WPI-CS-TR-06-20 October 2006

Measuring the Queue Sizes of IEEE 802.11 Wireless Access Points

by

Feng Li
Rui Lu
Mingzhe Li
Huahui Wu
Mark Claypool
Robert Kinicki

Computer Science
Technical Report
Series

WORCESTER POLYTECHNIC INSTITUTE

Computer Science Department
100 Institute Road, Worcester, Massachusetts 01609-2280

Measuring the Queue Sizes of IEEE 802.11 Wireless Access Points

Feng Li, Rui Lu, Mingzhe Li, Huahui Wu,Mark Claypool, and Robert Kinicki

Computer Science Department at Worcester Polytechnic Institute
Worcester, MA, 01609, USA
{lif,kkoy,lmz,flashine,claypool,rek}@cs.wpi.edu

Abstract. The increasing power, lower cost and improved convenience of wireless networks is fueling the spread of IEEE 802.11 wireless access points (APs) in the residential and commercial environments. While there has been considerable study of wireless performance, little is publicly known about the queue sizes of wireless APs. This paper presents the QFind methodology for measuring wireless AP queue sizes. QFind determines the wireless saturation point, measures the baseline delay, induces the saturation rate and measures the delay with queuing, and computes the queue size. The accuracy of QFind is validated in a wireless testbed with a Host AP, where the controlled queue size is compared to the QFind measured queue size. QFind is then used to measure the queue sizes of seven wireless APs, three commercial class APs and four residential class APs targeted for residences. The results show that wireless AP queue sizes are packet-based and vary considerably across vendor and AP classes, with queue sizes ranging 50 packets to over 350 packets. This suggests additional engineering and science to determine the best mechanisms for providing AP queue sizes for a variety of traffic types.

1 Introduction

Over-provisioning core network routers has meant that Internet performance bottlenecks are likely to appear on the "last-mile" wireless link to and from the end host. While available wireless link capacities have increased to 54 Mbps for IEEE 802.11g networks, their effective capacities remain below standard wired uplink capacities of 100 and 1000 Mbps. This suggests that for downlink Internet traffic emanating from a wired Internet connection to a wireless receiver, the wireless AP is the performance bottleneck. With bursty Internet traffic, this scenario implies the significance of selecting the proper queue size for the AP downstream links to accomodate the variability of higher-speed wired network traffic passing through the AP onto lower-speed wireless links.

Since Internet traffic is often bursty [**?**], partially due to flows with high round-trip times or large congestion windows, the wireless AP designer's choice of queue size has a direct impact on a flow's achievable throughput and delay. A small queue can keep achieved bitrates significantly below the available capacity, while a large queue can negatively impact a flow's end-to-end delay. Interactive

applications, such as IP telephony and some network games with strict delay bounds in the range of hundreds of milliseconds, will experience degraded Quality of Service when large AP queues become saturated by other concurrent flows. However, there has been little research conducted on AP queue size choices.

Actual queue size measurement have primarily looked only at wired queues, such as access queues for network switches [**?**] or DSL or Cable modems [**?**]. General guidelines for determining the "best" queue sizes have often been debated on the e2e mailing list,[1] an active forum for network related discussion by researchers and practitioners. While general consensus has the access queue size ranging from one to four times the bandwidth-delay product of the link, round-trip times can vary by at least two orders of magnitude (10 ms to 1 second) [**?**]. Thus, this provides little guidance when considering the best queue size for wireless.

This paper presents a detailed measurement study of the queue sizes of IEEE 802.11 wireless access points. Since commercial APs are "black-boxes" in that internal configurations are not revealed, the QFind methodology for measuring queue sizes is proposed. After determining the saturation throughput rate for a specific wireless AP, Qfind sends high bitrate traffic from the wired uplink over the wireless downlink and filling the AP queue. Careful measurements of the added queueing delay and achievable throughput provide a means to accurately determine the queue size. QFind is applied to seven different APs in two different classes, commercial and residential. The results show that AP queue sizes are packet-based and vary considerably both across AP classes and even within a class. This suggests AP designers have different assumptions about the number of flows and round-trip times of traffic traversing their APs, or, worse, that AP queues are picked without careful consideration of their queue sizes.

These results have implications for wireless AP designers that may be able model their AP after those of other vendors or may wish to explore the impact of queue lengths in more detail. Bandwidth estimation tools that send trains of packets can also make use of the queue size information to tailor their tools not to overflow wireless AP buffers. Lastly, network researchers that make use of simulation or Host APs that explicitly control queue length can base their queue settings on the results presented in this paper.

The rest of this paper is organized as follows: Section 2 describes QFind; Section 3 validates the QFind method; Section 4 details the experimental setup for AP queue measurement; Section 5 presents AP queue measurement results; Section 6 discusses the implications of the results; Section 7 presents related work; and Section 8 summarizes our conclusions and presents possible future work.

[1] In particular, see the e2e list archives at: ftp://ftp.isi.edu/end2end/end2end-interest-1998.mail and http://www.postel.org/pipermail/end2end-interest/2003-January/-002702.html.

2 QFind

To measure the queue size of wireless access points, we have developed *QFind*. After determining the AP saturation rate, a QFind session sends downstream traffic to determine the maximum queuing delay due to the AP queue. Repeated sessions and statistical analysis provides details on the queueing delay and throughputs, that are then used to compute queue size.

While QFind can estimate the size of any AP queue, this paper focuses on the downlink queue on the AP where infrastructure traffic flows from a wired source through the AP to the wireless receiver. Given the 100 Mbps capacity typical of most APs is greater than the maximum IEEE 802.11g 54 Mbps capacity on the wireless downlink, any queueing is likely to happen in the downlink direction. If the wireless link is the bottleneck to performance, there may be queueing on end-hosts sending packets to the wireless AP. However, studying these queues is less interesting as end-hosts block on a full queue, not inducing loss if the queue is filled, and end-hosts typically have fewer flows contending for the queue resources than does an AP.

2.1 Concept

Based on previous work [**?**], the QFind technique has four steps: 1) measure end-to-end delay on an unloaded network; 2) induce enough network load to fill up the bottleneck queue; 3) while at network load measure the delay and throughput; and 4) use these measurements to infer the queue size of the access link. Let D_h be the highest delay measured (with a full queue):

$$D_h = D_l + D_q \tag{1}$$

where D_l is the lowest delay measured (with an empty queue) and D_q is the queuing delay. Therefore:

$$D_q = D_h - D_l \tag{2}$$

Given throughput T (measured during the loaded period), the AP queue size in bytes, q_b, can be computed by:

$$q_b = D_q \times T \tag{3}$$

For a packet size s (say 1500 bytes including IP headers, a typical MTU), the queue size in packets, q_p, becomes:

$$q_p = \frac{(D_t - D_l) \times T}{s} \tag{4}$$

2.2 Link Saturation

To ensure the AP queue is full, the downlink offered load must be greater than the effective wireless link capacity. Previous research [**?**] has shown effective wireless

capacities for 802.11b APs vary widely from about 5 to 7 Mbps, depending upon the specific AP. Thus, the first step is to determine the traffic load that saturates the wireless link for a given AP.

The Qfind saturation algorithm begins with a sending rate of $S_0 = 50$ packets/sec. The sending rate is incrementally increased by 50 packets/second until packet loss rates of 20% or more is observed for three successive sending rates, S_{i-1}, S_i, S_{i+1}. The saturation rate S_{sat} then used by subsequent QFind sessions is S_i. Several pilot studies informed our choice of 20% as sufficient to clearly indicate AP link saturation due to drops by the AP queue and not merely wireless errors. The same pilot studies also indicated using three consecutive periods with 20% loss, as fewer than three sometimes provided runs that merely had significant amounts of wireless loss, and more than three sometimes resulted in sending rates that overwhelmed the ability of the AP to forward packets from the incoming link to the outgoing wireless link.

Algorithm 1 depicts the algorithm to find the link saturation rate for a given AP.

Algorithm 1 Find AP Saturation Rate

1: $i \leftarrow 0$, $S_i \leftarrow 0$
2: **repeat**
3: $\quad i \leftarrow i + 1$
4: $\quad S_i \leftarrow S_{i-1} + 50$ packets/second
5: $\quad$ Send at rate S_i for 60 seconds
6: $\quad R_i \leftarrow$ received packet rate
7: $\quad p_i \leftarrow (1 - R_i/S_i)$
8: **until** $(p_{i-2} \geq .2)$ & $(p_{i-1} \geq .2)$ & $(p_i \geq .2)$
9: $S_{sat} \leftarrow S_{i-1}$

Clearly, the saturation rate obtained using Algorithm 1 depends upon the packet size used by the source. In general, smaller packets will require a higher packet rate to achieve link saturation. However, due to wireless link per-packet overhead, small-packet flows will yield lower throughput (measured in bytes per second) than flows sending larger packets. This study uses two packet sizes, 1500 and 750 bytes, to study the saturation rates for different packet sizes and, more pertinent to the goals of the paper, to investigate whether AP queue sizes are set on a per-packet or per-byte basis.

While the objective is to saturate the wireless link, the impact of high traffic loads on the sender and receiver have to be considered. High sending rates are sensitive to variance in measurements based on timing and throughput. The commercial desktop PCs used in the study have a system clock granularity of 10 ms. To saturate a 802.11g link with a theoretical capacity of 54 Mbps would require sending more than 3000 packets per second. At this sending rate, from Equation 4, the PC clock difference of 10 milliseconds can cause a measurement difference of approximately 30 packets. Compared to an AP queue of 50 or 100

packets (the default in the Linux 2.2 kernel), this 30 packet error is significant. Thus, to reduce timing measurement sensitivity, wireless link capacites were limited to 11 Mbps for all systems under test. For most APs this implies setting the AP to use only IEEE 802.11b, but some APs allow a strict upper bound on the rate limit. Reducing the link capacity from 54 Mbps to 11 Mbps reduces the impact of the clock granuality four-fold, and reduces sender and receiver load outliers.

2.3 Qfind Sessions

Once the saturation rate S_{sat} for a given AP is determined, a series of repeated QFind sessions (depicted in Figure 1) can be run to obtain a good estimate of the AP queue size.

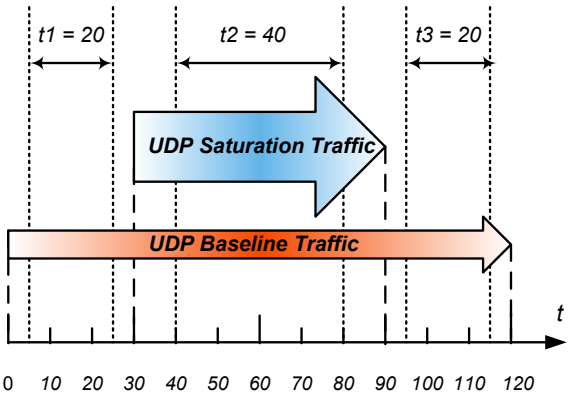

Fig. 1. QFind Session

At time $t = 0$, a low bitrate[2] UDP flow, labeled "UDP Baseline Traffic" in Figure 1, begins sending packets timestamped with the local PC time from the wired source through the wireless AP to a wireless receiver. Upon receiving the packet, the receiver computes the difference between the time stamp and its local time. Running by itself for 30 seconds, this UDP flow provides a baseline for the time difference that represents one-way delay without queueing.

At time $t = 30$ seconds, a second, concurrent UDP flow, labeled "UDP Saturation Traffic" in Figure 1, begins sending at the saturation rate rate S_{sat} from the sender to the receiver. The saturation flow runs for 60 seconds, providing time to fill up the AP queue. Similar to the baseline flow measurements, differences between receiver time and sender time for the saturation flow are recorded to provide delay information during queueing. The throughput and loss rate for the saturation flow is also computed.

[2] 10 packets/second

The baseline flow continues to run for an additional 30 seconds until time $t = 120$ seconds, at which time the session ends.

2.4 Calculations

The intricate part of QFind is the calculation of the queueing delay, D_q. While the QFind concept is to saturate the link to obtain the maximum AP queueing delay, one-way latencies on a wireless link also vary due to other events such as wireless channel errors and system events in the end hosts and wireless AP. This precludes simply using the largest observed delay as the maximum queueing delay and a more refined scheme for sampling from the delay distributions is needed to determine the values for D_h and D_l for Equation 2.

To obtain the delay samples for a sytem at steady state, the start and ending phases for the UDP flows are removed from the sampling range. Thus, as depicted in Figure 1, 40 seconds of time from t_1 and t_3 are used to compute the delay without queing, D_l, and 40 seconds of time at t_2 are used to compute the delay with queueing, D_h.

The baseline delay without queueing, D_l, is relatively straightforwardly computed as the median of the one-way delay during t_1 (time [5:25]) and t_3 (time [95:115]). Using both t_1 and t_3 measurements reduces the impact of differences in the clock drift rate at the sender and receiver while providing the same sampling interval (40 seconds) as that used to measure D_h.

However, obtaining the best estimate for D_h, is more complex. Four different schemes for determining the most accurate measure of D_h using the delay sample distributions from interval t_2 were considered, shown in Table 1. Since the objective is to characterize the delay for a full AP queue, both the 95th percentile and the 90th percentile delay values were considered. However, in validating the QFind result (Figure 2) these two metrics proved to be too sensitive to large outlier delays caused by system events other than a full queue. The mean of the delay samples during t_2 was also considered, as this may represent the typical full-queue delay. However, the mean was susceptible to the same outliers that plagued the 90th and 95th percentiles. Thus, the median of the delay samples during t_2 is used since it demonstrated the best robustness to system events while still providing the delay of a full queue.

Method	Details
Scheme 1	D_t = The 95th percentile of packet delay from Flow F_s during time t_2
Scheme 2	D_t = The 90th percentile of packet delay from Flow F_s during time t_2
Scheme 3	D_t = The median of packet delay from Flow F_s during time t_2
Scheme 4	D_t = The mean of packet delay from Flow F_s during time t_2

Table 1. Method to compute D_t

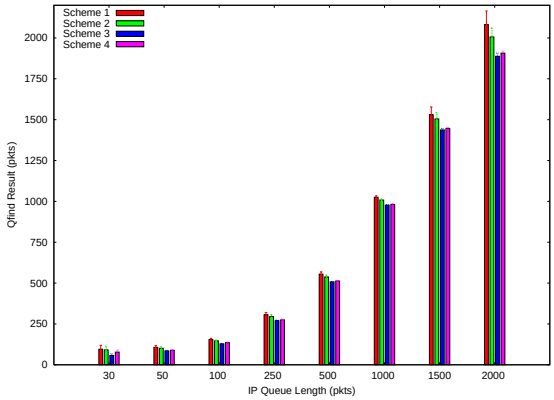

Fig. 2. QFind Validation Result with Different Compations

Q Len	Scheme1	Scheme2	Scheme3	Scheme4
1	53 ± 4.8	51 ± 4.6	39 ± 3.1	38 ± 0.6
30	96 ± 23.7	91 ± 21.8	57 ± 10.7	67 ± 0.6
50	107 ± 11.1	101 ± 9.0	85 ± 3.3	86 ± 0.9
100	154 ± 7.3	147 ± 4.8	128 ± 2.3	132 ± 1.1
250	306 ± 14.1	294 ± 11.9	270 ± 4.6	267 ± 4.8
500	554 ± 14.1	537 ± 11.7	507 ± 1.7	507 ± 3.5
1000	1025 ± 8.9	1008 ± 7.3	977 ± 3.8	977 ± 7.6
1500	1530 ± 47.8	1504 ± 38.2	1438 ± 9.4	1398 ± 21.22
2000	2082 ± 82.9	2006 ± 55.0	1888 ± 19.7	1773 ± 27.88

Table 2. QFind Validation Results with Different Computation Schemes

The throughput T is computed by multiplying the count of packets received during interval t_2 by the packet size and dividing by the time duration.

Algorithm 2 specifies a QFind session for a fixed packet size, L. To address the variability of experimental runs, the entire QFind method consisting of 30 QFind sessions where the mean and variance for computing confidence intervals is given in pseudocode in Algorithm 3.

3 Validation

To validate the QFind methodology, a testbed was setup where the wireless traffic and the AP queue itself were carefully controlled. In particular, the queue size of the AP could be both instrumented to measure queue occupancy and changed to different sizes, allowing testing of the robustness of QFind under different AP queue lengths.

3.1 Host AP

Our earlier efforts developed tools and techniques for wireless measurements [**?**] that include construction of a Host AP, namely, a PC turned into an IEEE

Algorithm 2 QFind Session

1: $t = 0$: start UDP baseline flow
2: $t = 30$: start UDP saturation flow at rate S_{sat}
3: $t = 90$: stop UDP saturation flow
4: $t = 120$: stop UDP baseline flow
5: $T \leftarrow$ throughput during t_2 ([40:80])
6: $D_h \leftarrow$ median delay for t_2 ([40:80])
7: $D_l \leftarrow$ median delay for t_1 ([5:25]) and t_3 ([95:115])
8: $q_p \leftarrow \frac{(D_h - D_l) \times T}{L}$

Algorithm 3 QFind Method

1: $S_{sat} \leftarrow FindSaturationRate()$
2: **for** $i = 1$ to 30 **do**
3: $\quad q_i \leftarrow QFindSession(S_{sat})$
4: $\quad$ Sleep for 10 seconds
5: **end for**
6: $q_{size} \leftarrow (\sum_{i=1}^{30} q_i)/(30)$

802.11 wireless access point using open source software and off-the-shelf wireless networking hardware. The general steps include (with specifics in parentheses):

1. Install the Linux operating system (SUSE Linux 9.2).
2. Integrate a prism[3] GT-based wireless network interface card (Netgear WG 511 version 1 PCMCIA) and update the driver (version 3.0).
3. Create an interface configuration file that handles the wireless interface and the wired interface.
4. Configure Dynamic Host Configuration Protocol (DHCP) to provide the addressing services common to most commercial APs.
5. Share the connection to the wired and wireless adaptors (using `iptables`[4]).

The Host AP allows end-host wireless clients that normally use a commercial wireless AP to transparently associate and use the Host AP. The Host AP Linux kernel is instrumented with a custom system call to a user level application that returns the current IP queue length of a given network device and a cumulative packet drop count for that queue. A Host AP user-level application records the queue size measurements using the custom system call via a timer that expires every 200 milliseconds.

The Host AP queue length is modified using *ifconfig*, the Unix system program that allows configuration of a network interface. Corresponding code in the Linux kernel source is under `linux/net/ethernet/eth.c` in `eth_setup()` to setup the transmission queue and `linux/netsched/sch_generic.c` `pfifo_fast_dequeue()` when a packet is dequeued.

[3] http://www.prism54.org/
[4] http://www.netfilter.org/

3.2 Validation Setup

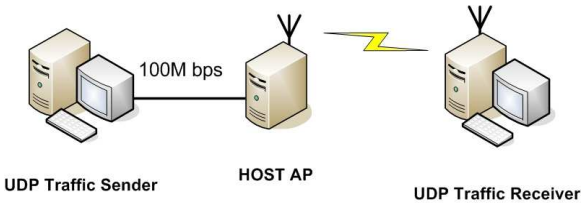

Fig. 3. Testbed Setup

Figure 3 shows the experimental testbed for validating Qfind with the Host AP. The UDP traffic sender is an Intel Celeron 750 MHz PC with a 100 Mbps wired Ethernet NIC running Linux SUSE 9.2. The Host AP is another Intel Celeron 750 MHz PC equipped with a prism GT-based wireless PCI NIC and a 100 Mbps wired NIC. The UDP traffic receiver is an Intel Pentium-4 2.8 GHz PC with a TI acx111 based wireless PCI NIC running Windows XP with Service Pack 2. The UDP traffic sender and the Host AP are connected via a Linksys 100 Mbps Ethernet hub. To avoid wireless throughput variations, the wireless client is placed 6 feet from the Host AP. The wireless capacity is restricted to 11 Mbps using `iwconfig`, and RTS/CTS is turned off.

The UDP traffic is generated using the Multi-GENerator toolset (MGEN)[5] using "precise on" option which attempts to accurately send packets at the user-specificed data rate. The MGEN UDP receiver creates a log file used for analysis of throughput, packet loss and delay after each experimental run. The validation experiments send 1500 byte packets to avoid Ethernet packet fragmentation.

3.3 Validation Results

These section analyzes results using the QFind methodology detailed in Section 2.

Figure 4 provides queue size results for two separate one session experiments where the Host AP queue length was set to 100 and 1000 packets[6].

The x-axis is time (in seconds) and the y-axis is queue size (in packets) reported by the Host AP. During the first 30 second period, only the baseline flow runs so there is almost no queueing. At 30 seconds, the saturation flow starts and the AP queue on the outgoing wireless link quickly fills up. The queue remains filled until the saturation flow stops at time 120 seconds where the AP queue quickly drains and returns to nearly zero. The trend in both curves is the

[5] http://pf.itd.nrl.navy.mil/mgen/

[6] Note, the chosen queue sizes reflect the default choices in recent Linux kernels v2.4 and v2.6, respectively.

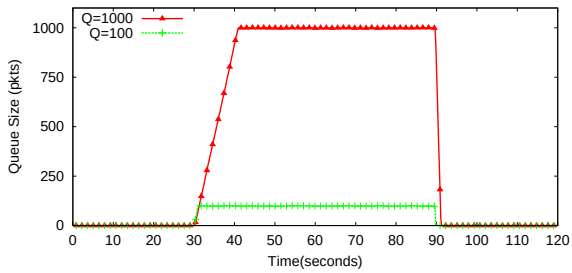

Fig. 4. Host AP with transmit queue length set to 100 and 1000 packets.

same with the only difference being the maximum queue size recorded when the saturation flow is running.

Having demonstrated the ability to control the Host AP queue size, the QFind method 3 was applied to the Host AP for a variety of AP queue settings to validate the Qfind method. The results of running QFind with Host AP queue sizes of 1, 30, 50, 100, 250, 500, 1000, 1500 and 2000 are graphed in Figure 5. The x-axis is the Host AP queue length setting and the y-axis is the queue size measured by QFind, with both axes shown in logscale. The bars represent the average over the 30 QFind sessions with 95% confidence intervals on top of the individual bar graphs. The numeric values are provided in Table 3. The fact that QFind generally reports accurate the actual queue size with tight confidence intervals validates the QFind approach.

Table 3. QFind Validation Results

Q Length (Actual)	Q Length (QFind)	Conf. Int. (95%)
1	39	±3.1
30	57	±10.7
50	85	±3.3
100	128	±2.3
250	270	±4.6
500	507	±1.6
1000	977	±3.8
1500	1438	±9.4
2000	1888	±19.7

Upon closer examination, for queue sizes up to 250 packets, QFind overestimates the Host AP queue size slightly. This is because the setting of the queue is done at the IP layer, while the wireless device driver provides its own, additional queuing [**?**]. Figure 6 diagrams the relationship between these different

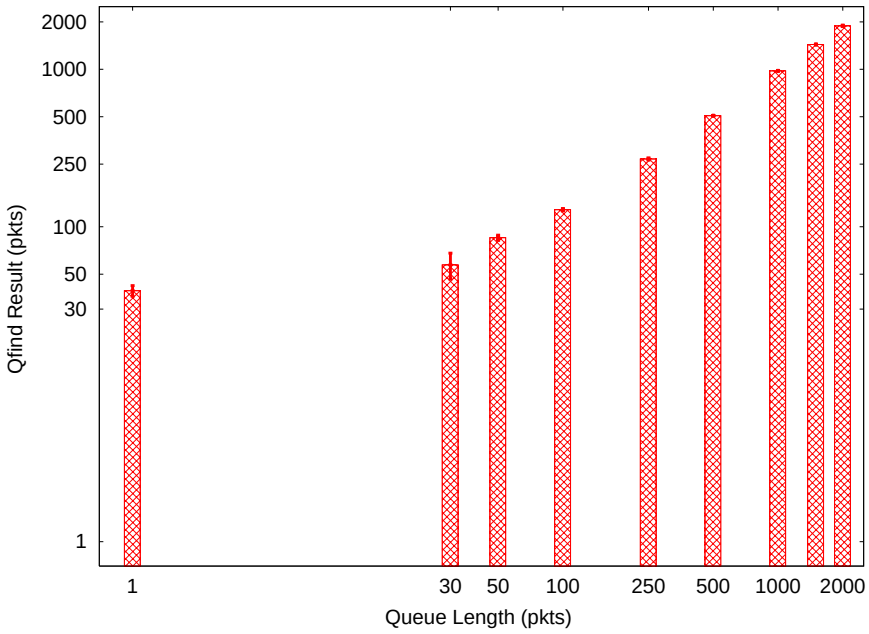

Fig. 5. QFind Validation Results (shown with 95% confidence intervals)

Linux packet transmission queues. From the QFind results with the Host AP queue set to 1, it appears that the wireless driver has a queue of about 40 frames. Note, an end-to-end application is only concerned about the total AP queue size and hence QFind does not need to differentiate the wireless driver queue from higher layer queues and only reports the total queue size.

For queue sizes over 1000 packets, QFind slightly underestimate the actual queue, due to the difficulty in keeping such a large queue consistently full. However, this underestimation is small and the actual APs measured in the next session do not have queues nearly this big.

4 AP Queue Measurement

To measure real AP queue sizes, a testbed configuration similiar to Figure 3 was used with the Host AP replaced by the specific real AP being evaluated. The APs were placed about 2 meters from the wireless UDP receiver to provide good radio transmission conditions. run in a wireless laboratory that was painted with DefendAir Radio Shield additive [7] to reduce the radio transmissions going through the walls. This was done to reduce possible interference between the test 802.11 APs and the general wireless users on the WPI campus. Since antenna

[7] http://www.forcefieldwireless.com/defendairadditive.html

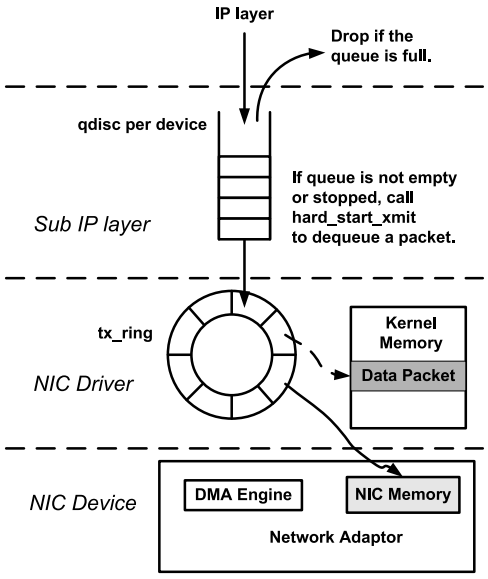

Fig. 6. Transmission of a packet in a Host AP, showing kernel queue and device driver queue.

orientation has been shown to be important for maximizing throughput [**?**,**?**], care was taken to position each AP antenna to provide good throughput to the wireless client.

Table 4 lists the details of the APs tested. There are two classes of APs available on the market - APs targeted for residential owners and those intended for business use by commercial industries. Home class APs typically cost less than 100 U.S. dollars, while commercial class APs typically cost over 300 U.S. dollars. Three APs from each class were selected from popular network vendors. Where possible, the same vendor was chosen for each class. However, there is no residential class AP provided by Cisco nor is there a commercial class AP provided by Linksys. Since Linksys was acquired by Cisco, these APs can perhaps be seen to represent APs from the same vendor. All APs selected were IEEE 802.11g compliant, with the exception of the Linksys BEFW11S4 which only supports IEEE 802.11b. The BEFW11S4 was selected to provide a case-study comparison of an older 802.11 product with a newer one. The BEFW11S4 was released in 2001 before 802.11g commercial products were readily available. Hereafter, the Linksys BEFW11S4 is called the "Old Linksys" AP.

During these queue measurements, all APs were rate limited to 11 Mbps. However, there is no single way to set rate limit for all the APs. For the Cisco and D-Link APs, the 11 Mbps rate limit is done by setting the maximum capacity to 11 Mbps, while for the Netgear and Linksys APs, the rate limit is done by

Class	Vendor	Model	Firmware
Commercial	Cisco	AIR-AP1121G	12.2
	D-Link	DWL-3200AP	1.00
	Netgear	PY3WAG302	1.0.3
Home	Linksys	WAP54G	v3
	D-Link	DI-524	1.00
	Netgear	PY3WGR61V5	1.0.3
	Linksys	BEFW11S4	v2

Table 4. Wireless APs Tested

putting the AP in IEEE 802.11b mode only. For all test cases, RTS/CTS is turned off.

The UDP traffic generator is an Intel Pentium-4 2.8 GHz PC with a Broadcom Net Xtreme 57xx Gigabit Controller. The UDP traffic receiver is an Intel Pentium M 1.7 GHz box with Intel(R) PRO/Wireless 2200BG network card running windows XP with Service Pack 2.

The UDP traffic is generated using MGEN as described in Section 3. For each AP, the QFind method 3) is run with the default IP packet size of 1500 bytes and the methodology is repeated with IP packets of size 750 to help determine if the individual AP queues are packet-based or byte-based.

5 AP Queue Measurement Results

5.1 Single Session

Figure 7 illustrates the throughput for one QFind session of the commercial Cisco AP and the commercial Netgear AP. For each AP, the throughput is low for the first and last 30 seconds when only the UDP baseline flow is running. At time 30 seconds, the UDP saturation flow begins whereupon the link throughput increases to a maximum, remaining at this maximum until time 90 seconds when the saturation flow stops. The throughputs in Figure 7 are fairly consistent, with only one dip in throughput for the Cisco AP. Notice the throughput differences during time 30-90 with the Cisco AP having an average throughput of about 8.2 Mbps and the Netgear AP having an average throughput of about 6.7 Mbps. This difference may be attributed to the Cisco AP using 802.11g with a maximum rate limit of 11 Mbps while the Netgear AP uses 802.11b.

Figure 8 depicts the delays for the same QFind session as in Figure 7. The delays recorded during the first and last 30 seconds are minimal since there is no queueing. At time 30, however, the UDP saturation flow quickly fills the queue and the delay reaches its maximum. Delay remains consistently near the maximum, corresponding to a consistently full queue, until the saturation flow stops and the queue drains, returning the delay to near 0. Notice how the delays for Netgear and Cisco APs are quite different, with the Cisco AP having a median delay of about 0.1 seconds and the Netgear AP having a median delay of about

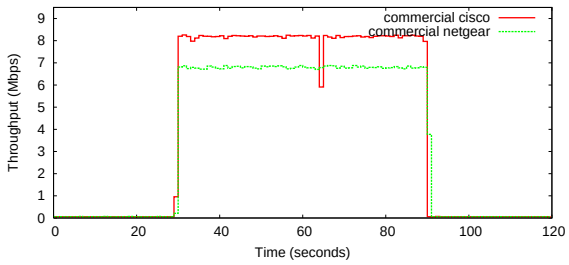

Fig. 7. Throughput during QFind Session

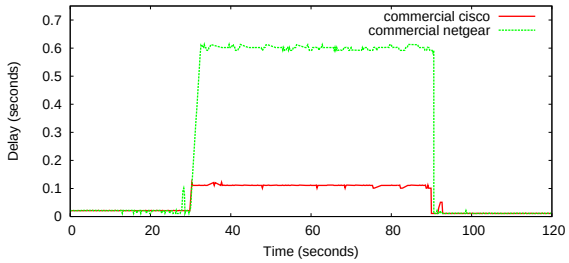

Fig. 8. Delay during QFind Session

0.6 seconds, suggesting different queue sizes, despite both being commercial class APs.

A fundamental QFind concept is to measure the delay when the queue is full compared to the delay when the queue is empty. Figure 9 shows the cumulative density functions (CDFs) of the delay distributions for the commercial Netgear session from Figure 8. When AP performance is consistent, the delays measured both with and without the saturation flow are consistent and delay values taken anywhere along either distribution yield the same queue size estimate. However, Figure 10 shows the delay CDFs for an old Linksys AP session that is less consistent. In this figure the period without the saturation traffic has consistent delay while the period with the saturation traffic produces a varying distribution. The maximum delay is nearly 300 milliseconds larger than the other delays, and their are several steps in the distribution towards the tail. This motivates the use of the median delay to limit the influence of outliers.

5.2 Multiple Sessions

The QFind methodology went smoothly for all the APs except for the commercial D-Link and the residential Netgear APs. For the residential Netgear AP, high offered load rates, sometimes even below saturation rate, cause the AP to become unstable in that there are numerous periods, sometimes as long as several

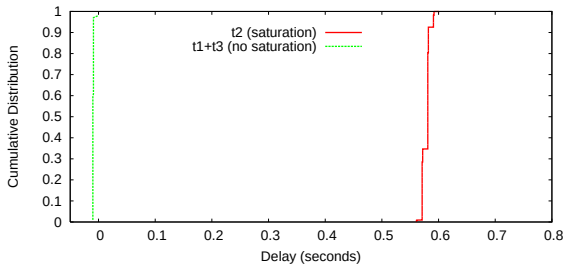

Fig. 9. CDF of Delay during QFind Session

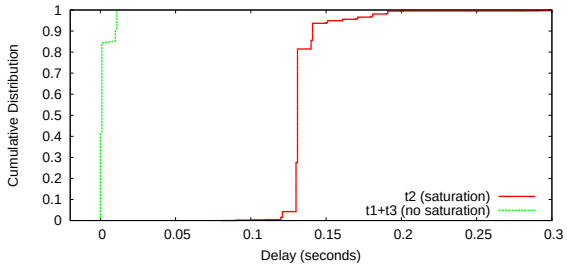

Fig. 10. CDF of Delay during QFind Session (inconsistent)

minutes, where little traffic makes it through the AP. For the commercial D-Link AP, data rates near saturation cause the AP to have inconsistent throughput, varying as much as 50% over the course of the 60 second UDP saturation flow. For both these APs, it is suspected that the difficulties arise when the offered load is greater than their packet forwarding capabilities. In applying the QFind method to the commercial D-Link AP, QFind sessions with extremely variable throughputs and loss rates are removed, and additional QFind sessions are run until 30 successful runs are obtained. In applying QFind to the residential Netgear AP, consistent results with throughputs near the saturation rate are not possible. Once the link is near saturation, the Netgear AP shuts down, stopping forwarding traffic from the wired to wireless link, often for several minutes. So, for the Netgear link only, the QFind results are estimated from data obtained before the AP shut down.

Figure 11 demonstrates the process of finding the saturation rate and the corresponding effect of offered traffic on throughput, loss rate and queuing delay for each AP. The x-axis for all graphs is the offered load (in Mbps). Each data point represents one run of Algorithm 1, with the delay also analyzed. All graphs are for a packet size of 1500 bytes, but the 750 byte tests show similar trends.

In Figure 11 (top), the throughput for each AP increases linearly with offered load until the AP approaches saturation, at which point the increase in throughput levels off. As noted earlier, the maximum throughputs differ considerable

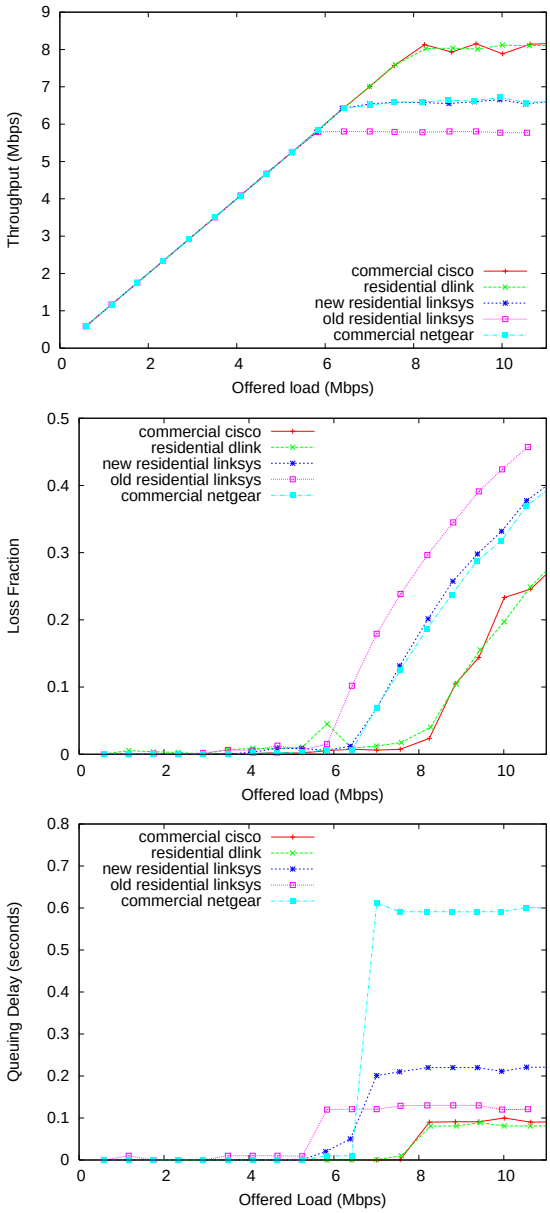

Fig. 11. Throughput, Loss and Delay versus Offered Load while Finding the Saturation Rate (1500 byte packets)

across APs. In Figure 11 (middle), the loss rates are negligible until the AP offered load nears saturation, at which time the loss rates increase linearly with offered load. As for loss, Figure 11 (bottom) shows the queuing delay is negligible until the link nears saturation, at which time the delay quickly increases to its maximum.

Table 5 gives AP data obtained during the QFind experiments, including saturation rates using Algorithm 1 and the throughputs and delays computed as in Algorithm 2 and averaged over the 30 QFind sessions using Algorithm 3. The QFind methodology was repeated for all APs using 750 byte packets in order to determine if APs have a queue size based on the total number of bytes (a byte queue) or have a queue size based on the total number of packets (a packet queue). The rates required to saturate the AP are higher for the smaller packets, but because of the per-packet overhead, the application layer throughput is lower for the smaller packets. From the table, since the delays are higher when the queues are filled with large packets than they are when the queues are filled with small packets, this suggests all the AP queues are per-packet.

Class	Access Point	Pk Size (bytes)	Satrtn (Pk/s)	Thrput (Mbps)	Delay (Sec)
Com.	Cisco	1500	1000	8.2	0.094
		750	1500	6.6	0.058
	D-Link	1500	900	7.1	0.406
		750	1050	5.5	0.331
	Netgear	1500	850	6.7	0.584
		750	1100	4.8	0.403
Home	Linksys	1500	850	6.6	0.217
		750	1150	5.0	0.138
	D-Link	1500	1050	8.1	0.083
		750	1500	6.6	0.048
	Netgear	1500	700	6.6	0.070
		750	1050	4.7	0.045
	old Lsys	1500	700	5.8	0.121
		750	900	3.9	0.089

Table 5. AP Performance Data

5.3 Queue Sizes

Figure 12 provides queue size results obtained using QFind (Algorithm 3), each average shown with a 95% confidence interval. Table 6 gives the numeric values. Despite some advantages that per-byte queues may offer to performance [**?**], the QFind results clearly demonstrate that all the APs use per-packet queuing. This is evident since the smaller packets provide approximately the same AP queue sizes as do the larger packets. If the APs used per-byte queuing, using the

QFind methodology with packets 1/2 the size would result in a queue that is approximately twice as large. In general, the queue sizes reported for the smaller packets are slightly smaller, but this can be attributed to the additional per-byte overhead at the wireless layer caused by more packets during saturation. The QFind results with with 1500 byte packets are probably slightly more accurate than the QFind results with 750 byte packets because the per-packet overheads cause some error in the measurements.

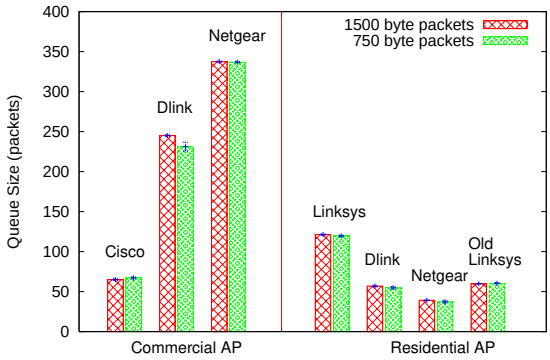

Fig. 12. Wireless AP Queue Sizes

Class	Access Point	Q Size (1500 B/Pkt)	Q Size (750 B/Pkt)
Com.	Cisco	65.0 ± 1.1	67.1 ± 1.3
	D-Link	245.2 ± 1.1	231.2 ± 5.7
	Netgear	337.5 ± 0.7	336.9 ± 0.8
Home	Linksys	121.3 ± 0.9	119.6 ± 1.0
	D-Link	56.9 ± 1.1	54.8 ± 1.5
	Netgear	39.2 ± 0.2	37.3 ± 1.6
	old Linksys	59.9 ± 0.3	60.4 ± 0.8

Table 6. Wireless AP Queue Sizes (± 95% confidence intervals).

Overall, the wireless AP queue sizes vary widely from the smallest (residential Netgear) of about 40 packets to the largest (commercial Netgear) of over 330 packets. Even within a class (residential or commercial), queue sizes vary considerably, with the residential Linksys AP queue (121 packets) being 3 times as large as the residential Netgear AP queue (39 packets) and the commercial

Netgear AP queue (337 packets) being 5 times as large as the commercial Cisco AP queue (65 packets). For Linksys and Netgear, their commercial APs have larger queue sizes than their residential APs, but for Cisco and Linksys (owned by Cisco), this relationship is reversed. The old Linksys AP has a smaller queue than the newer Linksys AP, possibly because the new AP supports IEEE 802.11g with higher capacities than the IEEE 802.11b that the old AP supports.

6 Discussion

The measured variability in queue sizes for different wireless APs begs the question: what size *should* access queues be?

With the steady decrease in the price of memory, it might be assumed that buffers should be large as over-buffering can reduce packet drops. However, in the presence of congestion large queues cause large delays that severely degrade interactive applications that have strict real-time constraints. For example, VoIP applications desire one-way delays of 150 milliseconds or less [**?**] and online first-person shooter games need round-trip delays to be under 100 milliseconds [**?**] to avoid performance degradations. During congestion, the commercial D-Link and Netgear APs can have almost 500 milliseconds of delay. Even for residential use, the maximum queuing delay of 200 milliseconds for the Linksys AP could be problematic for interactive applications.

A wireless AP is essentially a bridge, forwarding packets from a wired uplink downstream to a wireless downlink or forwarding packets between connected wireless clients. The conventional guideline is that a bridge needs queuing equal to the average round-trip time of a flow multiplied by the capacity of the bridge interface (the well known bandwidth-delay product, or $B = R\bar{T}T \times C$) [**?**]. This guideline comes from the goal of maximizing throughput by avoiding underflow of a queue during TCP's congestion control algorithm. A study of a large trace of TCP connections from an Internet backbone [**?**] shows TCP RTTs that traverse the Internet can vary from 10-1000 milliseconds with the median of the minimum[8] round-trip time about 200 milliseconds. While IEEE 802.11g links have a theoretical capacity of 54 Mbps, typical application level throughputs can be as low as a third of this capacity [**?**]. Putting this round-trip time and this bandwidth together in the guideline suggests queue sizes should be about 300 packets.[9] Of the APs tested, only the commercial Netgear and commercial D-Link APs are close to this queue size.

However, this queuing guideline assumes TCP sender and receiver windows can grow to reach the queue limit. If a TCP sender window size is limited to under 300 packets, then the queue sizes may not need to be so large. For actual TCP receiver window settings, older versions of Microsoft Windows have a default of 8192 bytes, Windows 2000 has a default of 17520 bytes, Linux has a default of 65535 bytes, and Windows XP may have a window size of 17520, but it also has

[8] Assuming the minimum round-trip time experienced by a connection represents the delay without queuing.

[9] 18 Mb/s $\times$ 0.2 seconds, assuming 1500 byte packets

a mostly undocumented ability to scale the receiver window size dynamically. Practically, this suggests AP queue sizes may not need to be larger than about 60 packets. Of the newer APs tested, the commercial Cisco and the D-Link residential AP meet this revised guideline.

If a wireless AP supports many flows, the queuing requirements of an AP may decrease. Appenzeller et al. [**?**] show that a queue should be approximately $B = R\bar{T}T \times C/\sqrt{n}$ (where n is the number of flows), as long as there are 250 or more flows [**?**]. A typical residential AP will not carry anywhere near 250 flows. However, a commercial AP deployed at a corporation or university may have a significant number of flows. Henderson et al. [**?**] reported the busiest APs service nearly 300 active clients on a wireless campus and Almeroth et al. [**?**] reported 200 to 500 users associated with a wireless AP during an IETF meeting. The number of flows these APs service would bring the queue requirement for the wireless AP down below 30 packets.

A queue needs to be big enough to accommodate bursts, such as during TCP congestion avoidance. However, bursts during slow start and even bursts during short-lived Web traffic are small relative to the queue dominance of long-lived flows [**?**].

If the wireless AP is considering UDP traffic, such as is present in some online games and streaming applications, and if this traffic is unresponsive to congestion and sends at a rate higher than the wireless link capacity, then the AP queue will fill to capacity no matter how large. The full queue will cause all responsive TCP traffic and interactive traffic to suffer from the full delay afforded by the queue for each packet. This reasoning suggests a wireless AP that serves unresponsive UDP traffic should have small queues.

It is quite possible that in an ideal case, a wireless AP would adapt the queue size to the current traffic conditions. A simple flow counting mechanism such as in [**?**] could provide a count of the number of flows, allowing adaptation of the queue size to flow count. Examining TCP ACK packets in response to data packets could be used to estimate round-trip times, allowing adaptation of the queue size to delay. Segregation of classes traffic could provide small queues for delay sensitive traffic (such as VoIP or online games) and larger queues for delay insensitive traffic.

Note that the queue sizes chosen for illustration in Section 3, 100 packets and 1000 packets, reflect default choices in recent Linux kernels. The default queue size in Linux kernels v2.4 is 100 packets, while the default queue size in Linux kernels v2.6 is 1000 packets. This has implications for users the make use of Linux in a PC acting as a Host AP, whether for functional or research purposes.

Regarding queue size and wireless AP class, performance is often not cited as a reason for choosing a commercial class AP over a residential class AP [**?**]. Commercial class APs have some features such as such as better security, centralized management, and rogue access point detection that may be of interest to administrator of heavily used wireless networks. However stability is frequently mentioned, making the instability of the commercial D-Link AP problematic.

7 Related Work

Our previous study [**?**] measured the queue sizes of DSL and Cable access networks. The intent was obtain a broad set of measurements from volunteers, so the emphasis was on non-invasive traffic generation to limit the impact of the measurements on the volunteers' ISPs. Thus, TCP bulk transfers were used to generate the traffic load and ICMP packets were used to obtain round-trip latencies. While this non-intrusive method did allow queue measurements for over 45 different access networks, it had limited accuracy since the maximum queue sizes reported were limited by the largest TCP window sizes and the ISP handling of ICMP packets. To measure wireless AP queue sizes, this paper uses the core ideas from [**?**], but with modifications that improved Qfind accuracy.

Pelletta and Velayos measure saturation thoughput for five different IEEE802.11b access points [**?**]. Their key contributions include a method to determine AP saturation throughput and results showing significant differences in maximum throughputs across APs. While maximum throughput is an import component of end-to-end performance, the AP queue sizes studied in our work are also important in effecting loss rates for responsive (TCP) flows and determining delay for real-time applications such as VoIP and online games.

Hirabaru applies a method similar to QFind to estimate the queue sizes for several wired switches and routers [**?**] and explores how the device queue size impacts TCP performance. He reports a wide range of queue sizes, from 50 packets to 15000 packets, with switches having much smaller queue sizes than routers. However, the method used to measure queue sizes assumes all loss is due to queue overflow, making it unsuitable for direct application to lossy wireless networks. Moreover, wired network capacities are quite different than wireless network capacities where determining network saturation isan important aspect of determining AP queue sizes.

Appenzeller et al. [**?**] argue that the "rule of thumb" that router buffer size should be equal to the bandwidth-delay product, is obsolete and incorrect for the backbone routers. Instead, backbone routers should have buffer sizes reduced by $\sqrt{n}$, where n is the number of flows traversing the router. These results hold for both short-lived (Web) and longer lived flows. While useful for practioners designing core routers, their results are only relevant for a large (greater than 250) number of flows and so do not apply to many wireless APs that carry far fewer flows.

8 Conclusions

The increased utilization of 802.11 APs demands a better understanding of wireless configurations that affect performance. One critical component to AP performance is the queue of packets waiting to be transmitted downstream. During traffic bursts or times of congestion from the higher-speed wired uplink, the AP queue can fill, dropping packets and causing increased delay. Despite the importance of the AP queue, there have been few measurements of actual AP queue

sizes. This has lead to AP designers implementing their own queue sizes and network researchers arbitrarily selecting queue sizes during experiments without knowledge of the queue sizes used in practice.

This paper presents QFind, a methodology to accurately measure the queue sizes of wireless APs. QFind uses a multi-step process to saturate a wireless access link and then determine delay from queuing, allowing a computation of the queue size. Validation with an controlled AP shows QFind accurately measures existing AP queue sizes. The QFind methodology is further applied to seven black-box wireless APs (three commercial and four residential class APs). The measurements show that AP queue sizes vary widely, differing as much across vendors as across classes. The largest AP queue sizes (about 300 packets) can be problematic for interactive applications during congestion and may not be needed for cases of a few or many TCP flows. The smallest AP queue sizes (about 50 packets) are likely sufficient for single TCP flows but further study needs to evaluate their performance under a range of traffic conditions.

These observations suggest future work to rigorously evaluate the performance of wireless APs with a variety of queue sizes under a variety of traffic conditions. This research could be conducted using a Host AP as done in the validation section of this paper, or by using a variety of commercial APs with queue sizes provided by this paper or determined by the QFind method presented in this paper.

References

1. G. Appenzeller, I. Keslassy, and N. McKeown. Sizing Router Buffers. In *Proceedings of ACM SIGCOMM*, Sept. 2004.
2. T. Beigbeder, R. Coughlan, C. Lusher, J. Plunkett, E. Agu, and M. Claypool. The Effects of Loss and Latency on User Performance in Unreal Tournament 2003. In *Proceedings of ACM NetGames*, Sept. 2004.
3. M. Claypool, R. Kinicki, M. Li, J. Nichols, and H. Wu. Inferring Queue Sizes in Access Networks by Active Measurement. In *Proceedings of the 5th Passive and Active Measurement Workshop (PAM)*, Antibes Juan-les-Pins, France, Apr. 2004.
4. S. Floyd. RED: Discussions of Byte and Packet Modes, 3 1997. With additional comments from January 1998 and October 2000.
5. T. Henderson, D. Kotz, and I. Abyzov. The Changing Usage of a Mature Campus-wide Wireless Network. In *Proceedings of MobiCom*, Philadelphia, PA, USA, Sept. 2004.
6. M. Hirabaru. Impact of Bottleneck Queue Size on TCP Protocols and Its Measurement. *IEICE Trans Inf. Syst*, E89-D(1), 2006.
7. International Telecommunications Union. One-Way Transmission Time. Technical Report G.114, ITU-T Recommendation, 1996.
8. S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley. Inferring TCP Connection Characteristics Through Passive Measurements. In *Proceedings of IEEE Infocom*, Hong Kong, Mar. 2004.
9. A. P. Jardosh, K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer. Understanding Congestion in IEEE 802.11b Wireless Networks. In *Proceedings of IMC*, Berkeley, CA, USA, Oct. 2005.

10. H. Jiang and C. Dovrolis. Source-Level IP Packet Bursts: Causes and Effects. In *Proceedings of IMC*, Miami, FL, USA, Oct. 2003.
11. F. Li, M. Li, R. Lu, H. Wu, M. Claypool, and R. Kinicki. Tools and Techniques for Measurement of IEEE 802.11 Wireless Networks. In *Proceedings of WiNMee*, Boston, MA, USA, Apr. 2006.
12. R. Morris. Scalable TCP Congestion Control. In *Proceedings of IEEE Infocom*, Tel-Aviv, Israel, Mar. 2000.
13. E. Pelletta and H. Velayos. Performance Measurements of the Saturation Throughput in IEEE 802.11 Access Points. In *Proceedings of WiOpt*, Apr. 2005.
14. M. Rio, T. Kelly, M. Goutelle, R. Hughes-Jones, and J.-P. Martin-Flatin. A Map of the Networking Code in Linux Kernel 2.4.20. Technical Report DataTAG-2004-1, DataTag project (IST-2001-32459), Mar. 2004.
15. P. Rubens. Your Next WAP: Hold the Cheese? Wi-Fi Planet, Oct. 2005. Online at: http://www.wi-fiplanet.com/tutorials/article.php/3556226.
16. C. Villamizar and C. Song. High Performace TCP in ANSNET. *ACM Computer Communications Review*, 24(5):45–60, 1994.
17. A. L. Wijesinha, Y. tae Song, M. Krishnan, V. Mathur, J. Ahn, and V. Shyamasundar. Throughput Measurement for UDP Traffic in an IEEE 802.11g WLAN. In *Proceedings of SNPD/SAWN*, 05 2005.
18. M. Yarvis, K. Papagiannaki, and W. S. Conner. Characterization of 802.11 Wireless Networks in the Home. In *Proceedings of WiNMee*, Riva del Garda, Italy, Apr. 2005.