. and Varadhan, K.: ns Notes and documentation, Technical Report, the VINT Project (UC Berkeley, LBL, USC/ISI, and Xerox PARC) (2002).
11) *Data service options for wideband spread spectrum systems: Radio link protocol*, TIA/EIA/IS-707 (1998).
12) Jain, R.: *The Art of Computer System Performance Analysis*, Wiley (1991).
13) Ratnam, K. and Matta, I.: WTCP: An Efficient Mechanism for Improving TCP Performance over Wireless Links, *Proc. IEEE Symposium on Computers and Communications '98* (*ISCC'98*) (1998).
14) Durst, R., Miller, G. and Travis, E.: TCP extensions for space communications, *Proc. ACM International Conference on Mobile Computing and Networking '96* (*MOBICOM'96*) (1996).
15) Brown, K. and Singh, S.: M-TCP: TCP for Mobile Cellular Networks, *ACM Computer Communications Review*, Vol.27, No.5 (1997).
16) Balakrishnan, H. and Katz, R.H.: Explicit Loss Notification and Wireless Web Performance, *Proc. IEEE Global Internet Mini-Conference* (1998).
17) Akyildiz, I.F., Morabito, G. and Palazzo, S.: TCP-Peach: A New Congestion Control Scheme for Satellite IP Networks, *IEEE/ACM Trans. Networking*, Vol.9, No.3, pp.307–321 (2001).
18) Utsumi, S., Zabir, S.M.S. and Shiratori, N.: An Efficient Approach to Improve TCP Performance over Wireless networks, *Lecture Notes in Computer Science* (*LNCS*) *2510*, pp.916–925, Springer (2002).
19) Ludwig, R.: A Case for Flow-Adaptive Wireless Links, *Technical Report UCB/CSD-99-1053, University of California at Berkeley* (1999).
20) Ludwig, R. and Katz, R.H.: The Eifel Algorithm: Making TCP Robust Against Spurious Retransmissions, *ACM Computer Communication Review*, Vol.30, No.1 (2000).

(Received March 20, 2003)
(Accepted January 6, 2004)

Editor's Recommendation

The authors propose a new TCP congestion window control mechanism which is effective at the occurrence of packet loss in wireless network, and evaluates its performance advantage by simulation. Proposed method is very useful and practical in the wireless network environment. This method is able to be adopted to next-generation wireless networks.

(Chairman of SIGMBL, Osamu Takahashi)

Satoshi Utsumi received his Ms. degree in 2003 from Tohoku University. He was employed at Advantest Corporation in 2003. His research interests include wireless networking and transport layer protocols.

Salahuddin Muhammad Salim Zabir joined the Department of Computer Science and Engineering of Bangladesh University of Engineering and Technology as a Lecturer in 1995 and later got promoted as Assistant Professor in 1997. At present, he is with RIEC, Tohoku University, Japan. He received a Best Paper Award in SCI, 2001. He is a member of IEEE, BCS and BAAS.

Gen Kitagata received his Dr.Eng. degree in information engineering from Tohoku University, Japan in 2002. Currently, he is a research associate of Research Institute of Electrical Communication of Tohoku University. His research interests include agent-based computing and network middleware design. He is a member of IEICE.

Norio Shiratori after receiving his doctoral degree at Tohoku University, joined the Research Institute of Electrical Communication (RIEC) where he is now a Professor. He has been engaged in research on distributed processing systems and flexible intelligent networks. Professor Shiratori also contributes to the field through assuming Editorial responsibilities of different journals like the Journal of High Speed Networks (1993–1997), Journal of Information Science and Engineering (1999–2001), Journal of Communication and Networks (1999–present) and Guest Editorship of the Special Issue on Recent Advances in Communication and Internet Technology of Telecommunication Systems journal (2002–2003). He has strong association with organizing various conferences in different capacities including General Chair of FORTE/PSTV'97, ICOIN-12, Program Chair of ICNP-95, ICPP-99 etc. He received IPSJ Memorial Prize Winning Paper Award in 1985, Telecommunications Advancement Foundation Incorporation Award in 1991, the Best Paper Award of ICOIN-9 in 1994, the IPSJ Best Paper Award in 1997, etc. Professor Shiratori is a Fellow of the IEEE, IPSJ and IEICE.