

Measuring Queue Capacities of IEEE 802.11 Wireless Access Points

Feng Li, Mingzhe Li, Rui Lu, Huahui Wu, Mark Claypool and Robert Kinicki
Computer Science Department at Worcester Polytechnic Institute
Worcester, MA, 01609, USA
{lif, lmz, kkboy, flashine, claypool, rek}@cs.wpi.edu

Abstract—While queue capacities have a direct impact on loss and latency during congestion, and wireless networks continue to spread in university, corporate and home networks, little is publicly known about the queue capacities of wireless access points (APs). This paper presents and deploys the Access Point Queue (APQ) methodology for externally estimating the queue capacity for a wireless AP. APQ determines the AP saturation point, measures the baseline delay, induces the saturation rate to measure the delay with a full AP queue and computes the queue capacity. APQ is deployed to determine the queue capacities of three commercial class and four residential class APs. The wireless AP queue capacities are shown to be packet-based and to range from 50 packets to over 350 packets. The fact that queue capacities vary so much among devices targeted for the same network configuration suggests future work to determine the most appropriate queue capacity.

I. INTRODUCTION

The performance bottleneck for many Internet flows is the last hop from a wireless access point (AP) to an application on a wireless end host. With wired IEEE 802.11 AP uplink capacities greater than or equal to 100 Mbps and wireless IEEE 802.11g downlink capacities of 54 Mbps, the wireless AP can often become the bottleneck for downstream Internet traffic. Since most Internet traffic is bursty [10], the designer's choice for wireless AP queue capacity can impact a flow's achievable throughput and delay. However, since little has been published on the best choice for AP queue capacity, AP queue capacities used in practice remain unknown.

If the AP queue capacity is too small, a flow's throughput can be significantly below the available bandwidth of the wireless link. However, simply over-provisioning the AP with a large queue can negatively impact a flow's end-to-end delay. Some interactive applications, such as IP telephony and network games with strict delay bounds in the hundreds of milliseconds range, experience degraded Quality of Service when a large AP queue saturates due to high concurrent flows with a high bitrate that saturate the queue.

Actual queue measurement and analysis has focused on queue capacities of wired devices, such as queues for network switches [6], DSL or Cable modems [3], or core Internet routers [1]. General guidelines for determining the "best" queue capacities have often been debated on the e2e mailing

list,¹ an active forum for network related discussion by researchers and practitioners. Consensus suggests the reasonable range for queue capacity is from one to four times bandwidth-delay product. Unfortunately, since wireless data rates vary with connection quality and round-trip time delays vary by at least two orders of magnitude (10 ms to 1 second) [8], the consensus queue range provides little guidance when selecting a queue capacity for wireless APs.

In a previous study [3], the QFind scheme was used to non-intrusively estimate queue capacities over 45 different access networks. While effective for providing broad participation of end-users, QFind's accuracy was hampered by using TCP bulk transfers and ICMP packets that caused the results to be affected by the size of the TCP windows and the specifics of ICMP packet handling by individual ISPs.

Applying a technique similar to QFind, Hirabaru [6] estimated queue capacities for several wired switches and routers and explored the impact of queue capacity on TCP performance. A wide range of queue capacities were reported, ranging from 50 to 15000 packets with switches using smaller queue capacities than routers. The queue capacities were also shown to have an impact on the throughput of TCP traffic.

This paper explains in detail the Access Point Queue (APQ) method for providing black-box measurements of commercial AP queue capacities and provides a measurement study of queue capacities for seven IEEE 802.11 access points. Upon determination of an AP's saturation throughput rate, APQ sends high bitrate traffic from a wired source over the wireless downlink to fill the AP queue. Careful measurement of the added queuing delay and achievable throughput provide an accurate determination of queue capacity. APQ is applied to seven different APs in two different classes, commercial and residential. The results show that AP queue capacities are packet-based and vary considerably both across and within the AP classes. This suggests that either the individual AP designs are based on differing, but unstated, traffic assumptions or that the AP queue capacity is not viewed as a critical factor in wireless LAN performance, a view not supported by measured performance.

The paper is organized as follows: Sections II and III explain and validate the APQ method, respectively; Section IV

¹In 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>.

provides details on the experimental AP queue measurements and results; Section V discusses the implications of the results; and Section VI presents conclusions and possible future work.

II. ACCESS POINT QUEUE (APQ) METHOD

The basic steps in the Access Point Queue technique for measuring wireless AP queue capacities are to fill the AP queue, to measure the queuing delay and to compute the queue capacity using the measured delay and throughput. Let D_h be the delay measured with a full queue, D_l be the delay measured with an empty queue and T be the saturation throughput measured by sending packets of size s (including IP headers). Then the queue capacity in packets, q_p , is:

$$q_p = \frac{(D_h - D_l) \times T}{s} \quad (1)$$

While QFind [3] utilized the above equation, it did not incorporate a repeatable methodology that provides reliable measurements of D_h over a variety of configurations. Hence, a major contribution of APQ is the Saturation Rate algorithm designed to ensure that the UDP flow used in an APQ session will saturate the AP queue.

To fill the AP queue, the downlink offered load must be greater than the effective wireless link capacity. Since prior research [14] has shown that, due to specific 802.11b AP implementation variances, effective wireless capacity can vary from approximately 5 to 7 Mbps, the first step of the APQ scheme to determine the traffic load that saturates a specific AP wireless link.

Starting from a sending rate of $S_0 = 50$ packets/sec, the APQ saturation rate algorithm (given in Algorithm 1) increases the sending rate by 50 packets/sec until packet loss rates of 20% or more are observed for three successive sending rates, S_{i-1}, S_i, S_{i+1} . The saturation rate, S_{sat} , is set to S_i for subsequent APQ sessions. The key to choosing the 20% loss as a saturation indicator was finding a metric that indicated that for a given offered load, the throughput result was beyond the ‘knee’ of the throughput curve and at the front edge of the throughput plateau. After numerous pilot studies, the concept of three consecutive high loss rates was chosen to avoid choosing a point below saturation due to spurious wireless errors. Moreover, the pilot studies indicated not using more than three consecutive sending rates with 20% loss because super-saturated rates can overwhelm the AP’s ability to forward packets onto the outgoing wireless link.

Algorithm 1 AP Saturation Rate Algorithm

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

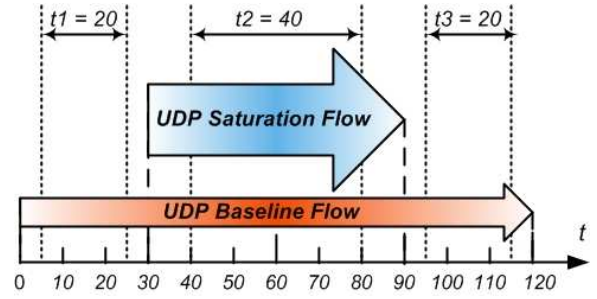


Fig. 1. APQ Session

Two distinct packet sizes, 750 and 1500 bytes, were used to study AP saturation rates to account for the dependency of saturation rate on packet size, but more importantly to determine if the AP queues are managed on a per-packet or per-byte basis. Once S_{sat} for a given AP and packet size is determined, a series of APQ sessions (shown in Algorithm 2) is run to estimate the AP queue capacity. Each APQ session is repeated 30 times to address experimental run variability. APQ queue capacity, $q_{capacity}$, is the mean of the 30 q_p measurements.

Algorithm 2 APQ 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: $q_p \leftarrow \frac{(D_h - D_l) \times T}{L}$
-

At time $t = 0$, the UDP baseline flow (a low bitrate, 10 packets/second CBR flow) 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 the receiver’s local time. Running alone for 30 seconds, this UDP flow provides a baseline for measuring one-way delay with no AP queuing.

At time $t = 30$ seconds, the UDP saturation flow, a concurrent CBR flow, begins sending packets at the saturation rate, S_{sat} , from the sender to the receiver. The saturation flow runs for 60 seconds to account for variability in the AP queue management. Similar to the baseline measurements, differences between receiver time and sender time for the saturation flow are recorded to provide delay information when the AP queue is full. Throughput and packet loss rate for the saturation flow are also computed. After the saturation flow stops, the baseline flow continues for an additional 30 seconds $t = 120$ seconds whereupon the APQ session ends.

The subtle part of the APQ technique involves the delay sample filtering used to estimate queuing delay, D_q , when the

AP queue is full. While the APQ concept is to saturate the wireless link to obtain the maximum AP queuing delay, one-way latencies on a wireless link vary with wireless channel error rates and system events at the end hosts and wireless AP. This precludes simply using the largest observed delay as the maximum queuing delay. Hence, a more refined technique is needed for sampling the delay distributions to determine reliable values for D_h and D_l in Equation 1.

To obtain delay samples for a system in steady state, the starting and ending phases for the UDP flows are removed from the sampling range. As shown in Algorithm 2, 40 seconds of time from t_1 and t_3 are used to compute the delay without queuing, D_l , and 40 seconds of time at t_2 are used to compute the delay with queuing, D_h . The baseline delay without queuing, D_l , is straightforwardly defined as the median of the one-way delay during t_1 (time [5:25]) and t_3 (time [95:115]). Both t_1 and t_3 samples are used to reduce the impact of differences in clock drift rate at the sender and receiver while providing the same 40-second sample interval duration used to measure D_h . The median of the delay samples during t_2 is used for D_h to provide robustness in the face of spurious system events while still measuring the delay of a full queue. Section IV-B shows evidence supporting the choice of median for the delay samples. 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.

Note that based on the IEEE 802.11 standard, when the AP has more than one packet to send (ie- when measuring D_h) the AP sets a random backoff timer uniformly distributed over the range 0 to 31 slot times. The backoff timer decreases by one when the media is idle for a slot time (20 microseconds) and is frozen when the channel is sensed busy. When the backoff timer reaches zero, the station sends the frame. This backoff mechanism is not used when the AP transmits a new packet (ie- when measuring D_l). When measuring D_h , this results in an additional average delay 310 microseconds per frame that is not included when measuring D_l . However, since this additional delay is small relative to the size of the D_h measurements, it can effectively be ignored without compromising the accuracy of the queue measurement, q_p .

III. APQ VALIDATION

Utilizing previously developed tools and techniques for wireless measurements [11], a testbed that includes a Host AP was set up to validate the APQ methodology. Building a Host AP (i.e., a PC that provides 802.11 AP functionality) from open source software and off-the-shelf wireless networking hardware provides the ability to vary the Host AP queue capacity and measure queue occupancy levels.

Figure 2 shows the experimental testbed for validating APQ 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

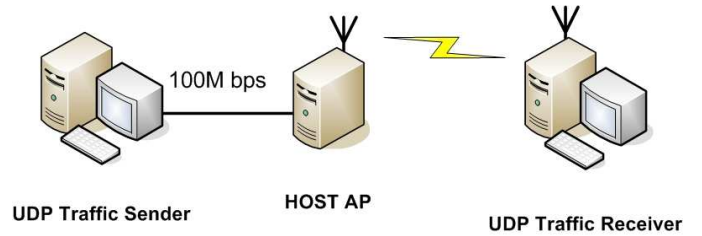


Fig. 2. Testbed Setup

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. The wireless capacity is fixed at 11 Mbps, RTS/CTS is disabled and the wireless client is 6 feet from the Host AP to minimize throughput variations.

The UDP traffic is generated using the Multi-GENerator toolset (MGEN)² using the ‘precise on’ option which attempts to accurately send packets at the user-specified data rate. MGEN creates a log file on the UDP receiver after each experimental run that is used to determine throughput, packet loss and delay. 1500-byte packets are sent during validation experiments to avoid Ethernet packet fragmentation.

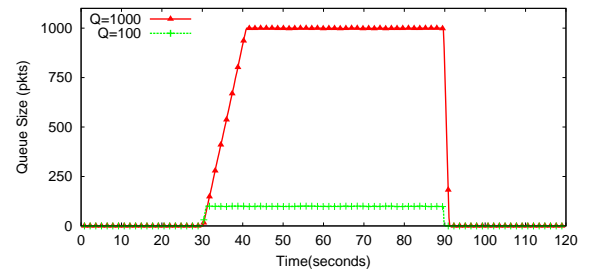


Fig. 3. Host AP with Transmit Queue Capacity Set to 100 and 1000 Packets.

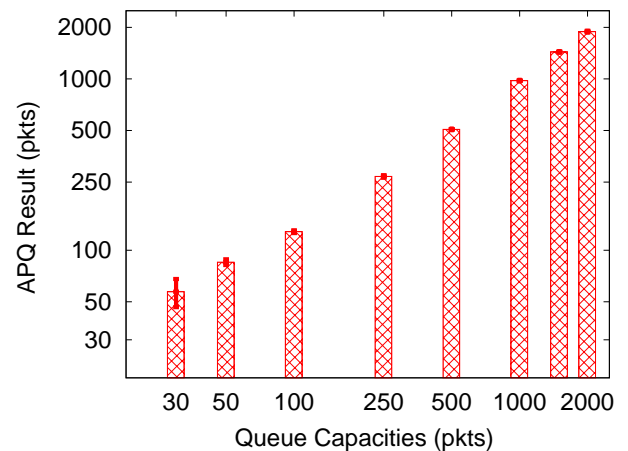


Fig. 4. APQ Validation Results (shown with 95% confidence intervals)

²<http://pf.itd.nrl.navy.mil/mgen/>

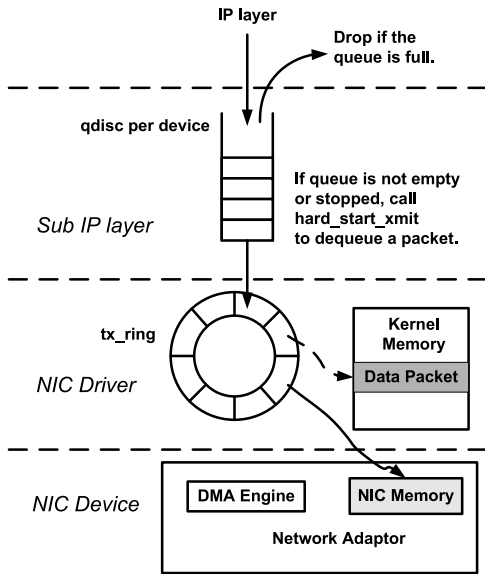


Fig. 5. Transmission of a Packet in a Host AP, Showing Kernel Queue and Device Driver Queue.

Figure 3 provides queue capacity results for two separate, one session experiments where the Host AP queue length was set to 100 and then 1000 packets.³

The x-axis is time (in seconds) and the y-axis is queue capacity (in packets) reported by the Host AP. During the first 30 second period, only the baseline flow runs so there is almost no queuing. 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 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 capacity, a range of AP queue capacities were used to validate the APQ method. The results of APQ validation runs on the testbed with Host AP queue capacities of 30, 50, 100, 250, 500, 1000, 1500 and 2000 packets are presented in Figure 4. The x-axis is the Host AP queue capacity setting and the y-axis is the queue capacity measured by APQ. Both axes use logscale. The bars represent the average queue capacity over 30 APQ sessions with 95% confidence intervals provided on the top of the individual bar graphs. Numeric values are provided in [12]. Generally, APQ accurately estimates the actual Host AP queue capacities with tight confidence intervals.

Closer examination of the numeric values reveals that for queue capacities up to 250 packets, APQ over-estimates the Host AP queue capacity slightly. This is attributed to using the IP layer queue setting in the Host AP while the AP wireless device driver provides an additional queue for frames [15].

³Note, the chosen queue capacities reflect the default choices in recent Linux kernels v2.4 and v2.6, respectively.

Figure 5 diagrams the relationship between these different Linux packet transmission queues. The APQ results with the IP queue for the Host AP set to 1 suggest that the wireless driver maintains a queue of approximately 40 frames. Note, since the end-to-end application’s main concern is the impact of maximum AP delay on QoS, APQ need not differentiate the wireless layer queue from higher layer queues. Thus, APQ estimates the sum of the AP queues.

For queue capacities above 1000 packets, APQ slightly under-estimates the Host AP queue capacity. This is attributed to difficulty in keeping such a large queue consistently full. However, this under-estimation is small and none of the real APs measured in the next section have queue capacities nearly this large.

IV. AP QUEUE MEASUREMENTS

A. Experimental Setup

APQ was used to measure queue capacities for the seven wireless APs in Table I on a testbed similar to Figure 2 where the evaluated AP replaces the Host AP. For the APQ measurements, the UDP traffic sender is an Intel Pentium-4 2.8 GHz PC with a Broadcom Net Xtreme 57xx Gigabit Controller and 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. Table I provides model specifics for the three APs classified as intended for residential users and the four APs intended for commercial use in industrial settings.

For all APQ measurement runs, the AP data rate is fixed at 11 Mbps, RTS/CTS is disabled and the AP antenna is positioned carefully to provide good throughput in response to studies showing the importance of antenna orientation in maximizing throughput [14], [19]. As described in Section III, MGEN generates the UDP traffic for all the experiments. The APQ method was run twice for each AP, once with 1500-byte IP packets and once with 750-byte IP packets to determine if the individual AP queues are packet-based or byte-based.

| 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 I
WIRELESS APs TESTED

B. Results

Figure 6 provides a throughput comparison of one APQ session for the Cisco commercial AP and one APQ session for the Netgear commercial AP. In both cases, 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 wireless link throughput increases to a maximum and hovers around this maximum until time 90 seconds when the saturation flow stops. Except for one throughput dip for the Cisco AP, Figure 6 demonstrates that both throughput profiles are fairly consistent. The figure clearly shows the higher throughput of the Cisco AP during the 30-90 second interval. The higher average Cisco AP throughput of 8.2 Mbps versus the Netgear AP's average throughput of 6.7 Mbps can be attributed to the fact that while both APs are fixed at 11 Mbps, the Cisco AP uses 802.11g and the Netgear AP uses 802.11b.

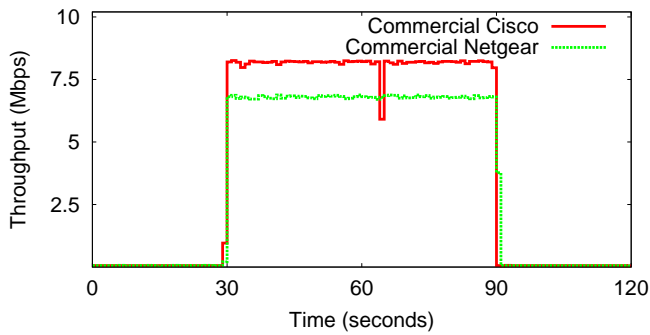


Fig. 6. Throughput during APQ Session

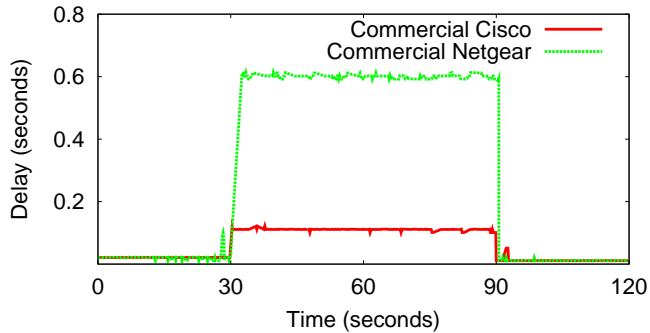


Fig. 7. Delay during APQ Session

Figure 7 depicts delay measurements for the same APQ sessions as in Figure 6. The recorded delays during the first and last 30 seconds are minimal since there is no queuing. At time 30, however, the UDP saturation flow quickly fills the queue and 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. Note that Netgear and Cisco AP measured delays are different. The approximate half second difference between 0.584 second Netgear delays and the 0.094 second Cisco delays clearly suggests a significant difference in queue capacities and design philosophies for two access points intended for the same market.

A fundamental APQ concept is to measure the delay when the queue is full compared to the delay when the queue is empty. Figure 8 shows the cumulative density functions (CDFs) of the delay distributions for the commercial Netgear

session from Figure 7. When the 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 capacity estimate. However, Figure 9 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 delay distribution. The maximum delay is nearly 300 milliseconds larger than the other delays with several steps in the distribution towards the tail. This motivates the use of the median delay to limit the influence of outliers.

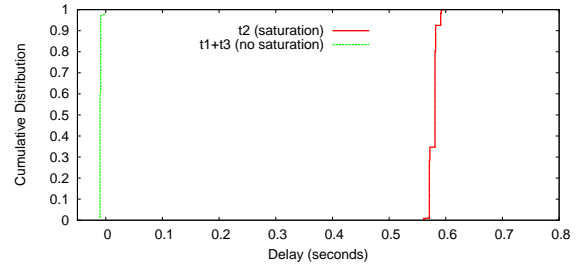


Fig. 8. CDF of Delay during APQ Session

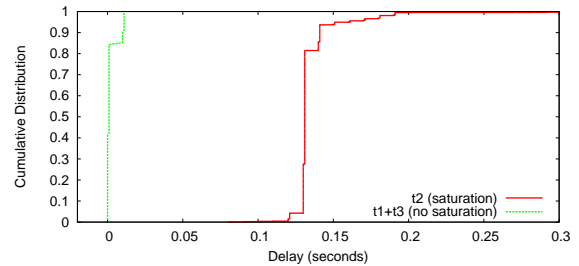


Fig. 9. CDF of Delay during APQ Session (inconsistent)

The APQ methodology went smoothly for five of the APs but problems were encountered with the Netgear residential AP and the D-Link commercial AP. During the Netgear residential AP measurements, high sending rates, even below the saturation rate, would sometimes yield unstable AP behavior evidenced by numerous periods, some lasting for several minutes, where little downlink traffic passed through the AP to the client. During the D-Link commercial AP measurements, near-saturation sending rates produced inconsistent AP throughputs varying as much as 50% during the 60 seconds of the UDP saturation flow. In both cases, strange AP behavior is attributed to offered load exceeding the AP's capability to forward packets. In the D-Link situation, measurements from APQ sessions with extremely variable throughputs and loss rates were discarded and additional APQ sessions were run to accumulate 30 acceptable runs. Unfortunately, once the Netgear AP shuts down there are no reasonable remedies. Hence, for the Netgear residential queue capacities are based only on data obtained prior to when the AP shuts down.

Figure 10 demonstrates the process of finding the saturation rate and the effect of the corresponding offered load for 1500-

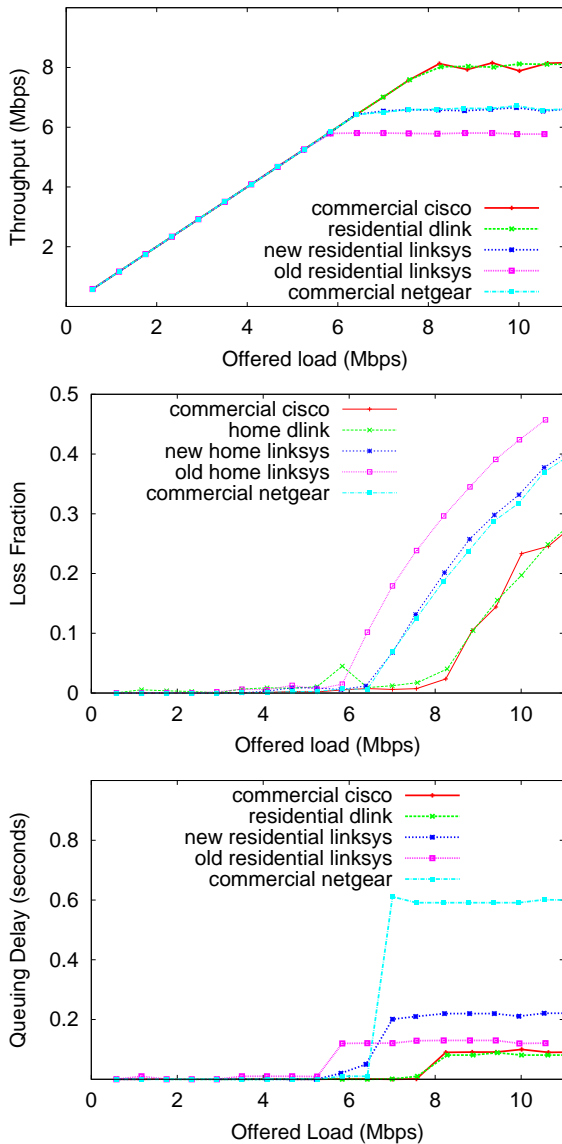


Fig. 10. Throughput, Load and Delay versus Offered Load while Finding the Saturation Rate (1500 byte packets)

byte packets on throughput, loss rate and queuing delay for the five APs that remain stable at the saturation point. The x-axis for all graphs is the offered load (in Mbps). Each data point represents one run of Algorithm 1 that also includes delay analysis. The results for the 750-byte experiments are not shown because the throughput and delay trends are similar.

In Figure 10 (top), the throughput for each AP increases linearly with offered load until the AP approaches saturation, at which point the throughput levels off. Note, maximum throughput varies across the five APs. Figure 10 (middle) shows negligible loss rates until the offered load nears saturation, whereupon the loss rates increase linearly with offered load. Similar to the loss behavior, Figure 10 (bottom) indicates the queuing delay is negligible until the AP wireless link nears saturation, at which point the delay quickly increases to its

maximum.

C. Queue Capacities

Figure 11 provides queue capacity bar graphs with 95% confidence intervals for both 750-byte and 1500-byte packet APQ measurements for the seven access points tested. Table II gives the numeric values for the seven APs. Although some researchers [4] have considered the performance advantages offered by per-byte queues, the APQ queue capacity results given in Figure 11 provide clear evidence that all seven APs manage their queues on a per-packet basis. If per-byte queuing were employed at an AP, the 750-byte APQ sessions would result in bar graphs twice as large as the 1500-byte APQ sessions in the figure. However, the nearly equal queue capacity measurements for differing packet sizes implies all the AP queues are packet-based. While the queue capacities reported by APQ for the smaller packet sessions are, in general, slightly smaller, this is attributed to additional per-packet AP overhead resulting from processing twice as many packets at the smaller packet saturation rate. This argument also suggests that the APQ measurements with 1500-byte packets are slightly more accurate.

The first observation from Table II is that the queue capacity range for the seven APs tested varies widely from 40 to 338 packets and that the Netgear APs occupy both endpoints. Clearly, Netgear and D-link believe a commercial AP requires a larger AP queue than a residential class AP. Within the two classes, the residential new Linksys AP (121 packet queue) and the Cisco commercial AP (65 packet queue) both appear to be out-of-step relative to the queue capacities of their competitors. Note, the new Linksys AP is double the capacity of the old Linksys AP. This can partially be attributed to the new Linksys AP supporting the higher capacities of 802.11g versus 802.11b for the old Linksys AP. However, it may also be that performance analysis of the old Linksys AP caused the significant change in the new Linksys AP queue capacity.

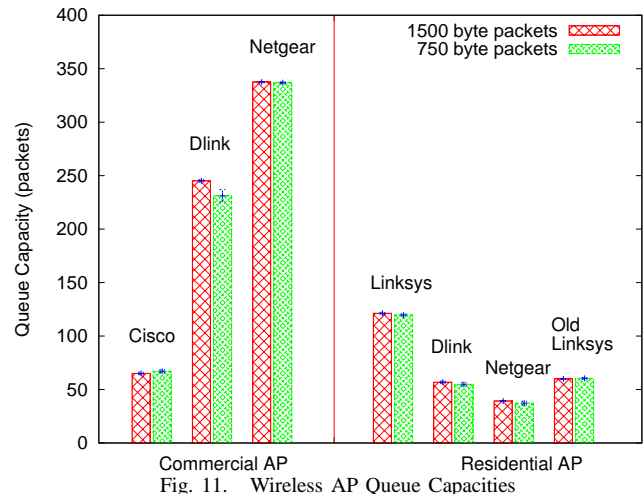


Fig. 11. Wireless AP Queue Capacities

| Class | Access Point | Q Capacity (1500 B/Pkt) | Q Capacity (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 |
| Residential | Netgear | 39.2 ± 0.2 | 37.3 ± 1.6 |
| | D-Link | 56.9 ± 1.1 | 54.8 ± 1.5 |
| | old Linksys | 59.9 ± 0.3 | 60.4 ± 0.8 |
| | Linksys | 121.3 ± 0.9 | 119.6 ± 1.0 |

TABLE II

WIRELESS AP QUEUE CAPACITIES ($\pm 95\%$ CONFIDENCE INTERVALS).

V. DISCUSSION

The measured range in queue capacities for different wireless APs, even for APs in the same class, begs the question: what capacity *should* access queues be?

With the steady decrease in the price of memory, one might assume that simply by over-provisioning APs with large queue capacities will improve performance by reducing packet drops at the AP. However, in the presence of congestion, large queue capacities will produce large average delay times in the AP queue that severely degrade interactive applications with strict real-time constraints. For example, VoIP applications desire one-way delays of 150 milliseconds or less [7] and online first-person shooter games need round-trip delays to be under 100 milliseconds [2] to avoid performance degradation. During congestion, the commercial D-Link and Netgear APs recorded almost 500 milliseconds of delay. Moreover, in a home environment with the Linksys AP, the maximum queuing delay of 200 milliseconds is problematic for interactive applications.

A wireless AP in some sense acts as a router, forwarding packets from a wired uplink downstream to a wireless downlink or forwarding packets between connected wireless clients. The conventional guideline is that a router needs queuing equal to the average round-trip time of a flow multiplied by the capacity of the router interface (the well known bandwidth-delay product, or $B = R\bar{T}T \times C$) [17]. 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 [8] shows TCP round-trip times that traverse the Internet can vary from 10-1000 milliseconds with the median of the minimum⁴ 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 [18]. Putting this round-trip time and this bandwidth together in the guideline suggests queue capacities should be about 300 packets.⁵ Of the APs tested, only the commercial Netgear and commercial D-Link APs are close to this queue capacity.

However, this queuing guideline assumes TCP sender and

⁴Assuming the minimum round-trip time experienced by a connection represents the delay without queuing.

⁵18 Mb/s \times 0.2 seconds, assuming 1500 byte packets.

receiver windows can grow to reach the queue limit. If a TCP sender window size is limited to under 300 packets, then the queue capacities 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 a mostly undocumented ability to scale the receiver window size dynamically. Practically, this suggests AP queue capacities may not need to be larger than about 60 packets. Of the newer APs tested, the commercial Cisco and the D-Link home AP meet this revised guideline.

If a wireless AP supports many flows, the queuing requirements of an AP may decrease. Appenzeller et al. [1] 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 [1]. A typical home 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. [5] reported the busiest APs service nearly 300 active clients on a wireless campus and Almeroth et al. [9] reported 200 to 500 users associated with a wireless AP during an IETF meeting. The number of flows served by these APs brings the queue requirement for the wireless AP down below 30 packets.

Queue capacity needs to accommodate packet 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 [1].

If the wireless AP is handling UDP traffic, such as found in some online games and streaming applications, that is unresponsive to congestion and sends at a rate higher than the wireless link capacity, then the AP queue will fill to capacity regardless of the queue capacity. 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 possible that in an ideal case, a wireless AP would adapt queue capacity to current traffic conditions. Implementing a simple flow counting mechanism such as in [13] at the AP would enable the AP to adjust the queue capacity based on the flow count. Examining TCP ACK packets at the AP in response to data packets could be used to estimate round-trip times and permit adjusting queue capacity with changes in delay estimates. Segregating traffic into queue classes as proposed in 802.11e 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 capacities chosen for illustration in Section III, 100 packets and 1000 packets, reflect default choices in recent Linux kernels. The default queue capacity in Linux kernels v2.4 is 100 packets, while the default queue capacity in Linux kernels v2.6 is 1000 packets. This has implications for users that utilize Linux in a PC acting as a Host AP, whether for functional or research purposes.

Finally, performance is often not cited as a reason for

choosing a commercial class AP over a home class AP [16]. Commercial class APs have features such as better security, centralized management, and rogue access point detection that are important to administrators of heavily used wireless networks. However stability is frequently mentioned as important. This implies the instability of two of the commercial D-Link AP evaluated in this study represents a serious problem.

VI. CONCLUSIONS

The rapid growth of infrastructure wireless local area networks (WLANs) as the last hop for both commercial and residential Internet applications puts the spotlight on key design decisions for a variety IEEE 802.11 access points. Moreover, the secretive nature of the major AP providers with respect to internal designs that they consider proprietary has stifled active performance analysis of commercial AP products.

Given the wide variety of user applications that reside on a wireless client and send requests through an infrastructure AP to an Internet server, it is clear that it is a difficult task for an AP to provide good QoS to satisfy distinctly different applications. During periods of bursty packet arrivals or when the AP nears a saturating offered load due to a high speed wired link into the access point, the AP queue will fill and produce undesirable droptail packet losses that will cause intolerable delay for applications such as VoIP or computer gaming. Despite the importance of the AP queue for performance, the blackbox nature of AP details has produced few reported AP queue capacity measurements of commercial products. AP designers have selected their own queue capacities without a healthy understanding in the research community of the rationale for the choice. Hence, WLAN performance papers report simulation results where the queue capacity is picked without knowledge of queue capacities of real APs. The results from this paper are intended to improve this situation.

This paper presents APQ, a methodology that accurately measures AP queue capacity. APQ uses a two step process that first determines the saturation offered load for a given AP configuration before carefully measuring the maximum queuing delay at saturation and thereby determining a blackbox estimate of the internal AP downstream queue capacity. Employing a controlled Host AP, the APQ method was validated to provide solid evidence that APQ accurately measures AP queue capacity. The APQ methodology is used to estimate maximum queue capacity for seven wireless APs (three commercial and four residential class APs). The measurements demonstrate that current AP queue capacities vary across vendors and AP classes. Large AP queue capacities are problematic for interactive applications that encounter congestion due to contending WLAN traffic. While sufficient for single TCP flows, small AP queues must be studied further to understand their impact when there are several contending wireless applications expected good QoS from the infrastructure AP.

These observations suggest future work that rigorously evaluates IEEE 802.11 AP performance over a range of queue capacity implementations and a variety of traffic scenarios. The queue capacities reported here provide a starting point for research that uses a Host AP to vary queue capacities and evaluate AP design strategies, for further performance measurements of the commercial APs, or for providing realistic queue capacities for WLAN simulations. Furthermore, bandwidth estimation tools could be employed in a complementary fashion to tailor application behavior to avoid overflowing the wireless AP buffers.

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] F. Li, L. Rui, M. Li, H. Wu, M. Claypool, and R. Kinicki. Measuring the Queue Size for IEEE 802.11 Wireless Access Points. Technical Report WPI-CS-TR-06-20, CS at WPI, June 2006.
- [13] R. Morris. Scalable TCP Congestion Control. In *Proceedings of IEEE Infocom*, Tel-Aviv, Israel, Mar. 2000.
- [14] E. Pelletta and H. Velayos. Performance Measurements of the Saturation Throughput in IEEE 802.11 Access Points. In *Proceedings of WiOpt*, Apr. 2005.
- [15] 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.
- [16] P. Rubens. Your Next WAP: Hold the Cheese? Wi-Fi Planet, Oct. 2005. Online at: <http://www.wi-fiplanet.com/tutorials/article.php/3556226>.
- [17] C. Villamizar and C. Song. High Performance TCP in ANSNET. *ACM Computer Communications Review*, 24(5):45–60, 1994.
- [18] 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.
- [19] 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.