

Queue management in Home Gateways: Potential impact on IoT traffic flows

Shahana Cumaranayagam*

Centre for Advanced Internet Architectures, Technical Report 160216A

Swinburne University of Technology

Melbourne, Australia

sahana.cis3@gmail.com

Abstract—In today’s fast-forwarding world, bandwidth has become a necessity with the crucially advanced Internet infrastructure, broadly delivering rapid growth to the number of IoT (Internet of Things). IoT has majorly contributed towards the increasing interconnectedness and remote access to devices. However, the need for bandwidth does not just stop with more bandwidth but requires reliable and resilient Internet service distribution. Our main focus is to compare and contrast the performance of various home-based services and to emulate a range of combinational network traffic parameters. For this we use CAIA’s high performance TEACUP testbed. The testbed allows us to emulate, the impact of different network conditions such as speed, latency, reliability and analyse the relationships between TCP flows with its bottleneck buffering, path BDPs and different AQM’s.

Index Terms—CAIA, TCP, AQM, fq_codel

I. INTRODUCTION

Conventional home gateways concurrently share overlapping home application traffic flows. These traffic flows consist of bandwidth-intensive applications that generate long lived bulk TCP transfers and interactive applications that generates comparatively smaller bursts or short lived traffic flows among multiple simultaneous users. The interactive traffic flows are mostly generated by IoT applications which tend to generate low bit rate traffic (such as home power monitoring systems, telemetry devices and so forth) that needs to be protected by the bulk transfers and cloud backups. In a typical home network common applications and their traffic flows are all bottle-necked through the same home gateway router, using certain amount of speed links depending on each households ISP. The congested single bottleneck router

*This work was performed while the author was an undergraduate summer intern under the supervision of Professor Grenville Armitage and Dr Hong Suong Nguyen.

being shared by all traffic of the home gateway causes speed imbalance resulting in high network congestion.

In this report we propose a feasible solution that provides prioritisation to the low bandwidth application such as IoT traffic, through flow segregation/ traffic isolation by implementing active queue management algorithms at the physical routers of a home gateway network. We find that implementing fq-codel algorithm at the bottleneck router can reduce the delay of low-rate IoT flows significantly in comparison with pfifo.

The rest of this paper is structured as follows. In Section II we outline the problem background and motivation of the paper. In this section we give a brief overview of TCP flow behaviour, bottleneck buffering, the paths bandwidth delay product (BDP) and illustrate the difference between two different queue management algorithms, pfifo and fq_codel. We show how using these queue management schemes improve or degrade TCP flows performance through the bottleneck router. Section III explains the simulation set-up of the testbed and the combination of simulations used in the experiments. The results of the experiments are reviewed in Section IV and Section V concludes the report with summarising the key findings and evaluation of the preferred queue management algorithm.

II. PROBLEM BACKGROUND AND MOTIVATION

The objective is to provide good end user experience over the Internet when these regular TCP connections collide with each other inside a home gateway. The primary concern compounded on networks that uses TCP/IP schemes is protecting lower bandwidth requiring interactive applications such as VoIP, online games and telemetry devices from bandwidth intensive applications such as HD television or video streaming. The challenge of bearing both traffic flows in the same router/link is greatly reliant on and influenced by many factors [1]. The

coexistence of any given traffic is greatly manipulated by connection latency, packet loss, link speed, bottleneck buffering and queuing impositions. We revisit the characteristic and behavioral patterns of the aforementioned concepts to draw a clear understanding of the problem.

A. TCP flow behaviour

Transmission control protocol is a connection-oriented protocol used in traffic flows with a range of application behaviors. It possesses the key responsibility of carrying traffic across the IP layer while ensuring to provide nominal network congestion. TCP is well acknowledged for its robustness in delivering bulk distributed and diverse traffic, reliably and securely through a network. TCP's intertwined congestion control algorithm, defined by its performance characteristics such as slow start, congestion avoidance [2], fast re-transmission and fast recovery contribute towards optimisation of a networks performance. TCP uses sliding window mechanism, which encompasses two windows, Congestion window and receive window.

TCP's limitation on window sizes act as source of problems. TCP's performance starts to drop considerably when multiple number of completing flows share the same congested bottleneck [3]. This further lead to substantial impacts such as packet loss causing the congestion window to half its window size resulting in noticeable degrade in performance.

B. Bottleneck buffering

Bottleneck typically occurs when the buffer inside a network has traffic arriving at a higher rate than it can depart. If substantially large amount of packets are retained in a path, the congestion will lead to network performance degradation and speed imbalance. If such an imbalance between arrival and departure rates tend to come in brief bursts, packet loss can be avoided by putting the excess arrivals into a buffer. This allows the system to process the information according to its capacity since there will be more buffer space available for packets to be stored until being processed. The packets in the buffer will then be processed at the departure rate of the outbound link. However the packets that arrived later are delayed by the time it takes to process the packets ahead of them in the queue.

Buffers are built into bottleneck routers and switches to absorb transient bursts of traffic arriving faster than the rate at which the router or switch can forward. When the buffering is too small, long/sharp arrival bursts of traffic may lose packets. But packet losses are often needed to

enable the TCP sender to estimate the traffic carrying capacity of the path. TCP sender will keep putting more unacknowledged packets in-flight until it sees a loss. When products ship with very large bottleneck buffers, TCP senders can put significantly more packets in flight, than packets that are strictly necessary for 100% utilisation of the path. The existence of buffers that are in excess of requirements is 'bufferbloat' [4] and the consequences are significantly excessive when any additional delay for packets passing through a bottleneck whose (overly-large) buffer has lots of packets queued up inside at any given time.

C. The bandwidth delay product

The BDP measures the amount of data that a protocol should have "in flight" in order to fill a network pipe. It is the buffer size deciding factor and has significant impact on throughput. TCP's sender and receiver must be capable of handling and keeping larger number of packets (unacknowledged data) in flight to obtain maximum throughput on the TCP connection over a path. BDP is calculated by multiplying the path Bandwidth with the imposed Delay. More information about BDP and its effect on competing traffic can be found in [5].

D. Choice of Active Queue Management

Deployment of queue management algorithm has proved to enhance network performance through significantly reducing network latency over the Internet path. AQM is a combination of two related algorithms dedicated to perform controlling of queue lengths, by managing or dropping packets adequately and scheduling the order of when and which packet should be sent next [6]. The goal here is to allow bandwidth to be shared equally between flows. In our experiment we investigate and compare the performance of pfifo and fq_codel queue management algorithms.

PFIFO (Packet First In First Out) allows en queuing until the currently queued number of packets exceeds the configured limit. This algorithm is used to deal with the issues associated with buffer overflow. However, buffer filling in the presence of excess buffering causes high RTT spikes and poor overall network performance.

Fq_codel algorithm is a combination of both a scheduler and queue management scheme. It is based on a modified Deficit Round Robin (DRR) queue scheduler, with the CoDel AQM algorithm operating on each subqueue [7]. The "fq_" part of fq_codel makes an explicit attempt to share capacity equally. Fq-codel mixes multiple flows of packets and uses its hashing functionality to

separate each flow into segregated queues, this provides scheduled capacity sharing of flows while reducing the impact of head-of-line blocking [8] and its subsequent peak throughput rates incurred by bursty traffic [5].

Fq_codel provides isolation for low-rate traffic used in a daily basis to connect IoT's, such as VoIP, web and medical telemetry traffic. The isolation of low rate traffic helps maintaining shorter queue lengths and low path latency in networks.

III. TESTBED TOPOLOGY AND SIMULATION SETUP (TECHNICAL REQUIREMENTS)

To emulate typical home gateway scenarios, CAIA's TEACUP (TCP Experiment Automation Controlled Using Python) testbed was used. The testbed topology in Figure 1 consists of several host machines, which connect to a PC-based router and belong to either of two different subnets. Further comprehensive description of the configurations and system information of the testbed can be found in [9]. TEACUP allows to emulate different network conditions by configuring different delay, loss, bandwidth, AQM and bottleneck buffer size at the router.

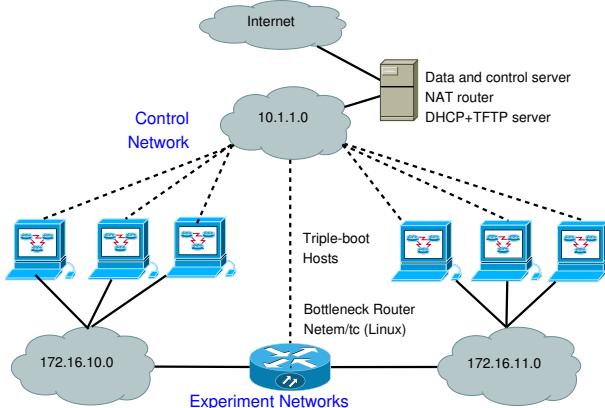


Figure 1. TEACUP testbed topology [10]

The experiments summarised in this report were conducted for the scenario of three overlapping upload TCP flows where the effect of different AQM's can be clearly seen. Besides, the starting time of three flows are staggered so that the interaction between flows and their transient behaviours can be observed. Three TCP flows were generated using iperf tool with three hosts in one subnet acting as clients and three hosts in the other subnet acting as servers. The bottleneck router in each experiment was Linux-based (Linux version 3.17.4) while all the hosts ran FreeBSD version 9.1 operating system. TEACUP was configured to emulate network base RTT of 40ms and loss rate of 0. Besides,

two bottleneck bandwidths are considered, which are 50 Mbps down /20 Mbps up and 4 Mbps down /1 Mbps up. Hereinafter the bandwidth speeds will be represented by "X/Y Mbps" where X is downstream speed and Y is upstream speed in this report. The router was configured to have the buffer size of 90 packets, which is calculated using the largest down link 50 Mbps and delay of 40ms for margin of safety. An iperf window size of 300000 bytes was used as a limiting factor. Each trial was run for 100s.

Moreover, to illustrate the impact of different queue managements on IoT applications, we also ran experiment for the scenario with 3 staggered TCP flows and two low-rate UDP flows representing IoT flows. The bottleneck bandwidth was configured to be 4/1 Mbps in this scenario.

IV. RESULTS - UNDERSTANDING THE OUTPUT

This section will illustrate how the choice of AQM on three FreeBSD NewReno flows sharing the same bottleneck link, influences the throughput and overall SPP RTT (all RTT measurements are calculated by TEACUP using SPP) of the path over time.

A. 50/20 Mbps and 4/1 Mbps pfifo - Throughput vs Time

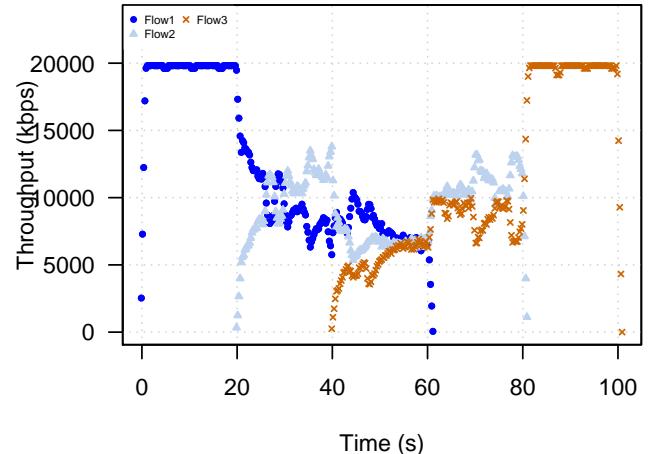


Figure 2. Throughput vs Time of 50/20 Mbps - pfifo

Figure 2 illustrates the Throughput vs Time graph of the three-flow traffic path over 40ms RTT. In here the three flows share the bottleneck capacity very unevenly. This is clearly noticeable at the time interval 20s - 80s when the three flows crudely overlap with each other. The throughput sharing worsens noticeably once flow 3 joins at time $t=40s$.

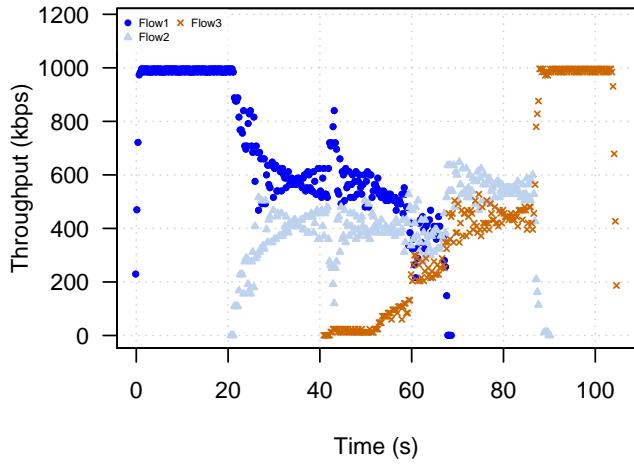


Figure 3. Throughput vs Time of 4/1 Mbps - pfifo

In Figure 3 with the lower bandwidth rate of 1 Mbps link, similar throughput fluctuation pattern is noticeable. Initially when only one flow is present, flow 1 reaches full capacity until flow 2 starts, as seen in $t=20s$. Thereafter throughput sharing becomes prominently uneven. The throughput sharing further worsens once the third flow joins the two existing flows. A similar effect is noticeable when first two flows cease at approximately $t=90s$, where the third flow is able to utilise full capacity.

B. 50/20 Mbps and 4/1 Mbps pfifo - RTT vs Time

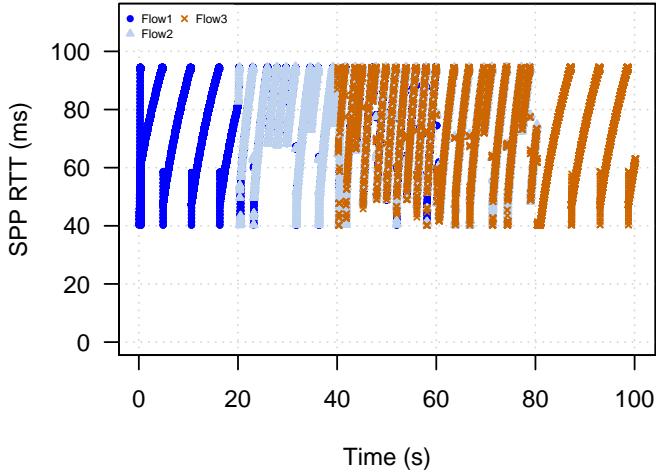


Figure 4. RTT vs Time of 50/20 Mbps - pfifo

Figure 4 illustrates very high RTT swings causing rapid periodic cycles of RTT. The high RTT swings are caused due to queuing of packets. Since all traffic share the dominant congested single bottleneck router,

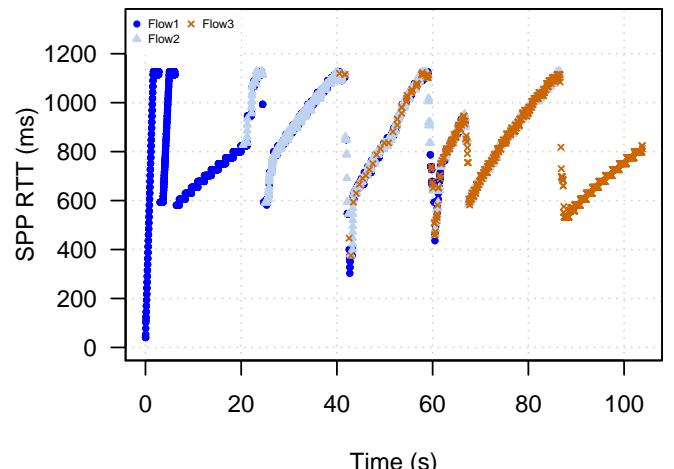


Figure 5. RTT vs Time of 4/1 Mbps - pfifo

significant queuing delays observed cause high RTT swings. The bottleneck buffer of 90 packets further endues filling of the buffer space and rapid RTT cycles. Figure 5 RTT vs Time graph demonstrates RTT cycling between $\sim 600\text{ms}$ and $\sim 1100\text{ms}$, with the round trip time of each flow closely following each other.

C. 50/20 Mbps fq_codel - Throughput vs Time

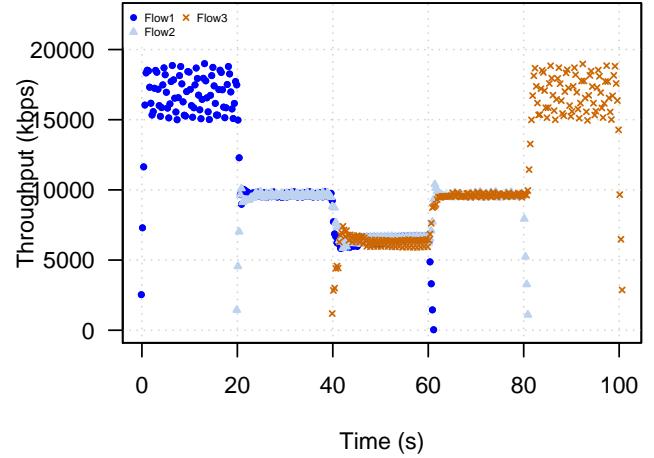


Figure 6. Throughput vs Time of 50/20 Mbps - fq_codel

Figure 6 illustrates the bottleneck results of the flow where quite consistent throughput and balanced capacity sharing is achieved. Fq_codel's better capacity sharing is evident during the instance where all three flows start overlapping from time $t=40s$, where sharing is made completely smooth.

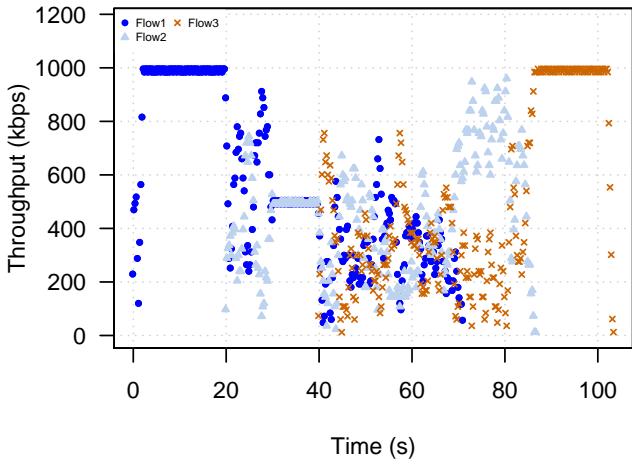


Figure 7. Throughput vs Time of 4/1 Mbps - fq_codel

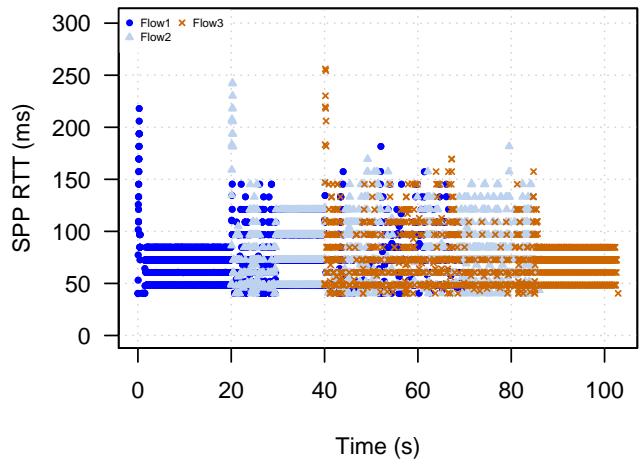


Figure 9. RTT vs Time of 4/1 Mbps - fq_codel

D. 4/1 Mbps fq_codel - Throughput vs Time

The throughput graph of the 4/1 Mbps bandwidth link shown in Figure 7 depicts rather fluctuating throughput over time. The cause of this fluctuation has not been known, thereby it is not discussed further in this report. However, it could be noticed that at time intervals 0s to 20s, 30s to 40s and 90s to 100s the capacity is smoothly shared among the competing flows making fq_codel's queue management sustainable than the pfifo's rapid swings of throughput and RTT.

E. 50/20 Mbps and 4/1 Mbps fq_codel - RTT vs Time

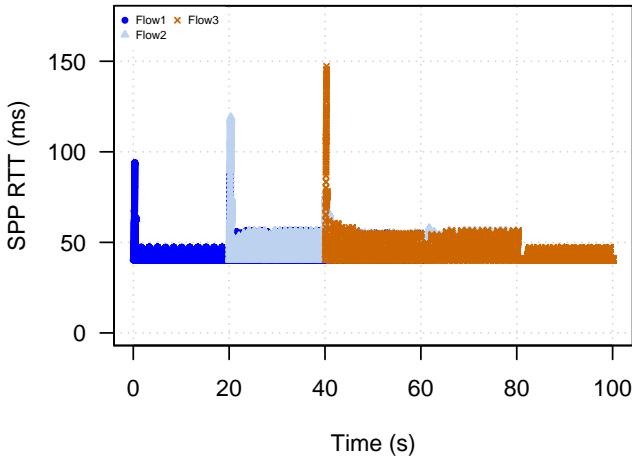


Figure 8. RTT vs Time of 50/20 Mbps - fq_codel

As seen in both bandwidths, RTT vs Time graphs (Figures 8 and 9) it could be observed that there is a significant RTT spiking immediately after the start of each

flow, this is quickly adjusted and the RTT's are brought back to approximately 100ms down than what it was originally. Such initial high RTT spikes are caused due to TCP's slow start mechanism and fq_codel's tolerance of bursty traffic. Towards the very end of each stream the base RTT of the traffic stabilises with a difference of approximately 10ms for Figure 8 and 30ms difference for Figure 9 resulting in significantly reduced bottleneck queue delays. This proves that fq_codel has better control over delay, while being insensitive to round-trip link and traffic loads compared to pfifo's.

F. Effect of AQMs on IoT flows - RTT vs Time

As illustrated by Figures 10 and 11, we have introduced a bi-directional UDP flow that imitates a low data rate IoT application. Applications such as VoIP generally works through a systematic pattern of pushing out small amount of data at low bit rates, however such applications sequentially push data almost at a continuous phase and the cycle has to be maintained without interruptions. These types of traffic flows are mandatory for medical telemetry applications, where acquisition of data from a source and the consequent transmission of the sequential data to its destination. Protecting packet loss while maintaining the RTT swings is a paramount requirement for such flows.

We could observe that over a path of 20ms base RTT and 4/1 Mbps bottleneck using pfifo queue management as illustrated in Figure 10, resulted in perusing high swings of RTT. There is no segregation for low rate traffic, leading to equal weight given to all flows despite being low or high in bandwidths. On the other hand using the fq_codel scheme as shown in Figure 11, it could be

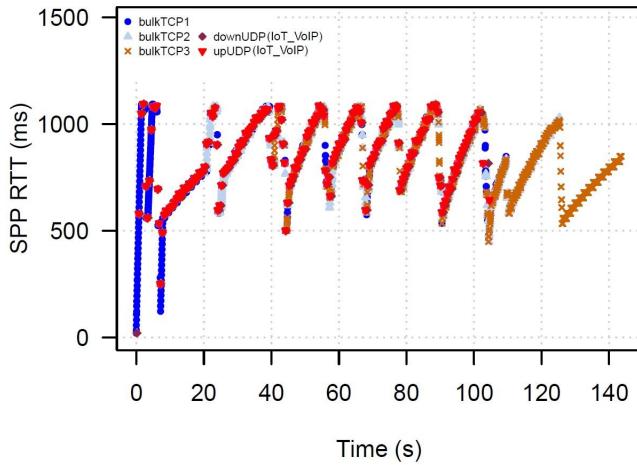


Figure 10. RTT vs Time of 4/1 Mbps - UDP flow - pfifo

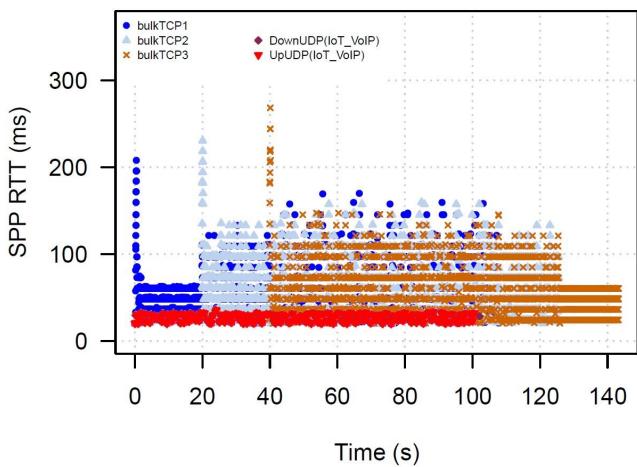


Figure 11. RTT vs Time of 4/1 Mbps - UDP flow - fq_codel

observed that the hashing of the fq_codel algorithm has distinctly isolated the emulated IoT VoIP traffic flow into separate queue. Whilst providing even capacity sharing and secure path. Further fq_codel algorithm has allowed low rate traffic flows have priority queuing, making bulk TCP traffic and low rate UDP traffic co-exist in the same bottleneck.

V. CONCLUSIONS

The aim of this report was to investigate how the choice of queue management algorithm alters the experience of IoT traffic through a conventional home gateway, sharing home application traffic. The queuing of fq_codel utilises isolated queues to more readily distribute available bandwidth between traffic flows. Fq_codel's packet drop mechanism makes sure that the packets being dropped are from the largest flow available. This provides additional security to interactive real-

time IoT type traffic flows, so that these type of traffic are provided with the network update before any other existing TCP flows [11]. In addition to prioritisation of packet loss and path isolation the low rate flows are also given the intensive of significantly improving the behavior of the network when new flows commence [7].

However it was noticed that with low bandwidth connection such as the 4/1 Mbps flow the throughput graph was still noisy but not to the extent of the pfifo resultant throughput. Thus it could be concluded that fq_codel is a plausible queue management algorithm that has the potential to alter the experience of IoT traffic through a conventional home gateway sharing.

ACKNOWLEDGEMENTS

I would like to acknowledge the extended support and guidance of Prof Grenville Armitage who has given me constructive comments and feedback throughout my internship. I would like to thank Dr Hong Nguyen for her support with the testbed configurations and her valuable comments. I would also like to thank Jonathan Kua for his version of the TEACUP testbed topology and other CAIA members for their help and guidance.

REFERENCES

- [1] N. Khademi, M. Welzl, G. Armitage, C. Kulatunga, D. Ros, G. Fairhurst, S. Gjessing, and S. Zander, "Alternative Backoff: Achieving Low Latency and High Throughput with ECN and AQM," Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, Tech. Rep. 150710A, 10 July 2015. [Online]. Available: <http://caia.swin.edu.au/reports/150710A/CAIA-TR-150710A.pdf>
- [2] M. Welzl and W. Eddy, "Congestion Control in the RFC Series," RFC 5783 (Informational), Internet Research Task Force, Feb. 2010. [Online]. Available: <http://www.ietf.org/rfc/rfc5783.txt>
- [3] R. Morris, "Tcp behavior with many flows," in *Network Protocols, 1997. Proceedings., 1997 International Conference on*, Oct 1997, pp. 205–211.
- [4] J. Gettys and K. Nichols, "Bufferbloat: Dark buffers in the internet," *Queue*, vol. 9, no. 11, pp. 40:40–40:54, Nov. 2011. [Online]. Available: <http://doi.acm.org/10.1145/2063166.2071893>
- [5] G. Armitage, "Three staggered-start TCP flows through PIE, codel and fq_codel bottleneck for TEACUP v0.4 testbed," Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, Tech. Rep. 140630A, 30 June 2014. [Online]. Available: <http://caia.swin.edu.au/reports/140630A/CAIA-TR-140630A.pdf>
- [6] F. Baker and G. Fairhurst, "IETF Recommendations Regarding Active Queue Management," RFC 7567 (Best Current Practice), Internet Engineering Task Force, Jul. 2015. [Online]. Available: <http://www.ietf.org/rfc/rfc7567.txt>
- [7] T. Hoeiland-Joergensen, P. McKenney, D. Taht, J. Gettys, and E. Dumazet, "Flowqueue-codel," IETF Draft, <https://tools.ietf.org/html/draft-ietf-aqm-fq-codel-03>, November 2015. [Online]. Available: <https://tools.ietf.org/html/draft-ietf-aqm-fq-codel-03>

- [8] V. Puente, J. A. Gregorio, R. Beivide, and C. Izu, "Impact of the head-of-line blocking on parallel computer networks: Hardware to applications," pp. 1222–1230, 1999. [Online]. Available: http://link.springer.com/chapter/10.1007/3-540-48311-X_173
- [9] S. Zander and G. Armitage, "TEACUP v1.0 - A System for Automated TCP Testbed Experiments," Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, Tech. Rep. 150529A, 29 May 2015. [Online]. Available: <http://caia.swin.edu.au/reports/150529A/CAIA-TR-150529A.pdf>
- [10] ——, "CAIA Testbed for TEACUP Experiments Version 2," Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, Tech. Rep. 150210C, 10 February 2015. [Online]. Available: <http://caia.swin.edu.au/reports/150210C/CAIA-TR-150210C.pdf>
- [11] H. Tschofenig, L. Eggert, and Z. Sarker, "Report from the IAB/IRTF Workshop on Congestion Control for Interactive Real-Time Communication," RFC 7295 (Information), Internet Architecture Board, Jul. 2014. [Online]. Available: <https://tools.ietf.org/html/rfc7295.txt>