

# DODR: Delay On-Demand Routing

Dong Wan-li, Gu Nai-jie, Tu Kun, Bi Kun and Liu Gang

**Abstract**—As originally designed for wired networks, TCP (transmission control protocol) congestion control mechanism is triggered into action when packet loss is detected. This implicit assumption for packet loss mostly due to network congestion does not work well in Mobile Ad Hoc Network, where there is a comparatively high likelihood of packet loss due to channel errors and node mobility etc. Such non-congestion packet loss, when dealt with by congestion control mechanism, causes poor TCP performance in MANET. In this study, we continue to investigate the impact of the interaction between transport protocols and on-demand routing protocols on the performance and stability of 802.11 multihop networks. We evaluate the important wireless networking events caused routing change, and propose a cross layer method to delay the unnecessary routing changes, only need to add a sensitivity parameter  $\alpha$ , which represents the on-demand routing's reaction to link failure of MAC layer. Our proposal is applicable to the plain 802.11 networking environment, the simulation results that this method can remarkably improve the stability and performance of TCP without any modification on TCP and MAC protocol.

**Keywords**—Mobile ad hoc networks (MANET), on-demand routing, performance, transmission control protocol (TCP).

## I. INTRODUCTION

A significant amount of today's Internet traffic, including WWW (HTTP), file transfer (FTP), email (SMTP), and remote access (Telnet) traffic, is carried by the TCP transport protocol [1]. TCP together with UDP form the very core of today's Internet transport layer. A mobile ad hoc network (MANET) is formed by a group of mobile nodes connected by wireless links [2]. Started as a "toy" problem in 1979 in the Naval Research Laboratory, ad hoc networking has since matured into a well-defined research area [3],[4]. Now, multihop ad hoc wireless networks are one of the most commonly found in many applications of sensor networks,

Manuscript received November 13, 2005.

Dong Wan-li is with the Department of Computer science and technology, University of Science and Technology of China, Hefei, Anhui, P.R.China (phone: 86-551-3601547; e-mail: danlin@mail.ustc.edu.cn).

Gu Nai-jie is with the Department of Computer science and technology, University of Science and Technology of China (e-mail: gunj@ustc.edu.cn).

Tu Kun is with the Department of Computer science and technology, University of Science and Technology of China (e-mail: tukun@ustc.edu).

Bi Kun is with the Department of Computer science and technology, University of Science and Technology of China (e-mail: bikun@mail.ustc.edu.cn).

Liu Gang is with the Department of Computer science and technology, University of Science and Technology of China (e-mail: liugang@mail.ustc.edu).

mesh networks, and home/office networks. But, TCP protocol generally suffers from poor bandwidth utilization and network unfairness problems over 802.11 multihop networks. It is primarily because the wireless ad hoc networking environments have many different unique features, such as: multihop shared channel, node mobility, channel noise and location-dependence etc. However, most wireless applications depend on legacy TCP for communicating with TCP-dominant wired hosts, and it is likely that TCP will remain as major transport protocol for the clients of 802.11 networks. Thus, it is important to understand the behavior of legacy TCP and improve its performance in the 802.11 networking environment.

Dynamic Source Routing protocol (DSR) [5], [6] and Ad Hoc On-Demand Distance Vector protocol (AODV) [7],[8] are two prominent on-demand routing protocols for mobile ad hoc networks. They are both the most commonly used in the design and evaluation of mobile ad hoc networks. This reactive nature of these protocols is a significant departure from more traditional proactive protocols [9], which find routes between all source-destination pairs regardless of the use or need of such routes. The key motivation behind the design of on-demand protocols is the reduction of the routing load. High routing load usually has a significant performance impact in low bandwidth wireless links. If the route is believed to have failed (through loss detection) a route maintenance/rediscovery is triggered. In general, ad-hoc routing protocols in 802.11 networks do not distinguish between MAC contention loss, channel errors, mobility-induced losses or other. It is primarily because 802.11MAC, which is the only interface of the routing protocols to the underlying networks, has only limited intelligence about the networking events and conditions.

Basically, in order to better improving channel utilization, TCP window mechanism make a TCP sender sends data at a higher rate beyond network capacity, eventually create more signal interference in ad hoc wireless networking environment. So, higher spatial density of packets in an area leads to higher chance of signal interference and collision in wireless medium. As is pointed out in [11], [12], 802.11MAC however cannot perfectly handle signal interference of general multihop topologies. TCP's pushing more packets beyond a certain limit drives excessive link layer retransmission and eventually leads to MAC contention loss instead of buffer flow loss. Hence, on-demand routing protocols in 802.11 networks keep responding to any MAC contention loss and performs unnecessary routing maintenance over 802.11 multihop networks even in the static topologies without channel noise[12], [13].

Our discovery also sheds some light on how to improve TCP performance over multihop wireless networks. In this paper, we propose a cross layer technique to improve TCP throughput and stability: add a sensitivity parameter  $\alpha$ , which means on-demand routing protocols' reaction to link loss of MAC layer. This simple method leads to a 75%-100% increase in throughput compared with standard TCP over general on-demand routing protocol. Our proposal is applicable to the plain 802.11 networking environment, the simulation results that this method can remarkably improve the stability and throughput of TCP without any modification on TCP and MAC protocol.

The rest of this paper is organized as follows: Section II compares with related work. Section III reviews link-layer contention and spatial channel reuse in an IEEE 802.11-based multihop wireless network. Then, through presents a thorough study of the events caused routing dynamics, we propose our cross layer method to improve TCP performance in Section IV. The performance of the proposed method is evaluated in 802.11 networks of various topologies via simulation in Section V. Finally, Section VI concludes this paper and proposes some future research directions.

## II. RELATED WORK

TCP over wireless networks has been an active research topic. Holland and Vaidya investigate the effect of mobility-induced link breakage of wireless ad hoc networks upon TCP performance [14]. The focus of their study is on the interaction between DSR routing dynamics and TCP window adaptation. Since most packet losses are due to node mobility, congestion control mechanisms of TCP should not be applied to such loss events. Studies in [15]–[18] mainly address the issue of congestion detection in improving TCP over mobile ad hoc networks. In particular, [15], [16], [18] use end-to-end measurements to detect whether the packet losses are due to congestion or non congestion conditions.

Recently, another effort has been made for the intelligent adaptation of the end users to wireless channel errors, routing changes, connection failures, and other wireless ad hoc networking events. TCP-ELFN [19] suggested an explicit link failure notification (ELFN) from the network to distinguish congestion from the routing change in a wireless environment. ADTCP [20] monitored the change of the end-to-end network conditions to distinguish congestion from other networking events. They are based on the idea of freezing TCP states and keeping large congestion windows without decreasing the transmission rate at the occurrence of routing changes. Meanwhile, most observation is confirmed in many recent studies [10], [11], [21], [22] by showing that TCP with a small congestion window tends to outperform TCP with a large congestion window in 802.11 multihop networks. Fu et al. [10] proposed a link layer active queue management algorithm called LRED that exploits ECN marking to stabilize TCP window. Nahm et al. [23] proposed fractional window increment (FeW) scheme for TCP to prevent the over-reaction

of the on-demand routing protocol by limiting TCP's aggressiveness. Since wireless ad hoc networks are typically stand-alone, a new rate-based approach was proposed for some transport design [24], [25]. They do not guarantee TCP compatibility because the network components across layers are closely tied with each other, exploiting side-information of the intermediary nodes.

## III. MAC CONTENTION LOSS

We consider a stationary, multihop wireless network using IEEE 802.11 distributed coordination function (DCF) [26]. A single wireless channel is shared among all nodes in the network. Only receivers within the transmission range of the sender can receive the packets. In IEEE 802.11 DCF, each packet transmission is preceded by a control handshake of RTS/CTS messages. Upon overhearing the handshake, the nodes in the neighborhood of either the sender or the receiver will defer their transmissions and yield the channel for subsequent DATA-ACK transmissions.

A hidden terminal is a sender in the neighborhood of the receiver of another ongoing transmission, but out of the transmission range of the sender. Because it may not receive the receiver's CTS due to various reasons such as collisions, a hidden terminal may disrupt the ongoing transmission by initiating another transmission. We show this situation in left part of Fig. 1. On the other hand, an exposed terminal is a potential receiver in the neighborhood of the sender of another ongoing transmission. It cannot receive or respond to another sender's RTS, we illustrate this in right of Fig. 1. According to the IEEE 802.11 protocol, a sender drops the packet after retransmitting DATA four times without receiving an ACK, typically caused by hidden terminals. Besides, a sender drops the packet after sending the RTS message seven times without receiving CTS, typically caused by exposed terminals.

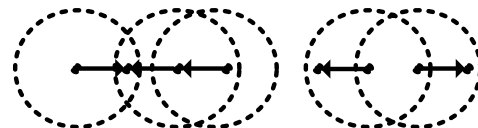


Fig. 1 Hidden/Exposed terminal problem

We illustrate the extended hidden terminal problem in Fig. 3. In Fig. 3, two adjacent nodes are 200m apart. The transmission range of a node is set to 250m, the carrier sensing range is 550m, and the interference range is 550m.

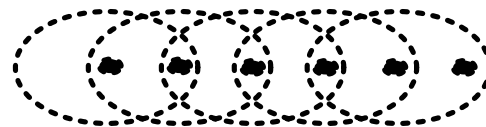


Fig. 2 Extended hidden terminal problem

In this example, node *D* is a hidden terminal of the ongoing transmission  $A \rightarrow B$ . Node *D* cannot decode *B*'s CTS since it is out of the 250m transmission range. Besides, *D* cannot

sense  $A$ 's DATA transmission since  $A$  is out of  $D$ 's 550m carrier sensing range. Therefore, node  $D$  may transmit to another node, say node  $E$ , at any time, disrupting the ongoing transmission  $A \rightarrow B$ . If the DATA transmission between  $A$  and  $B$  is corrupted four times in a row, node  $A$  will drop the packet. On the other hand, node  $C$  is an exposed terminal since it is within the 550 carrier sensing range of the transmitting node  $E$ . Node  $C$  cannot respond to the RTS message from another node, say  $B$ . After seven unsuccessful RTS retries, node  $B$  will drop the packet. The location-dependence of contention also allows for spatial channel reuse in a multihop wireless network. Specifically, any two transmissions that are not interfering with each other can be scheduled simultaneously. In Fig. 3,  $A \rightarrow B$  and  $E \rightarrow F$  can transmit concurrently, reusing the shared wireless channel. Spatial channel reuse can greatly improve the network throughput, especially in a large network that spreads a wide area.

#### IV. CROSS-LAYER METHOD

##### A. A Model for Congestion

According to [10], consider a generic ad hoc network setting. The  $m$  backlogged nodes in the network will use the RTS-CTS-DATA-ACK handshake of 802.11 protocols for data transmission. Let us denote that  $c(m)$  nodes are able to successfully initiate RTS request, and denote the number of nodes that are able to successfully transmit the DATA packets as  $b(m)$ . Using Markov chain models can derive each node's link drop probability  $P_r$  as follows:

$$P_r(m) = \frac{\frac{b(m)}{m}}{1 - \left(1 - \frac{b(m)}{c(m)}\right)^{r+1}} \left[1 - \frac{b(m)}{c(m)}\right]^r \quad (1)$$

Where  $r = 7$  is the maximum retry count for RTS in 802.11 MAC.

In the random topology of  $N$  nodes,  $B^*$  denotes the maximum number of nodes that can transmit their DATA packets concurrently without collision. At this value, the network achieves highest channel spatial reuse. Among  $N$  nodes,  $C^*$  denotes the maximum number of nodes that can initiate RTS messages.

**Corollary 1:** When the number of backlogged nodes  $m$  is smaller than  $B^*$ , then the link drop probability  $P_r \approx 0$ .

Since  $m < B^*$ , on average all  $m$  nodes can transmit simultaneously. Therefore,  $b(m) \approx c(m) \approx m$  in steady state. According to (1), the drop probability over each link is  $P_r \approx 0$ . This means that, as long as the network is under loaded, the link drop is negligible.

**Corollary 2:** When the network is overloaded (i.e., the number of backlogged nodes  $m$  is greater than  $B^*$ ), the link drop probability  $P_r$  increases as  $m$  increases.

We use (1) to see why the above is true. In this case, all  $m$

nodes can successfully initiate an RTS message but only  $B^*$  nodes can transmit their DATA without collisions. That is,  $b(m) \approx B^*$  but  $c(m) \approx m$ . Therefore,  $B^* < m < C^*$ . It is easy to see that  $P_r(m)$  is an increasing function of  $m$  since  $\frac{dP_r(m)}{dm} > 0$ .

This shows that link drop probability increases as the network load (as expressed by  $m$ ) further increases.

**Corollary 3:** Once network is heavily loaded in the sense that  $m > C^*$ , then the link drop probability  $P_r$  remains stable in the saturated state.

In this case, among the  $m$  nodes, only  $C^*$  out of  $c(m)$  nodes can initiate RTS, and only  $B^*$  nodes can transmit DATA packets without collision. Therefore,  $c(m) \approx C^*$  and  $b(m) \approx B^*$ . Then long term  $P_r$  remains statistically flat according to (1).

Basically, in order to better improving channel utilization, TCP window mechanism make a TCP sender sends data at a higher rate beyond network capacity, by doing this, much of packets have been pushed to this resource limited network. The result is increase the number of nodes which in the network will use the RTS-CTS-DATA-ACK handshake of 802.11 protocols for data transmission. As illustrated in corollary 2, the number of backlogged nodes  $m$  will greater than  $B^*$  eventually, and the link drop probability  $P_r$  will increases as  $m$  increases drastically. Unfortunately, as we have detailed depicted in section III, the 802.11MAC however cannot perfectly handle signal interference of general multihop topologies.

TCP's pushing more packets beyond a certain limit drives excessive link layer retransmission and eventually leads to MAC contention loss instead of buffer flow loss. Hence, on-demand routing protocols in 802.11 networks keep responding to any MAC contention loss and performs unnecessary routing maintenance over 802.11 multihop networks even in the static topologies without channel noise.

##### B. Link Failure and Stability

Except congestion cause link failure, there have many other events lead to link failure, such as: channel errors, node mobility etc. But the ad-hoc routing protocols in 802.11 networks do not distinguish between MAC contention loss, channel errors, mobility-induced losses or others. It is primarily because 802.11 MAC, which is the only interface of the routing protocols to the underlying networks, has only limited intelligence about the networking events and conditions.

As our above analyzed in section IIV and part A of IV, MAC loss with congestion is frequent and persistent during the entire course of the TCP connection. On-demand routing protocols in 802.11 networks keep responding to any link failure and performs unnecessary routing maintenance or re-discovery over 802.11 multihop networks even in the static topologies without channel noise and node ability etc. In fact, on-demand routing protocols in 802.11 networks needn't to response to link failure caused by congestion as soon as possible, especially in static topologies. So, the link failure needs to be treated

differently depending on what caused the loss.

- 1) Congestion: Need the TCP congestion mechanism to adjust the value of CWL or do some other TCP operations, and the MAC contention loss can be looked as an important sign of congestion to reduce the network overhead. But, the route should remain intact, needn't unnecessary routing maintenance or re-discovery.
- 2) Channel noise: Channel noise affects only the quality of the end to end connection, not the connectivity itself. So, noisy routes are tolerable. The noisy route should remain intact until a better alternative new route is found. Sometimes, retransmission can improve the quality of the end to end connection.
- 3) Mobility: This situation can be divided into two parts: First, the node is no longer available, the current route should be invalidated and a new route needs to be established no matter how it might cost. Second, in some random topologies, the moved nodes may be back to the original place in a few minutes, so if the routing protocols can delay dynamics, not only improve the stability of the end to end TCP connection, but also the throughput of the connection.

### C. DODR

From above link failure and stability analyzed, we conclude that the on-demand routing need to less sensitive to MAC loss in order to improve the stability and throughput of TCP connection. But the transport protocols need the MAC contention loss information to adjust the network load because of MAC contention loss is an important sign of congestion to reduce the network load.

So, this is a cross layer problem. In order to achieve this goal, we propose the delay on-demand routing cross layer method. We add a sensitivity parameter  $\alpha$ , which represents on-demand routing protocol's reaction to link failure of MAC layer. By shift the value of  $\alpha$  can adjust the actual maximum retry count for RTS in 802.11 MAC. This is according with corollary in part A of section IV. The number of successive link failure  $F$  is counted, and the on-demand routing protocol takes the routing maintenance or re-discovery action only after  $F$  exceeds a certain limit  $\alpha$ , the toleration limit of successive link failures. So, our method is as follows:

- 1) When come in to beginning link failure, compare the value of  $F$  and  $\alpha$ , if  $F < \alpha$ , the routing agent keeps the route intact, doesn't perform a routing maintenance or re-discovery operation. So, the current packet is dropped and the transmission is resumed with a next packet, the value of current  $F$  is increased by 1 meanwhile.
- 2) Otherwise,  $F = \alpha$ , it means the current route should be invalidated and a new route needs to be established no matter how it might cost. So, the routing agent needs to response to this link failure, perform a route maintenance or re-discovery operation immediately. Meanwhile, reset the value of  $F$  to 0.

- 3) Whenever the transmission is completed successfully, the route agent needs to reset the value of  $F$  to 0.

Accord to this method, we can control the route agent response to link failure immediately by adjust the value of  $F$ . As proved before, many routing maintenance and re-discovery triggered by link failure are unnecessary and affect the TCP stability and performance seriously.

The  $\alpha$ 's value of original on-demand routing is 1 obviously. If we increase the  $\alpha$ 's value, the routing protocol responds to link failure at least after  $7\alpha$  times of RTS retransmission in the 802.11 MAC layer. The result is enabling much a stronger end to end connectivity in presence of channel noise and signal interference. But we must notice that, the route agent sends a packet only once for any value of  $\alpha$ . If the packet is lost, despite the retransmission of 802.11 MAC layer, the routing agent sends out the next packet without saving the lost packet. So, the end users still experience the MAC contention loss created during the interaction between transport and 802.11 MAC layers because of such packet loss is indispensable to enable TCP properly adjust congestion window within the network capacity. So, this operation won't interference the TCP's congestion mechanism at all, the number of retransmission of the same packet by on-demand routing protocol is not changed when we specify the value of  $\alpha$ . It also means that the number of retransmission of the same packet by on-demand routing protocol is not changed according with the value of  $\alpha$ .

## V. SIMULATION

In this section, we evaluate our proposed method in NS-2(version ns-2.27) over a variety of scenarios and compare it to traditional TCP schemes in ad hoc networks. In our simulations, each node was 200m apart and several overlapping TCP connections were established between the two nodes. The data rate of the wireless channel was 2 Mbps and the radio propagation model was the two-ray ground model with transmission range 250 m, carrier sensing range 550 m, and interference range 550 m.

### A. Chain Topology

In these simulations, the topologies are as Fig. 3. We evaluate our method from 4-hop to 22-hop. In a chain topology, TCP packets travel along a chain of intermediate nodes toward the destination. The successive packets of a single connection interfere with each other as they move down the chain, resulting in link layer contention. We study the performance of the TCP flow whose source and destination are placed at both ends of the chain respectively.

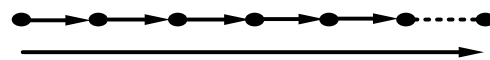


Fig. 3 n-hop chain topology

The TCP throughput results over networks of the 7-hop

chain topology under different conditions are compared in Fig. 4. This is a stationary scenario, so the TCP throughput with static routing can be used as an important reference of the performance bound. From 0 to 80s, the throughput of TCP over DSDV is around 0, because DSDV is based on the classical Bellman-Ford routing algorithm. After 80s, the DSDV can be looked as a static routing protocol. Noticeably, TCP with our method over the on-demand routing protocols achieves a throughput that is relatively close to the performance bound by TCP with static routing. As show, TCP over DODR achieves drastic improvement in throughput as compared to TCP over any other routing protocols, such as AODV, DSR or DSDV. For one flow, the TCP over DODR reaches 30%-70% increase in throughput compared with TCP over AODV, and reaches 100%-300% increase in throughput compared with TCP over DSR.

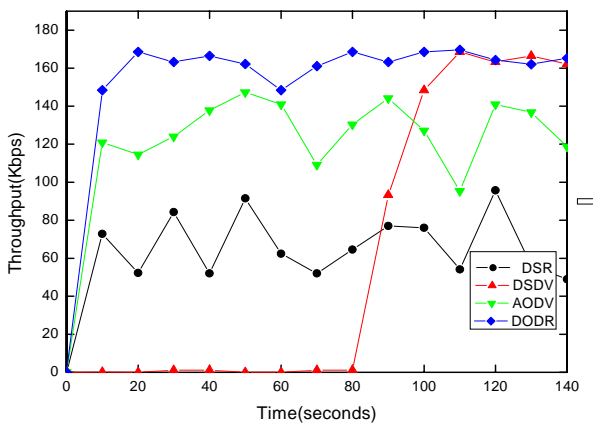


Fig. 4 7-hop chain topology comparison

Then, we compared two flows from 4-hop to 22-hop chain topology in Fig. 5. The simulation results show the similar trends of TCP throughput in which TCP throughput is noticeably improved with our proposed DODR. For 4-22 hop chain topologies, TCP over DODR achieved over 60% increase in throughput as compared with TCP over DSR or AODV.

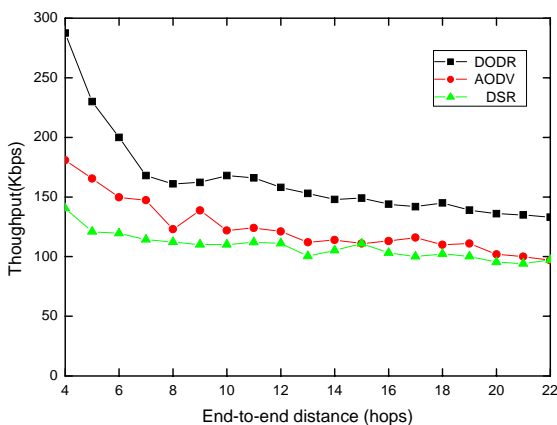


Fig. 5 4-hop to 22-hop chain topology (2 TCP flows)

*B. More Complex Topology*

We extend our study to scenarios of multiple TCP flows and more complex topologies including cross, grid. In the cross network topology shown in left part of Fig. 6, we run two TCP flows: one from node 0 to node 6 and the other from node 7 to node 12. The right part of Fig. 7 shows a 7\*7 grid topology.

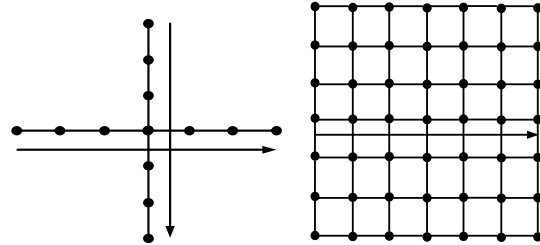


Fig. 6 Cross topology and Grid topology

In Fig. 7, the general routing protocols experienced over 100 times of route changes during a 150-second session. The instability and inefficiency of the on-demand routing protocol is noteworthy. Furthermore, the 6-hop distance route exists only for 50% of the simulation session in Fig. 7. This indicates that the shortest path was not efficiently utilized even though we didn't consider node mobility and channel errors. The network resource consumption for the alternative paths of more than 6 hops was more than needed. In the meanwhile, DODR experiences only one route change. As compared with general on-demand routing, DODR increasing one parameter makes a huge difference in routing dynamics over 802.11 multihop networks.

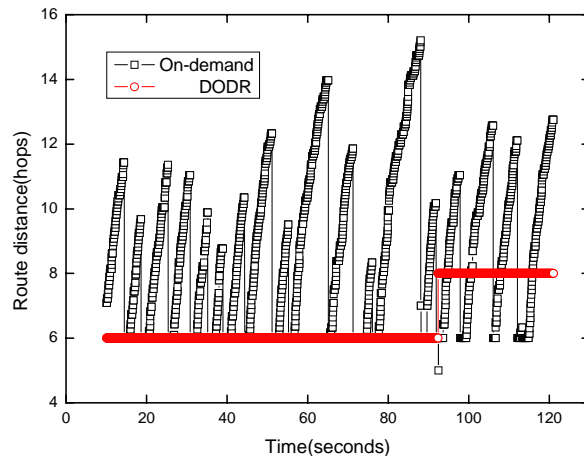


Fig.7 Routing change over time in 7\*7 grid topology

*C. Random Topology*

Finally, we also run extensive simulations in realistic mobile scenarios with random network topologies generated by the setdest tool in ns-2 distribution. We place 200 nodes uniform randomly in a rectangular area of size 1000\*2500. There are 20 TCP flows with their sources and destinations randomly chosen. With our proposed DODR, TCP performance is improved at least over 30% and the routing protocol is more

stabilized (50%-75% less routing changes).

## VI. CONCLUSION

By investigated the impact of cross layer interaction between transport protocols and on-demand routing protocols on the performance and stability of 802.11 multihop networks, we found that link loss of 802.11 networks was treated the same by routing protocols, even though it represents different types of networking events requiring different reactions of routing protocols. Another way to address this issue is to change the sensitivity to link failure. We have delay on-demand routing protocols respond only to bulk losses (of length  $\alpha$ ) instead of reacting to every single link loss. This simple policy turns out to make noticeable differences of TCP performance and routing stability in general, even though the policy may not provide a straightforward advantage to the other networking events unrelated with congestion (e.g., node mobility and noisy channel). Our proposal is applicable to the plain 802.11 networking environment, the simulation results that this method can remarkably improve the performance of TCP without any modification on TCP and MAC protocol. In the near future, we would like to continue our effort to reduce the inefficiency of the cross layer interaction in the 802.11 networking environment.

## REFERENCES

- [1] K. Thompson, G. Miller, and M. Wilder, "Wide-area internet traffic patterns and characteristics," *IEEE Network*, 11(6), November-December 1997.
- [2] (2004) Home page of IETF mobile ad-hoc networks (manet) WG. [Online]. Available: <http://www.ietf.org/html.charters/manet-charter.html>
- [3] A. Ephremides, "Ad hoc networks: not an ad hoc field anymore," *Wiley Journal of Wireless Communications and Mobile Computing*, vol. 2, pp. 441-448, 2002.
- [4] I. Chlamtac, M. Conti, and J.-N. Liu, "Mobile ad hoc networking: imperatives and challenges," *Elsevier Ad Hoc Networks Journal*, vol. 1, pp. 13-64, 2003.
- [5] Josh Broch, David Johnson, and David Maltz, "The dynamic source routing protocol for mobile ad hoc networks," <http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-01.txt>, Dec 1998. *ietf Internet Draft (work in progress)*.
- [6] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," In T. Imielinski and H. Korth, editors, *Mobile computing*, chapter 5. Kluwer Academic, 1996.
- [7] Charles Perkins and Elizabeth Royer, "Ad hoc on-demand distance vector routing," In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90-100, Feb 1999.
- [8] Charles Perkins, Elizabeth Royer, and Samir Das, "Ad hoc on demand distance vector (AODV) routing," <http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-03.txt>, June 1999. *ietf Internet Draft (work in progress)*.
- [9] Z. J. Haas and M. R. Pearlman, "The performance of query control schemes for the zone routing protocol," In *Proceedings of ACM SIGCOMM'98 Conference*, pages 167-177, Vancouver, Sept. 1998.
- [10] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on TCP throughput and loss," in Proc. *IEEE INFOCOM*, San Francisco, CA, March 2003.
- [11] K. Chen, Y. Xue, S. Shah, and K. Nahrstedt, "Understanding bandwidth-delay product in mobile ad hoc networks," Special issue on protocol engineering for wired and wireless networks, *Elsevier Computer Communications*, vol. 27, pp. 923-934, 2004.
- [12] K. Nahm, A. Helmy, and C. J. Kuo, "The impact of probe traffic scale on the stability of multihop 802.11 networks," *IEEE INFOCOM '05 poster*, <http://dawn.cs.umbc.edu/INFOCOM2005/Infocom2005-Posters.html>, Miami, FL, March 2005.
- [13] K. Nahm, A. Helmy, and C. J. Kuo, "TCP over multihop 802.11 networks: Issues and performance enhancement," in *Proc. ACM MobiHoc '05*, Urbana-Champaign, IL, May 2005.
- [14] G. Holland and N. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks," Proc. *ACM MOBICOM*, 1999.
- [15] P. Sinha, N. Venkitaraman, R. Sivakumar, and V. Bharghavan, "WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks," Proc. *ACM MOBICOM '99*, 1999.
- [16] Z. Fu, B. Greenstein, X. Meng, and S. Lu, "Design and Implementation of a TCP-Friendly Transport Protocol for Ad Hoc Wireless Networks" Proc. *IEEE Int'l Conf. Network Protocols*, 2002.
- [17] J. Liu and S. Singh, "ATCP: TCP for Mobile Ad Hoc Networks," *IEEE J. Selected Areas in Comm.*, vol. 19, no. 7, pp. 1300-1315, July 2001.
- [18] F. Wang and Y. Zhang, "Improving TCP Performance over Mobile Ad-Hoc Networks with Out-of-Order Detection and Response," Proc. *Third ACM Int'l Symp. Mobile Ad Hoc Networking & Computing*, 2002.
- [19] X. Yu, "Improving TCP performance over mobile ad hoc networks by exploiting cross-layer information awareness," in Proc. *ACM MobiCom 04*, Philadelphia, PA, September, 2004.
- [20] Z. Fu, X. Meng, and S. Lu, "A transport protocol for supporting multimedia streaming in mobile ad hoc networks," *IEEE journal on selected areas in communications*, vol. 21, no. 10, pp. 1615-1626, December, 2004.
- [21] S. Xu and T. Saadawi, "Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks?" *IEEE Communications Magazine*, pp. 130-137, June 2001.
- [22] Z. Fu, X. Meng, and S. Lu, "How bad TCP can perform in mobile ad-hoc networks," in Proc. *IEEE ICNP '02*, Paris, France, 2002.
- [23] Kitae Nahm, Ahmed Helmy, C.-C. Jay Kuo, "TCP over multihop 802.11 networks: Issues and performance enhancement," *ACM MobiHoc*, Urbana-Champaign, IL, May 2005
- [24] K. Sundaresan, V. Anantharaman, H. Hsieh, and R. Sivakumar, "ATP: a reliable transport protocol for ad-hoc networks," in Proc. *ACM MobiHoc '03*, Annapolis, MD, 2003.
- [25] K. Chen, K. Nahrstedt, and N. Vaidya, "The utility of explicit rate-based flow control in mobile ad hoc networks," in Proc. *IEEE WCNC*, Atlanta, GA, 2003.
- [26] *IEEE Computer Society*, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," IEEE Standard 802.11, 1997.

**Dong Wan-li** was born in 1981. He received the BS degree in computer science from the Southwest University of Science and Technology. He is currently a PHD candidate in the Department of Computer Science and Technology, USTC. His research interests include wireless and mobile networking, parallel and distributed computing.

**Gu Nai-jie** was born in 1961. He is a Professor and Doctoral Advisor in the Department of Computer Science and Technology, USTC. His research interests include parallel computing architecture, interprocessor communication, and high-performance computing.

**Tu Kun** was born in 1980. He is a Ph.D. student in the Department of Computer Science and Technology, USTC. His research interests include wireless and mobile networking, parallel and distributed computing.

**Bi Kun** was born in 1981. He is a Ph.D. student in the Department of Computer Science and Technology, USTC. His research interests include wireless and mobile networking, parallel and distributed computing.

**Liu Gang** was born in 1978. He is a Ph.D. student in the Department of Computer Science and Technology, USTC. His research interests include wireless and mobile networking, parallel and distributed computing.