Performance Improvement of TCP on a Wireless Ad Hoc Network

Masashi Sugano
Osaka Prefecture College of Health Sciences
3-7-30, Habikino, Habikino, Osaka 583-8555, Japan
Tel: +81-729-50-2111, Fax: +81-729-502131
E-mail: sugano@osaka-hsu.ac.jp

Masayuki Murata
Cybermedia Center, Osaka University
1-3, Machikaneyama, Toyonaka, Osaka 560-8531, Japan
Tel: +81-6-6850-6588, Fax: +81-6-6850-6589
E-mail: murata@cmc.osaka-u.ac.jp

***Abstract*— We describe a new technique for improving TCP performance in an ad hoc network that uses a table-driven type of routing protocol paying attention to short-duration link failure. We took as the object of research a Flexible Radio Network (FRN) which is an ad hoc network for data collection. We also evaluated the case in which the effect of the collision of a data packet and an ACK packet is suppressed by Delayed ACK and resending the ACK packet preferentially. We showed through simulation the combination of these improvements can increase TCP throughput about 20%.**

I. Introduction

Recent years have seen an expanded use of ad hoc networks, the interconnection of ad hoc and wired networks, and an increasing demand for ad hoc networks offering services similar to those offered by wired networks. The TCP (Transmission Control Protocol) should be used in ad hoc networks because it is used as a transport layer protocol in the wired network [1-4]. An ad hoc network, however, is multi-hop network composed of radio links, and the transmission quality of a radio link is more unstable than that of a wired circuit. Packet loss thus occurs frequently in an ad hoc network, and the consequent connection failure results in a severe deterioration of TCP performance.

When link failure blocks communication, the routing protocol usually restores communication by changing the route of packets through links. That is, in an ad hoc network with a table-driven type of routing protocol, the routing table is updated by exchanging control information at a fixed interval. An available route, however, cannot be acquired when the link failure time is shorter than the updating interval. Coping with such brief link failures by simply renewing the table sufficiently frequently increases the load of control packets and thus degrades efficiency.

Furthermore, the TCP needs to return an ACK packet from a destination node to the data packet from a source node. Since all the terminals in an ad hoc network share the same radio channel, TCP performance also deteriorates because of the collision of data packets and ACK packets in the radio channel.

In this paper we propose a technique for suppressing degradation of the TCP performance by short-time link failure resulting from noise or an obstacle, and we report the results of a simulation evaluating this technique. The object of our research is a Flexible Radio Network (FRN) [5, 6], an ad hoc network for data collection, modified to use the proposed technique to maintain two or more routes for information. We also report our evaluation of the case in which the performance deterioration due to the collision of a data packet and an ACK packet is suppressed by delaying the ACK packet at the time of a packet collision and then resending it preferentially.

The rest of this paper is organized as follows. Section II describes the outline of the modified FRN, Section III explains the proposed technique for dealing with short-time link failure, Section IV shows the results of our evaluation by simulation, and Section V presents our conclusions.

II. System Description of FRN

A radio channel in a is divided into fixed-length time slots, and a collision is avoided by performing a carrier sense at the beginning of the time slot. Moreover, to perform the receipt check for a packet sent between nodes, relay transmission to the next node is used [7]. Since the destination node of a packet does not send a relay transmission to any other nodes, it simply returns an ACK packet to the previous node. Packet relay and ACK transmission are performed by the following slot which received the packet. When packet transmission fails because of a transmission error or collision, resending is performed at random at intervals of 3 to 5 slots in order to avoid a re-collision. Figure 1 shows an example of the packet transmission process in a FRN. Figure 1 (a) shows a case in which the packet transmission and acknowledgment at node A succeed. Node B receives the packet from node A and relays it to another node. At the same slot, node A receives this relayed packet because node A is within the range of radio transmission from node B. This acknowledgment is called an echo. If the echo is received by node A, node A can delete the packet from its buffer. The case in which the first transmission from node A fails is shown in Figure 1 (b). Node A detects a failure because it receives no echo from node B in this case, and node A retransmits the packet after three slots. Figure 1 (c)

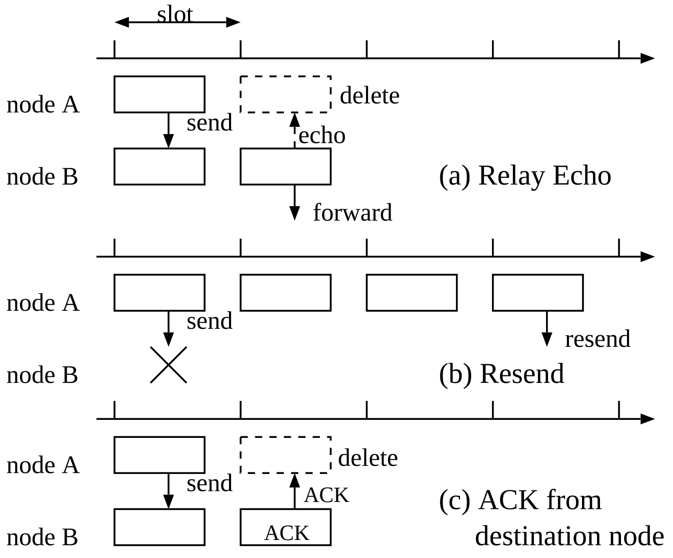

Fig. 1. Packet transmission timing in a FRN

shows a case in which node B is the destination node of the packet and node B returns an ACK to node A.

For every packet a maximum lifetime–that is, a maximum time for the packet to be allowed to exist in the network–is predefined by slots and is set at the source node. The lifetime is decreased by one for every timeslot even if the packet stays in a buffer. When the value reaches zero, the packet is discarded because its lifetime is exceeded. In other words, a packet is retransmitted repeatedly as long as a relay echo is not received within its lifetime. In the original FRN system the lifetime value long enough for the network scale is defined by users. However, the maximum lifetime is an important configuration parameter because a short lifetime makes it possible for long-living packets to be from the network effectively whereas a long lifetime gives packets more chances to try another route to the destination.

Moreover, a FRN uses a table-driven type routing protocol, and its routing table is updated by transmitting a configuration control packet for every fixed cycle. Every node maintains multiple route information for each destination node in the routing table. Consequently, bypassing quickly to another route is possible even when transmission in all the shortest routes fails. Figure 2 shows an example of the route classification in a FRN. When node A is a source node and node D is a destination node, the shortest route between them has two hops and the routes through node B and C are forward routes. The path through node I has three hops, so this route is classified as a sideward route. There is a detour, a backward route, to node D that passes through node E. The backward route ensures the reliability by providing a path when forward and sideward routes become temporarily unreliable because of the failure of a link or a node. At the source node a checkpoint on a backward route is defined for each destination in order

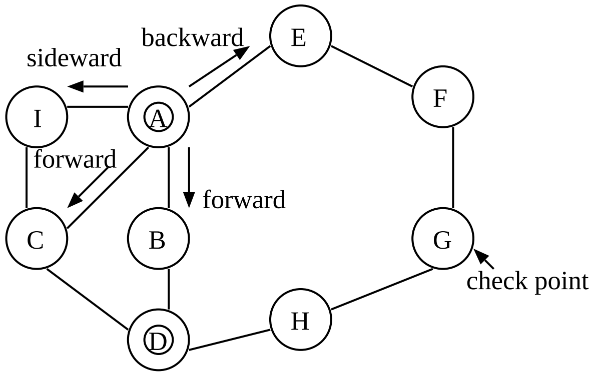

Fig. 2. Route classification

to avoid backtracking. When the checkpoint is defined in the packet header, nodes must first transmit the packet to the checkpoint node. When node A transmits a packet through node E, node G is defined as the checkpoint. If not, node E will relay the packet to node A because the shortest route to node D is through node A. We no not describe the details of the checkpoint detection method in this paper.

III. Performance Improvement of TCP

A. *Detection of link failure*

Explicit Link Failure Notification (ELFN) [3, 4] is proposed as a technique for preventing TCP performance deterioration due to the route change caused by the movement of a terminal. In ELFN, when a certain node in the TCP connection detects the failure of a link, a message called an Explicit Route Disconnection Notification (ERDN) message is sent to the source node in order to stop the transmission of a packet via that link. When the node that detected the link failure discovers a new route, it sends an Explicit Route Successful Notification (ERSN) message to the source node and transmission is resumed. The procedure of control of TCP transmission by ELFN is shown in Figure 3.

Even if a terminal does not move, a radio link can become unavailable because of a change of the radio environment or because of the movement of an obstacle. By detecting link failures whose duration is shorter than the length of the routing table updating cycle, the TCP can use another route to bypass the failed link and performance degradation can be suppressed. Transmission failure, however, can also be the result of packet collision, and when a transmitting error is made continuously several times it can appear to be a link failure. Then, the counter for every adjacent node is prepared in each node and the number of times of transmitting failure is recorded. When neither an echo nor ACK can be received, the value of the counter is incremented one. When the value of a counter exceeds a threshold, it is judged as link failure.

If there is another route that can be used when link failure is detected, the failure can be bypassed immediately. When

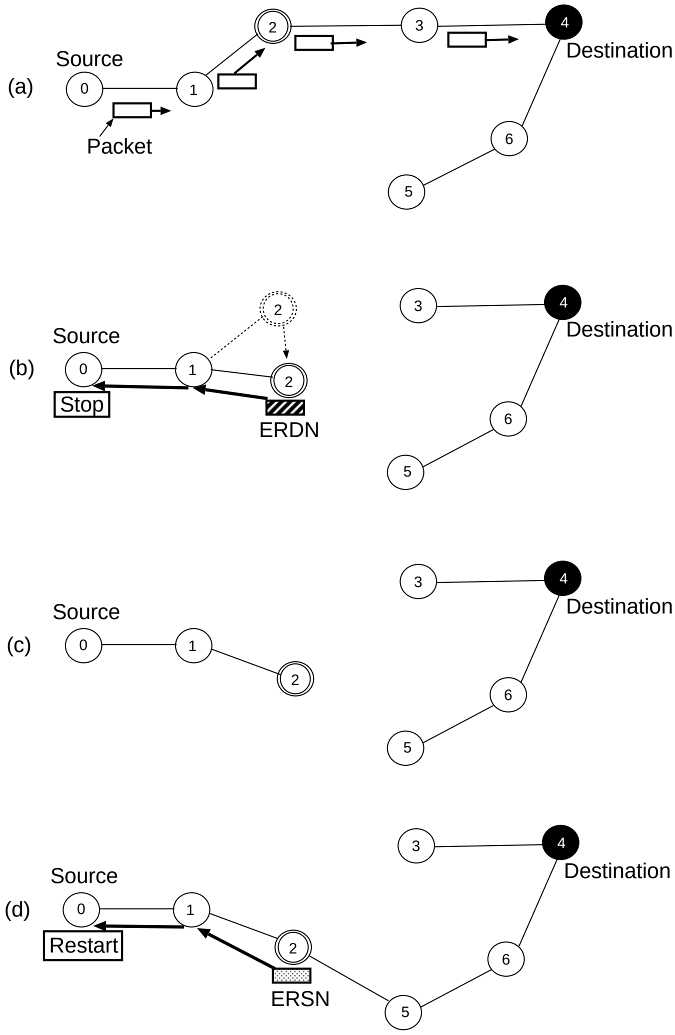

Fig. 3. Control of TCP transmission by ELFN

there is no alternate route that can be used, the previous node is notified of link failure and searches for another route from there. Thus, the transmitted packet goes back toward a source node until an available route is found. The case in which the connection is set up from Node 1 to the Node 6 as shown in Fig. 4 (a) is assumed. When Node 2 detects the failure of the link between itself and Node 3 as shown in Fig. 4 (b), it transmits an ERDN message to Node 1 because does not have any detour routes. When Node 1 then finds a detour route, it transmits packet to the new node (Node 7). When a detour is not found, an ERDN message will also be sent toward the source node.

At suitable intervals a probe packet for detecting the return of a link is transmitted to the node once judged to be a failed link. If the return of a link is detected, the use of a route through that link will be enabled again. Moreover, when a data packet from the node judged to be a failed link is received, the link shall be judged to be restored.

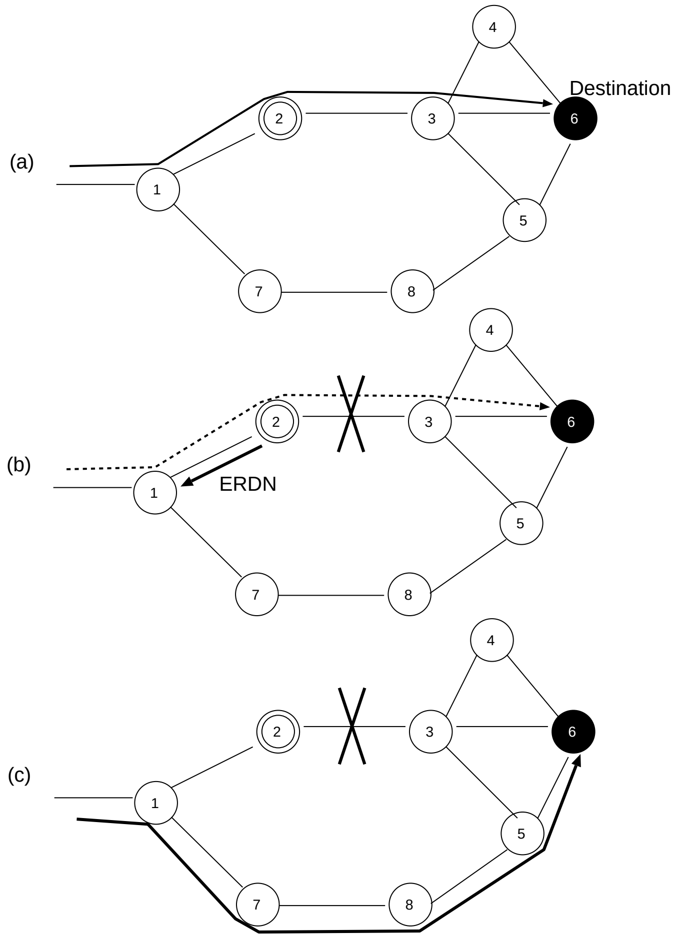

Fig. 4. Detection of the detour by ERDN

B. Control of the collision by Delayed ACK

Because all packets in a FRN may share the same radio channel, the collision of data and ACK packets can severely reduce TCP throughput. The Delayed ACK option reduces the waste of network resources by transmitting an ACK segment collectively citeBO:Stevens. That is, an ACK for a collection of data packets can be returned by delaying ACK for a fixed time after the reception a data packet (See Fig. 5 (b)). The number of ACK packets can thus be reduced, suppressing packet collision and increasing throughput.

In FRN, in order that all packets may share the same radio channel, the collision of data and ACK occurs and the throughput of TCP deteriorates remarkably. There is the Delayed ACK option aiming at holding down waste of network resources in TCP by transmitting an ACK segment collectively [8]. That is, it collects to the reception to a data packet in the meantime by delaying a fixed time reply, after receiving a data packet, and ACK is returned. By this, the number of ACK packets can be reduced, the collision of a packet is suppressed, and improvement in a throughput can be expected.

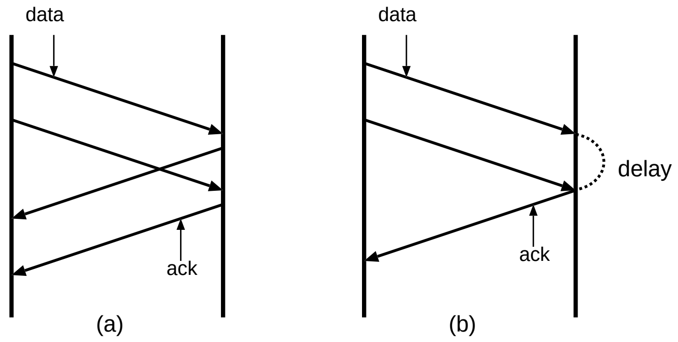

Fig. 5. ACK reduction by delayed ACK

C. Resending ACK preferentially

When resending occurs because of packet collision, repeated collision can be avoided and an ACK transmitted preferentially by sending assigning the ACK resending interval shorter than the data packet resending interval. This can be expected to increase TCP throughput.

IV. Performance Evaluation by Simulation

We evaluated the effect of the proposed technique by simulation. We assumed that link failures are generated according to the Poisson distribution that and the duration of link failure has an exponential distribution. Four FTP connections were given as input traffic on the network (which consists of 20 nodes as shown in Fig. 6), using ns-2 [9] as the simulation environment. Moreover, the version of TCP we used was Tahoe. We used TCP throughput as a performance measure and defined it as the average number of packets transmitted to their destinations successfully per time slot.

First we investigated the effect of the technique in the case of changing a route when link failure was detected. The average rate of the occurrence of link failure was set at 0.01 times per second, and the average duration of failures

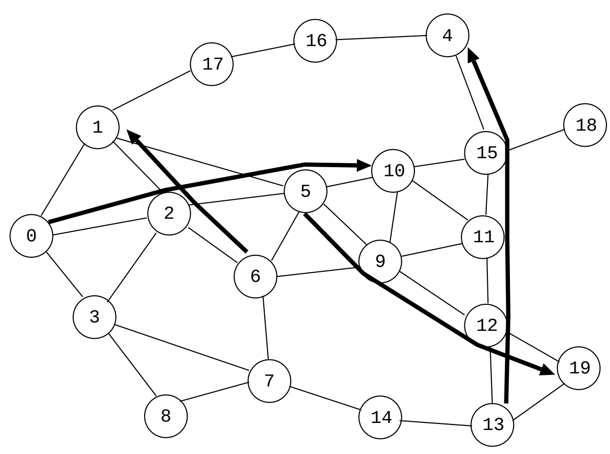

Fig. 6. Simulation model

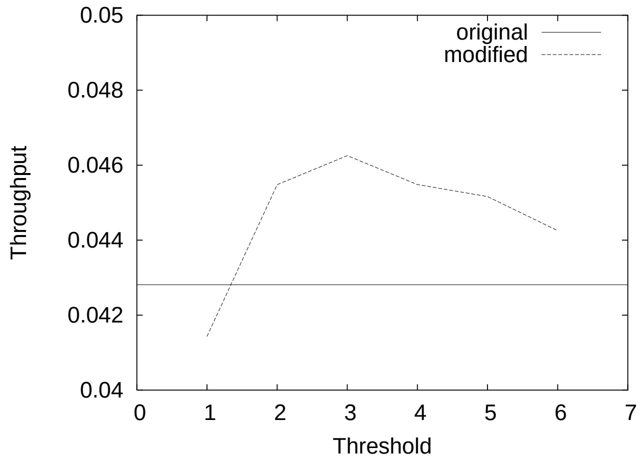

Fig. 7. Influence of the threshold value for judging link failure

was set at 10 seconds. The TCP throughput is plotted in Fig. 7 against the threshold value for the number of times transmission failure must occur in order to be considered to indicate link failure. The throughput of the original system is shown by the solid horizontal line labeled ”original,” and the throughput at the time of detecting link failure is shown by dotted line labeled ”modified.” It is clear that it is most effective to consider link failure to be indicated by three transmission failures.

Next, we investigated in what range it would be effective about the duration of link failure of a proposal technique. The rate of occurrence of link failure is fixed to 0.01, and the result to which average duration was changed from 2.5 seconds till 20 seconds is shown in Figure8. It is higher for a throughput to set the threshold as two in the range with the short term of link failure, and to perform route change quickly. When the term of link failure is long on the other hand, it became clear that it is better to set the threshold to three.

The result at the time of using Delayed ACK and priority resending of ACK is shown in Fig. 9. As a resending interval, it set up with 5 or 6 slots to the data packet, and set up with 3 or 4 slots to ACK. The threshold value for indicating link failure was set to 3, and the rate of generating of link failure was changed. This result figure shows that using Delayed ACK and priority resending of ACK can increase TCP throughput by about 20%.

V. Conclusion

In this paper we proposed a technique for improving TCP performance in an ad hoc network that uses a table-driven type of routing protocol paying attention to link failure of short period of time. We also clarified the effects of delaying the ACK is applied and resending the ACK preferentially.

The TCP throughput in an ad hoc radio network still needs to be increased, and we think this can be done by better detecting link failure. This will require consideration

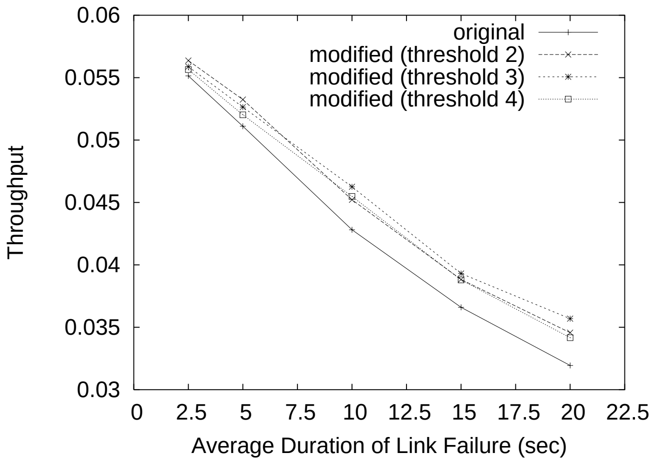

Fig. 8. Influence of the duration of link failure

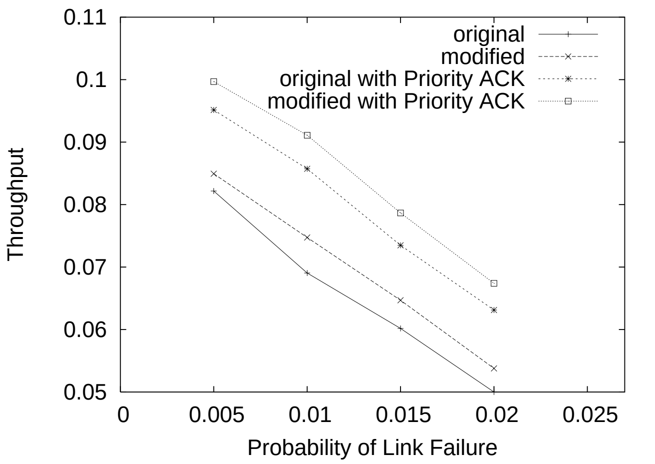

Fig. 9. Throughput increase due to the use of all the proposal techniques

of not only the number of transmission failures but also their duration. Furthermore, we think techniques for avoiding the collision of data packets and ACK messages will also be effective in a data link protocol or a routing protocol. We are therefore going to examine such techniques and evaluate systems using them.

Acknowledgments

This work was partly supported by the Special Coordination Funds for promoting Science and Technology of the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

[1] A. Ahuja, S. Agarwal, J. P. Singh, and R. Shorey, "Performance of TCP over Different Routing Protocols in Mobile Ad-Hoc Networks," in *IEEE VTC2000 Spring*, May 2000.

[2] G. Holland and N. H. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks," in *ACM/IEEE MOBICOM'99*, Aug. 1999, pp. 219–230.

[3] D. Kim, C. K. Toh, and Y. Choi, "TCP-BuS: Improving TCP Performance in Wireless Ad Hoc Networks," in *IEEE ICC2000*, June 2000.

[4] T. Goff, J. Moronski, and D. S. Phatak, "Freeze-TCP —A True End-to-End TCP Enhancement Mechanism for Mobile Environments," in *IEEE INFOCOM2000*, Mar. 2000.

[5] "Flexible Radio Network," available at http://www.fujielectric.co.jp/denki/p26/ecop˙contents2.html.

[6] M. Sugano, T. Araki, M. Murata, T. Hatauchi, and Y. Hosooka, "Performance Evaluation of a Wireless Ad Hoc Network: Flexible Radio Network (FRN)," in *IEEE ICPWC2000*, Dec. 2000, pp. 350–354.

[7] D. Bertsekas and R. Gallager, *DATA NETWORKS 2nd Edition*. Prentice-Hall.

[8] W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*. Addison-Wesley, 1994.

[9] "The Network Simulator - ns-2," available at http://www.isi.edu/nsnam/ns/.