TCP over Multihop 802.11 Networks: Issues and Performance Enhancement

Kitae Nahm
nahm@sipi.usc.edu

Ahmed Helmy
helmy@usc.edu

C.-C. Jay Kuo
cckuo@sipi.usc.edu

Department of Electrical Engineering
University of Southern California, Los Angeles, CA 90089, U.S.A.

ABSTRACT

Analyzing TCP operation over 802.11 multihop ad hoc networks involves a cross-layer study. In this work, we investigate the effect of congestion and MAC contention on the interaction between TCP and on-demand ad hoc routing protocol in the 802.11 ad hoc networks. Our study reveals several problems stemming from lack of coordination and sharing in such networks. It is observed that TCP induces the over-reaction of routing protocol and hurts the quality of end-to-end connection. So, one of the critical sources of lowering TCP throughput lies in the TCP window mechanism itself. To fix this problem, we propose a fractional window increment (FeW) scheme for TCP to prevent the over-reaction of the on-demand routing protocol by limiting TCP's aggressiveness. The proposed scheme is applicable to a wide range of transport protocols using the basic TCP mechanism, and the protocol behavior is analytically tractable. Our simulation results demonstrate that the proposed scheme can dramatically improve TCP performance and network stability in a variety of 802.11 multihop networks. For example, in some chain-like topologies, the proposed scheme outperforms basic TCP by over 90%, and recent related variants of TCP (ADTCP, LRED) by over 70%.

Categories and Subject Descriptors: C.2.5 [Computer-Communication Networks]: Local and Wide-Area Networks - Internet (e.g., TCP/IP).

General Terms: Algorithms, Performance, Design.

Keywords: TCP, on-demand routing protocol, 802.11, ad hoc networks, performance.

1. INTRODUCTION

Multihop ad hoc wireless networks are one of the most commonly found in many applications of sensor networks, 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 shared queues, half-duplex links, channel noise and mobility-induced effects. Nevertheless, 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 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.

Most of previous research on 802.11 multihop networking has one common ground. That is, the transport mechanism is affected by routing dynamics. Here, we examine this problem from another viewpoint; *i.e.*, the routing dynamics are affected by the transport mechanism. In wireless ad-hoc networks, resources are shared by all network participants across layers. However, little effort has been made to coordinate the operations of the transport and the routing protocols. TCP by nature tends to take the available network resources as much as possible and leaves little that might otherwise be used for routing maintenance. The lack of network resources at the critical moment of routing discovery/maintenance operation affects the quality of the end-to-end connection, and eventually hurts the TCP performance. In this study, we show that simply reducing network overload helps improving the overall system performance of TCP and on-demand routing protocols.

Considering network overload as a major factor affecting packet loss, our analysis of the operational range of the basic TCP window mechanism leads to the conclusion that it is the growth rate of the TCP congestion window that effectively controls the loss rate and, consequently, the network overload. To keep the network load at a reasonable level, we propose a fractional window increment scheme, called *FeW*, that allows a fractional increment of the TCP congestion window. The proposed FeW scheme can be validated mathematically by the well-known TCP-friendly equation [1]. Simply speaking, a modification of the window increment scheme corresponds to a shift of the TCP operation to a new region suitable for networks of a low bandwidth-delay product. It is confirmed by simulation results that the proposed scheme can dramatically improve the transport performance in 802.11 multihop networks. The improvement is observed for a variety of scenarios, and it is especially obvious (with over 100% improvement in throughput) in resource-limited environments such as networks with chain-like sub-topologies. Such sub-topologies are one of the

MobiHoc'05, May 25–27, 2005, Urbana-Champaign, Illinois, USA.

common in realistic scenarios of non-uniform node distribution. It is in these resource limited environments that such improvement is most needed. The success of the fractional window scheme in simulation demonstrates that TCP-style congestion control still provides a viable transport solution in 802.11 multihop networks.

The rest of this paper is organized as follows. The background of this work and related previous work are presented in Sec. 2. The interaction between transport and on-demand routing protocols over 802.11 multihop networks is discussed and analyzed in Sec. 3. Then, we continue the analysis and propose a fractional window increment scheme to improve the performance of the basic TCP window mechanism in a low bandwidth-delay-product network in Sec. 4. The performance of the proposed scheme is evaluated in 802.11 networks of various topologies via simulation in Sec. 5. Concluding remarks and some future research directions are given in Sec. 6.

2. BACKGROUND AND RELATED WORK

Most of previous transport design for ad hoc wireless networks is based on the viewpoint that routing dynamics affect the transport mechanism residing at the end users. So, much 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 [2, 3] suggested an explicit link failure notification (ELFN) from the network to distinguish congestion from the routing change in a wireless environment. ATCP [4], ADTCP [5], and ADTFRC [5] 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.

In the meanwhile, researchers have started to take a closer look at the detailed interaction between TCP and other networking layers recently, *e.g.* [6, 7]. Basically, the TCP window mechanism tends to create more signal interference in a wireless ad hoc network environment. It is because the TCP window mechanism drives wireless networks to be crowded with more packets, where a higher spatial density of packets in an area leads to a higher chance of signal interference and collision in a wireless medium. As pointed out in [6, 7], 802.11 MAC cannot perfectly handle signal interference of general multihop topologies. The push of more packets to go beyond a certain limit by TCP drives excessive link-layer retransmission and eventually leads to more MAC contention loss (rather than queueing loss). In addition, the link (MAC) loss without being related to queue overflow leads to severe queue under-utilization and little queue buildup. Thus, the bandwidth-delay product of a TCP connection over multihop 802.11 networks tends to be very small as compared with that of wired IP networks.

This observation is confirmed in many recent studies ([8, 9, 6, 10]) by showing that TCP with a small congestion window (*e.g.*, 1 or 2) tends to outperform TCP with a large congestion window in 802.11 multihop networks. Chen *et al.* [11] suggested to enforce an optimal window limit based on the end-to-end hop count. Fu *et al.* [6] proposed a link-layer active queue management algorithm called LRED that exploits ECN marking to stabilize TCP window. Despite the above effort, the utility of TCP and TCP-friendliness in 802.11 multihop networks has long been challenged [9]. Since wireless ad hoc networks are typically stand-lone, a new rate-based approach was proposed for some transport design [12, 13]. They do not guarantee TCP compatibility because the network components across layers are closely tied with each other, exploiting side-information (*e.g.*, bandwidth) of the intermediary nodes.

In this work, we further examine this problem from another angle; namely, the routing dynamics is affected by the transport mechanism. Our approach is a preventive one that reduces unnecessary routing reaction to the unique congestion phenomenon in 802.11 multihop networks. The proposed FeW scheme lets the basic TCP window mechanism guide itself to the appropriate window size at a low growth rate without imposing a maximum window size. Our scheme has TCP compatibility in the sense that the basic TCP window mechanism is utilized and the protocol behavior is analytically tractable using the well known TCP model [1]. In the evaluation and comparison section we shall show how using the fractional window increment (FeW) scheme outperforms recent related work in this area (e.g., ADTCP citefu03transport and LRED [6]), leading to significant improvements in end-to-end TCP throughput.

3. CROSS-LAYER INTERACTION OF TCP AND ROUTING PROTOCOLS

3.1 Stability of Multihop 802.11 Networks

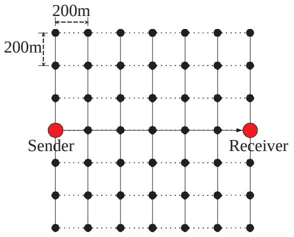
On-demand routing protocols (e.g., DSR [14] and AODV [15]) are among the most commonly used in the design and evaluation of ad hoc networks. In such routing protocols a route is established on-demand (during connection establishment) and is monitored throughout the session. If the route is believed to have failed (through loss detection) a route maintenance/re-discovery is triggered. In general, ad-hoc routing protocols do not distinguish between MAC contention loss, queue overflow, channel errors, mobility-induced losses or other. Hence, if MAC contention loss is frequent and persistent during the entire course of the TCP connection, on-demand routing protocols in 802.11 networks keep triggering unnecessary routing maintenance.

We investigate the cross-layer interaction between TCP and on-demand routing protocols using NS-2 simulator[16]. In the simulation, the data rate of the wireless channel was 2 Mbps and the radio propagation model was the two-ray ground model with transmission range 250m, carrier sensing range 550m, and interference range 550m. For this section, we only show the cases for two overlapping TCP flows over the 7x7 grid and 6-hop chain topologies shown in Fig. 1. To examine the routing dynamics, we consider two routing settings: **1.** a pre-configured static routing protocol and **2.** an on-demand routing protocol (i.e. the DSR protocol [14]) over chain and grid topologies. With the static routing protocol, the routing tables and paths do not change and each node simply forwards the packets along the pre-configured path (*e.g.*, the arrow between the sender and the receiver shown in Fig. 1(a)). With dynamic routing, route maintenance and re-discovery is triggered according to network dynamics as specified by the protocol.

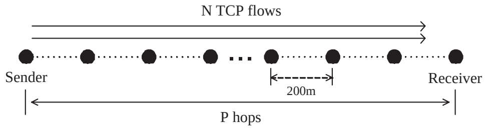
In Fig. 2, it is shown that the DSR protocol experienced 130 times of route changes during a 120-second simulation session over the 7x7 grid topology. Furthermore, the 6-hop-

Table 1: TCP throughput (Kbps)

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
$\frac{Static-DSR}{DSR}$	124.43%	141.43%	132.5%	1.98%	22.48%	13.5%



(a) 7x7 grid topology



(b) Chain topology

Figure 1: Simulation topologies.

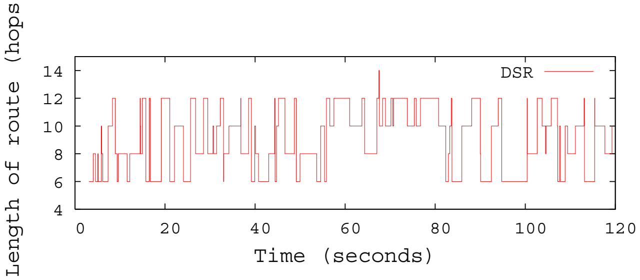


Figure 2: Routing change in the 7x7 grid topology.

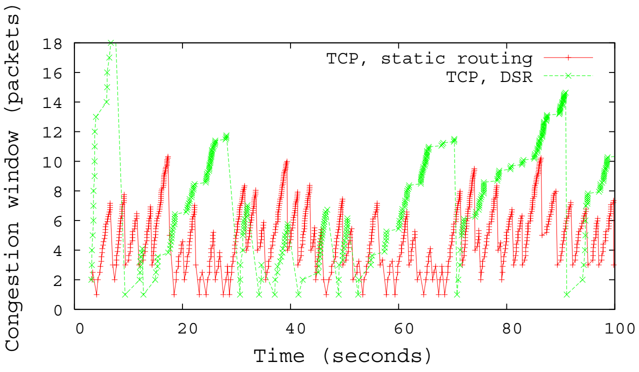


Figure 3: The TCP behavior in the 6-hop chain topology.

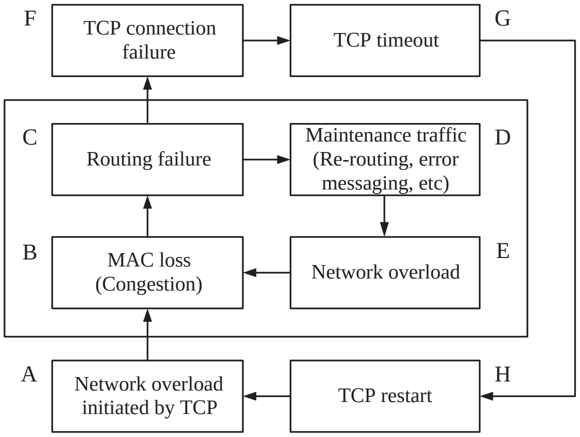


Figure 4: The connection blackout cycle for the chain topology.

distance route exists only for $\tilde{5}0\%$ of the simulation session. This indicates that the shortest path was not efficiently utilized even though we did not consider node mobility and wireless channel errors in the simulation. As a result, the TCP throughput with static routing is higher (by $\tilde{1}3\%$) than that with dynamic routing (Table 1). The instability and inefficiency of the on-demand ad hoc routing protocol is noteworthy. With dynamic routing, some network capacity was either wasted or used for other purposes such as routing maintenance, and the remaining capacity was allocated to the end user.

In Fig. 3, we compare the TCP behavior over the two routing protocols for the 6-hop chain topology. TCP congestion avoidance is frequently interrupted with the DSR protocol while a additive-increase-multiplicative-decrease (AIMD) pattern is clearly observed for the static routing protocol. Furthermore, as given in Table 1, the throughput of TCP with static routing is over 130% more than that for DSR. Even though the over-reaction of the DSR protocol to 802.11 link failure is commonly responsible for inefficiency and instability of both the systems, the throughput difference of the chain topology is far greater than that of the 7x7 grid topology. Fig. 4 elaborates the details of the cross-layer interactions leading to this severe network instability. Here, we refer to this sequence of actions as *connection blackout cycle* for a chain topology. The entire cycle consists of two loops: the inner loop with the interaction of MAC and routing protocols (Steps B, C, D, and E) and the outer loop with the interaction of TCP (transport) and routing protocols (Steps A, B, C, F, G, and H).

The entire cycle is initiated by TCP when a TCP sender sends data at a higher rate beyond network capacity (Step A). The network overload eventually leads to the hidden terminal problem and the MAC contention loss occurs after several retransmission attempts at the link layer (Step B). The packet loss at the link layer is perceived as a routing failure by the on-demand routing protocol such as DSR (Step C). Being confused with the routing failure, the routing agent enters a recovery process by sending error messages, updating and re-establishing the routing table, and salvaging some lost packets (Step D). The recovery process generates network traffic at the critical point of congestion and link failure (Step E). As a result, the network overload is not resolved while the MAC contention loss is repeated again (Step B). Meanwhile, due to the routing failure, the TCP connection is interrupted (Step F) and then times out (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 the timeout, TCP restarts (Step H) and soon leads to network overload again (Step A). Thus, the blackout cycle repeats.

In this diagram, it is only between Steps H and A (i.e., before network overload) that TCP can communicate over a quality end-to-end connection. The inner loop (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. Since a topology consisting of a large number of hops can have multiple points of routing/MAC failure at the same time, a full routing recovery is not easily achieved until the network load is reduced well below its capacity. The inner loop continues and the network remains unstable if no proper action is taken elsewhere (e.g., the end user) to reduce the overall network load.

The above phenomenon is a result of the lack of a coordination in sharing resources with transport and on-demand routing protocols in a resource-limited environment. But, the same phenomenon may still occur in a resource-rich environment, but at a lower impact level. We conclude that the on-demand routing protocols have a vulnerability with link failure over 802.11 networks. The interaction between the transport and routing protocols is a critical factor in determining the overall system performance of 802.11 networks.

3.2 Impact of TCP Window Mechanism

Since the MAC contention loss is elicited by TCP, we quantitatively examine the 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.* [1] derived 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 timeout and b is delayed-ACK factor with which the growth rate of TCP window at every round-trip-time is $\Delta W = 1/b$ [1]. 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 via

$$W = X \cdot RTT.$$

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

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

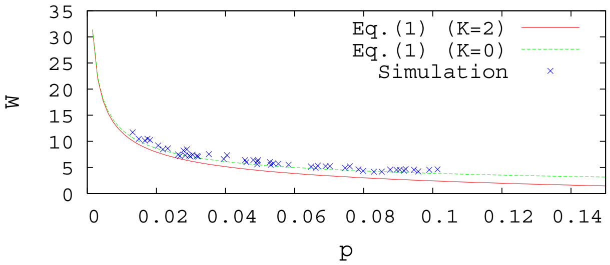


Figure 5: The TCP window and the loss rate with static routing.

In Fig. 5, we show the relationship between the average window size W and the packet loss rate p measured by Eq. (1) and simulation in several 802.11 multihop networks. 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-average in 500-second simulation sessions with 1, 2, 4, and 8 TCP flows for a chain topology with its hop number varying from 4 to 22 as shown in Fig. 1. Even though the loss in 802.11 multihop networks primarily comes from the MAC layer, the relationship between W and p shown in Fig. 5 conforms with the TCP-friendly equation of Eq. (1) originally established in the wired IP networks. Thus, we conclude that, if the routing dynamics does not interfere with TCP operation, the TCP behavior in the 802.11 networking environment would not be so much different from the TCP behavior in the wired IP networks.

For a low bandwidth-delay-product network, the TCP window mechanism creates a high loss rate. For example, when $\alpha = 1$ and $K = 2$, $W < 1$ demands $p > 0.19$ while $W < 2$ demands $p > 0.12$ under Eq. (1). Such a high loss rate with TCP in wireless networks exacerbates the blackout cycle as discussed in the previous subsection. In addition, allowing a high loss rate without channel errors is obviously a waste in the resource usage. The simulation result given in Fig. 5 implies that the TCP behavior in the 802.11 multihop networking environment is highly stimulus to the vulnerability of the on-demand routing protocols discussed in the previous subsection.

4. FRACTIONAL WINDOW INCREMENT (FEW) SCHEME

Since the MAC contention loss is elicited by TCP, we would like to include TCP as part of the whole solution to enhance the transport performance over 802.11 multihop networks. Even though properly chosen routing or MAC protocols may reduce the blackout cycle effect, allowing a high loss rate with TCP imposes huge workload on these possible network and link layer solutions. Thus, we consider a transport solution that prevents congestion loss from happening in this work. To achieve this goal, we consider shifting the TCP operational range to a lower loss rate.

Variables K and α are the two TCP parameters in Eq. (1). We choose proper K and α values to control the TCP operation range while preserving the basic TCP window mechanism. From Eq. (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 timeout factor K and the window growth factor α:

$$K \geq 0, \tag{2}$$

and

$$0 < \alpha \leq 1. \tag{3}$$

Fig. 5 also shows the relationship between W and p for the case of $K = 2$ for comparison. This curve has a lower loss rate than the curve with $K = 0$ with respect to a fixed window value.

The timeout factor K can be obtained as a function of W and p from Eq. (1), i.e.,

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

Let us 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 Eq. (2), we demand

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

or, equivalently,

$$p \leq \frac{1.5\alpha}{W^2} = \alpha \cdot p_1. \tag{6}$$

where 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 for the same loss rate p, or the TCP loss rate with factor α has a lower value for the same window value W.

In particular, let us consider the case where the window growth factor $\alpha = 1$. Following Eqs. (5) and (6), we have

$$W \leq \sqrt{\frac{3}{2p}}, \tag{7}$$

and

$$p \leq \frac{1.5}{W^2}, \tag{8}$$

respectively. Fig. 6 gives the plots of K for a loss rate range $0.015 < p < 0.06$ parameterized by the average window size $W = 1$, 2 and 5. We see clearly that to modify timeout parameter K appears to be ineffective at the low loss rate region, since K has to take a very large value.

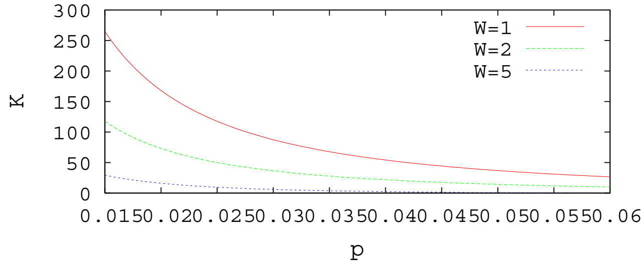


Figure 6: The relationship between the timeout factor K and the loss rate.

In Table 2, we show a couple of operational ranges as specified by the first column and the first row. The second column and the second row provide the operational range with $K = 0$ as the reference. Several scenarios are governed by Eqs. (4), (7) and (8) and labeled accordingly. Again, we see from the table that the value K becomes very large for lower W and p. For example, when $W < 2$, the loss rate p is greater than 0.375 with $K = 0$. By shifting the operational range to the lower loss rate region with $p = 0.06$ and $p = 0.015$ for the same window range ($W < 2$), the table indicates that $K > 10$ and $K > 118$, respectively. Since $T_0 = K \times RTT$, these values of K and T_0 are too large to be accepted in any practical system.

Thus, this motivates us to relax the condition of integer α in Eq. (3), i.e. setting $\alpha < 1$. Three different choices of timeout factor K and window growth factor α are shown in Fig. 7. The curve with $(K, \alpha) = (2, 1)$ is used as the reference. By enlarging the value of K from 2 to 50 or by reducing the value of α from 1 to 0.02, we can shift the TCP operational region to a lower loss region. However, the modification of α appears to be more effective. That is, the new alpha value $\alpha = 0.02$ allows both a small average window size W and a low loss rate p. Thus, we conclude that it is better to adjust the window growth rate α than timeout factor K so as to shift the TCP operational range.

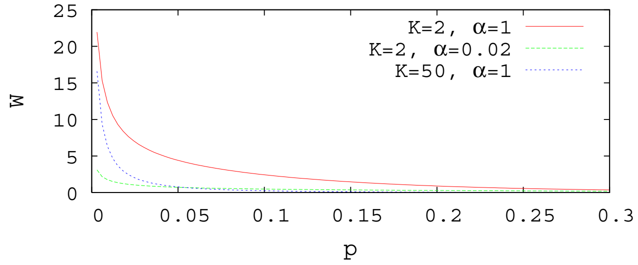


Figure 7: Comparison of the average window size with three sets of α and K values using the TCP-friendly equation.

Based on this result, 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 of one packet to the window size at every $\frac{1}{\alpha}$ round-trip-time. To be more precise, the TCP window update can be modified in the following way. Suppose that the current congestion window size is W, the TCP sender sends W packets at every round-trip-time, and

Table 2: Timeout factor (K) analysis (Eq. (4)*, Eq. (7)**, and Eq. (8)†)

	$K = 0$	$W < 4$	$W < 2$	$W < 1$
$K = 0$	N/A	$p > 0.167^\dagger$	$p > 0.375^\dagger$	N/A
$p = 0.06$	$W = 5^{**}$	$K > 1.7^*$	$K > 10^*$	$K > 26.6^*$
$p = 0.015$	$W = 10^{**}$	$K > 44^*$	$K > 118^*$	$K > 265^*$

receives W acknowledgements (ACK) during one round-trip-time from the TCP receiver. At every ACK reception, the TCP sender updates W by

$$W^{new} = W^{current} + \frac{\alpha}{W^{current}}, \tag{9}$$

where $0 < \alpha \leq 1$. A smaller value of α leads to a milder probing traffic. 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 side or network-assisted information.

Since parameter α represents the growth rate of the TCP window W at every round-trip-time (i.e. $\alpha = \Delta W$), we can say that α represents the probing traffic of TCP. The probing 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 network capacity. For wired networks whose bandwidth delay products are in the order of hundreds or thousands of packets, the relative scale of the probing traffic is small. However, for 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 2 packets, an increase of congestion window by 1 packet is equivalent to adding 50% of the total network capacity. Thus, TCP with $\alpha = 1$ can be actually too aggressive in 802.11 multihop networks, given the fact that the bandwidth-delay product of the network is low. The reduction of α makes the scale of the probing traffic more appropriate.

Finally, it is worthwhile to point out that we also allow a fractional window (i.e., $0 < W \leq 1$) as well as $W \geq 1$ without losing the generality to the 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 4 RTT by the timer. When the timer expires, the packet is sent and W is updated according to Eq. (9). A parameter W_{min} can be introduced as the lower bound of W (i.e., $W > W_{min}$) to avoid a possible long idle period for very small W. The TCP parameter `ssthresh` (slow-start threshold) also needs a lower bound (i.e., $\texttt{ssthresh} > \texttt{ssthresh}_{min}$, where $0 < \texttt{ssthresh}_{min} < 1$) so that TCP slow-start is not triggered when W is very small.

5. EVALUATION AND COMPARISONS

In this section, we evaluate the proposed scheme through simulation over a variety of scenarios and compare it to recent related schemes in ad hoc networks; ADTCP [5] and LRED [6].

5.1 Simulation Setup

We performed the NS-2 simulation using DSR protocol[14] which was the main routing protocol for the evaluation of various transport schemes. 2 to 50 long-lived TCP flows (of 200 seconds) were evaluated over different types of topologies such as chain, 7x7 grid, 13x13 grid, 13x5 grid, and mobile random topologies. The traffic patterns included the parallel, the cross and the random traffic over static and mobile topologies. The result of random traffic was obtained from the average of five runs with different random number seeds. For comparison, other transport schemes for ad hoc networks were also tested and compared in the simulation. We used ADTCP [5] as a representative transport protocol using the strategy of freezing the TCP state and window during route changes. As shown in the simulation, ADTCP tends to keep a large window despite frequent routing changes and link losses, whose behavior is good enough to be compared with our strategy of using a small window with a low window growth rate.

Explicit congestion notification (ECN) based on LRED proposal [6] was also examined to test the utility of ECN as a preventive measure for reducing the actual loss that triggers over-reactions of the on-demand routing protocol. Based on the observation that the queueing loss is negligible in 802.11 multihop networks, LRED proposed a link-layer ECN marking using the link-layer retransmission count instead of the queue length. The original LRED assumed a pre-configured manual routing, but the TCP performance improvement with commonly used on-demand routing protocols, which is of our current interest, was not addressed. 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 802.11 back-off changes on the overall performance and focus on the interaction only between transport and routing mechanisms, we kept the original back-off scheme of 802.11 MAC intact and implemented only the link-layer ECN marking of LRED without adaptive pacing in the experiments.

5.2 TCP-friendliness of the proposed scheme

In Fig. 8, we show the relationship between the average window W and the packet loss rate p, for different α values. The data were collected and organized with static routing in the same way used for the results shown in Fig. 5. With Eqs. (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 $W = 0.2W_1$ and $W \approx 0.3W_1$, respectively, under the same loss rate p. This figure shows that the shift of the operational range conforms with Eq. (1) in a networking environment free from routing dynamics and instability. The simulation result shown in Fig. 8 demonstrates that the TCP window mechanism can achieve a very low loss rate with a small average window, if a proper value of α is chosen.

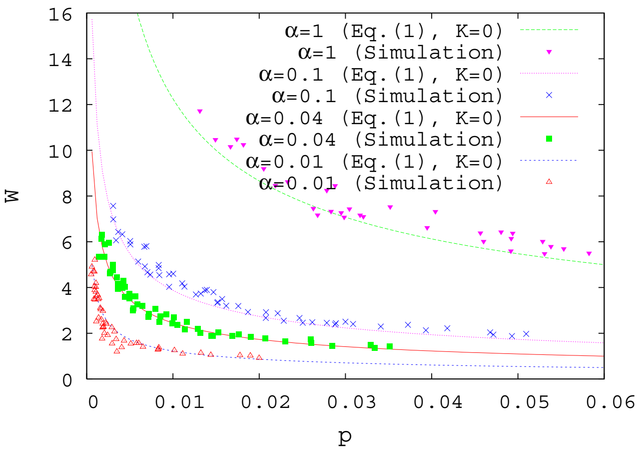


Figure 8: TCP-friendliness of the fractional window scheme with static routing.

5.3 Chain topology

The proposed fractional window scheme was tested over the 4- to 22-hop chain topologies. The TCP throughput results over networks of the chain topology under different conditions are compared in Figs. 9 (a) and (b). The TCP throughput with static routing was used as a reference of the performance bound while other curves in these two figures were obtained with the DSR routing protocol. Noticeably, TCP with $\alpha = 0.01$ (FeW) over the DSR protocol achieves a throughput that is relatively close to the performance bound by TCP with static routing. As shown, FeW (over dynamic routing) achieves drastic improvement in throughput as compared to basic TCP or ADTCP for 1 flow, with over 90% increase in throughput for end-to-end distance of 6 hops and 50% increase for 14 hops. For 4 flows the improvement reaches 90% (for 6 hops) and 85% (for 14 hops). It is worthwhile to point out that ADTCP mostly underperforms the ordinary TCP in the chain topology with a few exceptions. However, ADTCP also achieves higher throughput with a smaller α value.

The strategy of freezing the congestion window at routing changes can be found in many TCP proposals for ad hoc networks. Figs. 9 (c) and (d) show that ADTCP successfully maintains a large window despite routing changes due to packet loss events. However, the packet loss at the MAC layer due to congestion is confused with other types of packet loss such as the mobility and channel error at the physical layer. Thus, keeping a large window yields more losses, thus leading to more frequent unnecessary routing reactions. As a result, ADTCP underperforms the ordinary TCP for the same values of α in the chain topology as shown in Fig. 9 (a) and (b). The result is consistent with the observation in [17] about the TCP-ELFN [2]. Similar to ADTCP, TCP-ELFN is based on a window-freeze strategy. Its poor performance as compared with TCP over static topologies was reported in [17] since the link failure is possible due to congestion, node mobility and channel errors in 802.11 networks.

As shown in Figs. 9 (c) and (d), we see that the fractional window scheme works well for ADTCP, too. Thus, the preventive approach using the proposed fractional window scheme is effective and suitable for a wide range of window-based transport protocols. With respect to the on-demand ad hoc routing protocols, the chain topology is the most resource-limited case that offers no alternative path so that its impact on the TCP performance can be clearly observed. It is quite common to have chain-like topologies in ad hoc and mesh networks where deployment is not highly dense in the whole network. When a wireless network does not have alternate routes for some of its pairs, chain-like sub-topologies appear in which performance improvement is most needed. 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 data transmission. For a small value of W, the TCP sender actually sends one or two data packets at a time in the simulation of Fig. 9 while the TCP with $\alpha = 0.01$ behaves like a stop-and-go protocol.

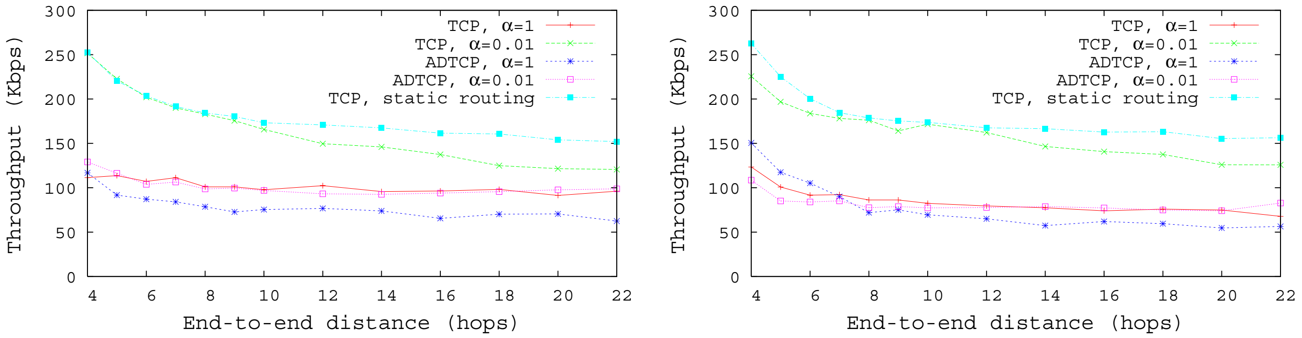
Finally, the behaviors of the TCP and ADTCP windows over time are shown in Figs. 9 (e) and (f), respectively, for the case of the 7-hop chain topology. With standard TCP of $\alpha = 1.0$, the congestion window variation is far from the ideal AIMD pattern of congestion avoidance. This is similar to the observation in Fig. 3. Frequent overshooting beyond the bandwidth-delay product of the network brings unreliable network feedback as a result of unstable routing dynamics. In contrast, as shown in Fig. 9 (e), TCP with $\alpha = 0.01$ leads to a more regular AIMD pattern of the congestion window whose average value is close to the optimal small window size in [7, 6]. As to ADTCP, the congestion window with $\alpha = 0.01$ is still smaller than that with $\alpha = 1$.

These above simulation results indicate that the basic TCP mechanism can keep track of an appropriate window size regardless of the bandwidth-delay product of the network if a proper α value is adopted. With milder aggressiveness (e.g., $\alpha = 0.01$), TCP can achieve a smaller average window with a lower loss rate and better performance in 802.11 multihop networks. For α close to 1, TCP drives the window size out of the proper level in 802.11 multi-hop networks, and then the system performance degrades significantly. Note that FeW scheme consistently outperforms legacy TCP at higher data rate (*e.g.*, 54 Mbps). It is because the connection blackout cycle exists regardless of the data rate [18].

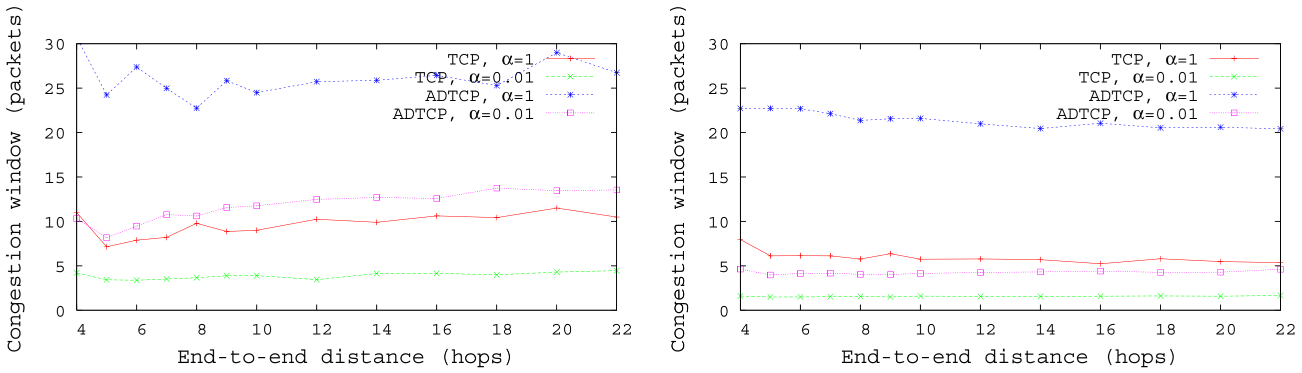
5.4 Grid topology and node mobility

The grid topology has much more redundancy that allows alternative routes and back-up resources than the chain topology. In this subsection, we tested the proposed fractional window scheme over the grid topology to see possible performance improvement. We used a small scale (7x7), a large scale (13x13) and a narrow-stripe (13x5) grid topology based on the grid topology shown in Fig. 1 (a) for the simulation. Fig. 10 shows some of the tested traffic pattern over the narrow-strip grid topology.

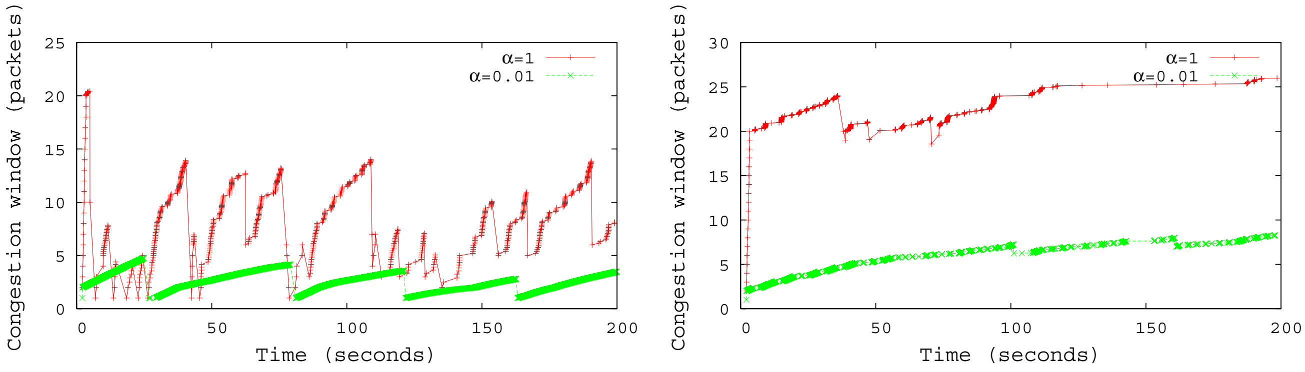
The results for the 7x7 grid are shown in Fig. 11 (a), where the total throughput of TCP with many flows is noticeably higher with $\alpha = 0.01$ than that with $\alpha = 1$. Even though the two-flow cases (overlap and cross) have a slight throughput loss with $\alpha = 0.01$, the loss is at a tolerable level considering the gain with respect to more flows. Also, we see that ADTCP hardly has any improvement with fewer flows and ADTCP fails to make improvement w.r.t. TCP with more flows. Nevertheless, the proposed fractional window scheme improves the performance of ADTCP as well as



(a) Throughput (1 flow)   (b) Throughput (aggregation of 4 flows)

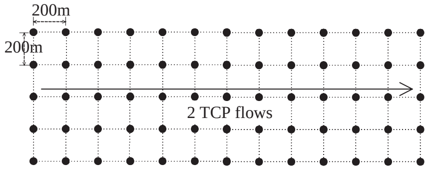


(c) Average congestion window (1 flow)   (d) Average congestion window (4 flows)

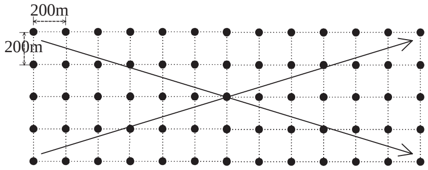


(e) Congestion window over time (TCP, 2 flows)   (f) Congestion window over time (ADTCP, 2 flows)

Figure 9: Performance comparison of different TCP schemes in a chain topology.



(a) 13x5 grid (overlapping traffic)



(b) 13x5 grid (cross traffic)

Figure 10: Examples of the traffic patterns for grid topology simulation.

TCP. This confirms our claim that the proposed fractional window scheme is applicable to a wide range of end-to-end adaptations using the TCP window mechanism.

The results for the 13x13 grid are shown in Fig. 11 (b). We observe a similar trend of the TCP throughput as that for the 7x7 grid, with less noticeable improvement. Since the network is more crowded with 14 flows than 8 flows, the TCP throughput ($\alpha = 1$) with 14 flows is slightly lower than that with 8 flows. However, TCP with $\alpha = 0.01$ maintains the maximum throughput achieved by 8 flows even in a more crowded setting with 14 flows.

Interestingly, when the flow number is equal to 2, TCP and ADTCP with $\alpha = 0.01$ do not perform as well as those with $\alpha = 1$. This can be explained as follows. The advantage and disadvantage of the fractional window scheme comes from less aggressive TCP traffic with a smaller α value. When the network has a higher capacity, the impact of probing traffic is less severe and can be absorbed by the network without affecting the overall network stability. Under this scenario, an aggressive adaptation strategy can be a good choice with fewer flows in a resource-rich environment. However, a real world system tends to use more network resources, and it is actually rare to have a resource-rich network in the real world. Similarly, ADTCP's strategy of window-freezing helps provide a performance improvement over TCP with fewer flows (say, 2 flows) in a resource-rich environment. However, this strategy does not help in a more crowded network with 4, 8, and 14 flows. In the latter case, the proposed fractional window increment scheme improves the performance of ADTCP and TCP.

We also tested the 13x5 grid topology to represent a more common network scenario which is between the resource-limited chain topology and the resource-rich grid topology. The results are shown in Fig. 11 (c). We observe a similar trend of throughput improvement as found in other topologies, yet with little or no performance degradation for the 2-flow scenario.

Finally, we evaluated a mobile scenario based on the random waypoint model with maximum speed 10m/s and no pause time. The results with random traffic among 150 mobile nodes in 2000m by 2000m space yield the similar trend of performance improvement of aggregate TCP throughput as shown in Fig.11 (d), with over 35% throughput improvement for 32 flows. This is because the link failure with congestion over multiple hops is more frequent and persistent than those due to node mobility and channel errors during the entire course of the connection lifetime. In other words, for 802.11 multi-hop networks, the effect of the fractional window increment scheme is more noticeable over the mobility adaptation effect. Mobility scenarios were the design focus of the ADTCP protocol.

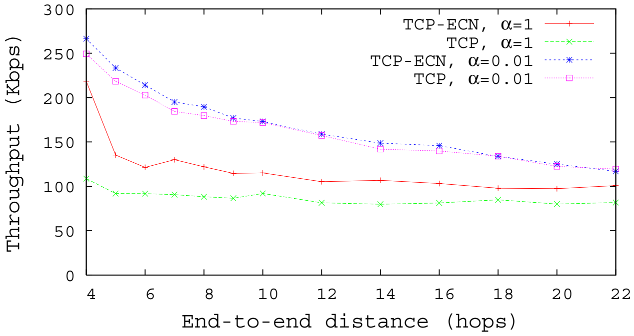
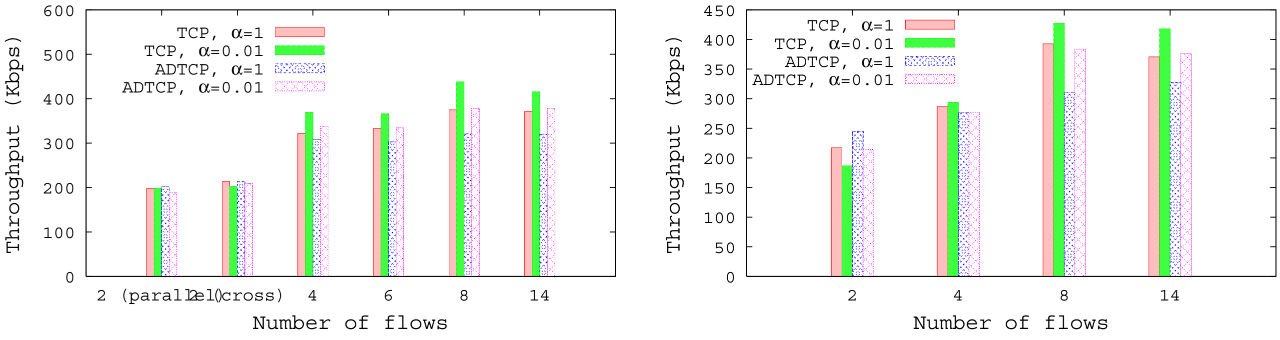


Figure 12: Effect of ECN on TCP performance in the chain topology (2 flows).

5.5 Utility of ECN

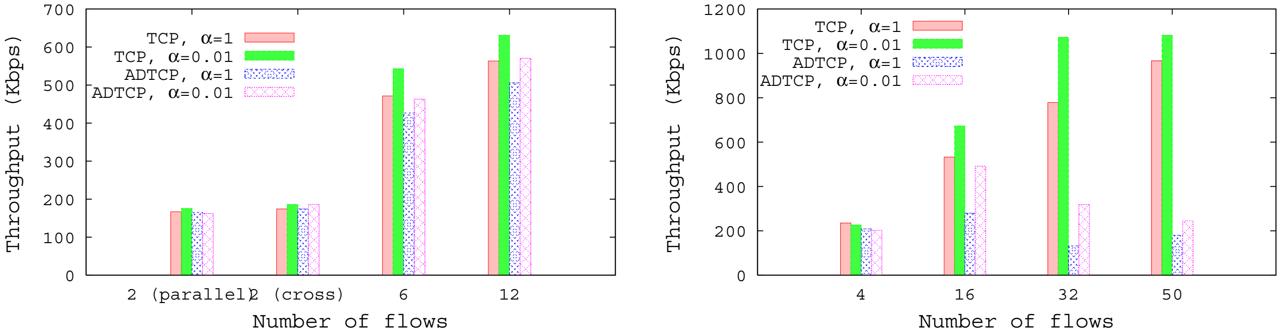
We implemented the link-layer ECN marking based on LRED [6] with NS-2 and evaluated the ECN effect with two TCP flows over the chain topology. Fig. 12 shows the aggregate throughput of TCP and TCP-ECN for different α values. With $\alpha = 1$, TCP-ECN makes a noticeable difference in the average throughput. However, its performance is still poorer than that of TCP with $\alpha = 0.01$ and no ECN. FeW (with or without ECN) achieves 70% improvement over TCP-ECN for end-to-end distance of 6 hops, 50% improvement for 10 hops and 45.5% improvement for 14 hops distance. Even though ECN can be used as a part of the preventive measure to reduce the over-reaction of routing protocols, the result in Fig. 12 indicates that the TCP growth rate (i.e., aggressiveness of the probing traffic) has more impact on the behavior of routing protocols in 802.11 multi-hop networks. One way to explain this phenomenon is that, despite that ECN can reduce the frequency of actual losses before MAC reaches the maximum retransmission, TCP with $\alpha = 1$ remains to be the severe network stress factor affecting MAC retransmission. The usage of smaller α can reduce such network stress in the link layer more effectively.

We see from Fig. 12 that, with $\alpha = 0.01$, TCP-ECN performs slightly better than TCP. This means that ECN can be integrated with the fractional window increment scheme. However, the loss rate is so low with a small α value, the performance gain due to ECN is hardly noticeable. Thus, controlling the window growth rate provides a very powerful preventive tool of reducing unnecessary routing dynamics.



(a) 7x7 grid

(b) 13x13 grid



(c) 13x5 grid

(d) Mobile random

Figure 11: The TCP throughput in grid and mobile random topologies.

6. CONCLUSION AND FUTURE WORK

Improving the performance of TCP over 802.11 multi-hop ad hoc networks is truly a cross-layer problem. In this work, we studied the interaction between the transport and the on-demand routing protocols, and showed that the utility of TCP in 802.11 multihop networks lies in the ability to keep the network robust and stable. The contributions in this paper are manifold. In particular, we:

- thoroughly examined and quantified the interaction between TCP, on-demand ad hoc routing and wireless 802.11 MAC protocols, illustrating the significance of such interaction,
- analytically investigated the TCP operational range and used the insight from that investigation to propose the fractional window increment (FeW) scheme,
- evaluated our proposed scheme over a variety of scenarios and showed significant improvement over legacy TCP,
- compared our proposed scheme to recent related work (namely, ADTCP and LRED) showing the significant advantages gained by using the FeW scheme.

Even though 802.11 multihop networks impose many technical challenges on TCP, it can be concluded from the current study that the TCP window mechanism actually provides a good solution to 802.11 networks with a proper α value. In the near future, we would like to continue to investigate the cross-layer interaction for various mobility and channel error models.

7. REFERENCES

[1] J. Padhye, V. Firoiu, D. Towsley, and J. Krusoe, “Modeling TCP throughput: A simple model and its empirical validation,” in *Proc. ACM SIGCOMM ’98*, Vancouver, CA, September 1998, pp. 303–314.

[2] G. Holland and N. Vaidya, “Analysis of TCP performance over mobile ad hoc networks,” in *Proc. ACM Mobicom ’99*, Seattle, WA, 1999.

[3] 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.

[4] J. Liu and S. Singh, “ATCP: TCP for mobile ad hoc networks,” *IEEE J-SAC*, vol. 19, no. 7, pp. 1300–1315, 2001.

[5] 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 communicatioins*, vol. 21, no. 10, pp. 1615–1626, Decemeber 2004.

[6] 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.

[7] 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.

[8] 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.

[9] K. Chen and K. Nahrstedt, “Limitations of equation-based congestion control in mobile ad hoc networks,” in *Proc. IEEE WWAN*, Tokyo, Japan, March 2003.

[10] 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.

[11] K. Chen, Y. Xue, and K. Nahrstedt, “On setting TCP’s congestion window limit in mobile ad hoc networks,” in *Proc. IEEE ICC 2003*, Anchorage, Alaska, May, 2003.

[12] K. Sundaresan, V. Anantharaman, H. Hsieh, and R. Sivakumar, “ATP: a reliable transport protocol for ad-hoc networks,” in *Proc. ACM MobiHoc ’03*, Annapolis, ML, 2003.

[13] 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.

[14] D. Johnson, D. Maltz, and Y. Hu, “The dynamic source routing for mobile ad hoc networks,” IETF Internet Draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt, July 2004.

[15] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc on-demand distance vector (AODV) routing,” IETF Internet Draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-13.txt, February 2003.

[16] 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),” *IEEE Computer Magazine*, vol. 33, no. 5, pp. 59–67, May 2000.

[17] J. Monks, P. Sinha, and V. Bharghavan, “Limitations of TCP-ELFN for ad hoc networks,” in *Proc. IEEE MOMUC ’00*, Tokyo, Japan, 2000.

[18] K. Nahm, “On the scale of probe traffic for Internet transport protocols,” Ph.D. dissertation, University of Southern California, Los Angeles, CA, May 2005.