Household internet and the ‘need for speed’: evaluating the impact of increasingly online lifestyles and the internet of things

Grenville Armitage*, Jenny Kennedy‡, Suong Nguyen†
Julian Thomas‡, Scott Ewing§

Centre for Advanced Internet Architectures, Technical Report 170113A
Swinburne University of Technology
Melbourne, Australia

***Abstract*—The present and future broadband needs of households remains a contentious topic, with political, industry and consumer stakeholders rarely in agreement. We illuminate the policy debate by drawing upon both social and technical studies to consider what happens in household networks. We identify the influence of latency-sensitive and latency-tolerant classes of application traffic in driving consumer perceptions of how fast their last-mile broaband services need to be. Debate on household bandwidth requirements frequently focuses on applications such as streaming TV services. We identify another indirect driver as poor end-user experiences with interactive and internet of things applications using lower speed broadband access services. Using experimental results we illustrate how current home gateway technologies fail to protect latency-sensitive applications when sharing with latency-tolerant traffic, and conclude that adding emerging active queue management (AQM) techniques to home gateways can address the needs of latency-sensitive applications, with the benefit of a reduction in the need for more expensive, additional access bandwidth. Drawing on ethnographic fieldwork on household rhythms we show how household demographics can be used to estimate future bandwidth requirements.**

CONTENTS

*garmitage@swin.edu.au, School of Software and Electrical Engineering, Swinburne University of Technology, Australia
†nhsuong@gmail.com
‡{jenny.kennedy,julian.thomas}@rmit.edu.au, School of Media and Communication, RMIT University, Australia
§sewing@swin.edu.au, Swinburne Institute for Social Research, Swinburne University of Technology, Australia

I. Introduction

One of the more contentious topics in Australian broadband policy is the present and future needs of households for higher bandwidth services. The question has been the topic of a long debate, in industry, governmental, academic and consumer circles, with ramifications for the larger communications sector. There are of course a wide range of policy instruments available to governments wishing to improve communications. However, in recent years these arguments have taken on a more polarised, political dimension.

If we think that higher bandwidth services are likely to be essential in the near future, then we may be more likely to agree to bear the additional costs of a fibre-to-the-home (FFTH) network. Skepticism as to the value of high speed internet may justify a more incremental adaptation of new and existing infrastructure, sometimes referred to as a 'mixed technology' network.

In the meantime, while arguments over the national broadband network (NBN) continue, broadband technologies are developing rapidly, with the emergence of a new, more complex ecosystem of domestic connections. We see three core areas of change:

- Mobile services are now an alternative and complementary mode of access for large numbers of Australians
- Connected devices and applications have also proliferated both within and beyond households
- Traffic is increasingly a mix of both latency-sensitive applications (such as interactive remote/online education services) and latency-tolerant applications (such as content streaming and data sharing services)

In this report we aim to illuminate the policy debate by considering more closely what happens in household networks, drawing upon both social and technical studies modelling household applications and their uses. Questions include:

- How, and to what degree, are household internet connections a constraint on internet use, including for applications such as online education, medical telemetry and life-style/home automation (the *Internet of Things*)?
- How, and to what degree, are speed demand predictions skewed higher by a perceived need to protect the Quality of Service (QoS) experienced by latency-sensitive services when using current gateway technologies?
- What do users perceive to be indicators of poor interactive service, and when, and to what degree, do they occur?

The report is structured as follows: Section II describes three broad categories of consumer traffic whose interactions drive bandwidth demands. Section III describes how demand for higher bandwidths can be a side-effect of seeking low latency for interactive applications. Section IV summarises emerging analyses of Australian household demographics and application profiles that drive demand for both low-latency and high-performance broadband services. Section V elaborates on the home gateway's role as a key traffic congestion point, demonstrates how emerging *active queue management* (AQM) technologies enable better multiservice support, and discusses challenges in deploying AQM-enabled gateways. Section VI makes some detailed bandwidth requirement estimates based on selected population demographics, application requirements and possible future deployment of AQM in gateways. The report concludes in Section VII.

Our experimental methods for measuring the bandwidth consequences of different use-scenarios and technologies are described in detail in Appendices A to D.

II. Three categories of consumer traffic

A range of network and end-host characteristics impact on the end-user's perception of Internet quality or performance. Figure 1 captures the modern reality of an increasingly diverse range of applications competing for a share of each home's broadband link to the Internet.

Fig. 1: Traditional, interactive and emerging IoT traffic collide at the home gateway – congestion causes traffic queues and delays, driving perceived need for higher access link speeds

For example, a typical home-based remote education service (such as participation in online interactive tutorial groups, etc) will share the home broadband service with a range of other common applications (such as streaming TV, smartphones/tablets app updates/downloads, smartphones and cameras syncing photos to 'the cloud', Skype and similar two-way video conferencing and online games). The challenge occurs when a mix of traffic overlaps in time, whether due to explicit action of the end-user(s) or background activities launched by in-home devices without end-user intervention.

To better estimate broadband speed requirements, we map diverse consumer applications into three broad categories: Latency-sensitive / interactive traffic, latency-tolerant (*elastic*) traffic, and streaming content traffic. Consumer experience of poor interactive service is often due to the way current home gateway designs allow mutual interference between traffic belonging to each of these three categories.

Significantly, concurrent sharing of a home's internet connection can dramatically increase the latencies experienced by all traffic in and out of the home. Whether brief or long-lived, competition for last-mile resources degrades the quality of interactive services in use at the same time.

The total bandwidth requirements depend heavily on the consumer's tolerance of such mutual interference, and how often such mutual interference is likely to occur. In this section we discuss these categories in more detail, and describe how they blend together to create demand on home broadband connections.

A. Latency-sensitive / interactive traffic

Latency-sensitive traffic covers a broad range of applications where timeliness of information transfer is important. Some of these applications are continuously interactive in nature, involving a human at one or both ends of the network path, generating and consuming data sent over the network in real-time. A non-exhaustive list of examples include:

- Voice over IP (VoIP, which includes ISP-supplied Internet Telephone products)
- Multi-party voice/video conferencing (such as Skype, Facetime, and remote education services)
- Online games (particularly 'twitch' games like First Person Shooters, or other highly interactive immersive environments)
- Real-time, remote medical monitoring services

Continuously interactive applications typically involve (relatively steady) streams of packets between two points on the network, at modest data rates dictated (and limited) by the applications themselves.[1] It is usually far more beneficial for a network to exhibit low *round trip time*[2] (RTT) than to offer bandwidth in excess of what the application typically needs.

A related class of sporadically interactive applications generate short bursts of traffic infrequently, but benefit from low RTT when human users are triggering events and waiting for responses. Examples include cloud-based Internet of Things (IoT) / Home automation services. This is a relatively new, but rapidly growing segment currently typified by products like Google Home, Apple's HomeKit, Amazon's Alexa, and 3rd-party programmable automation platforms such as 'If This Then That'.[3]

The quality of our Internet experience also relies on behind-the-scenes, latency-sensitive services, such as:

- Resolving domain names into numeric IP addresses using domain name system (DNS) lookups [1]
- The *three way handshake* while setting up new transmission control protocol (TCP) connections [2]

DNS lookups occur frequently during normal home Internet use, and usually involve a host inside the home talking to a DNS server on the ISP side of the home gateway. For example, when you click on a new link on a web page your browser does a DNS lookup to determine the target site's actual IP address before it connects and attempts to retrieve the remote site's content. Any slow-down of DNS lookups makes web browsing more tedious and less responsive.

Similarly, as RTT goes up it takes longer to exchange the control (signalling) packets for establishing and tearing down TCP connections. Services that rely on TCP connections (like web browsing, sending emails, starting new file downloads or uploads, and so forth) can feel far less responsive when a home's RTT to the outside world is inflated by congestion in the home gateway.

B. Latency-tolerant (elastic) traffic

Latency-tolerant traffic covers a broad range of applications that tend to be more concerned with efficient use of bandwidth between two ends of a network path, and less concerned about the precise RTT experienced throughout the lifetime of any particular data transfer. We also refer to this sort of traffic as *elastic*, in that the application generating the traffic is flexible in terms of RTT and tolerates slowing down or speeding up as dictated by available bandwidth. A non-exhaustive list of examples include:

- Photo and video sharing applications sync'ing content to/from "the Cloud"
- Web browsers retrieving embedded digital objects to render pages
- Sending and receiving emails with large attachments
- Peer to peer file transfer applications
- Remote/offsite backup systems
- Application update systems (such as triggered by Microsoft Windows updates, or iOS Android App Store updates)
- Downloading podcasts, movies, TV show episodes or music tracks in their entirety for later, offline playback
- Instant messaging / notifications with multimedia attachments (Twitter, Apple's iMessage, and so forth)

Applications generating elastic traffic tend to be judged more on *time to completion* (TTC) – how long to upload or download a podcast, photo or app update, and so forth. TTC goes up as the size of objects being transferred goes up and/or the bandwidth of a path goes down. Consequently, bandwidth requirements for elastic applications depends on consumer tolerance for TTC, which in turn depends on the social context within which a particular application finds itself used.

For example, consider downloading a 60-minute podcast for later playback. TTC would be the time it takes for the podcast to be sync'ed and your portable device becomes ready to be taken offline. One consumer might consider 15 minutes an acceptable TTC for this podcast, while another consumer considers any TTC longer than 6 minutes to be unacceptable. Similarly, when pushing raw photos to "the cloud" one consumer might be happy with a TTC of 30 seconds per photo while another demands (or at least wishes for) less than 5 seconds per photo.

Elastic applications often use TCP as their underlying transport, resulting in two key issues. First, unless limited by the application itself, elastic traffic will consume as much bandwidth as TCP can extract from the network at any given instant. This ends up *causing* the home gateway congestion that (as noted next in Section III) increases RTT for everyone

[1] For example, video codecs generate data at variable rates, but no faster than required to encode the content given the chosen encoding algorithms.
[2] Time taken by a message to go back and forth between two points
[3] https://ifttt.com/

sharing that gateway. Second, TTC can degrade as RTT increases because (a) as noted previously, new TCP connections take longer to start-up, and (b) recovery from packet losses on active TCP connections takes longer, at times causing brief slow-downs and at other times resulting in multi-second stalls as competing TCP connections step in an briefly consume all of a path's capacity.

C. Streaming content traffic

This category covers multimedia content delivery where content is 'streamed' across the network just fast enough to be consumed by multimedia clients in real-time, but where interactivity is relatively limited. Examples include:

- Internet TV and movie services (such as Netflix)
- Internet radio services

Streaming services are typically only interactive (and latency sensitive) when consumers are selecting content to consume (analogous to channel surfing). Once a piece of content has been selected and the stream has begun playing, the service can be more relaxed about variations in network latency.

The average bandwidth requirements of streaming services can be estimated from the audio and/or video encoding rates of the content being streamed. However, depending on choice of streaming technology, the short-term behaviour of streaming traffic can have distinctly aggressive characteristics akin to repeated bursts of short-lived elastic traffic.

Services built on technologies such as DASH (Dynamic Adaptive Streaming over HTTP) [3] will usually begin a stream by pulling down tens of seconds of content as fast as possible, then settling into a regular pattern of short bursts of data traffic as the client retrieves 'chunks' of content piece-meal from the server over time. In addition, DASH-like services will usually adapt the content quality (and hence size in bytes) of newly requested chunks depending on the speeds achieved while retrieving previous chunks. Commonly deployed on top of conventional TCP, such traffic causes periodic short bursts of congestion on the home broadband link as each new content chunk is retrieved.

D. Combining the bandwidth needs of each category

Today's consumers are likely to be dissatisfied by an internet service offering (a) poor streaming quality, (b) long TTC for important elastic applications, and/or (c) high latencies for latency-sensitive applications. Our goal is to provide sufficient downstream and upstream bandwidth to ensure satisfactory service during periods where applications from all three categories are simultaneously active.

Streaming applications only *require* enough bandwidth to ensure that, over tens of seconds, the average bandwidth supports the video and audio quality desired by the consumer. This makes streaming traffic an attractive category on which to base rough bandwidth estimates. We can put reasonably concrete numbers on, for example, the number of MBytes it takes to stream TV or DVD-quality movie content per hour using common encoding and compression schemes. However, most streaming does not (currently) result in smooth traffic flow. Rather, every few seconds the home gateway is hit with short bursts of elastic-like traffic consuming as much spare bandwidth as is available at the time, each time.

Elastic applications are more complicated. They only *require* enough bandwidth to achieve an acceptable TTC. But by using TCP for transport, elastic applications will typically consume whatever *extra* bandwidth happens to be available at the time, leading to even shorter TTC than the user may need. What constitutes an acceptable TTC (and hence minimum bandwidth requirement) depends greatly on the application itself, a consumer's usage-patterns and their sensitivity to the financial cost of obtaining higher broadband speeds.

Interactive applications can appear to have only modest or low bandwidth requirements, provided there is little RTT inflation due to competing traffic. Yet, as discussed next in Section III, the need for low RTT while coexisting with elastic traffic can drive bandwidth requirements far higher when using conventional home gateways.

III. The need for speed

Public discussion of bandwidth has traditionally been dominated by latency-tolerant and streaming content service, due to the deceptively intuitive relationship between such applications' requirements for bandwidth and the broadband speeds offered by network providers. Here we argue that the consumer sensitivity to the first category should also be considered a significant driver for higher bandwidth demands.

A. Importance of latency for interactive services

The user's perception of an interactive application's service quality is strongly influenced by how long actions take to complete – such as clicking a web link, changing channels on an IP TV console, trying to converse interactively with participants of a video conference call, and so forth. The responsiveness of these applications is in turn strongly impacted by the RTT of the network path between the client (inside the home) and the remote server (elsewhere on the Internet).

RTT is made up of transmission or propagation delays (dictated largely by geography and physical distance) and queuing delays (due to temporary buffering at congested network routers and switches along the path). The home gateway is a key point where severe congestion can trigger significant addition queuing delays, and hence spikes in RTT.

Table I shows some typical RTTs in milliseconds (ms) that might be experienced by home-based applications in Melbourne, Australia, when accessing remote services over an otherwise idle home gateway (also referred to as RTT_{base}). The quality of interactive or latency-sensitive services improves when remote servers are closer.

However, current consumer home gateways are very easily congested (for example, by something as simple as an iPhone pushing multi-MByte photos to the iCloud, or Dropbox sync'ing a large file from a home PC to the central Dropbox server). When congested, the RTT to sites on the Internet can suddenly increase by 100s or even 1000s of milliseconds, depending in the particular brand of gateway. Furthermore, with

TABLE I: Ballpark RTTs from a home in Melbourne

RTT_{base}	Illustrative of connecting to....
10ms	domestic, intra-ISP servers
40ms	domestic, inter-ISP or off-net servers
180ms	US West Coast servers
240ms	US East Coast servers
340ms	European servers

most consumer home gateways this increased RTT impacts all traffic that happens to be passing through them at the time. A website located in Melbourne can suddenly *feel* like it is located in Paris. Or beyond the moon!

B. RTT inflation and home gateway buffering

RTT inflation during heavy traffic load is due to the interaction between conventional home gateways using first-in-first-out (FIFO) queues and many data-intensive applications (including web surfing, multimedia streaming and file/photo backup or publishing services) using the Transmission Control Protocol (TCP)[4] for reliable data transfer.

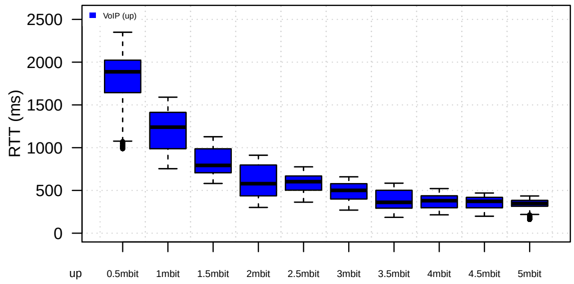
The Internet requires certain amounts of buffering (queue storage) in routers and gateways to absorb transient bursts of traffic. But when bulk data transfers cause long-lived or cyclical queue build up, the conventional FIFO queue architecture ensures *everyone* gets backed-up inside the queue, with worst case wait time (queuing delay) up to $\frac{BufferSize}{LinkSpeed}$ seconds.

Over the last decade many router and gateway vendors have focused on minimising packet losses by TCP connections, and erred on the side of including significant amounts of buffering – often referred to as *bufferbloat* [4], [5]. This leads to high, and often excessive, RTTs for all traffic during periods of congestion.

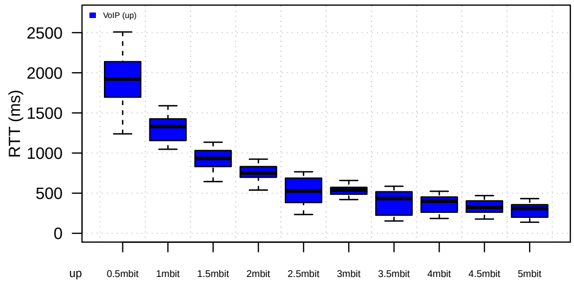
C. Higher connection speeds reduce RTT

During times of congestion, experienced RTT is higher through gateways with more buffer space, and lower as broadband link speeds go up in the congested direction. Doubling a link's speed will halve the additional RTT, and interactive applications seem to 'work better' as link speeds go up.

We illustrate this effect in Figure 2. Here we measure the actual RTT experienced by two-way VoIP traffic that is competing across a home gateway with an application inside the home pushing data to Cloud-based servers (and hence congesting the upstream link with a bulk, elastic data transfer).[5] We vary the hypothetical offered upstream speeds from 0.5 to 5Mbps for downstream rates of 12Mbps (Figure 2a) and 25Mbps (Figure 2b). In each case we see extremely high VoIP flow RTT for low upstream speeds, and a distinct drop in VoIP flow RTT as upstream bandwidth increases. Varying the downstream speed has little impact (because only the upstream



(a) VoIP flow RTT, fixed 12Mbps downstream, varying speed upstream



(b) VoIP flow RTT, fixed 25Mbps downstream, varying speed upstream

Fig. 2: VoIP flow competes with upstream bulk data transfers across gateway with 180-packet FIFO buffers. RTT inflation drops as upstream bandwidth increases. Downstream bandwidth has limited impact when congestion is upstream.

is congested).[6] The elastic TCP flow fully utilised the upstream link's capacity at each offered speed.

Figure 2 captures a key reason why many consumers perceive higher speed offerings from ISPs as critical to having a positive Internet experience. For example, Australia's NBN currently offers 12/1Mbps or 25/5Mbps service plans. Consumers with a mix of interactive and latency-tolerant, upstream, cloud-based applications will have a far better experience on the 25/5Mbps plan, even if their downstream requirements would be otherwise satisfied by a 12/1Mbps plan.

Later in this report we discuss how adding Active Queue Management (AQM) to home gateways can reduce the upstream bandwidth required to support satisfying consumer experiences.

IV. The Social and Technical Context of Australian Households

Now we consider the contemporary networked household in more detail. In 2015 NBN commissioned Telsyte, a telecommunications advisory firm, to undertake primary and secondary data collection on emerging trends in Australian household internet use. The Telsyte report [6] is our primary source for models of current household activity being based on a sample weighted against the Australian population. We supplement the Telsyte research findings with other relevant

[4]The Internet's most widely used reliable transport algorithm. While transfering data between clients and servers, standard *loss-based* TCP's capacity probing strategy cyclically fills and drains FIFO queues at bottlenecks.

[5]$RTT_{base} = 40ms$ for each flow in all cases. Additional background to these experiments is provided in Appendix B

[6]RTT inflation depends on downstream speed and queue management *at the ISP end* when we're congested with heavy downstream traffic

survey research and empirical research on concurrent device use to ground our work in a robust evidence base, and provide models of internet use which is shared with government and industry.

Drawing upon this evidence base, we can point to four critical trends: the proliferation of household connections; a combination of intensive and extensive growth in internet use; increasing diversity in Australian households; and the emergence of new connected services.

A. The proliferation of household connections

Wireless networks and mobile devices have enabled a multiplication of network activity, which is no longer confined to particular domestic rooms or places. Telsyte's survey found that, on average, Australian households now have nine Internet-connected devices; and 9% of households have 20 or more Internet-connected devices. They predict that concurrent device use within the home will rise rapidly in future years, with the average household having up to 29 connected devices by 2020. Much of this dramatic increase is credited to a proliferation of connected objects within the home, including white goods, health monitoring devices, and the Internet of Things (IoT).

The IoT is predicted to vastly increase the number of connected devices in households. Corporate research firm International Data Corporation [7] forecast 30 billion devices to be connected by 2020 globally, while Gartner Inc [8] report a prediction of 20.8 billion connected devices by 2020. At present, globally around 5.5 million new devices are connected daily.

B. The combination of intensive and extensive growth in internet use

The proliferation of connected devices, combined with mobility, has led to both more use, and additional use. According to Telsyte, the average household is now using around eight Internet applications at once during periods of peak usage. Examples of concurrent uses include updating smartphone apps,video chatting, watching streaming video, online study, web-based work applications, downloading podcasts, and backing up data (such as photos) to the cloud. Note that not all of these uses depend on deliberate user control, and also that connected general purpose devices, such as PCs or laptops, are likely to run multiple connected applications at once.

As devices proliferate, so too does concurrent application use. Concurrent usage behaviours vary based on a number of contributing factor. For example, users are more likely to multitask if they have access to multiple connected devices [9]. Other studies on the contemporary networked home further demonstrate that simultaneous device attention is becoming the norm [10], [11], [9].

An increase in devices will impact the average peak number of applications simultaneously connecting to the home network up, increasing the average from 8 applications to 12.

Broadly, we identify a range of applications being reported as used in connected homes:

- General Web Browsing
- Casual Games/Text Messaging
- Social Media
- Operating System, Software or App Updates
- Streaming Video (SD) / Streaming Video (HD)
- Online Games (HD)
- Downloading/Uploading Files
- Video Calling/Conferencing
- Streaming Music
- Radio/Podcasts
- Work- or School-Related
- Web Applications
- Online Storage Sync
- VoIP (Internet Phone Calls)
- Smart HomeVideo-Monitoring Services
- Streaming Video (UHD/4K)
- "Internet of Things" At Home Services

These applications vary in bandwidth requirements. Kenny and Broughton [12] provide some classifications on concurrent application use and bandwidth requirements. They identify that a crucial dilemma in future bandwidth use within the home is less to do with the number of devices, and more to do with anticipated peaks of simultaneous application use and bandwidth restrictions. Kenny and Broughton distinguish between four different types of application: primary (which they categorise as forms of streaming multimedia content, inc TV, YouTube, HD video calls, and streamed and interactive games); secondary (content down and uploads, i.e. cloud storage, torrents, software and OS downloads, and non-HD video calls); web surfing; and finally low-bandwidth traffic, which incorporate all other forms of internet usage. These categories are based upon use characterised by individual spheres of practice rather than application bandwidth specificities, e.g., "primary applications are those apps that are primarily used 'one at a time' by a given individual." (p. 16). Subsequently, the combinations of concurrent application use depict rhythmic patterns of household activities and presume peak application stacks equate to peak bandwidth use. The applications are also characterized by bandwidth requirements rather than latency sensitivity. Still, these categories give a broad overview of what types of applications people are typically using concurrently. In Section II we describe our categorisation of applications based on latency sensitivity, and in Section VI we demonstrate how calculations of bandwidth requirements are less effective when they ignore the latency sensitivity of key applications.

C. The increasing diversity of Australian households

Considerations of household internet use need to take into account the range of living and working arrangements in contemporary Australia. The average household in Australia has 2.6 people [13]. Ostensibly though, traditional family structures should not be assumed, nor should patterns of Internet usage be assumed to be uniform. The Telsyte report identifies the following simplified household profiles, each with distinctive patterns of use that are productive for thinking through how users connect at home:

1) Dual professional households with children: "The Hectic Household": Peak stack of 12 apps today, growing to 19 in 2020

2) Single or dual-parent households with children: " Suburban dreamers": Peak stack of 7 apps today, growing to 13 in 2020

3) Couples without children: "City Living": Peak stack of 11 apps today, growing to 15 in 2020

4) Shared living: Peak stack of 8 apps today, growing to 12 in 2020

Each of these household types represent different models of social activity around internet use within the home, with varying patterns of peak concurrent application use.

Education, work, social and entertainment connections are increasingly mixed [14]. "Peak times" for usage may vary and involve different things depending on the household. The effect of this complication is to undermine assumptions as to how and when users connect at home, and for what purpose.

In Section VI we describe in detail a typical pattern of Internet use for 'The Hectic Household' model drawing on our own empirical fieldwork on domestic internet usage.

D. The emergence of new connected services

Household demand is responding to new internet services: in Australia, a notable example is the emergence of video streaming services such as Netflix, which includes higher resolution content ('4K', or 'ultra high definition'). Another example is the increasing popularity of online education, which makes use of a wide range of media, but particularly relies on video and audio. Education is just one example of new media use that should require us to rethink common assumptions, such as the idea that demand for 'video' is mainly about entertainment. These are examples of high bandwidth applications, but at the other end of the spectrum, 'Internet of Things' services — such as connected appliances — also place increasing demands on household networks.

E. User frustrations with RTTs

RTT delays have long frustrated users [15]. According to the Telsyte report, users reporting latency issues are most likely to do so in regards to lags in cloud storage and backup RTTs. Wider research shows users are also likely to report latency issues based on issues when streaming content or conducting video calls, describing frustration with buffering and call dropout [16]. For example, users are reported to have a threshold of up to 500ms for web page retrieval before becoming frustrated when searching for information [17] and prefer sub-100ms for highly interactive first-person shooter (FPS) games [18].

In our view, the current research on household internet use holds important lessons for the broadband policy debate. Different kinds of households are likely to use devices and applications in different configurations, and at different times. A technically-informed analysis of household internet use and connectivity needs should be sensitive to the number and kind of applications in concurrent use, and the interactions between them.

V. Potential benefits of AQM

In this section we recap the issue with FIFO queue management, review the industry development of new AQM techniques to home gateways, and demonstrate how FQ-CoDel (a specific AQM described later in this section) can potentially *reduce* broadband access speeds requirements when consumers combine both interactive and latency-tolerant services.

A. FIFO queuing and RTT inflation

As noted in Section III-B, the Internet requires certain amounts of buffering (queue storage) in routers and gateways to absorb transient bursts of traffic. During periods of low (or no) congestion, buffers are mostly empty and packets pass through with minimal additional delay. However, during periods of congestion late arriving packets can experience additional queuing delays of up to $\frac{BufferSize}{LinkSpeed}$ seconds.

A widely accepted design goal for maximising TCP performance is for buffers to equal or exceed the *bandwidth-delay product* (BDP) experienced by traffic passing through a gateway. This has important ramifications for consumer gateways that must be manufactured to work well even under worst-case (high) bandwidths and RTT.

For example, consider a gateway intended to support conventional TCP-based traffic over the NBN's highest speed tier over paths between Australia to Europe (approximated by $RTT_{base} = 340ms$). The gateway vendor must aim to please customers for whom their 40Mbps NBN uplink is the bottleneck, and hence build at least 1.7Mbyte of buffering into their gateway. But at lower NBN speed tiers in the upstream direction, the BDP can be significantly lower. The built-in 1.7Mbyte buffer is now in excess of path BDP and simply becomes a source of unnecessary additional RTT inflation. This leads directly to the negative consumer experience illustrated in Figure 2.

B. Active queue management in the home gateway

Recent Internet Engineering Task Force (IETF) interest in new AQM schemes[7] has been motivated by the proliferation of oversized buffers in network devices. Modern AQM schemes such as PIE (Proportional Integral controller Enhanced [20]), CoDel (Controlled Delay [21]) and FQ-CoDel (FlowQueue-CoDel [22]) aim to provide congestion signaling[8] at far lower levels of queuing delay than is typical of classical FIFO (or *tail drop*) queue discipline. A summary of work studying the performance of different AQMs for a variety of traditional Internet applications can be found in [23].

[7] Revisiting ideas from at least the late 1990s, e.g. RFC 2309 [19]

[8] By dropping or Explicit Congestion Notification (ECN) marking of packets

1) PIE : PIE operates on a single queue, and keeps queuing delays low by dropping packets when queuing delays persistently exceed a target delay T_{target}.

The latest PIE draft [20] introduces a burst tolerance parameter of 150ms (by default) which allows packets arriving within the first 150ms of an empty queue to pass successfully. After this, when a packet arrives, it is randomly dropped with a certain probability which is periodically updated and based on how much the current delay is different from $T_{target} = 15ms$ and whether the queueing delay is currently going up or down. The queueing delay can be estimated from the queue length and the dequeue rate. ECN capable packets will be marked instead of being dropped when the dropping probability is $<$10%.

2) CoDel: Like PIE, CoDel operates on a single queue, and keeps queuing delays low by dropping packets when queuing delays persistently exceed a target delay T_{target}.

CoDel tracks the (local) minimum queuing delay experienced by packets in a burst tolerance interval (which is initially 100ms). When the minimum queuing delay is less than $T_{target} = 5ms$ or the buffer size is less than one full-size packet, packets are neither dropped nor ECN marked. When the minimum queueing delay exceeds T_{target}, CoDel enters the drop state where a packet is dropped and the next drop time is set. The next drop time decreases in inverse proportion to the square root of the number of drops since the dropping state was entered. When the minimum queueing delay is below T_{target} again, CoDel gets out of the drop state.

3) FQ-CoDel : With default settings, FQ-CoDel classifies flows into one of 1024 different queues by hashing the 5-tuple of IP protocol number and source and destination IP and port numbers. Each queue is seperately managed by the CoDel algorithm. FQ-CoDel uses a deficit round-robin scheme to service these queues in which each queue can dequeue up to a quantum of bytes (one MTU by default) per iteration. This scheme gives priority to queues with packets from new flows or from "sparse" flows with packet arrival rate small enough so that a new queue is assigned to them upon packet arrival.

An FQ-CoDel bottleneck achieves latency reductions (due to CoDel), relatively even capacity sharing (due to the round-robin scheduling of hashed flows) and priority for low-rate or transactional traffic (such as DNS, TCP connection establishments and VoIP). (The "FQ-" aspect of FQ-CoDel has also been re-purposed to create an open-source FQ-PIE implementation [24], but there is currently no related standardisation effort in the IETF.)

C. Using FQ-CoDel to protect VoIP traffic

There has been significant work on FQ-CoDel in the Linux community, and many home gateways are based around Linux kernels. Here we illustrate the potential benefits of FQ-CoDel AQM in home gateways by evaluating what happens when a VoIP flow, downstream DASH flow and an *upstream* bulk data transfer (elastic flow) compete for resources.

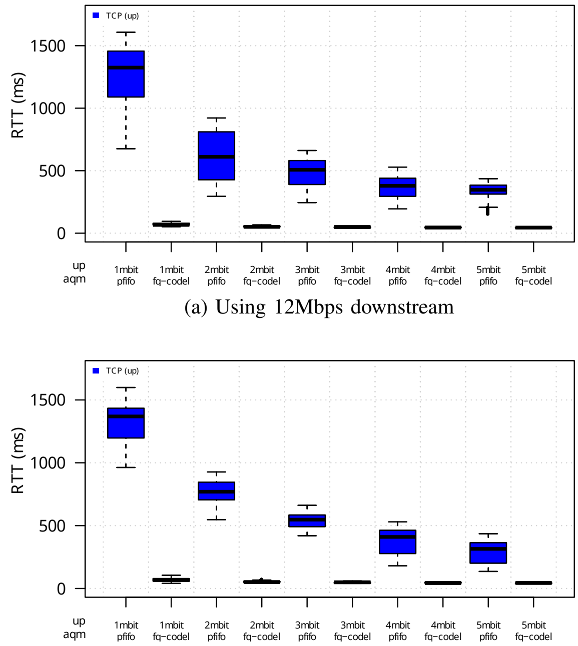
In Section III-B we saw an upstream elastic flow create significant RTT inflation through a conventional home gateway using a range of modest upstream speeds. Using the environment described in Appendix B, emulated broadband rates of $R_{down/up} = \{12, 25\}/\{1, 2, 3, 4, 5\}Mbps$ and each flow having $RTT_{base} = 40ms$, Figure 3 shows that a gateway utilising FQ-CoDel in the upstream direction delivers a dramatically better outcome for VoIP flow RTT compared to the pure FIFO results in Figure 2.

However, FQ-CoDel does incur a known performance trade-off for bulk data transfers over paths with high RTT_{base}. Figure 4 shows the performance of the upstream elastic flow when transferring data to increasingly distant destinations (represented by $RTT_{base} = \{40, 80, 120\}ms$) using $R_{down/up} = 25/\{1, 2, 3, 4, 5\}Mbps$. With FIFO queue management the elastic flow fully utilises the link for each upstream speed and each RTT_{base}. With FQ-CoDel, performance begins to drop off for $RTT_{base} = \{80, 120\}ms$.

More succinctly, for in-country destinations FQ-CoDel makes it feasible for a consumer to concurrently use bulk data applications and interactive applications on entry-level Australian home broadband services. For international destinations, the loss in throughput when using FQ-CoDel is compensated for by the significant reduction of RTT inflation imposed on all traffic sharing the home gateway.

D. Benefits of FQ-CoDel when competing elastic/bulk flows share a last-mile link

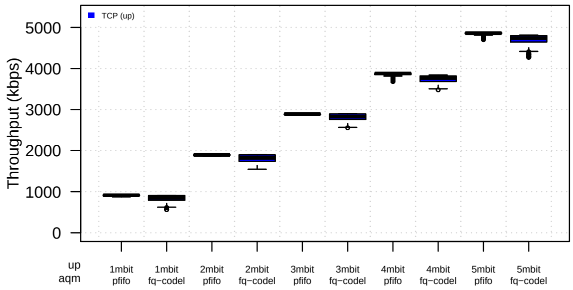
The use of AQM is also beneficial during periods when multiple bulk (elastic) TCP flows compete with each other over a home broadband link.



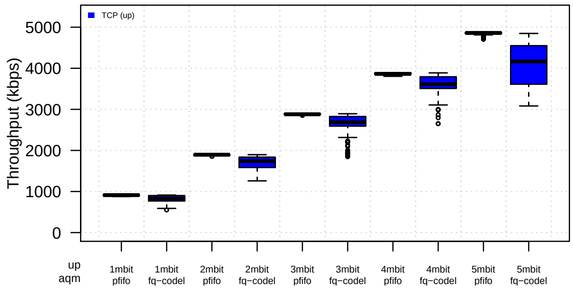
(a) Using 12Mbps downstream

(b) Using 25Mbps downstream

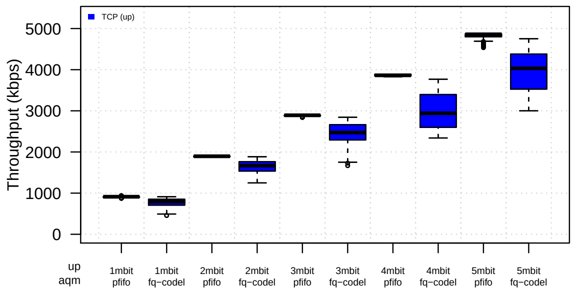
Fig. 3: VoIP flow RTT can be orders of magnitude lower using FQ-CoDel rather than FIFO when competing with an upstream elastic flow across range of upstream rates.



(a) Destination 40ms away ($RTT_{base} = 40ms$)



(b) Destination 80ms away ($RTT_{base} = 80ms$)

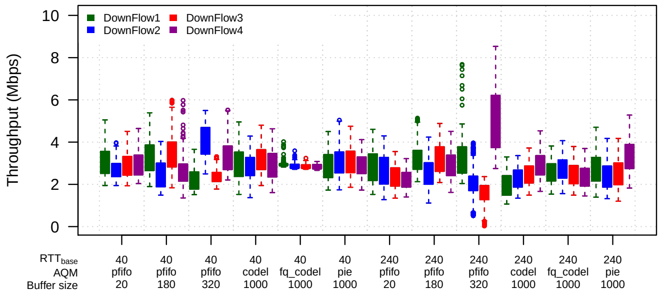


(c) Destination 120ms away ($RTT_{base} = 120ms$)

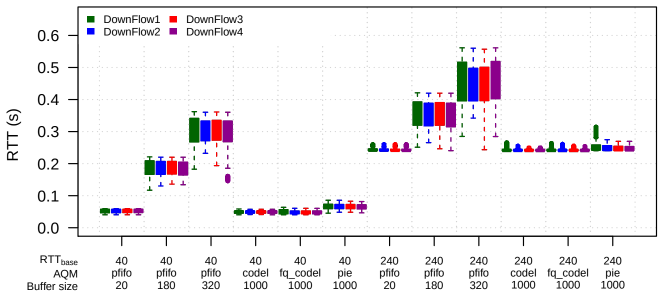
Fig. 4: With FQ-CoDel, performance of upstream elastic flow drops off to more distant destinations. (FIFO with 180-packet buffer, FQ-CoDel with 1000-packet buffer, fixed 25Mbps downstream)

Consider a situation where four elastic flows are concurrently downloading content into a home over a 12/1Mbps service.[9] Using experiments documented further in Appendix C, Figure 5 shows the impact on per-flow throughput and induced RTT when FIFO, CoDel, FQ-CoDel and PIE are used for queue management at both ends of the last-mile link[10], $RTT_{base} = \{40, 240\}ms$ and we try FIFO with very small (20 packet) and moderate (180 and 320 packet) buffers.

Figure 5a reveals that when $RTT_{base} = 40ms$ (e.g. pulling data from in-country cloud services) the four flows achieve roughly similar throughputs whichever queue management schemes are used. Figure 5b then reveals that under these same



(a) Boxplots show throughput is roughly similar across all schemes for $RTT_{base} = 40ms$ (but shared most evenly with FQ-CoDel), some degradation for the AQM scenarios for $RTT_{base} = 240ms$



(b) Boxplots show CoDel, FQ-CoDel and PIE result in significantly lower total RTT than conventional FIFO queue management

Fig. 5: AQMs provide better RTT when four elastic download flows (with same RTT_{base}) share a 12/1Mbps link

circumstances, the lowest RTTs are experienced when using some form of AQM or FIFO with very small buffers. However, real networks are unlikely to see a gateway utilising 'FIFO + very small buffer' (because FIFO bottlenecks are usually built with enough buffering to support high BDP flows). One of the AQM technologies is required if we want other latency-sensitive traffic to experience low RTTs over such a congested last-mile link.

Consider a related situation where three elastic flows are concurrently downloading content into a home, and a fourth elastic flow is pushing content out of a home over a 25/5Mbps service. Again, using experiments documented further in Appendix C, Figure 6 shows the impact on per-flow throughput and induced RTT when FIFO, CoDel, FQ-CoDel and PIE are used for queue management at both ends of the last-mile link[11] and all flows are subject to $RTT_{base} = \{40, 240\}ms$.

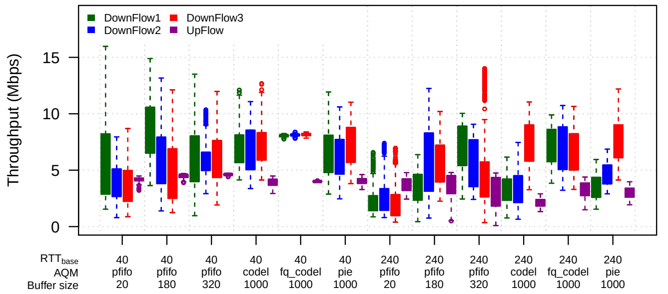
Figure 6a reveals a number of things. When $RTT_{base} = 40ms$ the upstream flow achieves much the same throughput regardless of queue management. However, only FQ-CoDel creates consistent and evenly shared throughput for the three downstream flows. The FIFO scenarios lead to significantly unbalanced throughput across the three flows[12], and the CoDel and PIE scenarios provide somewhat balanced (although noisy) throughput sharing. When $RTT_{base} = 240ms$ all queue

[9] As might occur when devices around the home decided to retrieve a large email, initiate some app updates or synchronise files from offsite storage such as DropBox, OneDrive, and so forth.

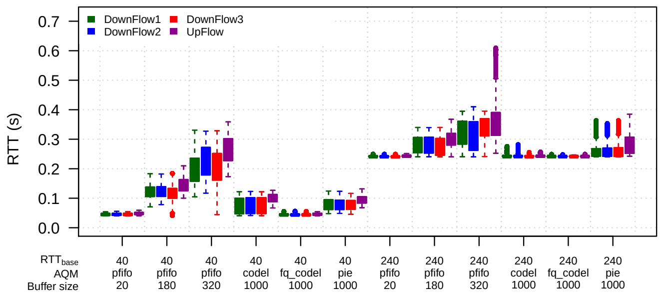
[10] In this case, queue management at the ISP end (in the downstream direction) is most relevant

[11] In this case, queue management at both ends (upstream and downstream) is relevant

[12] In part due to their return path Acnowledgment packets competing with the upstream flow's Data packets

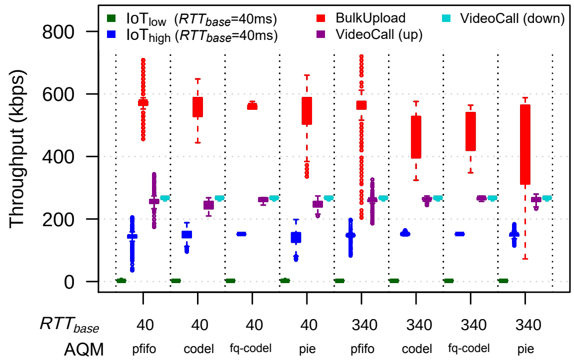


(a) Boxplots show FQ-CoDel shares throughput evenly when $RTT_{base} = 40ms$, with performance degradation for all scenarios when $RTT_{base} = 240ms$

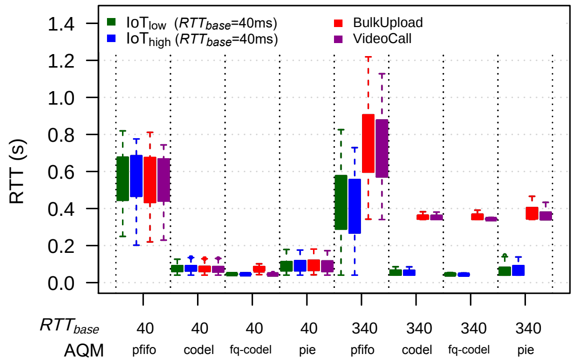


(b) Boxplots show FQ-CoDel results in lowest total RTT of the queue management schemes

Fig. 6: FQ-CoDel provides lowest RTT and best capacity sharing when three elastic download flows and one elastic upload flow (with same RTT_{base}) share a 25/5Mbps link



(a) All flows achieve desired throughput when $RTT_{base} = 40ms$ regardless of queue management. BulkUpload (elastic flow) sees some performance degradation running over AQM with $RTT_{base} = 340ms$



(b) All three AQMs ensure IoT-like flows experience low RTTs during competition with VideoCall and BulkUpload traffic

Fig. 7: Two upstream IoT-like flows competing with bi-directional UDP-based Video call and upstream elastic TCP flow, $R_{down/up} = 12/1Mbps$, $RTT_{base} = 40ms$ for IoT flows and $\{40, 340\}ms$ for non-IoT flows

management strategies show degraded and more inconsistent throughputs in both downstream and upstream directions. Despite the degraded performance, FQ-CoDel again provides the best sharing of available downstream capacity.

Figure 6b reveals that, as noted earlier in Section III-B, an elastic flow in the upstream direction through FIFO causes RTT inflation for all traffic. Aside from the undeployable 'FIFO + very small buffer' scenario, we see very high total RTT when using FIFO, lower when using CoDel and PIE, and lowest total RTTs when using FQ-CoDel. (The presence of upstream traffic causes CoDel and PIE to struggle more in keeping RTT low compared to Figure 5b.)

FQ-CoDel is again showing potential if we want other latency-sensitive traffic to experience low RTTs over a congested last-mile link. Related literature has also shown FQ-CoDel to be beneficial when elastic flows and DASH flows interact [25].

E. Benefits of FQ-CoDel for protecting medical telemetry/IoT flows and online game traffic

In Appendix D we explore how AQM improves the ability of IoT-like telemedicine traffic to co-exist with other classes of traffic. Figure 7 is based on a home utilising a local telehealth/telemonitoring service over a 12/1Mbps last mile link, during a period where two telemonitoring TCP flows (*IoT_{low}* at 2Kbps and *IoT_{high}* at 100Kbps upstream) compete with a 280Kbps UDP-based bi-directional Video call (*VideoCall*) and an elastic upstream TCP flow (*BulkUpload*). We set $RTT_{base} = 40ms$ for the telemonitoring traffic (to represent monitoring by a local hospital or related medical provider) and experiment with the VideoCall and BulkUpload traffic having $RTT_{base} = \{40, 340\}ms$.

Figure 7a shows that regardless of queue management scheme, when $RTT_{base} = 40ms$ the IoT_{low}, IoT_{high} and VideoCall flows basically receive their target throughputs and BulkUpload consumes all spare capacity in the upstream. When $RTT_{base} = 340ms$ for the BulkUpload and VideoCall, any AQM causes moderate performance loss for the BulkUpload flow. However, Figure 7b shows that across all those scenarios the use of AQM (and FQ-CoDel in particular) is an excellent way to keep total RTT much lower than what would be experienced with FIFO.

Additional scenarios are discussed in Appendix D. We conclude that FQ-CoDel can provide good protection in terms of throughput and delay for the IoT flows, better than that achieved with PIE, CoDel or FIFO queue management. Although we also observe FQ-CoDel having some negative impact on loss-sensitive UDP flows at low bandwidth, and on the throughput of bulk TCP flows with high RTT_{base} or at high bandwidth, we conclude it remains a better option than PIE or CoDel where IoT applications share broadband connection with traditional Internet traffic.

Similar observations are made in [26] regarding the benefit of using FQ-CoDel to protect interactive online game traffic. While single-queue AQMs (CoDel and PIE) do keep RTTs low, they are more likely to inflict packet losses on the low-rate, latency-sensitive flows sharing a bottleneck. FQ-CoDel's segragation of flows into separate queues ensures packet losses (as congestion signals) are more precisely targetted on the TCP flows causing congestion.

F. AQM deployment challenges

A number of practical issues mean that AQMs will take time to deploy. In the long term, equipment on either side of the broadband link between homes and their ISPs need to be upgraded, although they do not need to be upgraded at the same time. In the short term, significant benefits will accrue simply from having home gateways incorporate something like FQ-CoDel to manage buffers on the upstream side of their broadband service port(s).

ISPs typically have tight control over their end of the last mile link[13], and use equipment from relatively large and experienced data networking device vendors. However, they are constrained by existing multi-year equipment or software refresh schedules, and their vendors' willingness to incorporate AQM technologies into existing equipment or software. Ideally ISPs would incorporate AQM capabilities in the downstream direction in the medium to long term.

In contrast, home gateways are a significantly more diverse market. Gateways may be owned and supplied by the ISP, or independently purchased by the consumer from a wide range of low-margin vendors. This raises a number of challenges.

Gateways owned by the ISP might, in principle, be upgraded to FQ-CoDel through automated, remote firmware update and reconfiguration. But such an approach presumes the gateway's original vendor can be convinced to release new firmware with FQ-CoDel or similar AQM, and that the original (previous) firmware even allows remotely-controlled firmware updates.

Upgrading consumer-owned gateways relies on having technically-adept and aware owners, who seek out and apply firmware updates with FQ-CoDel or similar AQM capabilities.[14] The potential improvement to interactive application behaviour will likely motivate many consumers to try upgrading their own gateway's firmware. However, as in the ISP-owned case, they are dependent on low-margin vendors actually bothering to release updated firmware for gateways likely to be at least 6-12 months old. Well-known brands have been inconsistent in supplying updates to devices sold more than a few years earlier. This may well continue in regards to adding AQM.

Many consumer gateways are based around an embedded version of the Linux operating system. A sticking point for updates is that FQ-CoDel was first introduced to late 3.x series Linux kernels, but low-end gateways sold even in recent years are often based on older 2.x series Linux kernels. It may simply not make any financial sense for a vendor to spend time backporting FQ-CoDel to the 2.x series kernels of their previous gateways.

On the other hand, a significant fraction of basic consumer gateways retail for less than $100. So if upgrading to 3.x or 4.x series Linux kernels is on a vendor's product roadmap, incorporating FQ-CoDel to their upcoming models is easy. Consumers would then have a choice between replacing their old FIFO-based gateway with a new sub-$100 AQM-capable unit, or paying monthly for a higher downstream/upstream speed tier from their ISP. The former is likely to be a very attractive option for people whose downstream speed requirements are already met by one of the lower speed tiers.

VI. Predictions of consumer bandwidth needs

To demonstrate the arguments made in this report we emulate the Internet usage scenarios of a household, based on the data presented in Section IV. In this section, we describe the results of calculating household Internet usage when using FIFO or AQM techniques at either end of the home's broadband link to the outside world.

Among the household types in Section IV, Telsyte's *Hectic Household* is the most likely one to generate diverse and high traffic during peak hours [6]. We focus on this type of household to give a rough estimation about the maximum bandwidth an Australian household likely demands, both if they continue using FIFO queue management versus if gateways begin using AQM techniques. The other household types can be similarly be emulated and studied by changing the number and mix of application flows of our calculations accordingly.

Our emulation of the *Hectic Household* contains the three categories of consumer traffic described in Section II.

A. Rhythms of a 'Hectic Household'

We model our household as having two parents, and two children based on Telsyte's *The Hectic Household.* This household averages 12 apps in use during peak times. Drawing on our own ethnographic fieldwork on household rhythms we can describe in some detail a realistic profile of such a hypothetical household [28], [29], [30], [31], [32].

Both parents work full time in professional roles. Parent#1 works frequently from home in a designated study area in the spare bedroom. Parent#2 works from home less frequently, and works flexibly to accommodate childcare. The parents as technology enthusiasts encourage their children to engage with technology within age appropriate recommendations. Work and leisure times within the home happen simultaneously. Their technology mix might be as follows:

- Each parent has their own laptop, tablet, and smart phone. Parent#2 also has a FitBit.

[13] At least from the internet protocol (IP) layer and up
[14] Vendor-supplied, or third-party alternatives such as OpenWRT [27]

- There are Sonos speakers in the living room, kitchen, and study area. These are controlled from an app on both Parent#1 and Parent#2's smart phones, and stream music from Spotify. An old smart phone is used to play music in the children's shared room when sleeping, it is usually operated in aeroplane mode.
- In the living area of the house there is a desktop computer used as a home server, and two games consoles, both connected to a smart TV.
- Parent#1 likes to try out new technology, and has an early-adopter interest in IoT devices. They recently purchased wifi-enabled weighing scales, an LED lightbulb, and Chromecast.
- Children use Parent#2's tablet to Skype or Facetime family members, explore Google maps, watch Youtube videos, and ABC iView.

A typical mid-week day might play out as follows:

- Both parents wake at and immediately turn to their smart phones. Parent#1 checks emails, but doesn't respond until later. Parent#2 checks notifications and newsfeeds on several social media apps.
- The children wake, and Parent#1 gets them up, dressed and into the kitchen for breakfast. Parent#2 is already in the kitchen, eating breakfast and reading the news on their smart phone. Parent#1 gives the children breakfast and eats while reading the news of their own smart phone. The children are entertained while eating by watching a YouTube video on Parent#2's tablet.
- Parent#2 heads off to work, taking both children to school on the way. Parent#1 works from home in the study.
- Parent#1's day is filled with a mix of emailing and report writing, with at least 1-2 Skype calls per day. Parent#1 works from files located on a company server, accessed via VPN.
- Parent#2 returns home with the children. The children settle into the living room with Parent#2's tablet.
- Parent#2 works in the late afternoon and into the early evening period while also preparing the family's dinner, Parent#2 does so on their laptop in the kitchen where they can also observe the children's technology use. It is a very busy time of day. Parent#2 is responding to emails while also doing the family's weekly shop online, and coordinating school aand social activities by social media.
- The children are on Parent#2's tablet in living room, alternating between playing games and watching YouTube videos, the TV is on the background streaming an on-demand TV show.
- Parent#1 continues to work on their laptop in the study area.
- Both parent's stop working for family dinner. When the children go to bed, they continue to monitor emails and social media from their phones while watching TV.

Together with the applications users are actively using, there are several other applications running in the background that the family may not be acutely aware of. These include smart phones syncing images 'to the cloud', software updates, and data transfers (e.g. Parent#2's Fitbit monitoring activity and physiological repsonses).

B. The peak 'app stack'

Many applications start and stop throughout the day. We estimate that peak application use ('peak app stack') in this household occurs in the early evening, at which time the following devices are actively accessing the Internet:

- Parent#1's Laptop
- Parent#1's Tablet
- Parent#1's Smart phone
- Parent#2's Laptop
- Parent#2's Tablet
- Parent#2's Smart phone
- Parent#2's FitBit
- Sonos speakers x3
- XBox
- Smart TV (or TV with set-top box)

The following categories approximate the type of applications in use:

- Latency-sensitive applications:
  - Online game
  - Video call
- Latency-tolerant applications:
  - Bulk downloads
  - Bulk uploads
  - Short-lived TCP notifications
  - Web browsing (with multiple windows open)
- Streaming applications:
  - HD video streaming
  - Audio streaming

C. Individual app bandwidth requirements

Even within clearly defined classes of applications, bandwidth requirements are notoriously variable from one specific implementation to another. And user expectations make allocating specific speeds against things like uploading or downloading photos, sync'ing files to offsite storage, and so forth, fraught with uncertainty. As noted in Section II-B, acceptable time to completion (TTC) drives the bandwidth needs of most bulk transfer applications. Two people may accept wildly different TTC for the same activity, and hence need (or tolerate) quite different broadband speeds. We have picked somewhat arbitrary TTC goals for the bulk download, upload and short-lived data burst categories below.

With those caveats in mind, we approximate the ball-park bandwidth requirements for individual categories of hectic household applications as follows:

- One online game
  - 100 kbps down/up
- One video/audio call
  - Audio-only call, 100 kbps down/up

- Video & audio, 200-500 kbps down/up
- One video streaming session
  - ABC iView @ high quality[15]:
    - 1.5Mbps down
  - Netflix[16]:
    - SD@ 480p: 3Mbps down
    - HD@720p: 5Mbps down
    - Ultra HD (4K): 25Mbps down
  - Stan[17]:
    - SD: 3Mbps down
    - HD (720p): 4.5Mbps down
    - HD (1080p): 7.5Mbps down
- One high quality audio streaming session
  - Sonos[18]
    - 256 - 320 kbps down
- One session of web browsing
  - In Nov/Dec 2016 the average web page was 2.4MBytes of content spread over 106 HTTP requests across 35 different TCP connections.[19]
  - Requires both high speed (to download content) and low RTT (minimising TCP connection set up times)
    - 3Mbps down[20]
- One bulk download
  - Retrieving photos from 'the cloud' / file downloads / receiving emails with attachments / software/firmware updates
    - 5Mbps down[21], but will consume whatever excess capacity exists
- One bulk upload
  - Sync'ing photos or documents 'to the cloud', sending emails with attachments
    - 0.5Mbps up[22], but will consume whatever excess capacity exists
- One short-lived TCP data burst / notification (either direction)
  - Social media / instant messages with attachments
    - 0.2Mbps up/down[23], but will consume whatever excess capacity exists

Note that although TCP-based data transfers are often a one-way flow of user-content, the network path must support a matching flow of TCP acknowledgement (ACK) packets in the opposite direction. Depending on circumstances, this ACK traffic may require reverse path bandwidth of around $1/40$ to $1/50$ of the forward path bandwidth.[24] (For example, we should allow roughly 125Kbps of upstream capacity to support a 5Mbps downstream flow.)

D. Hectic Household requirements with FIFO

We now illustrate a rough estimation of the bandwidth required to provide uninterrupted and timely service for the hectic household's worst-case, peak app stack when multiple instances of each application class are interfering with each other over conventional gateways using FIFO queue management. Adding the speeds required by each category in each direction separately gives us:

Downstream direction: 16.9Mbps

- Streaming: 7.46Mbps
  - 1.5Mbps (iView) + 5Mbps (HD TV) + 3 x 0.32Mbps (concurrent audio streams)
- Latency tolerant: 8.8Mbps
  - 5Mbps (bulk download) + 4 x 0.2Mbps (concurrent notifications being received) + 3Mbps (prompt web page retrieval)
- Latency sensitive: 0.6Mbps
  - 0.1Mbps (game) + 0.5Mbps (video call)

Upstream direction: 2.0Mbps

- Streaming: 0.4Mbps
  - 0.4Mbps (ACK traffic to support combined downstream streaming and latency tolerant download traffic)
- Latency tolerant: 1.3Mbps
  - 0.5Mbps (bulk upload) + 4 x 0.2Mbps (concurrent notifications being sent)
- Latency sensitive: 0.3Mbps
  - 0.1Mbps (game) + 0.2Mbps (video call)

A superficial conclusion would argue this household's needs could be met by a (hypothetical) 18/2 Mbps service. However, this would fail to account for the RTT inflation of bulk transfer and streaming services hitting standard FIFO bottlenecks in the upstream and downstream directions (Section III-B). The hectic household would notice significant degradation of their latency-sensitive interactive online activities during peak periods if their home was only serviced at 18/2 Mbps with FIFO queue management.

Revisiting Figure 2 suggests that in order to keep everyone's RTT moderately bounded during bursts of elastic upstream traffic, this household requires at least 5Mbps in the upstream. Assuming the FIFO buffers aren't too long in the downstream direction, a hypothetical 18/5Mbps service might services this household with tolerable degradation from time to time. In

[15] http://iview.abc.net.au/support/faqs#high-quality-video

[16] https://help.netflix.com/en/node/306

[17] https://help.stan.com.au/hc/en-us/articles/202845004-What-kind-of-Internet-speed-do-I-need-to-run-Stan-

[18] https://en.community.sonos.com/music-services-and-sources-228994/streaming-your-music-collection-to-sonos-6736382

[19] Analysed over 490K URLs at http://httparchive.org/interesting.php?a=All&l=Dec%202%202016 and http://httparchive.org/trends.php over same period. Note, 36% of pages average less than 1MB, and 61% are under 2MB. Using just the top 100 web sites, 56% of pages averaged less than 1MB.

[20] TTC goal: retrieve roughly 50% of popular websites in 2-3 seconds

[21] TTC goal: transfer 10Mbytes every roughly 16 seconds

[22] TTC goal: transfer a 2Mbyte photo, file, etc in roughly 30 seconds

[23] TTC goal: send/receive message + 100Kbyte photo in roughly 4 seconds

[24] Consider a TCP connection is carrying one 64-byte ACK packet for every two 1500-byte IP (data) packets on average. The upstream bandwidth due to ACKs is roughly $64/(2*1500)$, or $1/47$ of the downstream bandwidth for data

practice, they would need to purchase at least the 25/5Mbps NBN speed tier.

This household *could* try living with the lower 12/1Mbps NBN speed tier. But during peak app stack periods, while their streaming services may remain arguably acceptable (albeit degraded), their web browsing, bulk uploads and notifications would experience noticeably longer TTCs, and their interactive applications would be regularly disrupted to the point of uselessness. The unsatisfactory solution for households today is to adapt their usage patterns and eliminate simultaneous use of latency sensitive, latency-tolerant and streaming applications. However, this solution is increasingly ineffective as more unattended applications on tablets, phones and IoT devices launch bursts of elastic traffic at unpredictable times throughout the day.

E. Hectic Household requirements with AQM

We now consider the household's overall bandwidth requirements if their broadband service used FQ-CoDel for queue management in both directions. The speeds required by each category of traffic in each direction are the same as in Section VI-D. What changes is the overall bandwidth required from our ISP to keep everyone happy.

As noted in Section V, FQ-CoDel isolates the different traffic flows from each other and keeps RTTs low. The household's latency sensitive traffic will be protected and experience low RTTs, regardless of what the elastic and streaming traffic sources are doing from one instant to the next. The hypothetical 18/2 Mbps service conceivably becomes acceptable.

FQ-CoDel is also beneficial to elastic and streaming applications, as the underlying round-robin scheduling gives even sharing of available capacity during peak app stack periods. Compared to FIFO scenarios, with FQ-CoDel it is far less likely for one elastic traffic flow to dominate other flows (starve them of capacity) for periods of time. Consequently, the household will perceive the TTCs of various activities to behave far more consistently throughout the day.

Another implication is that even the 12/1Mbps NBN speed tier could become a realistic option if the household was willing to accept a modest degradation in TTCs for elastic applications and not-quite-top streaming quality during peak app stack periods. During peak periods, the interactive traffic is protected (due to its low speed demands) and the remaining upstream and downstream capacity is divided equally between all other competing traffic.

F. Implications for alternative household models

The numbers quoted in section VI-D depend critically on assumptions we make regarding the peak app stack – the number and type of applications in use concurrently – and consumer TTC expectations for their elastic applications.

If the two video streams were instead UltraHD (4K) from Netflix and HD (1080p) from Stan, downstream demand would go up by an additional 26Mbps (with roughly 650Kbps in additional upstream demand due to the associated ACK traffic). Conversely, if we had three ABC iView streams only during peak app stack, demand for downstream (content) and upstream (ACK traffic) bandwidth would drop.

If tolerable TTCs were doubled, we could halve the elastic traffic bandwidth requirements. Conversely, households with demand for shorter TTCs (or whose web-surfing habits tend towards the 'heavier' websites) will exhibit higher bandwidth requirements for elastic traffic.

The bandwidth requirements of most current latency-sensitive applications are dwarfed by the requirements of streaming media or elastic applications with demands for low TTC. However, it only takes one latency-sensitive application to be impacted by RTT inflation, and the household will perceive their broadband service to be inadequate. Consequently, demand for significant (e.g. greater than 5Mbps) upstream bandwidth will exist whether we envisage one VoIP call during peak app stack periods, or multiple VoIP calls and multiple online games. Alternatively, we deploy an AQM (like FQ-CoDel or similar) at least in the upstream direction of home gateways to mitigate RTT inflation.

While the Telsyte report provides some alternative models for household populations (and hence peak app stack), there remains a need for greater insight into TTC tolerances and expectations of typical consumers for activities such as sending multi-media notifications, sync'ing content to and from portable devices, and so forth. It is also important to constrain overly optimistic estimates by recognising limits imposed by the physical context of typical households. For example, the number of living areas, bedrooms, and so forth puts practical upper bounds on the number of streaming or internet access devices operating concurrently at any given time. The physical sizes of viewing areas also puts bounds on the sizes of practical TV screens (and hence, the degree to which a household may be satisfied with combinations of SD, HD and/or ultraHD streaming).

G. Implications for future connected services

Using the preceding calculations we can make some inferences towards implications for future connected services. For example, as discussed in Section IV, households are increasingly making use of high definition streaming services, and online education is growing in popularity.

Online education involes a sub-class of semi-interactive applications for the following purposes:

(a) online tutorials where there's a primary video/audio feed to/from tutor, and lots of remote participants;
(b) live streaming of a lecture with limited audience participation;
(c) replay of pre-recorded lectures.

The traffic profiles of these applications will look very similar to traffic already described in this report. Online tutorials with video/audio feed are very similar to applications such as Skype and WebEx. Given Cisco's WebEx application is widely used beyond education, for similar purposes, WebEx bandwidth requirements are a useful guide.[25] Live streaming with audience participation will look much like online tutorial traffic with the addition of an interactive component much like text based IM, or perhaps low-res audio/video Q&A between student users. Replaying of pre-recorded lectures can be classified as streaming content traffic.

H. Implications for households stuck with 'slow' broadband

As discussed in section V-F, consumers and ISPs are in a position to leverage AQM technologies in the upstream direction of home gateways, independent of the political and market forces driving country-wide upgrades to broadband last-mile services. Deployment of FQ-CoDel (or similar) will provide significant improvement to consumer experience of interactive services for those on a 12/1Mbps speed tier or ADSL2+ plans.[26] And even once 25/5Mbps and higher tiers become available, many households may find their needs for VoIP, games and casual internet use met by a 12/1Mbps plan coupled with FQ-CoDel.

Another perspective is to consider how deploying FQ-CoDel could increase the *reach* of emerging services that rely on acceptable interactivity. Online education is an important example. Ideally, remote students are more than passive consumers of pre-recorded content. They should seamlessly participate in real-time, interactive group discussions, tutorials, and so forth with their classmates and instructors.

Unfortunately, such particpation is currently limited to a small population of students capable of either (a) affording very high speed broadband, or (b) curtailing all other internet activities in their household while they participate in online tutorials, etc. Deploying FQ-CoDel in home gateways would eliminate both of these requirements. By enabling those with modest broadband service to better engage with their fellow students online and in real-time, we grow the population of people who would consider becoming remote students in the first place.

VII. Conclusion

We can now see how changing demographics and Internet usage patterns combine with the technical characteristics of home gateway queue management to drive broadband speed requirements in contemporary households. Experiments confirm the poor performance of current home gateway technologies when latency-sensitive and latency-tolerant Internet applications attempt to share a home's broadband access link. A large proportion of bandwidth demand is being driven by applications such as streaming TV services. Another historical driver of demand for access speed is the (arguably misguided) belief that increasing speeds are needed to help keep latencies lower for interactive and IoT applications. However, our findings suggest that relatively low-cost improvements in home gateways — specifically the deployment of emerging active queue management (AQM) techniques — have the potential to address the needs of latency-sensitive applications, with the benefit of a reduction in the need for more expensive, additional access bandwidth.

There are significant areas of future work open to the community as we aim to better understand drivers and constraints on the need for speed. For example, we need deeper insights into TTC tolerances and expectations of typical consumers for elastic activities such as web surfing, sending and receiving multi-media notifications, sync'ing content to and from portable devices, cloud-based backups, and so forth. Bandwidth estimates go up as consumers develop a taste for increasingly shorter TTCs. But we may be over-estimating bandwidth requirements by under-estimating consumer willingness to trade TTCs against the cost of higher speed tiers. It is also important to constrain overly optimistic estimates by recognising the extent to which physical household designs in different communities limit the number of internet attached devices, the sizes of streaming TV screens, and so forth.

Acknowledgements

All the authors worked on this report while employees of Swinburne University of Technology (SUT). Grenville Armitage and Suong Nguyen were members of the Centre for Advanced Internet Architectures (CAIA), while Julian Thomas, Scott Ewing and Jenny Kennedy were members of Swinburne Institute for Social Research (SISR).

We gratefully acknowledge the support and advice of the SUT/Cisco Australia *Innovation Fund* Committee (2014 – 2016), especially Derek Whitehead, Leon Sterling and Reg Johnson. The Fund provided financial support for our project titled "An evaluation of household broadband service requirements for educational innovation and Internet of Things".

We thank the following past and present SUT colleagues for their assistance with and impact on the report: Sebastian Zander for his development of TEACUP that significantly eased the challenge of running AQM performance experiments; Jonathan Kua for developing the TEACUP extensions that enabled us to add DASH traffic to various experimental scenarios; Shahana Cumaranayagam (project-funded CAIA staff) for her preliminary study of the potential for AQMs to protect IoT/telemedicine flows [33] and analysis/verification of results in the Appendix C [34]; Djuro Mirkovic (project-funded CAIA staff) for wrangling bibtex, analysing anomalies in the Appendix C results [34] and running/verifying the multi-flow experiments described in Appendix B; and Russell Collom (project funded CAIA intern) for his analysis of AQMs and first person shooter online games [26].

Appendices

The appendices are structured as follows: Appendix A describes our in-house testbed used to experimentally explore scenarios involving various types of emulated Internet traffic running over emulated broadband links with different queue management strategies. Appendix B describes our experimental emulation of VoIP, DASH and bulk/elastic traffic competing

[25] http://www.cisco.com/c/en/us/products/collateral/conferencing/webex-meeting-center/white_paper_c11-691351.html

[26] Particularly those getting low ADSL sync speeds due to distance from, or poor quality copper wires to, their local telephone exchange

over a FIFO or FQ-CoDel gateway (with selected results appearing in Sections III-B and V-C). Appendix C describes a series of experiments illustrating the impact of AQM on the performance of simultaneous bulk/elastic TCP flows competing across a range of RTTs and bottleneck bandwidths. In Appendix D we look at Telemedicine as a subset of the broader Internet of Things (IoT), and experimentally show how AQM can significantly protect IoT traffic from bandwidth-intensive bulk data flows.

Appendix A
Experimental testbed

CAIA's testbed is custom hardware controlled by a software tool called TEACUP [35]. TEACUP was previously developed in-house for automating data network performance experiments across emulated network paths having highly configurable RTT_{base}, bottleneck bandwidth levels and bottleneck AQM choices.

Subsets of our testbed were used to generate the experimental results presented in the body of this report and later appendices. Much of this Appendix is drawn from [36].

A. Topology

Our testbed consists of the following hardware:

- 20 PC-based hosts that can be used as traffic sources or sinks;
- Two Mac Minis that can be used as traffic sources or sinks;
- Two PC-based routers that route between testbed networks;
- Two managed switches to which all PCs are connected;
- PC used for testbed control and data collection;
- Five power controllers to control the power of the testbed hosts and routers;

Two IP KVMs that allow remotely accessing the console of each host and router. For historical reasons our overall testbed is separated into two logical testbeds, where each logical testbed consists of two experiment networks each with PCs and Macs connected by a configrable bottleneck router. All hosts across both logical testbeds are connected to a single control network. The control network and the four different experimental networks are switched Gigabit Ethernet networks, each mapped to a different VLAN on one of our managed switches (one switch is responsible for one logical testbed).

Figure 8 depicts the IP-layer setup of one of the two testbeds (logically both testbeds are identical). All testbed hosts, the testbed router and the data and control server are connected to the common control network (10.1.1.0/24), which is connected to the public Internet via a NAT gateway (running on the data and control server). The NAT gateway connects all testbed hosts to the public Internet. Testbed hosts are connected to a test network 172.16.10.0/24 or a test network 172.16.11.0/24. The testbed router routes between the two 172.16. networks. For clarity Figure 8 omitted the power controllers and KVMs.

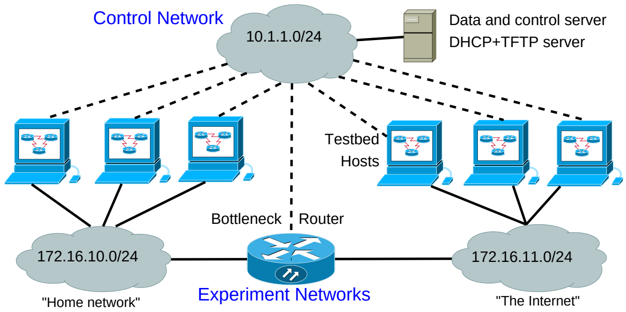


Fig. 8: Testbed used to emulate home connected to the Internet (image adapted from [36]).

The 20 PC testbed hosts are installed as triple-boot machines that can run Linux openSUSE 13.2 64bit, FreeBSD 10.1 64bit, and Windows 7 64bit. This allows experiments with all major TCP congestion control variants. The testbed router is dual-boot and runs Linux openSUSE 13.2 64bit or FreeBSD 10.1 64bit. To automatically boot PCs with the required operating system (OS) we use a Preboot Execution Environment (PXE) based booting approach. The two Mac Minis only run Mac OS X and they do not use PXE booting.

The data and control server runs TEACUP [35] to setup and control the experiments. It also runs DHCP and TFTP servers to assign IP addresses for the control network automatically.

Logical testbed 1 uses six HP Compaq dc7800 machines (4GB RAM, 2.33GHz Intel Core™ 2 Duo CPU, Intel 82574L Gigabit NIC for test traffic) as end hosts, and a Supermicro X8STi (4GB RAM, 2.80GHz Intel Core™ i7 CPU, two Intel 82576 Gigabit NICs for test traffic) as the router.

Logical testbed 2 uses fourteen Acer Veriton X6630G (8GB RAM, Intel Core™ i5, Intel 82574L Gigabit NIC for test traffic) as the end hosts and a Supermicro X9SRi-F (16GB RAM, 3.70GHz Intel Xeon ® E5- 1620v2, Intel I350-T2 dual port Gigabit NIC for test traffic) as the router.

B. Emulated bottleneck and path conditions

Linux `netem` and `tc` modules provide packet FIFO (PFIFO), PIE, CoDel and FQ_CoDel queue management schemes and emulate the bottleneck bandwidths and the base Round Trip Time (RTT), RTT_{base} (by adding $RTT_{base}/2$ on both forward and reverse paths).[27]

Packets sit in a 1000-packet buffer while being delayed, then sit in a separate *bottleneck buffer* while being rate-shaped to the bottleneck bandwidth. The bottleneck buffer is varied depending on the case study for PFIFO experiments and 1000 packets for PIE, CoDel and FQ_CoDel experiments. As they are largely intended to require minimal operator control or tuning, we used PIE, CoDel and FQ_CoDel at their default settings. ECN is disabled on all hosts.

Note the current Linux PIE (and hence the one used in our testbed) is based on an earlier draft [37], with some differences in traffic burst handling (a default $T_{target} = 20ms$, shorter

[27] See section IV-B of [35] for details on how TEACUP appropriately configures `netem` and `tc` for this purpose.

initial burst tolerance of 100ms, and lacks drop probability autotuning and de-randomisation of [20]).[28]

C. Traffic generators

TEACUP supports a variety of traffic generators which can be used to emulate the traffic of different types of applications. A subset of those are described as follows.

a) iperf: This tool can be used to generate TCP bulk transfer flows in which the iperf client pushes data to the iperf server. The duration to generate traffic or the total data size can be specified.

b) nttcp: This tool can be used to generate UDP or TCP traffic where packet size and packet interval can be manually specified. Therefore, we use this tool to generate bidirectional UDP flows to emulate real-time traffic such as VoIP and video calls. Besides, we also use it to emulate applications which send packets periodically and are TCP-based, e.g. Internet of Things applications.

c) DASH: This tool is used to generate a Dynamic Adaptive Streaming over HTTP [3], [38] (over TCP) video flow retrieving video segments (*chunks*) every fixed interval.

d) lighttpd: This tool is used to emulate a web server. It can be used as traffic source for httperf-based sinks.

e) httperf: This tool can be used to simulate an HTTP client. It can generate complex workloads based on work session log files. We use httperf together with lighttpd to emulate web browsing.

f) pktgen: This tool is used to generate the game traffic in our First Person Shooter game with configurable number of clients, game type, base packet size of packets send by clients, time interval between packets sent by client and time interval between packets sent by server.

D. Logging

TCP connection statistics were logged using SIFTR under FreeBSD and Web10g under Linux. Packets captured at both hosts with tcpdump were used to calculate throughput, packet loss rate and non-smoothed end to end RTT estimates using CAIA's passive RTT estimator, SPP [39].

E. Measuring throughput, delay and packet loss rate

a) Throughput: "Instantaneous" throughput is an approximation derived from the actual IP-layer bytes transferred during constant (but essentially arbitrary) windows of time. Long windows smooth out the effect of transient bursts or gaps in packet arrivals. Short windows can result in calculated throughput that swings wildly (but not necessarily meaningfully) from one measurement interval to the next.

b) Delay: Non-smoothed RTT values are derived using Synthetic Packet Pairs (SPP) RTT estimation [39] from packets captured at each host.

c) Packet loss rate: UDP flow packet loss rates are calculated from the total packets sent and received, matched at each end by each packet's unique 'IP identification' field.

[28]Linux PIE has not changed between kernel 3.14 to 4.5, http://lxr.free-electrons.com/diff/net/sched/sch_pie.c?diffvar=v;diffval=3.14

APPENDIX B
MUTUAL INTERFERENCE BETWEEN VOIP, DASH AND BULK DATA TRANSFERS

The following experiments used a subset of Appendix A's testbed to illustrate how different upstream bandwidths would impact VoIP, DASH and bulk/elastic traffic competing over a FIFO or FQ-CoDel gateway. A subset of these results appeared in Sections III-B and V-C.

Three hosts were assigned on either side of the bottleneck router to be sources and sinks of the following traffic categories:

(a) latency-sensitive (e.g. a bidirectional G.711 VoIP call)
(b) streaming (single DASH flow into home)
(c) latency-tolerant (TCP bulk data from home to internet)

The following hosts were configured on the 172.16.10.0/24 (home side):[29]

- 172.16.10.11 (newtcp1), a DASH client (data sink)
- 172.16.10.12 (newtcp2), a bidirectional nttcp client and server emulating G.711 VoIP
- 172.16.10.13 (newtcp3), an iperf server (sends data upstream)

The following hosts were configured on the 172.16.11.0/24 (internet side):[30]

- 172.16.11.14 (newtcp4), a bidirectional nttcp client and server emulating G.711 VoIP
- 172.16.11.17 (newtcp7), a DASH content source (lighttpd webserver)
- 172.16.11.18 (newtcp8), an iperf client (receiving data from newtcp3)

The bottleneck router emulated both a home gateway and last-mile broadband link with the following parameters:

- A *constant* RTT_{base} of 40ms for traffic between newtcp{1,2} and newtcp{4,7}
- A *varying* RTT_{base} of {40, 80, 120, 180}ms for traffic between newtcp3 and newtcp8
- Two queue management algorithms – PFIFO (buffers of 20, 180 and 320 packets) and FQ-CoDel (with buffer of 1000 packets)

ECN was disabled for all trials, and no artifical loss was added (no losses beyond those induced by bottleneck buffer congestion).

TABLE II: Downstream/upstream speeds (in Mbps)

Rate	Representative of...
12/{0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0}	entry-level NBN downstream and *hypothetical upstreams*
25/{0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0}	mid-tier NBN downstream and *hypothetical upstreams*

[29]newtcp1 used FreeBSD and NewReno TCP, newtcp2 & newtcp3 used Linux and CUBIC TCP
[30]All hosts used Linux with CUBIC TCP

We trialled each combination with the *downstream/upstream* bottleneck speeds shown in Table II (emulating some hypothetical Internet access speeds).

Each trial ran for 120 seconds, with staggered starts for the different flows:

- Flow 1: Start at t=0sec, one bidirectional emulated G.711 VoIP flow between newtcp{2,4}
- Flow 2: Start at t=30sec, one DASH flow from newtcp7 to newtcp1
- Flow 3: Start at t=60sec, one *upstream* TCP flow from newtcp3 to newtcp8

Note:

- Flows 1 and 2 have a *fixed* RTT_{base} of 40ms
- Flow 3'sRTT_{base} varies across the range {40, 80, 120, 180}ms.

Allowing a 5sec guard-band, we considered all three application flows to be stable and concurrently competing for resources between t=65 and t=115sec. Resulting throughput and RTT measurements were calculated during this more restricted time period.

APPENDIX C
BULK TCP EXPERIMENTS

Here we describes a series of experiments illustrating the impact of AQM on the performance of simultaneous bulk/elastic TCP flows competing across a range of RTTs and bottleneck bandwidths. We use the physical testbed described in Appendix A.

A. Introduction

Our focus here is on the behaviour of bandwidth-demanding applications that perform 'bulk' download or upload over TCP. This is typical of devices exchanging photos with 'cloud' storage providers, large email attachments, offline video delivery, app downloads and dropbox-like file synchronisations. The traffic flows generated by such applications interfere with each other, and interfere with any real-time interactive flows sharing the home gateway (or *bottleneck router*) at the same time. Our first goal is to begin quantifying the extent and nature of such interference for scenarios where a handful of concurrent TCP connections compete for Internet access over a range of (a) common home access speeds, (b) plausible underlying network latencies and (c) home gateway queue management technologies.

B. General test conditions

We use a separate pair of hosts from the two subnets 172.16.11.0/24 (internet side) and 172.16.10.0/24 (home side) for each concurrent TCP flow.

a) Emulated bottleneck and path conditions: We trialled each traffic generation scenario using combinations of the following conditions:

1) The range of RTT_{base} drawn from Table I to emulate a range of real-world scenarios. Higher RTT_{base} (e.g. 340ms) may also proxy for satellite-based internet access even to domestic ground-based servers
2) The emulated Internet access speeds show in Table III, reflecting a mix of ADSL, DOCSIS cable modem and NBN-like services
3) Four different types of queue (buffer) management algorithms inside the emulated home gateway: PFIFO, CoDel, PIE and FQ-CoDel

TABLE III: Common last-mile downstream/upstream speeds

Down-/Up-stream rate (Mbps)	Representative of...
1.5/0.5	~ADSL1
4/1	low-end ADSL2+
12/1	mid-tier ADSL2+ / entry-level NBN
20/1	high-end ADSL2+ / low-end DOCSIS
25/5	mid-tier NBN
50/2	mid-tier DOCSIS
50/20	mid-/high-tier NBN
100/40	high-tier NBN

For the bottleneck router, a given buffer size can be sufficient, too small or too big relative to the Bandwidth Delay Product (BDP) of traffic passing through the bottleneck. Table IV shows the BDP (in packets) for a number of speed and RTT_{base} combinations. For PFIFO trials we used buffers of 20, 180 and 320 packets. For CoDel, PIE and FQ-CoDel trials we choose the buffer size of 1000 packets for burst absorption (because each AQM will aim to keep sojourn times low anyway).

TABLE IV: Bandwidth delay product (BDP) for some representative combinations of speed and RTT_{base}

Speed and RTT	BDP in packets
1.5Mbps @ 10ms	1.25
1.5Mbps @ 40ms	5
12Mbps @ 10ms	10
12Mbps @ 40ms	40
12Mbps@ 140ms	140
50Mbps @ 10ms	42
50Mbps @ 40ms	167
100Mbps @ 10ms	83
100Mbps @ 40ms	333

Other test conditions:

- No artifical loss added (no losses beyond those induced by bottleneck buffer congestion)
- ECN disabled for all trials

b) Operating systems and TCP algorithms: We trialled each of the RTT_{base}, bandwidth and buffer size combinations using three different types of TCP algorithm:

- NewReno (using FreeBSD, but similar to the TCP used by Apple devices)
- CUBIC (using Linux, and the TCP algorithm underpinning Android devices)
- CompoundTCP (Windows 7 and later)

c) Traffic generator: We use `iperf` to send full-size packets over a TCP connection as fast as possible in each direction (separately) to represent upstream (uploading) or downstream (downloading) flows.

C. Experiment setup

We have explored four main traffic generation scenarios:

- *1upload*: A single TCP flow upstream for 90 seconds
- *1download*: TCP flow downstream for 90 seconds
- *4uploads*: Four concurrent TCP flows upstream for 90 seconds: four flows starting 2 seconds apart. We split this into two subscenarios below.
  - All flows have the same RTT_{base}
  - All flows have different RTT_{base}
    - Flow1 @ $RTT_{base} = 240ms$
    - Flow2 @ $RTT_{base} = 180ms$
    - Flow3 @ $RTT_{base} = 40ms$
    - Flow4 @ $RTT_{base} = 10ms$
- *4downloads*: Four concurrent TCP flows downstream for 90 seconds: similar to the four concurrent TCP flows upstream in terms of flows' starting time as well as base RTT.
- *3uploads&1download*: Three TCP flows upstream for 90 seconds starting 2 seconds apart, one TCP flow downstream starting 30sec through: all flows have the same RTT_{base}.
- *3downloads&1upload*: Three TCP flows downstream for 90 seconds starting 2 seconds apart, one TCP flow upstream starting 30sec through: This includes two subscenarios below.
  - All flows have the same RTT_{base}
  - All flows have different RTT_{base}
    - Flow1 @ $RTT_{base} = 240ms$
    - Flow2 @ $RTT_{base} = 180ms$
    - Flow3 @ $RTT_{base} = 40ms$
    - Flow4 (upload) @ $RTT_{base} = 10ms$

These represent a variety 'lightly loaded' scenarios where one or more devices in the home decide to engage in short period of concurrent or semi-concurrent Internet usage.

D. Results and discussion

Here we will choose two representative scenarios, *4downloads* and *3downloads&1upload*, to illustrate our results and the corresponding analysis. For each scenario, the throughput and RTT of flows are provided for the case of all flows having the same RTT_{base} and the case of all flows having different RTT_{base}.

- Case 1: 4 download flows, same RTT_{base}, see Figs. 9, 10 and 11.
- Case 2: 4 download flows, different RTT_{base}, see Figs. 12, 13 and 14.
- Case 3: *3downloads&1upload*, same RTT_{base}, see Figs. 15, 16 and 17.
- Case 4: *3downloads&1upload*, different RTT_{base}, see Figs. 18, 19 and 20.

1) Discussion:

a) Change of throughput and delay with bandwidth: Over the discrete set of bandwidths we consider, it is observed that when bandwidth increases, throughput of flows increases. However, the increasing rate may not be the same for different flows, which depends on their staring time and RTT_{base}. Besides, their delay also decreases with the increase of bandwidth. Those imply that user experience is improved. In particular, the throughput improvement means that data transfer such as photo uploads and application updates takes less time to complete and video can be streamed at higher quality (e.g. higher resolution and higher bitrate). Meanwhile, the delay improvement means that applications are more responsive, which is specially important for real-time traffic such as voice/video call and other traffic requiring responsiveness such as web browsing.

From the throughput graphs in Figs. 9 to 20, the throughput with AQMs is similar to slightly less than that with PFIFO. However, the RTT graphs show that the delay is much smaller with AQMs than with PFIFO at high buffer size (e.g. 180 or 320 packets). One can argue that PFIFO has small RTT if the buffer size is set to small value such as 20; however, the default buffer size in commercial home gateway is usually set to a reasonably large and throughput degrades noticably with PFIFO at small buffer size when bandwidth is high enough (e.g. 25/5Mbps) as in Figs. 11, 14, 17 and 20. Besides, PFIFO at small buffer size does not handle bursty traffic such as web browsing well as AQMs do. Therefore, it can be said AQMs have clear advantage over PFIFO.

Also note that the AQMs at 1.5/0.5Mbps has smaller RTT than PFIFO at buffer size 180 or 320 at 25/5Mbps. This means that using AQMs at the home gateway can improve the responsiveness of applications without the need to increase bandwidth if a household often uses low-rate applications which require high responsiveness but not bandwidth-intensive applications such as video streaming and has no concern about the time it takes to download or upload bulk data. If a household has high demand for video streaming and the short completion time of bulk data transfer, using higher bandwidth is the solution.

Among the AQMs, FQ-CoDel has the smallest mean and variance of delay while those are highest with PIE. This makes FQ-CoDel a better candidate to be deployed at the home gateway in the existence of real-time traffic.

b) Fairness issue when all flows have the same RTT_{base}: When flows in a certain direction have the same RTT_{base}, with PFIFO, the first starting flow will gain higher throughput than the other flows in the same direction for at least the first 90 seconds at low bandwidth (e.g. 1.5/0.5Mbps) and large buffer size (e.g. 320). This unfairness is consistently observed under NewReno as shown in Figs. 9 and 15 as well as under CUBIC. Under CompoundTCP, we also observe similar "unfairness"; however, the flow with the highest throughput can be the third flow instead of the first flow in some trials. Although this unfairness goes away if these flows exist for much longer than 90 seconds, there are many bulk data flows which do not stay long in the network such as photo upload and application update. When the unfairness exists, the gap between the highest throughput and the lowest throughput of flows in the same direction can be very large. This can

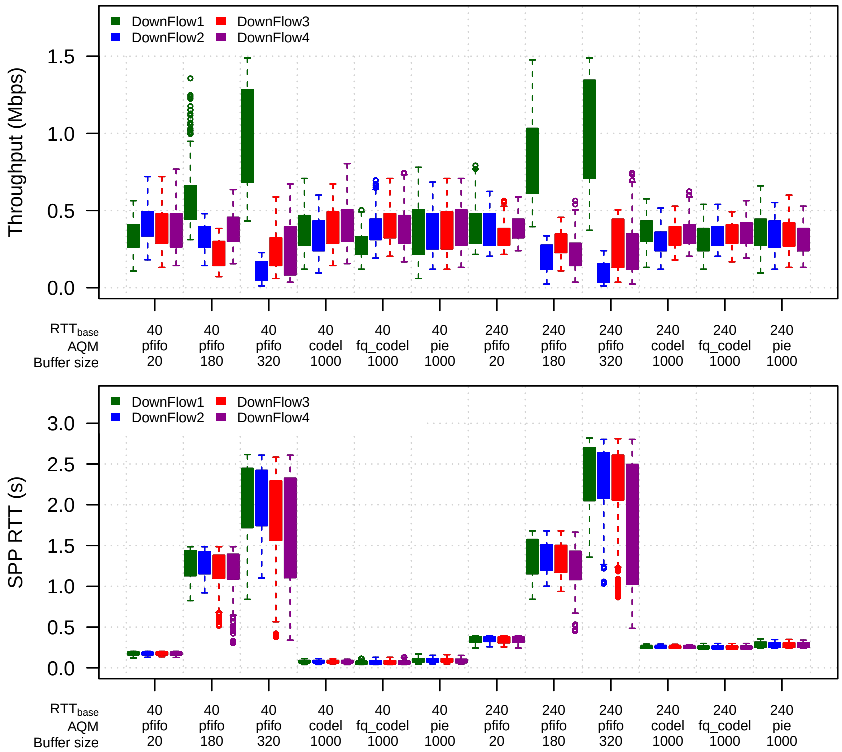


Fig. 9: 4 download flows: same RTT_{base} at 1.5/0.5Mbps.

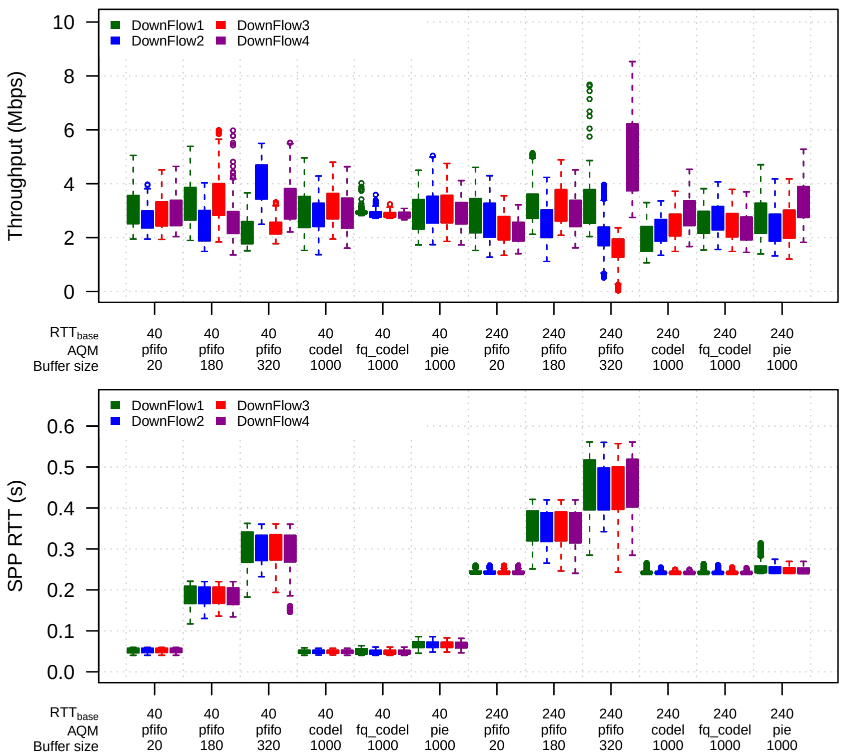


Fig. 10: 4 download flows: same RTT_{base} at 12/1Mbps.

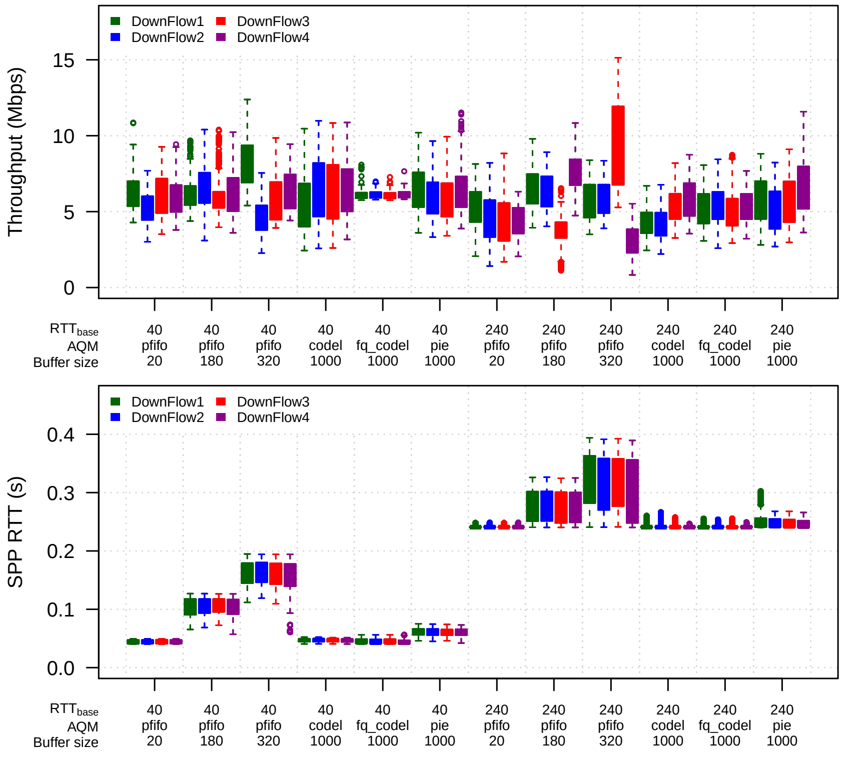


Fig. 11: 4 download flows: same RTT_{base} at 25/5Mbps.

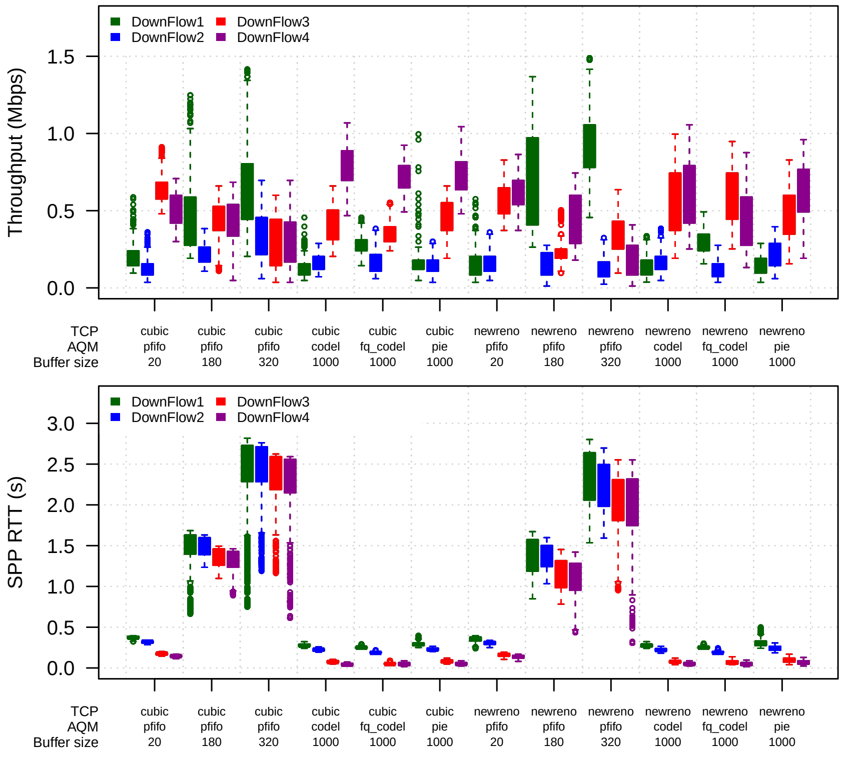


Fig. 12: 4 download flows: different RTT_{base} at 1.5/0.5Mbps (Flow1 @ $RTT_{base} = 240ms$, Flow2 @ $RTT_{base} = 180ms$, Flow3 @ $RTT_{base} = 40ms$, Flow4 @ $RTT_{base} = 10ms$).

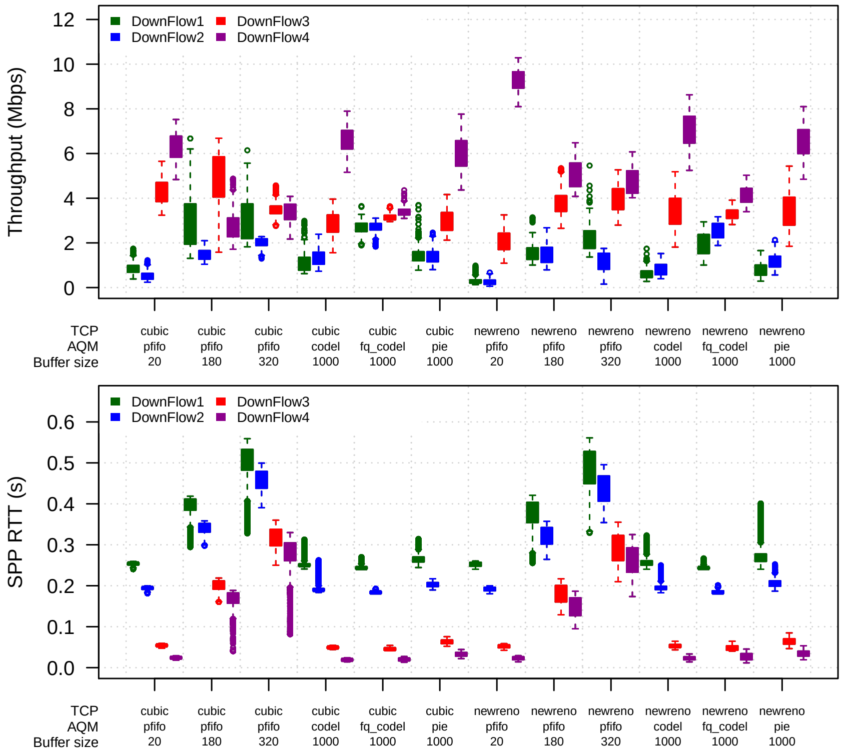


Fig. 13: 4 download flows: different RTT_{base} at 12/1Mbps (Flow1 @ $RTT_{base} = 240ms$, Flow2 @ $RTT_{base} = 180ms$, Flow3 @ $RTT_{base} = 40ms$, Flow4 @ $RTT_{base} = 10ms$).

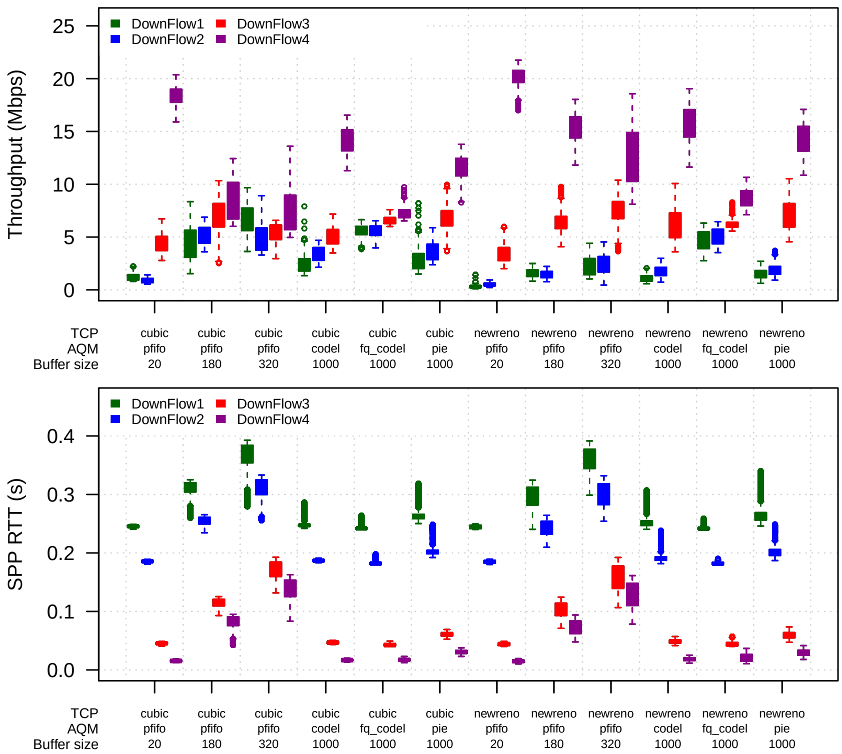


Fig. 14: 4 download flows: different RTT_{base} at 25/5Mbps (Flow1 @ $RTT_{base} = 240ms$, Flow2 @ $RTT_{base} = 180ms$, Flow3 @ $RTT_{base} = 40ms$, Flow4 @ $RTT_{base} = 10ms$).

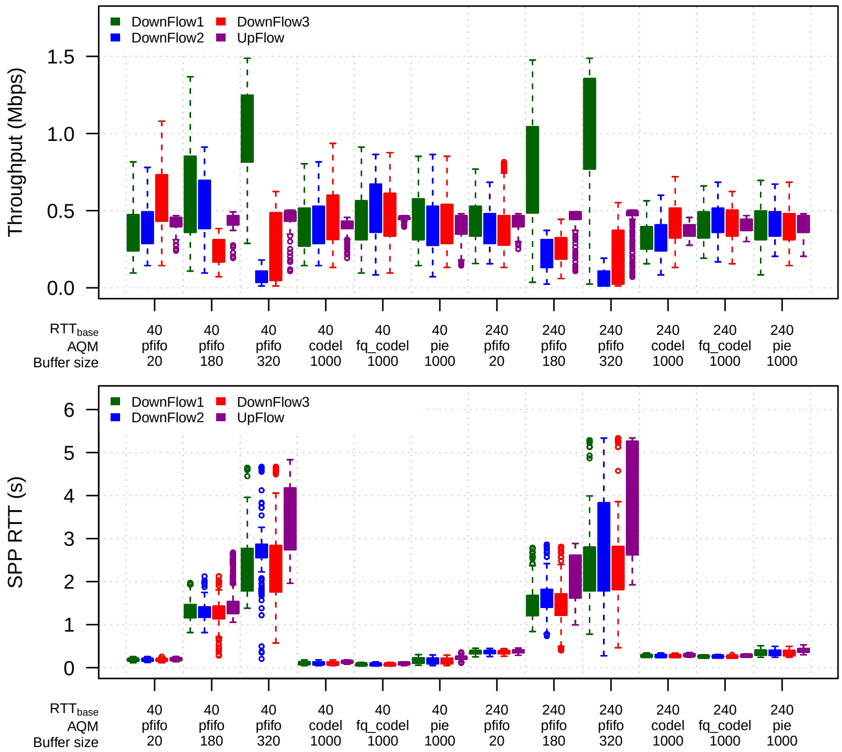


Fig. 15: 3 downloads & 1upload: same RTT_{base} at 1.5/0.5Mbps.

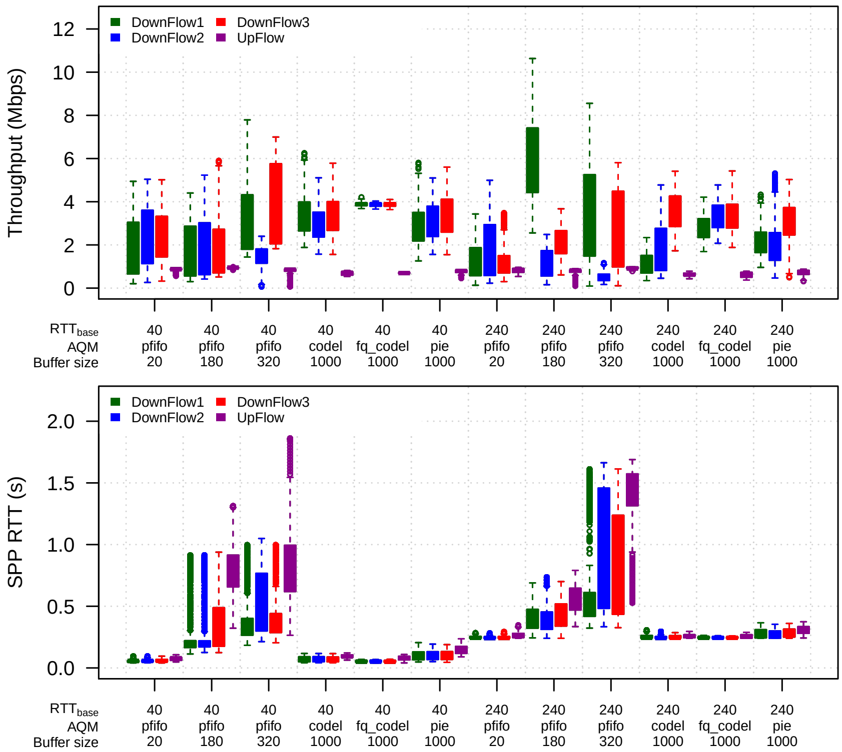


Fig. 16: 3 downloads & 1upload: same RTT_{base} at 12/1Mbps.

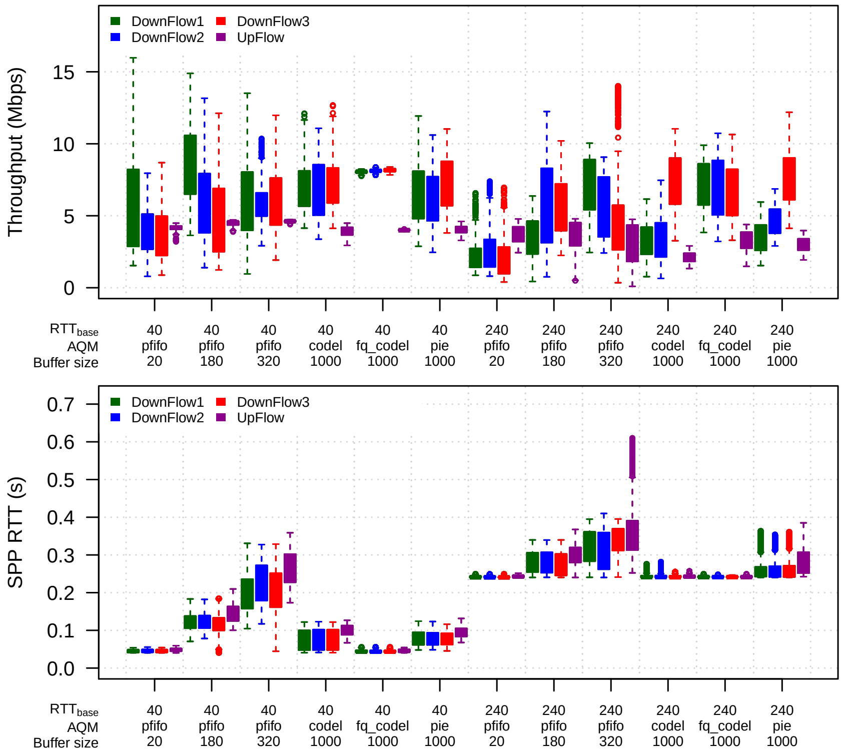


Fig. 17: 3 downloads & 1upload: same RTT_{base} at 25/5Mbps.

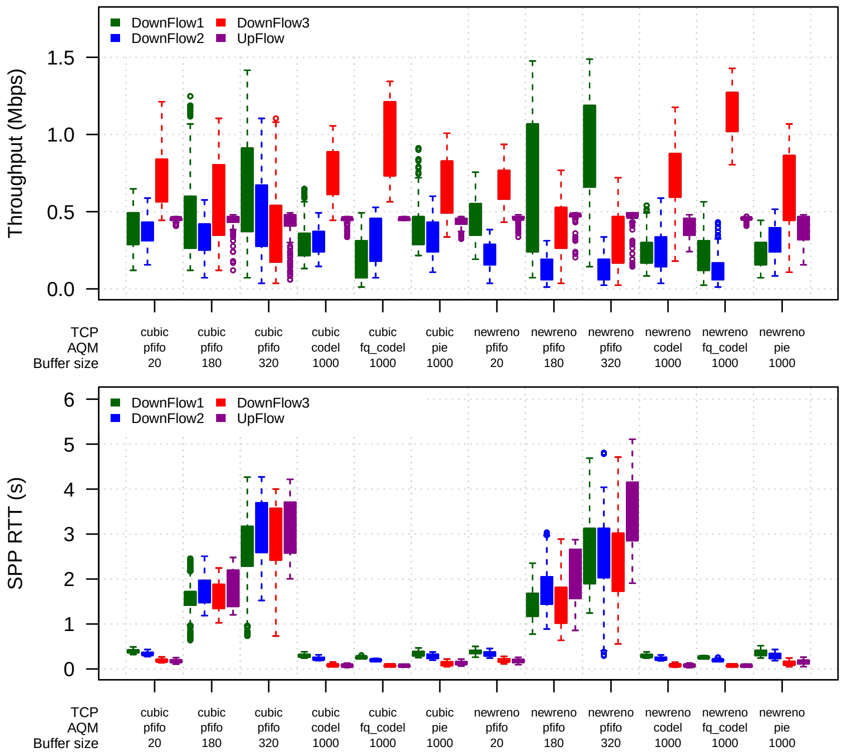


Fig. 18: 3 downloads & 1upload: different RTT_{base} at 1.5/0.5Mbps (DownFlow1 @ $RTT_{base} = 240ms$, DownFlow2 @ $RTT_{base} = 180ms$, DownFlow3 @ $RTT_{base} = 40ms$, UpFlow @ $RTT_{base} = 10ms$).

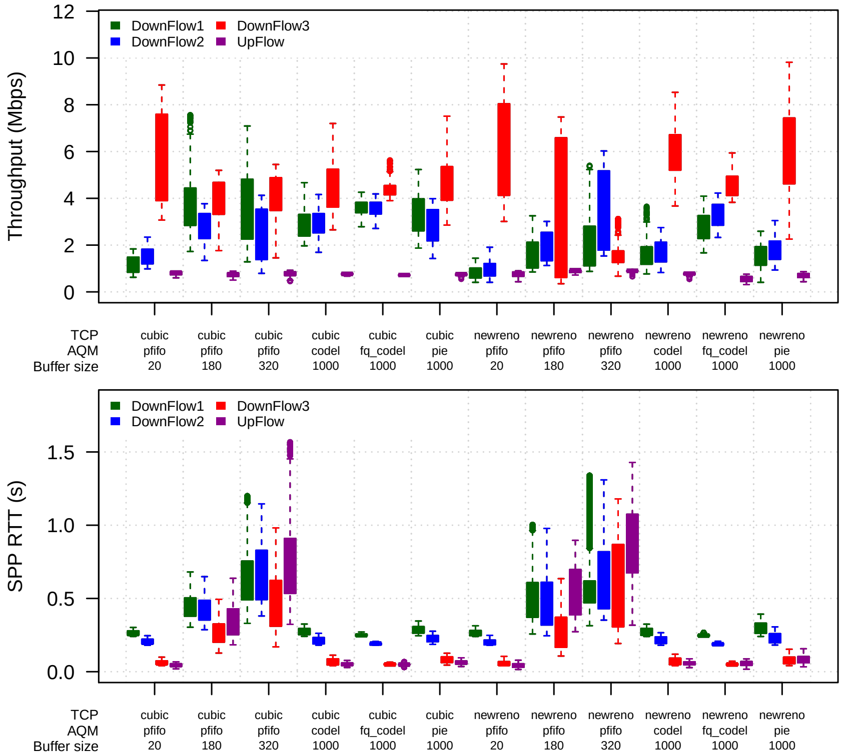


Fig. 19: 3 downloads & 1upload: different RTT_{base} at 12/1Mbps (DownFlow1 @ $RTT_{base} = 240ms$, DownFlow2 @ $RTT_{base} = 180ms$, DownFlow3 @ $RTT_{base} = 40ms$, UpFlow @ $RTT_{base} = 10ms$).

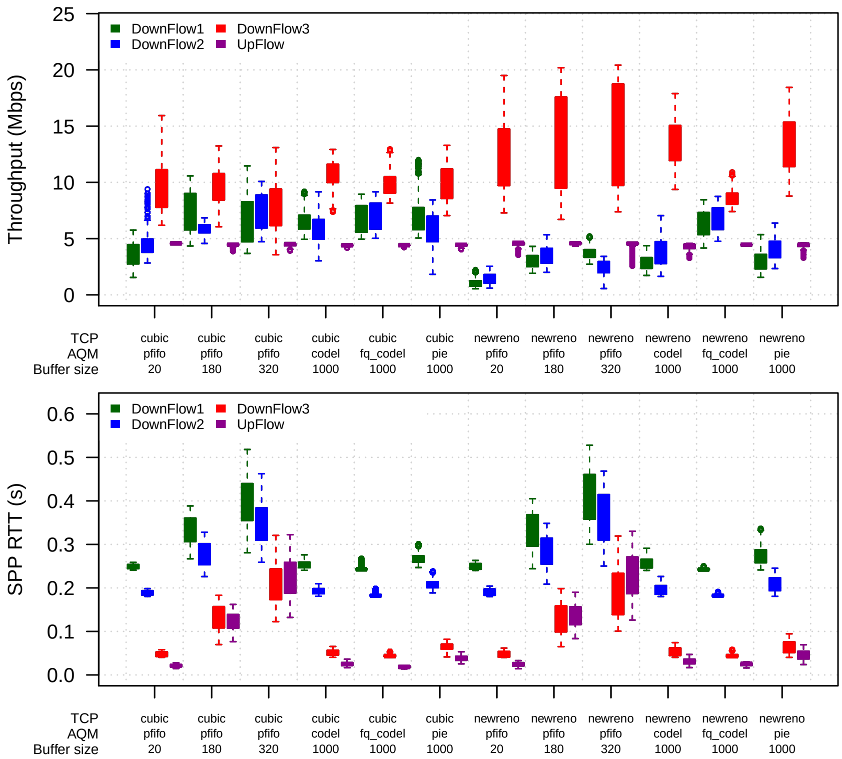


Fig. 20: 3 downloads & 1upload: different RTT_{base} at 25/5Mbps (DownFlow1 @ $RTT_{base} = 240ms$, DownFlow2 @ $RTT_{base} = 180ms$, DownFlow3 @ $RTT_{base} = 40ms$, UpFlow @ $RTT_{base} = 10ms$).

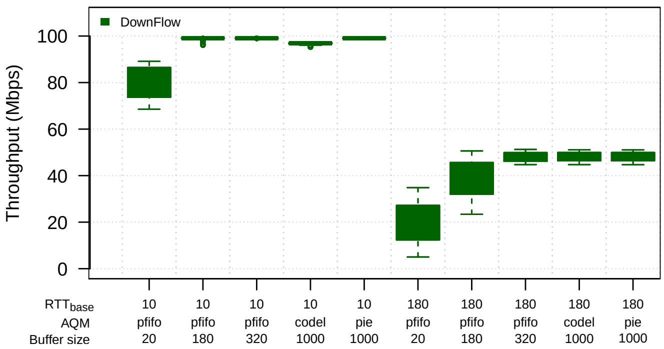


Fig. 21: Performance of one download flow at 100/40Mbps limited by receiving host's `rwnd` at high RTT_{base}

cause the "unstable" Internet experience for the same user in a household.

However, this unfairness does not happen with AQMs and PFIFO with small buffer size (e.g. 20) where flows in the same direction have roughly fair bandwidth share. Besides, among AQMs and PFIFO, FQ-CoDel provides the highest fairness, especially at low to moderate RTT_{base}.

c) Fairness issue when all flows have different RTT_{base}: When all flows in a certain direction have different RTT_{base}, we still observe similar "unfairness" like in the case of flows having same RTT_{base} with PFIFO at low bandwidth and high buffer in Figs. 12 and 18. In constrast, at low bandwidth, among flows in the same direction, the AQMs and PFIFO with small buffer size (20) give noticably higher throughput to the flows with much lower RTT_{base}.

When bandwidth increases, among the flows in the same direction, the flow with the lowest RTT_{base} seems to gain significantly higher bandwidth share under PFIFO, CoDel and PIE. This "unfairness" is more visible under NewReno and CompoundTCP than under CUBIC. When the unfairness exists, the difference between the highest throughput and the lowest throughput of flows in the same direction can be pretty high, which can cause the same user in a household to have noticably different Internet experience.

Importantly, FQ-CoDel provides much higher fair share among these flows than other AQMs and PFIFO, which makes FQ-CoDel again a good candidate for the home gateway.

2) Factors limiting throughput (other than access link bandwidth): From our preceding results, it can be seen that the throughput of flows increase with Internet speed of the home network. However, there are other factors limiting the throughput of flows, regardless of how large the bandwidth is.

One factor is the TCP receive window (as discussed further in CAIA Technical Report 160923A [34]). In particular, TCP has two components, congestion window (`cwnd`) and receive window (`rwnd`), that affect the number of packets in flight at a given time, which consequently determines the throughput of a TCP flow. `cwnd` increases with the link bandwidth while `rwnd` can be constrained at a particular value. The constraint of the `rwnd` makes the throughput unable to scale with bandwidth.

This can be observed in Fig. 21, which shows the throughput achieved by single flows (the *1download* scenario) over a 100/40Mbps link using different AQM, small and large FIFO buffer sizes and $RTT_{base} = \{10, 180\}ms$. When $RTT_{base} = 10ms$ the only combination achieving less than full speed is PFIFO with a 20-packet buffer (this bottleneck buffer is less than path BDP). But at higher $RTT_{base} = 180ms$ the maximum `rwnd` used in our experiments limits achieved throughput to $\sim 50Mbps$ with either FIFO (with 320-packet buffer), CoDel or PIE.[31]

When throughput is limited by factors such as receive window limits, increasing the broadband connection speeds will not result in any increase in performance for bandwidth-intensive applications. A positive side effect for interactive applications under such conditions is that average bottleneck queueing delays are smaller (as the elastic flow is unable to fully congest the home gateway).

E. Summary

In this section, we have investigated the performance of bulk data traffic at a variety of network conditions over a range of scenarios with different queueing disciplines deployed at the home gateway. We have demonstrated that with PFIFO, the flows can have high delay when the home gateway is configured with large buffers. In addition, PFIFO can also lead to throughput "unfairness" among flows with the same RTT_{base} at low bandwidth and large buffers and among flows with different RTT_{base}. Setting the buffer size of PFIFO too small can also cause throughput degradation due to link underutilization.

On the other hand, we have illustrated that using AQMs can help to reduce the delay experienced by flows, which means that end users will experience higher responsiveness of their TCP-based applications and better interactivity for any other flows sharing the bottleneck. In addition, AQMs improve bandwidth sharing for flows with the same RTT_{base} while FQ-CoDel in particular provides good throughput fairness among flows with different RTT_{base}. FQ-CoDel seems a good candidate for deployment at either end of future broadband last-mile links.

[31]Throughput is even lower through a PFIFO bottleneck with 20- or 180-packet buffers, as these are much smaller than the BDP.

Appendix D
Protecting IoT/Telemedicine Application Flows

A wide range of enabling technologies have reached a state where they can be safely applied by the healthcare industry to improve treatment quality and to make healthcare more accessible, especially to those in remote regions. The use of telecommunication and information technologies to provide clinical health care at a remote distance is called telemedicine.

Here we use the testbed in Appendix A to explore how replacing FIFO queue management with AQM in home gateways will improve the ability of IoT-like telemedicine traffic to co-exist with other classes of traffic. We find that FQ-CoDel can provide good protection in terms of throughput and delay for the IoT flows in all tested cases, which cannot be achieved with PIE, CoDel or FIFO queue management. Although we also observe FQ-CoDel having some negative impact on loss-sensitive UDP flows at low bandwidth, and on the throughput of bulk TCP flows with high RTT_{base} or at high bandwidth, we conclude it remains a better option than PIE or CoDel for home gateways in homes where IoT applications share broadband connection with traditional Internet traffic.

The rest of this section is structured as follows: In subsection D-A, we introduce the challenge of mixing telemedicine and regular Internet traffic in the consumer context, then provide an overview of telemedicine including system architecture and telemedicine applications in subsection D-B. Subsection D-C presents types of traffic we consider in our experiments and how we generate them. The experiment setup is described in subsection D-D, which is followed by experimental results and discussion presented in subsection D-E. We offer concluding remarks in subsection D-F.

A. Mixing telemedicine and regular Internet traffic

Technologies used in telemedicine include the transmission of text, image, vital signs, sound, voice and video via wireless or wired networks. The development of portable, wearable or implantable devices help to make telemedicine more popular. Patients now can access good healthcare service such as diagnosis, treatment guidance and remote prescription from world-wide healthcare professionals without the need of travelling. Telemedicine not only enables patients to use healthcare service from home but also allows healthcare professionals such as nurses, doctors and specialists to cooperate with each other more timely and efficiently. In this report we focus on telemedicine traffic in home networks.

There are a variety of telemedicine applications which are in use or expected to become popular in the futurre. Such examples include teleconference consulation, transfer of medical images and monitoring vital signs such as heart rate, glucose level and blood pressure. Different applications handle different types of traffic with different characteristics as well as different QoS requirement. Because telemedicine traffic is very important, its QoS should be guranteed to the highest level. Therefore, it is important to understand the characteristics of different types of telemedicine traffic. This will help to analyze the performance of these traffic types under different network settings, from which recommendations on bandwidth choice and network technologies can be made.

Among different types of telemedicine traffic, we are particularly interested in remote health monitoring because it represents Internet of Things (IoT) applications that are becoming more and more popular while other telemedicine traffic such as video consulation and medical data transfer is naturally like traditional Internet traffic. IoT is the network of physical objects with the capability to connect to each other and to other Internet-enabled devices and systems. IoT within the home is being realised through the emergence of Internet enabled devices including smart appliances, smart sensors and health and fitness monitors. Such systems can provide services such as home security and safety, energy management and remote health monitoring [40].

IoT applications usually exhibit event-triggered or low-rate traffic patterns. Telehealth telemonitoring is of the second type. Many IoT scenarios involve objects sending telemetry offsite, so IoT flows compete for (often scarce) upstream capacity out of the home. Even when requiring only low bandwidth, IoT flows often have strict QoS requirements (such as consistent delay and low packet loss) due to their importance such as realtime telehealth monitoring or realtime surveilliance [41]. Competition over a shared broadband connection with aggressive Internet traffic (such as bulk data transfer or video streaming) can violate these QoS requirements. Protecting IoT flows is an important new task for modern home gateways, for which there are several approaches.

As noted in Section V, congestion in FIFO-based home gateways can easily lead to significant spikes in RTT experienced by all traffic – conventional, latency-tolerant and IoT flows alike. One solution is multi-queue traffic differentiation inside the home gateway, with IoT flows being assigned to a high priority queue that always gets good service and low RTT [42]. The alternative we investigate here is the impact of replacing FIFO queues in the home gateway with the AQM techniques noted in Section V-B.

B. Overview of telemedicine

1) End-to-end architecture of a telemedicine system: In telemedicine, the communication can happen between machine and machine (M2M), between machine and human, or between human and human. Among those, communication between human and human such as teleconsulation can happen over traditional PSTN or Internet. Communication between machine and machine is an important part of telemedicine, example of which is the transmission of vital signs of a person from his house to a remote application which monitors patients' status and gives alarm to authorized people if necessary.

There has been much work to propose different architectures for M2M telemedicine systems. Many of those mainly use wireless technologies to transmit medical traffic to application servers. In general, these architectures can be generalized by the 3-domain horizontal architecture proposed by ETSI for M2M system: (i) device-gateway domain where all devices communicate with a gateway in a short-range area networks,

(ii) network domain which connects gateway to application servers via long-range access and core networks, (iii) application domain where various application services are defined [43].

Devices in the device-gateway domain are those which can transmit data autonomously or after receiving a data request. For telemedicine, these devices are low-power medical sensors and actuators which are wearable or implanted inside human body. Each device has built-in module for wireless communication with gateway via short-range network.

The gateway in the device-gateway domain is responsible for communicating with sensors, process and forward sensor data seamsly to the network domain. We refer to this as a "sensor gateway" to distinguish from the home gateway which connects to the ISP network. The network domain (i.e. home broadband Internet connection) can itself be wireless or wired.

2) Classification of telemedicine applications: In this section, we provide a summary of potential telemedicine applications which may be widely used in the near future. Those can be classified into five groups: tele-monitoring, telediagnosis, tele-consulation, tele-education, tele-management [44]. In the following, we only mention about the three most common groups in home care: telemonitoring, telediagnosis and teleconsulation.

a) Telemonitoring: To monitor the status of patients, their vital signs such as ECG, blood pressure, pulse oximetry and glucose level are periodically collected by implanted or wearable sensors. Then, they are transmitted to a patient portal which stores and/or process all information. The patient portal will be accessed by healthcare professionals either immediately to respond to any alarms or later to check/diagnose patients' condition. Besides, the measurement of these vital signs can be used by actuators to perform immediate action upon detected abnomalities. For example, when the sensor detects a sudden drop in glucose level, it will send a signal to the actuator to start the injection of insulin [45].

Beside the common vital signs mentioned above, there are a variety of other signs to be monitored depending on the status of each patient. A summary of the signs and their application bitrates are provided in Table V. Note that this bitrate is given at the application level. There have been not much information about how to map these bitrates to packet rates in the literature. One of the reasons can be the dependence of packet size on the chosen networking technology. For example, if ZigBee is chosen as the short-range access technology, the packet size will be limited by 128 bytes. Secondly, the packet size also depends on how many samples to pack in one packet, which depends on how often the value of a sign needs to be updated.

As can be seen in Table V, the bitrates vary significantly for different types of sign. Besides, for a particular sign, there can be different possible bitrates due to several differences such as sampling rate, ADC resolution and number of channels, which are appropriate in different contexts. For example, to monitor people with stable health condition, a lower bitrate device many be enough while the high bitrate ones are suitable for patients under supervision.

b) Telediagnosis: Telediagnosis services are generally characterized by asynchronous point-to-point communication. For example, a remote specialist review transmitted patient information and then return a diagnosis report [44]. The patient information can be text, image, recorded audio or video. Table Table VI on page 30 gives a brief summary of medical images and their sizes.

c) Teleconsulation: For home care, teleconsulation is the synchronous communication between patient and a remote healthcare professionals such as doctors/specialist/nurses. This involves real-time audio or audio transmission.

At the moment, there have been a few applications developed for teleconsulation which are in use [51].

3) Quality of Service: For services in telemonitoring and telediagnosis categories, their QoS requirements in terms of delay and packet loss depend on whether the service is in real-time (emergency scenarios) or not (non-emergency scenarios) [44]. An example of emergency scenario is the monitoring of a patient under treatment while non-emergency case can be the monitoring of a patient in stable condition. A summary of QoS requirements for common telemedicine services are shown in Table Table VII on page 31.

4) Transport protocols: Another important topic of telemedicine is the choice of transport protocol to be used to transmit telemedicine traffic. Traditional Internet traffic such file transfer, video streaming, email and web browsing uses UDP or TCP for end-to-end communication. The choice of the transport protcocol depends on traffic characteristics. For example, UDP is used for delay-sensitive but loss intolerant traffic such as audio/video conference. Meanwhile, TCP is used for delay-tolerant but loss-intolerant such as file transfer.

To communicate with remote health professionals or remote patient portals over Internet, UDP or TCP should also be used to send telemedicine traffic. Telemedicine traffic such as voice/video conference, image/recorded video transfer are just like traditional Internet services. For telemonitoring traffic, the communication between the gateway and remote enitity can use UDP or TCP, depending on the messaging protocol such as CoAP [52] (run over UDP) or MQTT (run over TCP). Meanwhile, the communication between sensors and the gateway in the short range access network can be IP-based or non-IP. For IP-based, UDP can be used as transport protocol while for non-IP communication such as ZigBee, MQTT-SN is an example of potiential transport protocol.

Note that MQTT and CoAP are two typical examples of IoT application protocols. MQTT is messaging protocol based on Publish/Subscribe model. It is lightweight and data agnostic. In this model, all clients connect to a MQTT server to subscribe to a given topic or publish the content of a given topic. When a client has data to send, it will publish data to the server and the server will automatically send the data to clients subscribing to the topic of the sending client. This protocol runs on top of TCP and requires that clients have to maintain a persistent TCP connection to server which is not suitable for power-constrained clients such as sensors. Therefore MQTT-SN [53], a version for sensor networks, has been proposed which allows

TABLE V: Signs to be monitored and their bitrates

Name	Description	Application bitrate
Pulse oximeter (SpO2)	Hemoglobin oxygen saturation	2 kbps [43], 16 bps (1Hz, 8bits) [45][46]
Accelerometers	Motion detection, acceleration	35kbps (500Hz, 12bits) [45][46], <10kbps [47]
Glucose monitoring		1.6Kbps [45]
Body temperature		2.4 bps [43], 80bps (5 samples*16bits/sample) [48],120bps [46]
Blood pressure		16bps (1 sample*16 bits/sample) [48], 1.9Kbps [49]
Respiration		<10kpbs [47][48]
Electrocardiogram (ECG)	Heart waveform characteristics	3.2kbps (1 channel, 3 lead, 16 bit) [49], 9.6 kbps (3 channels) [43], 15kbps [48], 71kbps (6 leads, 500Hz sample, 12 bits) [45]
Electroencephalo-gram (EEG)	Brainwave activity	100 kbps [43], 98.304 kpbs [49], 86.4 kbps (300Hz sample, 12-bit ADC, 24 channels) [47], 43.2kpbs (12 leads, 150Hz, 12 bits) [45], 10kbps (350 samples/s*12bits/sample) [48]
Electromyograph (EMG)	Muscle movement	100 kbps [43], 320kbps (10kHz, 16bits) [45][46], 1.536Mbps (8kHz, 16bit ADC, 12 channels) [47], 600kpbs (50000 samples*12bits/sample) [48]
Fall detection		250 kbps [43]

TABLE VI: Medical images

Name	Size
Ultrasound, cardiology, radiology	256kB [48][50]
Magnetic resonance image	384kB [48]
Scanned X-ray	1.8MB [48]
Digital radiography	6MB - 50MB [48]
Mammogram	24MB [48][50]
Digital cameras	User defined [50]

clients to sleep if they don't have data to send.

CoAP is based on request/response model like RESTful HTTP but it has much smaller overhead and runs on top of UDP, which makes it suitable for limited-resource devices. It has the bulit-in QoS mechanism via non-confirmable or confirmable messages. A confirmable message needs to be acknowledged before a timer expires or it will be restransmitted with exponential backoff between retransmissions.

C. IoT applications and traffic generator

We focus on IoT applications that generate regular offsite traffic, categorized into three groups: health telemonitoring, energy efficiency, and home security [54]. Because health telemonitoring applications were described in subsection D-B2a, here we briefly summarize the other two groups. Then, we present how IoT traffic is generated in our experiments.

1) Non-health IoT applications:

a) Energy efficiency: The emergence of smart appliances such as ovens, fridges, dryers, washing machine, light bulbs, meters and thermostats can lead to the development of home energy management to reduce the cost of energy provision in households. These smart appliances can send their real-time energy consumption to a remote server which uses it to analyse, optimize and then control the energy usage of these appliances in an effective manner.

The bitrates required by smart appliances for energy management purpose can vary between 1.4Kbps and 250Kbps [54].

b) Home security: These applications range from simple event-based alarm/detection sensors to real-time video surveillance. While event-based traffic is very low bitrate, real-time surveillance generates consistent outbound traffic from tens of Kbps to more than 10Mbps depending on camera quality [55].

2) Traffic generator for telehealth monitoring IoT flows:

a) Traffic pattern: Note that we only consider the traffic of monitored signs from the sensor gateway onward. We assume there is wired connection between sensor gateway and home gateway. Packets received by sensor gateway will be re-encapsulated if the short range access network is non-IP so that they can be sent via IP network. Besides, we assume that traffic pattern of each flow sent by the sensor gateway is the same as that sent by the corresponding sensor.

Each sensor will take the measurement of the corresponding sign every sampling interval. There are no standards on how to combine these samples into a packet. It can be one sample per packet or multiple samples per packet depending on the sensor application. In this report, we assume that the sensors will send a packet at least every second as long as the packet size does not exceed the maximum packet size allowed by underlying networking protocol. In particular, we assume that the maximum payload size of a packet (excluding TCP/IP header) is 100B as in ZigBee protocol [49]. For a 16-bit ADC resolution, this will be equivalent to a maximum of 50 samples per packet.

In summary, we model telehealth monitoring traffic as fixed-size packets sent at fixed intervals, where intervals and packet sizes derive from the sampling rate and ADC resolution of each sensor. We emulate IoT applications that use TCP transport for reliability.

b) Parameter choice: From Table V, the signs can be divided into 2 groups: high bitrates ($\geq$10Kbps, such as EMG and EEG) and low bitrates ($<$10Kbps, such as blood pressure and glucose level). We choose 2Kbps and 100Kbps to represent low and high rate IoT flows. These two bit rates are also representative for IoT applications of other groups.

We assume 16-bit ADC resolution. According to the assumption in subsectionD-C2a, a sensor will send a packet of 100B every 50 samples if the sampling rate is at least 50Hz. Otherwise, it will send a packet of smaller than 100B every second.

- For 2Kbps, the sampling rate is (2000/16) or 125Hz. This is greater than 50Hz; therefore, this sensor sends 100B packets every 50 samples which is 50*(1/125) or 0.4s.

TABLE VII: Summary of QoS requirements for e-Health services [44].

Types of e-health services	Example e-health application	Comonly used media types	General QoS requirements	
			E2E one-way delay	Packet loss ratio
Real-time conversational teleconsulation	Audio conferencing between patient/doctor or doctor/doctor	Audio	$<$150ms	$<$1% , prefered $<$3% limit
Real-time conversational video-based teleconsulation	Video conferencing between patient/doctor or doctor/doctor	Video	$<$250ms (upper bounds reported as 400ms)	1%
Real-time telemonitoring	Transmission of patient vital signs and streaming video in emergency situtations	Biomedical data collected by sensors	Depends on sensors and applications, $<$300ms for hard real-time ECG (certain applications may tolerate $<$1s for ECG)	Zero
Non real-time telemonitoring	Transmission of patient vital signs for post-hospital home care	Biomedical data collected by sensors, context data (e.g., collected by environmental sensors)	Not Available (N.A.)	Zero
Real-time telediagnosis	Transfer of medical images to remote location in emergency situations	Images, text, data	N.A. (Depends on image size. Smaller images should be transferred within a few seconds.)	Zero
Non real-time telediagnosis	Non-emergency remote diagnosis: transfer of medical images to a remote location where specialists analyze data and return a diagnostic report.	Images, text, data	N.A.	Zero
Real-time EHR data access	Emergency medical personnel at accident/disaster site accessing a patient's EHR	Data, text, graphics, images	N.A.	Zero
Non real-time EHR data access/storage	Web-based end user (patient, doctor, additional health personnel) application for access to EHR during patient check up	Data, text, graphics, images	N.A.	Zero

This means the packet rate is 1/0.4 or 2.5packets/s.

- For 100Kbps, the sampling rate is (100000/16) or 6250Hz. This is greater than 50Hz; therefore, this sensor sends 100B packets every 50 samples which is 50*(1/6250) or 0.008s. This means that the packet rate is 1/0.008 or 125packets/s.

D. Experiment setup

Here we describe our testbed and the simple scenarios for exploring interactions between IoT and non-IoT traffic.

1) A home network blending IoT and traditional traffic: We envisage a home where IoT devices and other consumer services simultaneously access remote servers across their ISP's broadband last-mile link. We explore the case where the ISP's last-mile provides rates of $R_{down/up} = \{1.5/0.5, 12/1, 25/5\}Mbps$ (a subset of Table III).

We use a range of RTT_{base} drawn from Table I for non-IoT flows involving both domestic and international servers, and $RTT_{base} = 40ms$ for IoT flows to emulate tele-health/telemonitoring services based nearby (such as a local hospital or related medical provider) .

Our experiments emulate the case where IoT-like flows are already in progress when overlapping non-IoT traffic flows begin. Low-rate and high-rate IoT flows (IoT_{low} and IoT_{high}) start at $t = 0s$, and $t = 10s$ respectively. A video call begins at $t = 30s$ and a bulk file upload begins at $t = 70s$. A DASH (Dynamic Adaptive Streaming over HTTP [3]) video flow starts at $t = 120s$, followed by a bulk file download at $t = 180s$. All flows end at $t = 250s$.

2) Testbed topology and test conditions: The experiments ran on the testbed described in Appendix A, except that hosts run 64-bit FreeBSD 10.2-RELEASE-p7 with NewReno as the default TCP algorithm. Hosts emulating telemonitoring traffic have Nagles algorithm [56] disabled on sources and delayed ACKs disabled on sinks.[32]

a) Traffic generation: The following traffic types were generated at the application layer:

- IoT_{low}: Low-rate telemonitoring traffic at 2Kbps, using `nttcp` to send 100B packets at 2.5 packets/sec upstream over a TCP connection to a remote server.
- IoT_{high}: High-rate telemonitoring traffic at 100Kbps, using `nttcp` to send 100B packets at 125 packets/sec upstream over a TCP connection to a remote server.
- VideoCall: A 280Kbps video call, using `nttcp` to generate concurrent upstream and downstream UDP flows of 700B packets every 20ms.
- BulkUpload/BulkDownload: Bulk file transfer, using `iperf` to send full-size packets over a TCP connection as fast as possible upstream or downstream .
- DASH: A DASH over TCP video flow retrieving video segments between 46Kbps and 4.2Mbps every 2s.

To generate IoT_{low} and IoT_{high} we use `nttcp` to send TCP traffic with the gap betweens packets set by the option `-g` and the packet size set by `-l` and the total number of packets set by `-n`. The total number of packets is equal to duration divided by the packet gap. The TEACUP (Appendix A) configuration

[32]When enabled, these features can be detrimental to low-rate TCP flows.

lines to generate 2Kbps low-rate traffic and 100Kbps high-rate traffic are:

```
Traffic_lowrate_IoT = [ ('0.0', '1',
``start_nttcp_tcp, client='newtcp1',
server='newtcp4', port=5004, duration=100,
interval=400, psize=100, extra_params_client='-D'
''), ]
```

```
Traffic_highrate_IoT = [ ('0.0', '1',
`` start_nttcp_tcp, client='newtcp1',
server='newtcp4', port=5004, duration=100,
interval=8, psize=100, extra_params_client='-D'
``), ]
```

E. Results and discussion

In this section we review the experienced RTT and throughput distributions see during different combinations of IoT and non-IoT flows.

We ran the experiments as described in subsection D-D1 using the traffic generators in subsection D-D2a. In order to analyse the results we split each experiment into four *scenarios* corresponding to the four intervals with different combinations of overlapping application flows. We limit our discussion to trials covering $R_{down/up} = \{1.5/0.5, 12/1, 25/5\}Mbps$ and with $RTT_{base} = \{40ms, 340ms\}$ for the non-IoT flows (VideoCall, BulkUpload, DASH, and BulkDownload). Keep in mind that RTT_{base} of IoT_{low} and IoT_{high} flows was fixed at 40ms for all experiments.

Note, throughput is measured at the IP layer. So the transport and IP layer headers should be added to traffic sources in subsection D-D2a to get their IP-layer bitrates (2.8Kpbs and 140Kbps for the IoT_{low} and IoT_{high} traffic respectively).

1) Scenario 1: We emulate 40 seconds of only IoT_{low}, IoT_{high} and VideoCall flows overlapping. Their combined bitrates are less than 500Kbps. Hence all three flows achieved their target bitrates, with no significant queuing delays, for all $R_{down/up} = \{1.5/0.5, 12/1, 25/5\}Mbps$ using PFIFO, PIE, CoDel and FQ-CoDel.

2) Scenario 2: We emulate 50 seconds of two pre-existing telemonitoring flows (IoT_{low} and IoT_{high}) and a video consulation (VideoCall) being overlapped with a bulk upload of medical data (BulkUpload).

a) $R_{down/up} = 1.5/0.5Mbps$: This case is captured by Figs. 22 and 23. With PFIFO, all four flows experience RTT>600ms when $RTT_{base} = 40ms$ and the non-IoT flows experience RTT>1s when $RTT_{base} = 340ms$. In contrast, with PIE, CoDel and FQ-CoDel, the RTT of all four flows are <250ms when $RTT_{base} = 40ms$ and the RTT of the non-IoT flows are <500ms when $RTT_{base} = 340ms$. The IoT flows experience the smallest RTT mean and variance with FQ-CoDel.

IoT_{high} consistently achieves its required bitrate with FQ-CoDel while its throughput is noticably smaller with PFIFO, PIE and CoDel. IoT_{low} throughput is similar among PFIFO and the AQMs and is equal to its average bitrate. Moreover, VideoCall experiences no packet loss in the downstream. However, while the packet loss rate in the upstream is <2%

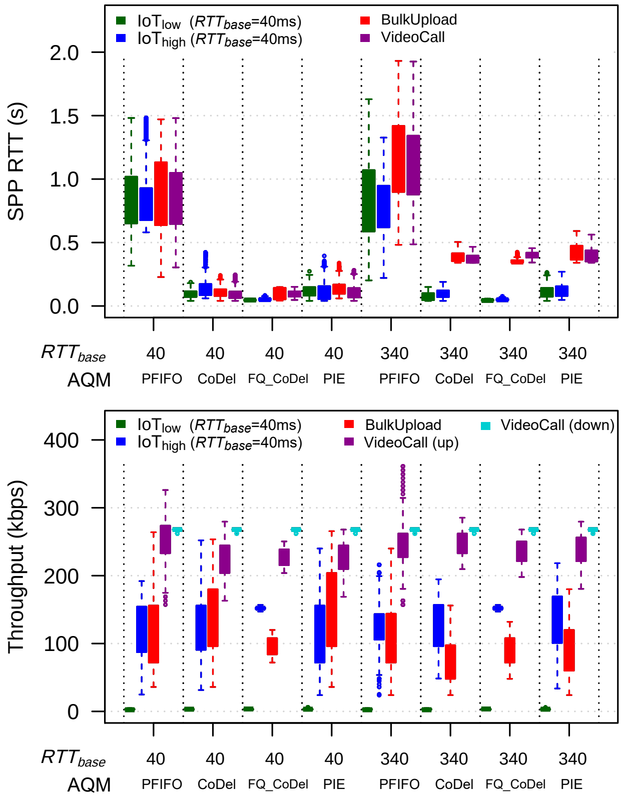


Fig. 22: Throughput and RTT of all flows at $R_{down/up} = 1.5/0.5Mbps$ (Scenario 2). $RTT_{base} = \{40, 340\}ms$ for non-IoT flows

with PFIFO, it is >10% when $RTT_{base} = 40ms$ and >5% when $RTT_{base} = 340ms$ with the three AQMs. BulkUpload throughput is similar among PFIFO, CoDel and PIE while it is smaller with FQ-CoDel when $RTT_{base} = 40ms$. With the three AQMs, when RTT_{base} increases to 340ms, BulkUpload throughput noticably decreases because it takes longer for the TCP congestion window to recover after a packet drop. The reduction of BulkUpload throughput leads to the throughput improvement of other flows.

b) $R_{down/up} = 12/1Mbps$: The results at $12/1Mbps$ are shown in Figs. 24 and 25. Compared with$1.5/0.5Mbps$, all flows observe smaller RTT at $12/1Mbps$. With PFIFO, all four flows experience RTT<700ms when $RTT_{base} = 40ms$ and the non-IoT flows experience RTT<1s when $RTT_{base} = 340ms$. With PIE, CoDel and FQ-CoDel, the RTT of all four flows are mostly <150ms when $RTT_{base} = 40ms$ and the RTT of the non-IoT flows are <400ms when $RTT_{base} = 340ms$. Again, the IoT flows experience the smallest RTT mean and variance with FQ-CoDel.

While IoT_{high} throughput consistently equals its required bitrate with FQ-CoDel at all RTT_{base}, it is slightly less with PFIFO, PIE and CoDel when $RTT_{base} = 40ms$. IoT_{low} has throughput equal to its bitrate with PFIFO and the AQMs. There is no packet loss in the downstream of VideoCall. While the packet loss rate in the upstream is <1% with

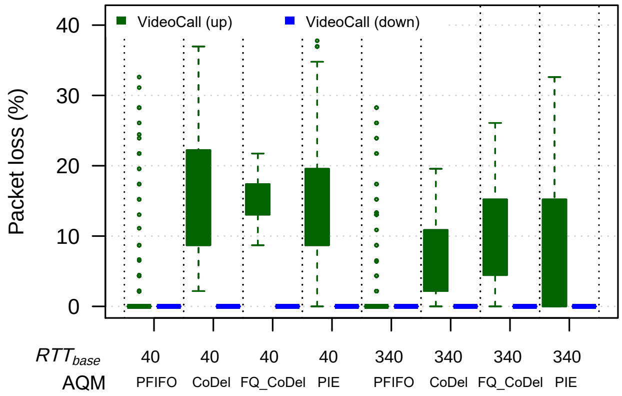


Fig. 23: Packet loss of the video call in both directions at $R_{down/up} = 1.5/0.5Mbps$ (Scenario 2).

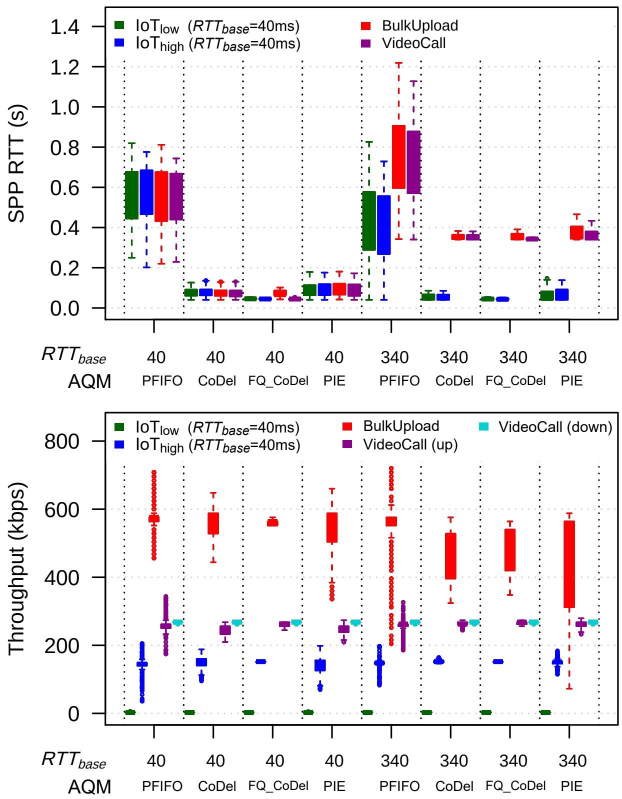


Fig. 24: Throughput and RTT of all flows at $R_{down/up} = 12/1Mbps$ (Scenario 2). $RTT_{base} = \{40, 340\}ms$ for non-IoT flows

PFIFO at all RTT_{base}, it is >5% with PIE and CoDel when $RTT_{base} = 40ms$. Among the AQMs, FQ-CoDel has the lowest packet loss rate, <3% when $RTT_{base} = 40ms$ and <1% when $RTT_{base} = 340ms$. BulkUpload throughput is higher with PFIFO than with the AQMs, especially at high RTT_{base}. With the three AQMs, BulkUpload throughput also decreases noticably when RTT_{base} increases to 340ms.

c) $R_{down/up} = 25/5Mbps$: The results at $25/5Mbps$ in Figs. 26 and 27 show that all four flows observe smaller

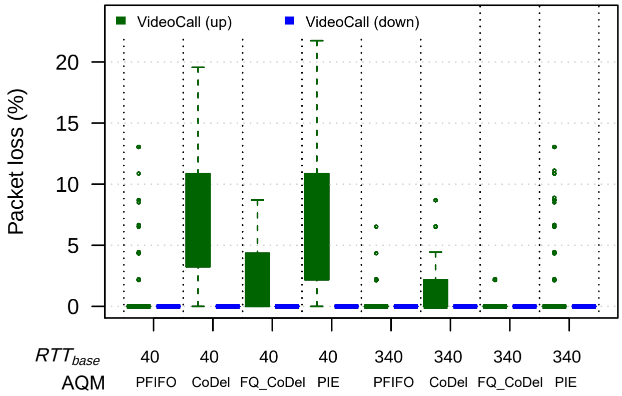


Fig. 25: Packet loss of the video call in both directions at $R_{down/up} = 12/1Mbps$ (Scenario 2).

RTT than at $12/1Mbps$. With PFIFO, all four flows experience RTT<300ms when $RTT_{base} = 40ms$ and the non-IoT flows experience RTT<600ms when $RTT_{base} = 340ms$. With PIE, CoDel and FQ-CoDel, the RTT of all four flows are <100ms when $RTT_{base} = 40ms$ and the RTT of the non-IoT flows are <400ms when $RTT_{base} = 340ms$. The IoT flows again have the smallest RTT mean and variance with FQ-CoDel.

IoT_{low} and IoT_{high} achieve their required bitrates with PFIFO and the three AQMs. Similarly, VideoCall throughput in both upstream and downstream is approximately equal to its sending bitrate, with negligible packet loss. BulkUpload throughput is noticably higher with PFIFO than with the AQMs because the effect of packet dropping is more pronounced at high bandwidth. This throughput difference between PFIFO and the AQMs becomes clearer when RTT_{base} increases to 340ms. Among the AQMs, BulkUpload has the highest throughput with PIE probably due to its highest T_{target}.

3) Scenario 3: We emulate 60 seconds of two pre-existing telemonitoring flows (IoT_{low} and IoT_{high}), a video consulation (VideoCall) and a bulk upload of medical data (BulkUpload) being overlapped with a DASH video stream. Adding the DASH traffic reduces BulkUpload throughput and, except under FQ-CoDel, degrades IoT_{high} throughput at low upstream bandwidths. The RTT and throughput of all flows improves as$R_{down/up}$ increases.

a) $R_{down/up} = 1.5/0.5Mbps$: Figs. 28 and 29 show the results at $1.5/0.5Mbps$. With PFIFO, all five flows experience RTT>1s. In contrast, with PIE, CoDel and FQ-CoDel, the RTT of all five flows are <250ms when $RTT_{base} = 40ms$ and the RTT of the non-IoT flows are <500ms when $RTT_{base} = 340ms$. Among the AQMs and PFIFO, the IoT flows experience the smallest RTT mean and variance with FQ-CoDel.

IoT_{high} achieves its required bitrate with FQ-CoDel while its throughput is noticably smaller with PFIFO, PIE and CoDel. IoT_{low} throughput is similar among PFIFO and the AQMs and is equal to its average bitrate. VideoCall experiences negligible packet loss in the downstream. However, while the packet loss rate in the upstream is negligible with PFIFO, it is >10% when $RTT_{base} = 40ms$ and >5% when $RTT_{base} = 340ms$

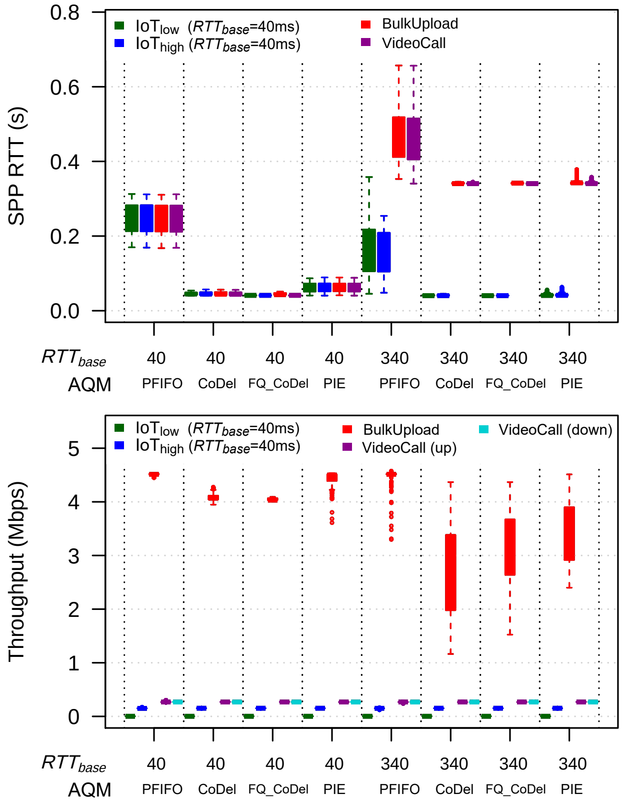


Fig. 26: Throughput and RTT of all flows at $R_{down/up} = 25/5Mbps$ (Scenario 2). $RTT_{base} = \{40, 340\}ms$ for non-IoT flows

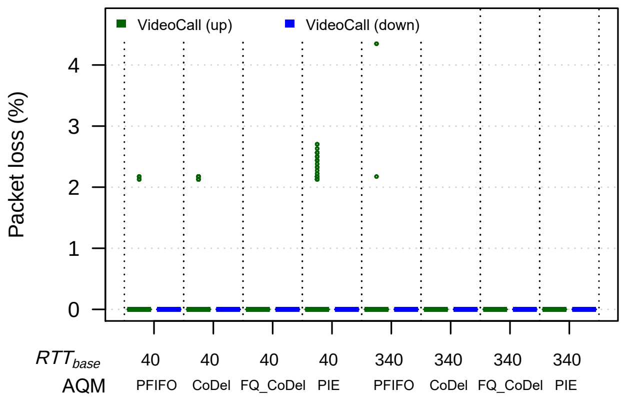


Fig. 27: Packet loss of the video call in both directions at $R_{down/up} = 25/5Mbps$ (Scenario 2).

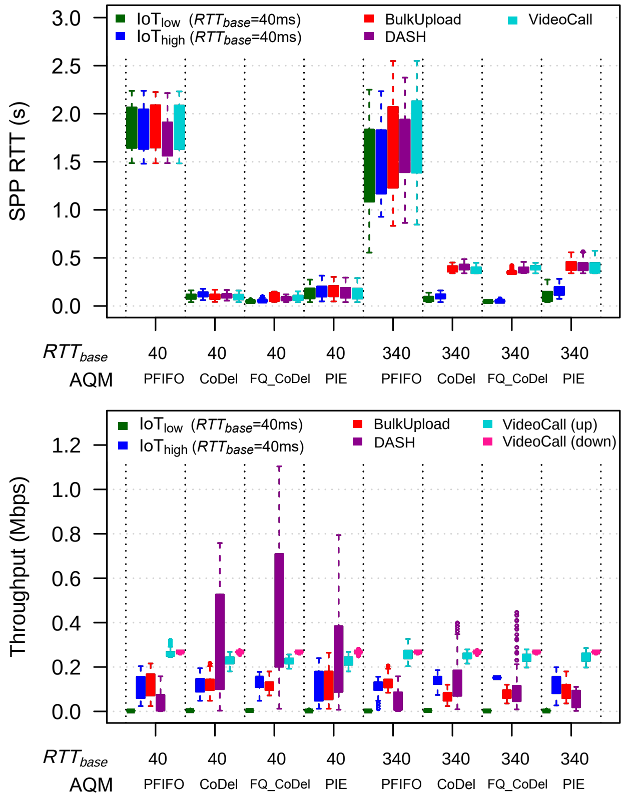


Fig. 28: Throughput and RTT of all flows at $R_{down/up} = 1.5/0.5Mbps$ (Scenario 3). $RTT_{base} = \{40, 340\}ms$ for non-IoT flows

with the three AQMs. BulkUpload achieves similar throughput among PFIFO and the AQMs when $RTT_{base} = 40ms$. With the three AQMs, when RTT_{base} increases to 340ms, BulkUpload throughput noticably decreases due to their packet dropping schemes. The reduction of BulkUpload throughput leads to the throughput improvement of the IoT flows and VideoCall. DASH throughput is much smaller with PFIFO than with the three AQMs when $RTT_{base} = 40ms$, which may be because the DASH client requests lower quality video segments when it detects poor network conditions such as high RTT. With the three AQMs, DASH throughput noticably reduces when RTT_{base} increases to 340ms in a similar way to BulkUpload but more noticeably because of the rate adaptation scheme of the DASH client described above. The reduction of DASH throughput at higher RTT_{base} also helps to improve the performance of both IoT flows and VideoCall.

b) $R_{down/up} = 12/1Mbps$: The results at $12/1Mbps$ are shown in Figs. 30 and 31. With PFIFO, all five flows experience RTT<700ms when $RTT_{base} = 40ms$ and the non-IoT flows experience RTT<1s when $RTT_{base} = 340ms$. With PIE, CoDel and FQ-CoDel, the RTT of all five flows are <150ms when $RTT_{base} = 40ms$ and the RTT of the non-IoT flows are <500ms when $RTT_{base} = 340ms$. The IoT flows again experience the smallest RTT mean and variance with FQ-CoDel.

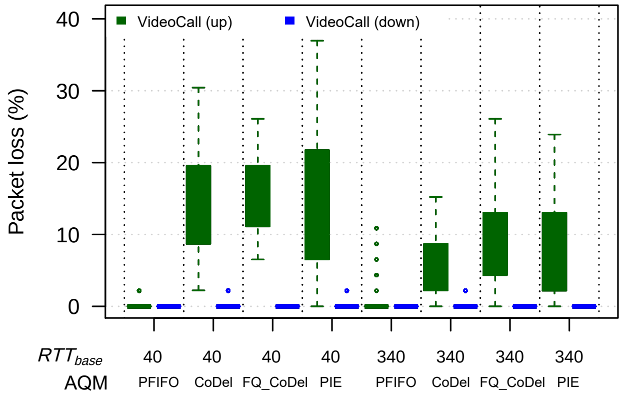


Fig. 29: Packet loss of the video call in both directions at $R_{down/up} = 1.5/0.5Mbps$ (Scenario 3).

IoT$_{\text{high}}$ throughput consistently equals its required bitrate with FQ-CoDel at all RTT_{base} while it is slightly less with PFIFO, PIE and CoDel when $RTT_{base} = 40ms$. IoT$_{\text{low}}$ has throughput equal to its bitrate with PFIFO and the AQMs. Negligible packet loss is observed in the downstream of VideoCall. While the packet loss rate in the upstream is again negligible with PFIFO at all RTT_{base}, it is >4% with PIE and CoDel and <2% with FQ-CoDel when $RTT_{base} = 40ms$. When $RTT_{base} = 340ms$, packet loss rate in the upstream is <2% with the three AQMs. Like the case$12/1Mbps$ in Scenario 2, BulkUpload throughput is higher with PFIFO than with the AQMs especially at high RTT_{base}, and it noticably decreases with RTT_{base} under the AQMs. As at $1.5/0.5Mbps$, DASH throughput is much smaller with PFIFO than with the AQMs when $RTT_{base} = 40ms$. Besides, DASH throughput decreases when RTT_{base} increases to 340ms not only with the AQMs but also with PFIFO due to the downlink buffer much smaller than the bandwidth delay product at this RTT_{base}.

c) $R_{down/up} = 25/5Mbps$: This case is represented in Figs. 32 and 33. With PFIFO, all five flows experience RTT<200ms when $RTT_{base} = 40ms$ and the non-IoT flows experience RTT<500ms when $RTT_{base} = 340ms$. With PIE, CoDel and FQ-CoDel, the RTT of all five flows are mostly <100ms when $RTT_{base} = 40ms$ and the RTT of the non-IoT flows are <500ms when $RTT_{base} = 340ms$. Again, the IoT flows have the smallest RTT mean and variance with FQ-CoDel.

Both IoT$_{\text{low}}$ and IoT$_{\text{high}}$ achieve their required bitrates with PFIFO and the three AQMs. Similarly, VideoCall throughput in both upstream and downstream approximately equals its sending bitrate, with negligible packet loss. BulkUpload throughput has qualitatively similar performance to the case $25/5Mbps$ in Scenario 2 but is quantatively smaller due to the existence of DASH. With the AQMs and PFIFO, DASH throughput reduces significantly when RTT_{base} increases to 340ms, like at $12/1Mbps$. However, DASH throughput under PFIFO is now comparable with that seen with the AQMs because DASH RTT under PFIFO is smaller in this case.

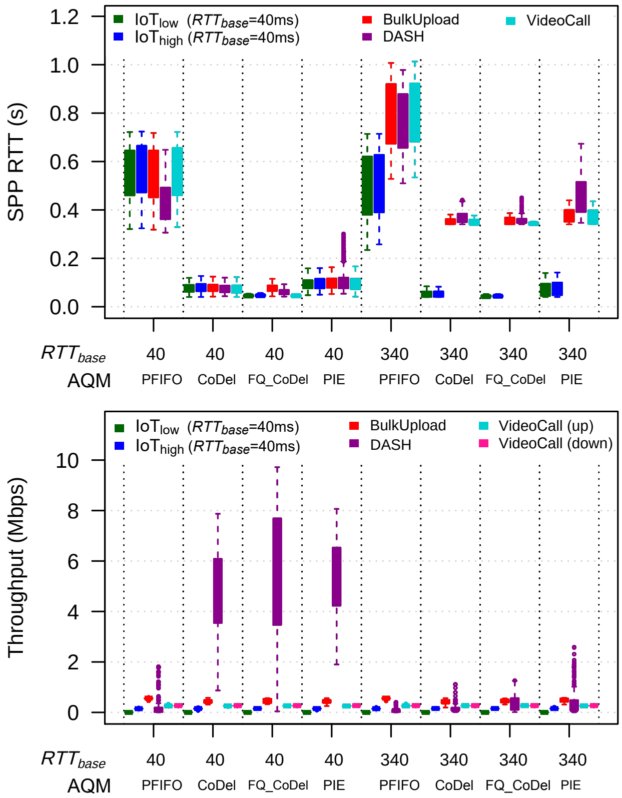


Fig. 30: Throughput and RTT of all flows at $R_{down/up} = 12/1Mbps$ (Scenario 3). $RTT_{base} = \{40, 340\}ms$ for non-IoT flows

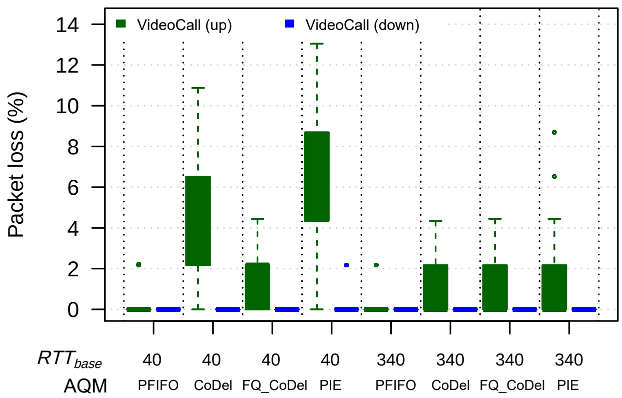


Fig. 31: Packet loss of the video call in both directions at $R_{down/up} = 12/1Mbps$ (Scenario 3).

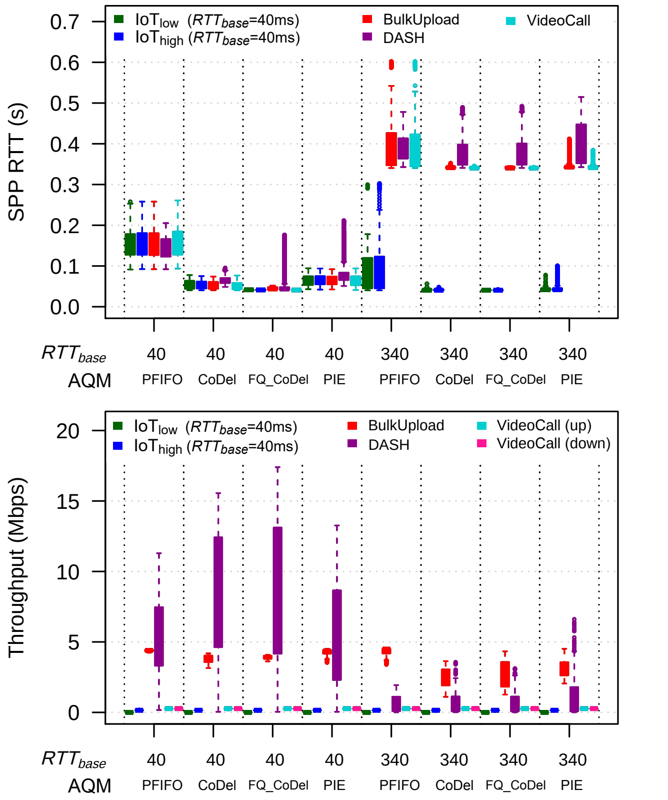


Fig. 32: Throughput and RTT of all flows at $R_{down/up} = 25/5Mbps$ (Scenario 3). $RTT_{base} = \{40, 340\}ms$ for non-IoT flows

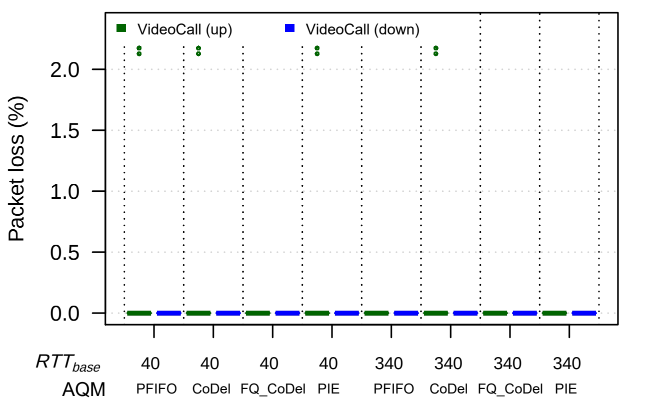


Fig. 33: Packet loss of the video call in both directions at $R_{down/up} = 25/5Mbps$ (Scenario 3).

4) Scenario 4: We emulate 70 seconds of two pre-existing telemonitoring flows (IoT_{low} and IoT_{high}), a video consulation (VideoCall), a bulk upload of medical data (BulkUpload) and DASH video stream being overlapped with a BulkDownload. The BulkDownload causes degradation to the throughput performance of BulkUpload and DASH. The RTT and throughput of all flows improves as$R_{down/up}$increases.

a) $R_{down/up} = 1.5/0.5Mbps$: The results at $1.5/0.5Mbps$ are shown Figs. 34 and 35. With PFIFO, all six flows experience RTT>1s. In contrast, with PIE, CoDel and FQ-CoDel, the RTT of all six flows are mostly <250ms when $RTT_{base} = 40ms$ and the RTT of the non-IoT flows are <500ms when $RTT_{base} = 340ms$. Among the AQMs and PFIFO, the IoT flows experience the smallest RTT mean and variance with FQ-CoDel.

IoT_{high} throughput equals its required bitrate with FQ-CoDel while it is noticably smaller with PFIFO, PIE and CoDel. IoT_{low} throughput is similar among PFIFO and the AQMs and is equal to its average bitrate. With PFIFO, VideoCall has packet loss rate <2% in the upstream and negligible in the downstream at all RTT_{base}. However, with the AQMs, the packet loss rate in the upstream is >10% when $RTT_{base} = 40ms$ and >5% when $RTT_{base} = 340ms$ while the loss rate in the downstream is <2% when $RTT_{base} = 40ms$ and negligible when $RTT_{base} = 340ms$. For BulkUpload and DASH, their throughput performance is qualitatively similar to the case $1.5/0.5Mbps$ in Scenario 3 but quantatively smaller due to the existence of BulkDownload. BulkDownload throughput is significantly larger than DASH because of its aggressive use of available bandwidth. With the three AQMs, BulkDownload throughput decreases noticably when RTT_{base} increases to 340ms due to their packet dropping schemes.

b) $R_{down/up} = 12/1Mbps$: This case is captured in Figs. 36 and 37. With PFIFO, all six flows experience RTT<600ms. With PIE, CoDel and FQ-CoDel, the RTT of all six flows are <150ms when $RTT_{base} = 40ms$ and the RTT of the non-IoT flows are <400ms when $RTT_{base} = 340ms$. The IoT flows experience the smallest RTT mean and variance with FQ-CoDel.

With FQ-CoDel, IoT_{high} throughput consistently equals its required bitrate at all RTT_{base} while it is slightly less with PFIFO, PIE and CoDel when $RTT_{base} = 40ms$. IoT_{low} has throughput equal to its bitrate with PFIFO and the AQMs. VideoCall has negligible downstream packet loss with PFIFO and the three AQMs. Besides, with PFIFO and FQ-CoDel, the packet loss rate in the upstream is <2% when $RTT_{base} = 40ms$ and <1% when $RTT_{base} = 340ms$. With PIE and CoDel, it is >3% when $RTT_{base} = 40ms$ and <2% when $RTT_{base} = 340ms$. BulkUpload throughput is slightly higher with PFIFO than with the AQMs at all RTT_{base} and it is lowest with FQ-CoDel when $RTT_{base} = 40ms$. Besides, DASH throughput is smaller with PFIFO than with the AQMs when $RTT_{base} = 40ms$ and it noticably decreases when RTT_{base} increases to 340ms like the case $12/1Mbps$ in Scenario 3. As at $1.5/0.5Mbps$, BulkDownload has throughput significantly larger than DASH. Besides, DASH throughput decreases when

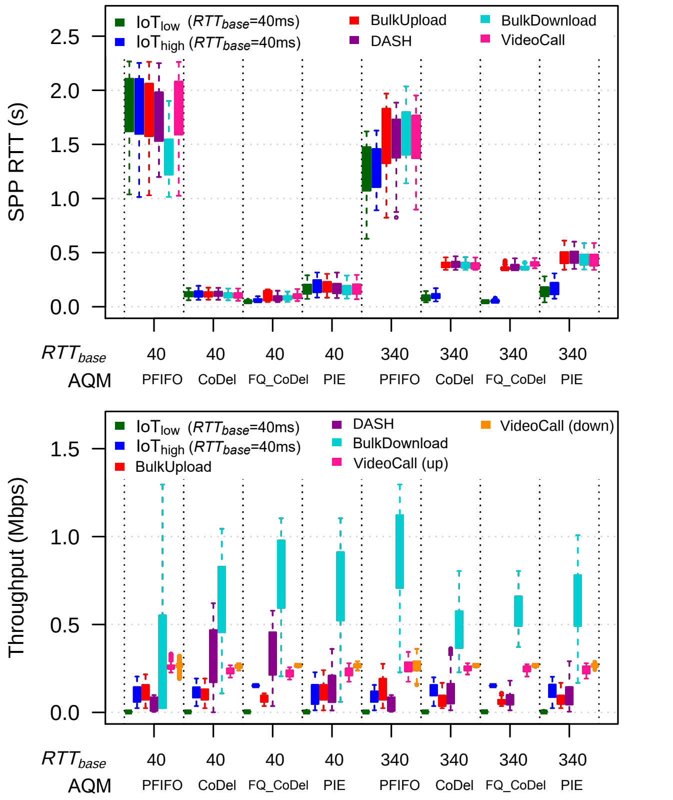


Fig. 34: Throughput and RTT of all flows at $R_{down/up} = 1.5/0.5Mbps$ (Scenario 4). $RTT_{base} = \{40, 340\}ms$ for non-IoT flows

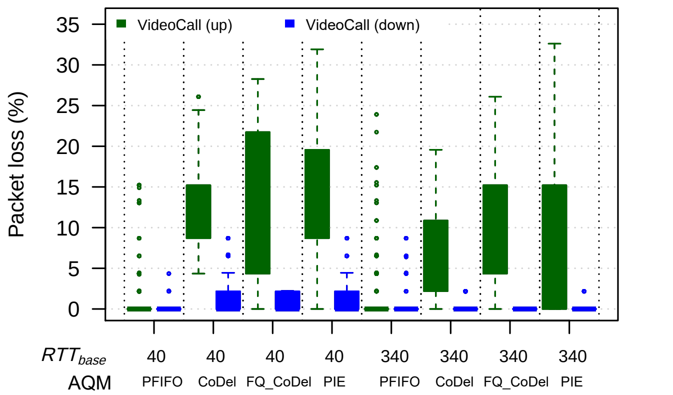


Fig. 35: Packet loss of the video call in both directions at $R_{down/up} = 1.5/0.5Mbps$ (Scenario 4).

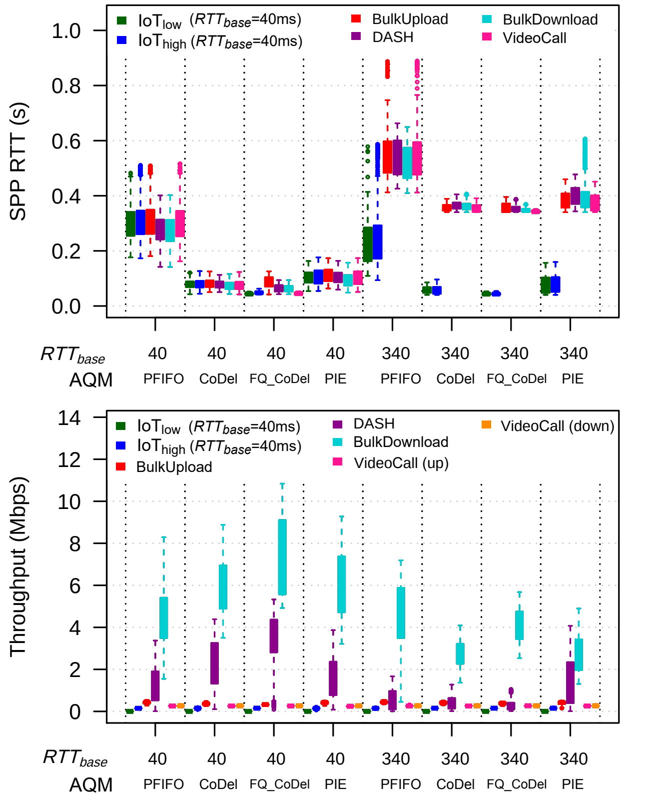


Fig. 36: Throughput and RTT of all flows at $R_{down/up} = 12/1Mbps$ (Scenario 4). $RTT_{base} = \{40, 340\}ms$ for non-IoT flows

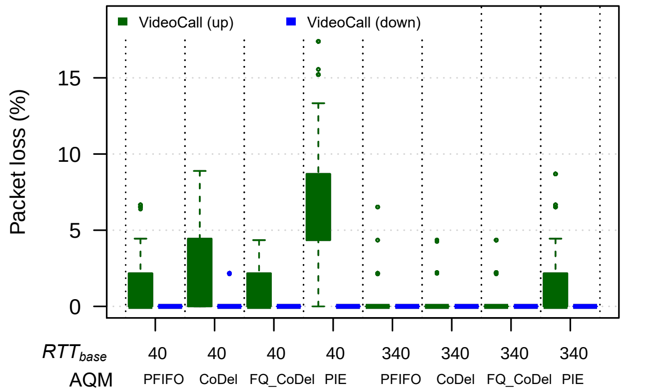


Fig. 37: Packet loss of the video call in both directions at $R_{down/up} = 12/1Mbps$ (Scenario 4).

RTT_{base} increases to 340ms not only with the AQMs but also with PFIFO due to the downlink buffer much smaller than the bandwidth delay product at this RTT_{base}.

c) $R_{down/up} = 25/5Mbps$: Figs. 38 and 39 show the results at $25/5Mbps$. With PFIFO, all six flows experience RTT<150ms when $RTT_{base} = 40ms$ and the non-IoT flows experience RTT<450ms when $RTT_{base} = 340ms$. With PIE, CoDel and FQ-CoDel, the RTT of all six flows are <100ms when $RTT_{base} = 40ms$ and the RTT of the non-IoT flows

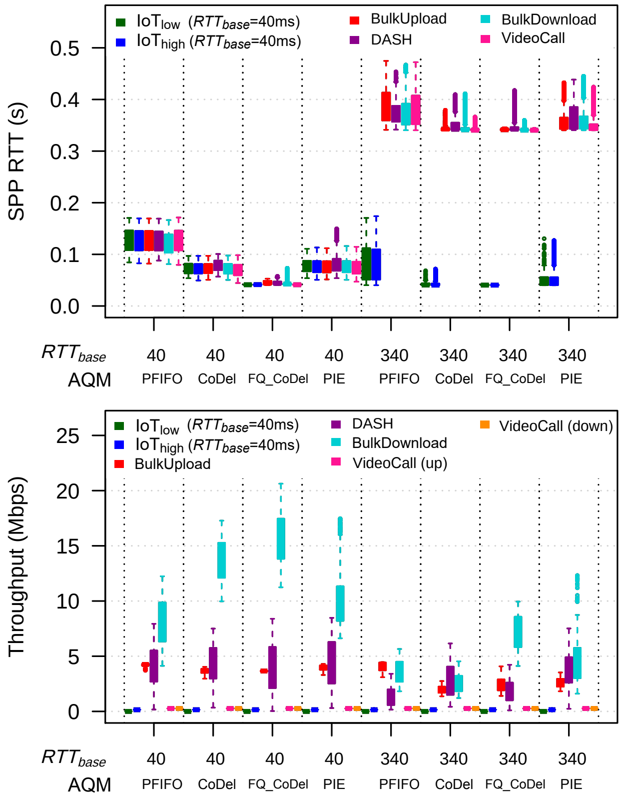


Fig. 38: Throughput and RTT of all flows at $R_{down/up} = 25/5Mbps$ (Scenario 4). $RTT_{base} = \{40, 340\}ms$ for non-IoT flows

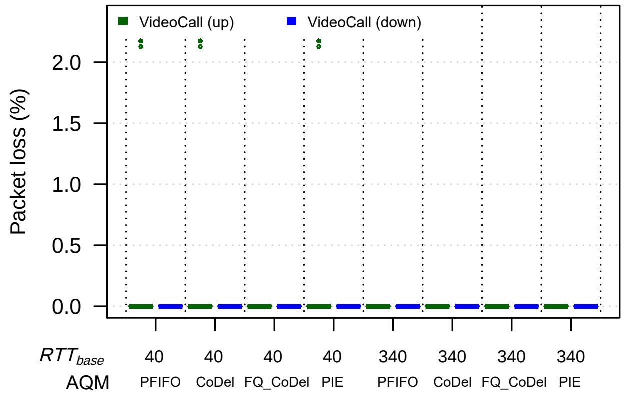


Fig. 39: Packet loss of the video call in both directions at $R_{down/up} = 25/5Mbps$ (Scenario 4).

are <400ms when $RTT_{base} = 340ms$. Again, the IoT flows have the smallest RTT mean and variance with FQ-CoDel.

Both IoT_{low} and IoT_{high} achieve their required bitrates with PFIFO and the three AQMs. Likewise, VideoCall throughput in both upstream and downstream is approximately equal to its sending bitrate, with negligible packet loss. The throughput performance of BulkUpload and DASH are qualitatively similar to the case $25/5Mbps$ in Scenario 3. BulkDownload throughput is noticably larger than DASH when $RTT_{base} = 40ms$ and significantly decreases when RTT_{base} increases to 340ms, like at $12/1Mbps$.

F. Summary

In this section, we have studied the performance of low-rate and high-rate IoT flows when three AQM algorithms, PIE, CoDel and FQ-CoDel, are deployed at the home gateway in a variety of home network scenarios. We confirm that traditional FIFO queueing discipline provides no protection to the delay and throughput of IoT flows, making it inappropriate for real-time IoT traffic. In contrast, our results have shown that with FQ-CoDel, the IoT flows achieve their required bitrates with consistently small delay in all the tested cases. We have observed that all three tested AQMs can cause packet loss in UDP flows such as video conferencing in low bandwidth scenarios and degrade the throughput performance of bulk TCP flows with high RTT or at high bandwidth. Nevertheless, we conclude that FQ-CoDel is more suitable than PIE, CoDel or traditional FIFO for deployment at either ends of last-mile broadband links to support the coexistence of IoT applications and traditional Internet applications in home broadband networks.

REFERENCES

[1] P. Mockapetris, "Domain names - concepts and facilities," RFC 1034, Nov. 1987. [Online]. Available: https://tools.ietf.org/html/rfc1034
[2] M. Allman, V. Paxson, and E. Blanton, "TCP Congestion Control," RFC 5681, Internet Engineering Task Force, Sep. 2009. [Online]. Available: https://tools.ietf.org/html/rfc5681
[3] "ISO/IEC 23009-1:2014 Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats," ISO/IEC, May 2014. [Online]. Available: http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=65274
[4] F. Baker and G. Fairhurst, "IETF Recommendations Regarding Active Queue Management," RFC 7567 (Best Current Practice), Internet Engineering Task Force, Jul 2015. [Online]. Available: https://tools.ietf.org/html/rfc7567
[5] J. Gettys and K. Nichols, "Bufferbloat: Dark Buffers in the Internet," *Queue*, vol. 9, no. 11, pp. 40:40–40:54, Nov 2011.
[6] "Internet Uninterrupted: Australian Households of the Connected Future," A Telsyte research paper commissioned by nbn, Australia, Nov 2015. [Online]. Available: http://www.nbnco.com.au/content/dam/nbnco2/documents/Internet%20Uninterrupted%20Australian%20Households%20of%20the%20Connected%20Future.pdf
[7] V. Turner, "The internet of things: Getting ready to embrace its impact on the digital." International Data Corporation, 2016. [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=DR2016_GS4_VT
[8] Gartner, Inc. (2015) Gartner says 6.4 billion connected "things" will be in use in 2016, up 30 percent from 2015. [Online]. Available: http://www.gartner.com/newsroom/id/3165317
[9] D. Hassoun, "Tracing attentions," *Television & New Media*, vol. 15, no. 4, pp. 271–288, 2014. [Online]. Available: http://dx.doi.org/10.1177/1527476412468621
[10] H. J. L. . M. Andrejevic, "Second screen theory: From democratic surround to the digital enclosure." in *Connected Viewing: Selling, Streaming, & Sharing in the Digital Era*, J. Holt and K. Sanson, Eds. Routledge, 2014, pp. 40–61.
[11] E. Tussey, "Connected viewing on the second screen: The limitations of the living room," in *Connected Viewing: Selling, Streaming, & Sharing in the Digital Era*, J. Holt and K. Sanson, Eds. Routledge, 2014.
[12] R. Kenny and T. Broughton. (2014) Domestic bandwidth requirements in australia a forecast for the period 2013-2023. [Online]. Available: https://www.communications.gov.au/sites/g/files/net301/f/

Forecasting-Australian-Per-Household-Bandwidth-Demand-Commun.pdf
[13] Australian Bureau of Statistics. (2015) 3236.0 - household and family projections, australia, 2011 to 2036. [Online]. Available: http://www.abs.gov.au/ausstats/abs@.nsf/Latestproducts/3236.0Main%20Features42011%20to%202036
[14] M. Gregg, *Work's Intimacy*. Polity, 2011. [Online]. Available: http://www.polity.co.uk/book.asp?ref=9780745650272
[15] I. Ceaparu, J. Lazar, K. Bessiere, J. Robinson, and B. Shneiderman, "Determining causes and severity of end-user frustration," *International Journal of Human Computer Interaction*, vol. 17, no. 3, pp. 333–356, 2004. [Online]. Available: http://dx.doi.org/10.1207/s15327590ijhc1703_3
[16] J. Kennedy, R. Wilken, B. Nansen, M. Arnold, and M. Harrop, "Overcoming the tyranny of distance? high speed broadband and the significance of place," in *Making Publics, Making Places*, M. Griffiths and K. Barbour, Eds. University of Adelaide Press, Forthcoming.
[17] I. Arapakis, X. Bai, and B. B. Cambazoglu, "Impact of response latency on user behavior in web search," in *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, ser. SIGIR '14. New York, NY, USA: ACM, 2014, pp. 103–112. [Online]. Available: http://doi.acm.org/10.1145/2600428.2609627
[18] G. Armitage, M. Claypool, and P. Branch, *Networking and Online Games - Understanding and Engineering Multiplayer Internet Games*. UK: John Wiley & Sons, 2006.
[19] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet," RFC 2309 (Informational), Internet Engineering Task Force, Apr 1998. [Online]. Available: https://tools.ietf.org/html/rfc2309
[20] R. Pan, P. Natarajan, F. Baker, and G. White, "PIE: A Lightweight Control Scheme To Address the Bufferbloat Problem," IETF Draft, Sep 2016. [Online]. Available: https://tools.ietf.org/html/draft-ietf-aqm-pie-10
[21] K. Nichols, V. Jacobson, A. McGregor, and J. Iyengar, "Controlled Delay Active Queue Management," IETF Draft, Dec 2016. [Online]. Available: https://tools.ietf.org/html/draft-ietf-aqm-codel-06
[22] T. Hoeiland-Joergensen, P. McKenney, D. Taht, J. Gettys, and E. Dumazet, "FlowQueue-Codel," IETF Draft, Mar 2016. [Online]. Available: https://tools.ietf.org/html/draft-ietf-aqm-fq-codel-06
[23] T. Hoeiland-Joergensen, P. Hurtig, and A. Brunstrom, "The Good, the Bad and the WiFi: Modern AQMs in a residential setting," *Computer Networks*, vol. 89, pp. 90 – 106, 2015.
[24] R. Al-Saadi and G. Armitage, "Dummynet AQM v0.2 – CoDel, FQ-CoDel, PIE and FQ-PIE for FreeBSD's ipfw/dummynet framework," Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, Tech. Rep. 160418A, 18 April 2016. [Online]. Available: http://caia.swin.edu.au/reports/160418A/CAIA-TR-160418A.pdf
[25] J. Kua, G. Armitage, and P. Branch, "The Impact of Active Queue Management on DASH-based Content Delivery," in *41st Annual IEEE Conference on Local Computer Networks (LCN)*, Dubai, UAE, 7 - 10 November 2016. [Online]. Available: https://doi.org/10.1109/LCN.2016.24
[26] R. Collom, "The effects of an AQM enabled home gateway on First Person Shooter game traffic," Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, Tech. Rep. 160826A, 26 August 2016. [Online]. Available: http://caia.swin.edu.au/reports/160826A/CAIA-TR-160826A.pdf
[27] (2016) OpenWrt. [Online]. Available: https://openwrt.org
[28] R. Wilken, B. Nansen, M. Arnold, J. Kennedy, and M. Gibbs, "National, local and household media ecologies: The case of australia's national broadband network," *Communication, Politics & Culture*, vol. 46, pp. 136–154, 2013. [Online]. Available: http://mams.rmit.edu.au/bl8kdfli247i1.pdf
[29] B. Nansen, M. Arnold, M. Carter, R. Wilken, J. Kennedy, and M. Gibbs, "Proxy users, use by proxy: Mapping forms of intermediary interaction," in *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction*, ser. OzCHI '15. New York, NY, USA: ACM, 2015, pp. 294–298. [Online]. Available: http://doi.acm.org/10.1145/2838739.2838789
[30] B. Nansen, J. Kennedy, M. Arnold, M. Gibbs, and R. Wilken, "Digital ethnographic techniques in domestic spaces: Notes on methods and ethics," *Visual Methodologies*, vol. 3, no. 2, pp. 86–97, 2015. [Online]. Available: http://journals.sfu.ca/vm/index.php/vm/article/view/50
[31] J. Kennedy, B. Nansen, M. Arnold, R. Wilken, and M. Gibbs, "Digital housekeepers and domestic expertise in the networked home," *Convergence*, vol. 21, no. 4, pp. 408–422, 2015. [Online]. Available: http://dx.doi.org/10.1177/1354856515579848
[32] M. Gibbs, M. Arnold, B. Nansen, M. Harrop, and J. Kennedy, "An ontography of broadband on a domestic scale," *Transformations Journal*, vol. 27, pp. 1–12, 2016. [Online]. Available: http://www.transformationsjournal.org/issues/27/07.shtml
[33] S. Cumaranayagam, "Queue management in Home Gateways: Potential impact on IoT traffic flows," Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, Tech. Rep. 160216A, 16 February 2016. [Online]. Available: http://caia.swin.edu.au/reports/160216A/CAIA-TR-160216A.pdf
[34] S. Cumaranayagam and D. Mirkovic, "Impact of insufficient TCP receive window on bulk data transfers in home broadband environments," Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, Tech. Rep. 160923A, 23 September 2016. [Online]. Available: http://caia.swin.edu.au/reports/160923A/CAIA-TR-160923A.pdf
[35] 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/150210A/CAIA-TR-150210A.pdf
[36] ——, "CAIA Testbed for TEACUP Experiments Version 2," Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, Tech. Rep. 150210C, 10 Feb 2015. [Online]. Available: http://caia.swin.edu.au/reports/150210C/CAIA-TR-150210C.pdf
[37] R. Pan, P. Natarajan, C. Piglione, M. Prabhu, V. Subramanian, F. Baker, and B. VerSteeg, "PIE: A Lightweight Control Scheme To Address the Bufferbloat Problem," IETF Draft, draft-pan-aqm-pie-00, June 2013. [Online]. Available: https://tools.ietf.org/html/draft-pan-aqm-pie-00
[38] J. Kua and G. Armitage, "Generating Dynamic Adaptive Streaming over HTTP Traffic Flows with TEACUP Testbed," Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, Tech. Rep. 161216A, 16 December 2016. [Online]. Available: http://caia.swin.edu.au/reports/161216A/CAIA-TR-161216A.pdf
[39] S. Zander and G. Armitage, "Minimally-Intrusive Frequent Round Trip Time Measurements Using Synthetic Packet Pairs," in *The 38th IEEE Conference on Local Computer Networks (LCN 2013)*. IEEE, Oct 2013, pp. 264 – 267.
[40] S. Kim, J.-Y. Hong, S. Kim, S.-H. Kim, J.-H. Kim, and J. Chun, "Restful Design and Implementation of Smart Appliances for Smart Home," in *The 11th IEEE Intl Conf on Ubiquitous Intelligence and Computing*, Dec 2014, pp. 717–722.
[41] C. Pereira and A. Aguiar, "Towards Efficient Mobile M2M Communications: Survey and Open Challenges," *Sensors*, vol. 14, no. 10, p. 19582, 2014.
[42] P. Nanda and R. C. Fernandes, "Quality of Service in Telemedicine," in *The 1st International Conference on the Digital Society (ICDS 2007)*, Jan 2007.
[43] E. Kartsakli, A. S. Lalos, A. Antonopoulos, S. Tennina, M. D. Renzo, L. Alonso, and C. Verikoukis, "A survey on M2M systems for mHealth: a wireless communications perspective," *Sensors (Basel, Switzerland)*, vol. 14, no. 10, pp. 18 009–18 052, Jan 2014.
[44] L. Skorin-Kapov and M. Matijasevic, "Analysis of QoS Requirements for e-Health Services and Mapping to Evolved Packet System QoS Classes," *Int. J. Telemedicine Appl.*, vol. 2010, pp. 9:1–9:18, Jan 2010.
[45] B. Latré, B. Braem, I. Moerman, C. Blondia, and P. Demeester, "A survey on wireless body area networks," *Wireless Networks*, vol. 17, no. 1, pp. 1–18, Nov 2010.
[46] M. Shu, D. Yuan, C. Zhang, Y. Wang, and C. Chen, "A MAC protocol for medical monitoring applications of wireless body area networks." *Sensors (Basel, Switzerland)*, vol. 15, no. 6, pp. 12 906–31, jan 2015.
[47] M. Patel and J. Wang, "Applications, challenges, and prospective in emerging body area networking technologies," *IEEE Wireless Communications*, vol. 17, no. 1, pp. 80–88, Feb 2010.
[48] D. Vouyioukas, I. Maglogiannis, and D. Komnakos, "Emergency m-Health Services through High-Speed 3G Systems: Simulation and Performance Evaluation," *SIMULATION*, vol. 83, no. 4, pp. 329–345, apr 2007.

[49] J. Misic and V. Misic, “Bridging between IEEE 802.15.4 and IEEE 802.11b networks for multiparameter healthcare sensing,” *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 4, pp. 435–449, May 2009.
[50] D. Varma, “Managing DICOM images: Tips and tricks for the radiologist,” *Indian Journal of Radiology and Imaging*, vol. 22, no. 1, p. 4, jan 2012.
[51] 11 Telemedicine Tools Transforming Healthcare. [Online]. Available: http://www.informationweek.com/mobile/11-telemedicine-tools-transforming-healthcare/d/d-id/1103487?page_number=9
[52] C. Bormann, K. Hartke, and Z. Shelby, “The Constrained Application Protocol (CoAP),” RFC 7252, Jun. 2014. [Online]. Available: https://tools.ietf.org/html/rfc7252
[53] A. Stanford-Clark and H. L. Truong. MQTT For Sensor Networks (MQTT-SN) Protocol Specification Version 1.2. [Online]. Available: http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf
[54] T. D. P. Mendes, R. Godina, E. M. G. Rodrigues, J. C. O. Matias, and J. P. S. Catalao, “Smart Home Communication Technologies and Applications: Wireless Protocol Assessment for Home Area Network Resources,” *Energies*, vol. 8, no. 7, pp. 7279–7311, 2015. [Online]. Available: http://www.mdpi.com/1996-1073/8/7/7279
[55] “NVR Camera Capacity based on Bit Rate,” Hikvision, Aug 2012. [Online]. Available: http://oversea-download.hikvision.com/uploadfile/doc/NVRCameraCapacitybasedonBitRate.pdf
[56] J. Nagle, “Congestion Control in IP/TCP Internetworks,” Legacy RFC 896, Jan 1984. [Online]. Available: http://www.ietf.org/rfc/rfc896.txt