A Credit-based Home Access Point (CHAP) to Improve Application Performance on IEEE 802.11 Networks

Choong-Soo Lee, Mark Claypool, and Robert Kinicki
Department of Computer Science
Worcester Polytechnic Institute
100 Institute Road, Worcester, MA 01609, USA
[clee01|claypool|rek]@cs.wpi.edu

ABSTRACT

Expanded high speed Internet availability and decreased wireless technology costs have increased the number of residential computers wirelessly connected to the Internet. While residential users run applications with a wide range of network requirements, these applications receive identical treatment from most wireless access points (APs). Although delay sensitive applications can suffer increased latency in the presence of throughput intensive applications, wireless APs contain few tools to mitigate these effects beyond explicitly classifying traffic based on port numbers or host IP addresses. We propose a Credit-based Home Access Point (CHAP) that features credit-based queue management designed to eliminate the need for explicit AP configuration of per-application quality. Based on wireless conditions, CHAP dynamically adjusts flow priorities to better satisfy their application requirements. Preliminary comparisons with DropTail and Strict Priority Queuing (SPQ) demonstrate the merits of our CHAP approach.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication; Packet-switching networks*

General Terms

Design, Performance

Keywords

credit, queue management, 802.11, access point, wireless network

1. INTRODUCTION

Wireless APs have become popular for networking multiple home devices and allowing individual users to simultaneously share one Internet connection while running diverse applications. In addition to desktop and laptop computers, home users now have many other devices with network capabilities such as gaming consoles (handheld and stationary), Voice-over-IP (VoIP) phones, streaming video

servers, cell phones and PDAs. Applications run on these devices have their own distinct Quality of Service (QoS) requirements. For example, a game player on a game console wants low latency for enjoyable gameplay, while a user streaming a high-definition movie trailer prefers consistent, high throughput. Concurrent activities over the same Internet connection can cause congestion that induces degraded application performance by increasing loss or latency or restricting bandwidth. For example, when one user is in a VoIP session while another user downloads a large file, VoIP quality can suffer from choppy sound and increases in delay that affect interactivity.

Increased latency for delay sensitive applications indicates queue buildup somewhere along the network path. In a home Internet connection setup, there are three places where the queue is most likely to become congested: the ISP gateway, the broadband connection device, or the wireless AP. Home devices connect to a wireless AP at up to 54 Mbps (802.11g) or up to 270 Mbps (802.11n). Bandwidth between the wireless AP and the broadband device typically ranges from 100 Mbps to 1 Gbps. Link capacity between the broadband device and the ISP gateway depends on the Internet subscription. The fastest residential Internet access offered in the US is approximately 50 Mbps downstream and 20 Mbps upstream (from Verizon). However, ISPs plan to provide faster broadband access to homes worldwide. For example, Hong Kong, Japan, and the Netherlands have deployed or are testing residential connections of 1 Gbps and Sweden has tested 40 Gbps connections to select homes. When such high bandwidths become available in residences, the wireless AP will become the source of the bottleneck link and the wireless medium (shared by all devices and neighbors) is likely to be the capacity bottleneck.

There have been solutions proposed and implemented by wireless AP manufacturers to provide QoS support. Some APs prioritize a physical Ethernet port for a specific device connected to that port or prioritize flows based on their protocol type and port number. However, the average home user already has difficulty understanding and configuring basic wireless AP features without the added difficulty of configuring an AP with port and flow information. Even for experienced users, these solutions have limitations. For port-based priorities, users must know the protocol and ports for the applications ahead of time and new, unclassified applications are unable to receive appropriate treatment.

The limitations of static QoS assignments has prompted research in automatic classification of flows to satisfy specific QoS needs. Classification of flows removes the need for users to configure APs to prioritize wireless traffic. However, in addition to the limited ability to classify new applications, changes in the wireless network connectivity can reduce overall performance and home network quality when flows with poor connectivity are given higher

priority. Thus, giving priority to a flow with poor connectivity reduces the overall QoS of wireless applications.

To address the shortcomings of static or dynamic flow classification and treatment while providing varied per-application QoS, this paper presents the Credit-based Home Access Point (CHAP). CHAP includes a new queue management scheme that uses per-flow credits to prioritize packet transmissions. This mechanism focuses on improving overall application quality over home wireless networks with minimal configuration and no explicit classification. CHAP takes advantage of general trends between high delay tolerance and high bandwidth usage for wireless client activities. This relationship is captured when one flow has more credits than a concurrent flow that uses more bandwidth. Additionally, flows in bad wireless locations automatically have reduced credits and are given lower priority to gracefully mitigate degraded wireless conditions.[1] CHAP employs frame transmission time as the credit cost metric for each flow. Given the same wireless connectivity, flows using higher link capacities have lower credits compared to flows that require less link capacities. Given the same bandwidth, flows with good connectivity amass more credits than flows with poor connectivity.

CHAP is evaluated using NS-2 [6] simulations through a preliminary set of scenarios. Our simulation results show that CHAP: 1) provides better application quality than DropTail in all scenarios tested; 2) yields Strict Priority Queue (SPQ)-equivalent application quality when all applications are in good locations; and 3) achieves higher overall network performance than SPQ when multimedia applications move from good to bad locations. These application quality improvements are accomplished while providing performance similar to DropTail and SPQ with respect to application throughputs and overall network performance.

The rest of this paper is organized as follows: Section 2 focuses on the derivation and details of the CHAP mechanisms; Section 3 describes the set of simulation experiments run to evaluate CHAP over a range of network conditions and includes analysis of the simulation results and detailed comparisons of the performance of CHAP with DropTail and SPQ; and Section 4 summarizes findings and considers further evaluation and future work.

2. PROPOSED APPROACH

2.1 Bandwidth and Delay for Residential Activities

While there is a wide variety of Internet applications initiated from the home, such applications can be broadly grouped based on similar activities. Popular residential network applications incorporate activities including Web browsing, audio and video streaming, Voice over IP (VoIP), instant messaging, online games, email and file downloading. Furthermore, the wide range of network applications supporting these activities can be broadly characterized based on their bandwidth and delay constraints. Basically, applications of a similar often nature share common network characteristics.

Figure 1 depicts the bandwidth and delay tolerance for a variety of Internet user activities. As the figure suggests, for most applications there is a direct relationship between each activity's bandwidth and its delay tolerance, indicated by the tilted gray rectangle. Generally, the more bandwidth an activity uses, the more delay tolerant it is and vice versa.
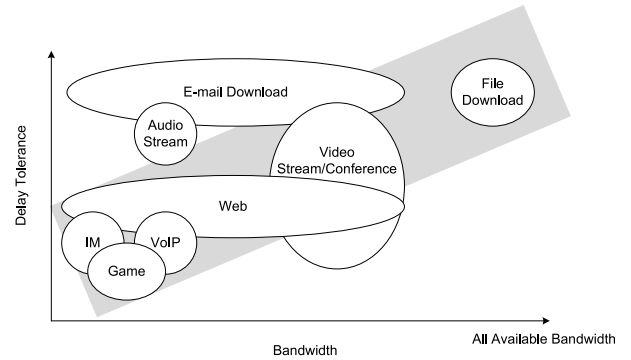
---

[1]Bad wireless locations are where nodes experience increased error/loss/retransmission rate and decreased data rate.

Figure 1: Characteristics of Network Applications

2.2 CHAP Design

Modern operating system schedulers favor interactive processes over CPU-bound processes. Credit-based Home Access Point (CHAP) draws upon the similarity of process scheduling for processes to packet transmission scheduling for network flows and exploits the tendency between delay tolerance and bandwidth usage shown in Figure 1. Bandwidth usage is easily observed by the AP because every packet that arrives and leaves the home network goes through the AP. Since the AP is responsible for routing traffic to the correct host within the home network, the typical AP keeps a Network Address Translation (NAT) table with a 5-tuple per entry: source address, source port, destination address, destination port and protocol. This same 5-tuple is used to identify flows when CHAP updates and monitors per-flow credits used to prioritize packet transmissions in light of the goal of improving overall application QoS.
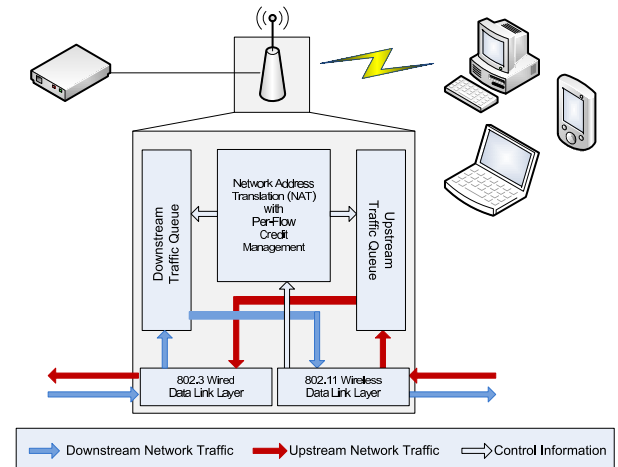

Figure 2: Block Diagram of CHAP

Figure 2 shows the CHAP components inside an AP. The solid arrows indicate downstream and upstream traffic in and out of the home network. Under CHAP the downstream and upstream queues utilize control information from the enhanced NAT component with credits to prioritize the packets in the queue. Furthermore, the NAT component requires control information from the 802.11 data link layer to update the credits.

To adjust credits, CHAP uses channel usage time at the wireless data link layer to reflect the combination of a flow's bandwidth usage and wireless connectivity. The more frames a flow transmits,

the fewer credits it retains. The larger the frame transmission time, the more credits the flow expends. Generally, flows that drain credits slower have higher priority. This scheme ensures that nodes with poor connectivity, even if they only require low application-level bandwidth, do not degrade overall wireless network performance.

Besides decrementing credits during transmission, CHAP boosts credits when all flows run out. The form of the credit boost (based on an earlier incarnation of the Linux process scheduler) is: $\alpha'_i = \frac{\alpha_i}{2} + I$, where i is the flow index, α_i is the credit of flow i, and I is the increment. This boost mechanism triggers when every flow with packets in the CHAP queue has 0 (or fewer) credits. A boost of I time units (consistent with the unit of credits) is then applied to *every* active flow.

Algorithm 1 CHAP::dequeue()

1: find k such that $\alpha_k = max[\alpha_1, ..., \alpha_n]$ and $k \in$ set of back-logged flows
2: **if** $\alpha_k \leq 0$ **then**
3:     $\alpha_i = \frac{\alpha_i}{2} + I$ for all $i \in$ active flows
4: **end if**
5: $p =$ the first packet from flow k in the queue
6: return p
7: $\alpha_k = \alpha_k - cost(p)$

Algorithm 2 CHAP::enqueue(p)

1: $j =$ flow ID of p
2: add p to the end of the queue
3: **if** the queue is full **then**
4:     find k such that $\alpha_k = min[\alpha_1, ..., \alpha_n]$ and $k \in$ set of back-logged flows
5:     $p_{tmp} =$ the first packet from flow k in the queue
6:     drop(p_{tmp})
7: **end if**

Algorithms 1 and 2 summarize the dequeue and enqueue functions for downstream traffic in pseudo code format. CHAP::dequeue() returns the first packet of the flow with the most credits in the queue. Step 1 finds the flow with the most credits. If a credit boost is necessary in steps 2-4, CHAP boosts credits for all active flows. Steps 5-6 dequeue the packet from the flow picked in Step 1. Step 7 decreases credits according to the packet transmission time spent at the wireless data link layer. This transmission time is a function of data rate, channel error rate and the number of wireless link layer retries.

CHAP::enqueue(p) adds the incoming packet p to the queue. If the incoming packet will overflow the queue, CHAP drops the oldest packet from the flow with the fewest credits in the queue. Step 1 retrieves the flow ID of the incoming packet p. Step 2 adds the incoming packet to the queue. Step 3 checks if the queue is full. If the queue is full, CHAP proceeds to find a packet to drop from the backlogged flow with the fewest credits following Steps 4-6. Steps 3-6 ensure there is room for an incoming packet by limiting the maximum number of packets in the queue to $q_{size} - 1$.

To illustrate CHAP's credit-based mechanism, Figure 3 tracks the credits for three different flows from scenarios explained in Section 3. Figure 3a shows the interaction between a video streaming application and two FTP applications in good locations. Being in good locations, the wireless channel usages to send a packet for all three flows are approximately the same. Since the FTP flows always have data to send, this produces a steady drain on their credits, represented by the angled slopes downward. The video

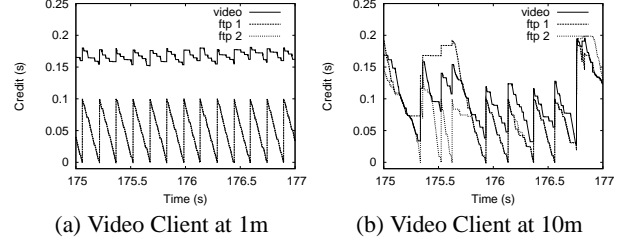(a) Video Client at 1m   (b) Video Client at 10m

Figure 3: Credit of Video and FTPs

sends a video frame every 33 ms, represented by the vertical steps present in the credits of the video application. Each time both FTP applications run out of credits, all three applications get a credit boost. The credit for the video application is highest because the two FTP applications drive CHAP's boosting mechanism by using more bandwidth. Hence video packets retain a higher priority than FTP packets.

Figure 3b graphs credits for the same traffic scenario but the video application has been moved to a bad location. Although the video uses the same amount of application-level bandwidth as in the previous location, the cost to send the video packets has gone up due to the additional channel time required. The overall video credits are more in line with the credits for the FTP applications and video packets have the same priority as packets sent by the FTP applications.

3. PERFORMANCE

This section compares CHAP against DropTail and SPQ where priorities are specified for known applications. CHAP is implemented in Network Simulator 2 (NS-2) [6] which provides frame-level behavior for IEEE 802.11 networks as well as DropTail queuing and FTP traffic modules. We implemented CHAP and SPQ queuing and added game, video and Web applications into NS-2 (details are provided in Section 3.1). DropTail, the default queue management for wireless home APs, serves as a performance baseline. Since SPQ uses predetermined flow priorities, its performance is close to the optimum quality when flows have been correctly classified.
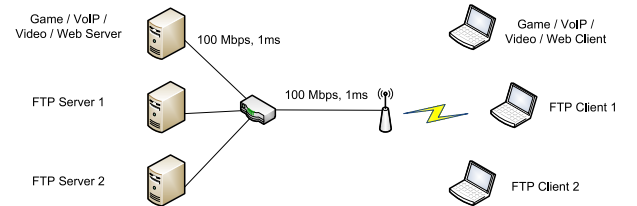

Figure 4: Simulation Topology

Figure 4 illustrates the simulation topology used to model a typical home network. Connections between application servers and the gateway and between the gateway and the AP are 100 Mbps Ethernet links with 1 ms latencies. Each wireless client communicates with the infrastructure-mode AP via IEEE 802.11g. The AP sends beacon frames every 100 ms with scanning mode set to passive. The AP queue size is 35 packets, a typical queue length for residential wireless access points [5]. All nodes utilize DropTail queuing except for the AP on the wireless interface. Unless other-

wise noted, all wireless nodes are 1 meter from the AP. CHAP is initialized such that I and the default credits are 100 ms per flow.
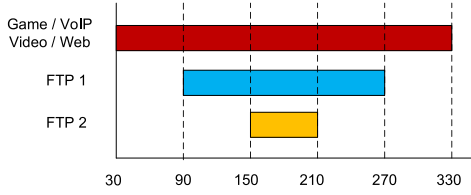

Figure 5: Simulation Scenario

Figure 5 gives the flow arrival and departure schedules. Four distinct applications are tested separately in different scenarios. The application of interest starts at 30 seconds and finishes at 330 seconds.

3.1 Simulated Applications

NS-2 includes FTP, Constant Bit Rate (CBR) and FullTCP applications. FTP applications send data over TCP CUBIC connections. Both game and VoIP applications are simulated with a CBR application periodically sending fixed-size packets. Web traffic is modeled by a FullTCP application sending data in both directions over a single TCP NewReno connection.

The game application uses a traffic model for Halo 2 [10], a popular First-Person-Shooter (FPS) game. The average client packet payload is 44 bytes while the average server packet payload is 72 bytes. Both the server and client send 25 updates per second. Hence, the game is modeled as two flows where the server CBR flow sends 72 byte payloads every 40 ms to a client over a UDP connection and the client CBR flow sends 44 byte payloads every 40 ms to the server over a different UDP connection.

The simulated VoIP application models a G.711 codec, generating 160 bytes of voice data every 20 ms. Unlike the game application, the two VoIP CBR flows are symmetric with both flows using UDP to send 160 byte payloads every 20 ms in opposite directions.

The video application is based on traces of an actual, high-quality video [7, 3]. Specifically, an Indiana Jones movie is encoded using H.264 Scalable Video Coding (SVC) at 30 frames per second. Only the 30 frames-per-second layer drives the video application. The resolution of the movie is CIF (352×288) with 16 frames in a Group of Pictures with 3 B frames in between each I/P frame [9]. The available quantization parameters for encoding are 10, 10 and 12 for the I, P and B frames, respectively. The average frame size of the video is 10.31 Kbytes while the average bit rate is 2.47 Mbps. The largest frame size is 50 KBytes with a peak bit rate of 12.10 Mbps. The video application sends a frame every 33 ms, showing up in the network as a burst of UDP packets. Lost frames or packets are not retransmitted. This implementation represents video streaming that is unresponsive to network congestion and packet loss.

Based on a recent Web browsing behavior model [4], the Web application characterizes HTML and object sizes in typical Web pages. HTML container page size, embedded object sizes and reading time follow a lognormal distribution while the number of embedded objects follows a gamma distribution. A FullTCP client sends an initial 350-byte request and the server responds with an HTML object where object size is drawn from a truncated (maximum 2 Mbytes) lognormal distribution. Upon receiving the HTML object response, if there are any embedded objects in the Web page, the client sends another 350-byte request. Upon receiving the second request, the server responds with the corresponding number of objects transmitted back to the client. After receiving the final response from the server, the client sleeps for some time (representing user reading time) before generating another request. The FullTCP connection is reset between requests.

3.2 Performance Analysis

Each application of interest has its own quality performance metric. The game applications use the G-Model MOS [8] developed for Quake IV. Although there are differences between Halo 2 and Quake IV, both are FPS games. The G-Model yields a Mean Opinion Score (MOS) ranging from 1.00 (worst) to 4.37 (best) based on a function of the game's average ping and jitter times. Loss is not included in the G-Model due to its negligible impact on gameplay [1].

The VoIP applications use the E-Model [2] which provides a MOS mapped from R-factor, a measure of audio impairments from delay and the codec. Delay degrades VoIP quality when one-way delay exceeds 177 ms. In the home scenarios simulated, delay is less than tens of milliseconds and the E-Model is unaffected. However, wireless losses decrease VoIP quality controlled by the codec. Jitter does not factor into VoIP quality as the VoIP player buffering typically mitigates jitter effects. The E-Model MOS ranges from 1.0 (worst) to 4.5 (best).

Video application performance is gauged by the playout frame rate in a range from 0 to the maximum encoded frame rate. In the simulations, it is assumed the receiver plays all playable frames when they arrive and it never discards late frames. Specifically, I frames are playable if all packets are received. B and P frames are playable if all packets are received and dependent frames are playable.

Web application performance is measured by response time calculated from the time the client sends the request until the last object is completely received. Response time varies greatly due to variability in object sizes and distances to the server.

FTP application performance uses throughput. Higher throughput results in shorter download time (better performance) while lower throughput results in longer download time.

Scn.	App. (Unit)	DropTail	CHAP	SPQ	% Impr
Game	Game (MOS)	3.67	4.27	4.27	+16%
	FTPs (Mbps)	26.42	26.60	26.00	+1%
VoIP	VoIP (MOS)	4.37	4.42	4.42	+1%
	FTPs (Mbps)	26.18	26.34	26.17	+1%
Video	Video (fps)	9.97	30.00	30.00	+201%
	FTPs (Mbps)	23.03	23.12	22.95	0%
Web	Web (ms)	109.52	39.70	33.96	+64%
	FTPs (Mbps)	26.59	26.73	26.54	+1%

Table 1: Summary of Performance Metrics

Table 1 summarizes the performance from the simulations of the four distinct applications of interest. Game, VoIP and video application performance metrics and FTP flow throughputs represent averages over the time interval between 150 and 210 seconds. Web response times are averaged over the entire duration of the simulation. Due to space limitations, only the game simulation results are analyzed in detail. Note that regardless of the application of interest, in Table 1 FTP throughputs remain essentially constant across the three simulated AP queues.

The VoIP application achieves a high MOS value of 4.37 with the DropTail queue because VoIP MOS is only degraded by packet loss. CHAP and SPQ both improve VoIP MOS by 1% to 4.42.
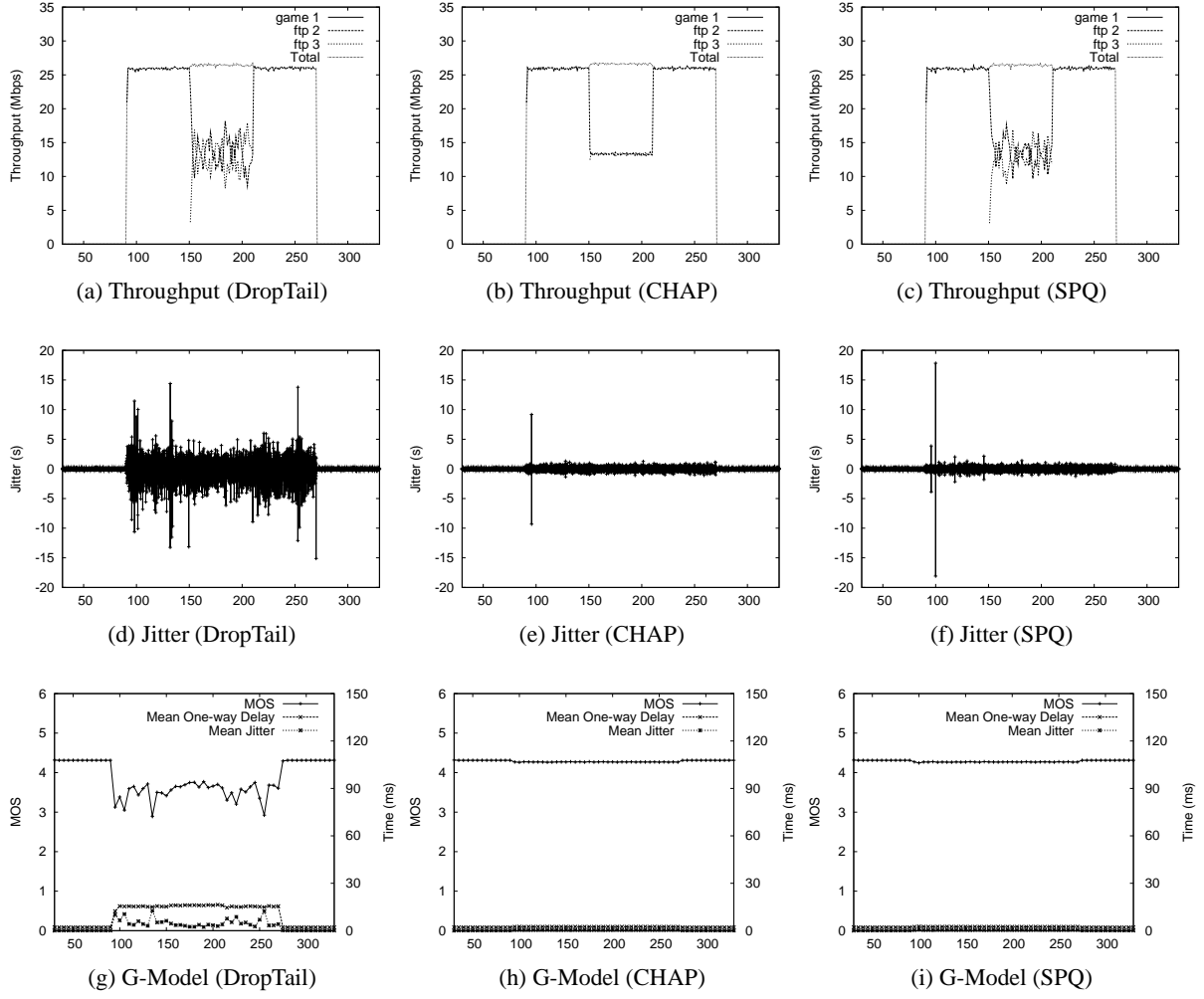
Figure 6: Game Performance: DropTail vs. CHAP vs. SPQ

While the video application plays at only 10 frames per second (fps) under DropTail, CHAP and SPQ both run at 30 fps, the maximum playable frame rate. This is about a 200% improvement over DropTail. In the Web traffic simulation, average Web response time for the DropTail queue is about 100 ms. SPQ lowers Web response time by 67% down to 34 ms and CHAP provides 39 ms Web response time, about a 65% improvement.

While the game application with DropTail achieves a 3.67 MOS, both SPQ and Chap AP queuing improve the MOS by 16% to 4.27. Figure 6 provides game performance results over the duration of the three simulations with the first row of graphs comparing game, FTP and total throughput for the three AP queues. Total throughputs in all three cases are about the same. CHAP provides the same bandwidth for the FTP applications but with reduced variance due to the round-robin nature of the credit scheme for flows with equal effective bandwidths. Figures 6d, 6e and 6f demonstrate that the jitter in packet arrivals for the game application has less variability when either CHAP or SPQ are used by the AP compared with DropTail. Figures 6g, 6h and 6i show the G-Model MOS and its factors. In Figure 6g, the MOS decreases once the first FTP application starts at 90 seconds. The delay is increased due to a nearly full queue and the jitter becomes pronounced. Figures 6h and 6i

show that both CHAP and SPQ provide nearly constant delay and jitter for the game application and a resultant high MOS value.

As mentioned in Section 1, by providing pre-classified applications with static priorities, SPQ may provide the best performance when all wireless nodes are in good range. However, when a device running an application with high priority moves to a bad location this yields an increased number of retransmissions and loss events. Overall performance is improved by lowering the priority of the application in the bad location to provide more channel capacity to flows running from better wireless locations. Figure 7 depicts such behavior by moving the wireless node running a video application farther away from the AP in subsequent simulation runs. Namely, simulations were run with the same setup and settings except the video client was placed at 1, 5, 10, 15, 20, 25 and 30 meters from the AP. Video and FTP throughputs are about the same for Drop-Tail and SPQ with FTP throughputs dropping close to zero at about 15 m. CHAP starts punishing the video application as early as 10 m, but CHAP keeps FTP throughputs higher as performance degrades gracefully all the way to 30 m. However, video throughput is not the only measure of quality for video applications. Figure 8 graphs video playable frame rate across different distances for DropTail, CHAP and SPQ. Although DropTail maintains about
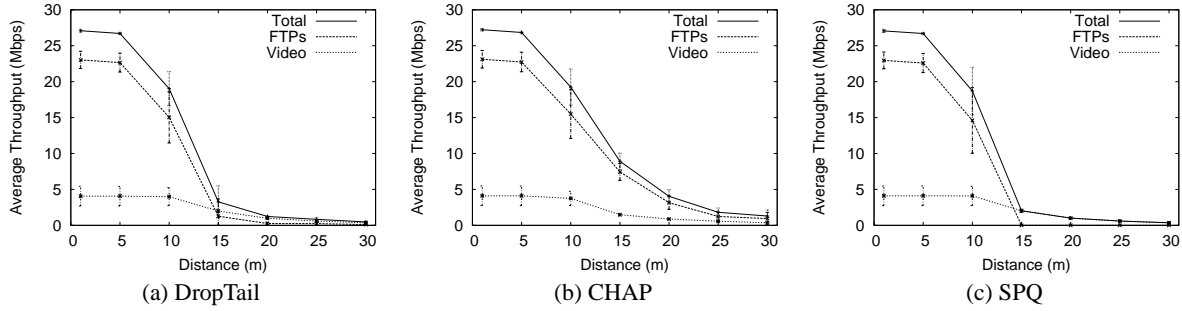
Figure 7: Throughput versus Distance (Video Scenario)

the same throughput as SPQ, DropTail video frame rate is far below SPQ video frame rate. The burstiness of Droptail packet drops causes a large degradation in video frame rate. The video frame rate with CHAP starts to degrade at 10 m. However, all three video frame rates are near zero at 15 m. Therefore, assisting the video traffic beyond 15 m does not help the video quality for users. Thus, SPQ at 15 m provides no real benefit to the video while FTP gets near zero throughput. Although CHAP degrades video quality earlier, it ends up helping the FTP applications in configurations where the video is unwatchable.
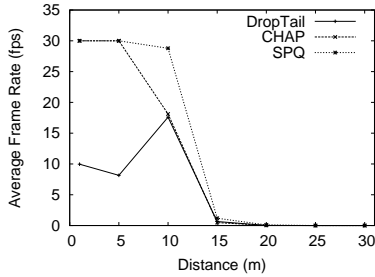

Figure 8: Video Frame Rate versus Distance (Video Scenario)

4. CONCLUSIONS

This paper introduces the Credit-based Home Access Point (CHAP), a queue management scheme designed specifically to improve application QoS over home wireless networks. CHAP retains and manages credits for all flows passing through a wireless AP. CHAP boosts credits when all active flows run out of credits and decrements a flow's credits as a function of wireless channel usage time. Depending on the flow characteristics of home wireless user activities, CHAP has the potential to increase the quality of delay sensitive activities while sustaining the quality of delay insensitive activities. We are unaware of any other AP queue management techniques that can improve quality of delay sensitive applications without using prior configuration and/or flow classifications.

A preliminary set of simulations where four applications individually compete with two FTP applications compare CHAP against DropTail and Strict Priority Queuing (SPQ) at an AP. When all wireless clients are in good locations, CHAP achieves significant quality improvement over DropTail for some applications and performs nearly as well as SPQ. Although SPQ provides the best performance in the presence of clients in good locations, it requires pre-configured priorities for known applications. In the presence

of clients in bad locations, CHAP performs better than SPQ with respect to FTP throughputs while still assisting the video application as much as is prudent.

Our ongoing work involves evaluating CHAP under a broader range of conditions and circumstances that include a wider range of wireless clients, more variability in wireless client location and AP queue size, and the presence of upstream and other background traffic such as Peer-to-Peer (P2P) applications. Furthermore, our plan is to evaluate more applications such as games of different genres, scalable and/or responsive video streaming and video conferencing. Also planned is exploration of conditions where CHAP may be less effective such as when delay sensitive and delay insensitive applications have nearly identical bandwidth demands. Formal analysis of CHAP in Algorithms 1 and 2 including analysis of the setting for the increment I is currently underway. A prototype implementation of CHAP on a Linux-based AP is planned to validate the simulations and to understand implementation nuances.

5. REFERENCES

[1] 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 Network and System Support for Games Workshop*, Portland, OG, USA, Sept. 2004.

[2] L. Carvalho, E. Mota, R. Aguiar, A. F. Lima, J. N. de Souza, and A. Barreto. An E-Model Implementation for Speech Quality Evaluation in VoIP Systems. In *Proceedings of the 10th IEEE Symposium on Computers and Communications*, pages 933–938, June 2005.

[3] G. V. der Auwera, P. T. David, and M. Reisslein. Traffic and Quality Characterization of Single-Layer Video Streams Encoded with H.264/MPEG-4 Advanced Video Coding Standard and Scalable Video Coding Extension. In *IEEE Transactions on Broadcasting*, volume 54, Sept. 2008.

[4] J. J. Lee and M. Gupta. White Paper: A New Traffic Model for Current User Web Browsing Behavior, Intel Research White Paper, Sept. 2007.

[5] F. Li, M. Li, R. Lu, H. Wu, M. Claypool, and R. Kinicki. Measuring Queue Capacities of IEEE 802.11 Wireless Access Points. In *Proceedings of the IEEE International Conference on Broadband Communications, Networks and Systems*, Raleigh, NC, USA, Sept. 2007.

[6] U. of California Berkeley. The Network Simulator - ns-2. Online at http://www.isi.edu/nsnam/ns/.

[7] P. Seeling, M. Reisslein, and B. Kulapala. Network Performance Evaluation with Frame Size and Quality Traces of Single-Layer and Two-Layer Video: A Tutorial. *IEEE Communications Surveys and Tutorials*, 6(3):58–78, 2004.

[8] A. F. Wattimena, R. E. Kooij, J. M. van Vugt, and O. K. Ahmed. Predicting the Perceived Quality of a First person Shooter: the Quake IV G-model. In *Proceedings of 5th ACM SIGCOMM Workshop on Network and system support for games*, Singapore, Oct. 2006.

[9] H. Wu, M. Claypool, and R. Kinicki. Guidelines for Selecting Practical MPEG Group of Pictures. In *Proceedings of IASTED International Conference on Internet and Multimedia Systems and Applications (EuroIMSA)*, Innsbruck, Austria, Feb. 2006.

[10] S. Zander and G. Armitage. A Traffic Model for the XBox Game Halo 2. In *Proceedings of the International Workshop on Network and Operating Systems Support for Digital Audio and Video*, June 2005.