TCP Behavior across Multihop Wireless Networks and the Wired Internet*

Kaixin Xu, Sang Bae, Sungwook Lee, Mario Gerla
Computer Science Department, UCLA
405 Hilgard Ave., Los Angeles, CA
{xkx, sbae, swlee, gerla}@cs.ucla.edu

ABSTRACT

Emerging wireless ad hoc networks find their most important applications in untethered, mobile, multihop scenarios where there is no wired infrastructure. Yet, when the wired infrastructure (say, the Internet) is within reach, opportunistic connections to Internet sites may be established across the multihop network to transfer files and update databases. These file transfers use TCP for reliability and congestion control. However, recent experiments with ad hoc, multihop 802.11 networks have exposed serious instabilities when TCP connections span both wired and wireless domains. In particular, some TCP connections capture the wireless channel and drive the throughput on other connections virtually to zero. This is most surprising in view of the fact that connections between 802.11 (single hop) wireless LAN stations and the Internet are well behaved. In fact they are routinely used in most Campuses, Businesses and Research Labs. This paper is an experimental study of the unstable behavior of TCP across 802.11 ad hoc networks and the wired Internet. We investigate the fairness issues of multiple TCP flows as well as the coexistence of TCP flows and video streams in the wired/wireless scenario. Detailed analysis of the measurement results is also presented. The paper will prove very valuable to future commercial and military ad hoc networks.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General – *data communications.*

General Terms

Measurement, Performance.

Keywords

TCP Performance, Fairness, Ad Hoc Network, MANET.

1. INTRODUCTION

As wireless multihop networks emerge, it will become necessary to communicate across multihop networks to servers back in the wired network. In the battlefield, for example, image files containing target profiles are downloaded from databases to mobiles in the tactical ad hoc network. At the same time, images of potential targets collected by mobiles may be sent back to processing centers in the wired Internet. In ad hoc collaborative net set up among members of a search and rescue team, one of the team members may occasionally connect to the Internet (via satellite, say) to download files from a remote server and share them with his colleagues. There may also be a need to upload files from mobiles to the Internet, for example pictures of objects requiring further processing.

Another opportunity for ad hoc (multihop) and wired segment interconnection arises in wireless LANs when multihopping is invoked to overcome and bypass a LAN Access Point (AP) fault. Namely, if the AP fails, and the mobiles in the affected area cannot connect to neighbor APs directly, they may try to reach the Internet indirectly by multihopping through mobile neighbors in adjacent areas.

In all the above cases there will be file transfers between wired and ad hoc wireless hosts, all running over TCP. Thus, it is vital thus to assure that TCP perform efficiently over mixed wired and multihop segments.

TCP performance over wired connection is well understood. Recently, good progress has also been made on TCP over paths that include one or more wireless links operated in a point to point mode (eg, satellite links, or last hop wireless LANs). These wireless links often introduce random packet errors and loss. Care must be taken to correctly handle such loss and distinguish it from congestion loss (the latter requires the intervention of congestion control mechanisms such as TCP window reduction). Recent extensions of conventional TCP (eg, TCP Snoop [16], TCP Peach [18], TCP Westwood [17], etc) can deal with such wireless loss situations.

Much more challenging is the problem of achieving good TCP performance within an ad hoc, multihop network. This has been an area of active research recently, and progress has been made in several directions. Three different types of challenges are posed to TCP design by such networks. First, as the topology changes, the path is interrupted and TCP goes into repeated, exponentially increasing time-out with severe performance impact. Efficient retransmission strategies have been proposed to overcome such problems [13][14][15]. The second problem has to do with the fact that TCP performance in ad hoc multihop environment

---

* This work is supported in part by ONR "MINUTEMAN" project under contract N00014-01-C-0016 and TRW under a Graduate Student Fellowship.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WoWMoM'02, September 28, 2002, Atlanta, Georgia, USA.

Copyright 2002 ACM 1-58113-474-6/02/0009…$5.00

depends critically on the window in use. If the window grows too large, there are too many packets (and ACKs) on the path, all competing for the same medium. Congestion builds up and causes "wastage" of the broadcast medium and severe throughput degradation [3]. This is different from wired networks, which are fairly tolerant of large windows. The third problem is due to the interaction of the 802.11 MAC layer protocol, more precisely, the hidden terminal problem and binary backoff scheme etc., with the TCP window mechanism and time out. This TCP/802.11 interaction was found to cause unfairness among competing TCP flows and (in extreme cases) "capture" of the channel by a few flows. Solutions to the second and third problems have been recently proposed [4][5].

Moving now to the wired/wireless ad hoc environment, we note that yet new challenges arise. The situation here is much more complex than that in the first scenario – ie, wired path with one or more wireless links introducing random loss. The wireless multihop section not only drops packets; it can also trap them. This invalidates the models on which some of the TCP variants (eg, TCP Westwood [17] and TCP Peach [18]) are based.

The presence of a wired section on the path also changes the terms of the problem with respect to an exclusively ad hoc environment. For example, in a pure ad hoc environment, analytic considerations and experimental results show that TCP operates best with small window sizes. Under proper assumptions, one can show that optimal window size is min (1, 1/3 path length). In most practical situations this translates to W = 1 or 2 [3][5]. The use of a larger window worsens performance and aggravates the capture problems. With the wired path extension, W = 1 is untenable. Assuming that the wireless multihop network is the bottleneck (as it is typically the case) and the maximum bandwidth achievable through it is B, the TCP window required to attain it is $W > B * RTT$, where RTT is the round trip delay of the connection. For example, if B = 500 Kbps, RTT is 200ms and packet size is 1.5 KB, the minimal window is W = 10 packets. Operating with W =1 in this situation would lead to drastic reduction in throughput. The use of a larger window, on the other hand, may lead to the accumulation of packets in the multihop section of the network, with adverse effects as well. This dilemma is at the core of the wired/multihop TCP design.

The above considerations indicate that, on the one hand, efficient TCP operation from ad hoc wireless networks to the wired Internet is critically important in many upcoming applications; and, on the other hand, TCP may behave quite differently from previously studied wireless scenarios. Thus, further research is needed in this area.

In this paper, we take a first step in this direction. In the following sections we first describe the environment at hand (Sect 2). Then, in Sect 3 we present experimental 802.11 testbed results on performance of multiple TCP flows in terms of fair sharing the wireless channel. The results expose the unique problems posed by the wired/wireless environment. In Sect 4 we further investigate how video streams and TCP flows perform when they are coexisting in the wired/wireless scenario. In Sect 5, we investigate the causes of the abnormal behavior and trace the problem to the interaction of TCP and the 802.11 MAC layer. Related work is reported in Sec 6. Finally, Sect 7 concludes the paper and discusses our future work.

2. ACCESS WIRED NETWORKS FROM MULTIHOP AD HOC NETWORKS

2.1 Targeted Network Structure

Typically, an ad hoc network connects to the wired infrastructure through one or more gateway nodes, which can be fixed or mobile. Each gateway node has multiple interfaces (at least two). One of them should be a wireless interface operating in ad hoc mode and is used to communicate with mobile nodes of the ad hoc network. A general picture of an ad hoc network connected to Internet is illustrated in Figure 1. As a difference from wireless LANs and cellular wireless networks the mobile nodes access the Internet through multihop wireless links. The wireless subnet is really an independent network running its own ad hoc routing protocol. Typically the IEEE 802.11 DCF MAC protocol is adopted, as it is so far the de facto standard for ad hoc networks. Ad hoc routing protocols and addressing schemes must be extended to operate in such a mixed network environment. Detailed discussion of the routing is out of the scope of this paper. In our testbed experiments, we use only one gateway node, which is manually configured at each mobile node as gateway to the Internet (e.g. static routing).

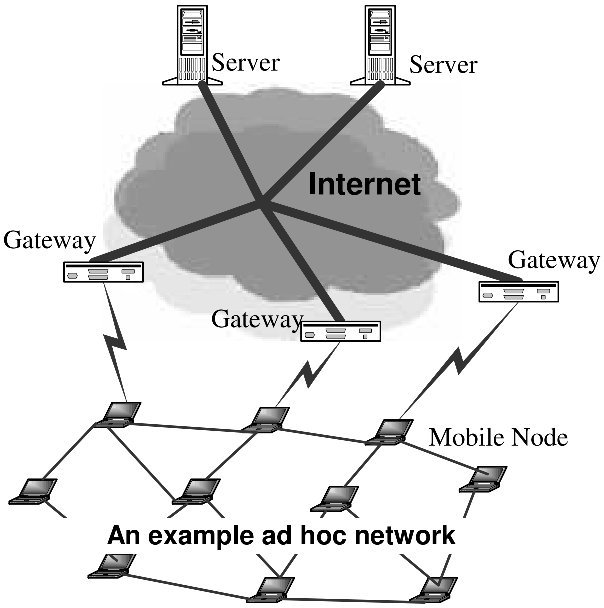
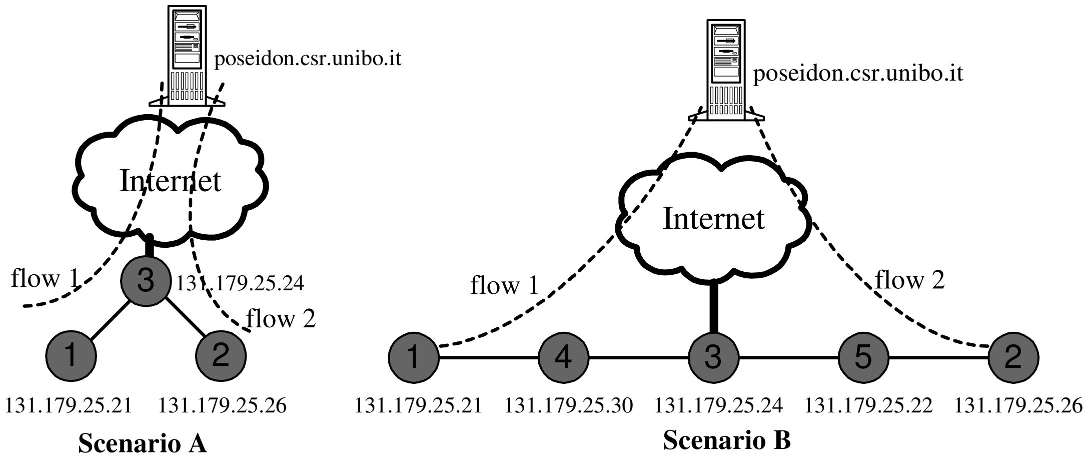


Figure 1. Illustration of an ad hoc network connected to the wired Internet

2.2 Experimental Testbed

Our experimental testbed consists of five Dell 1 GHz Pentium III Inspiron 4000 laptops equipped with network interface cards and a wired server with Ultra Spark 10 architecture located in Italy. In the wireless segment, the wired-to-wireless gateway is connected to wired and wireless networks with Xircom CardBus 10/100 ethernet card and Lucent Orinoco 802.11 pcmcia card respectively. Purely wireless nodes use only 802.11 devices. All nodes run Linux OS. The wired server has run on Linux kernel version 2.2.5 based on RedHat distribution 6.0. Laptops run on Madrake Linux distribution 8.1 with kernel version 2.4.3. We configured the Linux kernel to use TCP NewReno and disabled



Scenario A

Scenario B

Figure 2. Scenarios for testbed measurement

the TCP SACK and TCP window scaling options. Linux wavelan driver version 6.1 has been used for 802.11 devices and the devices transmission rate have been set to 2 Mbps. The MSS of TCP segment is set to 1460 bytes.

3. FAIRNESS OF MULTIPLE TCP FLOWS

We first study how multiple TCP flows share the wireless channel in our testbed. Two basic scenarios are investigated as shown in Figure 2. In both scenarios, node 3 plays the role of the gateway connecting the wired Internet with the wireless network. In scenario A, three nodes (ie, laptops) are placed in a single hop domain. This is similar to a wireless LAN scenario, with one of the nodes playing the role of access point. All radios are operating in the ad hoc mode (e.g. DCF mode of IEEE 802.11) and are in transmission range of each other. Scenario B is a simple scenario of connecting a small ad hoc network with 5 wireless nodes to the wired Internet. These 5 nodes are placed in a line. Each node can only reach its immediate neighbors. The gateway node 3 is in the middle of the line. Thus, the two end nodes are connected to the Internet via two wireless hops (two hops). The ad hoc network component of these two scenarios is quite elementary. However, it enables us to represent different situations. Scenario A is similar to a wireless LAN environment. In contrast, scenario B represents a real multihop ad hoc network. We investigate TCP performance under these two typical scenarios.

For each scenario, two FTP sessions (as illustrated in Figure 2) are established between nodes 1 and 2 and a server located at the University of Bologna, Italy. Before starting real experiments, we also measured the quality of the path from the gateway node (node 3) to the server (poseidon.csr.unibo.it). The RTT from node 3 to the Italy node is approximately 150ms; available bandwidth is around 1.2Mbps.

TCP behavior over a multihop wireless channel has been shown to be unstable and to undergo fluctuations. In order to capture these fluctuations (either short or long term) in our measurement, we run experiments with both small data files (1M) as well as large data files (8M). The direction of the TCP data flows is also important to determine performance. In our testbed experiments, there are only two FTP sessions transmitting at the same time. All three combinations of the different directions of two TCP flows are investigated. BOTH-OUT represents the case where both flows go from wireless to wired. The other combinations are BOTH-IN and MIXED. Throughput results are given in **Table 1** and **Table 2**

Table 1: Throughput measurement of scenario A

	Short term (1M file) (Kbps)		Long term (8M file) (Kbps)	
	Flow 1	Flow 2	Flow 1	Flow 2
BOTH-OUT	704.925	699.051	766.783	720.417
BOTH-IN	773.695	640.405	676.501	812.955
MIXED	755.730	702.547	779.822	710.147

Table 2: Throughput measurement of scenario B

	Short term (1M file) (Kbps)		Long term (8M file) (Kbps)	
	Flow 1	Flow 2	Flow 1	Flow 2
BOTH-OUT	241.921	156.504	185.855	283.914
BOTH-IN	211.005	224.859	240.101	278.199
MIXED	21.107	389.353	8.417	450.828

From **Table 1** and **Table 2**, we can see that two TCP flows in scenario A can roughly share the bandwidth fairly. This implies that in "last hop" wireless networks, fairness of multiple TCP flows is adequate. The wireless link doesn't bring in significant fairness issues. However, when multihop wireless links are introduced, we begin to observe serious unfairness between the two TCP flows of scenario B as shown in **Table 2**, where only the case with both flows from wired to wireless shows fair sharing. To further examine the behavior of these TCP flows, we also plot the instantaneous throughput of each TCP flow as a function of time. The instantaneous throughput is defined as following. Let X(t) denotes the instantaneous throughput of a TCP flow at time t and D_t denote the data transmitted during time period $[t+\Delta t]$. Then, we have

$$X(t) = D_t / \Delta t$$

By sliding the time window [t+Δt] and plotting the instantaneous throughput of the TCP flows, we can clearly see the dynamics of a TCP flow. Figure 3 and Figure 4 show testbed measurement results of Scenario B with MIXED traffic (e.g. FTP 1 from wireless node 1 to the wired server and FTP 2 from the wired server to wireless node 2). Clearly two TCP flows are extremely unfair. The TCP flow from wired to wireless captures the whole channel. The other flow from wireless to wired can only regain use of the channel after the first one is finished. This is not a short-term capture behavior. When we increased the data file size from 1M (Figure 3) to 8M (Figure 4), the same situation occurs. Certainly this is unacceptable. It implies that when there is traffic from the wired network to the multihop wireless network, no TCP traffic can go through from wireless to wired! When both sessions are OUT going (ie,. from wireless to wired), the unfairness problem as shown in Figure 5 is not as serious as for the MIXED traffic case. However, some unfairness is still observed. In Figure 5, we clearly see that FTP 1 finishes transmission 15 seconds later than FTP 2.

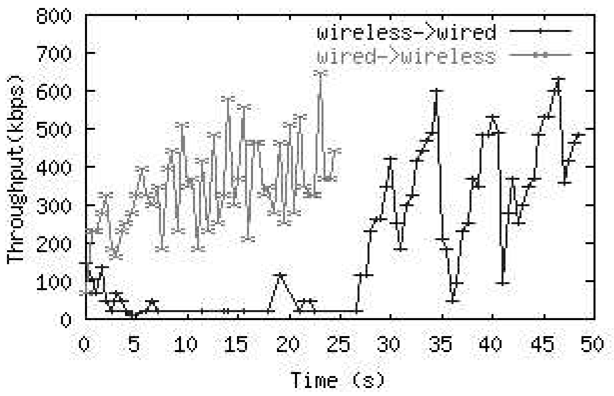


Figure 3. Instantaneous throughput of TCP flows in scenario B with FTP 1 transmitting a 1M file from wireless node 1 to wired server and FTP 2 vice versa.

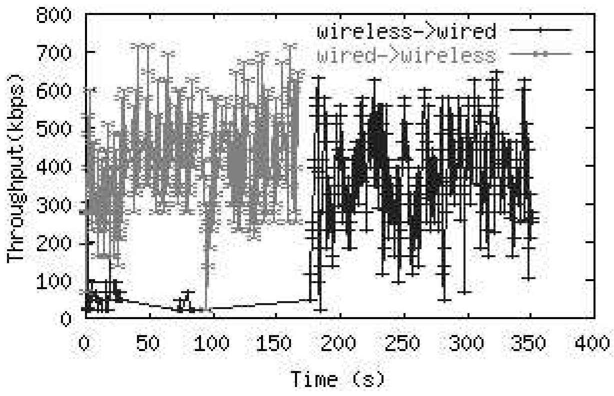


Figure 4. Instantaneous throughput of TCP flows in scenario B with FTP 1 transmitting an 8M file from wireless node 1 to wired server and FTP 2 vice versa.

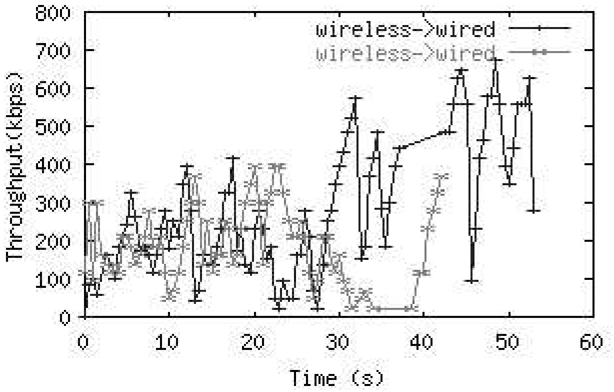


Figure 5. Instantaneous throughput of TCP flows in scenario B with both FTP 1 and FTP 2 transmitting a 1M file from wireless nodes to the wired server.

From Figure 5 above we note that, short-term unfairness is always significant. The same phenomenon was also observed by Koksal et al in [2], even though UDP traffic was used in that paper. In this study, however, we are more interested in the long-term unfairness of multiple TCP flows. This type of unfairness may starve some TCP flows, although the aggregate throughput may still be high. The extreme unfairness shown in Figure 3 and Figure 4 is certainly much more serious than low TCP throughput.

Based on our testbed as well as simulation experiments, we can draw some conclusions about fairness of multiple TCP flows across multihop wireless and wired networks.

- Fairness among multiple TCP flows across *multihop* wireless networks and wired networks is significantly different from that of *last hop* wireless LAN
- TCP flows from wired networks to wireless networks usually capture the wireless channel. Thus when mixed direction traffic coexists, flows from wireless to wired networks may experience starvation.
- Even flows from wireless to wired networks cannot share the channel among themselves in a fair way.
- Flows from wired networks to wireless networks roughly can share the channel equally.

4. TCP COEXISTENCE WITH VIDEO STREAMS

In the previous section, we have studied how multiple TCP flows share the wireless media in the wired/wireless scenario. In this section, we go further to study TCP behavior with coexisting multimedia streams. Here we assume the multimedia streams are UDP based and with fixed rate. The multimedia streams we generated in the experiments are video streams with a server located at a wireless node and a video player located in the wired network. This represents a typical scenario in the battlefield where the Unmanned Ground Vehicles (UGVs) or even Unmanned Air Vehicls (UAVs) detect intruders and transmit video taken from their cameras to the control center.

To test coexistence of TCP flows and video streams, we did a series of experiments with different video transmission rates. The topology is same as scenario B in Figure 1. The video stream is from node 2 to the wired server and a FTP/TCP connection is transmitting an 8M file from node 2 to the wired server at the same time (e.g. both of them from wireless to wired net.). We increase the transmission rates of the video stream from 80Kbps to 800Kbps representing different requirements of the quality. Selected experiment results for low video quality (80Kbps), medium video quality (540Kbps) and high video quality (800Kbps) are presented from Figure 6 to Figure 8.

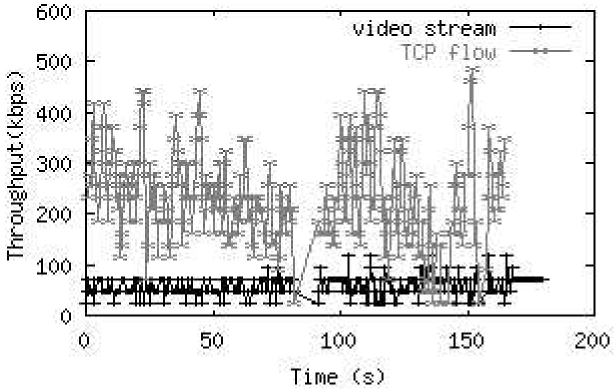


Figure 6. Instantaneous throughput of a FTP/TCP flow transmitting an 8M file from node 1 to wired server and a Video/CBR stream from node 2 to wired server at 80Kbps.

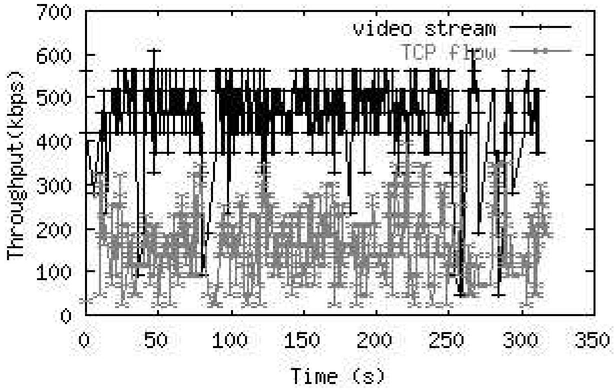


Figure 7. Instantaneous throughput of a FTP/TCP flow transmitting an 8M file from node 1 to wired server and a Video/CBR stream from node 2 to wired server at 540Kbps.

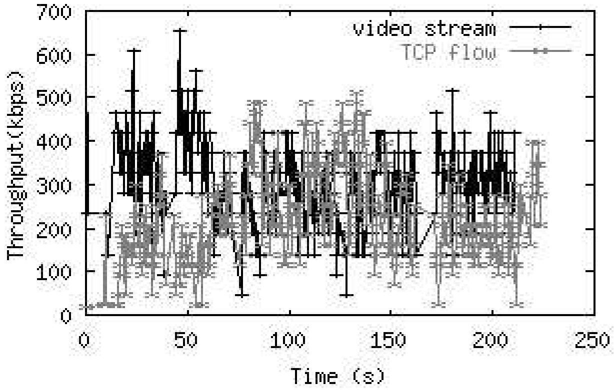


Figure 8. Instantaneous throughput of a FTP/TCP flow transmitting an 8M file from node 1 to wired server and a Video/CBR stream from node 2 to wired server at 800Kbps.

In Figure 6, the transmission rate of the video stream is as low as 80Kbps. We observe that it has no significant influence to the TCP flow, which achieves good throughput during most of the time. When the transmission rate of the video stream is increased to 540Kbps, the video stream is then dominant and drives TCP to low throughput. However, when the transmission rate of the video stream is further increased to 800Kbps, we notice that TCP performance is actually getting better and the performance of the video stream is worsened in the contrast. With a detail analysis of the trace files at all nodes, we find this is due to competition between video source node 2 and its next hop node 5. When the source node 2 transmits too fast, its next hop get little chance to send data out, thus, many packets will be dropped. This is a unique problem to multihop wireless networks where the source node shares the same wireless channel with its next hop forwarding node.

From above testbed measurements, we conclude that in the multihop wireless networks, low rate video streams are usually preferred to make it coexisting better with TCP flows. Besides, the high rate video stream may kill itself due to the fact that the source node will overwhelm its next hop node. Thus, limiting the maximum transmission rate of video streams in ad hoc networks would helpful for improving the whole network performance.

5. Analysis and Discussion

5.1 Reasons of TCP Unfairness

Several features of the multihop ad hoc network contribute to TCP unfairness. However, the most important and direct cause is the hidden and exposed terminal problem. Let us consider the two-hop domain configured around gateway node G as shown in Figure 9. There are two flows: the IN flow is from wired to wireless and OUT flow is from wireless to wired. They interact at the gateway node G. When gateway node G is transmitting a data packet to node 2, node 3 is an exposed node to this transmission. Since node 4 knows nothing about it, it may try to transmit packets to node 3 simultaneously. Under IEEE 802.11 MAC DCF mode, node 4 will first issue a RTS packet to node 3. However, node 3 cannot successfully receive the RTS packet due to the ongoing transmission from node G to node 2. Thus, node 4 will exponentially backoff. Such an uneven perception of contention limits the chances for node 4 to acquire the channel. The same thing will happen to node 3 when node 2 is transmitting to node 1. Of course, when node G or node 3 is transmitting TCP ACKs to node 3 or node 4, node 1 and node 2 will experience the same situation. However, due to small size of ACKs, their influence will be much smaller. More important, TCP ACKs may mostly be transmitted without RTS/CTS exchange (due to small size), thus the TCP ACKs from node 1 to node 2 and node 2 to node G usually will not suffer such exposed terminal problems. This may be further exacerbated by the exponential backoff of TCP's RTO timer. The final result is what we observed; namely, that the IN flows tend to overpower the OUT flows.

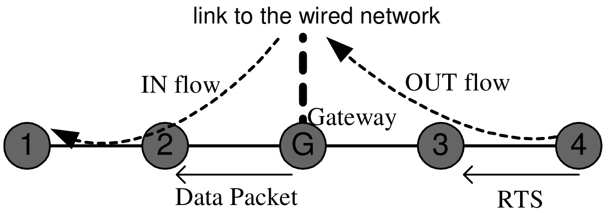


Figure 9. Unfairness of IN and OUT flows caused by hidden and exposed terminal problem

The hidden and exposed terminal problems can also happen among multiple OUT flows, although they are not as significant as for the mixed traffic case. However, we still observe serious unfairness when only OUT flows are present as shown in Figure 5. Detailed discussion of hidden and exposed terminal problems can be found in [11].

A Request to RTS (RRTS) packet is proposed in [10] to solve some kinds of hidden and exposed terminal problems. Such kind of hidden terminal problem happens frequently when all flows are from wireless networks to wired networks as illustrated in Figure 10. When node 2 is transmitting to node G, if node 4 tries to start a transmission to node 3 by sending a RTS packet to node 3, node 3 cannot reply CTS since it is hidden by transmission from node 2 to node G. However, different to the situation in Figure 9, node 3 can still successfully receive this RTS packet, although no CTS can be replied. When node 3 senses the transmission from node 2 to node G end, it then can send a RRTS packet to node 4 informing node 4 that it can resend RTS now. By this way, the two flows can share the media fairly. However, RRTS doesn't work for the situation in Figure 9 where, unfortunately node 3 cannot receive RTS successfully.

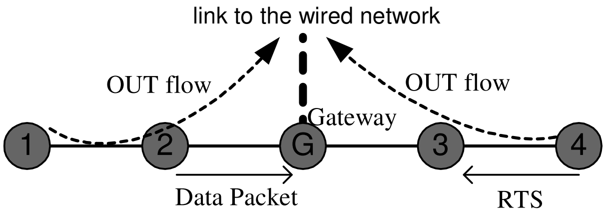


Figure 10. A hidden terminal problem happens when all flows are from wireless to wired networks

Note, the situations discussed above do not only happen around the gateway node. Actually, they can happen at any intermediate node in the multihop network part, when traffic contention occurs. But gateway nodes are hot points where such problems happen more frequently.

5.2 Optimal TCP Congestion Window Size

So far, in our experiments we have allowed the TCP congestion window to grow to its maximum, albeit we know that the multihop section performs best with a rather small window. In the next experiment, we consider the worst traffic pattern as far as fairness is concerned (i.e. Scenario B with mixed traffic) and we run repeated simulation experiments using QualNet [12] simulator with different max window values (by adjusting the receive buffer at the receiver). The throughput of the two TCP flows as a function of the maximum cwnd size is given in Figure 11. In addition, we also plot the aggregated throughput of the two flows.

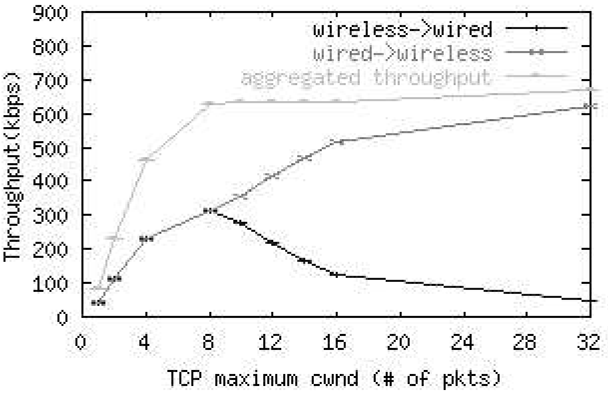


Figure 11. Relationship between maximum TCP congestion window size and fairness (Scenario B, mixed traffic)

From Figure 11, we can see that as long as the maximum congestion window size of TCP is smaller than 8 packets, the two flows can share the bandwidth pretty well. However, for W > 8, the flow from wired to wireless begins to capture a major fraction of the bandwidth. Directing now our attention to the curve of aggregated throughput, we interestingly notice that the aggregated throughput also nearly reaches the upper bound at the point where the maximum cwnd is limited to 8 packets. Clearly, 8 packets is the optimal window size for the scenario we are investigating. (Note, here the optimal cwnd size is same for both TCP flows since they share the same path in the wired network. If the two TCP flows have different paths, they will in general have different optimal window size). The extreme unfairness we observed implies that the conventional TCP cannot converge to the optimal window (in this case W=8). Rather, it overshoots badly. Further studies reveal that this optimal congestion window point is not a stable equilibrium point. When both the IN and OUT flows have increased their window size to this optimal point, a further increase has different effects on the two flows. The OUT flow will experience many more packet drops due to the hidden and exposed terminal problems discussed above. It gives bandwidth to the IN flow. This behavior is repeated every time the IN flow tries to increase its window size, leading to an equilibrium point where the IN flow has captured most of the bandwidth.

In the above experiments with manual control on the maximum TCP congestion window size, we achieved the maximum "fair" throughput for W = 8. To gain further insight into these results, consider the fact that the wired portion of the path has an RTT = 150ms and available bandwidth around 1.2Mbps. Thus, at least a window larger than 15 packets (the MDU is 1460 bytes) is needed to effectively fill the pipe. A much lower value (in fact, W =4) is required for each flow to support the 300 Kbps shown if Fig 9. This implies that up to 4 packets for each flow are present in the wireless section (Note, the IN and OUT flows in the wired part does not share the same bottleneck bandwidth since they are going in different directions). These results are further evidence that the wireless section is the bottleneck. In general, considering the low bandwidth of wireless links with respect to wired links, such a scenario will be the norm, at least in the near future.

To explore the fairness behavior in the case the wired network is bottleneck, we repeated the same experiment by connecting to a wired server located in Korea. The measured RTT of the "wired part" is only slightly increased to 160ms. But, this time the available bandwidth on the wired section is only 200kbps. In this case, we observe fair throughput of the two TCP flows in all the previous scenarios, as it should be. When the bottleneck is in the wired segment, most of the packets in the window accumulate in the wired bottleneck, leaving the wireless section of the path relatively lightly loaded and collision free.

5.3 Problems Caused by Wired Part

Other researchers [3][9] have pointed out that in a small scale ad hoc network, limiting the TCP congestion window size to one or two packets achieve the optimal throughput [9]. In [3], the authors claim that for a single chain with h hops, the optimal number of packets outstanding in the network is h/3. Thus, for the scenario B experiment, the optimum is 4/3, i.e. between 1 and 2 packets in the wireless network. Thus, 1 or 2 packets for each TCP flow. To further investigate this, we repeat scenario B this time without the access to the wired Internet. The two flows are now from node 3 to node 1 and node 2 to node 3 respectively as shown in Figure 12. We exercise control on the maximum congestion window of TCP. Individual TCP throughputs as well as aggregated throughput are shown in Figure 13. Optimal window size is now W = 2. Aggregate throughput also reaches its maximum at W=2. Further increase in window does not benefit aggregate throughput. It just hurts fairness.

This experiment also reveals the fundamental difference of TCP parameter tuning in pure ad hoc net versus ad hoc with Internet connection. While we can obtain decent behavior in the pure ad hoc network by controlling the window (in the case above, by setting the window W =1); we cannot do the same when the wired connection to the Internet is present. With a wired network path, the performance degradation caused by W =1 would be unacceptable. On the other hand, letting TCP adjust the window independently would lead to W values well beyond the optimal size which, as per Fig 9 results to be W = 8. A third option is to "force" the right value of window. However, the "right" window value is not easy to compute. Thus, separate mechanisms beside window adjustment must be developed. We are currently working on investigating some possible solutions.

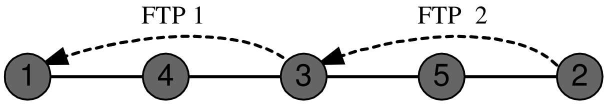


Figure 12. Scenario B without wired part (mixed traffic)

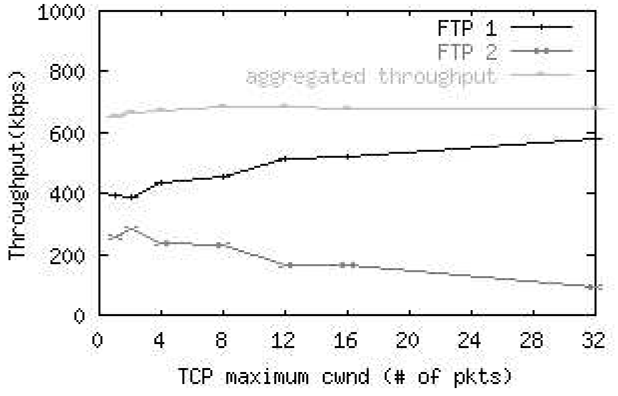


Figure 13. Relationship between maximum TCP congestion window size and fairness in pure ad hoc networks

6. RELATED WORK

Several researchers have investigated TCP fairness in pure ad hoc networks, especially under the IEEE 802.11 MAC protocol [5][6]. Xu et al have pointed out in [6] that the hidden and exposed terminal problem, the large interference range together with the binary exponential backoff (BEB) scheme of IEEE 802.11 are the major factors causing unfairness. In [5], a "yield" time is used to penalize the node that used the channel last. Such an additional yield time can improve fairness to some degree. These previous studies indicated that significant TCP unfairness/capture problems happen frequently in multihop ad hoc networks. In [5], the authors pointed out that the optimal TCP congestion window size in typical ad hoc networks is usually one or two packets, and simply proposed to limit the TCP congestion window. This scheme however will not work across wired and wireless ad hoc networks, which is the targeted scenario of our work. Moreover, we have built a testbed and performed our investigation with measurements. The previous work in the contrast is mostly simulation investigation.

In fact, the fairness of the MAC layer per se has been an active area of research. As early as in the MACAW [10] work, the *Request for RTS* (RRTS) packet and the backoff value "copying" scheme are proposed to solve some of the MAC fairness problems in a wireless LAN. In [7], a general "fair scheduling based framework" is presented to guarantee the fair share of the wireless media. A predefined fair share for each node is determined during the admission control. Then, each node will continuously monitor its currently achieved throughput. Based on this information, a fair index is calculated for each node and the BEB backoff scheme is replaced by a new scheme based on the fair index. In [8], a general algorithm is presented to translate any fair requirement into a matching backoff scheme. The fairness of the MAC layer has impact on network performance in general, regardless of the transport protocol used. However, none of the above studies has addressed TCP and the MAC protocol interactions. The authors of [9] reported that significant TCP unfairness/capture is still dominant over MACAW. In our future work, we will investigate how improving MAC fairness can improve TCP.

7. CONCLUSION AND FUTURE WORK

In this paper, we have investigated the TCP unfairness problems across multihop wireless networks (operating with IEEE 802.11 MAC) and the wired Internet using testbed measurement. The paper will prove very valuable to future commercial and military ad hoc networks, where accessing Internet is a MUST. Basically, this paper has three contributions. First, the significant unfair channel sharing among multiple TCP flows crossing ad hoc networks and the wired Internets is investigated through testbed measurements. Second, we also experimentally studied how well the video streams coexist with TCP flows in the intended scenario. Third, we explain the underlying reasons why such a significant unfairness happens. For example, even when in theory there is an optimal window size, the TCP flows cannot stabilize around those points due to different "perceptions" of channel quality. Our future work includes investigating how fair MAC schemes can improve TCP fairness as well as how adaptive video streams coexist with TCP flows.

8. REFERENCES

[1] IEEE Std. 802.11, "Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specifications," 1999. Downloadable at http://standards.ieee.org/getieee802/.

[2] C. E. Koksal, H. Kassab, and H. Balakrishnan, "An Analysis of Short-Term Fairness in Wireless Media Access Protocols," In *Proceedings of ACM SIGMETRICS 2000,* Santa Clara, CA, Jun. 2000.

[3] J. Li, C. Blake, D. Couto, H. Lee, and R. Morris, "Capacity of Ad Hoc Wireless Networks," In *Proceeding of ACM MobiCom 2001,* Rome, Italy, July 2001.

[4] K. Xu, and M. Gerla, "TCP Over an IEEE 802.11 Ad Hoc Network: Unfairness Problems and Solutions," UCLA Computer Science Department Technical Report – 020019, May 2002..

[5] K. Tang, and M. Gerla, "Fair Sharing of MAC under TCP in Wireless Ad Hoc Networks," In *Proceedings of IEEE MMT'99*, Venice, Italy, Oct. 1999.

[6] S. Xu, and T. Saadawi, "Does the IEEE 802.11 MAC Protocol Work Well in Multihop Wireless Ad Hoc Networks*?" IEEE Communications Magazine*, Volume 39 Issue 6, Jun. 2001 pp. 130-137.

[7] B. Bensaou, Y. Wang, and C. C. Ko, "Fair Medium Access in 802.11 based Wireless Ad-Hoc Networks," In *Proceedings of ACM MobiHoc 2000*, Boston, MA, Aug. 2000.

[8] T. Nandagopal, T.E. Kim, X. Gao, and V. Bharghavan, "Achieving MAC Layer Fairness in Wireless Packet Networks," In *Proceedings of ACM MobiCom 2000,* Boston, MA, Aug. 2000.

[9] M. Gerla, K. Tang, and R. Bagrodia, "TCP Performance in Wireless Multihop Networks," In *Proceedings of IEEE WMCSA'99*, New Orleans, LA, Feb. 1999.

[10] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LAN's", In *Proceedings of the SIGCOMM'94*, London UK, Aug. 1994.

[11] F. A. Tobagi, and L. Kleinrock, "Packet Switching in Radio Channels: Part II - The Hidden Terminal Problem in Carrier Sensing Multiple Access and Busy Tone Solution," *IEEE Trans. on Commun.*, Vol. COM-23, No. 12, pp. 1417-1433, 1975.

[12] QualNet simulator. http://www.qualnet.com

[13] G. Holland, and N. H. Vaidya, "Analysis of TCP performance over mobile ad hoc network," In *Proceeding of MobiCom, ACM Intern. Conf. on Mobile Computing and Networking*, pages 207-218, Seattle, WA, USA, Aug. 1999.

[14] B. S. Bakshi, P. Krishna, N. H. Vaidya, and D. K. Pradhan, "Improving Performance of TCP over Wireless Networks," *International Conference on Distributed Computing Systems*, Baltimore, MD, May 1997.

[15] P. Sinha, T. Nandagopal, N. Venkitaraman, R. Sivakumar, V. Bharghavan, "WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks," *Mobile Computing and Networking*, p231-241, 1999.

[16] H. Balakrishnan, S. Seshan, E. Amir, and R. H. Katz, "Improving TCP/IP Performance over Wireless Networks," In *Proceedings of ACM MobiCom'95*, Berkeley, CA, Nov. 1995.

[17] A. Zanella, G. Procissi, M. Gerla, M. Sanadidi, and R. Wang, "Bandwidth Estimation for Enhanced Transport over Wireless Links," In *Proceedings of ACM MobiCom'01*, Rome, Italy, Jul. 2001.

[18] TCP-Peach: http://users.ece.gatech.edu/~jtang/nasaproject_homepage/tcppeach.html.