Ad Hoc Networks 10 (2012) 586–603

Contents lists available at SciVerse ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Analysis of TCP performance on multi-hop wireless networks: A cross layer approach ☆

Adnan Majeed [a,*], Nael B. Abu-Ghazaleh [a], Saquib Razak [b], Khaled A. Harras [b]

[a] Dept. of Computer Science, State University of New York, Binghamton, United States
[b] Dept. of Computer Science, Carnegie Mellon University, Qatar

ARTICLE INFO

Article history:
Received 25 May 2010
Received in revised form 24 August 2011
Accepted 24 October 2011
Available online 17 November 2011

Keywords:
Wireless mesh networks
MAC interactions
TCP

ABSTRACT

In Multi-Hop Wireless Networks (MHWNs), wireless nodes cooperate to forward traffic between end points that are not in direct communication range. Specifically, traffic is forwarded from a source towards its destination through intermediate nodes that form a wireless multi-hop *chain*. Researchers have studied the performance of TCP over chains discovering properties such as how the number of hops reduces chain throughput as neighboring links contend for the shared medium. Moreover, the presence of hidden terminals has also been shown to negatively affect performance of example chains. In this paper, we leverage recent characterization of how competing wireless links interact to develop an in-depth analysis of TCP performance over wireless chains. In particular, there are a number of possible modes of interference between competing links with distinct implications on performance and fairness; to our knowledge, this is the first work that studies the impact of these different modes on TCP chain performance. We classify chains according to interference modes considering both the forward (data) and reverse (acknowledgment) traffic. Chain geometry limits the types of chains that arise most frequently in practice. We evaluate TCP performance over the most frequently occurring chain types and observe significant performance differences between chains that have the same hop count. Different four-hop chains, for example, show a throughput difference of up to 25% and a retransmission overhead difference of over 90%. We discuss the implications of these differences on network performance: specifically, route instability and bandwidth usage generated. We extend this analysis to two single-hop TCP flows and quantify the effect of interference interactions between two flows. This study is a first step towards completely understanding the performance of multiple TCP flows over multiple hops in a MHWN.

1. Introduction

Multi-Hop Wireless Networks (MHWNs) play an increasingly important role at the edge of the Internet. MHWNs allow ad hoc connectivity where wireless infrastructure is unavailable or expensive. Community mesh networks [1–3], many sensor networks [4], ad hoc networks [5] and vehicular networks [6] represent examples of MHWNs. Traffic in these networks traverses multiple hops in cases where the source and destination are not within direct transmission range. The sequence of nodes that forms a connection between a source and a destination is called a *path* or a *chain*;

☆ An earlier version of this work appeared in the 8th International Conference on Ad Hoc Networks and Wireless. The extensions in this paper are (i) a refinement of occurrence probabilities of different types of chains in a random network and (ii) an analysis of the effect of MAC interactions on two single-hop TCP flows. The latter extension is an important step towards understanding MAC interaction effects on performance in a MHWN carrying numerous multi-hop TCP flows. In the analysis of two TCP flows we also show that, unlike previous observations, in some cases unfairness can arise even in the presence of symmetric MAC interactions.

* Corresponding author.
E-mail addresses: adnan@cs.binghamton.edu (A. Majeed), nael@cs.binghamton.edu (N.B. Abu-Ghazaleh), srazak@cmu.edu (S. Razak), kharras@cs.cmu.edu (K.A. Harras).

1570-8705/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2011.10.005

it represents a fundamental communication structure in MHWNs. Understanding how current protocols behave when deployed over chains is critical to understanding and improving MHWN performance. Amongst these protocols, TCP is especially important due to its widespread use on the Internet.

Several studies have explored the performance of TCP over wireless chains [7–11]. A general observation of these works is that the overall throughput drops with increasing number of hops as neighboring links of the chain must share the available bandwidth; each packet must be retransmitted along the chain links consuming bandwidth. Subsequent studies improved the analysis by considering the presence of hidden terminals [12,13] which lead to collisions and dropped packets. These studies also consider a single chain (with a hidden terminal) and analyze its performance.

Recently, Garetto et al. have shown that two interfering links can interact in a number of different ways, each exhibiting substantially different performance and fairness characteristics [14,15]. For example, an asymmetric hidden terminal (one link causing collisions at the other but not vice versa) behaves differently from a symmetric hidden terminal (collisions occur at both interfering links). The ability of one or both links to capture packets [16] leads to yet other substantial difference in how they behave. Similarly, hidden terminals can occur among the ACK packets, or between ACK and data packets, symmetrically or asymmetrically. This paper leverages this analysis to provide a detailed analysis of TCP performance over wireless chains. We show that prior works have analyzed only one of the several possible chain types (e.g., the topology studied by the papers that consider hidden terminals [7,13,17] represents one type of many possible chain types and accounts for only 14% of chains occurring in our experiments). We show that these different chain types exhibit substantially different performance; a complete analysis of TCP over chains must take into account these possible chain types.

This paper presents a detailed analysis of TCP performance over wireless chains that for the first time accounts for the different modes of interference interactions between the chain links [14,15]. We discover that a large number of possible types of chains, based on the nature of the interference among the chain links, occurs in practice. The first contribution of the paper is a classification of the types of three and four-hop TCP chains and an analysis of the frequency of their occurrence under uniform node distribution and practical routing protocol assumptions. The second contribution of the paper is an analysis of the performance of the most commonly occurring chains under a number of operating conditions. We observe significant performance differences between chain types that have the same number of hops. In particular, in four hop chains, we observe a throughput difference of up to 25% and a retransmission overhead difference of over 90%. The analysis also provides insights into the behavior of TCP over MHWNs that can be used to improve performance at higher layers. For example, routing protocols may consider the chain type when selecting routes. Finally, we extend this analysis to two single-hop TCP flows. The effects observed for a single TCP flow extend to two single-hop flows but with more severe influence on performance. Being aware of these observations can enable design of protocols that improve network performance.

The remainder of the paper is organized as follows. In Section 2 we discuss related work. In Section 3 we discuss the different types of interactions that occur; we also define naming conventions for chains and present the measured occurrence percentages of the different types of chains. Section 4 evaluates the performance of TCP over the types of chains that occur most frequently. In Section 5 we discusses the effect of chain behavior on network performance. Section 6 extends our study to multiple TCP flows and shows that our findings for single TCP flow carry over to multiple flows with more severe performance effects. We present conclusions and briefly discuss future work in Section 7.

2. Related work

Several researchers have studied the behavior of chains in MHWNs. Li et al. [18] examine the performance of chains as the number of hops are increased and study the effect of cross-interference between chains. They analyze the effect of MAC 802.11 behavior on the performance of multi-hop chains. However, they do not categorize interference patterns that govern network performance. Ng and Liew [19] present a hop by hop analysis of a multi-hop chain, study the impact of hidden terminals on the throughput of chains, and present a quantitative approach towards estimating this throughput. They show that packet drops caused by hidden terminals lead to decreased throughput and route instability.

Razak et al. [20] have studied the impact of interference interactions on chains under saturated UDP traffic. They develop a systematic methodology for determining the interference interactions that are possible in 3 and 4 hop chains. They study the influence of these interference interactions on chain performance and extend their work to analyze chains of n hops. Other research [21] has conducted some preliminary analysis on the effect of interference interactions on the performance of bi-directional flows. This research, however, does not consider the effect of interference interactions on TCP performance over chains. TCP introduces several factors like bi-directional traffic, congestion control and round trip time estimations for timeout prediction. In this paper, we show that interplay between these factors and interference interactions of chain links greatly impacts TCP performance.

Previous work has identified that TCP does not perform well over wireless chains [7–10]. Researchers that analyze TCP performance over wireless chains show that TCP throughput decreases as the number of hops in the chain are increased [11]. The understanding of TCP performance over chains is further refined by works studying the impact of Hidden Terminals (HT) [22] on TCP performance [12,13]. HT causes collisions between data packets, and between data and ACK packets [13,23] resulting in a drop in TCP throughput. Researchers, therefore, have suggested various parameter settings to improve the performance of TCP over wireless chains. These settings include using fewer ACK transmissions [17,24,25], statically setting the TCP congestion window to a small value to avoid contention [9,10,23], putting in mechanisms that adaptively stabilize the congestion window to the optimal value [9,26], or using a particular TCP flavor that performs better over wireless chains [11]. Despite their contributions, however, prior works do not completely explain TCP performance over wireless chains. This is because they do not take into account the detailed characterization of how links in a chain interact [14,15].

Recent studies [14,15,27] show that links in a chain can interfere in a number of different ways with substantial implications on performance and fairness. For example, asymmetric hidden terminals (one link causing collisions at the other but not vice versa) behave differently from symmetric hidden terminals (collisions occur at both interfering links). The ability of one or both links to capture packets [16] leads to yet another substantial difference in how they behave. Our work shows how this link-level asymmetry and capture impacts TCP performance over chains.

Prior work on TCP performance over wireless chains either ignores the impact of these interference interactions or simply looks at example chains with one hidden terminal. For example, the topology studied by [7,13,17] represents one type of chain which accounts for only 14% of the chains occurring in a random MHWN. Therefore, TCP performance on over 80% of the chain types that occur in a network remains unanalyzed. This paper is the first to analyze TCP chains taking all these different modes of interference interactions [14,15] into account; in doing so, this paper also quantifies the effects of different types of asymmetry on TCP performance: packet size asymmetry, queue size asymmetry, capture asymmetry and interference asymmetry.

3. Chains in a MHWN

In this section we identify the possible interactions that occur in a chain carrying a bi-directional flow. We first present some general terminology used throughout the paper. We then describe the prominent types of link interactions possible in a chain in the presence of bi-directional TCP flows: carrying data and acknowledgment (ACK) packets. Finally, we determine the types of chains that are possible in a network and calculate occurrence probabilities for each of these chains. The analysis of these chains is the basic building block that enables generalizing this study for arbitrarily long chains. This generalization is part of our future work.

3.1. Terminology

Wireless networks use Carrier Sense Multiple Access (CSMA) protocols to share the wireless medium. IEEE 802.11, which is based on Carrier Sense Multiple Access (CSMA), is the most commonly used channel access protocol in wireless data networks. CSMA based protocols try to prevent multiple nodes from transmitting simultaneously. This is done by requiring each sender to sense the channel and transmit only after the channel has been idle for a specific amount of time. Since the channel is sensed around the sender, it is possible for the state of the channel to be different at the receiver. Hence, packets transmitted by a node that senses the channel to be idle may be lost due to collision if the medium around the receiver is busy. These collisions are detrimental to the performance of a wireless network. This is known as the *hidden terminal* problem [22].

Signal strength in wireless transmissions attenuates with distance and other environmental factors. A node can successfully receive a packet if the Signal to Interference and Noise Ratio (SINR) of a packet is above a certain threshold. That is, a node can successfully receive a packet in the presence of overlapping transmissions if the SINR is above the reception threshold. This is called the *capture effect* [16].

A chain is a sequence of nodes that forwards messages to enable communication between a source and destination that are not within transmission range. Two consecutive nodes in the chain can communicate with each other and can exchange packets with each other in both directions. We refer to this communication between two nodes as a *flow* and the communication channel is referred to as the *link*. Links that do not share a node can be active simultaneously and can affect each other at the MAC level; we refer to these effects as interactions. The location of the interacting links and the direction of flow on each link defines the interaction. The kind of effect these links have on each other defines the interaction type.

3.2. Link interactions

In multi-hop wireless networks, two interfering links can interact in several different ways. Given two source–destination pairs S1–D1 and S2–D2, there exist four secondary links: S1–S2, S1–D2, S2–D1, and D1–D2 that can interfere with each other. Although there can be a large number of combinations for these interactions [14,15], we discover that only a limited number of these interactions are possible between links of a chain. We summarize these interactions in Fig. 1, and briefly describe each one.

1. *Senders Connected (SC):* In SC interactions, the sources of the two links are within carrier sense range. CSMA prevents these senders from concurrent transmissions; and no collisions other than those arising when the two senders start transmission at the same time will occur. These collisions are low in probability and we refer to them as synchronized collisions. SC interaction allow equal share of the channel between the senders.
2. *Hidden Terminal (HT):* In this interaction, the source for one link is a hidden terminal for the destination of the second link but not vice versa. Both senders are disconnected and can transmit simultaneously. One destination drops its packets because of interference from the opposite sender. The other destination can successfully receive its packets because it does not experience any interference. In this interaction, one link will always be successful at obtaining maximum throughput, while the second link will experience frequent collisions and hence obtain a low throughput.

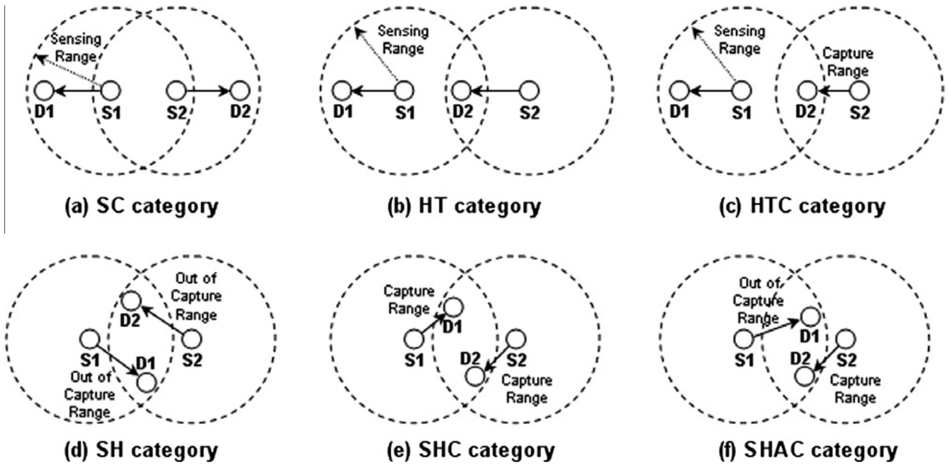


Fig. 1. The dominant types of link interactions observed in a MHWN.

3. *Hidden Terminal with Capture (HTC):* This interaction is similar to HT interaction but the destination that experiences interference is able to capture its packets. Assume that the links are S1–D1 and S2–D2 and that S2–D2 is the weak link, i.e. the link that suffers losses due to the interaction. When S1 starts transmitting before S2, D2 starts receiving the signal from S1. Therefore, when S2 starts transmitting, the signal from S2 is treated as interference at D2 and it causes a collision. However, when S2 starts transmitting first, D2 starts receiving this signal. In this case the signal from S1, which starts transmitting after S2, is not strong enough to cause a collision at D2. Therefore, whenever S2 starts transmitting before S1, D2 successfully receives the packet from S2. This decreases the number of collisions compared to HT case. Therefore, the weak link has better throughput in the HTC interaction, compared to the HT interaction [16].
4. *Symmetric Hidden Terminal (SH):* In this interaction, both links experience hidden terminals as both sources interfere with opposite destinations. Both links will experience collisions as the senders transmit concurrently causing throughput to be severely affected.
5. *Symmetric Hidden Terminal with Capture (SHC):* This interaction is similar to SH interaction except both links are able to capture their packets under interference from opposite sender. The capture effect is explained in the HTC interaction above. In this interaction, simultaneous transmissions will be successful at one link (the link whose sender starts first) and will fail on the other link. This interaction will suffer from short term unfairness. Overall throughput for both links would be similar to SC interactions except at a cost of much higher number of re-transmissions.
6. *Symmetric Hidden Terminal with Asymmetric Capture (SHAC):* This interaction has hidden terminal between both links. However, only one destination is able to capture its packets.

It should be noted here that symmetric interactions occur between links carrying data traffic in forward direction and links carrying ACK packets back to the source. Because of the geometric restrictions of chains, there are no symmetric interaction between links carrying unidirectional flows.

3.3. Chain interactions

In this section we identify the different chain types, develop a naming convention for them and present their occurrence probabilities in random networks. Fig. 2 illustrates the link interactions that we study in three and four hop chains. Each row in the figure has an interaction number along with the link pair in that interaction, the two links that give rise to the interference interaction. The arrows in each row identify the link pair considered in that interaction; the arrow direction depicts the flow direction considered. For the TCP flow running on a chain, node 0 is the source and node 3 and node 4 are the destinations for three and four hop chains respectively. So all flows towards the right carry TCP data packets, referred to as forward flow or forward direction in this paper, and all flows towards the left carry TCP ACK packets, referred to as backward flow or backward direction.

Razak et al. [15] have determined that, between any two links, there can be 10 different interference interactions, or interaction types, possible. This means that there can be 10^4 types of three hop chains; three interacting link pairs (i1–i4) and 10 interaction types for each link pair. In a four-hop chain, the possibilities are even more overwhelming. There can be 10^{12} types of four hop chains; 12 pairs of interacting links (i1–i12) each with 10 possible interaction types. Fortunately due to geometric limitations, in reality there are only 3 types of three-hop chains and 8 types of four-hop chains that occur most commonly.

We develop a nomenclature for identifying different chain types. Each chain is represented as F/B/C. F is the prominent interaction between links carrying the forward traffic (data). B is the prominent interaction between links carrying backward traffic (ACK), C is the prominent interaction between cross links, i.e. between a link carrying forward traffic (data) and a link carrying backward traffic

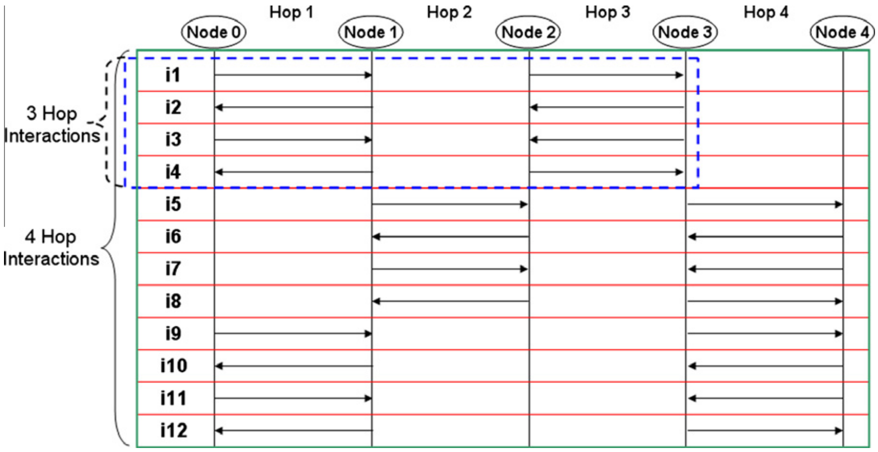


Fig. 2. All the interactions in a three and four hop chain. Each row shows the interaction between two specific links in specific directions.

Table 1
The types of four-hop chains that occur most frequently in a MHWN. The last three columns depict the interaction number of the dominant interaction (refer to Fig. 2) in the forward, backward and cross flows, respectively. A dash (–) means that no interaction dominates in that direction. Note that current routing protocols cannot differentiate between these chains even though they are significantly different.

Occurrence percentage	Chain type	Dominant interaction		
	Forw/back/cross	Forward	Backward	Cross
5.9	HT/SC/SHAC	i9	–	i3
7.7	HTC/HTC/SHAC	i9	i10	i7
14.2	HTC/HTC/SHC	i9	i10	i3
6.6	HTC/SC/HTC	i9	–	i11
7.8	SC/HT/HTC	–	i10	i11
3.6	SC/HTC/HTC	–	i10	i11
0.02	SC/SC/SC	–	–	–
41.7	SC/SC/SHC	–	–	i11

(ACK). A prominent interaction, as used in the above naming, is defined as the interaction number (from Fig. 2) that has the most effect on the performance of a chain in a particular direction, forward, backward or cross. Table 1 illustrates the use of this naming convention. The table shows four-hop chains that occur most frequently and states the interaction number of the prominent interaction in the forward, backward and cross flow. Previous works that study TCP performance over chains where each hop is 200 m long [7–9,13,17] only study the HTC/HTC/SHC chain. This chain type accounts for slightly over 14% of all four-hop chains that occur in a MHWN. Therefore, these works do not account for over 80% of the types of chains that occur in a MHWN. The table showing the same information for three hop chains is omitted due to space constraints.

We use simulations to determine the most frequently occurring chains and calculate their occurrence probability. We consider a 1500 m $\times$ 1500 m network with 500 nodes, randomly placed with uniform distribution. For routing, we use our implementation of NADV [28], which uses a greedy protocol to pick the best next hop at each node. The metric for selecting the hop is the product of link quality and how close this next hop gets a packet to the destination. We use this routing protocol to generate all possible three and four hop chains in random networks. We classify the chains according to their type, using the F/B/C convention described above. The percentage occurrence of a particular chain type is calculated using the ratio of the number of chains of that type that occur and the total number of chains, of the same length, that occur in the network. Using these percentages we identify the most frequently occurring chains. Fig. 3 shows occurrence probabilities of chain types for three and four hop chains as the carrier sense range is varied.

To avoid clutter in the figures we only show a subset of the chains that occur. This subset consists of the chains that occur most frequently in a random network given a realistic carrier sense to transmission range ratio. Most commercial radios set carrier sense range to slightly over twice the transmission range, and this is what we use as a realistic carrier sense to transmission range ratio. As the carrier sense range is increased more pairs of nodes in the network become sender connected until (as depicted in Fig. 3b) the carrier sense range is high enough to make all nodes in the network sender connected and the only chain type remaining is SC/SC/SC.

4. TCP evaluation over chains

In this section we evaluate, via simulation, the behavior of a TCP flow on different chains that occur in a MHWN. We start with three hop chains; the smallest chains that have the interactions we consider. We then evaluate TCP

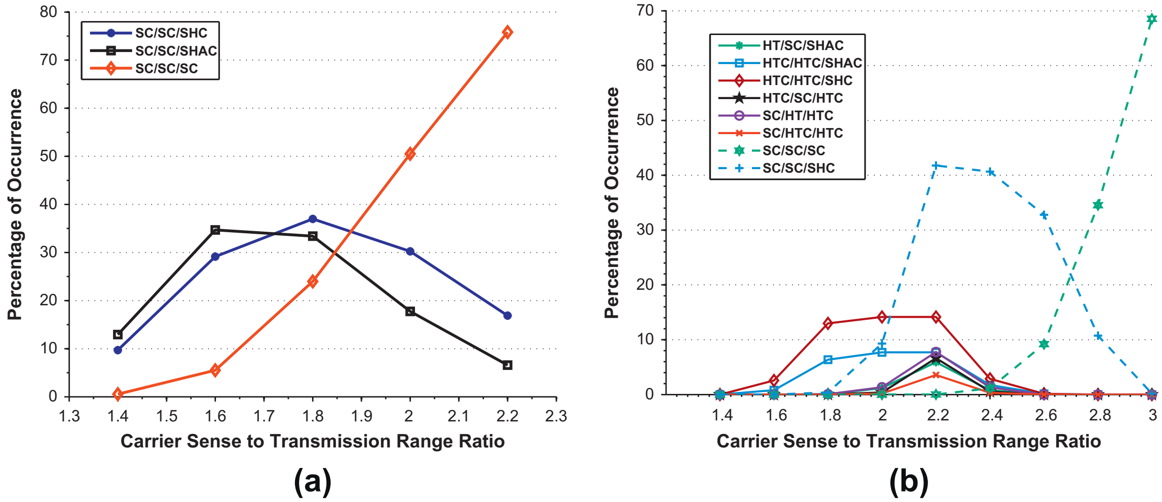


Fig. 3. Occurrences of different types of chains: (a) shows the occurrence probability of 3-hop chains, (b) shows the occurrence probability of 4-hop chains.

Table 2
Simulation parameters.

Parameter	Value
Application layer protocol	FTP
TCP version	Tahoe
TCP packet size	1040 bytes
TCP ACK packet size	40 bytes
Max TCP window size	20 packets
Routing	Static
Mobility	None
RTS/CTS	Disabled
MAC retries	7
Data rate	2 Mbps
AutoRate feedback	Disabled
Simulation time	65 s

performance on the four-hop chain types discussed in the previous section. Finally, we discuss the effect of varying parameters on the performance of different chain types. Analyzing TCP behavior over the chains discussed in the previous section is an important step towards understanding TCP performance over general n-hop chains.

We use ns-2 release 2.33 to perform this analysis. The simulation parameters used are shown in Table 2. All simulations use these parameters, unless stated otherwise. We use static routes as they ensure that the performance evaluation of TCP over MHWN is isolated from routing layer effects. We use TCP Tahoe at the transport layer. However, we show that the effects observed are the same for TCP Reno and TCP Vegas also. We vary some of the parameters shown in Table 2 and observe the effect on performance of different chain types in Section 4.3. Specifically, we vary TCP congestion window size, data rate, RTS/CTS setting (enabled) and delayed ACK setting (enabled).

4.1. TCP on three hop chains

This section discusses the performance of TCP over three hop chains. Refer to Fig. 2 for the numbers and locations of the nodes and hops used in the following explanation. Fig. 4a illustrates that the SC/SC/SHAC chain has approximately 25% lower throughput compared to the SC/SC/SC and SC/SC/SHC chain. In the SC/SC/SHAC chain links carrying backward flow, i.e. ACK traffic, have an SHAC interaction with links carrying forward flow, i.e. data traffic. This interaction causes unfairness towards the forward flow on Hop1. Our experiments reveal that all such cases, where a link carrying backward flow asymmetrically causes drops at a link carrying forward flow, behave similar to each other.

We explain this behavior by a detailed discussion of how the SHAC interaction acts in an SC/SC/SHAC chain. Nodes 0 and 3 can transmit concurrently since they are not sender-connected. Every time both these nodes transmit concurrently there is a collision at node 1 since it is in capture range with both nodes. However, a collision at node 2 occurs only when node 0 starts transmitting before node 3. When node 3 starts transmitting before node 0, node 2 is able to capture the packet [16]. In this interaction, the link between node 0 and node 1 is the *weak* link, i.e. it faces unfairness in terms of collisions. Fig. 4b shows that this unfairness causes the collision drops for TCP data traffic at Hop1 for the SC/SC/SHAC chain to be significantly higher than the other two types of chains. This translates to a high retransmission overhead as depicted in Fig. 4a.

At the MAC layer this effect causes the forward going link to drop packets anytime hops Hop1 and Hop3 are active simultaneously. The drops cause duplicate ACKs to be sent and reduces the congestion window to one. In the mean time, the ACK traffic acquires enough channel access to empty the chain of ACKs, eventually reducing the traffic in the chain. This reduction allows the sender to obtain access to the medium and finally resume transmitting successfully. This phenomenon keeps repeating, causing inefficient use of the medium. This pattern results in the 25% lower throughput for the SC/SC/SHAC chain. In Section 6 we observe that in the presence of two flows, and hence two sources, the flow control described above cannot take place. Therefore, the flow with the weak link looses packets indefinitely.

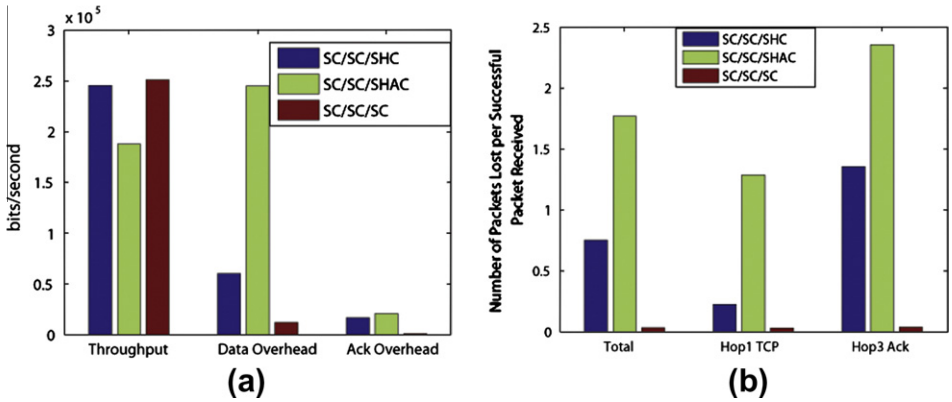


Fig. 4. Performance of three-hop chains. (a) Throughput achieved by each chain and the retransmission overhead of data and ACK packets. (b) MAC collisions in each chain and how many of these are data packet drops at Hop1 and ACK packet drops at Hop3.

In the case of the SC/SC/SHC chain, the SHC interaction is also between Hop1 carrying data traffic and Hop3 carrying ACK traffic. However in this case the interaction is symmetric and which ever of node 0 and node 1 starts transmitting first has a successful transmission. Hence, the collision overhead is much lower at Hop1 in this case causing comparatively better throughput and lower retransmission overhead.

The SC/SC/SC chain is the best in terms of performance and efficiency. It has the lowest overhead since all the sources are sender connected and the medium arbitration is synchronized. The only collisions in this case are synchronized collisions. These collisions are caused when two nodes that are sender connected are ready to transmit a packet, sense the medium as idle and then transmit at the same time. Such collisions occur infrequently [29].

The above explanation shows that running TCP over chains that are not sender connected, but have symmetric interactions, give approximately the same throughput as sender connected chains. However, as demonstrated by the higher overhead in Fig. 4a, they generate much more traffic to give the same throughput. The retransmission overhead of SC/SC/SC chains is less than 15% of the overhead of SC/SC/SHC chains. This is undesirable due to the effect this behavior can have on other flows in the network. This effect is discussed in more detail in Section 5.

The performance of a chain carrying TCP traffic worsens if the interaction types in the chain are both asymmetric and not sender-connected. These chains have a large number of collisions, leading to excessive retransmissions, leading to further collisions. This translates to a lower throughput for these chains.

4.2. TCP on four hop chains

Studying the performance of four hop chains is important for a number of reasons. Firstly, these chains can occur in a MHWN, so knowing how they perform is critical to predicting MHWN performance. Secondly, four hop chains present the opportunity to study more diverse interactions. Studying TCP behavior over such chains is a critical step towards analyzing TCP behavior over general n-hop chains.

Fig. 5a presents the throughput of the chains we study. We observe that chains with the worst performance are ones that have link pairs (a) that are not sender-connected, (b) that have an asymmetric interaction and (c) where the weak link carries forward flow (TCP data packets). For all other cases the throughput achieved is comparable. Based on these factors, the four-hop chains in a MHWN can be classified into three main performance classes. Chains (A) where links carrying data flow are the weak links in an interaction, (B) where links carrying ACK flow are the weak links in an interaction and (C) where none of the links are weak. Table 3 lists which chains fall in each of these classes. In some chains there are interactions that cause both the forward and the reverse flow to be weak. In such cases, performance of the chain suffers more due to the link carrying data flow being weak. Therefore, these chains are considered in the class (A), described above. We now discuss each of these classes in detail.

4.2.1. Chains where the forward going links are weak

These chains can be broadly described as having an interaction that causes collisions at an upstream link carrying data packets. The link causing these collisions is a downstream link carrying data or ACK packets. The performance of a chain in this category depends on (a) how many such weak links a chain has; and (b) the severity of the interaction.

In Section 4.1 we explained in detail how losses occur in the presence of asymmetric interactions. The same explanation also applies to four-hop chains. If the interaction type is an HT, then the data flow has collisions until all the packets at nodes downstream to this link have been transmitted. After that the weak link can successfully transmit; this pattern keeps repeating. An SHAC is another interaction that causes severe overhead. Anytime the two links that have an SHAC interaction transmit concurrently, the weaker link looses a packet. This degrades performance by causing excessive retransmissions at the weak link. However, with an HTC interaction the weak link starts transmissions before the stronger link roughly half the time and is successful. The effect of HTC is, hence, not as severe as SHAC or HT.

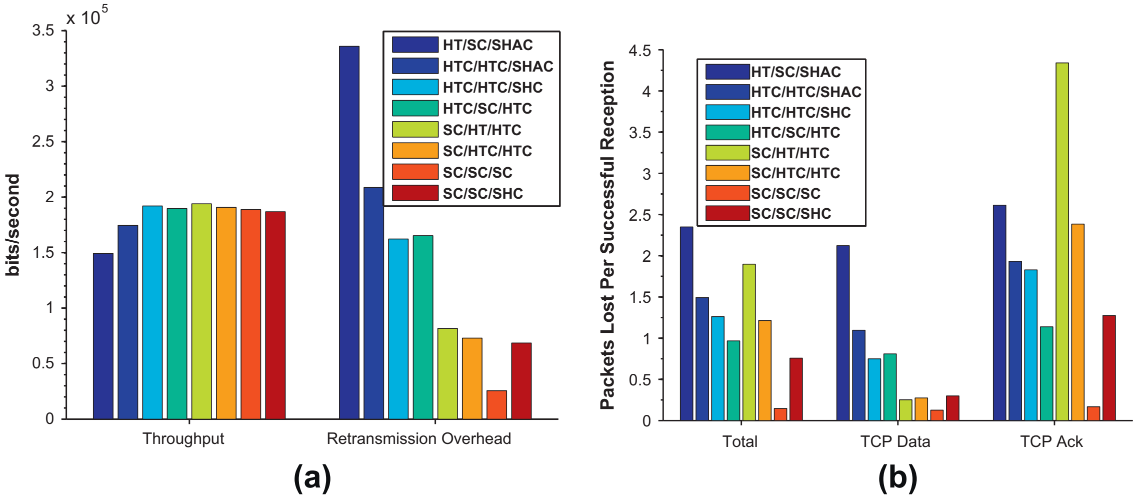


Fig. 5. Performance of four hop chain.

Table 3
The macro-effects shown by each chain type. The third and fourth columns state, for each chain type, whether or not there is a link interaction that causes a forward or backward flow to be weak, respectively. The fifth column qualitatively states the amount of interference generated by a chain of each type. The last column qualitatively states the number of route discovery generated due to consecutive collisions of the same MAC packet. Based on how the interference generated and the route instability caused, the chains are grouped into categories.

Cat.	Chain type	Weak interaction		Interference generated	Route instability
		Forw	Back		
A	HT/SC/SHAC	Yes	Yes	High	Low
	HTC/HTC/SHAC	Yes	Yes	Medium	Low
	HTC/HTC/SHC	Yes	Yes	Medium	Low
	HTC/SC/HTC	Yes	Yes	Medium	Low
B	SC/HT/HTC	No	Yes	Low	High
	SC/HTC/HTC	No	Yes	Low	Medium
	SC/SC/SHC	No	Yes	Low	Low
C	SC/SC/SC	No	No	Negligible	None

Among the chains in this class, HT/SC/SHAC behaves the worst in terms of throughput and overhead. This result is intuitive since this chain has two interactions where a link carrying data flow is weak, one HT and one SHAC. The HTC/HTC/SHAC chain has the HT replaced with an HTC interaction resulting in slightly better performance. However, the performance is still significantly lower than the other chains.

The HTC/HTC/SHC and HTC/SC/HTC chains are similar to each other and better than both HT/SC/SHAC and HTC/HTC/SHAC in terms of achieved throughput. The interactions of the links carrying TCP data packets in these two chains are similar. The only difference is that in HTC/SC/HTC Hop1 carrying forward flow is competing with both Hop3 and Hop4 carrying backward flow. The HTC/HTC/SHC, on the other hand, has Hop1 carrying forward flow competing with Hop3 carrying backward flow and Hop2 carrying forward flow competing with Hop4 carrying backward flow. So node0 in an HTC/SC/HTC chain has both node3 and node4 as hidden terminals causing TCP data packet drops while in the HTC/HTC/SHC chain node0 has only one hidden terminal. This difference causes HTC/SC/HTC to have more TCP packet drops at Hop1 resulting in worse performance, in terms of transmission overhead.

This analysis shows that chains where a link carrying Data traffic is the weak link in an interaction suffer from significant performance loss. These links should be avoided whenever possible.

4.2.2. Chains where the backward going links are weak

These chains have interactions that cause collisions at links carrying ACK traffic. The collisions could be from other links carrying ACK traffic or from links carrying data traffic. The chains that are affected on the most part by this category of interactions are SC/HT/HTC, SC/HTC/HTC and SC/SC/SHC. Qualitatively, the affects caused by HT, HTC and SHC on the ACK traffic is the same as what was described above for the case where links carrying TCP data packets were at a disadvantage. However, there are two main differences. (1) ACK packets are smaller and hence when they collide with a TCP data packet it is possible for multiple retransmissions of the same ACK packet to collide with a single TCP data packet. This is because the smaller size of ACK packets causes their transmissions to be shorter and hence retransmissions can be scheduled faster. This effect causes the number of collision drops per delivered packet in the SC/HT/HTC, SC/HTC/HTC and SC/SC/SHC chains (depicted in Fig. 5b) to be higher than the case where links carrying TCP data packets are at a disadvantage. (2) Due to the much smaller size of the ACK packets the overhead of ACK collisions is much lower. This is the reason the overhead observed in Fig. 5a is much lesser for these chains even though the number of ACK collisions are close.

4.2.3. Chains where none of the links are weak

The only chain that fits in this category is the SC/SC/SC chain. All the link pairs in this type of chain are sender-connected. As described in Section 4.1 the transmitters in an SC/SC/SC chains have a complete view of the medium and can arbitrate it in a way to avoid all the effects that cause performance degradation in the other chains. The only source of retransmission in these chains are synchronized collisions.

4.3. Effect of varying parameters

In this section we show the effect that changing certain parameters has on TCP performance over four-hop chains. The parameters we vary are maximum TCP congestion window size, RTS/CTS setting (enable), data rate, delayed TCP acknowledgment setting (enable) and TCP flavor.

4.3.1. TCP window size

Previous work has suggested putting a cap on the TCP window size to improve performance and reduce chain interference effects [10,30]. Fig. 6 shows the effect of varying maximum TCP window size on the chains we study. Fig. 6b shows that limiting the maximum window size to one almost completely eliminates the retransmission overhead and causes the throughput of all chain types to be almost identical. This shows that the only way to eliminate the performance difference between chain types is by setting the congestion window to one, effectively removing link interactions in a chain. However, for this to work properly for a TCP connection that spans both wired and wireless network, schemes such as split-TCP [31] have to be used. However, such schemes have not been developed for MHWNs.

For all other settings of congestion window size, the performance difference between the different chain types remains.

4.3.2. RTS/CTS

Researchers have suggested disabling RTS/CTS to improve TCP performance over chains [10]. However, other works [30] suggest that enabling RTS/CTS gives better performance in some cases without being specific of when these cases occur or what these cases are. Fig. 7b shows that enabling RTS/CTS significantly reduces overhead in all chains. Enabling RTS/CTS also reduces throughput in all but one chain type. For HT/SC/SHAC chains it is actually better to enable RTS/CTS as it slightly increases throughput, but more importantly, it significantly reduces retransmission overhead. This gives a much better overall TCP performance over this chain type. For all other chain types, it is better to disable RTS/CTS.

4.3.3. Data rate

Increasing link data rates increases the capacity of the chains and therefore improves throughput. However, as Fig. 8 shows, the performance difference between the chain types remains. The only difference in trends is the performance of SC/HT/HTC. The SC/HT/HTC chain type looses TCP ACK packets in the reverse direction due to the HT and HTC interactions. These ACK losses cause the RTT estimate to increase. For lower data rates (2 Mbps) the effect of this increase in RTT is not significant. However, at higher data rates the higher RTT causes the sources to slow down significantly causing an impact on the throughput. Fig. 8 shows that even though the throughput of SC/HT/HTC decreases significantly, the overhead is not very high. This is because ACK packets are small, and even though a large number of ACK packets are lost, the effective bits/second of overhead is small.

4.3.4. Using delayed TCP acknowledgments

Delayed TCP acknowledgments improve TCP performance on chains [8]. Fig. 9a shows that using delayed acks increases throughput of all chain types. However, the difference in throughput between chain types and the difference in retransmission overhead remains (Fig. 9b). This shows that using delayed acknowledgments improves performance but the overall difference, in performance, between chain types remains the same.

4.3.5. TCP flavor

We study the effect of varying TCP flavor on the performance of the chains studied. This enables us to ensure that the effects observed are an aspect of the MAC interactions and not a particular TCP flavor. We ran TCP Tahoe, TCP Reno and TCP Vegas over the chains studied above. The performance of all three flavors matched closely with the trends discussed above. Fig. 10 shows the results. For clarity we present the performance of only three of the chains studied. The figure shows that the throughput varies only slightly when the TCP flavor is changed. Also, the retransmission overhead is also very close for all three TCP flavors. This confirms that the performance observed is an effect of the MAC interactions and its effect on the TCP layer; changing the TCP flavor has minimal, if any, effect on the chain performance.

4.3.6. Summary

Changing the parameters discussed above improve performance of TCP over wireless chains, however, the performance difference between the different chain types remains. Therefore, choosing chains with better interference interactions will improve network performance irrespective of parameter settings.

5. Discussion

This section discusses the impact of chain behavior on a MHWN. We first discuss the interference effect of different

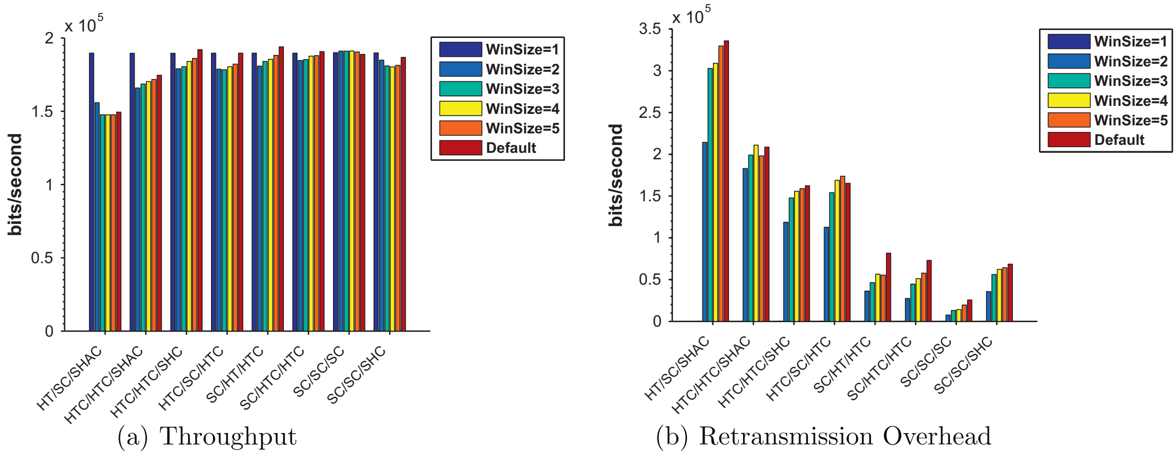


Fig. 6. Effect of changing maximum TCP window size on (a) throughput and (b) retransmission overhead.

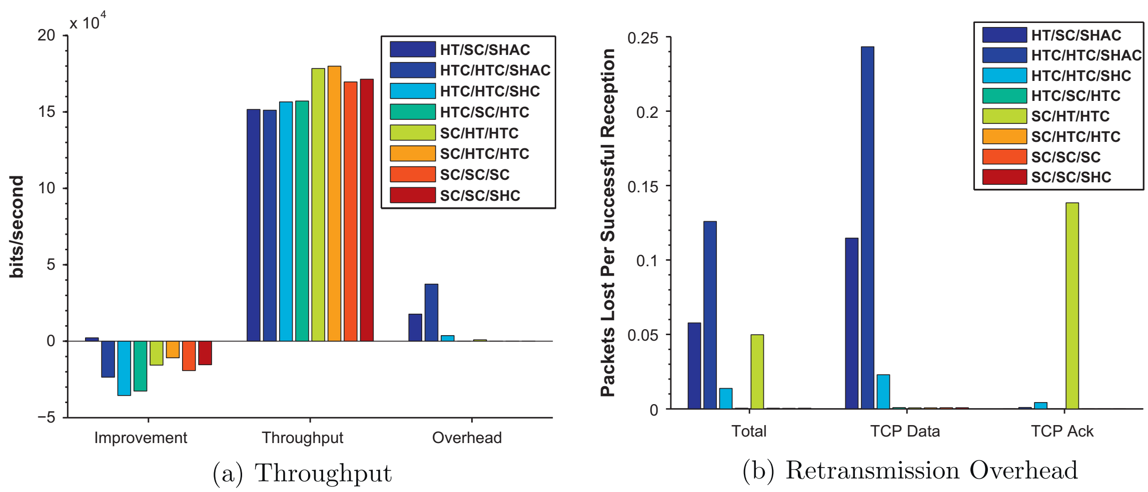


Fig. 7. Effect of enabling RTS/CTS.

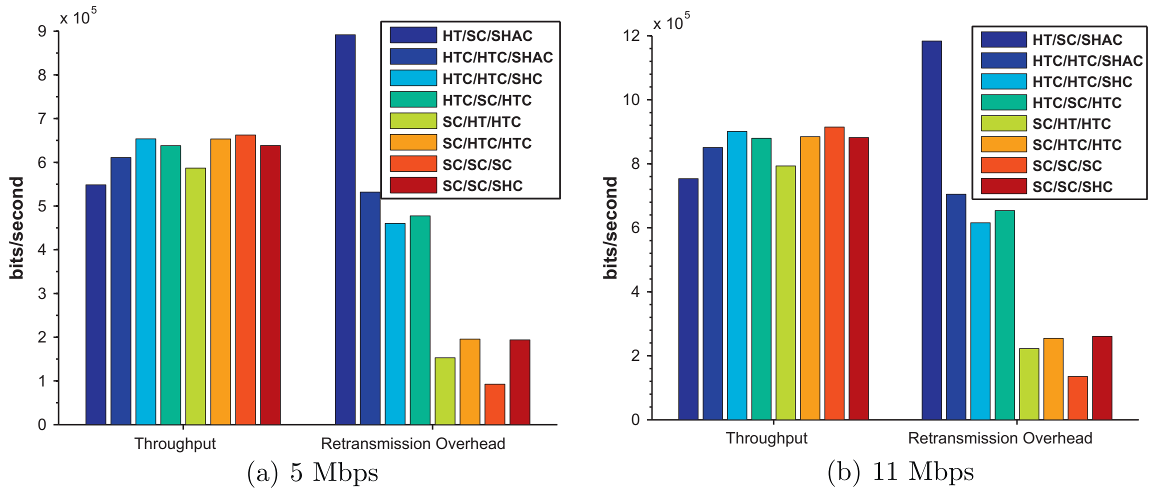
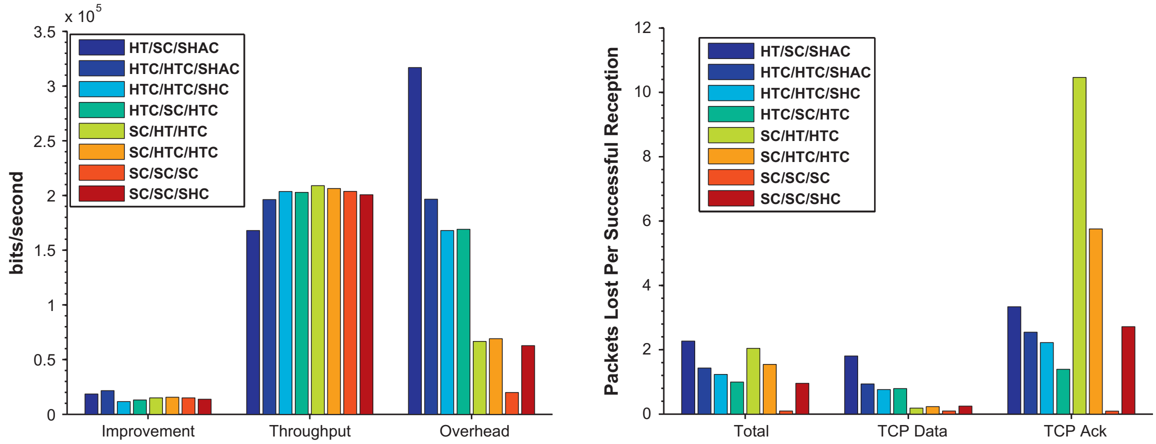
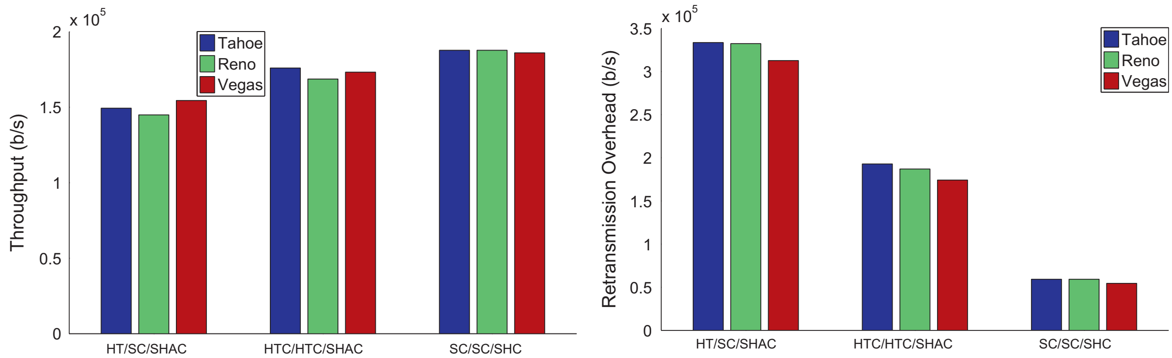


Fig. 8. Effect of varying data rate.



(a) Throughput (b) Retransmission Overhead

Fig. 9. Effect of using delayed acknowledgments.



(a) Throughput (b) Retransmission Overhead

Fig. 10. The performance of different flavors of TCP over three of the chains studied: (a) shows the throughput achieved by each of the TCP flavors and (b) shows the retransmission overhead for each of the TCP flavors.

chains due to retransmission traffic. We then discuss route instability caused by the behavior of different chains. Finally, we categorize chains according to their performance and describe the behavior of each category. Table 3 summarizes the discussion presented in this section.

5.1. Interference effects

Knowledge of how different types of chains perform can significantly improve MHWN performance. Figs. 4a and 5a illustrate the retransmission overhead of the chains we analyzed. In the three-hop case, SC/SC/SC gives 25% better throughput, compared to SC/SC/SHAC, using only 5% of the retransmission overhead. Since the overhead is measured in terms of unsuccessful transmissions, this means that by avoiding all of those additional transmissions the SC/SC/SC chain kept the medium busy for a smaller duration – still achieving better performance. This difference is also observed in the four-hop case when comparing SC/SC/SC with HT/SC/SHAC. In this case the HT/SC/SHAC chain has a 25% lower throughput while its retransmission overhead is over 90% higher compared to the SC/SC/SC chain. It is obvious that limiting connections to use chains that have low retransmission overhead would significantly increase MHWN throughput by facilitating medium reuse. On the other hand, using chains with undesirable interactions greatly increases interference in the network while providing similar or lower throughput to the sender-connected chains. Therefore, knowledge of chain performance can be used to significantly improve MHWN performance.

5.2. Route instability

Knowledge of relative performance of chain types can help improve MHWN routing. SHAC and HT interactions affect TCP performance due to numerous collisions (Sections 4.1 and 4.2). Some routing protocols decide that a route is lost if a certain number of consecutive MAC transmissions are lost [32,33]; the routing protocol then initiates route discovery. In the absence of RTS/CTS this number is set to seven. Using these routing protocols would initiate route discovery numerous times if the chains selected suffer from numerous collisions. Since the route is not lost, the same route will be found every time. Fig. 11 shows the number of such route drops in each chain and then breaks up the route drops depending on whether they were caused by TCP data packet losses or TCP ACK packet losses. Recall from Section 4.2 that due to smaller size collisions of TCP ACK packets are significantly higher in number. So even though ACK collisions do not cause significant overhead in terms of bits/s retransmitted, their effect on initiating route discovery is significant. Therefore, routing protocols should select chains that have fewer collisions and hence provide stable routes.

5.3. Ranking interactions

Table 3 separates the four-hop chains in categories based on how each chain affects the network. From the table we can see that chains that have interactions where the link carrying data traffic (forward flow) is at a disadvantage generate more retransmission traffic; keeping the medium busy longer. This is due to the high retransmission cost of TCP data packet collisions. The chains in category A are of this type. These chains can also have interactions where the link carrying ACK traffic (backward flow) is at a disadvantage. However, the forward flow being at a disadvantage causes a larger affect and so these chains show relatively lower route instability compared to category B. Category B is the set of chains that have interactions where the links carrying backward flow are at a disadvantage. For all these chains, except SC/SC/SHC, the large number of ACK collisions cause a large number of route drops and hence higher route instability.

The chains that have good performance are the SC/SC/SC and SC/SC/SHC chains. From Table 1 we see that over 40% of the chains in a random network are of these types. We observe that when these chains carry TCP traffic, the link interactions in the forward and backward path both enable efficient communication. Therefore, these chains have the least amount of collisions and retransmission traffic. This keeps their route instability and interference generated low.

The performance of the studied chains give guidelines for designing routing protocols. Routing protocols should pick routes that have as many sender connected interactions as possible; both in the forward and the backward direction. Furthermore, asymmetric interactions significantly affect throughput and the routing protocol should, as much as possible, avoid chains with such interactions. Previous work done on identifying interactions [34] in a chain can be used to accomplish this goal.

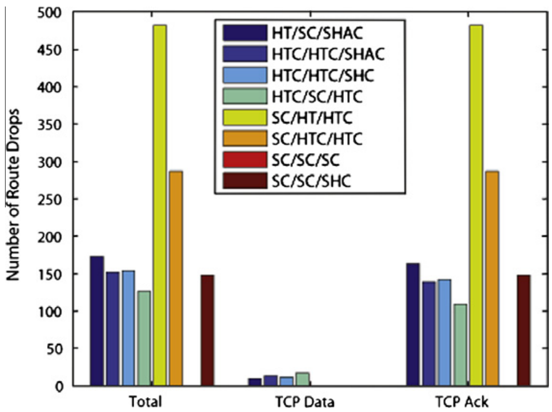


Fig. 11. Route instability due to TCP data and TCP Ack packet losses.

6. Towards cross-chain TCP analysis: two single hop TCP flows

Thusfar, we have been looking at the performance of TCP on an isolated multi-hop chain, examining the influence of self-interference within the chain on how it behaves. In more general settings, it is often the case that interference is present from other traffic within the network (as well as, possibly, from co-located technologies in the same frequency band). Given two TCP chains, they can interfere with each other in a large number of ways: there are two flows for each chain (data and ack) and there are no geometric constraints on how two chains are placed relative to each other. For example, consider the case of two interacting two-hop TCP chains. There are two flows for each chain (data and ack), each consisting of two links (four links per chain). Each of the links can have a different interaction with each of the three other links. If each interaction can be any one of the six interactions described in Section 3 then we have a total of 6^{16} individual interaction patterns.

While it is likely that these interactions can be grouped into fewer classes of interactions, their behavior is quite rich and complex. Thus, as a first step to understanding cross interference between two TCP chains, in this section we analyze the performance of two interacting single-hop TCP flows. We call these scenarios *two flow* scenarios to differentiate them from the single chain (*single flow*) scenarios that have been considered thusfar. We start with an explanation of our experimental setup followed by a nomenclature for different settings of two single-hop TCP flows. We then present our analysis of two single-hop TCP flows based on the interactions between data packets, ACK packets and data and ACK packets. Our future work will extend this analysis into interfering multi-hop wireless chains.

The presence of two sources in the two flow case significantly differentiates the two flow performance from the single flow case. In the single flow case the duration of an interference interaction adversely affecting TCP performance was limited. Either the strong link would be an upstream link causing collisions at a downstream link in which case the stronger link would timeout because it prevents data packets from reaching the destination or the ACK packets from getting back to the source. Alternatively, the strong link would be a downstream link causing collisions at an upstream link in which case eventually the queue of the strong links source would become empty since it does not allow packets from the upstream nodes to reach its queue. In either case, eventually the strong link would stop transmitting and give the weak link an opportunity to transmit without collisions. In the case of more than one TCP sources this observation does not hold. This would effectively cause the effect of interference interactions to be more severe in the case of more than one flow. This would at times allow one TCP flow to take over the medium and starve the other flow [14].

6.1. Experimental setup

We use a simple scenario with four nodes: two sources and two destinations. There is a TCP flow between each source/destination pair. The source sends data packets to the destination which in turn responds with an ACK. There are three main types of MAC interactions in this case: (i) between the TCP data transmissions of both flows (data interaction), (ii) between the TCP ACK transmissions of both flows (ACK interaction), and (iii) between the TCP data transmission of one flow and the TCP ACK transmission of the other flow (cross interaction). As described later, we use these interactions to name different instances of two single-hop TCP connections interacting with each other.

6.2. Nomenclature

Here we present a nomenclature for the different interaction groups present in the scenario. Our interaction nomenclature is similar to the one used for multi-hop chains above. The interactions between two single-hop TCP flows are represented as D/A/C. The 'D' describes the interaction between the data flow of both the TCP connections. The 'A' describes the interaction between the ACK flow of both the connections and the 'C' describes the interaction between the data flow of one connection and the ACK flow of the other connection. In all three cases (D, A and C) we study the interactions described in Section 3.2. The following example clarifies the naming convention. If we have two single-hop TCP flows classified as SC/SC/SHAC then this means that the data transmissions of these flows are sender-connected (SC) and the ACK transmissions also have an SC interaction. The cross interaction (data transmission of one flow and the ACK transmission of the other flow) have an SHAC interaction. In the cross interaction, as we will see later, it makes a significant difference if the weak link in the interaction is an ACK transmission or a data transmission. We will consider both cases separately.

6.3. Performance analysis

Two single-hop TCP flows have four link interactions; data and ack of each flow interacting with data and ack of the other flow. If each link interaction can be any one of the six interactions presented in Section 3.2 then there are 6^4 cases to study. Although some of these cases cannot occur due to geometric limitations, the number of cases that do occur is still quite large. We limit this number by studying the impact of varying one type of interaction at a time and, when possible, keeping the other two constant. For example, we start off by varying the interactions between the two data flows while keeping the ACK interactions and cross interactions sender-connected. However, when we make the data flow interactions HT or HTC, it becomes geometrically impossible to make the cross interactions SC and therefore the cross interaction becomes HTC. We repeat the same interaction variation for ACK flows and cross flows. Using this mechanism reveals how varying interactions over data, ack and cross flows effects the performance of two single-hop TCP flows.

We analyze the performance of fifteen scenarios. We run five instances of each scenario and present the mean of these instances along with the 95% confidence interval. The scenarios are divided into three groups based on which interactions are not sender connected. Each group is studied separately. The groups are (i) scenarios where the data transmissions are not sender connected, (ii) scenarios where the ACK transmissions are not sender connected, and (iii) scenarios where the cross transmissions are not sender connected. The one scenario which does not fit into any group is the SC/SC/SC case. In this case all transmissions

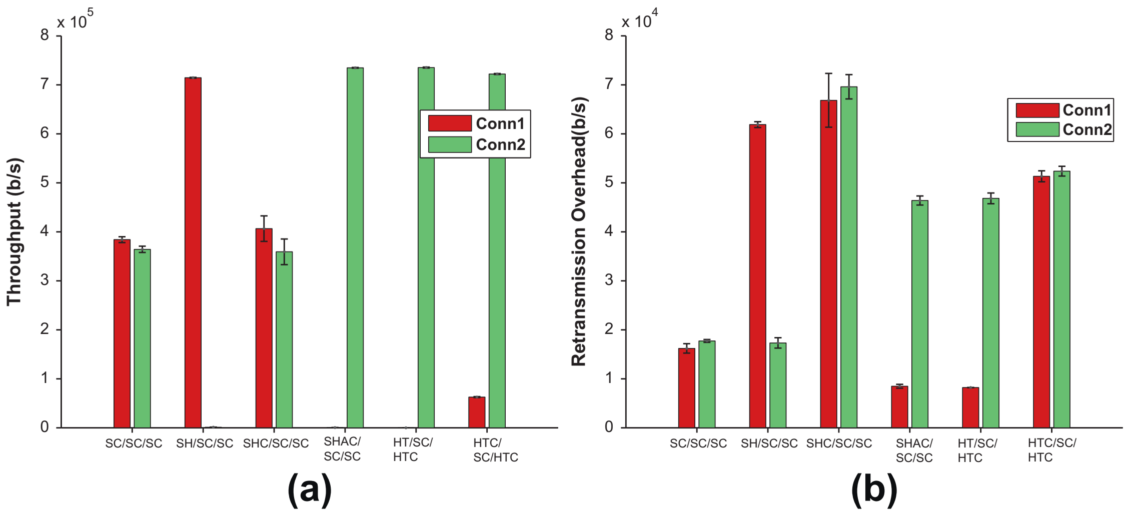


Fig. 12. Varying transmission interactions for two single-hop wireless TCP flows: (a) throughput and (b) retransmission overhead.

are sender connected. This serves as a control case to gauge the performance of the other cases. Fig. 12b shows that this scenario has the lowest overhead. The only collisions in this case are synchronous collisions.

Fig. 12a shows that the combined throughput of the two flows is approximately the same for all cases. The only differences between the scenarios are (i) fairness and (ii) retransmission overhead. Therefore, we will analyze the scenarios using these as metrics. We now study each of these three groups separately.

6.3.1. Cases where data transmissions are not sender connected

Here we analyze the performance of the cases where the links carrying data packets are not sender connected. We start with the SHC/SC/SC case. The SHC/SC/SC case has a symmetric interaction between the two flows. The SHC interaction causes one source to have a successful transmission at a time and as expected both the flows share the medium almost evenly. In this case, when both sources transmit together, the flow that starts transmission later looses the packet. The packet losses cause this case to have a higher overhead compared to SC/SC/SC.

The next case, SH/SC/SC, has non-intuitive performance. This case also has a symmetric interaction, however, one flow completely takes over the medium. This is in sharp contrast to previous works [7] where flows that have symmetric interactions suffer from short term unfairness but in the long run share the medium equally. This effect is due to transmission timing and queue sizes as described below.

Fig. 13a shows the transmissions, collisions, packet receptions and packet drops for a 0.9 s duration in the experiment. At time 10 s both sources transmit. At this time the MAC queues at both sources have only one packet, the one they are currently trying to transmit. Since the sources have a symmetric hidden terminal, both transmissions end in collisions. The sources repeatedly transmit and cause collisions with each others packets till 7 retries after which time the MAC layer drops the packet. This pattern is seen between time 10 and 10.1 s in Fig. 13a. Both sources then wait for a TCP timeout. Following the timeout the transport layers of both sources resends the dropped packet. The resent packet causes a similar transmission pattern between 10.2 and 10.3 s. The transport layer retransmits the packet a second time and another pattern of transmissions and collisions is observed between 10.6 and 10.7 s. In this case, however, one source gets 7 collisions sooner and the other source successfully transmits the packet. Connection 1 (Conn1), between nodes 3 and 4 is the successful flow and Connection 2 (Conn2), between nodes 1 and 2, is the unsuccessful flow. The source of Conn1 now gets an opportunity to continue transmitting unhindered whereas the source of Conn2 has to wait for a TCP timeout.

Conn1's successful transmissions enable it to increase its window size creating unfairness between the two flows. From Fig. 13b we see that Conn2 waits for a TCP timeout from 10.7 s to 11.4 s. During this time, the source of Conn1 is able to transmit 66 TCP data packets. Furthermore, because of these successful transmissions it's window size increases and it has packets in its MAC queue waiting for transmission. At 11.4 s (from Fig. 13b) Conn2 has a packet to transmit. We observe that the collision pattern described above is repeated: notice the empty region between 11.4 and 11.5 s when no new TCP packets are generated at either sources. These collisions end in a packet drop for both connections. However, since Conn1 has packets waiting to be transmitted, it does not wait for a TCP timeout and immediately restarts successfully transmitting data packets and ACKs. Hence, even though both the flows have a symmetric interaction, the asymmetry of queue sizes causes Conn1 to take over the medium. Therefore, the source that gets the first successful transmission always takes over the medium. Due to the inability of the source of Conn2 to send a significant number of

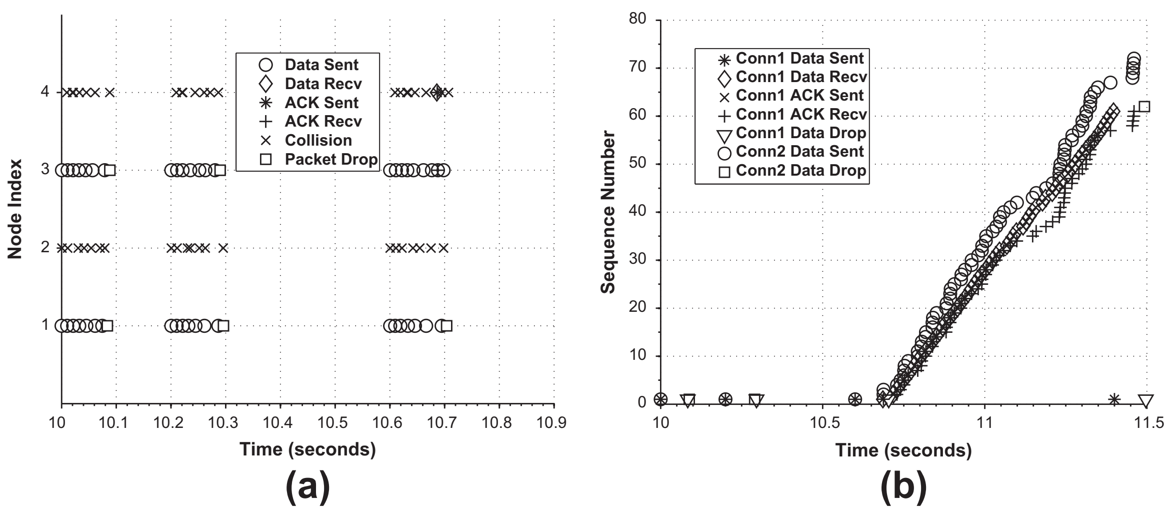


Fig. 13. A detailed look at the performance of the SH/SC/SC case. (a) Mac trace showing how the two flows repeatedly collide until one flow wins. (b) Once a flow has won, it takes over the channel and continues mostly unchallenged. This is due to asymmetry in the queue buildup of the two flows.

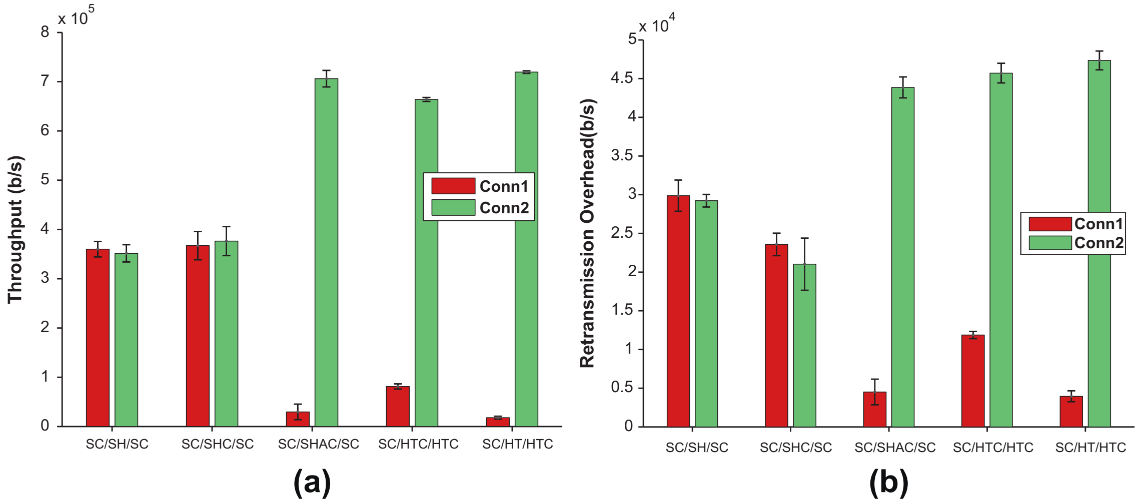


Fig. 14. Throughput and retransmission overhead for cases where the ACK interactions are not sender connected.

packets successfully it assumes the network is congested and keeps increasing its estimate of the RTT. The source of Conn2 therefore keeps increasing the time between successive retransmission attempts.

The unfairness caused by the asymmetry in the SH/SC/SC case is not short term. We ran simulations lasting over 1000 s, with different seeds and consistently got the same pattern. Different works suggest limiting the window size [9,35,36] to 3 or 4 or some fraction of path length to improve performance. However, we observed that the only way to ensure fairness between these flows is to limit the window size of both flows to 1. This ensures that there is no build up in the MAC queues of either flow and there is complete symmetry between them. However, for a window size of 1 to work properly for a TCP connection that spans both wired and wireless network, schemes such as split-TCP [31] have to be used. Such schemes have not been developed for MHWNs.

The rest of the interactions are asymmetric interactions. In these cases, as expected, the flow that is at an advantage gets a major portion of the throughput. In case of the HTC/SC/HTC interaction, the capture effect enables the weaker flow to transmit a few packets successfully. However, in the SHAC/SC/SC and HT/SC/HTC interactions the weaker flow does not get any successful transmissions.

Note that for asymmetric interactions, the flow at a disadvantage hardly gets any successful transmissions. This is in sharp contrast to the single flow case on three and four hop chains where the weak link in an asymmetric interaction was able to get considerable throughput. In the two flow case since the sources are different, the source of the flow loosing packets transmits at a lesser frequency,

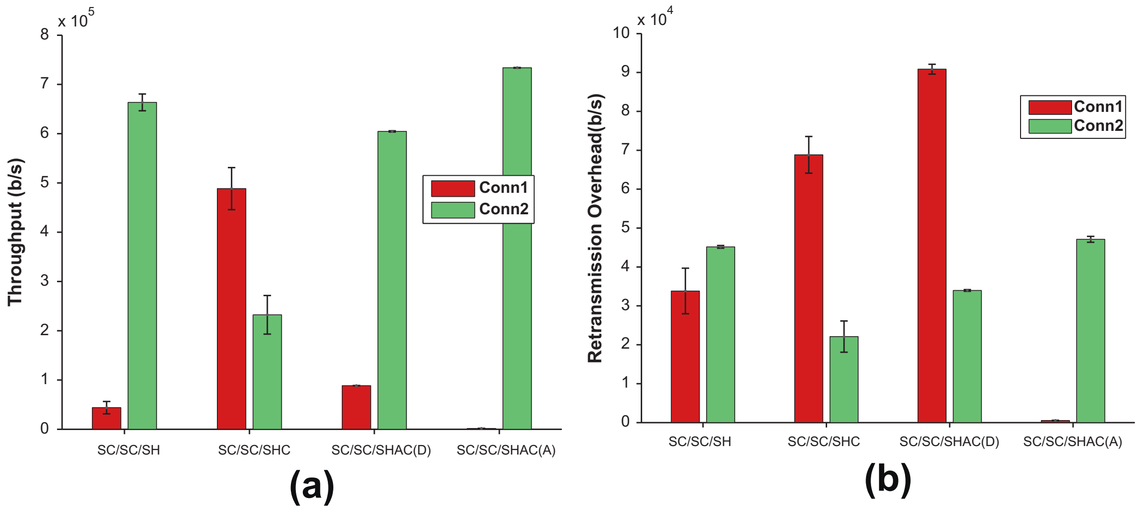


Fig. 15. Throughput and retransmission overhead for cases where the cross interactions are not sender connected.

allowing the stronger flow to get an even higher portion of the medium.

6.3.2. Cases where ACK transmissions are not sender connected

Fig. 14 shows the performance of the TCP flows when the ACK interactions are varied. In this case all the cases perform as expected. ACK packets are smaller than data packets so the retransmission overhead is significantly lower. The symmetric cases (SC/SH/SC and SC/SHC/SC) give almost fair share of the medium to both flows with SC/SHC/SC having comparatively lower retransmission overhead due to the capture effect. The asymmetric cases starve the flow with the weak link and the flow with the stronger link takes a major portion of the medium. The asymmetric cases show sever unfairness in terms of medium access.

6.3.3. Cases where cross transmissions are not sender connected

Fig. 15 shows the performance of the TCP flows as the cross interactions are varied. In this case there is a permanent asymmetry between the flows: size asymmetry. In the presence of capture (SC/SC/SHC) the connection that contributes data packets to the interaction fares better. This is because a successful ACK transmission will cause 1 data packet collision. However, a successful data packet transmission will cause multiple ACK packet collisions. This significantly affects RTT estimates and timeout estimates, causing the flow loosing ACK packets to slow down. For the case of SC/SC/SH, the asymmetry goes in favor of the flow contributing ACK packets to the interaction. In this case, the probability of a data packet transmission starting in the middle of an ACK packet transmission is much lower than the opposite case. Therefore, most of the data packets will be lost and hence the flow contributing the ACK packets to the SH interaction will take over a major portion of the medium.

For the SHAC case, when the weaker flow contributes a data packet to the interaction, SC/SC/SHAC (D), very few packets get through. When the weaker flow contributes ACK packets to the interaction, SC/SC/SHAC (A), due to the size asymmetry and the interaction asymmetry, no ACK packets get through at all.

6.3.4. Discussion

The above section shows that inter-flow interactions significantly affect TCP performance. Comparing Figs. 12a and 14a we see that the same interaction has a much worse effect on performance if the weak link is carrying data packets, rather than ACK packets. Furthermore, asymmetry of any form in interactions causes unfairness. This unfairness can be size asymmetry, capture asymmetry, queue size asymmetry or asymmetry caused by a hidden terminal. Therefore, it is important when routing TCP flows to ensure that asymmetric interactions are avoided as much as possible and when unavoidable, data transmissions should not be on the weak link of an interactions. These principles match closely with the findings from Section 4.

7. Conclusions and future work

In this paper, we have demonstrated that wireless chains, that otherwise appear identical to higher layers, can have significantly different performance. The differences occur due to the types of MAC layer interactions between the hops forming the forward (data) chain, the backward (ACK) chain as well as interactions between the two. We presented a classification of the types of three- and four-hop chains that groups them according to the most important interactions. We analyzed the frequency of occurrence of each chain type to identify the most frequently occurring ones. We then analyzed the performance of the most commonly occurring chains under a variety of conditions. We discovered substantial differences in the performance of the different chains. These differences cause up to 25% difference in throughput. Furthermore, some chains require a significantly higher number of MAC transmissions due to collisions. These retransmissions keep the medium busy and have substantial impact on the overall network performance.

Based on the observed performance, we have proposed a TCP performance based ranking of the chains in a MHWN. Sender-connected chains provide high throughput with low retransmission overhead and ought to be favored by higher layers. We extend our analysis further to study multiple TCP flows. We observe that interactions that cause performance degradation in a single TCP flow follow similar trends with multiple flows. However, these interactions have a much worse effect when multiple flows are involved; at times causing one flow to completely take over the medium and not allowing the other flow any successful transmission.

The analysis and results we have presented in this paper enable a better understanding of how TCP behaves over chains that occur in a general network setting and how performance is affected by cross chain effects in two single-hop TCP flows. For future work, we plan to extend our analysis to larger n-hop chains. Furthermore, we would like to evaluate and develop routing protocols that exploit this knowledge to improve network performance.

References

[1] W. Allen, A. Martin, A. Rangarajan, Designing and deploying a rural ad-hoc community mesh network testbed, in: LCN '05: Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary.

[2] V. Navda, A. Kashyap, S. Das, Design and evaluation of imesh: An infrastructure-mode wireless mesh network, in: WOWMOM '05: Proceedings of the Sixth IEEE International Symposium on World of Wireless Mobile and Multimedia Networks.

[3] J. Bicket, D. Aguayo, S. Biswas, R. Morris, Architecture and evaluation of an unplanned 802.11b mesh network, in: MobiCom '05: Proceedings of the 11th Annual International Conference on Mobile Computing and Networking.

[4] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, IEEECommunications Magazine 40 (8) (2002) 102–114.

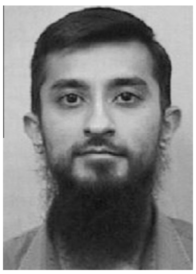
[5] I. Chlamtac, M. Conti, J.J. Liu, Mobile ad hoc networking: imperatives and challenges, Ad Hoc Networks 1 (1) (2003) 13–64. <http://dx.doi.org/10.1016/S1570-8705(03)00013-1>.

[6] A. Balasubramanian, R. Mahajan, A. Venkataramani, B.N. Levine, J. Zahorjan, Interactive wifi connectivity for moving vehicles, in: SIGCOMM '08: Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication, ACM, New York, NY, USA, 2008, pp. 427–438.

[7] S. Xu, T. Saadawi, Revealing the problems with 802.11 medium access control protocol in multi-hop wireless ad hoc networks, Computer Networks 38 (4) (2002) 531–548.
[8] S. Xu, T. Saadawi, Evaluation for tcp with delayed ack option in wireless multi-hop networks, in: Vehicular Technology Conference, 2001. VTC 2001 Fall. IEEE VTS 54th, vol. 1, 2001, pp. 267–271.
[9] Z. Fu, H. Luo, P. Zerfos, S. Lu, L. Zhang, M. Gerla, The impact of multihop wireless channel on tcp performance, IEEE Transactions on Mobile Computing 4 (2) (2005).
[10] V. Kawadia, P. Kumar, Experimental investigations into tcp performance over wireless multihop networks, in: E-WIND '05: Proceedings of the 2005 ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis.
[11] S. Xu, T. Saadawi, M. Lee, On tcp over wireless multi-hop networks, in: Military Communications Conference MILCOM, IEEE Communications for Network-Centric Operations: Creating the Information Force, vol. 1, 2001, pp. 282–288.
[12] H. Khalife, N. Malouch, Interaction between hidden node collisions and congestions in multihop wireless ad-hoc networks, in: ICC '06: IEEE International Conference on Communications, vol. 9, 2006, pp. 3947–3952.
[13] L.-P. Tung, W.-K. Shih, T.-C. Cho, Y. Sun, M.C. Chen, Tcp throughput enhancement over wireless mesh networks, IEEE Communications Magazine 45 (11) (2007) 64–70.
[14] M. Garetto, J. Shi, E.W. Knightly, Modeling media access in embedded two-flow topologies of multi-hop wireless networks, in: MobiCom '05, 2005.
[15] S. Razak, N.B. Abu-Ghazaleh, V. Kolar, Modeling of two-flow interactions under sinr model in multi-hop wireless networks, in: Proc. LCN, 2008, pp. 297–304.
[16] A. Kochut, A. Vasan, A.U. Shankar, A. Agrawala, Sniffing out the correct physical layer capture model in 802.11b, in: ICNP '04, 2004, pp. 252–261.
[17] B. Chen, I. Marsic, R. Miller, Improving tcp performance over multi-hop wireless networks, in: IEEE 68th Vehicular Technology Conference (VTC 2008-Fall), 2008, pp. 1–5.
[18] J. Li, C. Blake, D. DeCouto, H. Lee, R. Morris, Capacity of ad hoc wireless networks, in: MobiCom, 2001, pp. 61–69.
[19] P. Ng, S. Liew, Throughput analysis of 802.11 multi-hop ad hoc networks, IEEE/ACM Transactions on Networking 15 (2) (2007) 309–322.
[20] S. Razak, N. Abu-Ghazaleh, Self-interference in multi-hop wireless chains, in: Proc. ADHOC-NOW, 2008.
[21] S. Razak, U. Baroudi, Simulation analysis of bi-directional traffic characterization over wireless ad hoc networks, in: IWCMC '09: Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing, ACM, New York, NY, USA, 2009, pp. 1274–1278.
[22] L. Kleinrock, F. Tobagi, Packet switching in radio channels, in: IEEE Transactions on Communications, 1975.
[23] K. Chen, Y. Xue, K. Nahrstedt, On setting tcp's congestion window limit in mobile ad hoc networks, in: ICC '03. IEEE International Conference on Communications, vol. 2, 2003, pp. 1080–1084. doi:10.1109/ICC.2003.1204432 2.
[24] S. Gopal, S. Paul, D. Raychaudhuri, Investigation of the tcp simultaneous-send problem in 802.11 wireless local area networks, in: ICC 2005, IEEE International Conference on Communications, vol. 5, 2005, pp. 3594–3598.
[25] C.-W. Hsu, T.-C. Hou, Achieving fair throughput for tcp flows in multi-hop wireless backhaul networks, in: IEEE International Conference on Communications (ICC), 2010, pp. 1–5. doi:10.1109/ICC.2010.5501909.
[26] K. Nahm, A. Helmy, C.-C. Kuo, Tcp over multihop 802.11 networks: issues and performance enhancement, in: MobiHoc '05: Proceedings of the 6th ACM international Symposium on Mobile Ad Hoc Networking and Computing.
[27] A. Jindal, K. Psounis, The achievable rate region of 802.11-scheduled multihop networks, IEEE/ACM Transactions on Networking 17 (4) (2009) 1118–1131.
[28] S. Lee, B. Bhattacharjee, S. Banerjee, Efficient geographic routing in multihop wireless networks, in: MobiHoc, 2005, pp. 230–241.
[29] G. Bianchi, Performance analysis of the ieee 802. 11 distributed coordination function, IEEE Journal on Selected Areas in Communication 18 (3) (2000).
[30] D. Koutsonikolas, J. Dyaberi, P. Garimella, S. Fahmy, Y. Hu, On tcp throughput and window size in a multihop wireless network testbed, WinTECH '07: Proceedings of the Second ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization.
[31] S. Kopparty, S. Krishnamurthy, M. Faloutsos, S. Tripathi, Split tcp for mobile ad hoc networks, in: GLOBECOM '02: IEEE Global Telecommunications Conference, vol. 1, 2002, pp. 138–142.
[32] D. Johnson, D. Maltz, Y.-C. Hu, The dynamic source routing protocol for mobile ad hoc networks (DSR), in: IETF Draft, 2003.
[33] C.E. Perkins, E.M. Royer, Ad-hoc on-demand distance vector routing, in: WMCSA '99: Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications, IEEE Computer Society, 1999, p. 90.
[34] A. Greco, T. Milcher, V. Kolar, P. Mähönen, M. Petrova, Csma interaction detection and capacity estimation in cognitive radio networks, in: MobiHoc '09: Proceedings of the Tenth ACM International Symposium on Mobile Ad Hoc Networking and Computing.
[35] E. Ancillotti, M. Conti, A. Passarella, Experimental analysis of tcp performance in static multi-hop ad hoc networks, in: Chapter 6 in Mobile Ad TPA Protocol 23 The Computer Journal, Publisher, 2005.
[36] R.D. Oliveira, R.D. Oliveira, T. Braun, T. Braun, A dynamic adaptive acknowledgment strategy for tcp over multihop wireless networks, in: Proceedings of IEEE INFOCOM, 2004, pp. 1863–1874.

Adnan Majeed is a Ph.D. student in the Department of Computer Science at the State University of New York (SUNY) at Binghamton. He received his BS degree from GIK Institute in Pakistan and his MS degree from SUNY Binghamton. His research interests include performance analysis of wireless networks.

Nael Abu-Ghazaleh is an Associate Professor in the Department of Computer Science at the State University of New York at Binghamton. He received his Ph.D. and MS degrees in Computer Engineering from the University of Cincinnati. His research interests lie in Ad Hoc Networks, Computer System Design and Optimization.



Saquib Razak is currently working as a postdoctoral research associate at Carnegie Mellon University in Qatar. He has a Ph.D. student in Computer Science from State University of New York, Binghamton. He has an MS in Software Engineering and a BS in Electrical Engineering from University of Texas, Austin. His research areas include analysis and modeling of wireless network protocols.

Dr. Khaled Harras is an Assistant Professor of Computer Science at Carnegie Mellon University. He is the founder and director of the Networking Systems Lab (NSL) at the CMU-Qatar campus. His research interests spans different thrusts within computer networks including mobile ad hoc, wireless, sensor, and vehicular networks. He is particularly interested in the design and analysis of architectures and protocols for different forms of challenged, delay and disruption tolerant networks (DTNs). Dr. Harras holds a bachelors degree in computer science with a double minor in electronics and business administration from the American University in Cairo. He was awarded the president's cup and Ahmed. H. Zewail prize from AUC. He then received his M.S. and Ph.D. in Computer Science from University of California in Santa Barbara (UCSB). He is a professional member of the ACM and IEEE.