Treatment-Based Traffic Classification for Residential Wireless Networks

Feng Li, Mark Claypool and Robert Kinicki
Department of Computer Science, Worcester Polytechnic Institute
100 Institute Rd, Worcester, MA, 01609, USA
Email: {lif,claypool,rek}@cs.wpi.edu

***Abstract*—As broadband connection capacities increase, residential network users can experience performance degradation when multiple applications run concurrently over bottlenecked wireless access points (APs). This paper presents *Classification and Treatment iN an AP* (CATNAP), a plug-and-play solution for an AP that automatically classifies and treats flows passing through without any user intervention. CATNAP classifies application flows across three dimensions based on their response to flow traffic, bitrate greediness and interactivity. Then, based on the classification, CATNAP treats flows by rate limiting/preserving, adjusting packet time in queue or dropping packets. CATNAP is implemented in NS-2 and evaluated against DropTail and Strict Priority Queuing under various network and traffic conditions. Analysis of the results show CATNAP provides better QoS support for VoIP, games and streaming video applications.**

***Index Terms*—Wireless Network, IEEE 802.11, QoS**

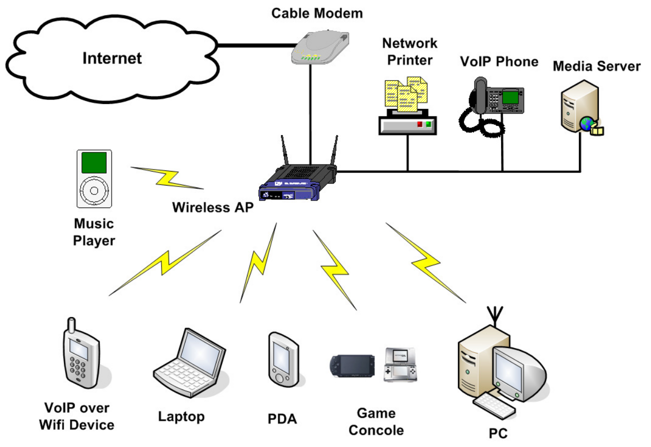

Fig. 1. Typical Residential Wireless Network

I. Introduction

The wireless access point (AP) (see Figure 1) connects residential computers and entertainment devices inside the home to the Internet. While IEEE 802.11 (a/b/g/n) wireless protocols have improved their nominal link rates (e.g., up to 260 Mbps for 11n), they rarely reach these top speeds [16]. In uncontrolled environments, real attainable AP throughputs are hampered by the shared medium, hidden terminals, low-end 11b and 11g devices, devices on overlapping frequencies (e.g., microwave ovens and Bluetooth devices), walls and other obstacles. Conversely, residential broadband connection capacities have grown – Akamai Technologies reports that the top four countries have more than 50 Mbps peak connections to the home and three U.S. states (Massachusetts, New Jersey and Maryland) provide 50 Mbps peak residential connections [17]. This often leaves the wireless AP link as the network bottleneck.

Wireless APs typically do not differentiate traffic. By treating concurrent applications with dramatically different quality of service (QoS) requirements equally, APs cause degraded QoS experiences for applications such as network games, real-time videos, or Voice over IP (VoIP) when another application simultaneously downloads a file through the same AP [7], [14].

Having studied various QoS mechanisms including IEEE 802.11e, Integrated Services (IntServ), and Differentiated Services (DiffServ), researchers have yet to deploy QoS functionality for residential use due to the lack of suitable mappings of application flows into specific QoS classes. Recent studies demonstrate that port-based traffic identification is error-prone [13], [15] since applications often intentionally avoid using well-known or consistent ports. Packet payloads can be inspected to achieve higher accuracy [13], but the high computational complexity and ineffectiveness for encrypted packet payloads make this approach impractical for real-time classification. While statistics-based classification methods use machine learning techniques and assume applications transfer data in repeatable, identifiable patterns [6], [12], [19], they are unusable for new, unclassified applications and ineffective in wireless environments. Since APs have limited ability to detect the application types running over their networks, they cannot easily shape traffic to provide better QoS. Currently, most residential wireless APs provide no QoS support (e.g., the 5th generation Apple Airport Extreme IEEE 802.11n wireless router [10]) or only support QoS with complicated manual configuration (e.g., the Cisco Wireless LAN controller 5500 series, which needs 14 steps to add per-user capacity limitations).

This paper presents a traffic classification technique that identifies applications based on their wireless network traffic characteristics while providing the AP with new traffic treatments to shape the traffic by mitigating the effects of wireless media [4]. *Classification And Treatment iN an Access Point* (CATNAP) automatically differentiates flow traffic into eight categories and treats traffic to improve QoS over residential wireless links. Implementation in NS-2 and comparison against DropTail and Strict Priority Queuing under various network and traffic conditions shows CATNAP provides better QoS support for multimedia applications.

The paper is organized as follows: Section II describes the CATNAP treatment-based classification approach; Section III discusses simulations experiments used to evaluate CATNAP; Section IV analyzes the results of the simulation experiments; and Section V summarizes our conclusions.

II. Treatment-Based Classification

CATNAP consists of two major components: the *classifier* and the *treater*. The classifier places the flows into different categories in real-time based on the nature of their traffic, without prior training and without relying upon ports or payloads. As wireless APs have limited processors and memory, the CATNAP treater uses four non-CPU-intensive operations: *push, delay, drop* and *limit* to help flows meet their QoS requirements.

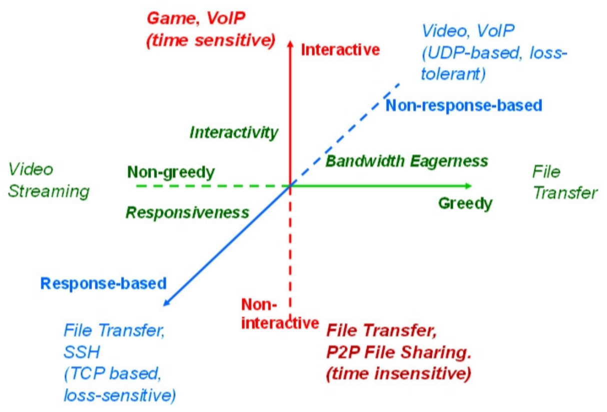

Fig. 2. QoS Dimensions for Most Home Applications

Figure 2 describes the QoS requirements of most home applications in a three dimensional space. The first dimension divides applications into response-based and non-response-based. Response-based flows directly or indirectly cause traffic in the reserve direction.

CATNAP further separates applications into interactive and non-interactive flows based on their delay tolerance. Since interactive applications such as VoIP or network games are sensitive to network delay, CATNAP pushes interactive applications to the front of the AP queue while non-interactive applications (e.g., P2P file sharing) can tolerate being delayed without significant degradation to QoS.

The final dimension in Figure 2 distinguishes greedy applications (e.g., FTP) that grab as much network capacity as possible from non-greedy applications (e.g., rate-limited streaming video) that do not consume additional capacity, even when it is available. CATNAP "tames" greedy TCP applications by reducing the advertised window in TCP ACK packets to limit transmission rates while reserving dropping as a treatment only for non-response-based, greedy UDP applications.

A. Identifying response-based/non-response-based flows

CATNAP classifies TCP flows as response-based and most UDP flows as non-response-based. DNS UDP flows running on port 53 are also classified as response-based. Classification is done by inspecting the packet header for protocol type TCP and port number for port 53. As dropping a packet triggers a TCP retransmission, CATNAP only uses the drop operation on non-response-based applications that generally tolerate some packet loss without significant quality degradation (e.g., video over UDP).

B. Identifying interactive/non-interactive flows

The CATNAP interactive/non-interactive classifier uses packet length to infer flow type. Interactive applications tend to transmit data in smaller packets relative to typical network MTUs. Non-interactive applications such as FTP usually send large packets near the network MTU. Some applications like the SSH session across a campus network shown in Figure 3 change their nature from interactive to non-interactive flows and back during their life cycle. In the figure, the x-axis is time and the y-axis is packet length in bytes. Each '+' depicts packet length for one packet. For this application, this SSH user executed around two dozen Linux directory commands before opening (via `cat`) a 30 MB file at the 40th second. The file is then displayed onscreen for about 17 seconds. After the 60th second, the user resumed the directory operations. This single SSH flow was interactive when browsing the directory and non-interactive when displaying the file. An interactive/non-interactive classifier must detect state changes quickly while ignoring packet length outliers (e.g., the occasional large packets between time 0 and 40 seconds that do not indicate a change in state).

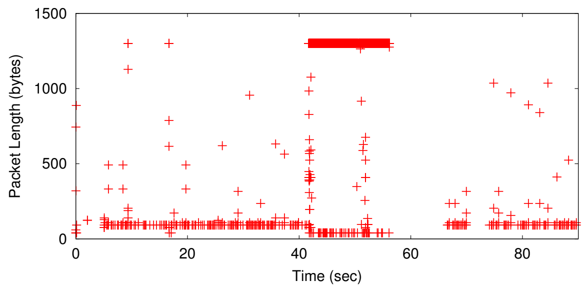

Fig. 3. Packet Length versus Time for SSH Flow

CATNAP uses as an exponential weighted moving average (EWMA) packet length to classify interactive/non-interactive flows. When packet p of flow i arrives, CATNAP updates for $\overline{L}$ flow i:

$$\overline{L} \leftarrow \overline{L} \times (1-\alpha) + \alpha \times length(p) \qquad (1)$$

using the weighting constant α. To meet the requirements of quick reaction time and stability, the algorithm uses different weights for interactive an non-interactive regions – $\alpha = 0.6$ for non-interactive regions and $\alpha = 0.05$ for interactive regions (see [9] for details). Figure 4 shows the EMWA packet length for the SSH flow in Figure 3.

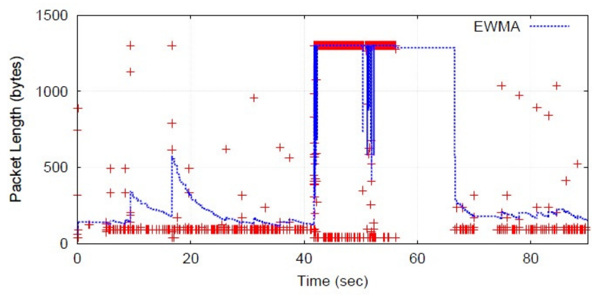

Fig. 4. EWMA Packet Length versus Time for Classifying Interactive Flows

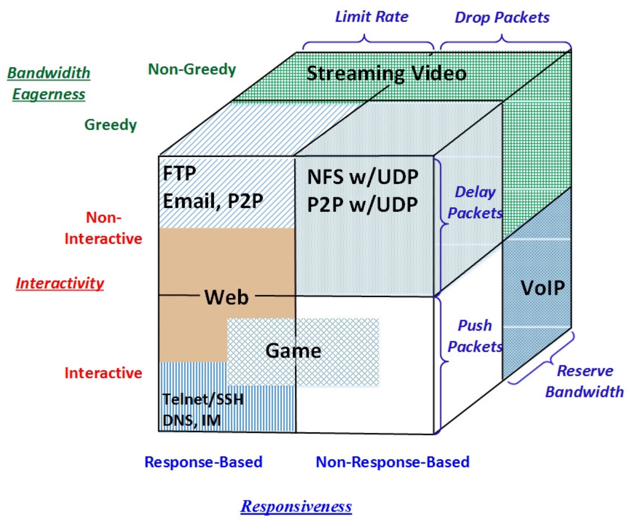

Fig. 5. Treatment-Based Traffic Classification

C. Identifying greedy/non-greedy flows

The CATNAP greedy/non-greedy classifier first computes the fair share bitrate (R_{nint}) for all non-interactive flows as the threshold to differentiate greedy/non-greedy flows:

$$R_{nint} \leftarrow \frac{C - \sum r_{int}}{N_{nint}} \tag{2}$$

where C is the estimated downlink capacity (see [9] for details), N_{nint} is the number of non-interactive flows, and $\sum r_{int}$ is the summation of bitrates for all interactive flows. R_{nint} is used as a threshold to differentiate greedy or non-greedy flows – flows consuming capacity less than $R_{nint}/2$ are classified as non-greedy, while flows consuming more than $R_{nint}/2$ are classified as greedy. There should be no congestion on the wireless AP downlink if all non-interactive flows transmit under R_{nint}, the non-interactive flow target rate.

After classifying flows as greedy/non-greedy, CATNAP calculates the fair share rate for greedy flows (R_{greedy}):

$$R_{greedy} \leftarrow \frac{C - \sum r_{int} - \sum r_{non-greedy}}{N_{greedy}} \tag{3}$$

where C is the estimated downlink capacity, $\sum r_{int}$ totals the capacity use for all interactive flows, $\sum r_{non-greedy}$ is the capacity use for all non-greedy and non-interactive flows, and N_{greedy} is the count of greedy flows.

D. Treatments

After classifying a flow, CATNAP maps that flow into one of the treatment methods highlighted by the blue text in Figure 5. The CATNAP treater pushes interactive flows to the head of the transmit queue to gain priority over non-interactive flows that it delays. While the treater drops greedy, non-response-based UDP flows in proportion to their bitrate excess above the fair share, it actively slows down greedy, response-based TCP flows by lowering their advertised window. CATNAP reserves without limiting the needed capacity of non-greedy flows.

E. Example

The top of Figure 6 repeats Figure 3 while the bottom provides colored horizontal bars that depict CATNAP's classification of this SSH flow over time. Being TCP-based, CATNAP always classifies the SSH flow as response-based. Through 40 seconds of small packets, CATNAP classifies the flow as interactive, but changes the SSH classification to non-interactive from time 40-65 seconds as the packets become large, and then returns it back to non-interactive at time 65 seconds when the packets become small again. As the SSH flow uses less than its fair share up to time 40 and after time 65, CATNAP identifies it as non-greedy except during its greedy stage (time 40-58 seconds). Although not described here due to lack of space, CATNAP considers flows active when they have sent packets during the past epoch, so the bottom bar indicates that CATNAP considers the flow active except for time period 58-65 seconds.

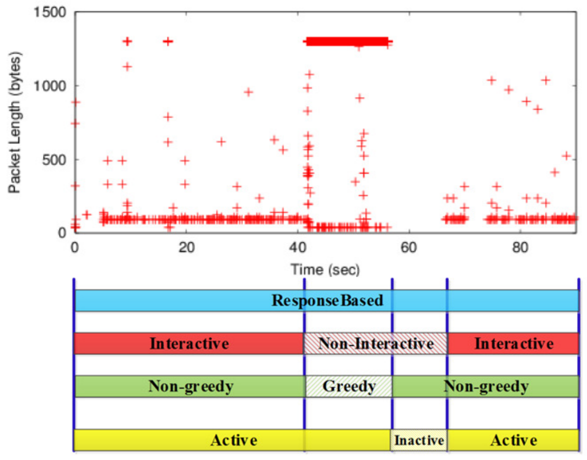

Fig. 6. CATNAP Classification Example

Figure 7 extends the previous SSH example by adding a concurrent FTP flow running in the background. The top of Figure 7 indicates FTP throughput in Mbps over time (in seconds on the x-axis) in green and SSH throughput over time as a red line. The bottom of the figure uses colored horizontal bars to depict the treatment CATNAP applies over time to the SSH flow. CATNAP does not drop packets from the SSH flow. When SSH is interactive, CATNAP pushes the packets to the head of the queue and when non-interactive, CATNAP delays SSH packets in favor of any other interactive flows. CATNAP only reserves capacity when the SSH flow is non-greedy. When the SSH flow is active, CATNAP fully treats the flow, otherwise CATNAP yields any SSH reserved capacity to the remaining active flows.

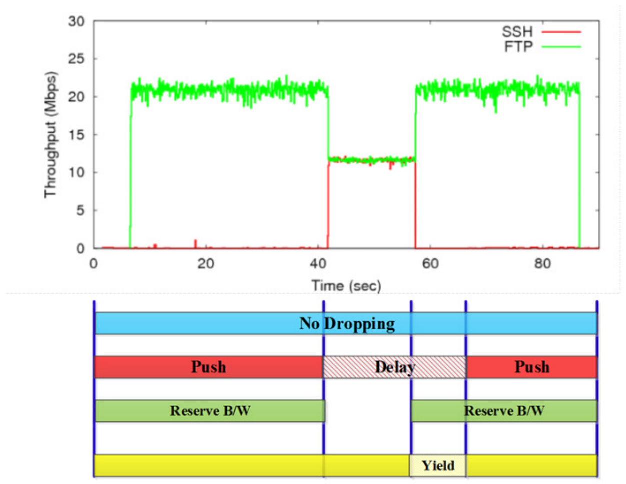

Fig. 7. CATNAP Treatment Example

III. Experiments

We implemented CATNAP in the NS-2 network simulator [1] and evaluated it over a wide range of network conditions and a variety of applications with diverse QoS requirements. Due to space limitations, this paper presents partial results using VoIP, network games, streaming video, Web and FTP applications (see [9] for the full set of results).

The simulation experiments evaluate the CATNAP AP against DropTail First-In, First-Out queuing (the default home AP), and Strict Priority Queuing (SPQ) which serves as a pre-configured "best case" competitor. Since CATNAP treats flows, its queue capacity is 1000 packets while DropTail and SPQ queues are set to the product of the bandwidth and round-trip time (RTT).

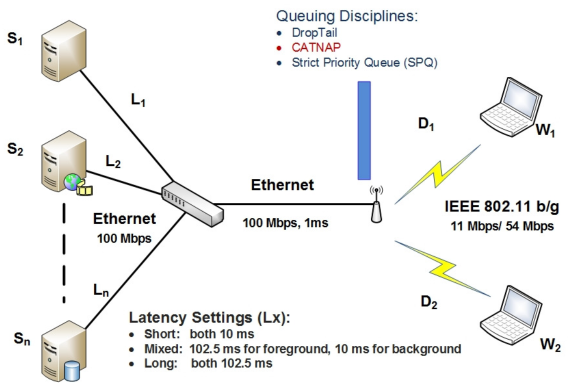

Fig. 8. Experiment Setup

In our simulated topology (see Figure 8), two wireless nodes, W_1 and W_2, connect to the access point (AP) from distances $D_1 = 5m$ and $D_2 = 5m$, respectively and communicate over a single channel infrastructure network using either 54 Mbps (IEEE 802.11g) or 11 Mbps (IEEE 802.11b). To mimic typical APs in homes partitioned into several rooms, we disable RTS/CTS, set MAC layer retransmissions to 4, set beacon frames to be sent every 100 milliseconds (ms), and set the NS-2 path loss exponent to 4.0 and the shadowing deviation to 7.0.

The wireless AP connects to the gateway via a 100 Mbps duplex Ethernet link with 1 ms latency and the gateway communicates with multiple wired servers over symmetric, 100 Mbps links with various latencies of L_1 to L_n. Our experiments set short, one-way latencies to 10 ms (half of the RTT mode measured over residential networks [11]) and long, one-way latencies to 102.5 ms. One wireless background client runs an FTP download from a server to induce congestion into the simulation experiments. In general, the remaining wireless clients serve as destinations for the UDP-based network game, UDP-based VoIP, TCP-based Web and TCP-based video flows, respectively. Due to space constraints, this paper reports only on experiments where a single foreground application competes with the background FTP application.

In mixed latency experiments, the foreground flow experiences long latencies while the background application experiences short latencies. The background flow starts at the 5th second and ends at the 115th second. The VoIP, game or video flows start at the 10th second, spaced out by 0.3 seconds, and end at the 110th second. The analysis covers from the 15th second to the 105th second to measure performance during steady-state.

The game traffic uses the NS-2 traffic generator for Quake 4 [5], a popular first person shooter game with an Auto Regressive Moving Average model, and sends packets with average payload size of 69.5 bytes, excluding headers, every 50 ms. The game client transmits packets with an average payload size of 64.5 bytes, excluding headers, to the game server every 10.75 ms. The quality metric for the game session is the 10th percentile of the G-Model Mean Opinion Score (MOS) value [18].

Using a traffic generator simulating a G.711 codec, UDP-based VoIP traffic is simulated by sending a constant bitrate stream of packets with a 172 byte transport layer payload between two nodes every 20 ms and passing TCP-based VoIP packets at the same rate to the NS-2 TCP agent. The VoIP session quality metric is the 10th percentile of the MOS produced by the E-Model [3].

Using a third-party plugin [2] based on frame sizes extracted from traces of *Indiana Jones I: Raiders of the Lost Ark*, the experiments simulate video traffic encoded in a single layer, Common Intermediate Format (CIF) at 30 frames per second (fps) with a resolution of 352×288 pixels. The Group of Pictures (GoP) consists of 16 frames with three B-frames between each pair of I/P frames. The quantization scales for I, P, and B frames are 10, 10, and 12, respectively. The average frame rate is the video session quality metric.

The Web traffic uses an NS-2 traffic generator based on a previously developed model [8]. The sequence of the simulated Web session assumes that all objects in the requested Web page are on the same Web server, and the Web client and server simulate a persistent, pipelined *HTTP/1.1* connection. After connecting to the Web server, the Web client sends a request to the Web server which replies with one HTML object. Upon receiving the HTML object, the Web client calculates the number and size of each embedded object, and sends the corresponding requests to the Web server to retrieve the embedded objects. The response time between the initial client request and when the client receives the last embedded object is the Web session quality.

IV. Results

This section presents results for two-client NS-2 experiments where flow 2 is always an FTP running in the background. Note SPQ always gives priority to flow 1, the foreground application, and the background FTP flow always sees short latencies in mixed latency experiments to induce congestion on the foreground flow.

Figure 9 graphs cumulative throughput for two simultaneous FTP flows on the y-axis. The x-axis has three clusters: left has both flows with short latencies, right has both flows with long latencies and the middle has flow 2 with short latency and flow 1 with long latency. Each cluster depicts CATNAP, DropTail and SPQ performance. The DropTail AP provides fair capacity allocation in the short case, but yields unequal throughputs in the mixed and long cases. SPQ's prioritization gives unfair treatment in the short and long cases, but CATNAP treats both flows fairly regardless of the latency.

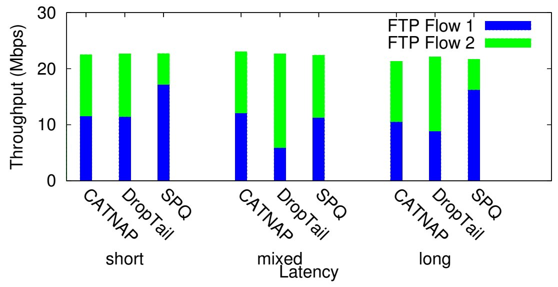

Fig. 9. Average Throughput for Two FTP Flows (54 Mbps link)

Figure 10 displays performance as the 10th percentile MOS for a VoIP flow running over UDP on an IEEE 802.11g AP. VoIP QoS degrades under DropTail with mixed and long round-trip times. While SPQ is configured to favor VoIP, CATNAP provides high MOS values regardless of the round-trip times and when facing congestion on a 11g AP from the background FTP flow.

The experiment shown in Figure 11 replaces the VoIP flow from Figure 10 with a network game flow. In this case, DropTail yields poor game quality compared to CATNAP and SPQ when competing with a short latency FTP flow. The game quality degrades severely under mixed and long latencies in all cases, however CATNAP is able to provide better QoS under mixed latencies and slightly better performance under long latencies.

Figure 12 displays the average response time for a Web flow to load a page with embedded objects. Regardless of the

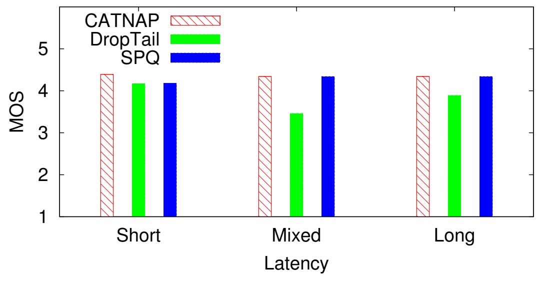

Fig. 10. 10th percentile of MOS for UDP-based VoIP (11 Mbps link)

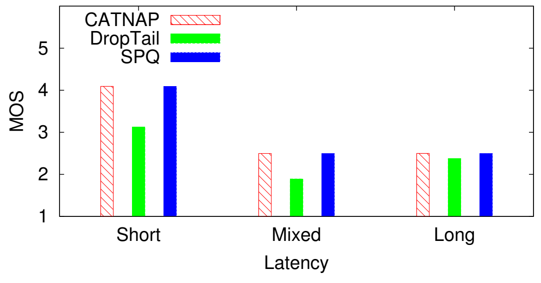

Fig. 11. 10th percentile of MOS for Network Game (11 Mbps link)

AP's queuing discipline, the long latency for the Web flow in both the mixed and long latency cases increases Web response times dramatically. DropTail has noticeably larger response times than SPQ or CATNAP in all cases. CATNAP produces performance similar to SPQ in mixed and long latency cases and yields the best response time in the short latency case.

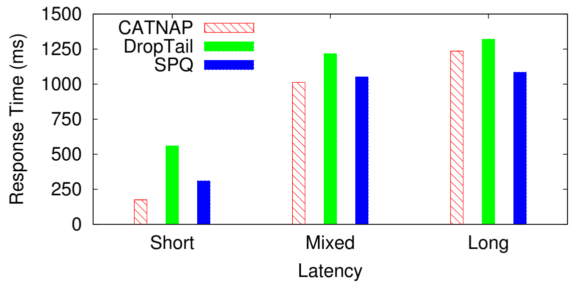

Fig. 12. Average Response Time for Web (52 Mbps link)

Table I (11 Mbps) and Table II (54 Mbps) provide a condensed summary to compare DropTail and CATNAP QoS results for many NS-2 experiments. Each cell in each row in both tables indicates the percentage improvement of CATNAP compared with DropTail for the appropriate QoS metric. Differences greater than 10% are shown in **bold**. The tables show that CATNAP always provides better performance than DropTail and produces particularly high benefits for network games and VoIP applications under higher latency conditions. While CATNAP greatly improves Web response times, the video QoS results are less impressive with CATNAP generating a significantly higher frame rate only under low capacity (11 Mbps) and mixed latency conditions.

TABLE I
QoS IMPROVEMENT FOR CATNAP OVER DROPTAIL (11 MBPS)

Foreground	Quality	Latency Settings		
		short	*mixed*	*long*
Game w/UDP	MOS (10% tail)	**32.6%**	**113.9%**	**93.7%**
Game w/TCP	MOS (10% tail)	**34.7%**	**117.7%**	**111.2%**
VoIP w/UDP	MOS (10% tail)	5.3%	**25.4%**	**11.6%**
VoIP w/TCP	MOS (10% tail)	5.2%	**29.6%**	**12.7%**
Video w/UDP	Fr Rate(10% tail)	**11.5%**	7.4%	**11.1%**
Video w/TCP	Fr Rate(10% tail)	7.4%	**20.0%**	3.4%
Web	Response Time (sec)	**54.6%**	**55.0%**	**57.9%**

TABLE II
QoS IMPROVEMENT FOR CATNAP OVER DROPTAIL (54 MBPS)

Foreground	Quality	Latency Settings		
		short	*mixed*	*long*
Game w/UDP	MOS (10% tail)	**58.1%**	**52.4%**	**15.7%**
Game w/TCP	MOS (10% tail)	**54.5%**	**55.6%**	**53.7%**
VoIP w/UDP	MOS (10% tail)	1.1%	1.8%	1.6%
VoIP w/TCP	MOS (10% tail)	0.2%	0.6%	0.9%
Video w/UDP	Fr Rate(10% tail)	0.0%	0.0%	0.0%
Video w/TCP	Fr Rate(10% tail)	3.4%	3.4%	0.0%
Web	Response Time (sec)	**13.6%**	**48.1%**	**56.4%**

V. CONCLUSIONS

This paper introduces Classification And Treatment iN an Access Point (CATNAP) as a mechanism designed to automatically classify and treat wireless network applications. Designed for low-end devices such as residential APs, CATNAP consumes a small amount of resources and requires no manual user configuration.

Extensive simulation experiments of CATNAP demonstrate it improves application QoS performance over a wide range of network conditions, especially benefiting network games and VoIP under high latency conditions. In all experiment cases, CATNAP provides better performance than DropTail and similar performance to SPQ, which is only effective with apriori knowledge and configuration.

Future work could include implementation and evaluation of CATNAP in an actual AP, perhaps using OpenWrt[1] to provide more detailed evaluation of the resource requirements for the CATNAP classification and treatment modules.

REFERENCES

[1] The Network Simulator - NS-2. Online at: http://www.isi.edu/nsnam/ns/, Last Access on April 10, 2014.

[2] G. V. D. Auwera, P. T. David, and M. Reisslein. Traffic and Quality Characterization of Single-Layer Video Streams Encoded with the H.264/MPEG4 Advanced Video Coding Standard and Scalable Video Coding Extension. *ACM Transaction on Internet Technology*, 54:698–718, 2008.

[3] C. Boutremans, G. Iannaccone, and C. Diot. Impact of Link Failures on VoIP Performance. In *Proceedings of the 12th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, pages 63–71, Miami, FL, 2002.

[4] M. Claypool, R. Kinicki, and C. Wills. Treatment-Based Traffic Signatures. In *Proceeding of IMRG (IETF Internet Measurement Research Group) Workshop on Application Classification and Identification (WACI)*, Cambridge, MA, October 2007.

[1]http://wiki.openwrt.org/

[5] A. Cricenti and B. Philip. ARMA(1,1) modeling of Quake4 Server to Client Game Traffic. In *Proceedings of the 6th ACM SIGCOMM Workshop on Network and System Support for Games*, NetGames, pages 70–74, Melbourne, Australia, 2007.

[6] J. Erman, A. Mahanti, and M. Arlitt. Internet Traffic Identification Using Machine Learning. In *Proceedings of the Global Telecommunications Conference (GLOBECOM)*, San Francisco, CA, November/December 2006.

[7] T. Guo, J. Cai, and C. H. Foh. Scalable Video Adaptation in Wireless Home Networks with a Mixture of IPTV and VoD Users. In *Proceedings of the Global Telecommunications Conference (GLOBECOM)*, Houston, TX, December 2011.

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

[9] F. Li. *Treatment-Based Classification in Residential Wireless Access Points*. PhD thesis, Worcester Polytechnic Institute, May 2014.

[10] S. Lynn. Apple AirPort Extreme Base Station (A1521), June 2013. On line at http://www.pcmag.com/article2/0,2817,2421124,00.asp, Last Access on May 1st, 2014.

[11] G. Maier, A. Feldmann, V. Paxson, and M. Allman. On Dominant Characteristics of Residential Broadband Internet Traffic. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference (IMC)*, Chicago, IL, 2009.

[12] A. McGregor, M. Hall, P. Lorier, and J. Brunskill. Flow Clustering Using Machine Learning Techniques. In *Proceedings of the 5th Passive and Active Measurement Workshop (PAM)*, Antibes Juan-les-Pins, France, April 2004.

[13] A. Moore and K. Papagiannaki. Toward the Accurate Identification of Network Applications. In *Proceedings of the 6th Passive and Active Measurement Workshop (PAM)*, Boston, MA, March/April 2005.

[14] M. Roccetti, C. Palazzi, S. Ferretti, and G. Pau. *Encyclopedia of Wireless and Mobile Communications*, chapter Wireless Home Entertainment Center: Protocol Communications and Architecture. Auerbach Publications, Taylor & Francis Group, London (UK), 2007.

[15] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield. Class-of-Service Mapping for QoS: a Statistical Signature-based Approach to IP Traffic Classification. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, Taormina, Sicily, Italy, October 2004.

[16] A. K. Singh and B. Mishra. Comparative Study on Wireless Local Area Network Standards. *International Journal of Applied Engineering and Technology*, 2(7), July 2012.

[17] A. Technologies. The State of the Internet, 3rd quarter, 2013 Report. Technical Report 3, Akami Technologies, Inc, 2013.

[18] A. Wattimena, R. Kooij, J. van Vugt, and O. Ahmed. Predicting the Perceived Quality of a First Person Shooter: the Quake IV G-model. In *Proceedings of the 4th ACM Network and System Support for Games (NetGames)*, Singapore, Oct. 2006.

[19] S. Zander, T. Nguyen, and G. Armitage. Automated Traffic Classification and Application Identification Using Machine Learning. In *Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary (LCN)*, pages 250–257, Sydney, Australia, November 2005.