Experimental Investigations into TCP Performance over Wireless Multihop Networks[*]

Vikas Kawadia
BBN Technologies, Cambridge, MA
vkawadia@bbn.com

P. R. Kumar
University of Illinois at Urbana-Champaign,
Urbana, IL
prkumar@uiuc.edu

ABSTRACT

The results of an extensive experimental study of the performance of the TCP protocol over wireless multi-hop ad hoc networks are presented. The investigations are performed in a real indoor environment over a network of laptops equipped with off-the-shelf IEEE 802.11b wireless cards. The cards were partially covered with copper tape to reduce their range, which enabled creation of manageable topologies. Several tools were written and assembled to make the entire process of experimentation including topology setup, traffic generation, trace collection, and archival and analysis of data repeatable, reliable and as automated as possible. The experimental observations are subjected to a thorough statistical analysis. The final result of the study is a recommendation of some TCP and IEEE 802.11 parameters that are best for TCP performance over wireless multi-hop networks. The most critical of these include setting a destination dependent clamp on the sender congestion window and disabling the RTC-CTS handshake. The methods and techniques used, as well as the support tools developed, and statistical analysis, may be of larger interest in wireless network experimentation.

Categories and Subject Descriptors

C.2.2 [**Computer Systems Organization**]: Computer Communication Networks—*Network Protocols*

General Terms

Experimentation

Keywords

Ad hoc networks, Experimentation, TCP

1. INTRODUCTION

The performance of TCP over wireless networks has been studied actively in the past few years. The observation that TCP does not distinguish between losses due to congestion and link quality related link layer losses has been the basis of a lot of studies and proposed modifications, which have been summarized nicely in [1] and [2]. However, this is not a problem in multi-hop ad hoc networks since IEEE 802.11, the dominant MAC protocol in use in multihop ad hoc networks, uses per-hop link layer ACKS which effectively screens TCP from losses at the link layer. The necessity of link-layer ACKs for an efficient wireless MAC protocol for ad hoc networks has been established through several studies, for example [3]. However node mobility causes lots of new problems in wireless multi-hop networks which have been the focus of several studies [4, 5, 6, 7, 8, 9, 10, 11]. Many solutions are feasible when feedback is available from the network layer about disruptive events such as link breakage.

However, it is not clear if current TCP would perform well even over relatively static multihop wireless ad hoc networks. Good performance of TCP over static networks is certainly a prerequisite for good performance over networks with mobility. Also problems caused by mobility are orthogonal to factors which affect performance over static multi-hop networks. Our concern is investigating factors which affect TCP performance over wireless multi-hop networks even in the absence of mobility. This issue has not received great attention, see Section 5 for a survey.

Another distinguishing feature of our investigation is that it is based on extensive experiments performed over a real testbed. It is important to conduct real experiments in studying wireless multihop ad hoc networking systems. Such experiments can provide a valuable baseline to identify important issues, provide reference values for simulation studies, and be used to set parameters of protocols appropriately to optimize performance. Since these systems are dynamic and difficult to model accurately, purely simulation studies could be quite removed from reality.

Several factors make a thorough experimental study challenging. These include methods for setting up desired topologies reliably and repeatably, conducting multihop experiments in a building environment of limited area, dealing with the constraints of control and measurements when working with off-the-shelf equipment, designing experiments so that one can study the actual experimental factors of interest, managing and presenting the huge volume of data that one typically collects in a manner comprehensible to the experimenter, and finally, drawing reliable conclusions. We address these challenges using a medley of several unique approaches which are described in Section 3.

Another important feature of our study is that the experiments over the ad hoc networking testbed are followed by a thorough statistical analysis, as per good practice of scientific experimentation

---

[*]This material is based upon work partially supported by AFOSR under Contract No. F49620-02-1-0217, NSF under Contract Nos. NSF ANI 02-21357 and CCR-0325716, USARO under Contract Nos. DAAD19-00-1-0466 and DAAD19-01010-465, DARPA/AFOSR under Contract No. F49620-02-1-0325, DARPA under Contact Nos. N00014-0-1-1-0576 and F33615-0-1-C-1905.

SIGCOMM'05 Workshops, August 22–26, 2005, Philadelphia, PA, USA.

in other fields. The analysis establishes that the improvements observed are not the result of random environmental artefacts, but are the result of changes that we introduce in the protocols.

To summarize, this works makes the following contributions: It undertakes the task of thorough experimental investigations over a multi-hop wireless networking testbed. The testbed and the associated software is set up so that experiments can be performed reliably and repeatably. Several issues affecting TCP performance are identified, which then lead to design of experiments followed by statistical analysis of the results. The final contribution is a set of recommendations for TCP which improve performance in wireless multihop networks. These are presented next.

2. MAIN CONCLUSIONS FROM EXPERIMENTS

TCP performance is potentially affected by many different parameters and factors at various layers of the network stack. The design of the actual experiments was preceded by several investigations to understand the interaction between various parts of the system and identify the parameters which made a critical difference to TCP performance. We then performed extensive experimentation and subsequent statistical analysis, to determine the combination of these parameters that is best for TCP performance. We present the main conclusions from our experimental investigations and subsequent statistical analysis.

1. Imposing a hard limit or a clamp on the sender congestion window significantly improves the end-to-end delay and delay jitter of a TCP connection. In our experiments, clamping the congestion window to $\lceil \frac{3}{2}n \rceil$, where n is the number of hops to be traversed by that flow, resulted in an order of magnitude improvement in delays and delay jitters. This is because, in IEEE 802.11 wireless ad hoc networks, the number of packets in flight is limited by the per-hop acknowledgements at the MAC layer. The TCP congestion window however grows far beyond this value, leading to excessive build-up of queues at the intermediate nodes, hence large delays and delay-jitters. Clamping the congestion window on a per-connection basis is a simple mechanism which enables more effective utilization of network resources. See Section 4 for more discussion.

2. Turning off the RTS-CTS handshake results in a statistically significant improvement in throughput, delay and delay jitter. Connections with fewer hops experience greater improvements. It is difficult to determine the exact cause of this improvement as various factors interact in a complicated way when both RTS-CTS and carrier sensing (which cannot be turned off on the commercial off-the-shelf cards) are on. Matters are further complicated by the different carrier sensing and communication ranges, both of which are also a function of the data rate. See [12] for a more elaborate discussion.

3. The TCP Selective acknowledgement (TCP-SACK) option does not have any statistically significant effect on TCP performance in our experiments. With no window clamping, SACK could aggravate buffer build-up problems at intermediate nodes by reducing feedback to the sender causing it to ramp up the congestion window, thus resulting in increased delays and delay jitters. But our experiments do not demonstrate any such negative effect and we recommend turning the TCP-SACK option on when congestion window clamping is on.

We evaluate all possible combinations of these modifications on TCP performance in different scenarios or topologies. During these experiments several other parameters were fixed at values which were determined to be good from a smaller set of experiments. These settings are as follows:

1. MAC *retry limit*, i.e., the maximum number of retransmission attempts for a packet at the MAC layer, is set to 8. A lower MAC *retry limit* is possibly useful only when there are multiple intersecting flows of at most three hops. Section 4 comments a little more on this issue.

2. The MTU is set to 2276 bytes which is the maximum possible value for the Cisco 350 cards such that there is no fragmentation of packets at the network layer or the MAC layer. Lower MTU increases the packetization overhead in our experiments.

3. The fragmentation threshold at the MAC layer is set to 2312 bytes, the maximum possible value allowed by the IEEE 802.11 specifications. Packets of size greater than the fragmentation threshold are fragmented at the MAC layer and sent in smaller packets. A high value for this threshold reduces the packetization overhead in our experiments.

4. The data rate was set to 11 Mbps and auto rate adaptation was turned off. However one should note that broadcast packets like RTS and CTS are always sent at the base rate of 1 Mbps in IEEE 802.11.

3. BRIEF DESCRIPTION OF THE EXPERIMENTAL SETUP

We now briefly describe our experimental setup. The nodes in our testbed are off-the-shelf laptops running a version of the 2.4.19 Linux kernel, modified to log various TCP connection level parameters like the congestion window and the slow start threshold. The wireless cards we use are the Cisco Aironet 350 series PCMCIA cards. They have six different power levels of 1, 5, 10, 30, 50 and 100 mW, which give us greater control in forming desired topologies. The open source driver we use [13], provides access to various registers in the card which keep a count of various MAC protocol related events.

The experiments are performed in an indoor environment. The antenna of the wireless cards is covered with copper tape (3M 1181 EMI Copper Foil Shielding type) to reduce their transmission and communication range. This works since copper changes the effective antenna impedance, thus inducing an impedance mismatch with the card circuitry resulting in less power being delivered on the air. We verified that the copper tape does not have any observable effect on the cards other than reducing their range (and causing them to heat up a little faster) by comparing experiments with and without copper tape, over few one hop topologies with different MAC settings. One also has to be careful in interchanging cards between nodes since each card is a little different and changing cards may alter the network topology. Our experiments are performed in a building where extraneous interference from other IEEE 802.11b devices (which are part of a production network and cannot be turned off) exists. Instead of bullet proofing our setup (some screening is provided by the copper tapes) from such interference, we accept it as a fact of life and rely on statistics to filter out such random factors. We however keep a record of the observed interference by running the Kismet wireless sniffer [14]. This sniffer works by putting the card in RF monitor mode, which lets it potentially decode any RF energy put on the air by any IEEE 802.11b device making it much more powerful than tcpdump.

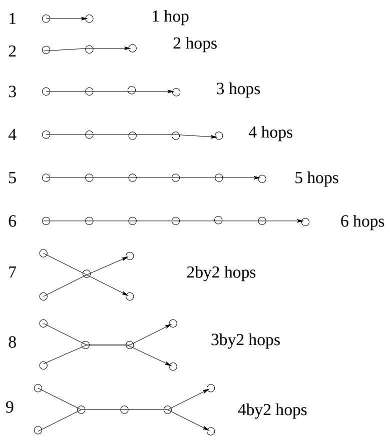

Figure 1: The experimental topologies or plots.

For generating traffic we use the nttcp tool. Logs were collected using tcpdump for TCP traffic, our custom kernel for TCP connection level parameters and IP parameters, and the wireless card driver for various MAC level parameters. The whole process of experimentation, including set up, data generation, logging and archiving, was completely automated using shell scripts and existing tools so that multiple runs of an experiment could be performed overnight without intervention. The data is also made available through a CGI interface which allows searching and limited online processing. We now proceed to describe the experiments and their results.

3.1 Design of experiments

Table 1: Description of treatments.

#	Name	Explanation
0	default	No clamping, sack off, RTS-CTS on
1	all	all=clamp+sack+rts_off
2	clamp	Cwnd clamped to 3*(# of hops/2)
3	rts_off	RTS-CTS mechanism switched off
4	sack	TCP selective acks turned on
5	clamp+rts_off	Treatments 2 and 3 combined
6	clamp+sack	Treatments 2 and 4 combined
7	sack+rts_off	Treatments 3 and 4 combined

In the language of statistics, a *treatment* is a control administered to an experiment. Treatments are represented by values of the independent variable (controlled by the experimenter) also called *factor*, which is typically a categorical variable. The eight treatments (in our case, protocol modifications) we have selected are all possible combinations of the three modifications: congestion window clamping, RTS-CTS handshake off, and TCP selective acknowledgment option on. They are listed in Table 1.

For each of the treatments we measure several *response variables* at different layers in the stack. In the subsequent analysis though, we focus on three main response variables: the end-to-end TCP throughput, the average delay or RTT, and standard deviation of RTT (also called delay jitter). The average and standard deviation of RTT are computed over the number of packets for each flow. The response variables are measured for each treatment on nine different experimental topologies, which are called *plots* in the language of statistics. We also refer to them as scenarios or topologies. The first six plots or scenarios consist of a single TCP flow of one through six hops, and the next three plots consist of two intersecting flows of two, three, and four hops, respectively. These are shown in Figure 1.

For each plot, 15 runs of an experiment are performed for each treatment, and all the response variables are recorded, which totals to 1080 (= 15 × 8 × 9) experiments. Each experiment consists of transferring a huge file over a TCP connection from each source to its destination. The size of the file is different for different plots.

3.2 Summary of measurements

This section presents some summary statistics and observations. The means for all the three response variables are plotted in Figure 2 , and displayed in Table 2. For the topologies with two TCP flows (2by2, 3by2 and 4by2), the mean throughput listed is the total of the throughput for the two TCP flows, whereas the delay and delay jitter values listed are averaged over the two flows.

From a visual inspection of the graphs in Figure 2 and observing the values in Table 2, the following observations are discernible. Treatments 1, 2, 5, and 6 seem to achieve considerably lower values for delay and delay jitter, while treatments 7 and 3 seem to get better throughput, though 1 and 5 are also close. This implies that congestion window clamping results in a significant reduction in delay and delay jitter, while turning RTS-CTS handshake off results in a small but noticeable increase in throughput. However, these variations may result from random experimental effects. To draw reliable conclusions, we need to subject these experimental observations to rigorous statistical analysis.

3.3 Statistical analysis

Before describing the results of the statistical analysis, some definitions [15] are in order. The null hypothesis for a statistical test is the assumption that the test uses for calculating the significance or the p value of a test. The p value (sometimes called just significance) is the probability of observing a test statistic at least as extreme as the value actually observed, assuming that the null hypothesis is true. This probability is then compared to the preselected significance level (also known as the α-level) of the test, which is the probability of (incorrectly) rejecting the null hypothesis when it is in fact true. Usually a small value such as 0.05 is chosen. If the p value is smaller than the significance level, the null hypothesis is rejected, and the test result is termed significant.

We estimate the significance of the experimental results using ANOVA techniques [16, 15]. ANOVA tests the null hypothesis that there is no difference in the means for different levels of the factor. We use SPSS [17] for our computations. ANOVA makes the following assumptions about the experimental data. It assumes that the observations are independent, the distribution of the response variable in each of the groups is normal, and that the groups have homogeneous variances. We perform further tests to check whether our observations satisfy these assumptions. The details for these tests are presented in [12]; a summary follows. Independence was a given since each observation is obtained from a new TCP connection at a different time. We test the normality of our data using a statistical test called the Shapiro-Wilk test [15] for normality, which reports that in a few cases (about one-third) the data is not normal. But given the robustness of ANOVA to non-normality, we conclude that it is an appropriate test for our data. The assumption for homogeneity of variances is tested using Levene's test [15] which reports that the assumption is not satisified with a low enough significance level for some of our data. Thus, modifications to the standard ANOVA F test, like the Welch and the Brown-Forsythe tests, which do not make the assumption of the homogeneity of variances, may be more appropriate for our data. However standard ANOVA is also robust to lack of homogeneity of

Table 2: Means for the response variables.

Treatment #	1hop	2hop	3hop	4hop	5hop	6hop	2by2	3by2	4by2
Mean (over runs) of throughput									
0	4.12	2.03	1.28	1.02	0.78	0.53	1.91	1.78	1.08
1	3.33	2.06	1.45	1.12	0.86	0.58	1.94	1.60	1.12
2	3.20	1.92	1.36	1.02	0.79	0.52	1.78	1.48	1.06
3	4.54	2.19	1.41	1.08	0.84	0.53	2.05	1.84	1.23
4	4.29	2.08	1.34	1.02	0.78	0.53	1.97	1.72	1.10
5	3.33	2.06	1.45	1.11	0.90	0.55	1.90	1.59	1.14
6	3.18	1.92	1.32	1.03	0.80	0.52	1.75	1.47	1.03
7	4.64	2.26	1.45	1.11	0.89	0.57	2.11	1.94	1.21
Mean (over runs) of average RTT									
0	68.9	750.6	1237.1	838.6	180.9	304.1	743.0	569.4	350.4
1	8.6	17.1	49.2	88.6	133.8	183.4	43.4	100.7	154.5
2	9.1	18.5	53.3	97.1	143.1	195.3	48.8	109.3	161.6
3	39.3	687.8	1148.8	919.1	170.5	270.4	702.6	513.3	340.3
4	70.2	655.9	1074.5	863.1	186.0	333.4	790.3	519.5	377.0
5	8.6	17.1	49.4	89.9	132.9	182.6	45.1	100.4	152.5
6	9.1	18.4	55.0	96.8	145.5	194.6	50.3	110.5	169.3
7	40.3	590.3	1077.5	760.0	174.2	291.3	716.6	520.4	342.2
Mean (over runs) of standard deviation of RTT									
0	78.3	205.6	307.4	248.9	69.6	191.2	277.0	274.5	241.2
1	3.9	5.5	16.0	22.9	41.9	77.5	21.4	40.5	60.3
2	3.7	6.0	16.7	25.2	47.1	87.2	23.4	41.4	62.3
3	34.5	182.5	282.1	378.8	71.5	191.9	252.3	245.8	251.0
4	74.4	203.0	286.2	332.2	70.8	246.9	240.8	220.3	241.8
5	3.9	5.4	15.4	23.7	40.3	74.8	22.1	40.7	57.3
6	3.8	5.8	17.8	24.9	43.2	77.8	27.5	41.4	61.5
7	36.3	160.5	249.5	326.0	70.4	189.5	247.2	256.7	245.9

variances if it is balanced; i.e., the sample sizes for the treatments are equal, which is the case for our data. We run all the three tests using SPSS.

Table 3 lists the p values (statistical significance), computed using the three tests mentioned above: the standard ANOVA F test, Welch test, and the Brown-Forsythe test. We find that the null hypothesis is rejected by all the three tests for all the three measured variables for all the topologies, since the p values are less than the α-level of 0.05.

ANOVA tests the null hypothesis but gives no further information on which treatment is better. This information can be obtained by running multiple comparison post hoc tests. There are various tests available which are suitable under various circumstances. Pairwise comparison tests compare the differences between each pair of means, whereas range tests identify subsets of means that do not differ from each other. We want such subsets for all our response variables, so that we can identify the most effective treatments. Scheffe's test, which is both a pairwise as well as range test, seems like the most appropriate choice, since it is somewhat robust to both non-normality of the response, and heterogeneity of variances among groups. The results from Scheffe's tests are shown in the appendix of [12].

Table 3: p values for ANOVA tests.

Plot	Throughput	Avg. RTT	St. Dev. of RTT
(ANOVA F test, Welch test, Brown Forsythe test)			
1hop	0.000,0.000,0.000	0.000,0.000,0.000	0.000,0.000,0.000
2hop	0.000,0.000,0.000	0.000,0.000,0.000	0.000,0.000,0.000
3hop	0.000,0.000,0.000	0.000,0.000,0.000	0.000,0.000,0.000
4hop	0.000,0.000,0.000	0.000,0.000,0.000	0.000,0.000,0.000
5hop	0.000,0.000,0.000	0.000,0.000,0.000	0.000,0.000,0.000
6hop	0.000,0.000,0.000	0.000,0.000,0.000	0.000,0.000,0.000
2by2	0.000,0.000,0.000	0.000,0.000,0.000	0.000,0.000,0.000
3by2	0.000,0.000,0.000	0.000,0.000,0.000	0.000,0.000,0.000
4by2	0.005,0.024,0.006	0.000,0.000,0.000	0.000,0.000,0.000

Table 4: The best homogenous subset for each response variable, obtained from Scheffe's test.

Plot	Throughput	Avg. RTT	St. Dev. of RTT
1hop	{7, 3}	{1, 5}	{2, 6, 1, 5}
2hop	{7}	{5, 1, 6, 2}	{1, 6, 2}
3hop	{7, 1, 5, 3, 2}	{1, 5, 2, 6}	{5, 1, 2, 6}
4hop	{7, 5, 3}	{1, 5, 6, 2}	{1, 5, 6, 2}
5hop	{5, 7, 1}	{5, 1}	{5, 1, 6, 2}
6hop	{1, 7, 5, 4}	{5, 1, 6, 2, 3}	{5, 1, 6, 2, 7, 0, 3}
2by2	{7, 3, 4, 1}	{1, 5, 2, 6}	{1, 5, 2, 6}
3by2	{7, 3}	{5, 1, 2, 6}	{1, 5, 2, 6}
4by2	{All treatments}	{5, 1, 2, 6}	{5, 1, 6, 2}

To interpret the results from Scheffe's multiple comparison test we make another table, Table 4, which lists the best homogeneous subset of treatments for each topology and each response variable. We see that the treatments 1, 2, 5, and 6 appear in the best homogeneous subset for all the topologies for the response variables: RTT average and standard deviation of RTT. In fact, they are far better than the rest, with delays and delay jitters orders of magnitude lower. Thus, we rule out treatments 0, 3, 4, and 7 from possible recommendations. With respect to the throughput, the division into homogeneous subsets is not that clean for all the topologies. In fact, the difference between the means for throughput is not much for different treatments, possibly because we are already operating close to "capacity" for our topologies. The real gains to be reaped are in average delays and delay jitter. However, the subsets containing treatments 1 and 5 are still statistically better in throughput than the subsets containing treatments 2 and 6, for all the topologies. This leaves us with treatments 1 and 5, and our experiments do not demonstrate any significant difference between them. Treatment 1 includes window clamping, RTS-CTS handshake off, and TCP SACK option on. Treatment 5 is the same as treatment 1 except that SACK is turned off. Thus, our experiments show that there is no statistically significant impact of turning on SACK. This may be because SACK does not really kick in since the MAC retries

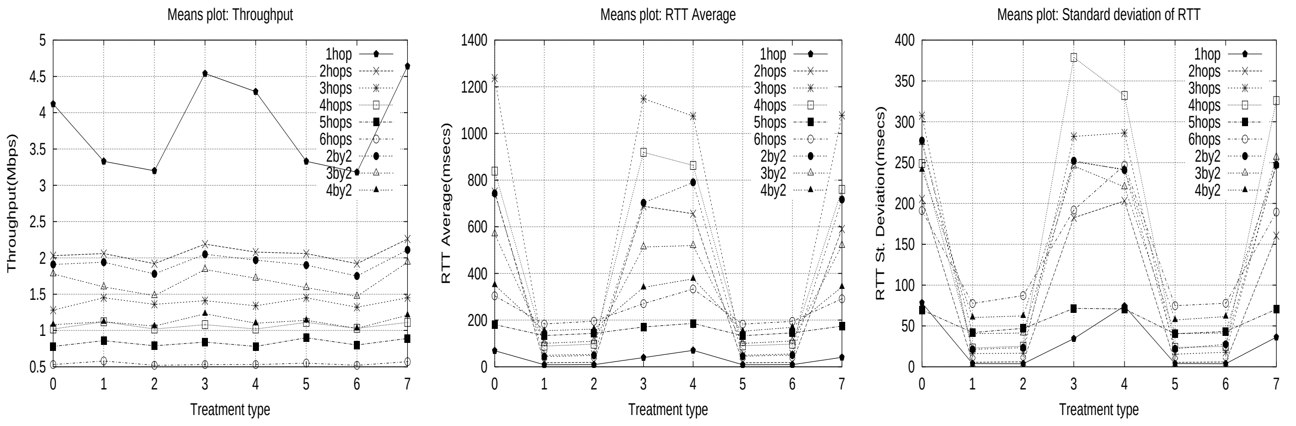

Figure 2: Mean values (over runs) of throughput, average RTT, and standard deviation of RTT.

shield TCP from end-to-end loss. But since turning on SACK does not degrade performance, we recommend turning it on whenever possible.

Thus, our recommendation based on the experimental and statistical analysis is treatment 1. Treatment 1 provides considerable improvements in mean delay and delay jitter, and small but statistically significant improvement in throughput. To conclusively demonstrate this, we display the percent change, with respect to treatment 1, in the means of all the three response variables (displayed in Table 2) when using each of the other treatments. For example, the percent change for throughput for treatment x is calculated as (throughput(x)-throughput(1))*100/throughput(1). These values are shown in Table 5. The entries in bold indicate that the improvement is statistically significant at an $alpha$-level of 0.05, as computed by Dunnett's T3 pairwise multiple comparison test [17]. We note that using other treatments can result in orders of magnitude increase in delays and delay jitter compared to treatment 1, and a small but statistically significant decrease in throughput. Thus, our recommendation is treatment 1. It includes setting the congestion window clamp to $\lceil \frac{3}{2}n \rceil$ where n is the number of hops to be traversed by that flow, turning off the RTS-CTS handshake and turning the TCP selective acknowledgment option on.

4. SOME OTHER FINDINGS

As a part of this study several issues were investigated in detail. We present a summary of some of these studies. Details can be found in [12].

Any packet at the MAC layer is retransmitted at most *retry limit* number of times if an acknowledgement is not received. Our experiments demonstrated that in a generic wireless network setting where a link is shared by multiple nodes, too high a value of *retry limit* is not good, as that causes nodes with bad channel conditions to grab the channel for too long and prevents other nodes with better channel to use it effectively. However if no other node desires to use the link then a very high value of retry limit is good. Thus we perhaps need an adaptive mechanism for setting the *retry limit* based on channel conditions, and perhaps also a smart scheduling policy at the MAC layer. However in the absence of such mechanisms, we determined experimentally that a *retry limit* of 8 is appropriate for a generic ad hoc network in an indoor environment

The effect of a congestion window clamp can be modeled simply through a closed migration process for the case of a linear chain of nodes. The throughput of such a chain of N nodes with traffic flowing from one-end to the other is proportional to W/(W+N-1) where W is the congestion window clamp. For detailed calculations and plots, please refer to [12]. We observe that the throughput quickly saturates as W is increased beyond a small multiple of N. The plots for throughput from real experiments match these analytical plots quite closely. The experiments also show how delays and delay jitters grow as the window size is increased and motivate the idea of congestion window clamping to keep the delays and delay jitters low.

Delays and delay jitters can also be reduced by reducing the network buffer size at every node or by some active queue management scheme so that buffer occupancy is mostly low. Congestion window clamping is a simple way to achieve that effect in ad hoc networks. The size of the buffer is set to 100 packets in our experiments.

Retransmissions at the MAC layer can made more efficient. The contention window, which is the time period after which the MAC attempts a retransmission, is doubled upon an unsuccessful transmission which could be due to the physical carrier being busy, the virtual carrier (NAV) being busy, timeout waiting for a CTS or timeout waiting for an ACK. If RTS-CTS is turned on, then the fourth event is most likely due to a channel error rather than due to contention by other nodes. Hence, if the contention window is not doubled upon an ACK timeout, then the MAC retransmissions can be made more efficient.

5. RELATED WORK

There have been few studies addressing concerns similar to those we have studied. Fu et al. [18] contend that there exists an optimal TCP congestion window and conclude from their primarily simulation based studies that TCP window size actually grows much larger than the optimum, resulting in reduced spatial reuse. They suggest two techniques, link RED and adaptive pacing, to improve TCP throughput. Similar observations about excessive TCP congestion window growth have been made in [19]. They also suggest the intuitive solution of sender congestion window clamping and study its effect through simulation based studies.

6. CONCLUDING REMARKS

We have performed a thorough experimental study of the TCP protocol over static wireless multihop networks. The results have been analyzed through statistical tests, and the recommendations made are demonstrated to improve performance. In the process we have discovered numerous subtleties in experimenting with commercial off-the-shelf equipment. More remains to be done, both in terms of analysis as well as experimentation.

Table 5: Percentage change in response variables for all treatments, with respect to Treatment 1.

Treatment	1hop	2hop	3hop	4hop	5hop	6hop	2by2	3by2	4by2
Throughput									
0	**23.7**	-1.46	**-11.7**	**-8.93**	**-9.30**	**-8.62**	-1.55	**11.2**	-3.57
1	0	0	0	0	0	0	0	0	0
2	**-3.90**	**-6.80**	**-6.21**	**-8.93**	**-8.14**	**-10.3**	**-8.25**	**-7.50**	-5.36
3	**36.3**	**6.31**	-2.76	-3.57	-2.33	**-8.62**	5.67	**15.0**	9.82
4	**28.8**	0.971	**-7.59**	**-8.93**	**-9.30**	-8.62	1.55	7.50	-1.79
5	0	0	0	-0.893	4.65	-5.17	-2.06	-0.625	1.79
6	**-4.50**	**-6.80**	**-8.97**	**-8.04**	**-6.98**	**-10.3**	**-9.79**	**-8.13**	-8.04
7	**39.3**	**9.71**	0	-0.893	3.49	-1.72	**8.76**	**21.2**	8.04
Average RTT									
0	**701**	**4.29e+03**	**2.41e+03**	**847**	**35.2**	**65.8**	**1.61e+03**	**465**	**127**
1	0	0	0	0	0	0	0	0	0
2	**5.81**	**8.19**	**8.33**	**9.59**	**6.95**	**6.49**	**12.4**	**8.54**	4.6
3	**357**	**3.92e+03**	**2.23e+03**	**937**	**27.4**	**47.4**	**1.52e+03**	**410**	**120**
4	**716**	**3.74e+03**	**2.08e+03**	**874**	**39**	**81.8**	**1.72e+03**	**416**	**144**
5	0	0	0.407	1.47	-0.673	-0.436	3.92	-0.298	-1.29
6	**5.81**	**7.6**	**11.8**	**9.26**	**8.74**	**6.11**	**15.9**	**9.73**	9.58
7	**369**	**3.35e+03**	**2.09e+03**	**758**	**30.2**	**58.8**	**1.55e+03**	**417**	**121**
Standard deviation of RTT									
0	**1.91e+03**	**3.64e+03**	**1.82e+03**	**987**	**66.1**	**147**	**1.19e+03**	**578**	**300**
1	0	0	0	0	0	0	0	0	0
2	-5.13	9.09	4.37	10	12.4	12.5	9.35	2.22	3.32
3	**785**	**3.22e+03**	**1.66e+03**	**1.55e+03**	**70.6**	**148**	**1.08e+03**	**507**	**316**
4	**1.81e+03**	**3.59e+03**	**1.69e+03**	**1.35e+03**	**69**	219	**1.03e+03**	**444**	**301**
5	0	-1.82	-3.75	3.49	-3.82	-3.48	3.27	0.494	-4.98
6	-2.56	5.45	11.3	**8.73**	3.1	0.387	28.5	2.22	1.99
7	**831**	**2.82e+03**	**1.46e+03**	**1.32e+03**	**68**	**145**	**1.06e+03**	**534**	**308**

7. REFERENCES

[1] R. E. Ludwig, “Eliminating inefficient cross-layer interactions in wireless networking,” Ph.D. dissertation, Aachen University of Technology, Germany, 2000.

[2] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, “A comparison of mechanisms for improving TCP performance over wireless links,” *IEEE/ACM Transactions in Networking*, vol. 5, no. 6, pp. 756–769, Dec 1997.

[3] V. Bharghavan, A. J. Demers, S. Shenker, and L. Zhang, “MACAW: A media access protocol for wireless LANs,” in *SIGCOMM*, 1994, pp. 212–225.

[4] G. Holland and N. Vaidya, “Analysis of TCP performance over mobile ad hoc networks,” in *Proceedings of ACM MOBICOM*, Aug. 1999, pp. 275–288.

[5] ——, “Impact of routing and link layers on TCP performance in mobile ad hoc networks,” in *IEEE Wireless Communications and Networking Conference*, 1999, pp. 1323–1327.

[6] V. Anantharaman, S.-J. Park, K. Sundaresan, and R. Sivakumar, “TCP performance over mobile ad-hoc networks: A quantitative study,” *Wireless Communications and Mobile Computing Journal, Special Issue on Performance Evaluation of Wireless Network*, 2003.

[7] J. Liu and S. Singh, “ATCP: TCP for mobile ad hoc networks,” *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 7, pp. 1300–1315, July 2001.

[8] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, “A feedback-based scheme for improving TCP performance in ad hoc wireless networks,” *IEEE Personal Communications Magazine*, vol. 8, no. 1, pp. 34–39, Feb. 2001.

[9] D. Kim, C.-K. Toh, and Y. Choi, “TCP-BuS: Improving TCP performance in wireless ad hoc networks,” in *IEEE International Conference on Communications*, June 2000, pp. 1707–1713.

[10] Z. Fu, B. Greenstein, X. Meng, and S. Lu, “Design and implementation of a TCP-friendly transport protocol for ad hoc wireless networks,” in *IEEE International Conference on Network Protocols’02*, 2002.

[11] F. Wang and Y. Zhang, “Improving TCP performance over mobile ad-hoc networks with out-of-order detection and response,” in *Proceedings of the Third ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2002, pp. 217–225.

[12] V. Kawadia, “Protocols and architecture for wireless ad hoc networks,” Ph.D. dissertation, University of Illinois at Urbana-Champaign, Sep 2004.

[13] “Cisco/Aironet driver for Linux (4500/4800/340/350 series),” http://airo-linux.sf.net/.

[14] M. Kershaw, “Kismet,” 2003, www.kismetwireless.net.

[15] “PROPHET statguide: One-way analysis of variance (ANOVA), BBN technologies,” 1996, www.basic.nwu.edu/statguidefiles/oneway_anova.html.

[16] R. S. Bogartz, *An Introduction to the Analysis of Variance*. Westport, Connecticut: Praeger, 1994.

[17] “SPSS 12.0: Base system reference manual,” 2004, www.spss.com.

[18] 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 *Proceedings of IEEE INFOCOM*, 2003, pp. 1744–1753.

[19] K. Chen, Y. Xue, S. H. Shah, and K. Nahrstedt, “Understanding bandwidth-delay product in mobile ad hoc networks,” *Elsevier Computer Commnications Journal*, vol. 27, no. 10, pp. 923–934, 2004.