Quantitative Assessment of IP Service Quality in 802.11b Networks

Thuy T.T. Nguyen, Grenville J. Armitage

Centre for Advanced Internet Architectures
Swinburne University of Technology
Melbourne, Australia
tnguyen@swin.edu.au, garmitage@swin.edu.au

3rd Workshop on the Internet, Telecommunications, and Signal Processing (WITSP'04), Adelaide, Australia, December 20-22 2004

***Abstract*- This paper experimentally studies the performance of 802.11b links in terms of round trip time (RTT) under load and TCP throughput in the presence of competing traffic. Our test scenarios are of specific interest to the emerging "hot spot" market, and the use of 802.11b in enterprise networks. We use commercial, standards-compliant 802.11b clients and access points, and demonstrate that CSMA/CA as currently used by 802.11b allows competing low-rate traffic (such as 128Kbps flow of 64 byte UDP or ICMP packets) to degrade concurrent TCP throughput by up to 50%. We also observe that data transfers from a remote server to a wireless client can create substantial latency spikes (over 50ms for MTU 1500 bytes) in the shared wireless segment, even when the TCP throughput is up around 4Mbps. Such spikes have a big impact on delay-sensitive applications, such as voice over IP (VoIP), online games and interactive streaming video that may be sharing the wireless medium. We also experimentally characterize the impact of the bottleneck link bandwidth, MTU sizes, and TCP window sizes in achieving maximum performance over 802.11b wireless links.**

Keywords- 802.11b, data throughput, wireless network, TCP performance

I. INTRODUCTION

With transmission speeds comparable to a wired 10Mbps Ethernet, 802.11b wireless networks are now able to provide high-speed files transfer, web surfing and support wireless interactive applications such as online gaming and videoconferencing. However, the service quality implications of mixed interactive and non-interactive applications and performance degradation that would be seen by a wireless client sharing the Access Point at a hot spot have not been fully explored.

This paper describes our experimental study based around real-world 802.11b wireless equipment. We emulate multiple wireless clients sharing the same Access Point (AP) at a hot spot (such as an Internet café or at the airport) and explore the interactions between TCP flow control behaviour and end-to-end latencies experienced by applications sharing a wireless medium. The results of our study will be useful in motivating and guiding future work on priority queuing and packet scheduling systems in 802.11b networks.

Limited link capacity between AP and wireless client can introduce substantial delay to other traffic sharing the wireless medium along their end-to-end path (in the order 100ms for 512-byte MTU). Delay-sensitive applications sharing the AP, such as voice over IP (VoIP) or interactive online game traffic, are affected by the increase in per-hop latency. We also characterize the impact of MTU and TCP flow control window settings on both TCP performance and the latency-jump imposed on interactive applications sharing the link.

Another major source of performance degradation has its origins in 802.11b's CSMA/CA link sharing and access mechanism. We show that low-rate, non-reactive packet flows to and from one client can steal substantial capacity from concurrent TCP flows to other clients. For example, a flow of 64 byte 'ping' ICMP packets at 250 packets per second (roughly 128Kbps) degrades a concurrent TCP flow's throughput by up to 50% (from 4Mbps to ~2Mbps). This has implications for the use of UDP-based audio and video conferencing applications at 802.11b hotspots or in enterprise networks. We also observe that typical wireless clients using a fixed, small TCP receiver window size see nearly half the throughput received by a well-tuned TCP client sharing the link.

The rest of the paper is organized as follows. Section II provides a background on 802.11b. Section III briefly discusses some related work and our contributions. Section IV outlines details of our experimental setup, and presents analysis of the findings and theoretical verification of our results. The paper is concluded in section V with some discussions of implications for mixing interactive and non-interactive traffic.

II. BACKGROUND

A. *Basic IEEE 802.11b operations*

The IEEE's 802.11b specifications define the physical layer and media access control (MAC) sublayer for communications across a shared, wireless local area network at up to 11Mbps [1]. At the physical layer, IEEE 802.11b radios operates at 2.45 GHz and use direct sequence spread spectrum (DSSS) transmission. At the MAC sublayer 802.11b uses carrier sense multiple access with collision avoidance (CSMA/CA).

802.11b operates in either ad hoc mode or infrastructure mode. In ad hoc mode wireless clients communicate directly with each other. In infrastructure mode, wireless clients

3rd Workshop on the Internet, Telecommunications, and Signal Processing (WITSP'04), Adelaide, Australia, December 20-22 2004

communicate with a wired network (an enterprise LAN or Internet connection) or other clients via an Access Point (AP). Infrastructure mode networks consist of APs, wireless clients (computing devices with 802.11b-based network interfaces) and a wired network. An AP acts as an Ethernet bridge between wireless clients and the wired network. Our study focuses on 802.11b in infrastructure mode.

B. *CSMA/CA with Positive Acknowledgment and Virtual Carrier Sense*

Because radio links are typically unreliable 802.11b utilizes Positive Acknowledgement (ACK) of every transmission and a Virtual Carrier Sense mechanism to reduce the probability of two clients colliding[2][3].

A client wanting to transmit senses the medium and defers if the medium is busy. The client transmits when the medium is free for a specified time (the Distributed Inter Frame Space, DIFS). First, a Request to Send (RTS) control packet is transmitted, carrying the source, destination, and duration of the desired transaction (a data packet and the corresponding ACK). The receiver responds (if the medium is free) with a Clear to Send (CTS) control packet which include the same duration information. Upon receiving a CTS, the sender waits for a Short Interframe Space (SIFS) then sends the data packet. The receiver checks the received packet for errors, waits for a SIFS and sends an ACK packet. The receipt of the ACK indicates to the sender that no collision occurred. The sender would retransmit the data frame until it gets acknowledged (and throws the data frame away if not ACKed after a number of unsuccessful retransmissions).

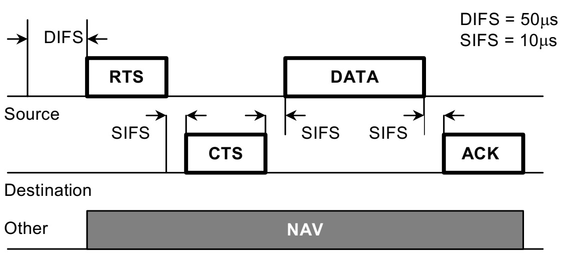

Fig. 1. RTS/CTS and data transactions[1]

All stations receiving either the RTS and/or the CTS, will set their Virtual Carrier Sense indicator (called Network Allocation Vector, NAV), for the duration. Would-be senders use the status of their current NAV, in conjunction with physical carrier sensing, to decide if the medium is likely to be in use at any given time. Fig. 1 shows the transactions between two wireless clients and the NAV status of a third, neighboring node[2].

C. *802.11b encapsulation*

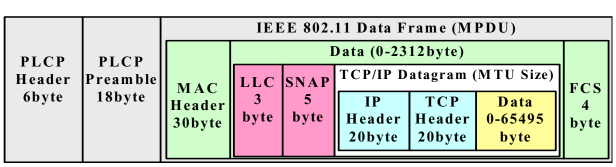

Fig.2. 802.11frame encapsulation

[1] Back off time scheme [4] is not considered here to simplify the analysis.

In addition to the payload data, the MAC frame encapsulation process adds 42 additional bytes of overhead (Fig.2). The 802.11 MAC header adds 30 bytes of data for various control and management functions, error detection, and addressing and a trailing 4 byte Frame Check Sequence (FCS). LLC/SNAP encapsulation adds another 8 bytes[5].

A PLCP (Physical Layer Convergence Protocol) header and a PLCP preamble is prepended to every frame before it is transmitted. These headers are transmitted at 1Mbps. The PLCP preamble may be either a "long" preamble of 18bytes, or a "short" preamble of 9 bytes. Long preamble is the default setting on most devices so we based our theoretical calculations on a long preamble that takes 192µs to transmit [6].

D. *802.11b timeline*

The MAC frame is transmitted as a series of 8-bit symbols at maximum 1.375 milion symbols per second. From this we can estimate the transaction time for a hypothetical TCP stack that requires one TCP ACK for every TCP Data packet. A 1500 byte TCP/IP data packet thus generates 1542 symbols in the MAC data frame, while the TCP ACK frame generates 82 symbols. The 802.11b ACK is 14 bytes long, and the RTS and CFS packets are 20byte and 14byte respectively. Based on this, we can calculate the overall transaction time as shown in Table 1[6].

	1500-byte MTU TCP Data (µs)	**TCP ACK (µs)**
DIFS & SIFS	50 + 10*3 = 80	50 + 10*3 = 80
RTS & CTS	192*2 + (20 +14)/0.125 = 656	192*2 + (20 +14)/0.125 = 656
802.11 Data	192 + 1542/(1.375) = 1,313.4	192 + 82/(1.375) = 251.6
802.11 ACK	192 + 14/(1.375) = 203	192 + 14/(1.375) = 203
Frame exchange total	2252.4	1190.6
Total Transaction	**3443**	

Table 1 - TCP transaction time

III. RELATED WORK

Many papers have studied the performance of 802.11b network. Effects of varying bit rate used by a wireless client on other mobile hosts sharing the link were studied in [7]. The situation considered was when a host was far away from an AP and hence was subject to signal fading and interference, which caused its bit rate to degrade from 11Mbps to 5.5, 2 or 1 Mbps. In such a case, other hosts sharing the AP, though they were transmitting at 11Mbps, would degrade to a rate of lower than 1Mbps due to the basic CSMA/CA channel access method. Reference [8] investigated the short-term unfairness of the CSMA/CA as implemented in the WaveLAN network. Reference [9] characterized the expected performance of the standard's ad hoc and infrastructure 802.11b networks. Its simulation models incorporate the effect of burst errors, offered load, packet size, RTS threshold and fragmentation threshold on network throughput and delay.

In this paper, we measure and characterise the negative impact of 802.11b's CSMA/CA on end-to-end TCP performance in the face of low bandwidth, non-reactive traffic. Our test scenario is where a number of 802.11b-enabled game clients cluster around an 802.11b hot-spot that

3rd Workshop on the Internet, Telecommunications, and Signal Processing (WITSP'04), Adelaide, Australia, December 20-22 2004

could easily lead to unexpected performance degradation for a client running TCP applications. We show that low-rate, non-reactive packet flows to and from one client can 'steal' significant capacity from concurrent TCP flows to other clients. This has implications for the use of UDP-based audio and video conferencing and game applications at 802.11b hotspots or in enterprise networks. Similar work had been done in [10], in which they conclude that in multihop wireless networks, low rate video streams are usually preferred to make it coexisting better with TCP flows. However, the authors did not go into detail analysis. In this paper, we demonstrate that even a low-rate, non-reactive packet flow of 128Kbps could degrade a concurrent TCP flow's throughput by up to 50%. We verify our finding based on theoretical analysis of the CSMA/CA and positive ACK scheme used in the current 802.11b networks.

Other parameters such as bottleneck bandwidth, MTU sizes and TCP receiver window sizes are also considered and investigated in our work.

While most other papers use simulation [9][11] or Markov chains [8] for their analysis, our major contribution is the use of direct trials on commercial equipment, rather than relying on simulations that (of necessity) do not always properly implement all aspects of the respective protocols.

IV. Test Setup & Findings

A. *Impact of a low-rate, non-reactive interference traffic in contention for the medium*

First we examined TCP performance from one wireless client while a low-rate, non-reactive traffic flow (in our case a flow of ICMP 'ping' packets) from another wireless client competes for resources on the 802.11b link (Fig. 3).

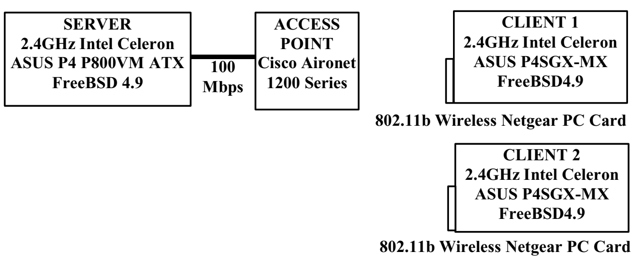

Fig. 3. Test1 setup

Our server and both clients ran FreeBSD 4.9. Repeated

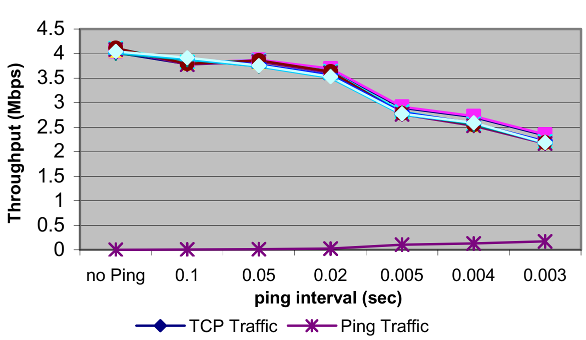

Fig. 4. nttcp throughput vs. Ping rate

runs of nttcp [12] were used to measure TCP performance between client 1 and the server. Each trial transferred 8Mbyte (using the nttcp default of 2048 4Kbyte buffers) three times. We injected interfering traffic over the wireless link by concurrently "pinging" the Server from Client2, using ping interval from 100ms down to 3ms.

The resulting degradation in TCP throughput was quite dramatic, particularly given the relatively low bandwidth of the competing ICMP traffic flow (Fig. 4). When the ping interval was 3ms (roughly 171Kbps given 64 byte ICMP packets) the TCP throughput dropped by 50% (from 4Mbps to approximately 2Mbps).

An explanation for this observed behavior could be found by closer analysis of the 802.11b frame transmission protocol. We know from Table 1 that a TCP transaction requires 3443μs. The time taken to complete one ping transaction (an ICMP echo request and ICMP echo reply) is calculated in Table 2 to be ~2416.2μs for a 64 byte ping packet. If we treat every ping transaction as a lost opportunity for transmitting TCP data, then we can predict the TCP degradation fairly well. For example, assume we send one ping every 4ms, i.e. 250 packets per second or roughly 128Kbps. The total time taken by 802.11b link to handle these transactions would be 250*2416.2μs = 604.05ms. During that time, 604.05ms/3443μs = 175.4 TCP transactions could have occurred if there had been no competing ping traffic. With a 1500byte MTU nttcp would lose ~2.05Mbps.

	64-byte Echo Request & Reply (μs)	**128-byte Echo Request & Reply (μs)**	**256-byte Echo Request & Reply (μs)**
DIFS + RTS +CTS + SIFS	736	736	736
802.11 Data	192 + (64+42)/(1.375) = 269.1	192 + (128+42)/(1.375) = 315.6	192 + (256+42)/(1.375) = 408.7
802.11 ACK	192 + 14/(1.375) = 203	192 + 14/(1.375) = 203	192 + 14/(1.375) = 203
Frame exchange total	1208.1	1254.6	1347.7
Total Transaction	1208.1*2 = **2416.2**	1254.6*2= **2509.2**	1347.7*2 = **2695.4**

Table 2 – Ping transaction time

Fig. 5 shows the nttcp throughput seen at client 1 (the continuous line) and the effective nttcp throughput 'stolen' by the ping flow from client 2 (the dotted line). The sum of

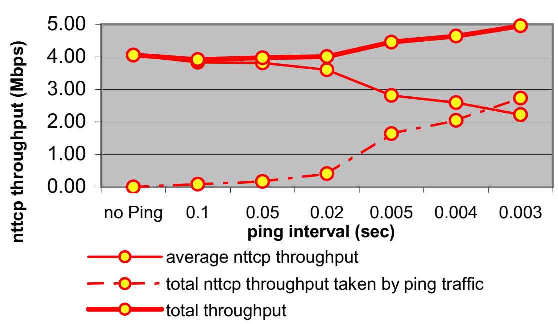

Fig. 5. nttcp throughput of TCP traffic and taken by ping traffic

these rates matches that achieved by nttcp in the absence of competing traffic. (The predicted total drifts higher than

3rd Workshop on the Internet, Telecommunications, and Signal Processing (WITSP'04), Adelaide, Australia, December 20-22 2004

4Mbps due to our simplified calculation of equivalent TCP throughput was stolen by the competing ICMP traffic.)

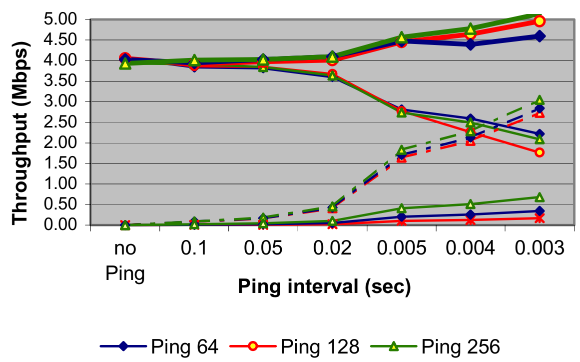

Fig. 6. Results for ping packet sizes of 64, 128 and 256 bytes

Fig. 6 shows the nttcp results when the interfering ping packets were 128 and 256 bytes long (thin solid lines), the predicted 'stolen' capacity due to the ping traffic (dotted lines), and the sum total (thick solid lines). These results indicate that an 802.11b's link shared link capacity can be substantially degraded with only modest level of competing traffic (e.g. 100 to 200 packets per second). At such low rates the very act of transmitting the ICMP packets was more significant then their size in 'stealing' capacity from other clients on the link.

We have related our experimental observations back to a theoretical model of performance estimation. From this basis we can work out the expected maximum throughput that the AP can provide under different circumstances. For example, if we were only sending ICMP ping packets the 802.11b link could handle no more than 867 pings per second (10^6/1154µs). If the traffic was solely TCP with 1500 byte TCP/IP data packets we would be limited to 458 TCP transactions per second (10^6/2180µs).

B. *Impacts of MTU size, Window Size and Server –AP link bandwidth on path latency and TCP throughput.*

We then examined the relationship between overall TCP performance and rate caps between the server and the AP, MTU on the link and TCP maximum window size. We used the simplified configuration show in Fig.7

We repeatedly ran nttcp from server to client with

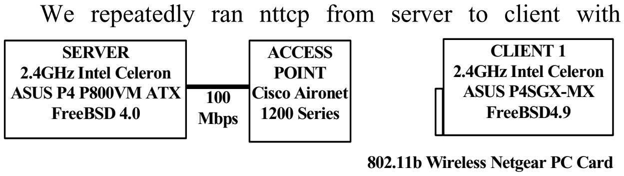

Fig.7 - Testing raw nttcp throughput as a function of external parameters

different link MTUs, and gathered round trip time (RTT) estimates before, during and after each run using ICMP ping (one per second) from client to server. Our RTT results are presented in Fig. 8. Each trial transferred 8Mbyte (using the nttcp default of 2048 4Kbyte buffers) three times with TCP client window of 32Kbyte.

The presence of traffic in the server to client direction causes a dramatic increase in RTT (with the actual increase

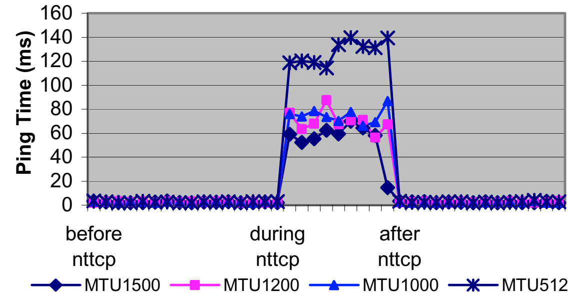

Fig. 8. Ping Time before, during and after nttcp transfer

depending noticeably on MTU). An idle link shows 2.6ms RTT. During the nttcp transfer phase the RTT jumps to just over 120ms at an MTU of 512 bytes, 70ms with 1000 byte MTU, 67ms with 1200 byte MTU and 55ms with 1500 byte MTU respectively.

To characterize this increase in RTT as a function of

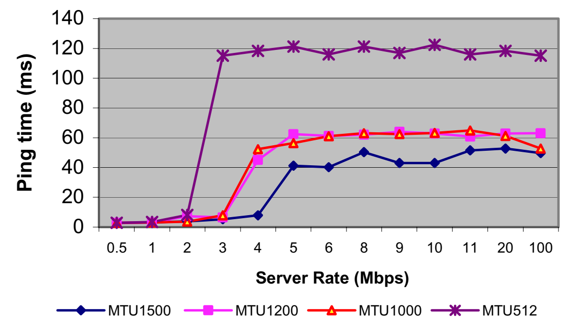

Fig.9. Ping Time during nttcp transfer for all MTUs

offered load we artificially throttled the server to client data rate at the server using FreeBSD's kernel-resident 'dummynet' module. We set dummynet's internal queue limit to 62Kbytes and applied bandwidth limits between 500Kbps and 100Mbps (the natural rate of the server to AP link). TCP ACKs and all ICMP packets were not rate limited.

We repeated the test with different MTU sizes of 1500, 1200, 1000 and 512 bytes, with the maximum TCP window sizes of 32Kbyte. Fig.9 shows the average RTT during an

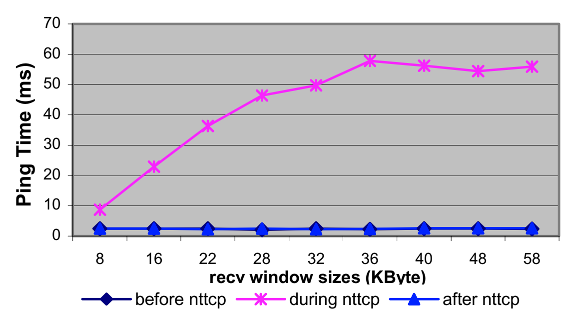

Fig. 10. Ping Time -MTU 1500-Server rate 100Mbps

nttcp transfer as a function of server-side rate limit. For each MTU the server rate is varied from 500Kbps to 100Mbps.

3rd Workshop on the Internet, Telecommunications, and Signal Processing (WITSP'04), Adelaide, Australia, December 20-22 2004

The RTT increases dramatically at the point where the server's offered load begins to exceed the wireless link's maximum rate for the specific MTU, at ~2Mbps for MTU 512, ~3Mbps for MTU 1200 and 1000, and ~4Mbps for

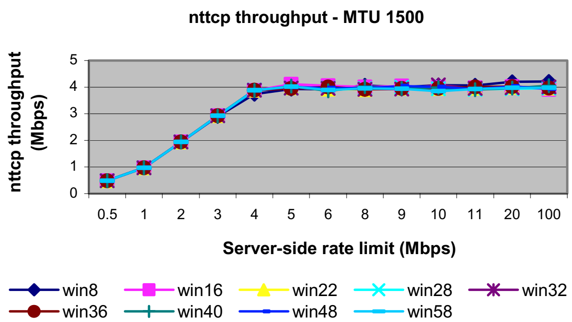

Fig. 11. nttcp throughput for different window sizes – MTU1500

MTU 1500 (taking into the account the 802.11b MAC overheads discussed in Section II). A smaller MTU implies a higher relative 802.11b overhead per frame, thus the

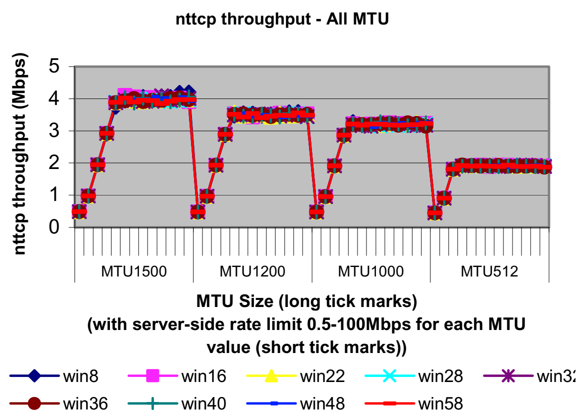

Fig. 12. nttcp throughput for different window sizes – all MTUs

effective wireless link capacity is smaller for smaller MTUs. When the server is rate limited to less than the wireless link's capacity, the RTT stays low. However, the size of the RTT

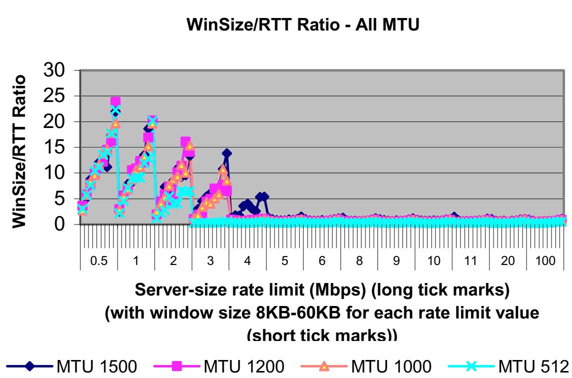

Fig. 13. Window Size/RTT ratio – All MTUs

itself increases significantly when the wireless link itself becomes the bottleneck.

Fig. 10 shows the RTT before, during and after an nttcp transfer with MTU of 1500 bytes, for different window sizes, and where the server-side rate limit is off (100Mbps). The interaction between offered load, total link RTT and optimal window size is intriguing. Up to a point the amount by which the RTT increases during an nttcp trial itself increases as the configured maximum TCP window is set to larger values. It is noteworthy, however, that the actual throughput reported by nttcp is not noticeably affected by the receive window size when the wireless link itself is the bottleneck (Fig. 11 and Fig. 12). The best TCP throughput is bounded solely by the lesser of the server-side rate limit or the wireless link capacity, even down to an 8Kbyte maximum TCP window.

The ratio of window size and RTT (WinSize/RTT) across all server-side rate limits and MTU sizes (Fig. 13) provides a useful perspective. WinSize/RTT increases with larger window size when the server-side rate limit is more restrictive than the wireless link. The reason is that despite the increase in the window size, the RTTs stay the same at ~2.6ms (see Fig. 10). It is so small that any TCP windows over 1.3Kbytes saturate the path. RTT increases substantially when the wireless link becomes the bottleneck. In this case the wireless link's bandwidth is saturated, RTT increases as the TCP receiver window size increases (WinSize/RTT ratio stays nearly constant for all window sizes at ~0.5Mbyte/sec

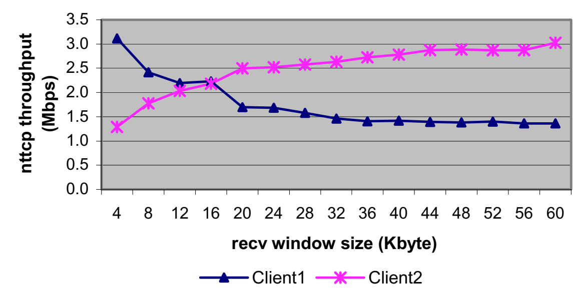

Fig. 14. Average nttcp throughput, Client 1 vs. Client 2

~ 4Mbps).

C. *Impacts of receiver window size on TCP performance in a contention for medium*

One might conclude that there's no benefit in tweaking a client's maximum TCP window if, for example, we're delivering local content at an 802.11b "hot spot". However, when multiple clients are involved it is possible to gain an advantage through tweaking the TCP window.

With the setup from Fig. 3 we ran nttcp on both clients to concurrently download 4 MBytes from the Server. Client 1 used a fixed 16Kbyte window while client 2 varied its window from 4Kbytes to 60Kbytes. Fig. 14 shows that client 2's optimization of their window degrades the service experienced by client 1.

V. Discussion and Conclusions

Although the IEEE's 802.11b wireless LAN standard is being adopted all over the world - becoming a popular choice for enterprise networks and public "hot spots" and Internet Cafes – it is clear that end to end service over 802.11b networks exhibits a few non-obvious characteristics. We have experimentally characterised two operational modes that would create confusion and dissatisfaction in the minds of customers – the ability of low-rate, non-reactive flows to ‘steal’ substantial capacity from TCP flows sharing an AP, and a large spike in RTT experienced by all users of a given AP when the AP is under load. We have also noted the relationships between client-controlled TCP window size and client-perceived performance. Our experimental results are arguably intrinsic to 802.11b and do not reflect peculiarities of the commercial equipment we used in our test bed.

These insights are of particular importance when engineering robust and predictable services to discerning customers – whether the general public, corporate colleagues, or small villages relying on 802.11b for their last/first hop Internet access. For example, a UDP-based IP telephony or video conferencing application might end up ‘stealing’ far more capacity than you expect because its small, yet frequent packet transmissions lead to disproportionate consumption of 802.11b MAC layer resources. In our experiments a fairly trivial 128Kbps packet stream from one client to AP caused another client to perceive the link to have lost almost 2Mbps of available capacity.

The RTT spike also has significant implications for ISPs who wish to concurrently host local content and yet support interactive applications such as voice, video and games through their AP(s). For example, consider an ISP who encourages their wireless clients to use local, well-connected content servers (either explicitly, or e.g. by transparently forcing client web browsing through a local caching proxy). All clients sharing the AP will find their RTT to other parts of the Internet jumping by over 50ms while someone is performing local content transfer– rather disruptive to other clients who may be engaged in online game play or tele-conferencing at the time. Our experiments revealed that smaller MTUs actually trigger higher RTT spikes (the TCP flow downloading local content requires more packets per second for the same TCP throughput, triggering more 802.11b MAC layer overhead).

We also noted experimentally that clients using a fixed, not-uncommonly small TCP receiver window of 16Kbyte could have their achieved throughput severely disadvantaged by other local clients adjusting their own received window size to a far higher value. Given that client-side TCP windows are not under control of the wireless ISP operator, this provides another avenue for inexplicable customer dissatisfaction if not well understood.

Our experiments and theoretical discussion illuminate a number of factors that wireless network operators must consider when deploying 802.11b services to customers or clients having heterogeneous operating environments and service requirements. We believe our results can be extrapolated to provide insights into real-world behaviour of end to end Internet paths containing 802.11b wireless links.

Our future work will take into account the impacts of some other factors, such as the backoff time in the CSMA/CA scheme, collision rate, packet size and transmission probability.

ACKNOWLEDGMENT

We thank Cisco Systems Australia for supporting our work at the Centre for Advanced Internet Architectures through equipment donations and post-graduate student stipends.

We would like to thank the anonymous referrees for their evaluation feedback to improve the presentation of the paper.

REFERENCES

[1]. IEEE 802.11b Wireless Networking Overview, March 2002 http://www.microsoft.com/technet/community/columns/cableguy/cg0302.mspx

[2]. Brenner,P., A Technical Tutorial on the IEEE 802.11 Protocol, 1997 http://www.sss-mag.com/pdf/802_11tut.pdf

[3]. Campbell, T.A, IEEE 802.11Wireless LAN http://www.columbia.edu/itc/ee/e6951/2002spring/LectureNotes/lecture3_4.pdf

[4]. Bianchi, G 2000, 'Performance Analysis of the IEEE 802.11 Distributed Coordination Function,' IEEE Journal on Selected Areas in Communications, vol. 18, pp. 535-547.

[5]. Leira, J. “Throughput”, March 2003 http://www.uninett.no/wlan/throughput.html

[6]. Gast,M.“ When Is 54 Not Equal to 54? A Look at 802.11a, b, and g Throughput” Aug 2003 http://www.oreillynet.com/pub/a/wireless/2003/08/08/wireless_throughput.html

[7]. M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, “Performance Anomaly of 802.11b,” in Proceedings of IEEE INFOCOM 2003, San Francisco, 2003.

[8]. C. Koksal, H. Kassab, and H. Balakrishnan, “An Analysis of Short-Term Fairness in Wireless Media Access Protocols,” in Proceedings of ACM SIGMETRICS, 2000.

[9]. B. P. Crow et al., “Investigation of the IEEE 802.11 Medium Access Control (MAC),” in Proceedings of INFOCOM 1997, 1997.

[10]. K. Xu, S. Bae, S. Lee and M. Gerla, TCP behaviours across multihop wireless networks and the wired internet, Proc. of the 5th ACM international workshop on Wireless mobile multimedia, (2002) pp 41-48.

[11]. M. Natkaniec et al., “Analysis of Backoff mechanism in IEEE 802.11 standard,” in Proceeding of ISCC 2000, 2000.

[12]. http://www.leo.org/~elmar/nttcp/

[13]. Tianbo,K.and Williamson, C. "RealMedia Streaming Performance on an IEEE 802.11b Wireless LAN" Proceedings of IASTED Wireless and Optical Communications (WOC) Conference, Banff, AB, Canada, pp. 306-311, July 2002

[14]. Sharma,S "Analysis of 802.11b MAC: A QoS, Fairness, and Performance Perspective," Research Proficiency Exam, Jan 2003 http://www.ecsl.cs.sunysb.edu/tr/wlanrpe.pdf

3rd Workshop on the Internet, Telecommunications, and Signal Processing (WITSP’04), Adelaide, Australia, December 20-22 2004