Performance of TCP/UDP under Ad Hoc IEEE802.11

Milenko Petrovic
Dept. of ECE
University of Toronto
Toronto ON. Canada

Mokhtar Aboelaze
Dept. of Computer Science
York University
Toronto ON. Canada

Abstract

TCP is the De facto standard for connection oriented transport layer protocol, while UDP is the De facto standard for transport layer protocol, which is used with real time traffic for audio and video. Although there have been many attempts to measure and analyze the performance of the TCP protocol in wireless networks, very few research was done on the UDP or the interaction between TCP and UDP traffic over the wireless link. In this paper, we tudy the performance of TCP and UDP over IEEE802.11 ad hoc network. We used two topologies, a string and a mesh topology. Our work indicates that IEEE802.11 as a ad-hoc network is not very suitable for bulk transfer using TCP. It also indicates that it is much better for real-time audio. Although one has to be careful here since real-time audio does require much less bandwidth than the wireless link bandwidth. Careful and detailed studies are needed to further clarify that issue.

1 Introduction

There is a huge growth in the wireless communication industry as can be shown by the huge increase in the number of cellular phones, wireless LAN's and the personal digital assistants. The convenience that portable computers bring will tend to displace desktop computers. The same can be said about wireless phones and in the future, smart appliances, which will become commonplace. Wireless phone popularity is mainly due to freedom of movement, that comes from the ability to use wireless phones from virtually anywhere. Voice is the first, and still the major driving force behind wireless technology, but the trend is to provide more services to the user including connection to the Internet either through the wireless phone or some other wireless device.

This research is supported by NSERC under Grant 504000. This paper was prepeared when the first author was at York University

0-7803-7661-7/03/$17.00 ©2003 IEEE

TCP, transfer control protocol, is the standard protocol for reliable delivery of data over the Internet. TCP relies on IP, Internet Protocol, for routing and data transmission. IP provides best-effort service, which is intrinsically unreliable. This makes Internet Protocol very simple, which is one of the reason for its popularity and the rapid growth of the Internet. IP is the de facto standard protocol for inter-networking. TCP is designed to go hand in hand with IP protocol, which resulted in it becoming the dominant reliable transport protocol. There has been a lot of research on how to make TCP work well in a wireline network [11, 12, 7]

Wireless communication is usually done in one of 2 different ways, cellular communication or ad-hoc communication. In cellular communication, pre-established base stations are distributed to cover the are. Each base station is responsible of managing the mobile users in each cell. Mobile users communicate via the base station in the cell they are in.

The other alternative is known as ad-hoc networks. In ad-hoc networks, there is no fixed infrastructure such as base stations or predefined geographical cells. Mobile users are roaming in a specified area, and they communicate by sending (receiving) messages to (from) each other. If two users are close enough to each other they can communicate directly. If the users are far apart then the rest of the users can forward packets to and from these two users in order to be able to communicate. That means every mobile user serves as a relay or a router in order for all the nodes to be able to communicate. Several routing protocol were proposed for ad hoc networks [13, 16, 19, 20, 17, 18]

Wireless medium is a difficult medium for communication. In free space, a typical wireless channel is susceptible to the problems of path loss, shadowing, multipath fading and interference. Usually the bit rate error for wireless channels is higher than wireleine channels, and its bandwidth is less. that makes using the wireless link with a protocol that was specifically designed for a wireline networks a bit challenging.

There has been a lot of research trying to measure and analyze the performance of TCP over wireless links for both cellular, and ad-hoc networks. However, for applications like audio or video, usually that will be carried out using UDP instead of TCP. Very few studies were carried out on the performance of UDP on wireless links, or in the interaction between TCP and UDP traffic over wireless links. In this paper, we present simulation results of the interaction between TCP and UDP traffic (both real-time and bulk) over ad-hoc networks using IEEE802.11 as a wireless link [10].

The organization of the paper is as follows, In section 2 we review some of the previous attempts in measuring TCP performance over wirless networks. In section 3, we present our network setup and the error model we will use throughout experiement. Section 4 presents some results on a string of wirless nodes, while section 5 presents results for a mesh topology. Section 6 is a conclusion.

2 Previous Work

There is a large volume of literature on the performance of TCP in wireless environment. Research on improving the performance of TCP over wireless networks can be classified into two categories. improvements at the link layer and improvements by making modifications to TCP. We will very briefly mention some of the previous work and classify it.

Snoop protocol [2] is designed to be TCP aware, and to mask unreliability of wireless layer. Snoop is implemented as a layer in TCP/IP architecture stack. It is located just below TCP layer. Snoop can be located at both the access point and the mobile nodes. It is not necessary to use it at mobile nodes, which makes it easier to implement, but transfer of data from mobile host to wired node will not benefit from snoop. Snoop at the access point is only able to improve TCP performance of connections from wired host to mobile hosts.

Explicit Feedback (EF) [1] is a mechanism used by the access point to inform TCP sender (located in the wired network) that wireless channel is currently experiencing a lot of errors and that it should not invoke congestion avoidance procedure on lost segment timeouts. This requires modifications at both the access point and the TCP sender. The explicit feedback messages are sent to the sender after every failed transmission to a mobile node from the access point. In [5] access point is assumed to send acknowledgments to senders on the wired network for every segment it receives. These acknowledgments indicate to the TCP sender the segment reached the access point and if it does not receive the acknowledgment for it, then the sender can assume that the loss occurred due to corruption over wireless medium, and congestion avoidance should not be initiated.

The last hop acknowledgment scheme assumes that losses over wireless network happen only due to corruption and that wireless network is the last hop on the TCP segment path (which is the case for cellular networks). The acknowledgment from the access point is called last hop ACK (LHACK). In the case that TCP sender does not receive LHACK, then congestion in the wired network caused packet to be dropped and therefore TCP sender should start congestion avoidance procedure.

In [3] , TCP segment inter-arrival times at TCP receiver are used to distinguish between congestion and wireless losses. It is assumed that TCP segments will queue at the access point in the case when TCP receiver is on a wireless node. Queuing occurs here because of small wireless bandwidth as compared to wired bandwidth. TCP receiver looks at inter-arrival time between every segment. If the inter-arrival time between two segments is a multiple of a segment transmission time over wireless network, but the two segments arrived out-of-order, then TCP receiver assumes that all segments between last in-order received segment and the segment just received are lost due to congestion in the wired network. This scheme assumes that due to queuing at the access point, all segments will be sent back-to-back to the wireless node. It also assumes that there is no congestion in the wireless link and that only bulk transfers are used. In the case that segments are lost because of congestion, the queue at the access point will have gaps in sequence numbers, but inter-arrival times at the mobile node will be the same for all packets. From these gaps, TCP receiver can conclude that congestion is the cause of the losses. On the other hand, if losses occurred in wireless part then the inter-arrival times will not be a multiple of segment transmission times. From this, TCP receiver can conclude that the losses occurred because of wireless error and it does not initiate congestion avoidance.

Mobile-TCP [22] is another solution that is designed mostly for problems of disconnections. Mobile-TCP informs TCP-sender (on wired network) that a disconnection occurred. If TCP sender detects a loss (duplicate acknowledgments or timeout) it will perform retransmissions but without reducing its send window. Once disconnection ends, TCP sender is informed to resume normal operation.

3 Experiment Setup

In this paper, a series of simulations is performed to determine the perofrmance and the interaction between TCP carrying bulk traffic and UDP carrying real-time audio traffic in wireless links. We used the ns-2 simulator [4] with the wireless extension form the Monarch project at CMU [15]. The main perofrmance criteria for bulk transfer is the throughput. While the main performance criteria for real-time audio is cell loss ration. These sets of simulations are similar. The cell loss in UDP traffic is mainly due to two different factors. A cell (frame) is lost if it will be tranmsitted up to the maximum number of times and always is delivered in error. Or if the cell is delayed due to queueing or multiple transmission up to the maximum delay limit, in this case it is not useful anymore and will not be transmitted and is dropped by the sender. to those in [8], although they only used bulk transfer with TCP without FEC. In all the setups DSDV is used as the routing protocol.

The model of errors in a wireless channel is Gilbert-Elliot [9, 6], which captures bursty nature of errors in radio channels. It is a time-based two state Markov chain, where a "good" state has a low error probability (10^{-6}) and a "bad" state has a high error probability (10^{-2}) (same as in [14]). The average length of the states is exponentially distributed with mean duration for "good" state of 0.1 second and 0.0333 seconds for the "bad" state. In two-sate Markov chain, "good" state is always followed by a "bad" state and vice versa. Each node uses the error model independently, which means that each nodes sees the radio channel differently. The original error model for wireless channel in ns2 has been modified to correct an error in its operation. The model failed to make any state transitions when the channel was idle regardless of the passage of time. The consequence of this is that states lasted for very long time.

In order to deal with errors, we used Reed-Solomon FEC in order to detect and errors when the channel is in the bad state. The choice of the code was such that the channel will have the same BER in the good state and the bad setate with FEC. That results in decresing the efficiency of the TCP by 40% due to the overhead of the FEC. Thus we eliminated the bad state on the expense of a reduced bandwidth.

We used two different topologies, first we used a linear string of 8 nodes where every node can communicate with its two neighbors only (one neighbor in case of the end nodes). Then we used a mesh topology where every node can communicate with its four neighbors in the row and column directions. For a complete results description, the reader is referred to [?].

4 String

4.1 Single bulk TCP transfer

Using a string topology we examine the performance of a multi-hop TCP connection. In this configuration, every node is only able to communicate with its immediate neighbor, so routing is needed to reach nodes that are not within transmission range. The source node initiates a bulk TCP transfer to one of the other nodes. The measure of performance is throughput. All nodes are assumed to be stationary so routing has no effect on the throughput, thus TCP performance depends mainly on MAC protocol performance. We look at TCP throughput for connections between nodes 0-1, 0-2, 0-3, 0-4, 0-5 and 0-6. we ran the simulation using FEC, without FEC, and without any errors (ideal channel) for comparison.

In figures 1 we notice that a larger segment size produces better reults than a smaller one. Also, RTS/CTS is almost having a negative impact on the peformance, and the system performs better without collision avoidance.

In figures 2 and 3 it can be seen that as number of hops increases, the use of larger window size results in increase in throughput. With large window size, TCP can have more segments to transmit at each node without waiting to receive the ACK for the transmitted segments (many will be lost and the TCP will perform slow start). It is therefore recommended to allow TCP to use larger window size at all times. Furthermore in figure 1, we can see that with errors, plain TCP connections are barely able to transfer packets more then 4 hops away. With increased window and large packet size as in figures 3(c) and 3(d), plain TCP is able to complete transfers, but at a very low throughput.

4.2 Audio

Here, we consider combination of real-time audio and bulk transfer. The objective is to investigate the interaction between real-time audio using UDP, and bulk transfer using TCP. First, we consider audio only, we simulate and measure the loss percentage for a call from node 0 to the seven other nodes. Table 1 shows the loss percentage for a call from node 0 to nodes 1..7. As

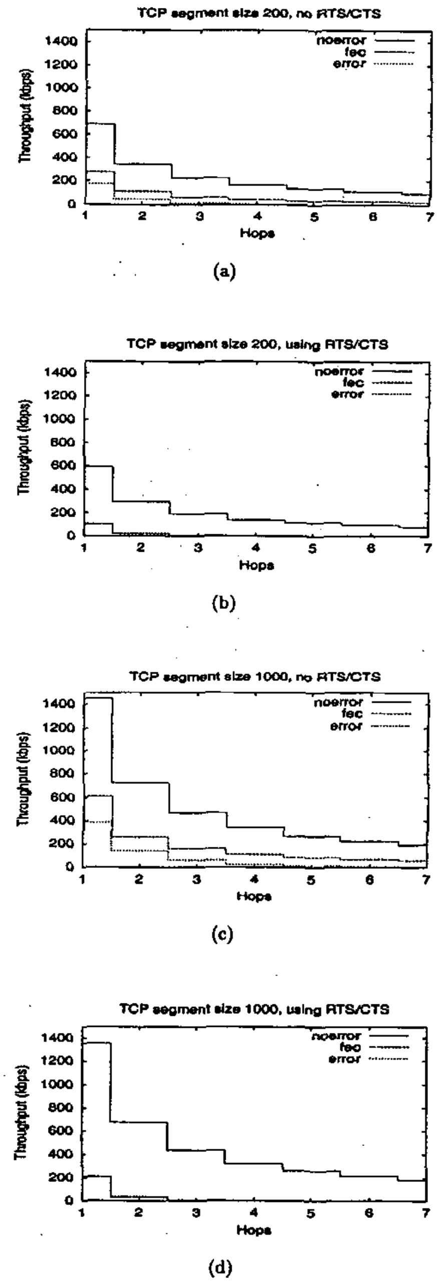
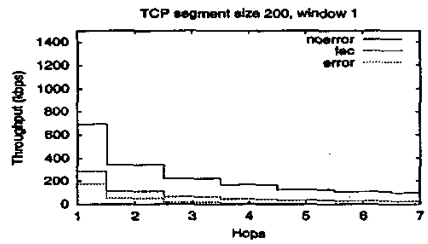
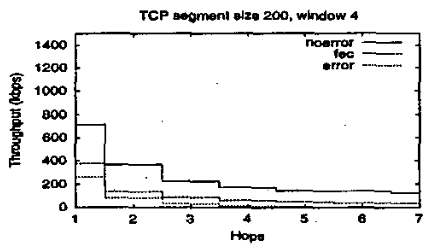


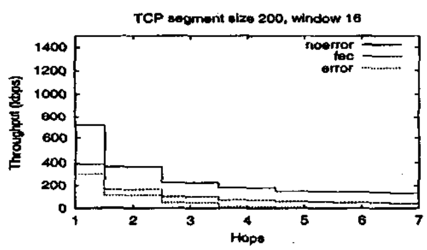
Figure 1: String: TCP window size = 1



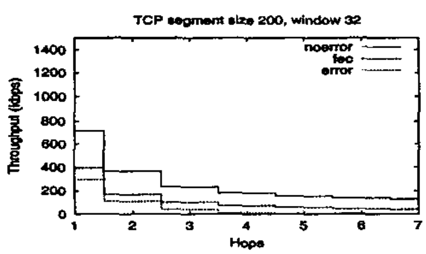
(a)



(b)

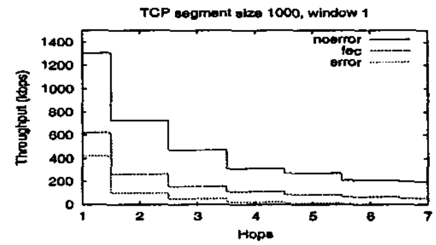


(c)

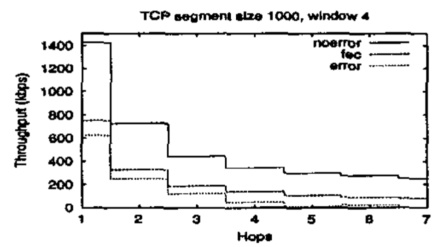


(d)

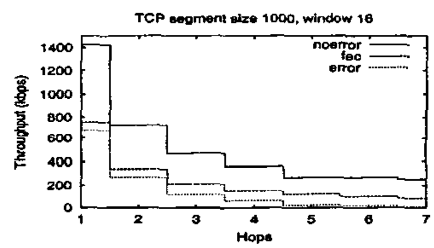
Figure 2: String: TCP segment size = 200 bytes



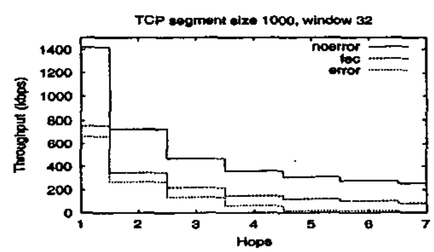
(a)



(b)



(c)



(d)

Figure 3: String: TCP segment size = 1000 bytes

	UDP Packet Size		
From node 0 to	600 bytes	400 bytes	300 bytes
1	0.25%	0.3%	0.4%
2	0.31%	0.85%	0.75%
3	1.5%	1.3%	1.2%
4	2.05%	1.4%	1.7%
5	2.1%	2.0%	1.7%
6	2.6%	3.5%	4 %
7	6.0%	6.0%	4.0%

Table 1: loss ratio for a single real-time audio call from node 0 to the rest of the nodes

UDP Packet Size				
100	200	300	400	600
19.7%	2.27%	3.23%	3.54%	3.13%

Table 2: Average loss ratio for 4 overlapping audio connections

expected the loss ratio increases with the number of hops (even for a single call), and more than few hops results in increasing the loss rate beyond the generally accepted 1-2%. We also notice that smaller UDP packet size leads to a better loss rate. We believe that although a smaller UDP packet means more packets, however it also means that by not waiting to collect a large packets, we can tranmsit cells with minimum delay thus reducing the probability of time out and discarding the cell in case of multiple retramsissions.

Next, we run two experiments with multiple voice connections. The first experiment, we run two way audio connection between nodes 0-7, 1-6, 2-5, and 3-4. In the second, we run 0-1, 2-3. 4-5, and 6-7. The first configuration produces the maximum overlaps between these four connections, while the later produce the minimum overlap. Tables

Table 2 shows the loss ratio for the first configuration,

UDP Packet Size				
100	200	300	400	600
0.36%	0.42%	0.36%	0.33%	0.47%

Table 3: Average loss ratio for 4 non-overlapping audio connections

	Window Size in packets		
Packet Size	1	2	4
500	(32, 5.0%)	(37, 4.6 %)	(34, 14%)
1K	(46, 5.0%	(54, 5.0%)	(66, 6.6%)
2K	(4.8, 5.0%	(8.3, 8.5 %)	(7, 4.5%)
4K	(0.0, 4.0%)	(0.23, 5.5%)	(0.0, 4.0%)

Table 4: Average throughput in Kbps and loss ratio for one audio and one bulk connection between nodes 0 and 7

while Table 3 shows the loss ratio for the second configuration. We notice that for multiple audio connections a UDP packet size of 300 bytes produces the best results.

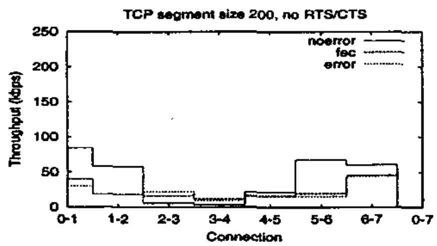
Table 4 shows both the throughput of one TCP connection and one audio connection between nodes 0 and 7. We notice that with a large packet size the TCP throughput is 0. We also notice that although a larger window size increases the TCP throughput, it also increases the loss ratio for UDP packets.

4.3 Multiple concurrent bulk TCP transfers

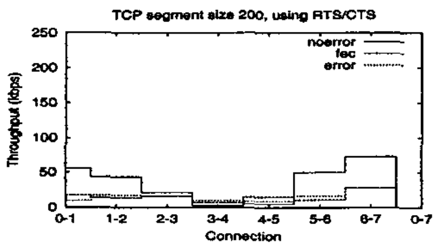
In a modified string experiment there is a connection between every two neighboring nodes in a string. In addition there is a connection between the last and the first node that spans all nodes in the string. This topology is used to investigate fairness between single and multihop transfers and how hidden terminal problem affects them.

Performance of one-hop TCP connections is much better when compared to the one multi-hop connection (figure 4). This setup shows how a multihop connection fails in the presence of many single hop transfers. Furthermore, since all nodes are active at the same time, the performance of single hop connections is also affected, since a single node cannot communicate with two different nodes simultaneously.

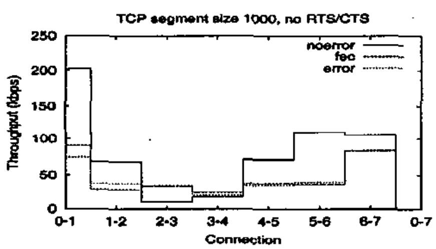
In the figures 4(a), 4(c), where RTS/CTS is not used, it can be shown that large packet sizes result in larger total throughput. It is interesting to note that FEC at 60% efficiency is more fair to middle connections (2-3 and 3-4) when compared to a situation either with or



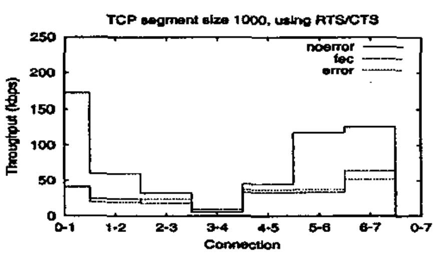
(a)



(b)



(c)



(d)

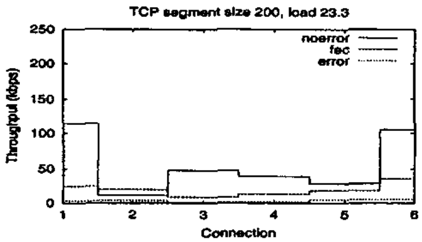
Figure 4: String with concurrent TCP transfers

without errors. Because of errors, a single connection is never able to fully capture the channel as is the case without errors. This results in reduced overall throughput, and better fairness. Use of RTS/CTS (figures 4(b), and 4(d)) reduces throughput and does not improve fairness. In all cases, the single multihop connection is able to transfer data, but at a very low rate (about 1 kbps).

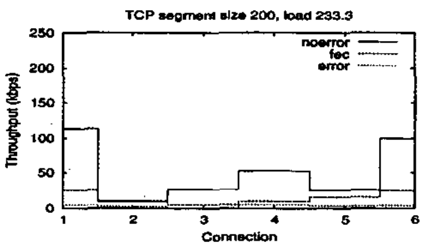
5 Mesh

Mesh topology is an example of a more realistic topology than a string from the previous section. Every node in a mesh is connected to either two (corners), three (sides) or four (inner) other nodes. Mesh topology allows us to see how TCP performs in a more realistic ad environment. There are two types of traffic passing through the mesh. Along all the vertical paths are bulk TCP connections. For example in a 6x6 mesh, the nodes are numbered in a row major fashion, with the bottom row numbered 0,1,2,..5, and the top row numbered 30-35. We established 6 TCP connections, with source nodes 30 to 35 and destination nodes 0 to 5 respectively (6 connections from the top row to the bottom row). They are numbered 1 to 6 respectively. Along the horizontal paths are constant bit connections. In a 6x6 mesh, the connections originate in nodes 0, 6, 12, 18, 24, 30 and terminate in 5, 11, 17, 23, 29, 35. (these are 6 horizontal connections between nodes in the left-most column and the corrersponding nodes in the righ-most column). Constant bit sources are similar to bulk sources in that they too have unlimited supply of packets. The difference is that constant bit (CBR) sources send packets at regular intervals. CBR sources use TCP as transport protocol. Packet size is fixed in all the simulations to 1000 bytes. The rate at which the CBR sources generate packets is one of 23.3kbps or 233.3kbps. CBR sources represent interference traffic, by introducing constant load on the network. With 23.3kbps sources, a CBR source sends one TCP segment every 1.5 seconds, and with 233.3kbps it is 0.05 seconds. Therefore network load increases with higher CBR source rate.

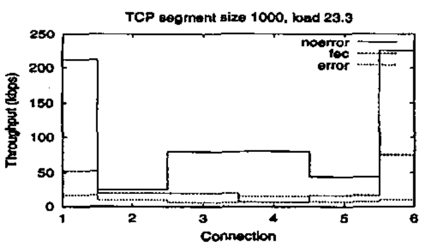
Figure 5 shows results for mesh of size 6x6. Here we can see that in the presence of errors and with-



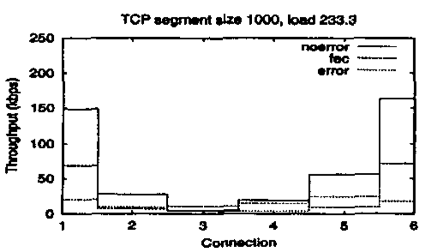
(a)



(b)



(c)



(d)

Figure 5: Ad hoc mesh

out FEC, TCP is barely able to transfer any packets, with throughput close to zero. This is because with 6 hops, probability that a packet reaches destination is very small. Here use of FEC helps considerably in that is allows all connection to transfer data. Again larger packet sizes show better throughput. As load increases, without FEC, some middle TCP connections fail completely for small packet sizes. With FEC, throughput of bulk connections is reasonable in the sense that it is about 20% to 50% on average of the throughout without any errors. In some cases, throughput of bulk connections with FEC is even better than their throughout would without any errors. This is because wireless errors also affect the throughput of interfering traffic and lower it considerably, so that bulk TCP connections experience less interference from CBR traffic.

A similar simulation in [8] differs in that static routing is used, wheres here DSDV is used. The consequence of this is that it is possible for DSDV to incorrectly determine that a node is unreachable due to wireless errors and it re-routes packets using a different route. It is therefore possible that the bulk and CBR traffic do not flow always along horizontal or vertical direction only.

6 Summary and Conclusions

In this paper we investigated the perofrmance of bulk traffic using TCP and real-time audio traffic using UDP over an ad-hoc networ using IEEE802.11. Our results indicates that 802.11 is suitable only for small ad-hoc networks with number of hops 2-3. A bigger network results in a much degraded perofrmance for both TCP and UDP.

References

[1] B. S. Bakshi, P. Krishna, N. H. Vaidya, and D. K. Pradhan, "Improving performance of TCP over wireless networks," in *International Conference on Distributed Computing Systems*, 1997.

[2] H. Balakrishnan, S. Seshan, and R. H. Katz, "Improving reliable transport and handoff performance in cellular wireless networks," *ACM Wireless Networks*, vol. 1, no. 4, 1995.

[3] S. Biaz and N. Vaidya, "Discriminating congestion losses from wireless losses using inter-arrival times at the receiver," in *IEEE Symposium ASSET'99*, 1999.

[4] L. Breslau, et al. "Advances in network simulation," in *IEEE Computer* 33(5) pp. 59-67, May 2000.

[5] J. A. Cobb and P. Agrawal, "Congestion or corruption? a strategy for efficient wireless TCP sessions," in *IEEE Symposium on Computers and Communications*, pp. 262–268, 1995.

[6] E. O. Elliot, "Estimates of error rates for codes on burst-noise channel," *Bell System Technical Journal*, vol. 42, pp. 1977–1997, September 1963.

[7] S. Floyd, "A report on recent developments in TCP congestion control," *IEEE Communications Magazine*, vol. 39, pp. 84–90, April 2001.

[8] M. Gerla, R. Bagrodia, L. Zhang, K. Tang, and L. Wang, "TCP over wireless multihop protocols: Simulation and experiments," in *Proceedings of IEEE ICC'99*, 1999.

[9] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell System Technical Journal*, vol. 39, pp. 1253–1265, September 1960.

[10] *IEEE Std 802.11-1997 Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* 1997.

[11] V. Jacobson, "Congestion avoidance and control," in *ACM SIGCOMM'88*, (Stanford), pp. 314–329, August 1988.

[12] V. Jacobson, "Modified TCP congestion avoidance algorithm," *end2end-interest mailing list*, April 1990.

[13] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing* (Imielinski and Korth, eds.), ch. 5, pp. 153-181, Kluwer Academic Publishers, 1996.

[14] J. G. Kim and M. M. Krunz, "Bandwidth allocation in wireless networks with guaranteed packet-loss performance," *IEEE/ACM Transactions on Networking*, vol. 8, no. 3, pp. 337–349, 2000.

[15] CMU Monarch Project, "The CMU monarch project's wireless and mobility extension to ns." available from www.monarch.cs.cmu.edu 1999

[16] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *IEEE INFOCOM*, pp. 1405–1413, 1997.

[17] G. Pei, M. Gerla, and T.-W. Chen, "Fisheye state routing: A routing scheme for ad hoc wireless networks," in *ICC*, pp. 70–74, 2000.

[18] G. Pei, M. Gerla, and X. Hong, "Lanmar: Landmark routing for large scale wireless ad hoc networks with group mobility," in *IEEE/ACM MobiHOC'00*, pp. 11–18, August 2000.

[19] C. Perkins, "Ad-hoc on-demand distance vector routing," in *MILCOM'97*, November 1997.

[20] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, pp. 234–244, 1994.

[21] M. Petrovic, "Improving TCP/IP performance over IEEE802.11 wireles LAN's," M.Sc. Thesis, Dept. of Computer Science, York University, Toronto Canada, 2002

[22] M. Stangel and V. Bharghavan, "Improving TCP performance in mobile computing environments," in *International Conference on Communications '98*, 1998.