

Performance of TCP in Multi-Hop Access Networks

Yang Su
ETH Zurich, Switzerland

Peter Steenkiste
Carnegie Mellon University

Thomas Gross
ETH Zurich, Switzerland

Abstract—Wireless multi-hop access networks are an increasingly popular option to provide cost-efficient last-mile Internet access. However, despite extensive research, performance of even basic communication services, such as TCP, is still problematic. Measurements collected on a wireless testbed indicate that the poor performance of multi-hop access networks is caused by poor interactions between TCP congestion control and link-layer bit-rate adaptation resulting in severely reduced network efficiency even over short wireless paths (< 6 hops). However, bit-rate adaptation improves fairness across TCP flows. The same principal observations hold for hybrid wireless/wireline paths. To investigate approaches to improve TCP performance, we present a simple model that captures the cause for the inefficiency of TCP over autorate links. We then examine several techniques at both the TCP level and the link layer (TCP Vegas, clamping, limiting the buffer size at the wireless routers) to alleviate contention. None of these techniques works for all scenarios, but the simple approach to limit the buffer size is attractive in many settings that include four or more wireless hops.

I. INTRODUCTION

Wireless multi-hop access networks based on IEEE 802.11 technology are an increasingly popular option to provide cost-efficient last-mile Internet access. In multi-hop wireless networks, where wireless stations are densely deployed and packets are relayed over multiple hops towards the destination, wireless links often experience large variations in link qualities caused by the interference from transmissions in the same network. Consequently, local error control techniques as defined in the 802.11 specification are used aggressively to reduce or recover from packet losses. The basic technique is MAC retransmission of lost packets. In addition, as channel quality changes over time, link-layer bit-rate adaptation dynamically selects the best transmission rate. The use of these local error control techniques clearly impacts both the behavior and performance of an upper layer protocol like Transport Control Protocol (TCP). However, whereas previous research established that link-level local error control can effectively hide link losses in a single-hop wireless network (WLAN) [1], [2], the interactions between local error control, especially link-layer bit-rate adaptation, and TCP in wireless multi-hop networks remain largely unexplored in real systems. The question of TCP performance over hybrid wireless-wireline paths, which we expect to exist frequently in the presence of multi-hop wireless access networks, has received even less attention.

The work presented in this paper was partially supported by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322

In this paper we report on an experimental investigation of the interaction between TCP flow control and IEEE 802.11 link-layer bit-rate adaptation in a wireless multi-hop network testbed. Based on the observations for the all-wireless paths, we further extend our experiments to paths that consist of both multiple hops in the wireless networks and the wireline Internet. Our major findings can be summarized as follows:

- 1) Although performing well in single-hop paths, link-layer bit-rate adaptation becomes very inefficient for TCP flows as the length of the path increases.
- 2) Poor interaction between TCP and bit-rate adaptation contributes to the inefficiency. When TCP grows its congestion window excessively and overloads the network, aggressive link-layer local error recovery may overreact to reduce the transmission bit-rate, hence hiding the congestion from TCP. Eventually these steps lead to reduced efficiency and throughput in the wireless network.
- 3) The principle observation about the interaction between TCP and link-layer bitrate adaptation and the impact of this interaction on the network performance also holds for the hybrid paths that consist of both multiple hops in the wireless network and the wireline Internet.
- 4) We found that a number of TCP-level and link-layer techniques to reduce contention in the wireless network are moderately effective in alleviating congestion and can improve the TCP performance over multi-hop paths. Our measurements show however that none of the existing approaches are entirely sufficient for efficient TCP over multi-hop wireless access networks, and further research is needed to devise a practical and effective solution for this problem.

The rest of this paper is organized as follows. Section 2 summarizes the related works. Section 3 briefly introduces the experimental environment. Section 4 presents the TCP performance over (all-)wireless multi-hop networks. The interaction between TCP flow control and the wireless link bit-rate adaptation is further studied in Section 5. Section 6 discusses the TCP performance in hybrid wireless/wireline networks. We then examine different ways to alleviate contention and improve the TCP performance in Section 7. We conclude the paper in Section 8.

II. RELATED WORK

There have been extensive studies that investigated the performance of TCP over wireless multi-hop networks using simulation [3], [4], [5], [6], [7]. Fu et al. [6] identified the

self-interference problem in multi-hop networks and showed that there exists an optimal TCP CWND size, but TCP often increases its CWND to a much larger than the optimal value, resulting in reduced spatial reuse. Several link-layer schemes are developed to alleviate the problem. Chen et al. [3] and ElRakabawy et al. [4] made a similar observation about excessive TCP CWND growth. TCP CWND clamping [3] and TCP pacing [4] are proposed to improve the situation. ElRakabawy et al. [5] simulated the performance of different TCP versions (NewReno and Vegas) in wireless multi-hop networks and found that TCP Vegas significantly outperforms TCP NewReno in terms of throughput and fairness. Nahm et al. [7] studied the interaction between TCP and routing protocols in wireless multi-hop networks and proposed a fractional window increment scheme for TCP to prevent the over-reaction of the on-demand routing protocol.

A number of experimental investigations of TCP performance over multi-hop wireless networks have been also conducted [8], [9], [10], [11]. Anastasi et al. [8] investigated the impact of the maximum congestion window size and routing protocol on TCP performance on a static wireless testbed. Gupta et al. [9] studied TCP performance on a testbed taking into account factors such as routing protocols, user mobility, and the number of hops traversed in the network. Kawadia et al. [10] presented their experimental study of the effect of TCP CWND clamping, RTS/CTS handshake, and SACK on the TCP performance. Koutsonikolas et al. [11] made an observation about the effect of TCP CWND clamping and RTS/CTS on TCP performance over both fixed and adaptive bit-rate wireless links. Other researchers [8], [10], [11] observed that properly setting the TCP CWND limit improves the TCP performance, and [10] shows that RTS/CTS hurts TCP performance, while [11] discovers that RTS/CTS might help when autorate links are used. Our experimental study confirms the above observation and further focuses on the interaction between TCP flow control and link-layer rate adaptation and its impact on the TCP performance over hybrid wireless paths.

While optimizing TCP performance over 802.11 access networks has been studied extensively, the primary focus has been on single-hop wireless LANs. We are aware of only one previous study (by Su et al. [12]) that considers multi-hop access networks. The study focuses on fairness among competing flows. It demonstrates cases where severe unfairness happens and identifies the cause for the unfairness. This study covers in-depth the fairness problem and is complementary to our work.

III. EXPERIMENTAL SETUP

All the experiments are conducted on IvyNet [13], a miniaturized wireless network testbed deployed at ETH Zurich. IvyNet consists of 96 mesh routers set up in a 4x24 grid. Each router has one Atheros 5212 based 802.11a/b/g wireless radio operating in b mode. To reduce the transmission range, the wireless NIC is connected to a low-gain antenna through RF attenuators. Validation experiments [14], [15] show that the miniaturization technique based on attenuation can effectively

reduce the wireless NIC's communication range while still obtaining high-fidelity experimental results.

The wireless routers run Linux (2.6.16.13) with the Madwifi wireless driver [16]. We use iperf to generate TCP workloads and measure performance. Logs were collected using tcpdump for TCP traffic. Our custom kernel also reports the statistics of various TCP parameters. Static routing is used. For experiments that involve Internet hosts, we use iptables to setup a NAT on the IvyNet gateway. We use the TCP Reno implementation from Linux 2.6 for the experiments. Before each experiment, we set the congestion control algorithm via sysctl. TCP uses 1460 bytes MSS, which is the value used in commodity Ethernet. In addition, TCP-SACK is enabled for all the experiments. The TCP congestion window limit is changed using the ip command from the iproute2 package. We use the TCP Vegas implementation from Linux 2.6 with its default parameter settings for the TCP Vegas experiments. MAC retransmission is enabled in all experiments, with the MAC retry limit set to 10 (default from the Madwifi driver). We use the SampleRate bit-rate adaptation [17] implementation from the Madwifi driver, which has the best performance over wireless mesh networks [17]. Samplerate keeps a long-term estimation of link-layer throughput for all the possible bit-rates and selects the transmission rate that can achieve the highest link-layer throughput. As the result, links with SampleRate experience moderate rate fluctuation and a reduced packet loss rate compared to the links with a fixed 11Mbps transmission bit-rate. We use iwconfig to turn on/off the RTS/CTS handshake. The Madwifi driver maintains a drop-tail buffer with default size of 50 packets.

IV. WIRELESS MULTI-HOP NETWORKS

A. Single TCP flow

First, we study the TCP bulk transfer performance over static topologies consisting of 2-6 stations arranged in a chain. The stations are chosen so only adjacent stations can directly communicate at 11Mbps. We repeat the experiments for chain lengths ranging from 1 to 5 hops, and with links using a fixed 11Mbps bit-rate and using rate adaptation. Each experiment runs a single TCP flow over one of the 50 randomly selected paths. Figure 1 shows the results for the single TCP flow measurements. Each bar presents the medium throughput from 50 independent experiments and the two endpoints of the errorbars show the corresponding tenth and ninetieth percentile. We observe that TCP achieves almost the same throughput in 1-hop and 2-hop chains both with rate adaptation and with a fixed bit-rate. But when it operates over longer chains, TCP becomes less efficient with rate adaptation. For example, over 5-hop chains, TCP with rate adaptation achieves only 64% of the throughput when using 11Mbps fixed bit-rate.

To understand the detailed interaction between TCP and link-layer rate adaptation, we conduct a microscopic study of a single TCP flow over a 5-hop path. Figure 2(a) shows the TCP congestion window dynamics. We observe that with a fixed bit-rate, TCP experiences extensive congestion window fluctuation that limits the congestion window to no more than

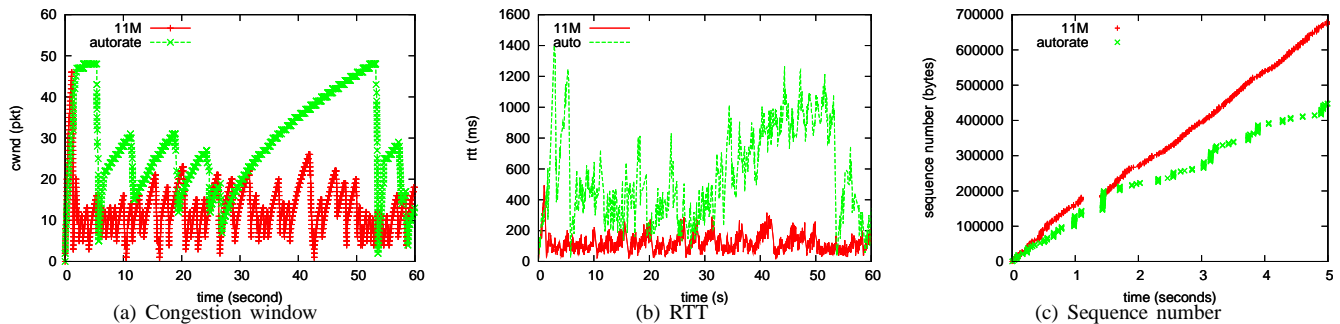


Fig. 2. TCP dynamics for single flow over 5-hop path

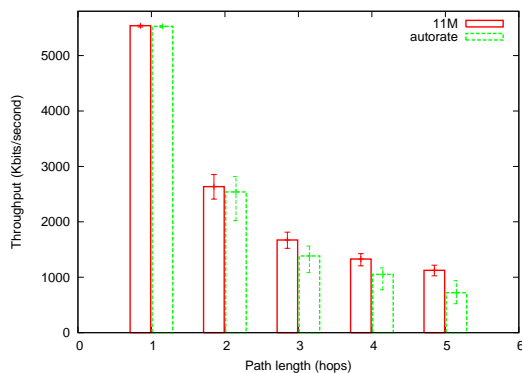


Fig. 1. TCP throughput

Bit-rate	TCP retry (pkt)	TCP tx (pkt)	Retx/tx
11M	83	5799	0.014
autorate	15	3025	0.005

TABLE I
TCP RETRANSMISSION: FIXED BIT-RATE VS. AUTO BIT-RATE

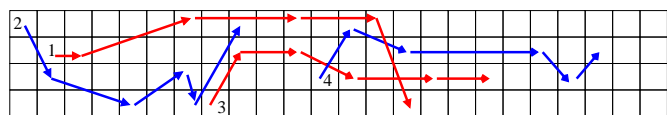


Fig. 3. Flow layout for multi-flow experiments

30 packets (except for the slow start phase). In contrast, with rate adaptation, TCP experiences fewer congestion window reductions and keeps a much larger congestion window. Figure 2(b) shows that the large congestion window for TCP over links with rate adaptation, not surprisingly, also results in an inflated RTT.

The drop in throughput is a result of the fact that rate adaptation reduces the packet loss rate on the wireless links. Table I confirms this: it shows that fewer TCP packets are retransmitted over the adaptive bit-rate links than over the fixed bit-rate links. In wireless multi-hop networks, packets from the same TCP flow may interfere with each other, a phenomenon known as self-interference [6]. TCP packets also compete with ACK packets. As TCP increases its congestion window, self-interference increases, and network efficiency is reduced; this observation is consistent with [3], [6]. Rate adaptation also increases the transmission time of packets, further increasing interference. These observations suggest that using bit-rate adaptation to aggressively reduce the packet loss rate can be counter productive.

Let us look at the dynamics of the congestion window in more detail, using Figure 2(a). When both flows are in the slow-start phase between 0s and 1.2s, they have similar behavior: as long as there is no packet loss, TCP increases its congestion window multiplicatively until it fills the interface buffer at the bottleneck (50 packets in the experiments). As a result, the TCP flows over paths with a fixed bit-rate and rate adaptation have similar throughput (Figure 2(c)). At the

end of slow-start, the TCP congestion window is more than 40 packets, and we start to have significant physical-layer losses in the network due to self-interference. In the fixed bit-rate case, link-layer retransmission cannot recover all the losses and TCP packets are dropped, leading to TCP timeouts. With rate adaptation, the physical-layer packet loss rate causes links to switch to a lower bit-rate. This step not only reduces the link-level packet loss rate but also reduces the capacity of the links since the packet transmission time is increased. Because of high load and the increased packet transmission time, physical layer losses remain, preventing bottleneck links from increasing their bit-rate. The result is lower throughput for TCP over rate adaptation links after 1.2s (Figure 2(c)).

B. Multiple TCP flows

Figure 4 shows the aggregate throughput for four TCP flows running simultaneously over four disjoint paths. The path layout is depicted in Figure 3. The TCP sender always resides on the first node of a path, and we start the TCP receiver at different nodes along the path to achieve various path lengths. Each point in the Figure 4 represents the average from three runs. Each run lasts 60 seconds.

The results are similar to the single-flow case: TCP over fixed bit-rate links significantly outperforms TCP over rate adaptation links in terms of aggregate throughput. For short paths (1-hop, 2-hop), the fixed-bitrate links achieve much better spatial reuse of the channel than the autorate links, e.g., TCP over 11Mbps links achieves 180% more aggregate throughput over 1-hop paths than with autorate links. For

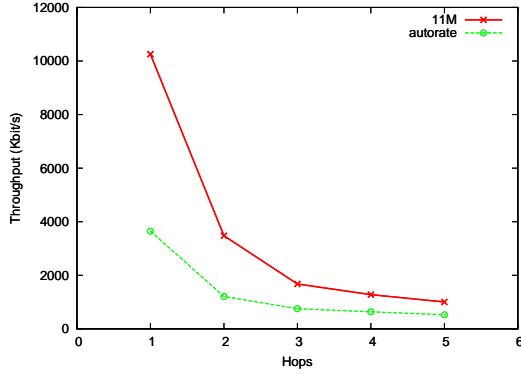


Fig. 4. Aggregate TCP throughput

Bit-rate	1 hop	2 hop	3 hop	4 hop	5 hop
11M	0.65	0.45	0.49	0.52	0.34
autorate	0.74	0.84	0.54	0.56	0.52

TABLE II
FAIRNESS INDEX

longer paths, the difference in aggregate throughput remains more than 91%.

We use the fairness index described by Jain et al. [18] to assess the fairness among competing TCP flows. The fairness index is a value between 0 and 1. A larger fairness index represents fairer resource allocation among competing flows. From Table II, we observe that, in contrast to the throughput results, TCP achieves better fairness with bit-rate adaptation links than with fixed bit-rate links over all the paths. With rate-adaptive links, fewer packets are dropped by the link layer, resulting fewer TCP timeouts, and leading to the better fairness.

V. INTERACTION BETWEEN TCP AND LINK RATE ADAPTATION

We develop a model of the TCP and bit-rate adaptation interaction to guide our investigation of different approaches.

A. Interaction Cycle

Figure 5 elaborates on how the cross-layer interactions between TCP and wireless MAC may lead to the TCP performance differences observed in the previous section. The entire cycle is initiated by TCP when a TCP sender sends data at a rate higher than the network capacity (Step A, B). The network overload leads to the increased packet loss rate at the physical layer (Step C). For fixed bit-rate links, this causes the link throughput to drop (Step D). Consequently, the network becomes even more overloaded (Step B). Staying in the cycle B-C-D, the physical layer eventually loses too many packets, so local retransmission can no longer recover, and the link starts to drop packets (Step F). The TCP sender senses the packet losses and reduces its congestion window (Step H), reducing the load on the network.

On the other hand, if wireless links use bit-rate adaptation, increased link losses (Step C) will trigger the link to switch

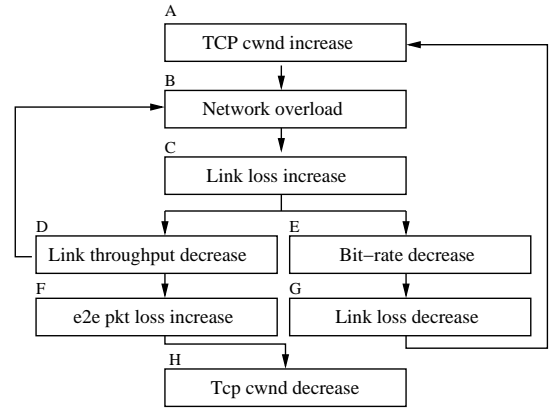


Fig. 5. Interaction: TCP vs. wireless MAC

Node id	11Mbps	5.5Mbps	2Mbps	1Mbps
0	11.1	79.4	5.4	4.1
1	29.8	13.8	50.5	5.9
2	97.9	1.9	0	0.18
3	75.5	23.5	0.1	0.9
4	97.7	2.2	0	0.14
5	0	0	0	0

TABLE III
BIT-RATE DISTRIBUTION (IN PERCENT) OF TCP DATA PKTS. NODE 0 IS THE TCP SENDER, NODE 5 IS THE TCP RECEIVER

to a lower bit-rate (Step E). As a result, the physical layer becomes more resilient to interference, thus reducing the number of losses at the physical layer (Step G). Local retransmission can now more often recover from remaining losses, so TCP Reno will further increase its congestion window (Step A) and the network becomes even more over-loaded (Step B). While the network stays in the loop A-B-C-E-G, the bottleneck link keeps reducing its transmission bit-rate and TCP continues to increase its congestion window size. Table III, which shows the bit-rate distribution on the links for the single flow case, confirms this reasoning. A high percentage of the packets on the first two hops are sent at the 5.5 and 2 Mbps bit rates. This arrangement significantly limits the TCP throughput.

The two cases also differ in how much time they spend in the A-B-C-E-G and B-C-D cycles. In our experiments with TCP over adaptive-rate links, the network spends quite a long time in the A-B-C-E-G loop. As shown in Figure 2(a), TCP maintains a large congestion window, and packet losses will be mostly caused by queue overflow at the bottleneck link. In contrast, in the experiments with fixed-rate links, the congestion window is much lower. The relatively small MAC retry limit (10) results in more frequent link-layer packet drops in a network with fixed bit-rate. These link-layer packet drops cause TCP to reduce its congestion window, so the network stays in the B-C-D cycle only for short time. Consequently, with a fixed bit rate, there is less contention and the packet transmission times are shorter, which translates into higher throughput.

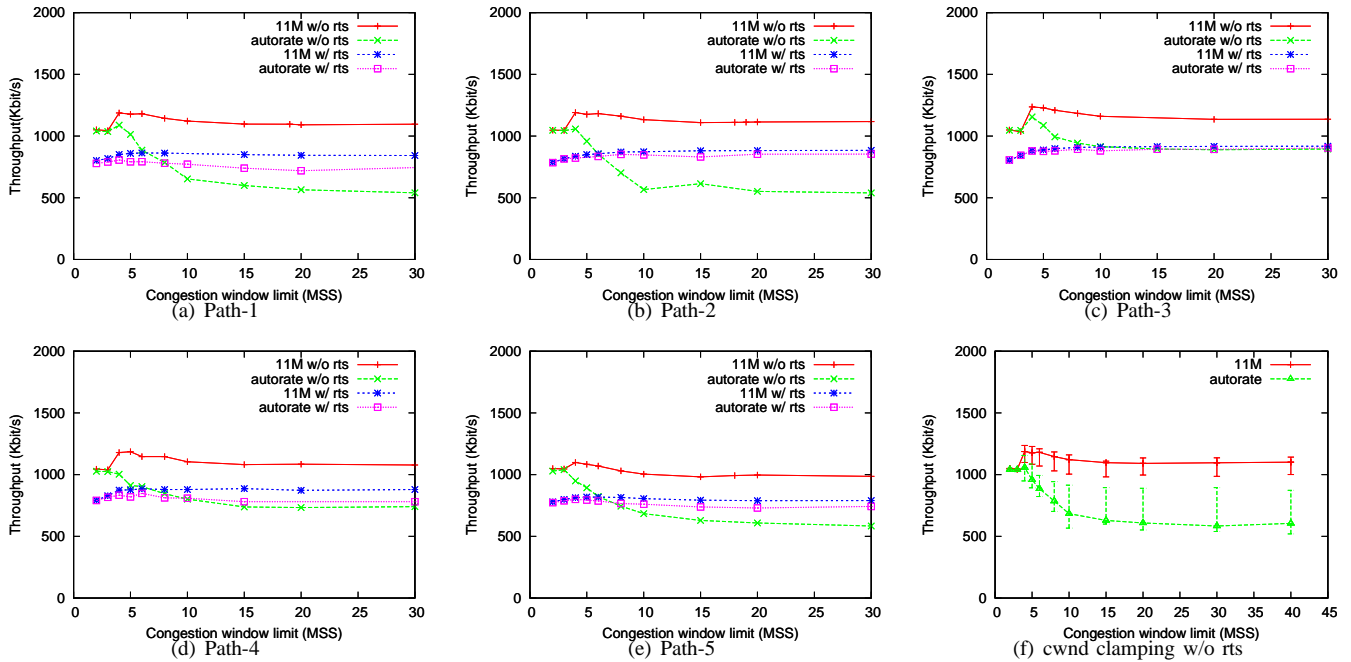


Fig. 6. Effectiveness of TCP congestion window clamping and RTS/CTS handshake

B. Effect of TCP congestion window clamping and RTS/CTS handshake

To confirm the above observations and to investigate the reasons for packet losses in our experiments, we repeat the TCP single-flow experiments (Section IV-A) with a controlled window limit and with RTS/CTS handshakes enabled. With TCP CWND clamping, we control TCP’s maximum congestion window, thus limiting the maximum number of packets a TCP sender can put into the network and controlling contention. The RTS/CTS handshake is designed to coordinate the network transmissions when there are hidden terminals. By comparing the results w/ and w/o RTS/CTS, we can infer the significance of hidden terminals in our experiments. The experiments are conducted over five 5-hop paths. Each data point in the Figure 6(a)-6(e) shows the average throughput from three runs of the same experiment.

Let us first consider the results without RTS/CTS. As we increase the CWND limit, we observe that for all paths TCP achieves improved throughput until the network becomes overloaded. Further increasing the CWND limit hurts the TCP throughput. With fixed bit-rates, the TCP throughput drops only slightly before it stabilizes. This indicates that the network enters the cycle B-C-D when the network gets overloaded; i.e., the network starts to drop TCP packets. This prevents TCP from overly increasing its congestion window and the network contention level remains moderate. With adaptive bit-rate links, TCP throughput drops more significantly when we increase the TCP CWND limit: the increased contention causes the network to operate in the A-B-C-E-G cycle. When the CWND limit is larger than 10 MSS, the network becomes saturated. Once the packet queues have

built up, the contention level no longer increases and TCP throughput stabilizes as well.

Figure 6 also presents the effect of RTS/CTS handshake on our experiments. When the network operates with saturated contention level (i.e., CWND limit > 10 MSS) over adaptive bit-rate links, using RTS/CTS significantly improves the TCP throughput for three out of five paths. For the other two paths, TCP achieves similar throughput w/ and w/o RTS/CTS. Path-1 is the path used in the microscopic study (Section IV-A). When the CWND limit is large, TCP over Path-1 achieves 45% higher throughput with RTS/CTS than without RTS/CTS. The effect of RTS/CTS indicates that the hidden terminal problem is the major cause of packet losses in the experiments with adaptive bit-rate links when the TCP CWND limit is large. However, when TCP runs over fixed bit-rate links, the network operates with moderate contention. Fewer packets are lost due to hidden terminal problems and the benefit of using RTS/CTS does not compensate for its overhead. Hence using RTS/CTS over fixed bit-rate links is always detrimental to TCP in the experiments.

VI. TCP OVER HYBRID WIRELESS/WIRELINE NETWORKS

In this section, we extend our experiments to paths that travel over both a multi-hop wireless network and the wireline Internet. We extended the five 5-hop wireless paths used in Section V-B to two remote sites, one at Europe and another at US. Table IV shows the path characteristics. The wireline (bottleneck) bandwidths are measured between the testbed gateway and remote sites with iperf, while the wireline RTTs are measured using ping with 1500 byte payloads. Both of the measurements are carried out before we started TCP tests. The bandwidths and RTTs of the hybrid paths are measured

Path	Wireline RTT (ms)	RTT (ms)	Wireline BW (Kbps)	BW (Kbps)	BDP (MSS)
Mesh-Europe	31	57	17596	1077	6
Europe-Mesh	31	56	34747	1074	6
Mesh-US	115	140	4719	1058	13
US-Mesh	115	140	3500	919	11

TABLE IV
CONFIGURATION OF THE INTERNET TEST.

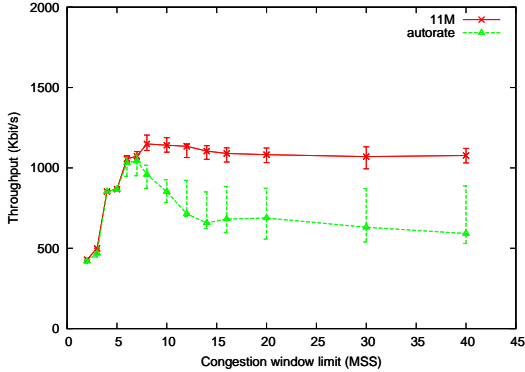


Fig. 7. Mesh-Europe

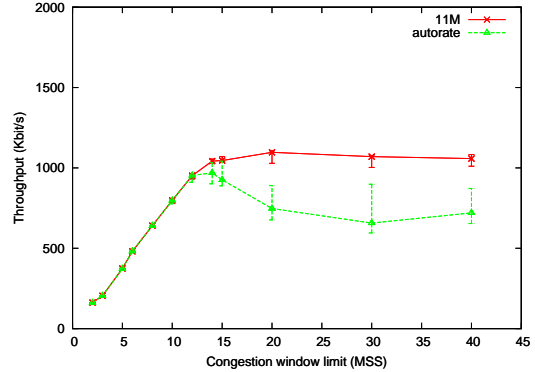


Fig. 8. Mesh-US

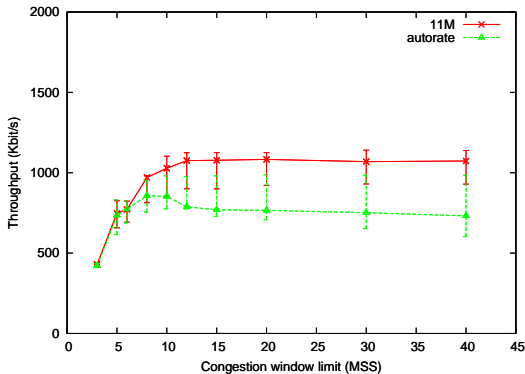


Fig. 9. Europe-Mesh.

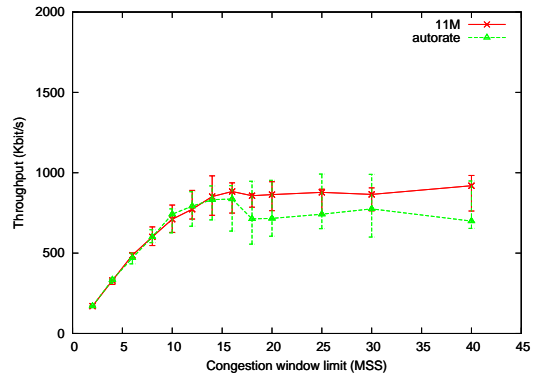


Fig. 10. US-Mesh.

in a similar manner between the wireless hosts and the remote sites with all the wireless links using 11Mbps bit-rate.

We categorise the hybrid paths into four groups: Mesh-Europe, Mesh-US, Europe-Mesh, and US-Mesh. Each group consists of five paths. The paths from the same group share the same Internet path but use different wireless paths. In Table IV, the figures for the hybrid paths represent the average over 5 paths from the same group. We ran a single TCP flow experiment over the hybrid paths with different TCP congestion window limits (RTS/CTS was turned off). The results are presented in Figure 7, 8, 9, and 10. For comparison, Figure 6(f) shows the results for all-wireless paths (Section V-B). Each data point in the figures shows the medium measurement value from the five paths. The two endpoints of the error bars show the corresponding maximum and minimum value.

We observe that TCP behaves similarly over hybrid paths and all-wireless paths. When the TCP congestion window limit increases, for the hybrid paths as for the all-wireless paths, bit-rate adaptation on wireless links eventually leads to

increased congestion and reduced TCP throughput. In contrast, TCP only experiences a small drop in throughput when the wireless links use 11Mbps fixed bit-rate. Consequently, even over the same path, there is a significant performance gap between TCP with fixed bit-rate wireless links and TCP with adaptive bit-rate wireless links. Furthermore, when the multi-hop wireless segment uses adaptive bit-rate links, there exists an optimal CWND limit, with which TCP achieves the highest throughput. Further increasing the congestion window will overload the wireless links and leads to reduced TCP throughput, although for the downstream links (Figure 9, Figure 10), the drop in throughput is more moderate probably because the packets spread out on the Internet before they arrive at the wireless bottleneck.

Fu et al. [6] show that in all-wireless multi-hop networks there exists a topology-dependent optimal CWND limit for a single TCP flow. As shown in Figure 6(f), the optimal CWND limit for the all-wireless paths in our experiments is 4 MSS. For the hybrid paths, our experiment results indicate

Test Scenario	Description
TCP Reno	The baseline TCP
Clamping 3	Reno with CWND limit set to 3
Clamping 4	Reno with CWND limit set to 4
Buf 2	Reno with router buffer size of 2
Buf 4	Reno with router buffer size of 4
Buf 50	Reno with router buffer size of 50. The default setting in Madwifi driver
TCP Vegas	TCP Vegas from linux 2.6.16

TABLE V
OPTIMIZATION APPROACHES.

that the optimal TCP CWND limit is also influenced by the wireline part, even when the bottleneck is on the wireless part. Table IV shows the average Bandwidth Delay Product (BDP) of the hybrid paths. Not surprisingly, TCP achieves the best throughput when the CWND limit matches the BDP on the path. When the BDP of the path increases, TCP requires a larger CWND limit to achieve the best performance, e.g., TCP achieves the optimal throughput over the Europe paths with a TCP CWND limit of 7 MSS, while the US paths requires a much larger CWND limit of 13 MSS.

VII. POSSIBLE SOLUTIONS

Contention introduced interferences are the primary reason that the bit-rate adaptation hurts TCP. In this section, we analyze various approaches, at both TCP-level and link-level, to alleviate contention and improve the TCP performance in the multi-hop wireless access networks:

- 1) *TCP CWND clamping*. In Section V-B we showed that setting the CWND limit to an appropriate value significantly improves the TCP throughput over 5-hop paths. In this section, we study the effectiveness of CWND clamping in more general settings, e.g., with competing traffic or multiple hop lengths.
- 2) *TCP Vegas*. As we indicated in Section V-A, TCP Reno cannot detect the incipient signs of network congestion before losses occur and thus often causes the congestion collapse when operating over autorate links. TCP Vegas enhances TCP Reno with an innovative congestion control scheme. A source in TCP Vegas anticipates the onset of congestion by monitoring the difference between the throughput it is expecting to see and the actually realized throughput. TCP Vegas tries to proactively detect congestion in its incipient stage, and subsequently reduces the TCP window in an attempt to resolve the congestion before it causes any packet loss. It thereby may prevent the network from entering the congestion collapse loop shown in Figure 5.
- 3) *Buffer tuning*. On wired Internet, buffer overflow at intermediate routers is the primary cause for congestion collapse. The conventional wisdom is to equip the routers with more and more buffer to mitigate the buffer overflow problem caused by traffic bursts. In contrast, in a multi-hop wireless network, contention is the primary cause of degraded network capacity. Since

Hops	Test scenario	Aggregate thr. (Kbps)		Fairness index
1 Hop	TCP Reno	5377	0%	0.9955
	CWND clamping 3	5594	4%	0.9999
	TCP Vegas	5575	4%	0.9980
	Buf 2	4321	-20%	0.8922
2 Hop	TCP Reno	2766	0%	0.9963
	CWND clamping 3	2835	3%	0.9999
	TCP Vegas	2841	3%	0.8640
	Buf 2	2188	-21%	0.7884
3 Hop	TCP Reno	1502	0%	0.9582
	CWND clamping 3	1576	5%	0.9996
	TCP Vegas	1564	4%	0.9510
	Buf 2	1432	-5%	0.7775
4 Hop	TCP Reno	1126	0%	0.9586
	CWND clamping 3	1116	-1%	0.9992
	TCP Vegas	1146	2%	0.9832
	Buf 2	1206	7%	0.6793
5 Hop	TCP Reno	824	0%	0.9806
	CWND clamping 3	850	3%	0.9985
	TCP Vegas	874	6%	0.9945
	Buf 2	1156	40 %	0.7856

TABLE VII
FOUR SIMULTANEOUS FLOWS SHARE THE SAME PATH (WIRELESS ONLY)

the contention level is a function of the number of nodes with non-empty queues, limiting the maximum length of the buffers at routers to a very small number, i.e., 1 packet, may actually reduce the number of backlogged nodes, and hence, keep the contention level low. Because the forward flow and the backward flow in a multi-hop network may traverse the same set of intermediate nodes, we set the maximum length of router buffer to 2 packets in our experiments to reduce the contention between TCP data packets and ACK packets.

We perform this analysis using experiments over the IvyNet testbed. Table V describes the test scenarios.

A. Multiple TCP flows

When there are multiple flows over multiple paths, there is significant contention in the network. We conduct multi-flow tests, using the same setting as the multi-flow experiments in Section IV-B, for each of the possible solutions. Autorate is turned on for all the experiments. Table VI summarizes the results for the wireless-only setup. The throughput for each path is the average of three runs. The percentage numbers in the table show the improvement of the aggregate throughput relative to the baseline case (TCP Reno). We observe that over 1-hop paths, Vegas and Buf2 achieve significant throughput improvements over TCP Reno, while CWND clamping shows better fairness. For longer paths, Buf2 and Clamp3 perform the best in both throughput and fairness. Overall, the simple link layer approach Buf2 effectively alleviates the contention level in the network and shows the best results both with respect to throughput and fairness.

B. Congested path

TCP's ability to share a bottleneck fairly and efficiently decreases dramatically as there are more flows than packets in the delay-bandwidth product. This is known as the TCP many

Hops	Test scenario	Path-1 (Kbps)	Path-2 (Kbps)	Path-3 (Kbps)	Path-4 (Kbps)	Aggregate thr. (Kbps)		Fairness index
1 Hop	TCP Reno	986	443	443	1777	3649	0%	0.74
	Clamping 3	1008	623	552	1540	3723	2%	0.85
	Clamping 4	1083	530	486	1719	3818	5%	0.78
	TCP Vegas	1641	866	664	2674	5845	60%	0.77
	Buf 2	1617	1121	299	2285	5322	46%	0.77
2 Hop	TCP Reno	144	276	278	511	1209	0%	0.84
	Clamping 3	254	351	294	460	1359	12%	0.95
	Clamping 4	202	287	280	468	1237	2%	0.91
	TCP Vegas	182	251	246	554	1233	2%	0.82
	Buf 2	261	404	233	449	1347	11%	0.93
3 Hop	TCP Reno	4	43	275	432	753	0%	0.53
	Clamping 3	13	104	332	452	901	20%	0.62
	Clamping 4	72	95	234	445	846	12 %	0.67
	TCP Vegas	46	78	236	469	829	10%	0.61
	Buf 2	8	227	328	351	914	21%	0.74
4 Hop	TCP Reno	6	67	187	379	639	0%	0.56
	Clamping 3	7	112	255	341	715	12%	0.66
	Clamping 4	10	136	181	386	713	12%	0.63
	TCP Vegas	5	81	190	389	665	4%	0.57
	Buf 2	13	184	249	284	730	14%	0.75
5 Hop	TCP Reno	4	48	142	331	525	0%	0.52
	Clamping 3	10	106	198	277	591	13%	0.69
	Clamping 4	3	79	206	262	550	5%	0.65
	TCP Vegas	27	54	157	319	557	6%	0.60
	Buf 2	23	141	201	241	606	15%	0.77

TABLE VI
FOUR DISJOINT SIMULTANEOUS FLOWS OVER THE SAME FOUR PATHS AS IN FIG.3 (WIRELESS ONLY)

flow problem [19]. One way to work around the problem is to equip routers with larger buffers, i.e., not just one round-trip time worth of buffering, but buffering proportional to the total number of active flows. An approach like using the drop-tail buffers with the minimum length can effectively reduce the contention level in a multi-hop network but at the same time, this solution makes the many flow problem more severe. We conducted an experiment to study this trade-off over the set of nodes from Path-1 in Figure 3. The TCP senders always reside on the first node of a path, and we start the TCP receiver at different nodes along the path to achieve various path lengths. The measured delay-bandwidth product for the 5-hop path is 3 packets. We run 4 simultaneous TCP flows over the same path for each test. Table VII presents the results. We observe that since the number of active flows is not significantly larger than the BDP of the path, Reno can still achieve good fairness among competing flows. With the aggregation of multiple flows, CWND clamping and TCP Vegas only slightly outperform Reno. Meanwhile, using small buffers suffers a lot from the many flow problem. Over paths shorter than 3 hops, using small buffers achieves at least 20% less throughput and worse fairness than using larger buffers. For long paths, using small buffers effectively alleviates the contention level in the network, and leads to a significantly improved aggregate throughput. However, the unfairness among the competing flows still remains.

C. Hybrid networks

Section VI shows that TCP clamping works for hybrid paths, i.e., TCP achieves the best performance when we set (clamp) its congestion window limit to the BDP of the path.

Hops	Test scenario	Thr. to US (Kbps)		Thr. to EU (Kbps)	
1 Hop	TCP Reno	4393	0%	5442	0%
	TCP Vegas	3345	-24%	4913	-10%
	Buf 2	470	-89%	2736	-50%
2 Hop	TCP Reno	2387	0%	2438	0%
	TCP Vegas	1890	-21%	2494	2%
	Buf 2	391	-84%	1410	-42%
3 Hop	TCP Reno	1174	0%	1173	0%
	TCP Vegas	1156	-2%	1350	15%
	Buf 2	340	-71%	813	-31%
4 Hop	TCP Reno	874	0%	863	0%
	TCP Vegas	940	8%	1018	25%
	Buf 2	265	-70%	688	-15%
5 Hop	TCP Reno	620	0%	618	0%
	TCP Vegas	713	15%	741	20%
	Buf 2	276	-55%	563	-9%

TABLE VIII
SINGLE TCP FLOW OVER HYBRID PATHS (MESH TO REMOTE HOSTS)

In this section, we conduct further experiments to study the performance of TCP Vegas and buffer tuning over hybrid paths. We use the same experimental setup as in Section VI¹ Results are summarised in Table VIII.

We observe that TCP Vegas is conservative over the wired Internet, and achieves less throughput than TCP Reno over all the paths with a 1-hop wireless segment. In contrast, over paths with longer wireless segments, TCP Vegas's delay-sensitive congestion control scheme starts to help. As a result, Vegas outperforms Reno by 15% and 20% in throughput over the 5-hop Mesh-US paths and Mesh-Europe paths respectively. In

¹The Europe host has now a round trip time of 17.9 ms to the Mesh gateway; the available bandwidth between the Mesh gateway and this host (measured by iperf) is 14.9 Mbps downstream and 31.4 Mbps upstream.

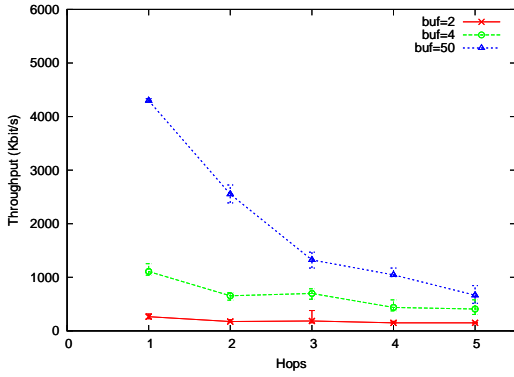


Fig. 11. US-Mesh.

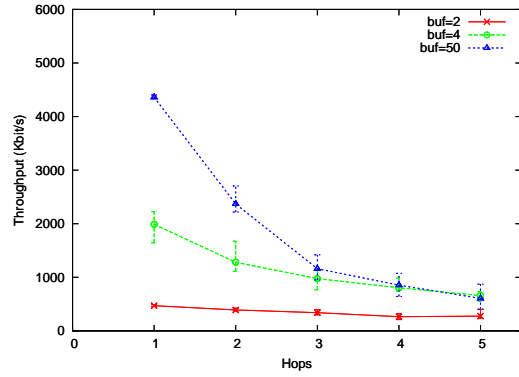


Fig. 12. Mesh-US.

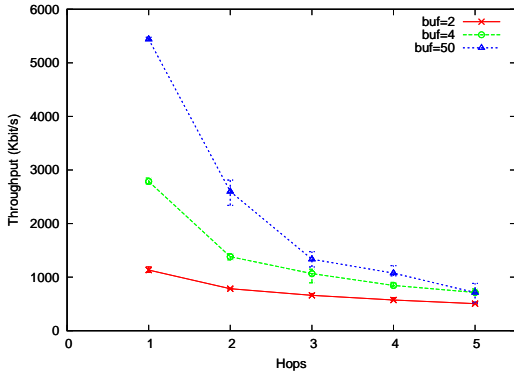


Fig. 13. Europe-Mesh.

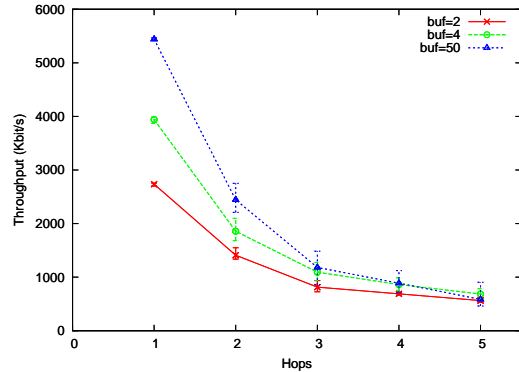


Fig. 14. Mesh-Europe.

addition, limiting the length of buffer to 2 packets significantly degrades the TCP throughput over all the experiments.

We conduct further experiments to study the impact of buffer size on TCP performance. In the experiments, we run TCP Reno between the remote hosts and wireless hosts while changing the buffer size at the wireless routers. Figure 11, 12, 13, and 14 present the results. We observe that overall, using small buffers hurts TCP performance over hybrid paths, especially for paths with a small number of wireless hops. Table IX shows the average CWND of TCP over upstream paths. With small buffer, TCP is unable to fully open its CWND to match the BDP of the path, e.g., over the 1-hop Mesh-US path, when we set the length of the buffer to 2, TCP maintains an average CWND of 5.6 MSS while the BDP of the path is 45 MSS. As a result, TCP achieves only 11% of the throughput obtained when wireless routers use the default buffer size of 50 packets. However, as the length of the wireless segment of a path increases, the performance gap between small buffers and large buffers rapidly diminishes.

D. Discussion

Although TCP is able to efficiently make use of the network capacity in paths with few wireless segments, there is still room for improvement for paths with many wireless segments, i.e., when there are more than 3 wireless hops. Any solution to improve the efficiency of TCP over multi-hop access networks should meet two requirements: First, TCP should be able to fill

its share of the end-to-end network pipe to fully take advantage of the pipelining effect of packet transmission. Second, the solution should also keep the contention level in the wireless part as low as possible to achieve better bottleneck bandwidth. Approaches we studied in this section are effective in certain scenarios, but none of them meet both requirements all the time in practice.

TCP level solutions estimate the fair share of the BDP for a flow and regulate TCP to not overshoot its share of the BDP. TCP CWND clamping achieves this goal by explicitly setting the CWND limit. It works well only when TCP knows the correct CWND limit. Although solutions [3] have been developed to estimate a tight upper bound of the CWND limit based on hop count or per-hop delay for wireless multi-hop networks, estimating a precise CWND limit with low overhead for a hybrid path remains challenging. TCP Vegas's congestion control scheme aims to proactively detect and react to congestion in its incipient stages, and hence, TCP Vegas experiences less BDP overshooting problems than Reno without the need to know the CWND limit. Our experiments confirm the effectiveness of this approach; TCP Vegas achieves higher throughput than Reno when the wireless segment of a path is long. However, TCP Vegas becomes over conservative when it competes with other traffics in the wired path, consequently, Vegas achieves less throughput than Reno over paths with a short wireless segment. When multiple flows share the same path, both CWND clamping and TCP Vegas are not effective

Buffer size	Mesh-US					Mesh-Europe				
	1 hop	2 hop	3 hop	4 hop	5 hop	1 hop	2 hop	3 hop	4 hop	5 hop
2	5.6	4.9	4.3	4.4	4.5	5.3	4.1	3.4	3.1	3.1
4	24.2	17.1	13.5	11.1	8.5	8.8	5.9	5.2	4.9	4.8
50	68.7	54.4	42.6	28.0	16.7	38.4	48.0	29.6	25.4	11.9

TABLE IX
AVERAGE CONGESTION CONTROL WINDOW SIZE (MSS)

in alleviating the contention level in the network anymore and, hence, achieve similar throughput as TCP Reno.

Tuning the buffer size at the link layer turns out to be the most effective method to alleviate the contention level in a wireless network in our experiments. However a wireless access network with small drop-tail buffers also suffers from other kinds of performance problems, e.g., when many flows share the same path and overly conservative behavior over hybrid paths. Recent work [20], [21] also investigated the effectiveness of buffer tuning in multi-hop wireless networks based on simulation. Their solutions, which combine buffer management techniques with link-layer flow control, aim to take advantage of the small buffering and at the same time eliminate its side effects. These approaches are interesting and worth further investigation in testbed or real-world experiments.

Several bit-rate adaptation algorithms [22], [23], [24] have been proposed to relieve the self-interference problem by discriminating between frame losses due to radio channel impairments from those due to self-interference. The interaction between those algorithms and TCP remains an interesting problem and worth further study.

VIII. CONCLUDING REMARKS

We conducted an experimental study of the TCP performance over a wireless testbed with an emphasis on the interaction between TCP flow control and 802.11 link bit-rate adaptation over multi-hop wireless paths. The results reveal that when operating over paths with adaptive bit-rate links, TCP becomes very sensitive to its congestion window size, and the current TCP implementations are often overly aggressive in increases their congestion window, resulting in significantly reduced throughput. We further examine various ways of reducing contention in the wireless network, and study their effectiveness on improving the TCP performance over multi-hop wireless access networks. Our results show that while the approaches often help in reducing contention in some scenarios, none of the approaches constitutes a general solution to the problem of efficient TCP over multi-hop wireless access networks, and further investigation into this problem is needed.

This work is a first step towards understanding the dynamics of large-scale wireless multi-hop access networks. As these networks become more prevalent, it may be necessary to reconsider some of the design choices made earlier, e.g., it may be worthwhile to consider cross-layer control of features (like autorate adaptation) that attempt to help the higher layers.

ACKNOWLEDGEMENT

We appreciate assistance by Peter Druschel, Roger Karrer, Nancy Steadle and James Stewart in performing measurements.

REFERENCES

- [1] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," in *Proc. ACM SIGCOMM*, 1996.
- [2] D. Eckhardt and P. Steenkiste, "Improving Wireless LAN Performance via Adaptive Local Error Control," in *Proc. IEEE ICNP*, 1998.
- [3] K. Chen, Y. Xue, and K. Nahrstedt, "On Setting TCP'S Congestion Window Limit in Mobile Ad Hoc Networks," in *Proc. IEEE ICC*, 2003.
- [4] S. ElRakabawy, A. Klemm, and C. Lindemann, "TCP with Adaptive Pacing for Multihop Wireless Networks," in *Proc. MOBIHOC*, 2005.
- [5] S. ElRakabawy, C. Lindemann, and M. Vernon, "Improving TCP Performance for Multihop Wireless Networks," in *Proc. ACM DSN*, 2005.
- [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. INFOCOM*, 2003.
- [7] K. Nahm, A. Helmy, and C. Kuo, "TCP over Multihop 802.11 Networks: Issues and Performance Enhancement," in *Proc. MOBIHOC*, 2005.
- [8] G. Anastasi, E. Ancilloti, M. Conti, and A. Passarella, *Experimental Analysis of TCP Performance in Static Multi-Hop Ad Hoc Networks*. Nova Science Publisher, 2007.
- [9] A. Gupta, I. Wormsbecker, and C. Williamson, "Experimental Evaluation of TCP Performance in Multi-Hop Wireless Ad Hoc Networks," in *Proc. IEEE MASCOTS*, 2004.
- [10] V. Kawadia and R. Kumar, "Experimental Investigations into TCP Performance over Wireless Multihop Networks," in *Proc. ACM E-WIND Workshop*, 2005.
- [11] D. Koutsonikolas, J. Dyaberi, P. Garimella, S. Fahmy, and Y. C. Hu, "On TCP Throughput and Window Size in a Multihop Wireless Network Testbed," in *Proc. ACM WiNTECH*, 2007.
- [12] K. Su, S. Bae, S. Lee, and M. Gerla, "TCP Behavior across Multihop Wireless Networks and the Wired Networks," in *Proc. WoWMoM*, 2002.
- [13] IvyNet wiki. [Online]. Available: <https://trac.id.ethz.ch/projects/IvyNet>
- [14] Y. Su, M. Heule, and T. Gross, "IvyNet: A Testbed for Multi-Hop Wireless Network Research," ETH Zurich, Tech. Rep., 2006.
- [15] P. De, A. Raniwala, S. Sharma, and T. C. Chiueh, "Mint: A Miniaturized Network Testbed for Mobile Wireless Research," in *Proc. IEEE INFOCOM*, 2005.
- [16] Madwifi driver. [Online]. Available: <http://madwifi.org>
- [17] J. Bicket, "Bit-Rate Selection in Wireless Networks," Master's thesis, Massachusetts Institute of Technology, Cambridge, 2005.
- [18] R. Jain, D. M. Chiu, and W. R. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System," Digital Equipment Corporation, Tech. Rep., 1984.
- [19] R. Morris, "TCP Behavior with many Flows," in *Proc. ICNP*, 1997.
- [20] O. Dousse, "Revising Buffering in Multihop CSMA/CA Wireless Networks," in *Proc. IEEE SECON*, 2007.
- [21] C. Lim, H. Y. Luo, and C. H. Choi, "Rain: A Reliable Wireless Network Architecture," in *Proc. IEEE ICNP*, 2006.
- [22] C. Hoffmann, M. Manshaei, T. Turletti, "CLARA: Closed-Loop Adaptive Rate Allocation for IEEE 802.11 WirelessLANs," in *Proc. WiMob*, 2005.
- [23] J. Kim, S. Kim, S. Choi, and D. Qiao, "CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs," in *Proc. IEEE INFOCOM*, 2006.
- [24] S. Rayanchu, A. Mishra, D. Agrawal, S. Saha, and S. Banerjee, "Diagnosing Wireless Packet Losses in 802.11: Separating Collision from Weak Signal," in *Proc. IEEE INFOCOM*, 2008.