Cross-Layer Interaction of TCP and Ad Hoc Routing Protocols in Multihop IEEE 802.11 Networks

Kitae Nahm, *Member*, *IEEE*, Ahmed Helmy, *Member*, *IEEE*, and C.-C. Jay Kuo, *Fellow*, *IEEE*

Abstract—In this research, we first investigate the cross-layer interaction between TCP and routing protocols in the IEEE 802.11 ad hoc network. On-demand ad hoc routing protocols respond to network events such as channel noise, mobility, and congestion in the same manner, which, in association with TCP, deteriorates the quality of an existing end-to-end connection. The poor end-to-end connectivity deteriorates TCP's performance in turn. Based on the well-known TCP-friendly equation, we conduct a quantitative study on the TCP operation range using static routing and long-lived TCP flows and show that the additive-increase, multiplicative-decrease (AIMD) behavior of the TCP window mechanism is aggressive for a typical multihop IEEE 802.11 network with a low-bandwidth-delay product. Then, to address these problems, we propose two complementary mechanisms, that is, the TCP fractional window increment (FeW) scheme and the Route-failure nOtification using BUlk-losS Trigger (ROBUST) policy. The TCP FeW scheme is a preventive solution used to reduce the congestion-driven wireless link loss. The ROBUST policy is a corrective solution that enables on-demand routing protocols to suppress overreactions induced by the aggressive TCP behavior. It is shown by computer simulation that these two mechanisms result in a significant improvement of TCP throughput without modifying the basic TCP window or the wireless MAC mechanisms.

Index Terms—TCP, on-demand ad hoc routing protocol, IEEE 802.11, ad hoc networks, cross-layer interaction.

1 Introduction

The IEEE 802.11 wireless LAN is widely available these days, whereas the multihop topology is frequently found in ad hoc network applications such as sensor networks, wireless mesh networks, and home/office networks. When being integrated, the IEEE 802.11 ad hoc network has quite a few features different from the wired network, for example, shared queues, half-duplex links, channel noise, and mobility-induced effects. It has been observed that TCP suffers from poor bandwidth utilization and extreme unfairness in this environment, and the utility of TCP in the multihop IEEE 802.11 network has been seriously questioned. On the other hand, TCP is the most dominant transport protocol that serves as a basis for many other protocols in wired and wireless networks. Interoperability with the TCP-dominant wireless or wired network is often critical to some ad hoc network applications using IEEE 802.11. For example, the wireless mesh network (as an extension of the hot-spot-based wireless LAN) has to support TCP over the multihop topology for wireless LAN clients. Thus, it is important to work with TCP and improve its performance in the IEEE 802.11 ad hoc network. This motivates our current study of TCP behavior and related cross-layer interaction in multi-hop IEEE 802.11 networks.

Most of previous transport design for ad hoc wireless networks was based on one common assumption. That is, the underlying layers (that is, routing and MAC) of wireless networks affect the transport layer at end hosts. Thus, many efforts have been made in the intelligent adaptation of end hosts to wireless channel errors, varying bandwidth, routing changes, and other wireless ad hoc networking events. TCP resets its state and reduces the transmission rate when an end-to-end connection is momentarily broken down due to the route change in a mobile environment. This observation leads to the window-freeze strategy, where wireless TCP resumes its previous transmission rate without resetting the state, if the event is not caused by congestion. To implement this idea, TCP-ELFN [1], [2] uses an explicit link failure notification (ELFN) from the network to distinguish congestion from routing change in a wireless environment. ATCP [3], ADTCP [4], and ADTFRC [4] monitor the change of end-to-end network conditions to distinguish congestion from other network events.

Recent in-depth studies on IEEE 802.11 have revealed more fundamental problems of multihop IEEE 802.11 networks rooted at contention-based wireless MAC. IEEE 802.11 MAC has advanced handshaking options such as RTS/CTS but cannot avoid signal interference in multihop topologies [5], [6]. It makes the bandwidth-delay product of multihop IEEE 802.11 networks very small. It

- *K. Nahm and C.-C.J. Kuo are with the Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089-2654. E-mail: {nahm, cckuo}@sipi.usc.edu.*
- *A. Helmy is with the Computer and Information Science and Engineering (CISE) Department, University of Florida, Gainesville, FL 32611. E-mail: helmy@cise.ufl.edu.*

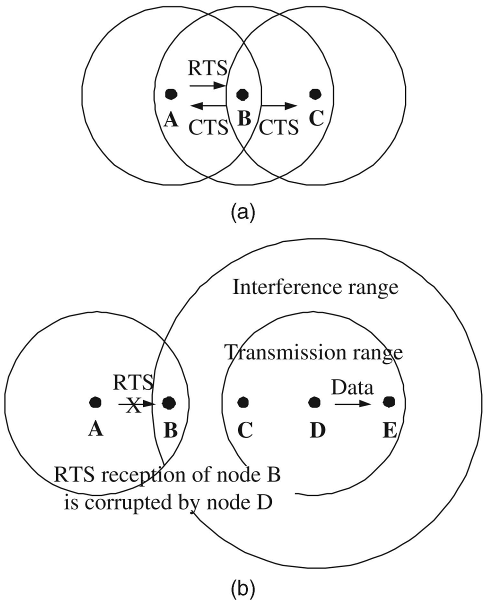
Manuscript received 27 Dec. 2006; revised 5 July 2007; accepted 16 Oct. 2007; published online 25 Oct. 2007.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0346-1206.
Digital Object Identifier no. 10.1109/TMC.2007.70779.

was observed in [5], [7], [8], and [9] that a TCP with a small congestion window (for example, 1 or 2 packets) outperforms a TCP with a larger one in the IEEE 802.11 ad hoc network. This observation led to a customized TCP for multihop IEEE 802.11 networks. For example, Chen et al. [10] suggested enforcing an optimal window limit based on the hop count of an end-to-end connection, since the bandwidth-delay product of a connection is proportional to the hop count of the connection. Fu et al. [5] proposed a link-layer active queue management scheme called LRED that exploits explicit congestion notification (ECN) marking to prevent the TCP congestion window from growing too much. It was shown by the authors in [11] that the additive-increase, multiplicative-decrease (AIMD) behavior of the TCP window mechanism is too aggressive for networks of a low bandwidth-delay product, resulting in frequent wireless link loss. A fractional widow increment scheme, where the TCP congestion window grows at a fractional rate (namely, with $\alpha < 1$ packets per round-trip time), was proposed to remedy this shortcoming.

In the meanwhile, the utility of TCP and TCP friendliness in the multihop IEEE 802.11 networks has long been challenged [8]. When wireless ad hoc networks are stand alone, it is possible to consider a cross-layer design [12], where network components across layers are closely coupled with each other. For example, Chen et al. [13] proposed a rate-based transport design utilizing the side information (that is, the bandwidth estimate) from the underlying wireless link. It was suggested by Lim et al. [14] that the wireless link layer plays a partial role in congestion control. This approach deviates from the traditional layered network architecture model [15] and does not assure compatibility with TCP (for example, TCP friendliness).

In this work, we examine this problem from a different angle; namely, how the transport layer at end hosts affects lower wireless network layers (for example, routing and MAC layers). We identify an ideal condition for IEEE 802.11 multihop networks, which facilitates our analysis on the impact of the TCP window mechanism on lower protocol layers. Then, we propose a couple of simple yet effective ways to reduce the difference between the ideal and the actual network conditions and show a significant improvement in routing stability and TCP performance experimentally without modifying the basic TCP window or wireless MAC mechanisms. Our study indicates that the TCP window mechanism is still a viable solution for IEEE 802.11 ad hoc networks as long as some key conditions (for example, routing robustness) are met.

The rest of this paper is organized as follows: We perform an in-depth investigation of cross-layer interaction in multihop IEEE 802.11 networks and conduct a quantitative study on the impact of TCP's greedy behavior using the well-known TCP-friendly equation under static routing in Section 2. Then, we introduce two complementary solutions, namely, the TCP fractional window increment (FeW) scheme and the Route-failure nOtification using BUlk-losS Trigger (ROBUST) policy, in Section 3. We evaluate these two schemes via computer simulation and compare their performance with other approaches such as ADTCP [4] and LRED [5] under various wireless network topologies in Section 4. The performance improvement is more evident in resource-limited environments such as chainlike topologies or networks with crowded traffic. Finally, concluding remarks are given and future research directions are pointed out in Section 5.



(a)

(b)

Fig. 1. Signal interference in multihop IEEE 802.11 networks. (a) Classic hidden-terminal problem. (b) Extended hidden-terminal problem.

2 Overview of Cross-Layer Interaction

2.1 Signal Interference and 802.11 MAC

In a wireless medium, the signal is transmitted to all directions, and it may suffer from interference due to another node's transmission. In a multihop topology, IEEE 802.11 MAC cannot completely prevent the signal interference problem, *a.k.a.*, the extended hidden-terminal problem [5], [6], [16]. The classic hidden-terminal problem, as shown in Fig. 1a, occurs when the transmission range of each node covers just a single-hop distance, thus making nodes A and C "hidden" from each other. However, as node B can see both of them, node B can coordinate the transmission of hidden nodes A and C using RTS/CTS to avoid concurrent transmission. The extended hidden-terminal problem, as shown in Fig. 1b, occurs when there are some other nodes beyond node B's transmission range. Since node D is beyond node B's control, node D can interfere with the RTS/CTS exchange of nodes A, B, and C. In Fig. 1b, the interference range spans two hops, where the signal is not strong enough to deliver meaningful data yet strong enough to corrupt data in another's transmission range. Consequently, node D keeps interfering with the RTS reception of node B when it is busy (for example, sending data to node E or C). As node B does not respond with CTS, node A keeps retransmitting RTS until it reaches the maximum count (7 in IEEE 802.11) and eventually drops the packet. In short, nodes A and D become new hidden terminals to each other.

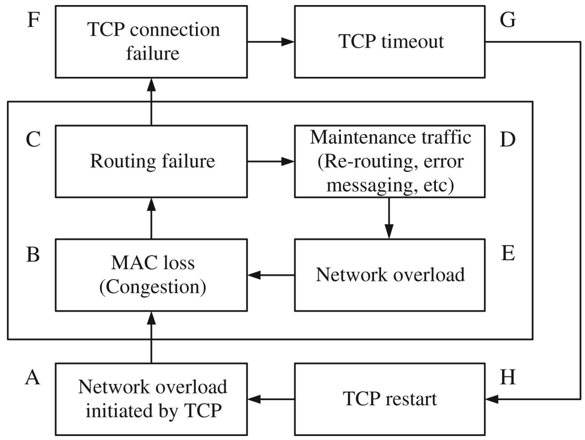


Fig. 2. The multiloop cross-layer interaction model.

The extended hidden-/exposed-terminal problem reduces the bandwidth-delay product of multihop IEEE 802.11 networks significantly [5], [6]. The bandwidth-delay product represents the total number of unacknowledged in-flight packets between the TCP sender and receiver, including all packets in the queue and under transmission in the link. The wireless link loss caused by signal interference starts before the queue overflow, and it triggers premature TCP reaction in reducing its sending rate. Because of little queue buildup, the bandwidth-delay product of multihop IEEE 802.11 networks is mostly determined by the number of packets transmitted at the wireless link. As shown in Fig. 1b, one packet at every four hops survives from the interference so that the bandwidth-delay product of the network is about 1/4 of the end-to-end hop counts [5], [6].

2.2 Multiloop Cross-Layer Interaction Model

On-demand routing protocols such as DSR [17] and AODV [18] establish a path on demand and monitor the path throughout the session. If the path is believed to have failed (typically due to the wireless link loss), the protocol performs a maintenance/rediscovery operation on the path. On the other hand, in multihop 802.11 networks, congestion produces the link loss rather than the queuing loss due to the extended hidden-/exposed-terminal problem. This interferes with the route maintenance task of the routing protocol. The whole process can be explained by a multiloop protocol interaction model, as shown in Fig. 2.

First, the TCP window mechanism overloads the network and intensifies the MAC contention (in steps A and B). The packet loss at the link layer is perceived as a routing failure by the on-demand routing protocol (in step C). Being confused with the routing failure, the routing agent enters a recovery process by sending error messages, updating and reestablishing the routing table, and salvaging some lost packets (in step D). The recovery process creates network traffic at the critical point of congestion and link failure (in step E). While the network overload is not resolved, the MAC contention loss occurs again (in step B). Meanwhile, due to the routing failure, the TCP connection is interrupted (in step F) and then times out (in step G). Since there is no external packet entering the network during this period, the network overload is reduced, and the routing and MAC functions are recovered. However, after time out, TCP restarts (in step H) and overloads the network again (in step A).

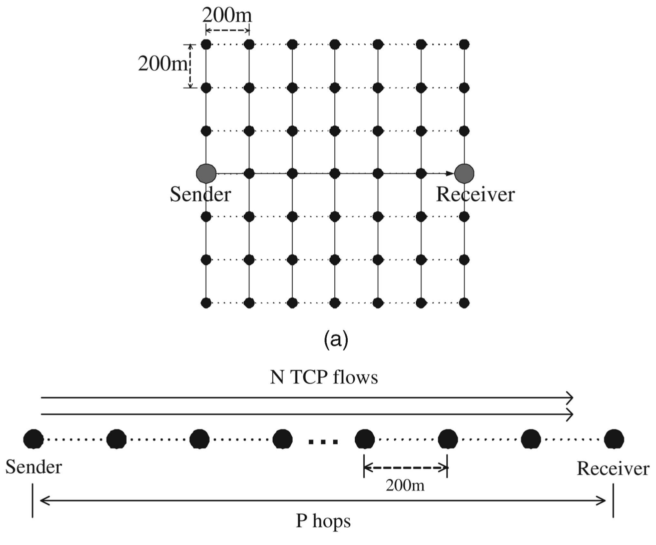
Basically, Fig. 2 represents the lack of a coordination in sharing resources between transport and on-demand routing protocols. TCP, by nature, tries to take the available network bandwidth as much as possible. Due to the contention-based 802.11 MAC, TCP's greedy behavior leaves little that might otherwise be used for the other critical functions in different layers (for example, routing maintenance here). MAC and on-demand routing protocols suffer from the lack of network resources at the critical moment when they are most needed. TCP's greedy behavior leads to the severe instability of the whole network, which hurts TCP performance in return.

In Fig. 2, it is only between steps H and A (that is, before network overloading) that TCP is given a quality end-to-end connection. The inner loop (formed by steps B, C, D, and E) is self-sustaining in the sense that the routing recovery attempt induced by the link failure results in another link failure. For a network with a long end-to-end hop distance, there could be multiple points of link failure, and the full path may not be easily recovered until the network load goes well below its capacity. While the inner loop repeats, the network remains unstable unless a proper action is taken elsewhere (for example, end hosts) to reduce the network load.

2.3 On-Demand versus Static Routing

In this section, we examine the multiloop cross-layer interaction model using the NS-2 simulator (version 2.29) [19]. It is well known that NS-2 uses the so-called "capture threshold" model for modeling interference. This model is known to be inaccurate, and in particular, it accounts for interferers one at a time, rather than cumulatively, which is the case in a practical single-antenna system. However, this shortcoming may be as severe in our current work since any interference model that produces "link loss" events should suffice to illustrate the interaction of TCP and ad hoc routing protocols.

The data rate of the wireless channel was 2 Mbps, and the radio propagation model was the two-ray ground model with a transmission range of 250 m, a carrier sensing range of 550 m, and an interference range of 550 m. We considered two routing settings: 1) a preconfigured static routing protocol and 2) the DSR protocol [17], which serves as an example of the on-demand routing protocol. With the static routing protocol, routing tables and paths do not change, and each node simply forwards packets along a preconfigured path. For the 7 × 7 grid topology, as shown in Fig. 3a, the static path is configured to be the shortest path (of six hops) between the sender and the receiver. The bandwidth-delay product of the six-hop path is theoretically $6/4 = 1.5$ packets [5], [6], and the per-flow bandwidth-delay product is equal to 0.75 packets as two TCP flows are tested.



(a)

(b)

Fig. 3. Illustration of two simulation topologies. (a) 7 × 7 grid topology. (b) Chain topology.

Fig. 4 shows the change of the path made by the DSR protocol during a 120-second simulation session in the 7 × 7 grid topology. There were 130 times of route changes, whereas the shortest (six-hop-distance) path was used only for about 50 percent of the simulation session. This result shows severe routing instability, even though there was neither node mobility nor wireless channel errors. We see in Table 1 that on-demand routing yields lower TCP throughput than the static routing. Their ratio is 88.1 percent. As explained by our cross-layer interaction model, this result indicates that some of the network capacity should be reserved for other purposes such as routing maintenance.

Fig. 5 shows the TCP congestion window changes over time for the six-hop chain topology. With the DSR protocol, TCP congestion avoidance is frequently interrupted. In contrast, static routing offers a clear and smooth AIMD pattern. In Table 1, the TCP throughput using the DSR protocol is about 43 percent of that using static routing. Although the overreaction of on-demand routing protocols gives a partial reason for poor performance, the significant difference in TCP throughput between the chain topology and the 7 × 7 grid topology needs another explanation. That is, the interaction between the transport and routing protocols actually plays a critical role in the overall system performance. It is also notable that static routing provides an ideal condition in which TCP behaves well. This motivates our quantitative study of the TCP behavior given below.

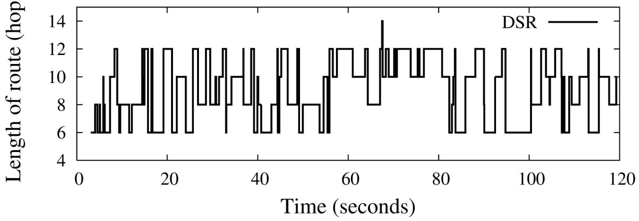


Fig. 4. Routing change in the 7 × 7 grid topology.

TABLE 1
TCP Throughput in the Unit of Kilobits per Second

Routing	6-hop chain			7x7 grid		
	Flow 1	Flow 2	Total	Flow 1	Flow 2	Total
Static	105.30	102.17	207.47	89.36	116.86	202.22
DSR	46.92	42.32	89.24	87.59	90.57	178.16
$\frac{DSR}{Static}$	44.46%	41.49%	43.01%	98.02%	77.5%	88.1%

2.4 TCP Operation Range Analysis

Since the MAC contention loss is elicited by TCP with static routing, we examine the quantitative relationship between the TCP window mechanism and the MAC contention loss in 802.11 multihop networks. The TCP congestion window defines the number of packets to be sent at every round-trip time, and TCP is designed to adjust its congestion window according to the bandwidth-delay product of the network. Without loss of generality, we assume the same size for all packets so that the bandwidth-delay product can be expressed in terms of the number of packets. Under the assumption of no wireless channel error, we consider the loss perceived by the end user without differentiating the MAC contention loss from the queue overflow loss.

The TCP-friendly equation is a mathematical model to characterize the steady-state TCP behavior. It describes the TCP window averaged over a long period of time. Padhye et al. [20] expressed the average TCP throughput as

$$X = \frac{1}{RTT\sqrt{\frac{2b}{3}p} + T_0 \min\left\{1, 3\sqrt{\frac{3b}{8}p}\right\}p(1+32p^2)},$$

where p and RTT are the loss rate and the round-trip time based on the end-to-end observation, respectively. T_0 is the time-out parameter, and b is the delayed-acknowledgment (ACK) factor with which the growth rate of the TCP window at every round-trip time is $\Delta W = 1/b$ [20]. For a typical configuration of the TCP-friendly equation in the TFRC protocol, we have $T_0 = 2RTT$ and $b = 1$. We can use the TCP-friendly equation to calculate the average window W by

$$W = X \cdot RTT.$$

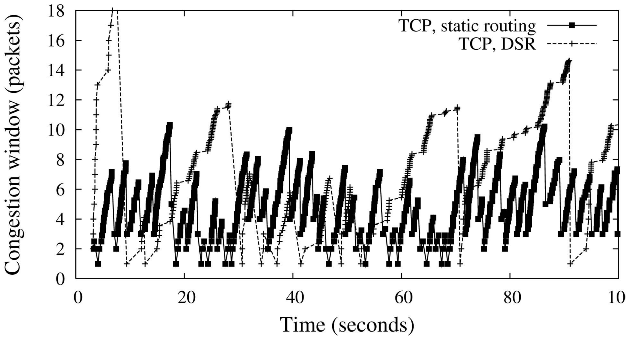


Fig. 5. The TCP behavior in the six-hop chain topology.

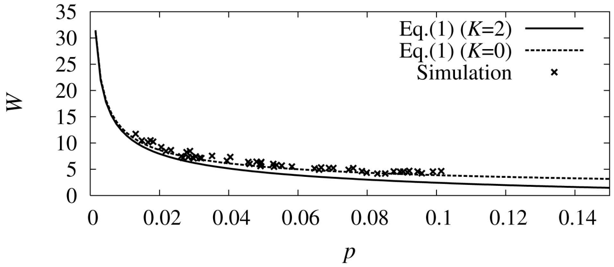


Fig. 6. The TCP window and the loss rate with static routing.

By setting $T_0 = K \cdot RTT$ and $\alpha = \frac{1}{b}$, the window version of the TCP-friendly equation can be written as

$$W = \frac{1}{\sqrt{\frac{2}{3\alpha}p} + K \min\left\{1, 3\sqrt{\frac{3}{8\alpha}p}\right\} p(1+32p^2)}. \quad (1)$$

We show the relationship between the average window size W and the packet loss rate p measured by (1) in Fig. 6, where several IEEE 802.11 multihop networks were simulated. The static routing scheme was adopted in the simulation to prevent any unpredictable effect induced by routing dynamics. The values of W and p were obtained by time-averaging 500-second simulation sessions with one, two, four, and eight TCP flows for a chain topology, with its hop number varying from 4 to 22, as shown in Fig. 3b.

Although the loss in IEEE 802.11 multihop networks primarily comes from the MAC layer, the relationship between W and p, shown in Fig. 6, conforms with the TCP-friendly equation in (1), which was originally established for wired IP networks. Thus, we conclude that if the routing dynamics does not interfere with the TCP operation, the steady-state TCP AIMD behavior of the 802.11 network would not be much different from that of wired IP networks. For a low-bandwidth-delay-product network, the TCP window mechanism leads to a high loss rate, as revealed in Fig. 6. For example, when $\alpha = 1$ and $K = 2$, $W < 1$ demands $p > 0.19$, whereas $W < 2$ demands $p > 0.12$ under (1). Such a high loss rate of TCP exacerbates the undesirable cross-layer interaction given in Fig. 2.

3 Improving TCP Performance and Network Stability

If the congestion-driven link loss can be clearly distinguished from channel or mobility errors, the TCP behavior can be improved accordingly. However, this approach would require some side information from other layers (for example, the node movement trace and/or the physical signal measurement). Here, we consider a simple alternative that suppresses the undesirable cross-layer interaction in Fig. 2 without the side information.

3.1 TCP: Fractional Window Increment

We choose proper K and α values to control the TCP operation range while preserving the basic TCP window mechanism. From (1), $K = 0$ and $\alpha = 1$ provide the upper bound for the shifted TCP operation range. Thus, we have the following two constraints on the time-out factor, K, and the window growth factor, α:

$$K \geq 0, \quad (2)$$

and

$$0 < \alpha \leq 1. \quad (3)$$

The relationship between W and p for $K = 2$ is also shown in Fig. 6 for comparison. This curve has a lower loss rate than the curve with $K = 0$ with respect to a fixed window value.

The time-out factor, K, can be obtained as a function of W and p by (1), that is

$$K = \frac{W^{-1} - \sqrt{\frac{2}{3\alpha}p}}{\min\left\{1, 3\sqrt{\frac{3}{8\alpha}p}\right\} p(1+32p^2)}. \quad (4)$$

We define $W_\alpha = \sqrt{\frac{1.5\alpha}{p}}$ and $p_\alpha = \frac{1.5\alpha}{W^2}$. From condition $K \geq 0$ given in (2), we demand

$$W \leq \sqrt{\frac{1.5\alpha}{p}} = \sqrt{\alpha} \cdot W_1, \quad (5)$$

or, equivalently,

$$p \leq \frac{1.5\alpha}{W^2} = \alpha \cdot p_1, \quad (6)$$

where the equalities in (5) and (6) hold when $K = 0$. Thus, the TCP congestion window with factor $0 < \alpha \leq 1$ has a lower dynamic range with the same loss rate p, or the TCP loss rate with factor $0 < \alpha \leq 1$ has a lower value for the same window value W.

Based on the above discussion, we propose a new TCP regulation scheme that allows the TCP congestion window to grow by a fractional rate $\alpha \leq 1$ (packets) at every round-trip time. This is equivalent to adding one packet to the window size at every $\frac{1}{\alpha}$ round-trip time. Suppose that the current congestion window size is W, the TCP sender sends W packets at every round-trip time and receives W ACKs during one round-trip time from the TCP receiver. At every ACK reception, the TCP sender updates W by

$$W^{new} = W^{current} + \frac{\alpha}{W^{current}}, \quad (7)$$

where $0 < \alpha \leq 1$. If $\alpha = 1$, the above rule is the same as the window update rule of legacy TCP. Since the proposed scheme preserves the TCP window mechanism, it can track the network capacity automatically without the aid of any network-assisted information.

Since parameter α represents the growth rate of the TCP window W at every round-trip time (that is, $\alpha = \Delta W$), we say that α represents the probe traffic of TCP. The probe traffic should be mild enough not to cause network instability yet aggressive enough to enable meaningful network feedback. To achieve congestion avoidance in legacy TCP ($\alpha = 1$), the TCP window increases by one packet at every round-trip time independent of the network capacity. For wired networks whose bandwidth delay products are in the order of hundreds or thousands of

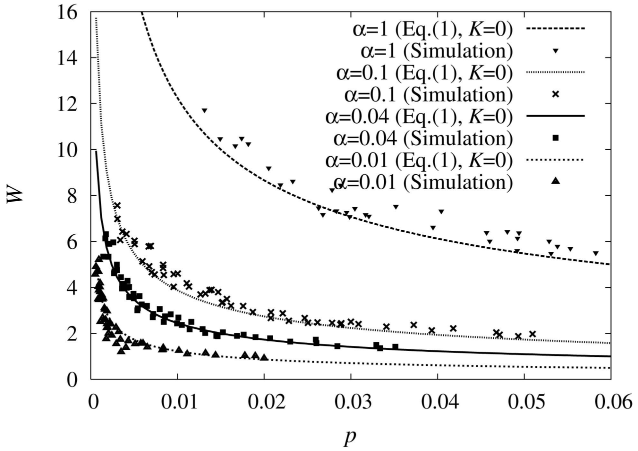


Fig. 7. TCP friendliness of the fractional window scheme with static routing.

packets, the relative scale of the probing traffic is small. However, for IEEE 802.11 multihop networks that typically have a low bandwidth-delay product, this probing traffic may not be mild at all. For example, if the bandwidth-delay product of a network is equal to two packets, an increase of congestion window by one packet is equivalent to adding 50 percent of the total network capacity. Thus, TCP with $\alpha = 1$ is too aggressive for IEEE 802.11 multihop networks. The reduction in α makes the scale of the probing traffic more appropriate.

We also allow a fractional window (that is, $0 < W \leq 1$), as well as $W \geq 1$, without losing compatibility with legacy TCP. We interpret $W < 1$ as sending one data packet at every $\frac{RTT}{W}$. To implement this, a new internal timer is introduced to avoid the possible deadlock with TCP ACK clocking at $W < 1$. For example, if $W = 0.25$, the next data transmission is scheduled in four RTT by the timer. When the timer expires, the packet is sent, and W is updated according to (7). A parameter W_{min} can be introduced as the lower bound of W (that is, $W > W_{min}$) to avoid a potential long idle period for a very small W. The TCP parameter ssthresh (slow-start threshold) also needs a lower bound (that is, $\texttt{ssthresh} > \texttt{ssthresh}_{min}$, where $0 < \texttt{ssthresh}_{min} < 1$) so that the TCP slow-start is not triggered when W is very small.

In Fig. 7, we show the relationship between the average window W and the packet loss rate p for different α values. The data were collected using static routing in the same way as that used in Fig. 6. With (5) and (6), the case of $\alpha = 0.01$ represents the shift of the operational range via $W = 0.1W_1$ at the same loss rate p. Likewise, $\alpha = 0.04$ and $\alpha = 0.1$ imply that $W = 0.2W_1$ and $W \approx 0.3W_1$, respectively, at the same loss rate p. Fig. 7 shows that the shift of the operational range conforms with (1) in a network free from routing dynamics and instability. The TCP window mechanism can achieve a very low loss rate with a small average window if a proper value of α is chosen.

3.2 On-Demand Routing: Route-Failure Notification Using Bulk-Loss Trigger Policy

We consider the ROBUST policy in this section, which is a simple link loss reaction policy for on-demand routing to improve the path robustness against the MAC contention loss driven by congestion. The typical implementation of on-demand ad hoc routing protocols takes a route maintenance/rediscovery action immediately upon detecting any wireless link loss. However, in IEEE 802.11 multihop networks, the link loss has to be treated differently according to its cause:

- Node mobility. When a node is no longer available, the current route should be invalidated, and a new route should be established regardless of its cost.
- Congestion. The congestion-driven loss is a sign of network overload, indispensable for a seamless TCP operation. Nevertheless, the existing route should not be affected.
- Channel noise. Noisy routes are tolerable in the sense that channel noise affects only the quality of the end-to-end connection, not the connectivity itself. A noisy path should be tolerated until a better alternative path is found. If needed, a hop-by-hop retransmission can improve the connection quality.

If we consider the fact that the congestion-driven MAC loss is frequent and persistent during the entire course of TCP connection, an immediate reaction of a routing protocol to any MAC loss is costly. Examples given in the previous section show that the reaction should be less sensitive to the MAC loss to keep a certain level of routing robustness and stability. In the meanwhile, the congestion-driven MAC loss should be signaled to end users so that transport protocols can react properly to the network change.

To account for the link failure tolerability, we introduce a generalized link failure sensitivity parameter, β, for on-demand routing protocols to control connection stability. The number of *successive* link failures L is counted, and a routing protocol takes action for routing maintenance only when L exceeds a certain threshold, β, which indicates the limit of tolerable successive link failures.

1. If the transmission fails and $L < \beta$, the current packet is dropped, and the transmission is resumed for the next packet. The current route is kept intact, and the value of L is increased by 1.
2. If the transmission fails and $L = \beta$, the routing agent responds to the link failure and sets $L = 0$. The routing agent is free to establish a new route and perform necessary routing maintenance operations.
3. Whenever a transmission is completed successfully, the routing agent resets $L = 0$.

The IEEE 802.11 standard specifies a maximum of seven times of RTS retransmission for a wireless link. However, factor β makes the routing protocol respond to a link failure at least after 7β times of RTS retransmission from 802.11 MAC's perspective. Please note that $\beta = 1$ is used in the conventional reaction policy for on-demand routing protocols, and $\beta = \infty$ in static routing. Thus, $\beta > 1$ enables a stronger end-to-end connectivity in the presence of channel noise and signal interference.

However, this policy has nothing to do with the retransmission of the same packet by the routing protocol. The routing agent sends a packet *only once* for any value of $\beta \geq 1$. If the packet is lost after retransmission in the

IEEE 802.11 MAC layer, the routing agent sends out the next packet without saving the lost packet. Later, when the end host detects the missing packet, TCP can respond to the MAC contention in the same way as it does to the ordinary network congestion. The packet dropped by the routing agent also helps relieve the MAC contention of the local host whose packet is lost. The possible drawback of $\beta > 1$ is the increase of reaction latency in the mobility scenario. However, the negative impact of small values of $\beta > 1$ appears negligible in a high-mobility scenario as demonstrated by the computer simulation presented in the next section.

4 Evaluation and Comparison

4.1 Simulation Setup

We evaluated the two proposed schemes using the NS-2 simulator (version 2.29) [19] for various scenarios and compared them with other related schemes. As discussed in Section 2, we adopted the DSR protocol [17] as the on-demand ad hoc routing protocol in the simulation. The ROBUST policy was evaluated when being applied to the DSR protocol. We used long-lived TCP flows running for 200 seconds in various topologies such as chain, grid, and mobile random topologies. Please note that the long-lived flow provides a condition emulating the stationary ergodic process, which makes the time-averaged TCP throughput the same as the ensemble average (that is, the average over different simulations) [21], [22]. This reduces the simulation-dependent deviation of the result for static topologies and makes the TCP throughput conform to the TCP-friendly equation with static routing.

The number of TCP flows was different, depending on the topology type. The simulated traffic patterns included linear, parallel, cross, and random traffics for static grid and mobile random topologies. The result of random traffic was obtained from the average of five runs based on different random number seeds. The result of static traffic was time-averaged before plotting. In some complicated scenarios such as grid and random topologies, the extreme starvation of local traffic is often found. As is addressed in [23] and [24], this is primarily due to the nature of the contention-based CSMA MAC of multihop IEEE 802.11 networks. Thus, we consider only the aggregate throughput as the primary TCP performance measure in this study.

For comparison with the window-freeze strategy, we considered ADTCP [4]. ADTCP utilizes various end-to-end metrics to determine the wireless network status and freeze the TCP state instead of resetting TCP for noncongestion wireless events. As shown in the next section, ADTCP tends to keep the window large despite frequent routing changes and link losses. It is interesting to compare this behavior with our strategy using a small window at a low window growth rate.

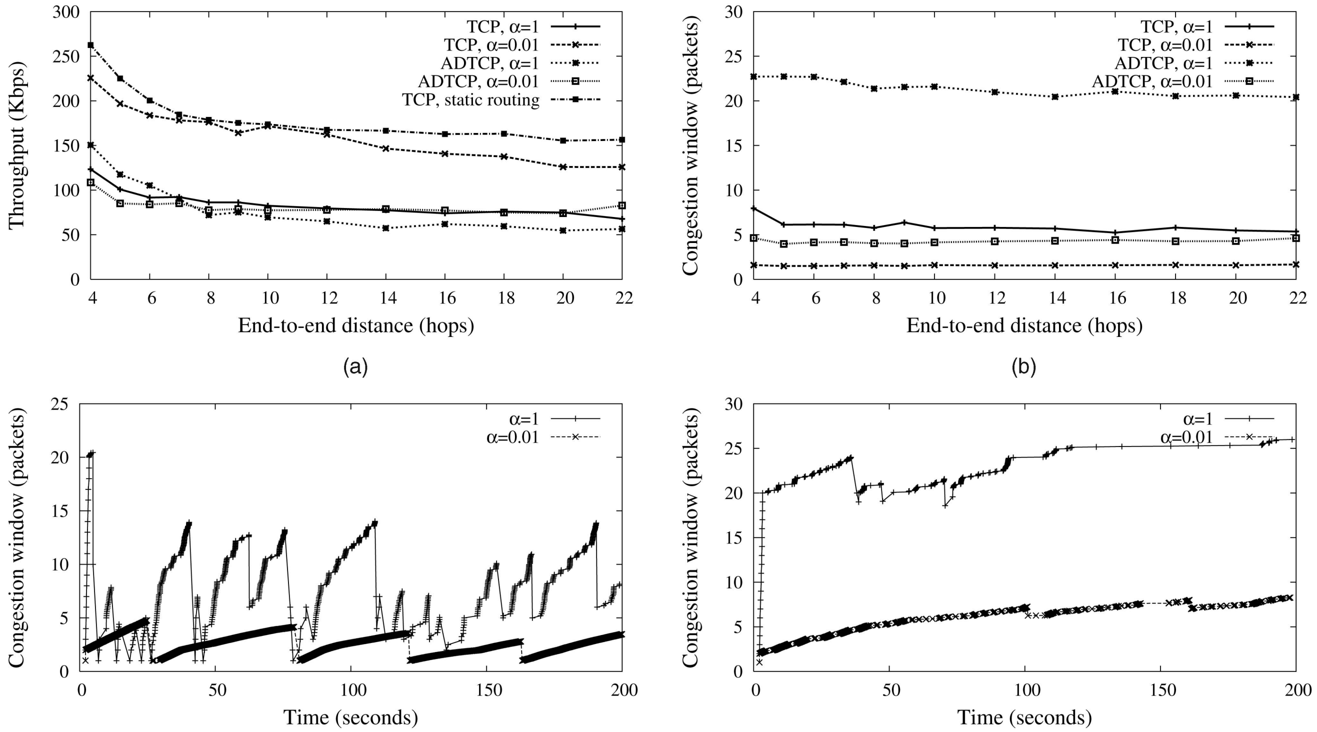
For comparison with other congestion preventive strategies, we examine the use of ECN in the LRED proposal [5]. ECN [25] uses packet marking rather than packet loss for network congestion notification, and TCP can reduce the sending rate before the queue overflow. Since ECN reduces the actual loss that triggers the overreactions of an on-demand routing protocol, ECN yields an effect similar to that of using TCP FeW. Like random early detection (RED) [26], LRED performs ECN marking. However, the ECN marking of LRED is based on the MAC retransmission count instead of the queue length. The original LRED assumed a preconfigured manual routing but did not address the interaction with on-demand routing protocols of our current interest. Instead, LRED cooperates with adaptive pacing, which requires a modification of the back-off mechanism of the 802.11 MAC protocol. To separate the effect of IEEE 802.11 back-off changes and focus on the interaction only between transport and routing mechanisms, we kept the original back-off scheme of IEEE 802.11 MAC intact and implemented only the link-layer ECN marking of LRED without adaptive pacing.

4.2 TCP Fractional Window Increment versus Window Freeze

The fractional-window and window-freeze schemes were evaluated over chain, grid, and random mobile topologies. The topologies are carefully chosen for fair comparison with the result of related studies [4], [5], [6]. The chain topology is the basic building block of more complex topologies. It is commonly found in ad hoc and mesh networks where the density of node deployment is not high. The chain topology also represents the path established by routing whether it is static or mobile. Please note that the bandwidth-delay products of test topologies are low, theoretically up to $22/4 = 5.5$ (packets) in the case of the 22-hop chain topology, plus a few more from little queue buildup [5], [6]. In this section, static routing is used to obtain the TCP performance bound, whereas the DSR protocol ($\beta = 1$) is used to observe the actual TCP performance as a result of the aforementioned cross-layer interaction.

Fig. 8 compares the throughput and the congestion window of TCP and ADTCP over a chain topology with and without the FeW scheme. As compared with the original TCP ($\alpha = 1$) or other ADTCPs, TCP FeW ($\alpha = 0.01$) achieves significant improvement in throughput (90 percent improvement for six hops and 85 percent improvement for 14 hops), as shown in Fig. 8a. ADTCP mostly underperforms the original TCP with a few exceptions, but ADTCP FeW ($\alpha = 0.01$) outperforms ADTCP ($\alpha = 1$) consistently. Note that a fractional value of the TCP congestion window has to be truncated to an integer value when the value is referred to by the TCP sender. For small values of the congestion window, the TCP sender actually sends one or two data packets at every round-trip time. Thus, Fig. 8b shows that TCP with $\alpha = 0.01$ behaves like a stop-and-go protocol in the low-bandwidth-delay-product environment of the chain topology.

Figs. 8c and 8d give a snapshot of the congestion window variation over time in the seven-hop chain topology using the DSR protocol ($\beta = 1$). Fig. 8c shows a severely disconnected AIMD pattern with $\alpha = 1.0$, which is far from being the ideal TCP congestion avoidance pattern. Frequent overshooting beyond the network's bandwidth-delay product results in the congestion-driven MAC contention loss, which disrupts the TCP congestion avoidance phase. In contrast, TCP FeW ($\alpha = 0.01$) leads to a more



(a)

(b)

(c)

(d)

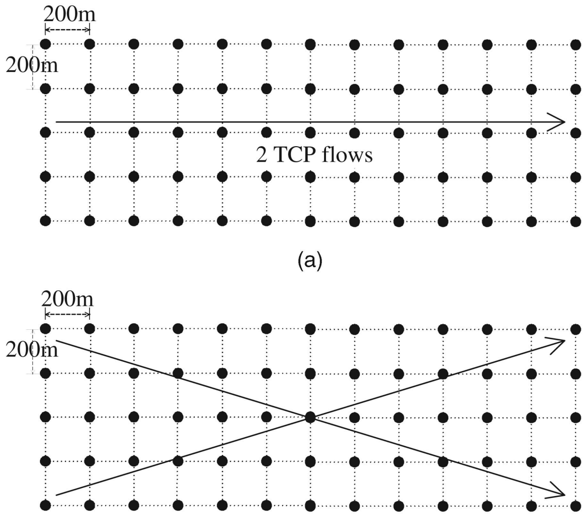
Fig. 8. Comparison of different TCP schemes as a function of time in a chain topology. (a) Throughput (aggregation of four flows). (b) Average congestion window (four flows). (c) Congestion window over time (TCP, two flows). (d) Congestion window over time (ADTCP, two flows).

regular AIMD pattern whose averaged value is close to the optimal window size. Fig. 8d shows how the FeW scheme ($\alpha = 0.01$) works for ADTCP. The window-freeze mechanism of ADTCP remains the same. However, the slower window growth with the FeW scheme makes ADTCP FeW ($\alpha = 0.01$) have a smaller average congestion window than the original ADTCP ($\alpha = 1$).

Next, we perform our evaluation study on grid topologies. The grid topology has much more redundancy that allows alternative routes and backup resources than the chain topology. In the simulation, we used a small (7 × 7), a large (13 × 13), and a narrow-stripe (13 × 5) grid topology. Fig. 9 shows some test traffic patterns over the narrow-strip grid topology.

Figs. 10a and 10b show results of the 7 × 7 and the 13 × 13 grids, respectively. For most cases, the total TCP throughput of $\alpha = 0.01$ is noticeably higher than that of $\alpha = 1$, and TCP FeW improves the performance of ADTCP and TCP. However, in the two-flow cases (parallel and cross), TCP FeW has little loss (or a small gain) in throughput for $\alpha = 0.01$ as compared with that for $\alpha = 1$. As more alternative paths and backup resources are available, the impact of network instability driven by the aggressive behavior of TCP is not much noticeable in the two-flow cases. In fact, the two-flow scenarios are very optimistic scenarios since the grid topology is severely underutilized. Fig. 10c shows the result of the 13 × 5 grid topology. The 13 × 5 grid topology represents a more common network scenario that lies between the resource-limited chain topology and the resource-rich grid topology. We observe a similar trend of throughput improvement as found in other topologies yet with little or no performance degradation for the two-flow case.

Finally, we evaluated TCP FeW and ADTCP in the mobile scenario where 150 nodes move in 2,000 m × 2,000 m space according to the random waypoint model. The random waypoint configuration used in our simulation has a uniform speed distribution between the maximum speed of 10 m/s and the minimum speed of 2 m/s with no pause time. The result is shown in Fig. 10d, where we see a



(a)

(b)

Fig. 9. Examples of traffic patterns in grid topology simulation. (a) 13 × 5 grid (overlapping traffic). (b) 13 × 5 grid (cross traffic).

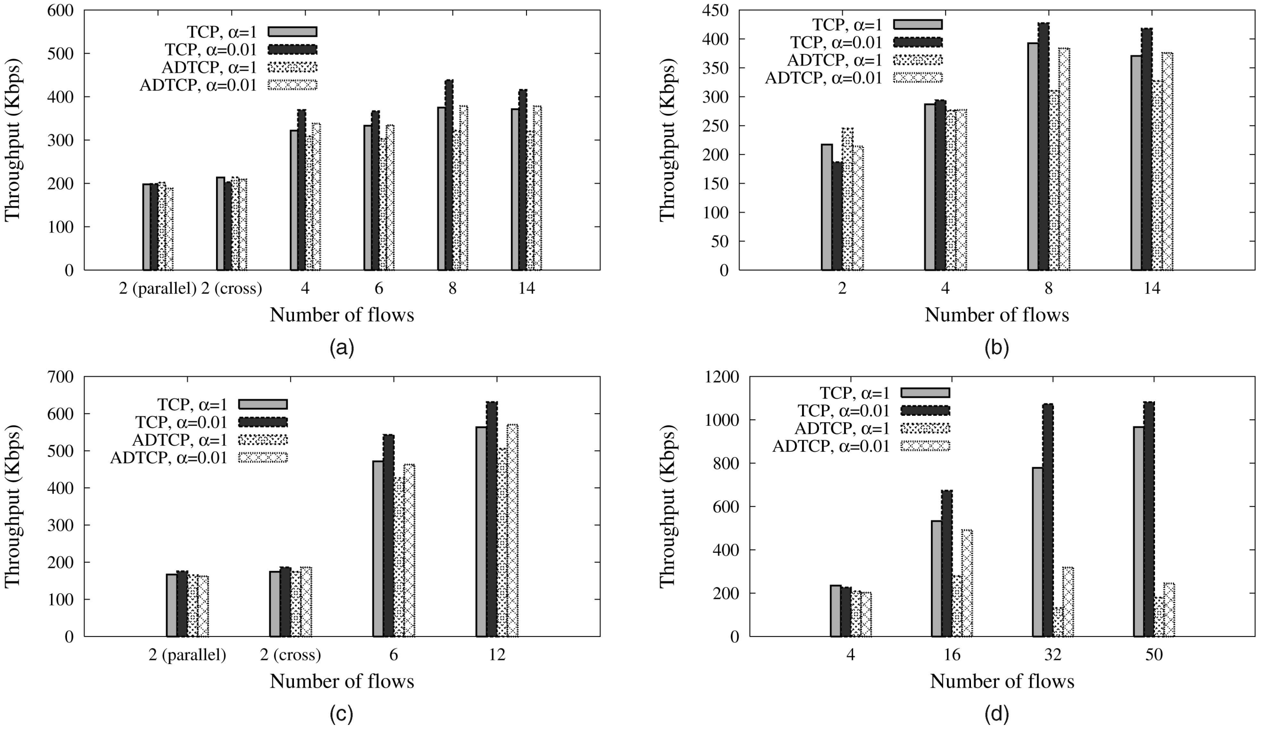


Fig. 10. Comparison of TCP throughput in grid and mobile random topologies. (a) 7 × 7 grid. (b) 13 × 13 grid. (c) 13 × 5 grid. (d) Mobile random.

similar trend of the aggregate TCP throughput in Fig. 8. TCP outperforms ADTCP by more than four times for 32 and 50 flows. TCP FeW has more than 35 percent throughput improvement over TCP ($\alpha = 1$) for 32 flows, and even ADTCP FeW has more than 100 percent throughput improvement over ADTCP with $\alpha = 1$. One of the primary design focuses of many window-freeze schemes, including ADTCP [1], [3], [27], is on mobility. Our simulation shows that the throughput improvement of TCP FeW is more noticeable than that of mobility-adaptive window-freeze schemes. The random mobile topology often has an unbalanced node distribution and contains chainlike subtopologies. Noticeable improvement is made in such a resource-limited environment with a low bandwidth-delay product.

The simulation reported in this section demonstrates that the basic TCP window mechanism can keep track of the low bandwidth-delay product of the multihop IEEE 802.11 network if a proper α value is chosen. For smaller values of α (for example, $\alpha = 0.01$), TCP can maintain a smaller average congestion window at a lower loss rate and achieve better performance in the multihop IEEE 802.11 network. When α is close to 1, TCP performance degrades since it makes the average congestion window go beyond the desirable operation range of IEEE 802.11 multihop networks.

4.3 Utility of Explicit Congestion Notification

We implemented the link-layer ECN marking based on LRED [5] in the NS-2 simulator and evaluated the ECN effect using two TCP flows over the chain topology. Fig. 11 shows the aggregate throughput of TCP and TCP-ECN with different α values. For $\alpha = 1$, the average throughput of TCP-ECN is noticeably higher than that of TCP yet much larger than that of TCP FeW ($\alpha = 0.01$) without ECN. TCP FeW (with or without ECN) achieves 70 percent improvement over TCP-ECN for an end-to-end distance of six hops, 50 percent improvement for 10 hops, and 45.5 percent improvement for 14 hops. Although ECN can be used as a part of the preventive measure to reduce the overreaction of routing protocols, the result in Fig. 11 indicates that the TCP window growth rate has more impact on the behavior of routing protocols. It is true that ECN can reduce the frequency of actual link loss, but TCP with $\alpha = 1$ remains as a severe network stress factor affecting the MAC contention. The use of smaller α values can reduce such stress more effectively.

In Fig. 11, TCP FeW ECN ($\alpha = 0.01$) performs slightly better than TCP FeW. Even though ECN can be integrated with the FeW scheme, the performance gain of ECN is hardly noticeable for smaller α values because the loss rate is so low. Thus, controlling the window growth rate is a very powerful tool in preventing unnecessary routing reactions.

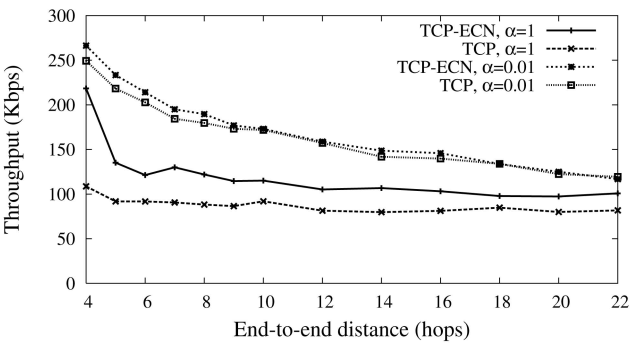


Fig. 11. Effect of ECN on TCP performance in the chain topology with two flows.

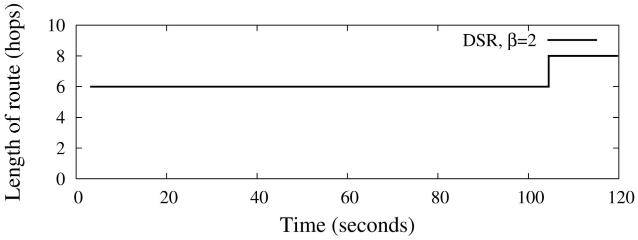


Fig. 12. Route change over time with the ROBUST policy ($\beta = 2$) in a 7 × 7 grid topology.

4.4 Route-Failure Notification Using Bulk-Loss Trigger Policy

The ROBUST policy was applied to the DSR protocol and evaluated over the 7 × 7 grid topology with the same configuration as discussed in Section 2.3. Unlike the result in Section 2.3, DSR with $\beta = 2$ changed the path only once, as shown in Fig. 12, and the total TCP throughput given in Table 2 is almost the same as that of static routing. Increasing β just by 1 makes a huge difference in routing dynamics over IEEE 802.11 multihop networks.

The simulation result over the chain topology with two different channel noise conditions (with the packet error rate equal to 0 and 0.04) is shown in Fig. 13. We see a similar trend as observed before. For the 6-12-hop chain topologies with no channel noise ($\text{PER} = 0.0$), the DSR protocol with the ROBUST policy ($\beta = 2$) achieved more than 100 percent improvement in TCP throughput, which is very close to the performance bound obtained with static routing. For $\text{PER} = 4$ percent, the DSR protocol with the ROBUST policy ($\beta = 2$) achieved more than 200 percent improvement of TCP throughput over DSR without the ROBUST policy ($\beta = 1$) in the 4-20-hop chain topologies. The vulnerability of TCP to channel errors remains the same for $\beta = 2$ and $\beta = 1$. However, we see an improvement in route robustness using the ROBUST policy, which is beneficial to the overall system performance including TCP throughput.

Finally, the effect of the ROBUST policy was evaluated in a more realistic mobile scenario using the Manhattan mobility model [28]. A total of 200 mobile nodes were created in the 500 m × 500 m space where six streets (three north-south and three east-west) were placed at equal distance. In each mobile scenario, mobile nodes move randomly along 12 lanes (two lanes for each street) at the maximum speed of 0, 1, 2, 5, 10, and 20 m/s, respectively. To ensure sufficiently large hop distances of the end-to-end connections, we used a transmission range of 60 m, a carrier sensing range of 140 m, and an interference range of 140 m.

TABLE 2
TCP Throughput (in Kilobits per Second) in the 7 × 7 Grid Topology with the ROBUST Policy

Routing	Flow 1	Flow 2	Total	$\frac{DSR_\beta}{Static}$
Static ($\beta = 0$)	89.36	116.86	202.22	100%
$\text{DSR}_{\beta=2}$	92.06	108.76	200.82	99.3%
$\text{DSR}_{\beta=1}$	87.59	90.57	178.16	88.1%

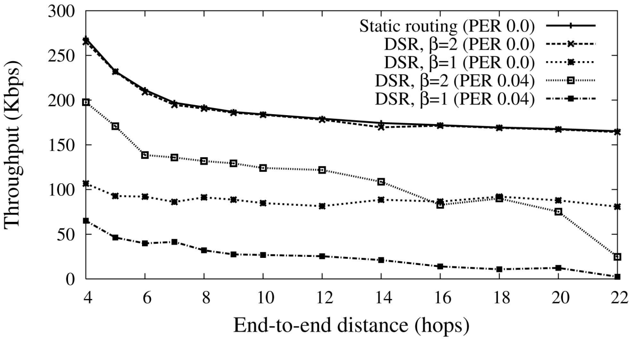
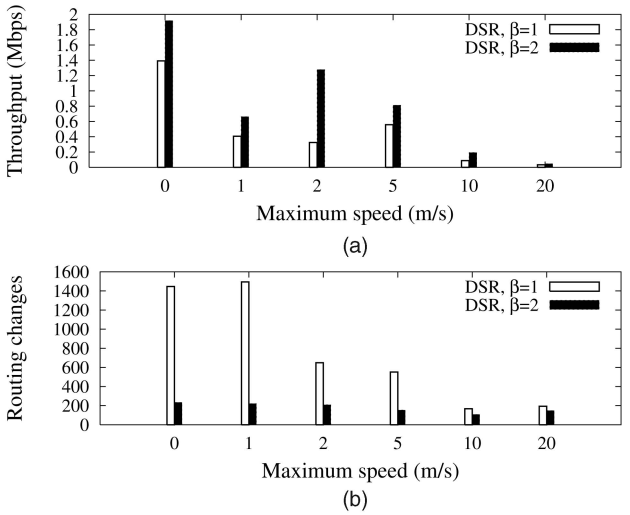


Fig. 13. TCP throughput comparison (which is the aggregate throughput of two TCP flows) with different routing policies in the chain topology.

The total throughput and the total number of routing changes of 20 TCP flows during the 150-second simulation session are shown in Fig. 14. With the original DSR protocol ($\beta = 1$), routing changes are more frequent at a lower speed (0 and 1 m/s). A small amount of mobility significantly reduces the total throughput as a result of the costly routing maintenance operations. However, for DSR ROBUST ($\beta = 2$), TCP performance is improved by at least more than 30 percent, and the routing protocol is more stabilized (65-85 percent less routing changes) in a moderate mobile scenario (up to 5 m/s) in our simulation where TCP throughput is practically meaningful. We also see that the impact of increased mobility-reaction latency on the overall system performance due to $\beta > 1$ is negligible.

4.5 TCP Fractional Window Increment and the Route-Failure Notification Using Bulk-Loss Trigger Policy

Although the multihop IEEE 802.11 network presents complex problems across layers, the TCP FeW scheme and the ROBUST policy provide stand-alone solutions that do not require changes of other layers. In this section, we examine the mixed usage of TCP FeW and ROBUST. Fig. 15 shows TCP throughput for different configurations of



(a)

(b)

Fig. 14. Performance comparison using the Manhattan mobility model. (a) Aggregate TCP throughput (20 flows). (b) Total number of routing changes (20 flows).

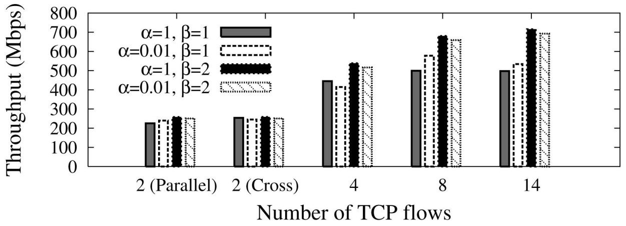


Fig. 15. TCP throughput in the 7 × 7 grid topology.

TCP FeW and DSR ROBUST over the 7 × 7 grid topology. The advantage and disadvantage of TCP FeW comes from its milder TCP traffic. If the hop-by-hop connection is robust or the network is resource rich, TCP's aggressive behavior may not disturb the overall network stability significantly. As demonstrated in the two-flow cases in Figs. 10a and 10b, TCP FeW may just slow down the TCP window growth and result in underutilization of the available network bandwidth.

For $\beta = 1$, the result is the same as Fig. 10a. For $\beta = 2$, DSR ROBUST improves the total throughput of TCP and TCP FeW. Since routing becomes more robust, TCP FeW with mild additive-increase traffic does not make a noticeable difference in throughput improvement as compared with legacy TCP. Basically, the two schemes solve the same cross-layer problem from two different angles. TCP FeW with ROBUST does not offer a dramatic improvement in throughput. The performance difference of TCP and TCP FeW for the same β value can be understood as a matter of finding an optimal value of α for the bandwidth-delay product of a given network topology and for loss tolerability of on-demand routing.

5 Conclusion and Future Work

Improving TCP performance in the IEEE 802.11 ad hoc network is a complicated cross-layer problem. The cross-layer interaction between TCP and routing protocols in the IEEE 802.11 ad hoc network was investigated. Based on the well-known TCP-friendly equation, we conducted a quantitative study on the TCP operation range using static routing and long-lived TCP flows. Based on the analysis, we proposed two simple yet effective ways, namely, TCP FeW and ROBUST, to improve the system performance. It was shown via computer simulation that TCP performance can be significantly improved without modifying the basic TCP window or the wireless MAC mechanism. Thus, the TCP window mechanism can still be a viable solution for IEEE 802.11 ad hoc networks as long as key conditions are met (for example, routing robustness).

The cross-layer interaction observed in this study can be understood as a part of a more general problem of the contention-based MAC with heavy traffic. Zhang et al. [29] reported that a sudden outburst of sensing traffic overloads the sensor network using S-MAC [30], [31] and B-MAC [32]. The outburst of sensing traffic interferes with the operation of the transport layer (for example, ACK delivery) and hurts the reliable delivery of critical sensing data. Likewise, the greedy behavior of TCP creates heavy traffic in the IEEE 802.11 ad hoc network and stimulates the vulnerability of the contention-based MAC and on-demand ad hoc routing protocols. The improvement achieved by TCP FeW demonstrates that simply limiting such heavy traffic can help improve the overall network performance.

There are a few issues to be considered in the future. In our simulation, TCP FeW performance varied depending on the network condition (for example, whether the routing is robust or not). We may improve TCP FeW by tuning factor α properly to meet the performance requirement for a given network condition. Besides, the TCP FeW design discussed in this work focuses on long-lived flows for the convenience of TCP-friendliness analysis. However, small files that take only a few round-trip times to finish may not get TCP into the steady state. Actually, the use of short-lived flows does not prevent the interaction between MAC and routing if the traffic goes beyond the bandwidth-delay product of the wireless network. Thus, if α is set correctly, TCP FeW can help alleviate such undesirable interaction regardless of long- or short-lived flows. Then, we can revise the transient behavior of TCP FeW to improve the efficiency of small file transfers.

One interesting problem associated with ad hoc routing is finding the interference-avoidance path. In view of the protocol architecture, the routing layer lies between transport and link layers, and the routing protocol has a direct access to the information of other neighboring layers (that is, transport and wireless link) of each host along the path. In this work, we have focused on the robust path maintenance problem. In the future, we may consider a more aggressive adaptive routing strategy by utilizing the side information such as the location, signal strength, traffic conditions, and other flows actively.

References

[1] G. Holland and N. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks," *Proc. ACM MobiCom*, 1999.

[2] X. Yu, "Improving TCP Performance over Mobile Ad Hoc Networks by Exploiting Cross-Layer Information Awareness," *Proc. ACM MobiCom '04*, Sept. 2004.

[3] 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, 2001.

[4] Z. Fu, X. Meng, and S. Lu, "A Transport Protocol for Supporting Multimedia Streaming in Mobile Ad Hoc Networks," *IEEE J. Selected Areas in Comm.*, vol. 21, no. 10, pp. 1615-1626, Dec. 2004.

[5] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The Impact of Multihop Wireless Channel on TCP Throughput and Loss," *Proc. IEEE INFOCOM '03*, Mar. 2003.

[6] K. Chen, Y. Xue, S. Shah, and K. Nahrstedt, "Understanding Bandwidth-Delay Product in Mobile Ad Hoc Networks," *Elsevier Computer Comm.*, special issue on protocol engineering for wired and wireless networks, vol. 27, pp. 923-934, 2004.

[7] S. Xu and T. Saadawi, "Does the IEEE 802.11 MAC Protocol Work Well in Multihop Wireless Ad Hoc Networks?," *IEEE Comm. Magazine*, pp. 130-137, June 2001.

[8] K. Chen and K. Nahrstedt, "Limitations of Equation-Based Congestion Control in Mobile Ad Hoc Networks," *Proc. IEEE Int'l Workshop on Wireless Ad Hoc Networking (WWAN '03)*, Mar. 2003.

[9] Z. Fu, X. Meng, and S. Lu, "How Bad TCP Can Perform in Mobile Ad-Hoc Networks," *Proc. Seventh Int'l Symp. Computers and Comm. (ISCC)*, 2002.

[10] K. Chen, Y. Xue, and K. Nahrstedt, "On Setting TCP's Congestion Window Limit in Mobile Ad Hoc Networks," *Proc. IEEE Int'l Conf. Comm. (ICC '03)*, May 2003.

[11] K. Nahm, A. Helmy, and C.J. Kuo, "TCP over Multihop 802.11 Networks: Issues and Performance Enhancement," *Proc. ACM MobiHoc '05*, May 2005.

[12] K. Sundaresan, V. Anantharaman, H. Hsieh, and R. Sivakumar, "ATP: A Reliable Transport Protocol for Ad-Hoc Networks," *Proc. ACM MobiHoc,* 2003.

[13] K. Chen, K. Nahrstedt, and N. Vaidya, "The Utility of Explicit Rate-Based Flow Control in Mobile Ad Hoc Networks," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC),* 2003.

[14] C. Lim, H. Luo, and C. Choi, "Rain: A Reliable Wireless Network Architecture," *Proc. 14th IEEE Int'l Conf. Network Protocols (ICNP '06),* Nov. 2006.

[15] H. Zimmermann, "OSI Reference Model—The ISO Model of Architecture for Open Systems Interconnection," *IEEE Trans. Comm.,* vol. 28, no. 4, pp. 425-432, Apr. 1980.

[16] J. Li, C. Blake, D.D. Couto, H. Lee, and R. Morris, "Capacity of Ad Hoc Wireless Networks," *Proc. ACM MobiCom '01,* July 2001.

[17] D. Johnson, D. Maltz, and Y. Hu, "The Dynamic Source Routing for Mobile Ad Hoc Networks," IETF Internet draft, work in progress, July 2004.

[18] C. Perkins, E. Belding-Royer, and S. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing," IETF Internet draft, work in progress, Feb. 2003.

[19] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu, "Advances in Network Simulation (NS)," *Computer,* vol. 33, no. 5, pp. 59-67, May 2000.

[20] J. Padhye, V. Firoiu, D. Towsley, and J. Krusoe, "Modeling TCP Throughput: A Simple Model and Its Empirical Validation," *Proc. ACM SIGCOMM '98,* pp. 303-314, Sept. 1998.

[21] E. Altman, K. Avrachenkov, and C. Barakat, "A Stochastic Model of TCP/IP with Stationary Random Losses," *IEEE/ACM Trans. Networking,* vol. 13, no. 2, pp. 356-369, Apr. 2005.

[22] M. Vojnovic and J.-Y.L. Boudec, "On the Long-Run Behavior of Equation-Based Congestion Control," *IEEE/ACM Trans. Networking,* vol. 13, no. 3, pp. 568-581, June 2005.

[23] M. Garetto, T. Salonidis, and E.W. Knightly, "Modeling Per-Flow Throughput and Capturing Starvation in CSMA Multi-Hop Wireless Networks," *Proc. IEEE INFOCOM '06,* Apr. 2006.

[24] K. Xu, M. Gerla, L. Qi, and Y. Shu, "Enhancing TCP Fairness in Ad Hoc Network Using Neighborhood Red," *Proc. ACM MobiCom '03,* Sept. 2003.

[25] S. Floyd, "TCP and Explicit Congestion Notification," *ACM Computer Comm. Rev.,* vol. 24, no. 5, pp. 10-23, Oct. 1994.

[26] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Trans. Networking,* vol. 1, no. 4, pp. 397-413, Aug. 1993.

[27] Z. Fu, G. Greenstein, X. Meng, and S. Lu, "Design and Implementation of a TCP-Friendly Transport Protocol for Ad Hoc Wireless Networks," *Proc. 10th IEEE Int'l Conf. Network Protocols (ICNP),* 2002.

[28] F. Bai, N. Sadagopan, and A. Helmy, "IMPORTANT: A Framework to Systematically Analyze the Impact of Mobility on Performance of Routing Protocols for Adhoc Networks," *Proc. IEEE INFOCOM '03,* Mar. 2003.

[29] H. Zhang, A. Arora, Y. Choi, and M. Gouda, "Reliable Bursty Convergecast in Wireless Sensor Networks," *Proc. ACM MobiHoc '05,* May 2005.

[30] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," *Proc. IEEE INFOCOM '02,* pp. 1567-1576, May 2002.

[31] *Wireless Embedded Systems,* http://webs.cs.berkeley.edu, 2007.

[32] J. Polatre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," *Proc. Second ACM Int'l Conf. Embedded Networked Sensor Systems (SenSys '04),* Nov. 2004.

Kitae Nahm received the BS degree in control engineering from Seoul National University, Seoul, in 1995 and the MS and PhD degrees in electrical engineering from the University of Southern California (USC), Los Angeles, in 2000 and 2005, respectively. He was with Intervideo Inc. in 2001 and is currently with the Digital Medial Solutions Laboratory, Samsung Electronics, Irvine, Calif. His research interests include the design, implementation, and analysis of various protocols of wireless mobile network and content delivery network and the various topics of the emerging network applications. He is a member of the IEEE.

Ahmed Helmy received the BS degree in electronics and communications engineering with highest honors and the MS Eng. Math. degree from Cairo University, Egypt, in 1992 and 1994, respectively, and the MS degree in electrical engineering and the PhD degree in computer science from the University of Southern California (USC) in 1995 and 1999, respectively. He is an associate professor and the founder and director of the wireless networking laboratory in the Computer and Information Science and Engineering (CISE) Department, University of Florida, Gainesville. From 1999 to 2006, he was an assistant professor of electrical engineering (EE) at the University of Southern California. He was also the founder and director of the wireless networking laboratory at USC. He was a key researcher in the network simulator (NS-2) and the protocol independent multicast (PIM-SM) projects in the Information Sciences Institute, USC. His research interests lie in the areas of network protocol design and analysis for mobile ad hoc and sensor networks, mobility modeling, design and testing of multicast protocols, IP micromobility, and network simulation. In 2002, he received the US National Science Foundation (NSF) CAREER Award. In 2000, he received the USC Zumberge Research Award, and in 2002, he received the best paper award from the IEEE/IFIP International Conference on Management of Multimedia and Mobile Networks and Services (MMNS). In 2003, he was the EE nominee for the USC Engineering Jr. Faculty Research Award and a nominee for the Sloan Fellowship. In 2004 and 2005, he got the best merit ranking in the EE-USC faculty. In 2007, he was a winner in the ACM MobiCom SRC research competition. He has been an area editor of the *Adhoc Networks Journal*, published by Elsevier, since 2007 (editor since 2004). He is the cochair for the IFIP/IEEE MMNS 2006 and IEEE INFOCOM Global Internet Workshop 2008 and the vice chair for IEEE ICPADS 2006 and HiPC 2007. He has been the ACM SIGMOBILE workshop coordination chair (for ACM MobiCom, MobiHoc, MobiSys, and SenSys) since 2006. He served on the program committees for numerous IEEE and ACM conferences in the areas of computer and wireless networks. He is a member of the IEEE.

C.-C. Jay Kuo received the BS degree in electrical engineering from the National Taiwan University, Taipei, in 1980 and the MS and PhD degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1985 and 1987, respectively. He is currently the director of the Signal and Image Processing Institute (SIPI) and a professor of electrical engineering, computer science, and mathematics in the Ming Hsieh Department of Electrical Engineering, University of Southern California (USC). His research interests are in the areas of digital image/video analysis and modeling, multimedia data compression, communication and networking, and biological signal/image processing. He is a coauthor of about 140 journal papers, 730 conference papers, and nine books. He is the editor in chief of the *Journal of Visual Communication and Image Representation* and an editor of the *Journal of Information Science and Engineering*, *LNCS Transactions on Data Hiding and Multimedia Security*, and *EURASIP Journal of Applied Signal Processing*. He was on the editorial board of the *IEEE Signal Processing Magazine* from 2003 to 2004. He served as an associate editor for the *IEEE Transactions on Image Processing* from 1995 to 1998, *IEEE Transactions on Circuits and Systems for Video Technology* from 1995 to 1997, and *IEEE Transactions on Speech and Audio Processing* from 2001 to 2003. He received the US National Science Foundation Young Investigator Award (NYI) and Presidential Faculty Fellow (PFF) Award in 1992 and 1993, respectively. He received the Northrop Junior Faculty Research Award from the USC Viterbi School of Engineering in 1994. He received the best paper award from the Multimedia Communication Technical Committee of the IEEE Communication Society in 2005. He is an IEEE Signal Processing Society Distinguished Lecturer in 2006. He is a fellow of the IEEE and the SPIE.