European Commission — FP7

R/TE

REDUCING INTERNET TRANSPORT LATENCY

Project acronym: **RITE**

Project number: 317700

Work package: Use-case trials

Deliverable number and name:

Deliverable 3.3: Deployment of RITE mechanisms in use-case trial testbeds report

Title: Deployment of RITE mechanisms in use-case trial testbeds report
Work Package: WP3

Version: 1
Date: January 25, 2016
Pages: 55

Author:
Ing-Jyh Tsang

Co-Author(s):
Mohammad Rajiullah, Gorry Fairhurst, Andreas Petlund, Raffaello Secchi, Bob Briscoe, Achilles Petras, Koen De Schepper, Olga Bondarenko

To:
Jorge Carvalho
Project Officer

Status:
[ ] Draft
[ ] To be reviewed
[ ] Proposal
[X] Final / Released to CEC

Confidentiality:
[X] PU — Public
[ ] PP — Restricted to other programme participants
[ ] RE — Restricted to a group
[ ] CO — Confidential

Revision:
(Dates, Reviewers, Comments)

Contents:
This report summarises the effect of the developed mechanisms in the use-cases trials in the industry partner testbeds. It constitute as a final report for the working done by WP3 tasks 3.3, 3.4 and 3.5. Three testbeds were designed and setup to evaluate the efficiency and deployability of selected mechanisms studied and developed in WP1 and WP2. This document is divided along the three use-cases i.e. online game, web application and interactive video. In addition, we include a session describing an emulated testbed used to evaluating ABE with Dual-Queue, which combined end-system and network mechanics in a single testbed.

Abbreviations

This section provides definitions of key terms and defines the abbreviations used in the remainder of the report.

3G 3rd Generation

3GPP 3rd Generation Partnership Project

AccECN Accurate ECN

ACK Acknowledgement

ADU Application Data Unit

API Application Programming Interface

AQM Active Queue Management

ARED Adaptive Random Early Drop

BE Best Effort

BCP Best Current Practice

BDP Bandwidth Delay Product

BoF Birds-of-a-Feather

CAIDA Cooperative Association for Internet Data Analysis

CC Congestion Control

CDF Cumulative Density Function

CDN Content Delivery Network

CE Congestion Experienced

CM Congestion Manager

CMT-SCTP Concurrent Multipath Transfer for SCTP

CoDel Controlled Delay

CUBIC Cubic function congestion control

cwnd Congestion WiNDow

CWR Congestion Window Reduced

CWV Congestion Window Validation

DASH Dynamic Adaptive Streaming over HTTP

DC Data Center

DCLC Data Center Latency Control (IRTF research group)

DCTCP Data Center TCP

DiffServ Differentiated Services

DNS Domain Name System

DOCSIS Data Over Cable Service Interface Specification

DoS Denial of Service

DRR Deficit Round Robin

DSCP Differentiated Services Code Point

DSL Digital Subscriber Line

E-UTRAN Evolved Universal Terrestrial Radio Access Network

E2E End-to-End

ECE ECN-Echo

ECN Explicit Congestion Notification

ECT ECN-Capable Transport

eNB eNodeB, Evolved Node B (LTE Base-Station)

FIFO First-In First-Out

FQ Fair queuing

FQ-CoDel Flow queuing CoDel

FSE Flow State Exchange

FTP File Transfer Protocol

GB Gigabyte

GPRS) General Packet Radio Service

GRE) Generic Routing Encapsulation

GTP) GPRS Tunnelling Protocol

GUTS Getting up to Speed

HAS HTTP Adaptive Streaming

HSS Hybrid Slow-Start

HTTP HyperText Transfer Protocol

HULL High-bandwidth Ultra Low Latency

ICMP Internet Control Message Protocol

IETF Internet Engineering Task Force

IP Internet Protocol

IRTF Internet Research Task Force

ISP Internet Service Provider

IW Initial Window

LAN Local Area Network

LEDBAT Low Extra Delay BAckground Transport

LKM Loadable Kernel Module

LTE Long Term Evolution

MAC Media Access Control

MB Megabyte

MPLS Multi Protocol Label Switching

MPTCP Multipath TCP

MSS Maximum Segment Size

NAT Network Address Translation

NS-2 Network Simulator 2

NV-GRE Network Virtualization using Generic Routing Encapsulation

NVO Network Virtualization Overlay

NVP Non Validated Period

OWD One Way Delay

PCI Packet Congestion Indication

PCN Pre-Congestion Notification

PDI Packet Drop Indication

PDPC Packet Discard Prevention Counter

PDV Packet Delay Variation

PGW Packet Gateway

PIE Proportional Integral controller Enhanced

PLT Page Load Time

PSP PipeACK Sampling Period

QV Queue View

RCP Rate Control Protocol

RED Random Early Drop

RFC Request For Comments

RITE Reducing Internet Transport Latency End-to-End

RMCAT RTP Media Congestion Avoidance Techniques (IETF Working Group)

RT Real-Time

RTCWEB Real Time Collaboration on world wide WEB (IETF Working Group)

RTO Retransmission Time Out

RTP Real-Time Protocol

RTT Round Trip Time

RW Restart Window

SACK Selective Acknowledgement

SAE System Architecture Evolution

SBD Shared Bottleneck Detection

SCTP Stream Control Transmission Protocol

SDO standards Development Organisation

SFQ Stochastic Fair Queuing

SGW Serving Gateway

SKB Socket Buffer

SQ Source Quench

SVD Singular Value Decomposition

SYN Synchronize packet

TFRC TCP Friendly Rate Control

TCP Transmission Control Protocol

TCP′ Paced and RTT Independent TCP (TCP-PRIme)

TCPM TCP Maintenance (IETF Working Group)

TOS Type Of Service

TRILL TRansparent Interconnection of Lots of Links

TSQ TCP Small Queue

UDP User Datagram Protocol

URG TCP Urgent flag

VCP Variable-structure congestion Control Protocol

VXLAN Virtual Extensible LAN

WiFi Wireless Fidelity

WG Working Group

WP Work Package

WRED Weighted Random Early Drop

Participant organisation name	**Participant Short Name**	**Country**
Simula Research Laboratory	SRL	Norway
BT	BT	UK
Alcatel-Lucent	ALU	Belgium
University of Oslo	UiO	Norway
Karlstad University	KaU	Sweden
Institut Mines-Télécom	IMT	France
University of Aberdeen	UoA	UK
Megapop	MEGA	Norway

1 Introduction

This report evaluates some of the mechanisms that RITE has developed through three testbeds. Each testbed is focussed specifically on one of the industry partner's use cases. The testbeds are notable for using production equipment, in which the overall transport latency have been enhanced with some of the mechanisms that RITE has developed. The three use cases are: online gaming, web / broadband applications and interactive video. The report also describes results from an emulation of one of the testbeds, which was used to evaluate other mechanisms that RITE has developed. The document summarises the work of WP3 tasks 3.3, 3.5 and 3.7 to evaluate the efficiency and deployability of various mechanisms developed in WP1 and WP2.

This deliverable follows the algorithms and analysis presented in deliverable *D1.3 - Report on Prototype Development and Evaluation of End-System, Application Layer- and API Mechanisms* [1] and *D2.3 Report on Prototype Development and Evaluation of Network and Interaction Techniques* [2]. The gaming testbed was only used to evaluate RITE end system mechanisms (described in D1.3). This was because a gaming provider (in our case Megapop) controls both the server and clients, but uses cloud-based servers to provide the network infrastructure, hence it was not be possible to deploy solutions that relied on network support. The second and third testbeds included operator equipment which it was possible to update with RITE mechanisms, and hence these testbeds were used to introduce both end system (D1.3) and network mechanisms described in D2.3.

1.1 Testbed 1: Megapop game environment

The first use case trial considered online gaming within the Megapop game environment. This testbed was based on the actual Megapop production environment. The Megapop game environment offered a unique opportunity to test end system solutions RITE mechanisms with real gaming traffic patterns. Initial tests considered the traffic of the Troll vs Vikings game application. This was followed by detailed evaluation of traffic representing a next generation real-time online game application being developed by Megapop, where latency is critical. Tests used a low latency update to the Linux kernel developed by the RITE project, which included RITE end-system mechanisms developed in WP1. This update was installed on the testbed servers and client to evaluate the expected impact when used with the real-time game prototype.

The key highlights for evaluation of end system mechanisms with online gaming can be summarised as:

- The testbed is based on the actual deployed infrastructure.
- The testbed end systems introduced WP1 (D1.3) RITE mechanisms at the server and clients e.g., RTOR [3], TLPR [4] , new-CWV [5], RDB [6] and ABE [7].
- The testbed traffic used the Trolls vs Vikings game, with clients developed specially for the testbed. These clients were able to play indefinitely without any human assistance. Since Trolls vs. Vikings is a turn-based game with some real-time elements, Megapop developed a real-time prototype model in order to get a realistic evaluation of real-time game patterns.
- Megapop games were developed for mobile devices, thus the RITE mechanisms to reduce latency were tested over a wireless network segment.

1.2 Testbed 2: Web application low-latency testbed

The second testbed, a joint ALU/BT testbed, was developed for evaluation of jointly optimised network and end system mechanisms developed in WP2 within an operator network environment. This testbed was used because of BT's experience on service delivery network for residential and small to medium-sized business customers. In the downstream direction (towards the customer), the network is deliberately designed to bottleneck at the last IP-aware hop; known as the broadband network gateway (BNG) function, also sometimes known as a Broadband Remote Access Server (BRAS). In the upstream

direction, the network is designed to bottleneck at the home hub, which is the last IP-aware hop before the DSL access link (sometimes called the home gateway). The set of trials used a web traffic workload to evaluate RITE low latency mechanisms. Much of the current network design and management complexity comes from the requirement of an architecture to share capacity, exploiting multiplexing efficiency, and still be able to provide each customer with performance assurances. This testbed was therefore also used to explore how the new methods could improve the deployment and self-management (i.e. auto-tuning) of networked services with different QoS/latency requirements.

The key highlights for the testbed activity for web application using Alcatel-Lucent low-latency testbed can be summarised as:

- The testbed was based on actual deployed network operator infrastructure. It incorporated WP2 RITE mechanisms to provide low latency using DCTCP and Dual Queue (DualQ) active queue management (AQM). This was able to transport sustainably higher video quality under the same congested traffic conditions.
- Web traffic was considered focusing on video delivered over-the-top using HTTP Adaptive Steaming (HAS) (i.e., Microsoft Smooth Streaming [8, 9]).
- This testbed was also used to evaluate how the new methods improved QoS and traffic management capabilities.

1.3 Testbed 3: Alcatel-Lucent interactive video testbed

The third testbed was used to evaluate an interactive video application with a novel Couple Dual Queue AQM system developed in WP2. The ALU testbed implements the full end-to-end path: data centre servers, the core network, backhaul network, access link, and a home network. The evaluation used RITE's AQM mechanisms implemented in a Linux machine, with break-out connectivity from ALU's 7750 multi-service edge (MSE) node. Since the end-to-end requirement for ultra low latency could only be achieved by minimising delays at all layers, this case also involved optimisation of the interactive video application system. The goal was to build capabilities to deploy ultra low latency network necessary for innovative services, such as cloud-based interactive video and networked virtual reality services.

A panoramic interactive video application was integrated in this testbed. This application was based on a system developed by FP7 FascinatE project, which originally assumed an unconstrained/uncongested network. The application was adapted/recoded to give the best experience under a congested network. Two different end user devices were integrated into the testbed, a touch screen laptop/tablet where users could zoom, pan and tilt a panoramic video image, and the Oculus Rift headset, showing an interactive virtual reality video application.

The key highlights for the testbed for interactive video in an Alcatel-Lucent testbed can be summarised as:

- The testbed was built and integrated in a setup mirroring an operator's residential service network. It was composed of RGWs, xDSLs, BNGs and service routers equipment deployed by Alcatel-lucent customers. The DualQ AQM [10, 11] developed in WP2 was implemented in a server and integrated in the testbed connected to the BNG equipment.
- Two applications were tested: a panoramic interactive video application and an interactive virtual reality video application.
- Extensive tests evaluated the ability of the RITE mechanisms to support delivery of a low latency interactive video service, including simulation of long-standing and dynamic traffic flows to understand the advantages and limitations of the DualQ AQM system.

1.4 Value of testbeds in the project

Together the RITE testbeds provided experimental data to evaluate the efficiency of the methods for a range of applications. A standalone version of the ALU testbed was prepared as an additional technology demonstrator. This provided a portable emulation of the system, and has assisted in project dissemination. The interactive video use-case was demonstrated to attendees at the IETF-93 meeting in the Bits-N-Bites session. It was also deployed in a virtualised environment at the University of Aberdeen to evaluate the updated RITE end system mechanisms in an environment that included the AQM algorithms from the ALU testbed.

Together the set of testbed trials provided many opportunities to learn from deployment of the mechanisms. The experience in selecting best practice parameters for the developed mechanisms provides an input to the definition of a set of guidelines describing how to tune mechanisms to minimise latency, reported in a deliverable *D3.4 - RITE Recommended Use Parameters Report*. Experience using the mechanisms within the testbeds provided the necessary understanding of opportunities for exploitation of the project outputs.

1.5 Structure of the report

The remaining of this report is divided in four section:

- Deployment and evaluation of end system mechanisms for online gaming in a Megapop game environment.
- Deployment and evaluation of RITE mechanisms for web application using Alcatel-Lucent low-latency testbed.
- Deployment and evaluation of RITE mechanisms for interactive video in an Alcatel-Lucent testbed.
- Deployment and evaluation of RITE mechanisms over UoA emulated testbed

The first three testbeds follow the Description of Work. Each of these testbeds focuses on their use case application, i.e. online gaming, web applications and interactive video. The fourth testbed was created as a laboratory testbed developed at UoA based on virtual machines (VMs) and used to carry out integration tests for mechanisms developed by both WP1 and WP2. The work carried out in this testbed, combining mechanisms tested on the online game application and the network mechanisms used in the web application and interactive video application, stood apart from the work of the other testbeds and thus is reported in a separate section.

Each section reports on the deployment and evaluation of mechanisms relevant to its use case. It describes how the testbed is composed, including the RITE mechanism it includes used by each of the use case. It reports the results, in particular the effectiveness of reducing latency, and discusses the benefits and possible drawbacks.

2 Deployment and evaluation of end system mechanisms for online gaming in a Megapop game environment

This section describes tests performed on the Megapop online game testbed to evaluate the selected RITE mechanisms for this scenario. The mechanisms were deployed in the game server using the custom patched Linux kernel described in deliverable 1.3. First, tests performed using Megapop's Trolls vs Viking game are described. Since the physics-based real-time game has unfortunately not yet reached prototype stage, we have modelled the expected traffic behaviour for this real-time game using specifications provided by Megapop . This emulated game server has been evaluated in the Megapop testbed and the results are presented in this document.

2.1 End System Mechanisms

In this section, we briefly describe the end system mechanisms that have been developed in RITE and chosen for testing on the Megapop testbed. A more detailed description of these mechanisms can be found in D1.3 or in the referenced publications.

RTO restart (RTOR) [3]
RTOR will reduce the time needed to retransmit when tail loss happens. In this case, loss happens at the tail of a flow or segment bursts within a connection. In TCP, the retransmission timer is restarted on the reception of an ACK that acknowledges new data while outstanding data are still available. A retransmission therefore occurs after RTO seconds from when the ACK was received, not RTO seconds after the transmission of the potentially lost segment(s). In most cases, total loss recovery time is exceeded by an extra delay of approximately one RTT. The delay could be even higher if the ACK that restarts the timer is a delayed ACK. In RTOR, retransmission timer is restarted with respect to the last transmitted segment, thus avoiding these extra delays.

TLP with Restart (TLPR) [12]
Using Tail Loss Probe (TLP) [4], a probe segment is sent after a Probe Time Out (PTO) when an ACK is overdue for a connection. The probe can induce extra selective ACKs or SACKs at the receiver when tail loss happens. The probe is generally sent PTO seconds after the ACK that restarts the timer. On the other hand, using TLP with Restart (TLPR), the logic of RTOR is applied in TLP, which causes the probe to be sent PTO seconds after the transmission of the last segment.

Redundant Data Bundling (RDB) [6, 13]
In interactive, time dependent flows, data transmission are often triggered by real life events, resulting in small data chunks with relatively high interarrival times. We call such flows "thin streams" [14]. For online games, such flows are often isochronous, meaning that data is sent periodically with the same time interval between, triggered by game server "ticks". RDB tries to redundantly bundle unacknowledged data with new data chunks as long as the resulting segment size is smaller than one Maximum Segment Size (MSS). By proactively retransmitting unacknowledged data, RDB potentially hides losses while never sending more packets than the application would otherwise have triggered. RDB will reduce latency for flows matching the target patterns by avoiding retransmissions by timeout and eliminating head-of-line blocking at the receiver.

Congestion Window Validation(new-CWV) [5]
Standard TCP is not efficient at handling application limited traffic. When an application is idle for a period larger than one RTO, the *cwnd* is reset to no more than the Restart Window (RW) [15]. However, during the application limited period, standard TCP allows the *cwnd* to grow arbitrarily large. In such a scenario, since packets are sent at a lower rate than permitted by the *cwnd*, the reception of an ACK does not show that the path can sustain the transmission rate reflected by the *cwnd*. A sudden rate increase by the application can cause severe congestion and increase latency for the concurrent traffic in the network. Although resetting the *cwnd* after an idle period is a safe course of action, it reduces performance substantially for bursty applications that needs to quickly

get back up to speed after idle time. new-CWV proposes an alternative to address this problem by conditionally freezing the congestion window during the rate limited period.

Alternative Backoff with ECN (ABE) [7]
ABE is based on the hypothesis that in the near future, all ECN enabled devices in the network will be equipped with state-of-the-art AQMs such as CoDel (or FQ-CoDel and its variants) or PIE that aim to maintain the queuing delay in the range of 5 to 20 ms by default with short term packet burst allowance. In such a scenario, the reception of an ECN mark indicates the low-delay marking thresholds of the AQMs and therefore the corresponding TCP sender may afford to reduce its sending rate by less as compared to halving the send rate in case of packet losses (three DupACKs). ABE leverages this potential and shows significant throughput gains without losing the delay-reduction benefits of CoDel or PIE [7].

2.2 Trolls vs Vikings

Megapop, an Oslo based game developer and RITE partner, developed the game "Trolls vs Vikings". This is a "tower defense" type casual game, designed to attract a large amount of players, with a "free to play" business model. The game is single-player, but has collaborative elements like rankings and social media updates being made in-between matches and at significant in-game events. The production game server is hosted at Microsoft Azure cloud and the game client is available for both Android and iOS. For RITE, Megapop has set up an experimental test server in Azure that mirrors the actual production deployment. The test server is used both for the game-company's own development testing and for RITE experiments into network traffic latency.

2.2.1 Experiment Setup

The Trolls vs Vikings game server is deployed with Linux 3.18.5. The kernel was patched with all the RITE mechanisms discussed in the above section. At the Karlstad university (KaU) side, the other part of the testbed, we have set up a Linux machine and enabled its wireless NIC into wifi mode. The NIC is of 802.11g type. We used 15 android devices that hosted Trolls vs Vikings game clients. These clients are a modified version of the original one. Megapop modified the client to be able to play indefinitely without any human assistance. The Azure cloud data centre hosting the Trolls vs Vikings server is located in Amsterdam. The measured RTT between KaU and the server is approximately 30 ms. Since, most of the RITE mechanisms are triggered when loss occurs, we set up 1.5% artificial loss at the wifi point using the Linux traffic control (tc) with network emulator (netem) [16]. We used tcpdump[1] to collect traces both at the server side and at the client side. We later processed the traces to calculate retransmission delays, packet delays used as the metrics to compare the effectiveness of the RITE mechanisms in the given scenario. The test setup is shown in Figure 2.2.

2.2.2 Traffic Characteristics

Before running experiments on RITE mechanisms, we collected traces at the Megapop server to assess the traffic patterns and make an initial judgement on the potential for mechanism effectiveness. Some of the basic statistics of the traffic pattern from the traces are shown in Figure 2.1. These statistics are based on flows that have more than two packets. Otherwise more than half of the flows are between one and two packets in size. According to the packet count sub-figure (x–axis in logscale) in Figure 2.1, still, more than 60% of the flows contain no more than 10 packets.

Furthermore, according to the avg packet len sub-figure in Figure 2.1, for most of the flows, the average packet lengths do not exceed 500B. Continuing withe the same figure, according to the average Inter-Arrival Times (IAT) subfigure, only less than 1% flows appear to have back to back packets with IAT up to 10 ms. Moreover, the throughput subfigure shows that for most of the flows, up to 60% according to the figure, the maximum throughput is only as much as 1 kBps. In general, most of the Trolls vs

[1] http://www.tcpdump.org

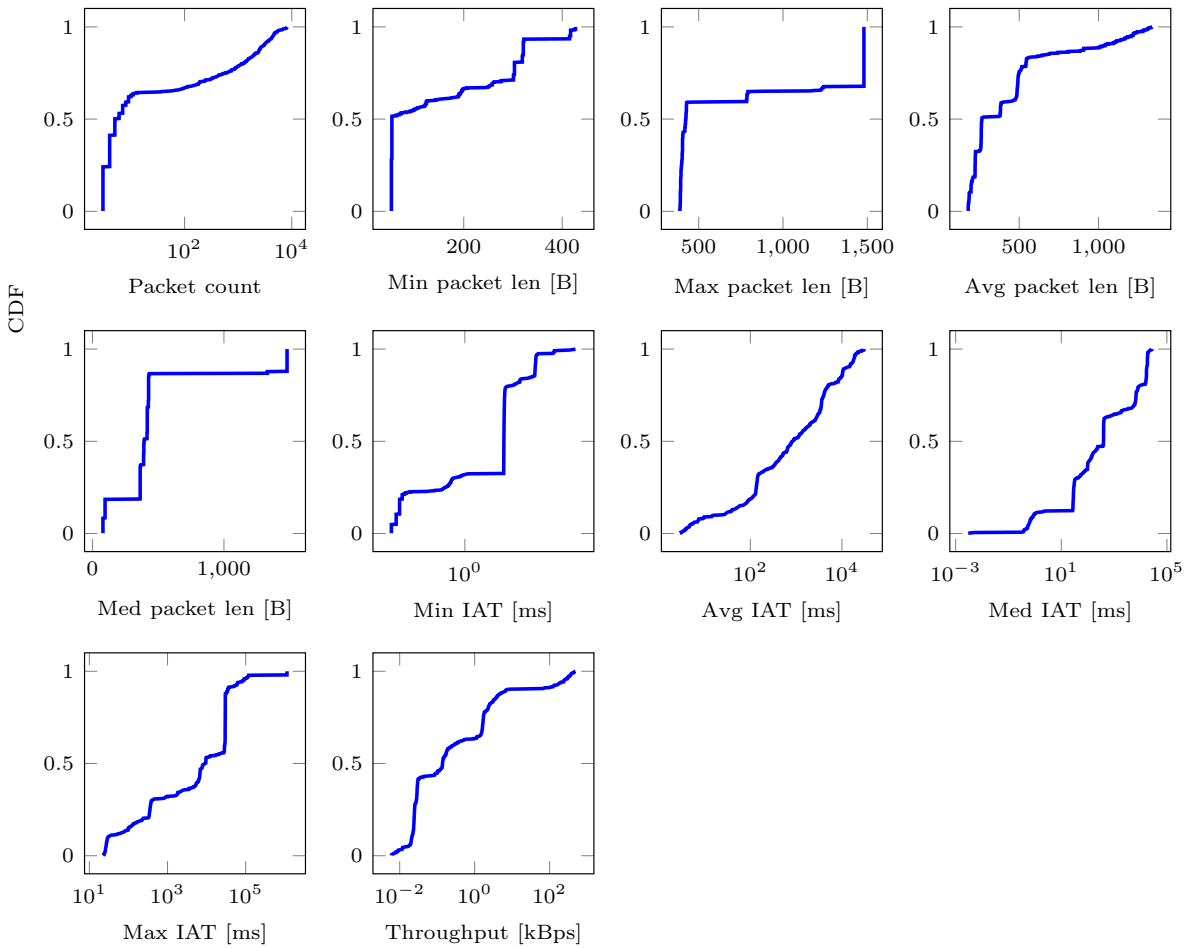


Figure 2.1: Traffic characteristics of Troll vs Vikings flows that are larger than two packets.

Vikings flows seemed to be thin streams or short flows with small packets and fairly large inter-arrival times. Due to fairly low intensity of Trolls vs Vikings game traffic, evolution of the congestion window is therefore not a concern.

Moreover, although we have seen a significant rise of interest for ECN in the networking community over the recent years, the measurement study in [17] found only a single ECN-enabled router in the Internet. However, mechanisms like ABE require ECN support in an end-to-end basis, which we can not ensure at the moment in our experiment involving real network. We therefore do not consider RITE mechanisms like ABE and new-CWV in our experiments.

2.2.3 Results

Since, most of the RITE mechanisms under test were designed to reduce loss recovery time, we calculate retransmission delays for each flow when loss occurs. Our first set of results showed very high values for retransmission delays for all cases. Our trace analysis revealed that we could detect no activations of the RDB mechanism, contrary to what we should expect. Moreover, we found that early retransmit did not work when TLP was turned on, which is a default setting in the linux 3.18.5 kernel. The TLPR mechanism was therefore inactive, as it depends on TLP. Later, our investigation revealed that Nagle's algorithm was not turned off on the game server, which is a necessary prerequisite for RDB to work as expected. We also found that when Nagle's algorithm is turned off, TLP behaves properly. Figure 2.3 shows the effect of Nagle's algorithm on the baseline TCP (no RITE mechanisms). In this case, TLP is turned on by default in baseline TCP which works properly in the case when Nagle's algorithm is turned

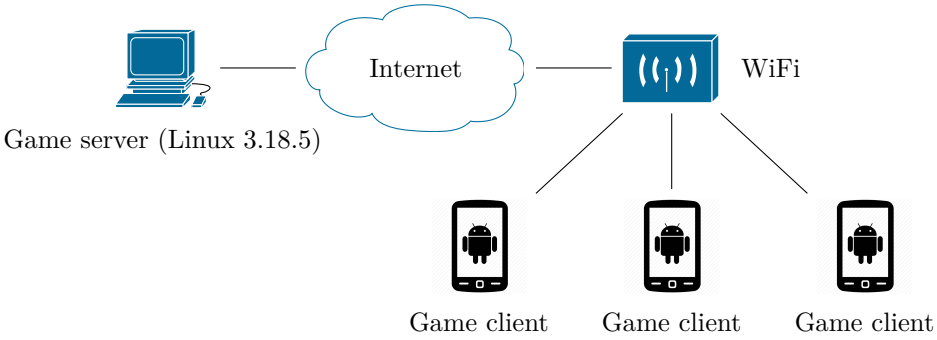


Figure 2.2: Trolls vs Vikings experiment setup

off.

After fixing the issue with the Nagle algorithm, we repeated the gaming experiment. The results in Figure 2.4 show the effectiveness of the tested RITE end system mechanisms for reduceing the retransmission delay when a loss occurs. We compared all the RITE mechanisms against the baseline TCP. We only show the results for the 99th percentile. Since, the loss at the network is very low ($\approx 1.5\%$), the difference in packet level delays is only visible at the upper tail of the Cumulative Distribution Functions (CDFs). The difference in this case between baseline and the RITE mechanisms is very small, because most of the RITE mechanisms like RTOR, TLPR did not come into effect for the loss recovery of the Trolls vs Vikings flows that mostly carry very few packets and large interarrival time. Figure 2.5 shows the comparison of retransmission delays between baseline TCP and RITE mechanisms, TLPR and RTOR. There is essentially no difference among the mechanisms in terms of retransmission delays. Our trace analysis suggested that the difference in Figure 2.4 is mainly due to proactive retransmissions of RDB. RDB helped to reduce the number of retransmissions and thus head-of-line blocking delay at the receiver. The retransmission counts are given in Table 2.1. RDB can help to avoid approximately half of the retransmissions, but the traffic pattern for this game does not allow for the full benefit of RDB latency reduction.

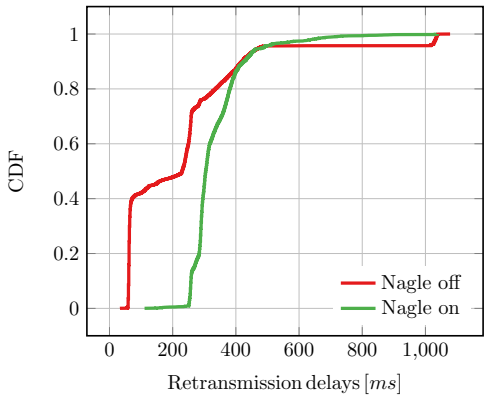


Figure 2.3: Impact of nagle algorithm on baseline TCP for retransmission delays.

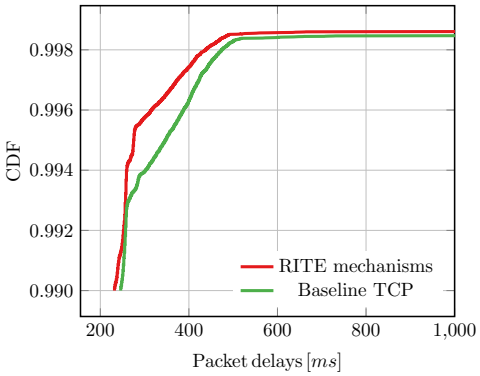


Figure 2.4: Comparison of packet delays between baseline TCP and RITE end system mechanisms.

2.3 Discussion

In both games, RDB provides the best networking latency by proactively bundling redundant data with the new transmission and thus preempt the experience of loss. Megapop can use the RDB mechanism to provide better gaming experiments for its user bases, but has to consider the bandwidth cost.

Moreover, from our experience, turning off Nagle's algorithm turned out to be crucial for the loss recovery

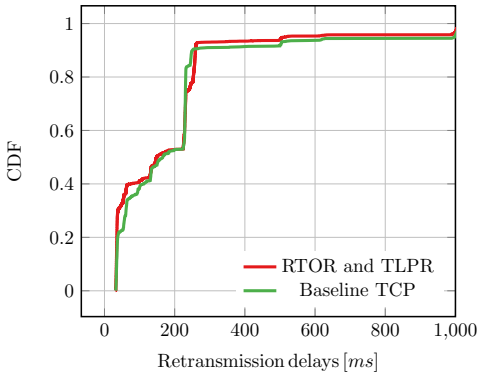
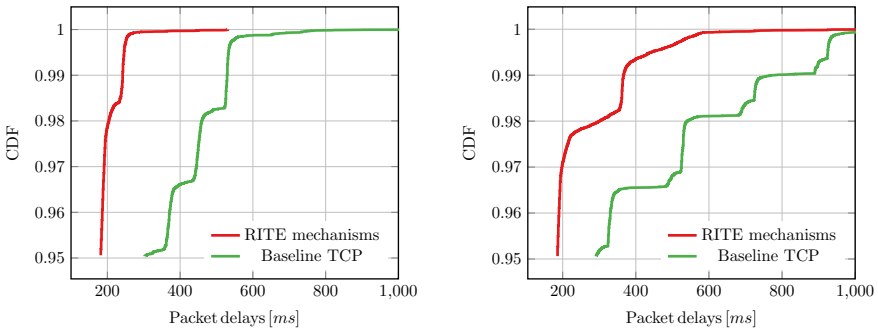


Figure 2.5: Comparison of retransmission delays between baseline TCP and RITE end system mechanisms, RTOR and TLPR.

Retransmission count	RITE mechanisms	Baseline TCP
1	2760	5584
2	135	194
3	0	3
4	0	1
Sum	2895	5782

Table 2.1: Comparison of number of retransmissions using different loss recovery mechanisms.



(a) IFT: 80 ms

(b) IFT: 200 ms

Figure 2.6: Packet delays for different IFTs with base RTT 165 ms.

mechanisms like TLP(R) and RDB. We will use this information when bringing the mechanisms to the Linux kernel mailing list. RDB should implicitly turn off Nagle's algorithm when it is active. Since TLP is on-by-default in the Linux kernel, as is Nagle's algorithm, we will document the effect and bring a suggestion for solution to the Linux kernel mailing lists in order to allow TLP to have its full potential for the scenarios where we documented the problems.

2.4 Conclusion

We have deployed the RITE mechanisms on a test server used by Megapop for internal development and testing. The applicable mechanisms were evaluated for the game "Trolls vs. Vikings" and for an emulated game server developed based on the specifications for a new realtime game under development by Megapop. The results show that the mechanisms that show effect are RTOR, TLPR and RDB. Due to the nature of the traffic patterns, the effect is good, but limited for "Trolls vs. Vikings". Most of the effect is due to the RDB mechanism. The real-time prototype shows a greater impact for the RITE mechanisms due to the smaller IFT in combination with small packet sizes. The experiments document the effect of the RITE mechanisms in the gaming scenario. The real-life deployment has also helped us uncover interactions between the Linux kernel mechanisms that should be taken into consideration when deploying online games with such traffic patterns.

3 Deployment and evaluation of RITE mechanisms for web application using Alcatel-Lucent low-latency testbed

3.1 BT's Objectives

BT provides broadband connectivity between homes with broadband and multiple CPs (Communication Providers). Apart from managing the capacity of the network end-to-end, BT offers different Classes of Service that in turn give CPs (including BT's own CP) the option to differentiate services in terms of performance. This differentiation is done by using a DiffServ [18] Quality of Service (QoS) model at BT's broadband network gateway (BNG). Broadly this is fit for the purpose but there is always the challenge that new services introduced by BT as well as its wholesale customers (i.e. CPs) [18] would encourage product development stakeholders to demand a new Class of Service that is more important than the others. That makes it hard to rationalise the relative importance of services and their mapping to a Diffserv queue at the BNG scheduler.

The use case is about serving HTTP Adaptive Streaming (HAS) in the presence of background TCP flows, when the downstream traffic per line is sufficient to saturate the queue at the BNG. Current TCPs deliver variable throughput which forces the HAS rate-determining algorithms to change the encoding rate frequently so deteriorating the end user s experience. Our work involves two aspects:

- At the end systems deploying a new TCP variant, DCTCP, which reacts "scalably" and more gently to indications of congestion.
- At the BNG (the bottleneck router) deploying a new queueing mechanism, DualQ, which provides an earlier and stronger signal of incipient congestion to the new TCP variant, whilst also ensuring that classic TCP traffic is still served fairly and in particular is not starved.

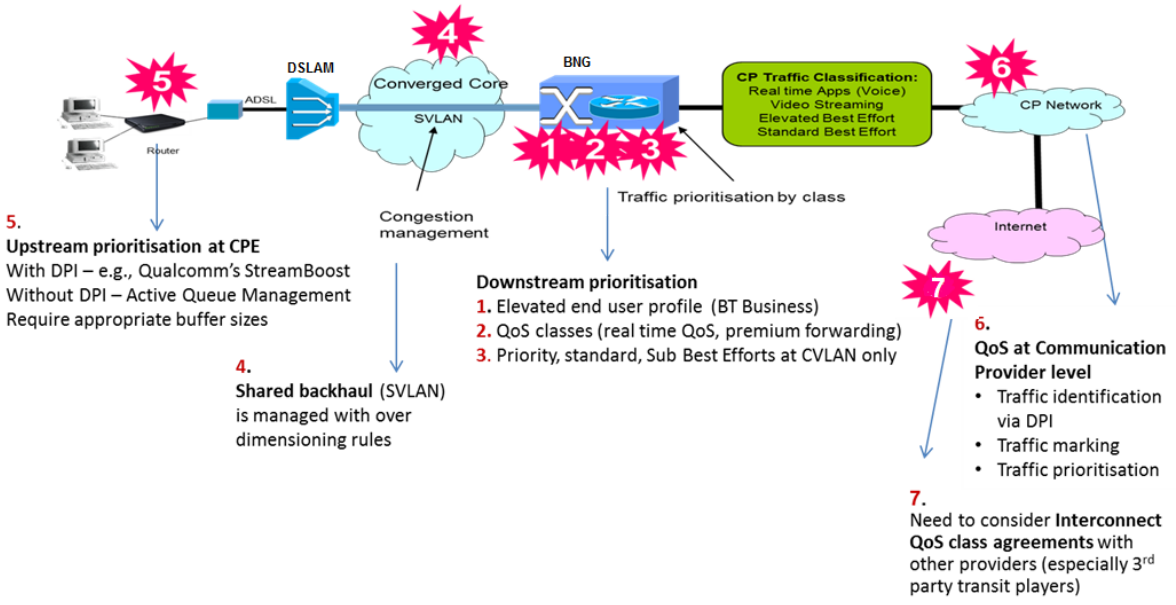


Figure 3.1: BT Network

3.1.1 Background

The scope of the RITE testbed is to demonstrate that the number of queues can be reduced by introducing an AQM to maintain low latency and best sharing of throughput to all value-added services (VAS) and

Over-The-Top (OTT) services. Normally, over-provisioning the capacity of the network minimises the need for QoS. However, bottlenecks like the shaper serving a line at the BNG could be saturated for short or long periods depending on the application concurrency within that line. It is then that a queue management technique that manages the queueing delay as well as the losses becomes critical to manage the distribution of 'quality' shared amongst different applications.

Here is an overview of BT's Broadband network and the reasoning for testing an AQM based solution at its downstream bottleneck, i.e. the multi-service edge (MSE) - BNG which is provided by ALU. BT's needs simpler QoS and traffic management. A large part of the complexity of BT architecture is because we have to share capacity for multiplexing efficiency, but we still want to provide each customer with performance assurances, and all this is crossed by the company's internal divisional boundaries.

BT operates a DSL-based broadband access network for residential and small to medium-sized business customers. Figure 3.1 shows the end-to-end performance is managed by different commercial entities (Openreach, BT WholeSale, Communication Providers) and different QoS and capacity management mechanisms. A crucial element is the BNG, which enforces the traffic downstream prioritisation per user, i.e. elevated (1), real-time/premium (2) and best-effort (3). In between the BNG and DSLAM (DSL Access Multiplexers) (4), the shared aggregated traffic per Service-VLAN (SVLAN) is managed with over dimensioning rulers. While prioritisation of upstream traffic (5) could be done at the customer premise equipment (CPE) using Diffsev, deep packet inspection (DPI) or AQM, in out particular case AQM is the focus of this testbed. From direct interconnect towards the CPs BT considers the QoS agreements and traffic are marked and priorirized (6) accordingly, the same QoS consideration must be taken into account for other providers such as third party transit players (7).

3.1.2 QoS Model Rationalisation

This section provides an overview of BT's current QoS model, the principles that it aims to address and its challenges in terms of complexity.

In general, there is an intuitive preference to map the 3 following concepts:

- **Internet services/applications** , e.g. Voice, Video, Gaming, Web
- **Classes of Services** , e.g. Real Time QoS, Premium Forwarding
- **Queues** , e.g. Expedited Forwarding (EF), Assured Forward (AF), Best-Effort (BE)

In almost a 1:1 basis assuming a pre-defined hierarchy of importance between them. This approach is reflected to the current model: multiple services classified to multiple Classes of Service, using Differentiated Services Code Point (DSCP) markings, and mapped to multiple queues. Figure 3.2 depicts BT current model.

Our attempt has been to challenge this approach in order to reduce the required number of queues, simplify the QoS scheduler at the BNG/MSE and reduce capital expenditure (Capex) (cheaper QoS card at MSE) and operational expenditure (Opex) (by shortening the time to market for new services with less QoS configuration and queue mapping). To achieve this conceptual leap we introduce three dimensions of QoS requirements:

- **Urgency**: Novel ways of reducing packet queueing latency
- **Order of importance**: Need for hierarchy of prioritisation
- **Isolation**: Need separation between greenside vs redside users (guest WiFi or Femtocell vs customer's own traffic) per customer-VLAN (CVLAN)

In addition it is important to appreciate that not everything can be solved by network control. There is a need to

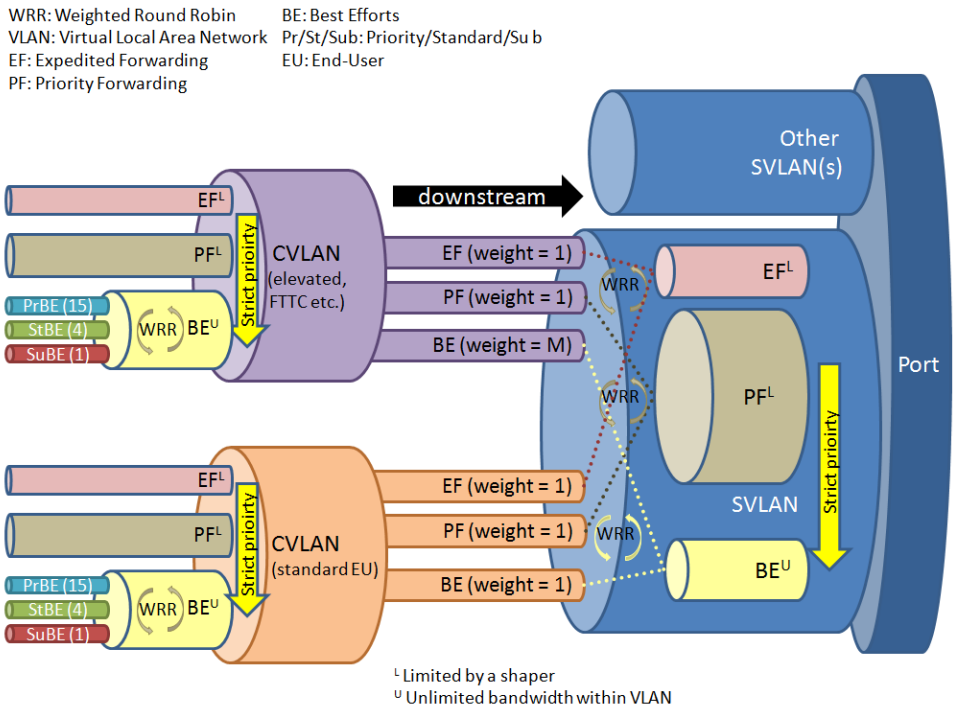


Figure 3.2: BT Current Model

- Decide what is best controlled by capacity planning
- Decide what is best controlled by the application
- Only the rest must be controlled in the network CVLAN

3.1.3 Proposed Model

Based on the principles a proposed model is described with its new pros and cons.

Here the end goal is to satisfy all three dimensions:

- **Urgency**: aim for low queuing delay for all traffic by using active queue management (AQM) and exploiting higher line rates
- **Importance**: could reduce priority (importance) to 2 levels: Value-Added Services (VAS) and everything else (always plan sufficient capacity for sum of all VASs then no need for a separate priority for each value-added service)
- **Isolation**: between traffic within CVLAN e.g. femtocell, or third-party Wi-Fi network (such as FON), i.e. (customer vs non-customer traffic) can be achieved within one queue (to be evaluated)

The proposed model comprises two Classes of Services and two queues: A 'VAS' queue for all value added services ie BT's services (TV, BT Sport) and services other CP's pay BT for low latency behaviour and a 'non-VAS' queue (for anything else). Apply either strict priority between the queues but with required shaping of the 'VAS' queue at let's say 90% of CVLAN rate (to avoid starvation of 'non-VAS' queue) or weighted round robin (WRR) between them without shaping [2]. On each queue we would have an

[2] Comparison between Shaped vs WRR for the anti-starvation function:

- WRR Pros:
  - Potentially no per-line tuning. Configure one pair of weights for all circumstances (e.g. 90:10 for all line rates)
  - VAS class gets full line rate when no non-VAS traffic

AQM mechanism either the same or different one. Perhaps in the 'non-VAS' queue we might need either subqueues or congestion policers to distinguish performance between greenside vs redside services (for WiFi or Femtocell), but this starts increasing complexity and requires further flow classification. In fact, this is not abolishment of the EF queue; the EF is actually reincarnated as the VAS queue (still the VAS queue would be policed not to be exceeded as currently EF is implemented). It is just that the VAS queue would now be able to use a better and scalable transport protocol so to use in a more efficient way the available bandwidth.

- Pros: simpler implementation, simpler required traffic classification, CPs can introduce new services fast. Just has to keep the bandwidth of the total bundle it sells within the capacity assigned to VAS
- Cons: Still we need convincing evidences that this would work, understand the packet loss trade-off especially for the interactive apps

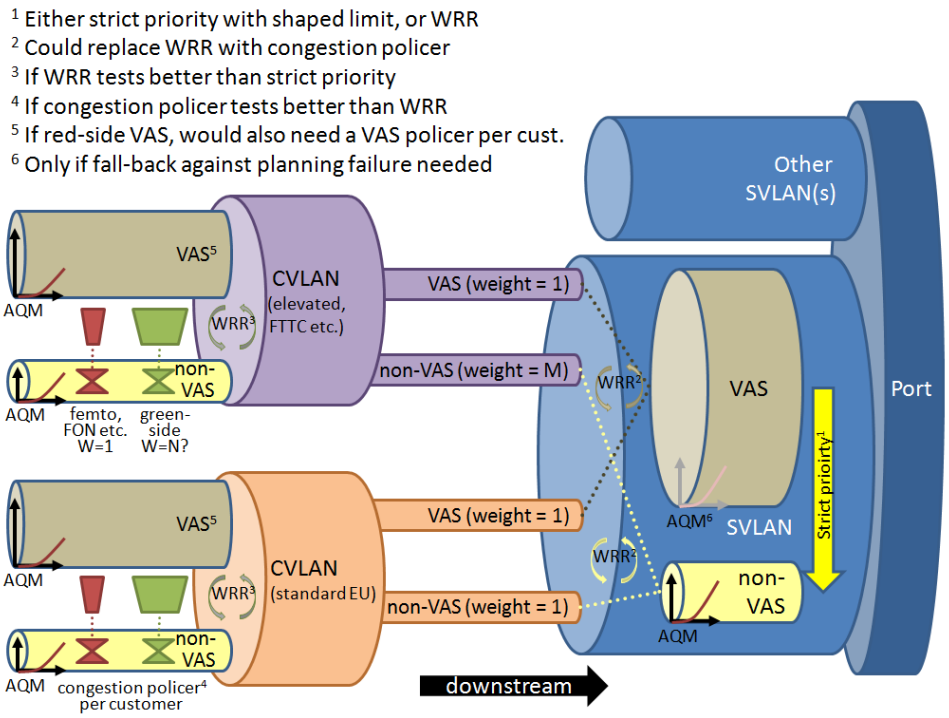


Figure 3.3: Proposed Model

At the SVLAN level we maintain the differentiation between elevated and standard CVLANs or different speed Fiber-to-the-X (FTTx) CVLANs with WRR. However, it is possible that such a separate elevated CVLAN will not prove to be needed; this will need further discussion with BT s market facing units and with the business customers themselves. In addition, there could be a need to pre-configure the weight ratio between greenside and redside (guest femtocell, FON etc) congestion policers, if this feature is enabled. Overall BTs objective is to simplify its QoS 'model' whilst improving the customer experience for end users.

- Shaping Cons:
  - Shaped rate is *absolute*. So, shaped rate will have to be reviewed each time line rates increase, whereas WRR weights are *relative* to line-rate.
  - VAS class sees higher delay because its queue only drains at shaped line rate not full rate (even without non-VAS traffic)
- WRR Cons:
  - Round robin could be additional computational burden to the scheduler and introduce bursts above the nominal CVLAN rate

3.2 Testbed for Web Applications

This section describes the testbed setup, the scenarios that are relevant to BT's QoS architecture simplification incentive, the traffic model used to replicate service application flows as well as stretching utilisation conditions, the novel AQM techniques that will be tested and the test results.

For evaluating the effectiveness of the aforementioned model, a cut-down version is proposed to be replicated at ALU research lab. It focuses on the CVLAN only. For simplicity the plan is to focus on the VAS queue and evaluate AQMs, at what load does local loss or delay degrade perceived QoE of video delivered using HAS technology. This is a simple scenario to prove its effectiveness in coping with future video service requirements.

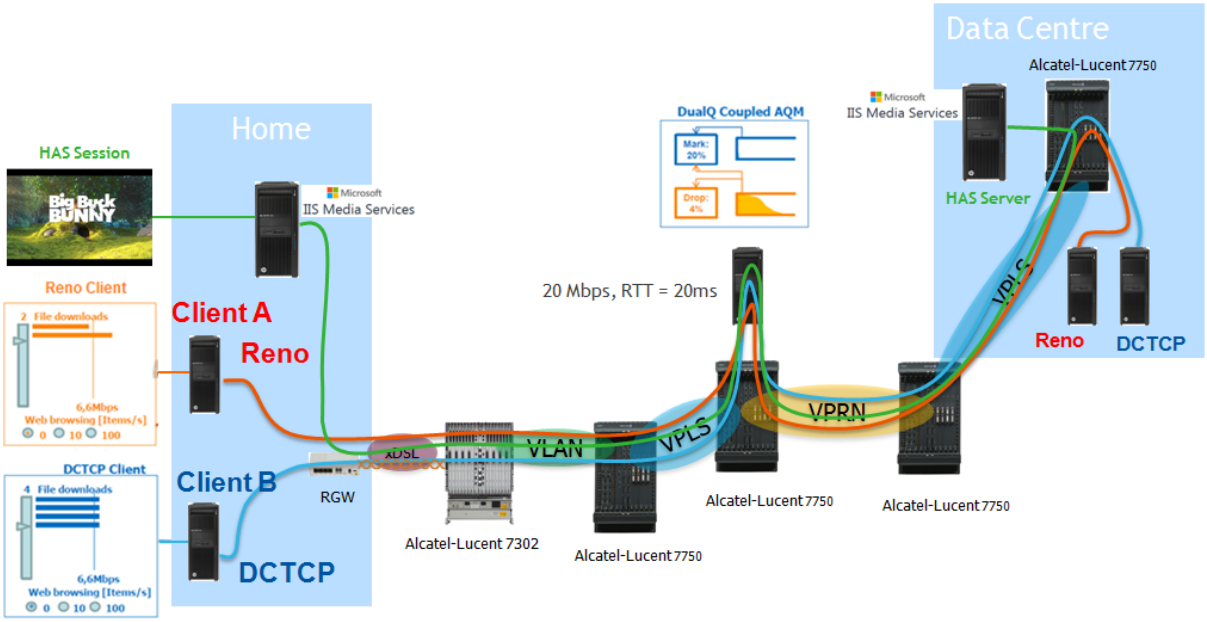


Figure 3.4: Testbed modelling BT deployment setup for HAS video service over DCTCP

3.2.1 ALU testbed

The Alcatel-Lucent testbed is already setup on an architecture which parallels BT residential network and incorporates Hierarchical-QoS (HQoS) and subscriber management models (see section 4.2). This makes it more relevant and straightforward to use the results of this work for BT. In order to reflect the range of applications a customer could generate whilst keeping the testbed manageable we have used three type of traffic: video (HAS), (emulated) web browsing, and long file download. We have tested RITE's DualQ AQM, in combination with "scalable TCP" (DCTCP) at the end hosts, in order to assess how effectively this improves network latency and transport efficiency.

3.2.2 Use case scenario

Video traffic is foreseen to continue dominate and even increase in Internet traffic [19, 20]. HAS video service is the main video delivered mechanism used by OTT services provides, such as YouTube and Netflix. Usually, this is done by overlay content delivery network (CDN) infrastructure, with nodes residing/co-hosted in operator's facilities. In addition, HAS is used by operators to complements their IPTV solution, which is normally deployed using IP multicast. In this case, HAS is deployed to serve the 2nd and 3rd screens, with the advantage of seamless reaching mobile devices and deployment in an OTT fashion.

Figure 3.4 shows the testbed setup and how it represent a typical Broadband scenario for BT. One server at the operator's data centre is tuned to use *Classic* TCP (TCP Reno, Cubic or Compound) and another to use DCTCP. They emulate long-running TCP and dynamic web traffic flows over the residential service network, as representative background traffic. The HAS flows use either *Classic* TCP or DCTCP. This test case scenario is designed to represent a realistic deployment scenario.

The next subsections show how the actual tests were done and results, which demonstrates that this model can fulfil BT's expectation for simplifying QoS, by deploying a novel AQM, i.e. DualQ, and at the same time deliver a better quality of experience to its customers.

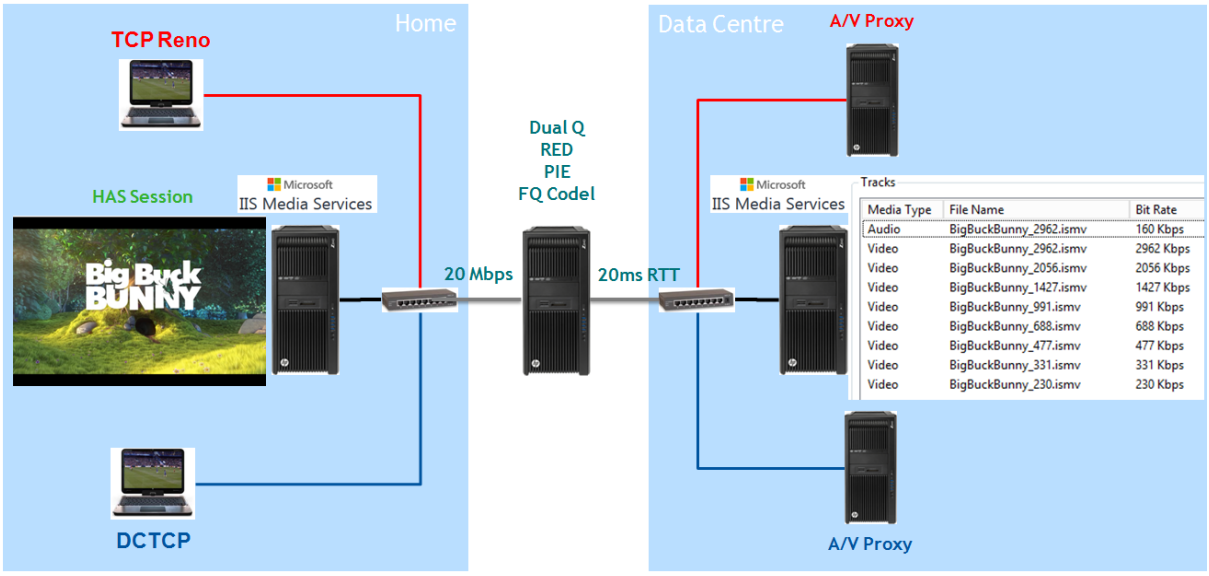


Figure 3.5: HAS demonstrator setup

3.3 Experiments and Results

We have added to the Alcatel-Lucent testbed, a pair of computer servers using Windows Server 2012 R2, including the Microsoft (MS) - IIS Media Services supporting Smooth Streaming [8, 9], see Figure 3.5. At the home side, a HAS session was initiated using Internet Explorer browser with MS Silverlight as the HAS client. At the server side, we have installed the IIS Smooth Streaming HD Sample Content, including both Big Buck Bunny and Elephants Dream content encoded with H.264 video and AAC audio codecs in 8 standard bit rates with a maximum 720p resolution. In all the experiment we have used Big Buck Bunny, which was composed of the following segment files and encoding bit rates:

- BigBuckBunny_230.ismv - 230 Kbps
- BigBuckBunny_331.ismv - 331 Kbps
- BigBuckBunny_477.ismv - 477 Kbps
- BigBuckBunny_688.ismv - 688 Kbps
- BigBuckBunny_991.ismv - 991 Kbps
- BigBuckBunny_1427.ismv - 1427 Kbps
- BigBuckBunny_2056.ismv - 2056 Kbps
- BigBuckBunny_2962.ismv - 2962 Kbps

MS Windows Server 2012 R2 supports, both DCTCP and Compound TCP (CTCP), which is another favour of *Classic* TCP compatible with TCP Reno or Cubic. For testing we could easily switch between serving HAS using DCTCP or CTCP. Two other pairs of client and server computers were deployed at the home and data center sides, they were used to initiate background traffic, which could be either long standing TCP file download and/or short flows simulating dynamic web traffic. Those computers were based on Ubuntu 14.04 with Linux kernel 3.18.9, which contains a recent DCTCP implementation, thus for the background traffic we could easily switch the TCP congestion controller from Reno, Cubic or DCTCP. A Linux server (AQM server) is used to create a bottleneck where packets are dropped/marked when congestion occurs, in addition it is used to configure different AQMs, i.e. DualQ [10, 11], RED [21, 22], PIE [23, 24, 25] and FQ-Codel [26, 25].

Each of the following sections presents a set of experiments with a particular AQM and with different types of congestion control (different TCP variants):

- DualQ: the AQM that RITE has developed
- RED: a traditional AQM
- PIE and FQ-Codel: two emerging AQMs that are regularly discussed at the IETF and Broadband Forum
- Further experiments with DualQ and different mixes of background traffic

In the experiments we measure the video segment quality received by the client, the throughput of each flow over the simulation, and the queue filling at the AQM. The Big Buck Bunny content is a video of almost 10 minutes, thus the experiment were set to last exactly 10 minutes, with statistics collected at one sample per second, i.e. 600 samples per experiment. At the AQM server we created a bottleneck of 20 Mbps with an RTT of 20 msec between the clients and application servers. A 20 Mbps bottleneck was chosen due to the combination of encoded HAS segmented qualities, as listed above, and typically ten TCP flows were initiated as background traffic. The HAS client initiated two flows which were used to download the video and audio file segments. In practice due to our 20 Mbps bottleneck, the share 'fair' TCP throughput over the flows never allowed the HAS session to achieve the highest quality i.e. 2962 Kbps; instead it varied between 2056 Kbps and the lower encoding bit rates.

3.3.1 DualQ AQM

This setup used the DualQ AQM in two experiments, the first used HAS over CTCP (HAS-CTCP), while the second experiment used HAS over DCTCP (HAS-DCTCP). The main objective of these experiments was to verify how HAS performs under *Classic* CTCP versus *Scalable* DTCP congestion control. We launch ten long TCP flows, i.e. long files downloaded at the same time a HAS session was initiated. In each of the experiments the background traffic used the same type of congestion control scheme as the HAS flows. For the background traffic we used Linux equipment, thus for the HAS-CTCP session, we used Reno TCP as background traffic. In the same way, Linux DCTCP was used as background traffic in conjunction with Windows DCTCP for the HAS-DCTCP experiment.

Figure 3.6 shows the results of the quality of the segments received using HAS- CTCP and HAS-DCTCP. It is very clear that HAS performs better under DCTCP, where it was able to achieve a sustainable download of video segments at 2056 Kbps. This was not the same for HAS-CTCP, which in general achieve download throughput rate at one quality below, with encoding segments at 1427 Kbps. It is worth noting that the same network condition was created for both experiments, i.e. one HAS content request, with ten background traffic flow under the same network bottleneck of 20 Mbps and RTT of 20 ms.

Figure 3.7 gives the plots of throughput per flow. The right hand side shows all the flows, whilst (for clarity) the left hand plot shows only the background flows. As mentioned before, the HAS clients initiated two flows which are used to request and download both the video and audio content segments. There is only one audio encoded file, while there are multiple possible video files depending on encoding bitrate see section 3.3. The HAS client used these two flows, as shown in the Figure 3.7 (right), the very

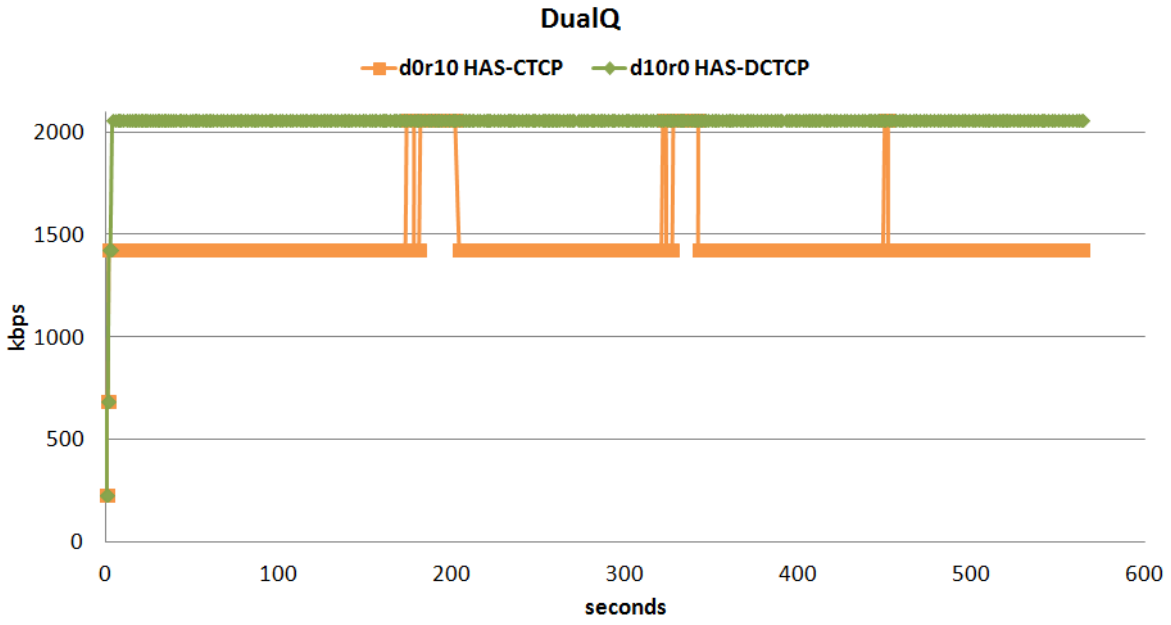


Figure 3.6: Quality of HAS segment for CTCP and DCTCP using DualQ AQM

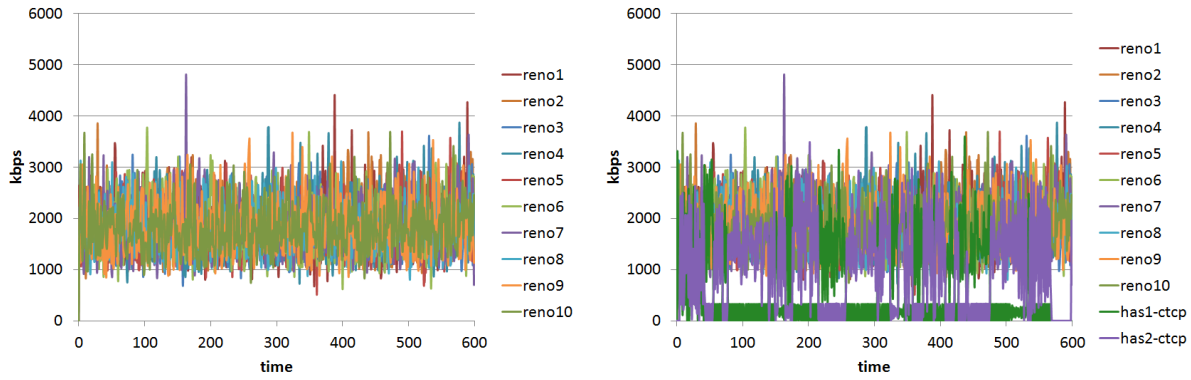


Figure 3.7: Throughput per flow of Reno TCP (left), plus HAS-CTCP (right)

low throughput flow is for the audio stream. But as the graphs show sometimes it peaked, alternating with a deep on the video flow. This is because, there is a more complex mechanism in which the MS - Smooth Streaming uses the audio TCP flow to send video segments. In practice, this does not affect the results at transport level and will not be discussed further, as this is related to how MS implement their HAS algorithm. The TCP flows behaves as expected with the normal sawtooth behaviour, halving the window size on a packet drop. While it maintains the expected average 'fair' shared throughput amongst the flows, it does limit HAS application to deliver higher bit rate segments as HAS is tuned to avoid freezes by chosing lower bit rate segments.

The throughput per flow for the DCTCP experiments are shown in Figure 3.8, on the left it shows ten DCTCP flows used as background traffic, one can immediatly see the difference between this graph and the one from the previous experiment with Reno TCP. DCTCP is able to control much better the sawtooth behaviour due to a better network congestion signally mechanism using ECN marking. On the right, the figure shows the overlap of the HAS-DCTCP, it can achieve on average higher throughput, thus in this case the HAS mechanism request segments with higher encoded rates. Notice that Windows DCTCP implementation follows the standard and sends retransmitted packets over an non-ECN flows. In this experiment just a couple of packet are retransmitted, thus they were not depicted in Figure 3.8.

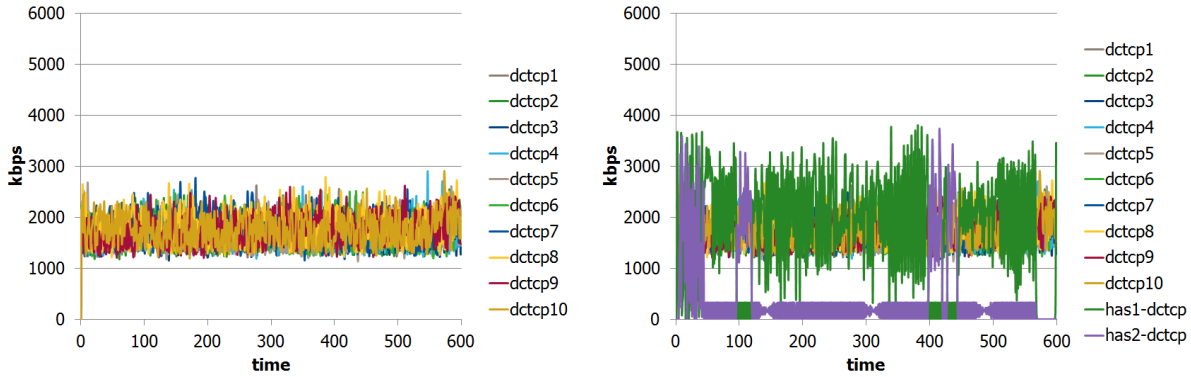


Figure 3.8: Throughput per flow of DCTCP (left), plus HAS-DCTCP (right)

Figure 3.9 shows the cumulative distribution of the queue size at the AQM. In the left picture there is no DCTCP traffic, as both the background flows and HAS used *Classic* TCP. The queue size was of around 20 ms on maximum for DualQ. In the right graph, it shows that the AQM keeps a shallower queue delay for DCTCP, as the DualQ was configure to allow just a couple of packets for the ECN marketed traffic. In addition, in this graph the red curve (Reno) indicates the non-ECN retransmitted packets (as mentioned above, DCTCP retransmit with no-ECN marks), which accounted to a very small percentage of packets on the overall duration of the experiment. They had low scheduling opportunity, since most of the traffic were ECN market packets, i.e. from DCTCP flows.

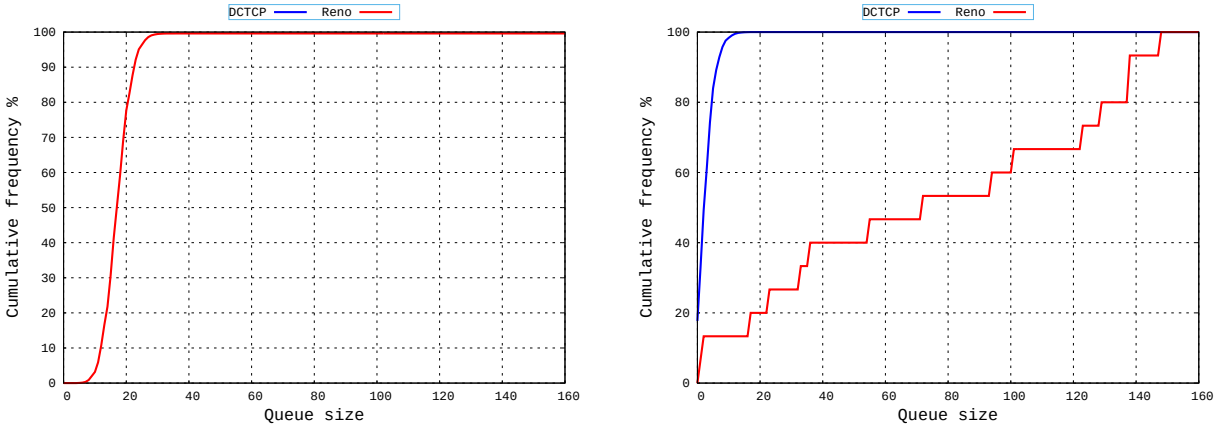


Figure 3.9: CDF of the queue fillings for the experiment with HAS-CTCP (left) and HAS-DCTCP (right). No blue line in the left graph indicates that there was no DCTCP/ECN traffic, queue size in msecs

3.3.2 RED

It was observed that the Linux Kernel 3.18.9 DCTCP implementation did not react to drop by falling back to Reno TCP. In this experiments we wanted to verify Microsoft Windows DCTCP implementation and how it behaves under traditional RED AQM, with no ECN signalling. The same two set of experiment was done, first HAS-CTCP with ten Reno TCP flows as background traffic was initiated over the network with the bottleneck of 20 Mbps and RTT of 20 ms. The second experiment used HAS-DCTCP with the same background Reno traffic.

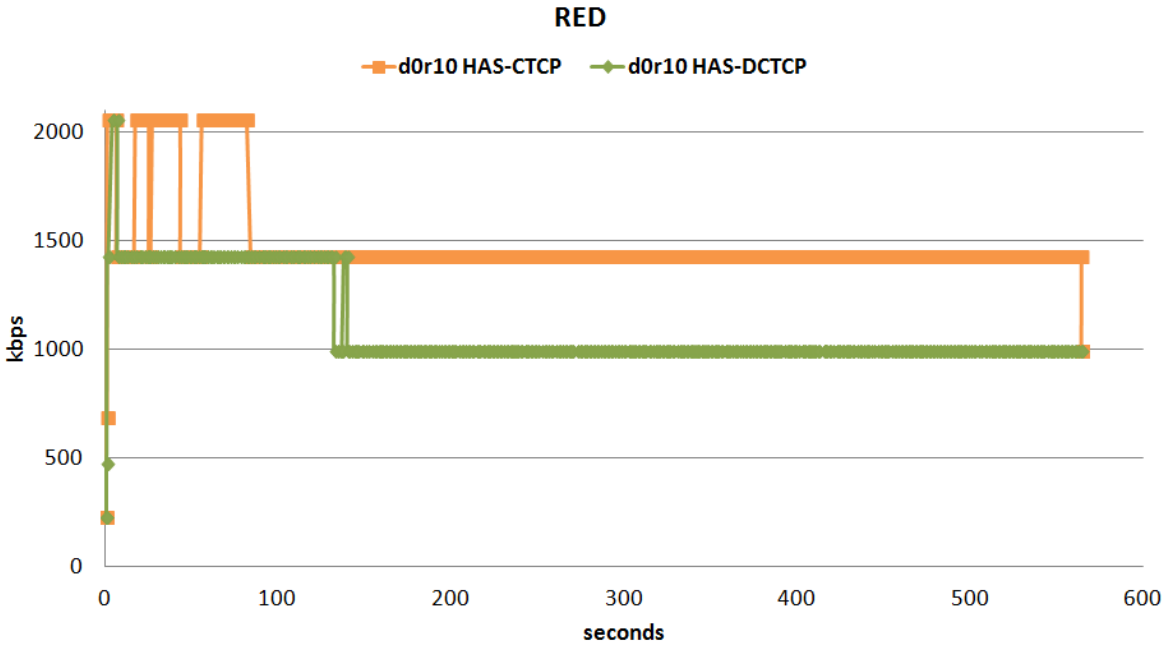


Figure 3.10: Quality of HAS segment for CTCP and DCTCP using RED AQM

Figure 3.10 shows of the quality of the segments received using HAS-CTCP and HAS-DCTCP under RED AQM. As expected, it shows similar result as the previous experiment using DualQ. While HAS-CTCP achieved a very similar video rate quality at around 1427 Kbps, HAS-DCTCP consistently showed to achieve a lower video rate quality at around 991 Kbps. We attribute this to the fact that HAS-CTCP can perform better against Reno TCP background traffic, than the fallback of DCTCP, which will put HAS-RenoTCP against Reno TCP background traffic.

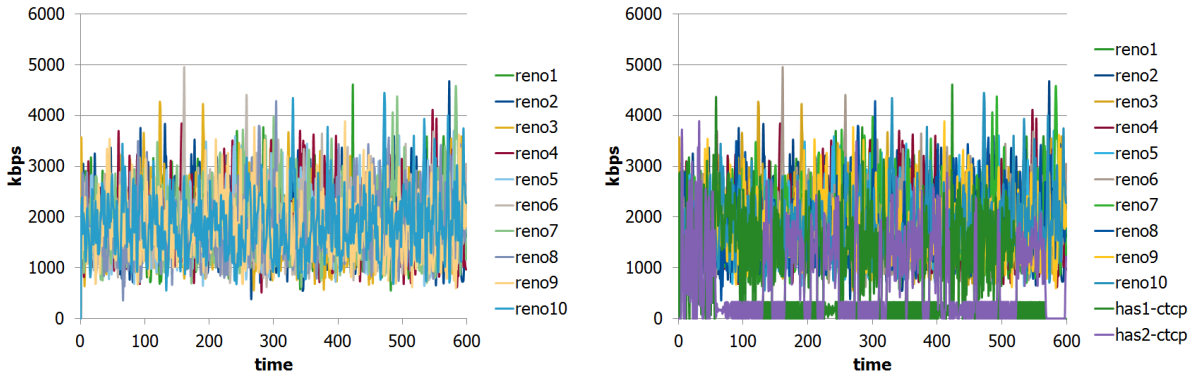


Figure 3.11: Throughput per flow of Reno TCP (left), plus HAS-CTCP (right)

Figure 3.11 depicts the throughput per flow plots, on the left it shows ten Reno TCP flows while on the

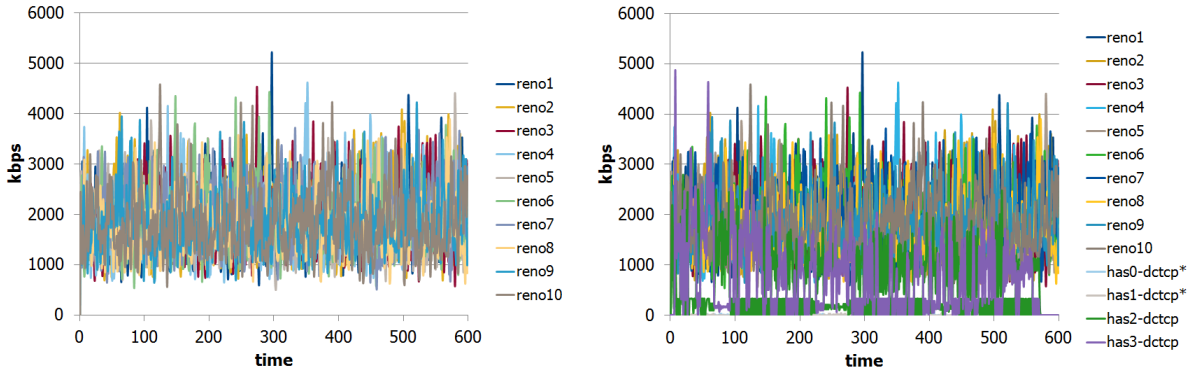


Figure 3.12: Throughput per flow of Reno TCP (left), plus HAS-DCTCP (right). The * denotes DCTCP flows used for packet retransmission with no-ECN markings

right side it shows the overlapped HAS-CTCP flows. The same type of picture is shown in Figure 3.12 but now for HAS-DCTCP, which reacts to drops by falling back to Reno TCP congestion control. Moreover, it still maintains the ECN marks in the packets. On the right graph there are two additional HAS-DCTCP flows, which represent the retransmitted packet from the HAS-DCTCP flows, with no-ECN marks. Unsurprisingly, these graphs show expected TCP behaviour, adding just some further insights on the DCTCP mechanism implemented by Microsoft Windows.

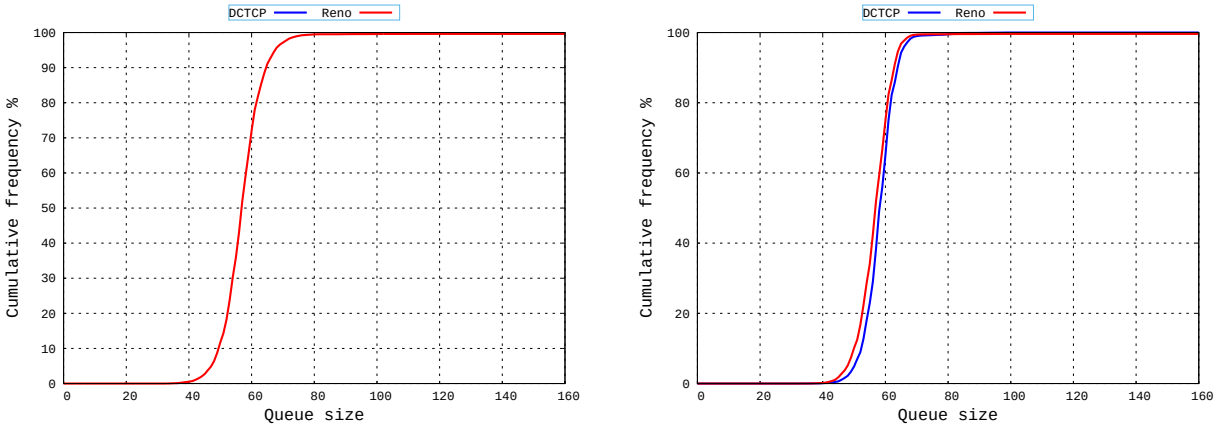


Figure 3.13: CDF of the queue fillings for the experiment with HAS-CTCP (left) and HAS-DCTCP (right). No blue line in the left graph indicates that there was no DCTCP/ECN traffic, queue size in msecs

The cumulative distribution function of the queue size for the RED AQM is shown in Figure 3.13. Both graphs are very similar as expected, on the left there was no DCTCP traffic, thus it just shows the no-ECN market packet on the queue most of the delay at around 60 ms. In the plot at the right, it shows both ECN (DCTCP) and no-ECN (Reno TCP) queue size delay at around 60 ms. This is an expected result since nothing has changed at the AQM, with the same single RED queue.

3.3.3 PIE and FQ-Codel

In these experiment we tested HAS over PIE and FQ-Codel, the main objective here was to understand how these AQM affects the performance of HAS application. Again we used the same experimental baseline configuration: bottleneck of 20Mbps and RTT of 20 ms, ten TCP long file download flows as background traffic and one HAS request, which launch two TCP flows. The experiment were done using HAS-CTCP, Figure 3.14 shows the achieved segment bit rate received by the HAS client for PIE and FQ-Codel over the 10 minutes experiment. HAS performance under PIE does not show much surprise, in general fetching segments at 1427 Kbps encoding rate and sometimes receiving segments at 2056 Kbps. This behaviour is basically the same as on the previous experiment with RED and DualQ. However, HAS under FQ-Codel showed to have a worst performance, fetching segments at 1427 Kbps encoding rate, but many times falling back to segments at 991 Kbps. The reason for that is clearly shown by the throughput per flow graphs below.

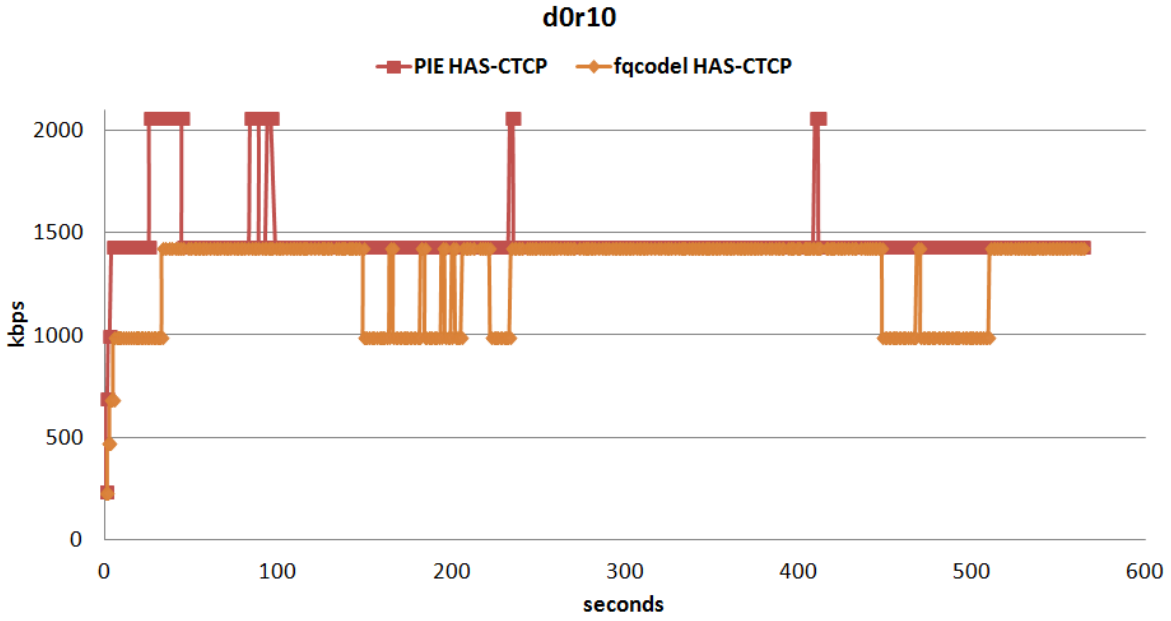


Figure 3.14: Quality of HAS segment for CTCP using PIE and FQ-Codel AQMs

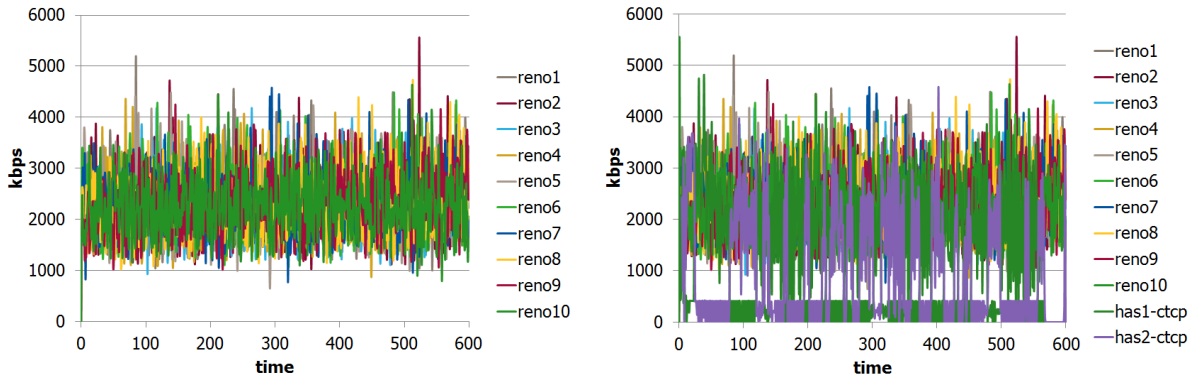


Figure 3.15: Throughput per flow of Reno TCP (left), plus HAS-CTCP (right) using PIE

Once more, the throughput per flow plots using PIE AQM is depicted in Figure 3.14. The left graph shows the plots of ten Reno TCP flows, while on the right side it shows the overlapped HAS-CTCP flows. Figure 3.16 plots the equivalent graph for FQ-Codel, the left graph shows that the FQ-Codel

scheduler strictly distributes the bandwidth amongst the flows, as a consequence it strongly limits the maximum throughput attainable by each flow. On the right side, the graph shows the superimposed HAS-CTCP flows. It clearly indicates why the overall HAS application had a worst performance. Any drop in throughput will indicate the HAS algorithm to requested segments with lower encoding rates to avoid freezes. While the upside with a possibility to peak in throughput cannot happens due to the maximum throughput cap imposed by the FQ-Codel scheduler.

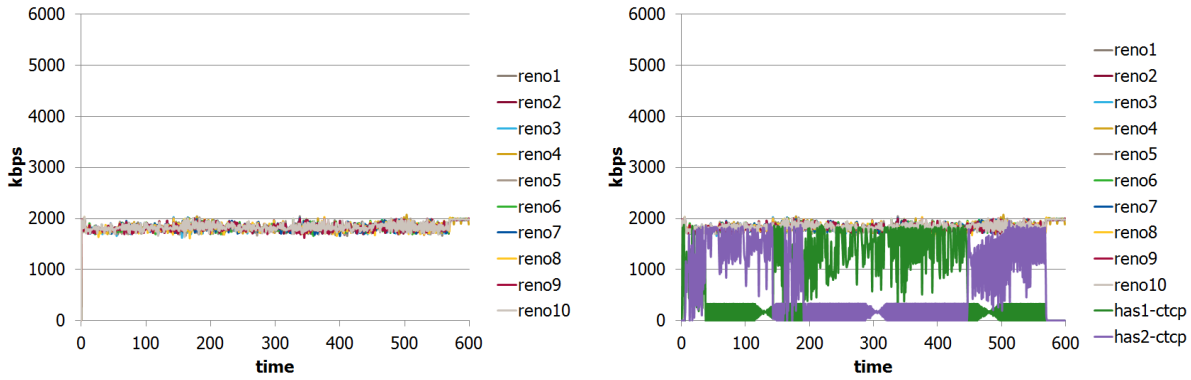


Figure 3.16: Throughput per flow of Reno TCP (left), plus HAS-CTCP (right) using FQ-Codel

Figure 3.17 depicts the cumulative distribution function of the queue sizes. The left graph shows the PIE queue size, which indeed by design maintain an average queue delay of 20 ms to achieve low network queue delay. The plot at the right side, shows the queuing behaviour for FQ-Codel, it achieves high utilization and low queue delays, by strictly distributing the scheduling time amongst the flows, however as shown previously this has negative consequence for HAS traffic.

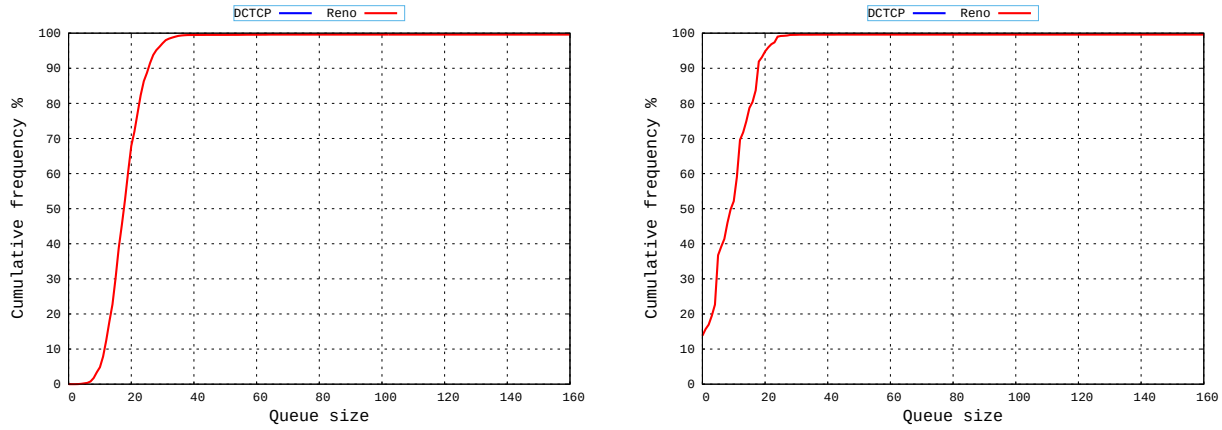


Figure 3.17: CDF of the queue fillings for the experiment with PIE (left) and FQ-Codel (right), queue size in msecs

3.3.4 DualQ with mixed TCP background traffic

The DualQ AQM classifies 'classic traffic' (such as TCP Reno) into a separate queue from 'scalable traffic' (DCTCP) so that the former doesn't delay the latter. The 'scalable traffic queue' is designed with a very low maximum queue size and hence the DCTCP flows are served with a very low latency. The DualQ is configured to equalise the congestion window for all flows, so 'classic traffic' still gets a fair share. In this set of experiments we further tested and evaluated the performance of the DualQ. One experiment tested HAS-CTCP and the second HAS-DCTCP. In both experiments five DCTCP and five Reno TCP long file download were initiated as background traffic.

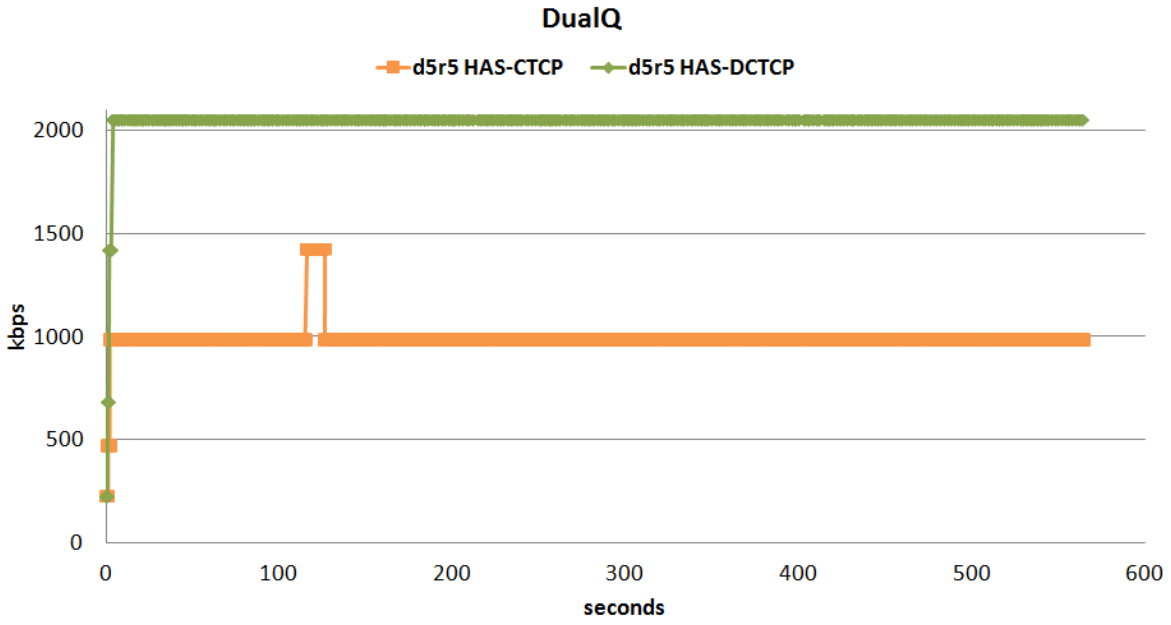


Figure 3.18: Quality of HAS segment for CTCP and DCTCP using DualQ AQM with both CTCP and DCTCP as background traffic

Figure 3.18 shows that HAS-DCTCP can attain a higher quality service delivery versus HAS-CTCP under the same network condition. Once again we observe that DCTCP can utilize the network in a more efficient way, since at congestion the ECN marks informs the sources to reduce the congestion windows in a smoother and more efficient way than halving the window on a packet drop as in TCP Reno, resulting in a more optimal per flow network utilization.

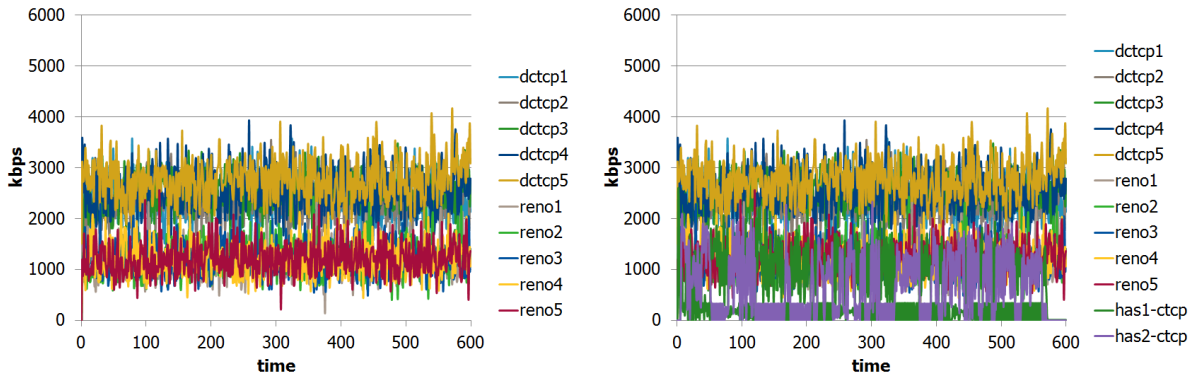


Figure 3.19: Throughput per flow of Reno TCP and DCTCP (left), plus HAS-CTCP (right)

Figure 3.19 and 3.20 show the throughput of the various flows, with HAS carried over respectively classic

TCP (compound TCP) and scalable TCP (DCTCP). In both figures the left hand graph doesn't show the HAS flows; this is for clarity. For the background flows, DCTCP achieved higher throughput than classic (Reno), because the RTT is shorter due to the lower queueing delay in the 'scalable traffic queue'. For the HAS flows, similarly DCTCP achieved higher throughput and consequently achieving a higher segment quality delivery by HAS-DCTCP.

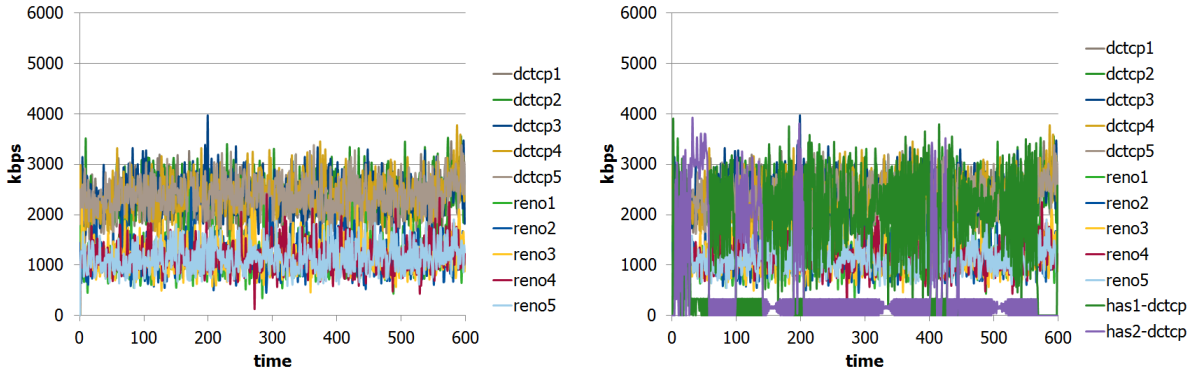


Figure 3.20: Throughput per flow of Reno TCP and DCTCP (left), plus HAS-DCTCP (right)

Figure 3.21 shows the cumulative distribution function of the queue size at the DualQ. The red curves (Reno) represents the *Classic* queue, while the blue curve (DCTCP) the *Scalable* queue. For both experiments *Scalable* queue shows to achieve its design purpose and maintain a very shallow queue, the left graph shows the experiment with HAS-CTCP, while the right graph represent the experiment with HAS-DCTCP flow.

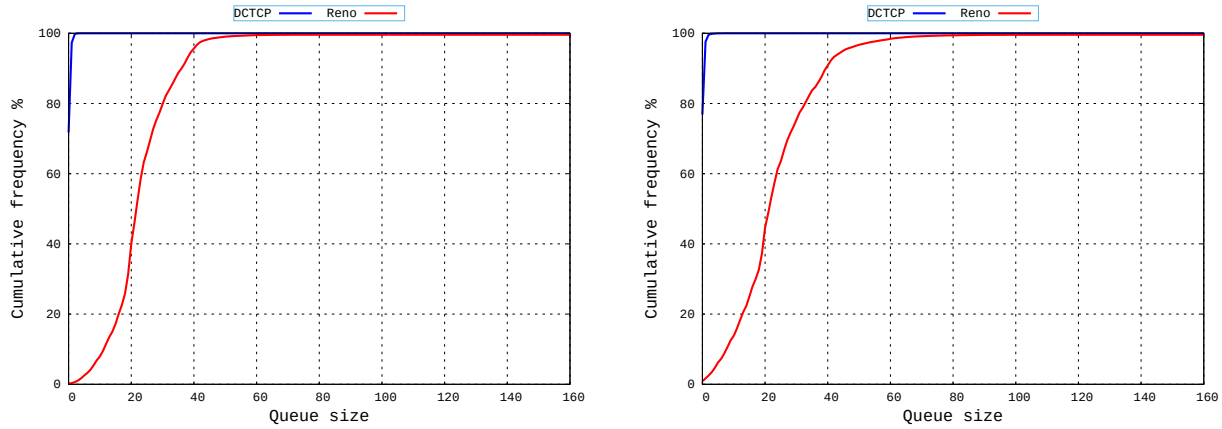


Figure 3.21: CDF of the queue fillings for the experiment with HAS-CTCP (left) and HAS-DCTCP (right), queue size in msecs

3.3.5 DualQ with steady state and dynamic web background traffic

These experiment extended the previous case by adding web traffic as background flows. Thus, it used the DualQ AQM with the same configuration setup, i.e. bottleneck of 20 Mbps and RTT of 20 ms, ten TCP long file download, five DCTCP and five Reno TCP. One experiment used HAS over CTCP while the second used HAS over DCTCP. At the same time the HAS request was initiated, the setup launched 100 web request per second emulating DCTCP web traffic and another 100 web request per second emulating Reno TCP web traffic. The web requests followed an exponential arrival process, while the size of the downloaded files were designed to represent actual Internet web objects following a Pareto distribution with size between 1 Kbyte and 1 Mbyte.

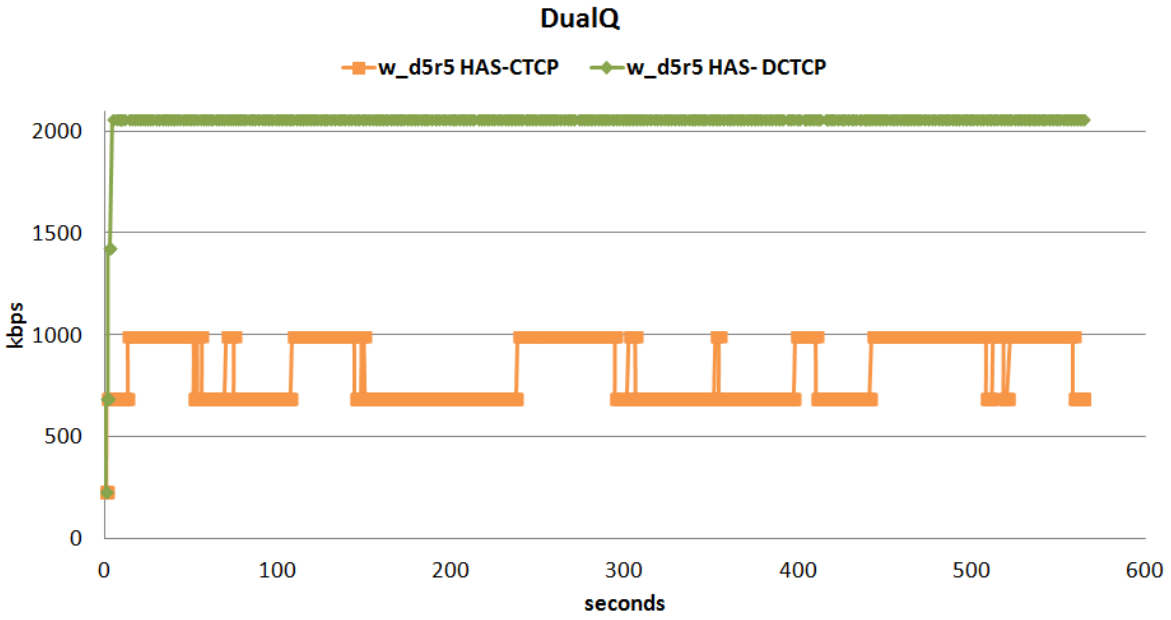


Figure 3.22: Quality of HAS segment for CTCP and DCTCP using DualQ AQM with both steady state and dynamic web background traffic

Figure 3.22 shows that HAS-DCTCP can maintain its performance even with the additional 100 web traffic request per second as background traffic. It consistently delivered segments of 2056 Kbps encoding bit rate. In contrast, under the same network condition, the additional web traffic deteriorate further the performance of HAS-CTCP as the delivered segment alternated qualities between 688 Kbps and 991 Kbps encoding rates. This had a lower overall quality when compared with the previous experiment, which without the additional web traffic could achieved a sustainable delivery of segment with 991 Kbps encoding rates, as shown in Figure 3.18

3.3.6 DualQ experiments with web traffic only

In this setup we focused on the web traffic performance using DualQ AQM, thus HAS traffic were not included in these experiments. The baseline configuration were the same, with a bottleneck of 20 Mbps and RTT of 20 ms. In one experiment we launched 100 DCTCP and 100 Reno TCP web requests per second, with objects sizes following a Pareto distribution as described in previous section 3.3.5. Figure 3.23 shows the completion time of the emulated web traffic. The empty downloads indication at the left shows that there were no long file downloads. The plot on the top-right shows the completion time of the DCTCP web objects, while the plot right below, in the bottom-rightshows the plot of completion time of the Reno TCP web objects. It is clear that DCTCP out performs Reno TCP, since in general DCTCP web traffic took last time, following closely the minimum completion time the objects can be downloaded according to their size. The completion time plot for Reno TCP shows many points taking longer time to download, independently of object size. Moreover, a cluster of points can be observed at around 200 ms which is due to the Retransmission Timeout (RTO). Thus, when a packet is lost, and the TCP stack does not receive a duplicate ack, it waits 200 ms (in Linux) to timeout and retransmit the packet.

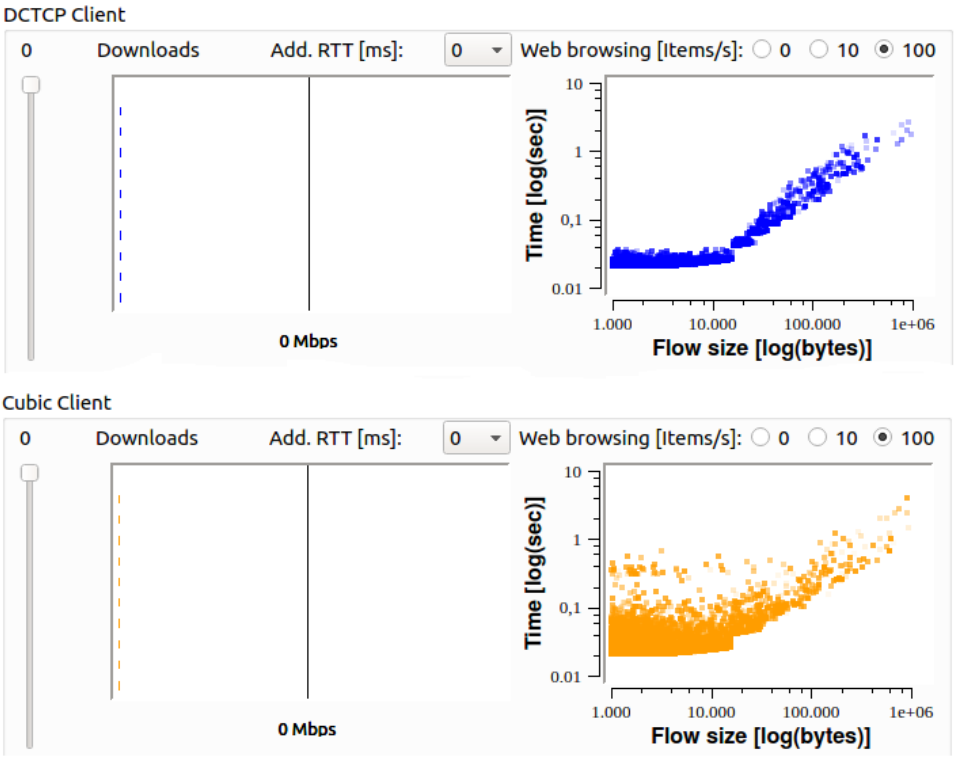


Figure 3.23: Download completion time of emulated web traffic using Cubic TCP (bottom) versus DCTCP (top)

Figure 3.23 shows the results of a similar experiment as described previous. In addition to the 100 DCTCP and 100 Reno web requests per second, it includes 5 DCTCP and 5 Reno TCP long file downloads. The left side of the figures shows the long file downloads, whereas the size of the bars indicated the congestion windows size for each flow. It shows that the DualQ AQM does a good job in giving a fair share for each of the flow, in terms of congestion window, irrespective if it is DCTCP or Reno TCP flows. In addition, it shows that DCTCP can maintain a very steady windows size, in comparison with Reno TCP, this is due to the reaction to drop or mark of each of these congestion control. This is also reflected on the sawtooth effect that is much wider for Reno TCP than in DCTCP. Finally, the plots on the right sides shows that DCTCP maintains a very good completion time for the emulated web traffic, while the performance of Reno TCP web traffic was even worst than compared with the previous experiment, without the long file downloads.

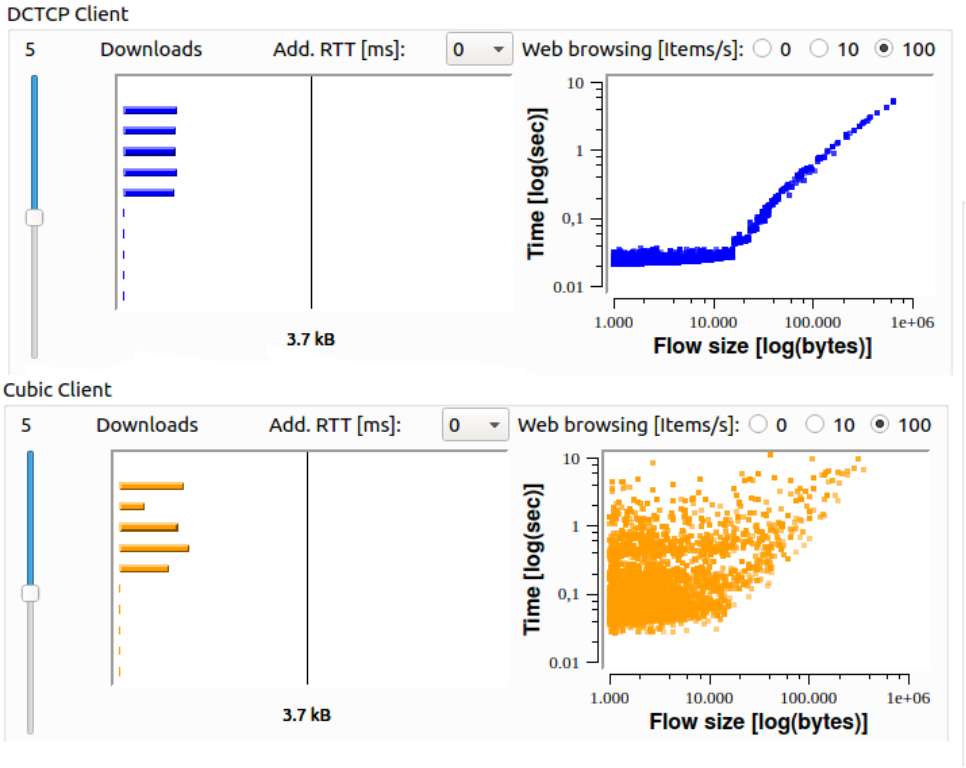


Figure 3.24: Download completion time of emulated web traffic using Reno TCP (bottom) versus DCTCP (top) with 5 DCTP and 5 Reno TCP long file download

3.4 Conclusion

We have developed a testbed and used it to assess RITE's innovative mechanisms for broadband applications. We used the ALU testbed which consists of a classical residential service delivery network composed of xDSL DSLAM (DSL Access Multiplexers), BNG (Broadband Network Gateway), Service Routers (SR) and application servers; further details are in the next section 4. We built on this testbed in order to test RITE's DualQ AQM (Active Queue Management) with various mixes of video, web and file downloads. We also assessed the performance with existing AQMs, such as RED since BT's equipment already has RED implemented, but not turned on, it is important to understand whether RED is 'good enough'. This work meets the requirements of RITE's Description of Work (which was amended to be a broadband testbed from the original specialised financial application).

In summary:

- We have shown that RITE's DualQ AQM mechanism in combination with Scalable TCP (DCTCP) delivers a much better quality of experience for video applications using HAS (HTTP Adaptive Streaming) than Classic TCP. The importance is that HAS traffic now dominates the Internet; Netflix accounts for about 35% and YouTube 15% of traffic at peak time [27].
- The DualQ AQM also substantially improves the latency /download time of other typical applications like web browsing and file downloads.
- We showed that DualQ AQM also allows applications that use Scalable TCP (DCTCP) and 'legacy' applications that use Classic TCP to coexist. There is no need to segregate bandwidth, as a flow of either type of application gets a fair share of the bandwidth. The particular importance of this is that it helps deployability: a smooth transition of transport protocols from legacy classis to scalable TCP.
- The DualQ AQM should enable an operator's Quality of Service Model to be substantially ra-

tionalised by delivering a better QoS with minimum management requirements. There are some details that need further investigation, including discussion with BT's market facing units for example, whether there needs to be a separate elevated CVLAN, the use of shaping vs Weighted Round Robin and the handling of 'redside' services like WiFi FON.

- We have also compared the DualQ AQM with the existing AQM, RED, and with two emerging AQMs, PIE and FQ-Codel. None gave a comparable performance.

4 Deployment and evaluation of RITE mechanisms for interactive video in an Alcatel-Lucent testbed

4.1 Alcatel-Lucent Objectives

In telecommunications, latency is a multidimensional problem that requires innovative solutions at all levels: from data/information acquisition, processing, encoding, packetization, scheduling, routing, transport, decoding, to converting it to the appropriate means of information consumption, i.e. by humans or machines. Figure 4.1 gives a schematic overview of the sources of latency from the *Latency Taxonomy Survey* work done in this project [28]. Alcatel-Lucent realizes that network latency can be a limiting factor for the deployment of several novel services, from interactive cloud base video application to the deployment of novel network architecture such as promoted by Network Function Virtualization (NFV) and Software-Defined Networking (SDN) paradigm. Moreover, as the access network moves towards infrastructure with higher and higher throughput, at gigabit levels, network latency will be more and more visible and a limiting factor to achieve the required QoS/QoE for possible upcoming services, such as network based virtual reality applications.

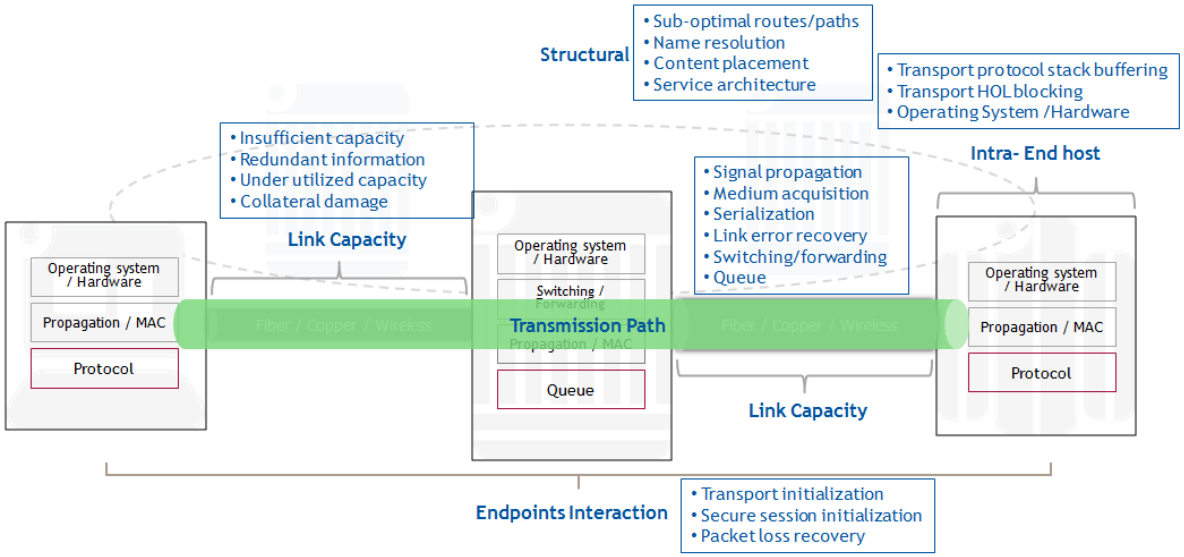


Figure 4.1: Sources of latency as classified by the latency taxonomy survey

4.1.1 Background

Alcatel-Lucent (ALU) develops and manufactures network equipment, from customer premise equipment (CPE) to carrier grade routers, for different types of network, e.g. from fixed to mobile infrastructure. Its equipment serves diverse customer segments from residential to business, industries and application service providers. This imposes a wide range of requirements on its equipment, as it should be capable of performing a variety of different functions under different network conditions. To maintain its competitiveness it is fundamental for Alcatel-Lucent to innovate and address not only the present problems its customers face, but the upcoming challenges of the future telecommunication industry. Network transport latency is fundamentally coupled with the capability of an autonomous system and the Internet as a whole to deliver services with the appropriate QoS/QoE. As already mentioned, network latency is a problem that will be more and more "under the microscope", as gigabit access networks become common elements of the deployed infrastructure.

The RITE project offered the opportunity for ALU to study, investigate, and implement novel techniques to address network latency. In addition to enable the collaboration with leading research centers and researchers in the field. The initial focus has been on network interaction mechanics (WP2), more

specifically on novel technologies that can be implemented in ALU's equipment as opposed to end system application (WP1), such as novel TCP mechanism that are implemented by computer operating system. However, at very early stage it was realized that network latency cannot be addressed solely with network equipment, as a multifaceted problem it needs to be jointly addressed at both the network elements and the end systems, so that we are able to escape the trap of incremental improvements and achieve fundamental changes and solutions. To this end, work on DualQ AQM and AQMs in general, DCTCP and going beyond the *Classic* TCP, with proposal such as *TCP Prime* and initiative such as *TCP Prague* The work of RITE had been disseminated within ALU to different business units with the objective of finding its way into the equipment it designs and manufactures; innovations to help create the network of the future. This work on the testbed use case offered a great opportunity to implement, test and integrate some of the novel techniques on a network architecture and conditions that is both actual and deployed around the globe by several operators.

4.2 Alcatel-Lucent Testbed Setup

The testbed was built using carrier grade equipment assembled in the same lab environment that customer solutions and deployment scenarios are tested and validated. We have used the testbed to evaluate the proposed DualQ AQM mechanism and other AQMs in a realistic setting, having run repeatable experiments in a controlled environment. Additionally it has allowed us to demonstrate realistic applications in an environment mirroring an network provider residential network and service provider cloud based applications. Interactive adaptive video applications requiring very low end-to-end latency were deployed over the setup. We could clearly demonstrate how network enhancements improve the end user experience for such demanding applications. The testbed was generic enough to be used for testing and validating any novel end system or network based mechanism, or mechanisms that combine both end systems and network elements.

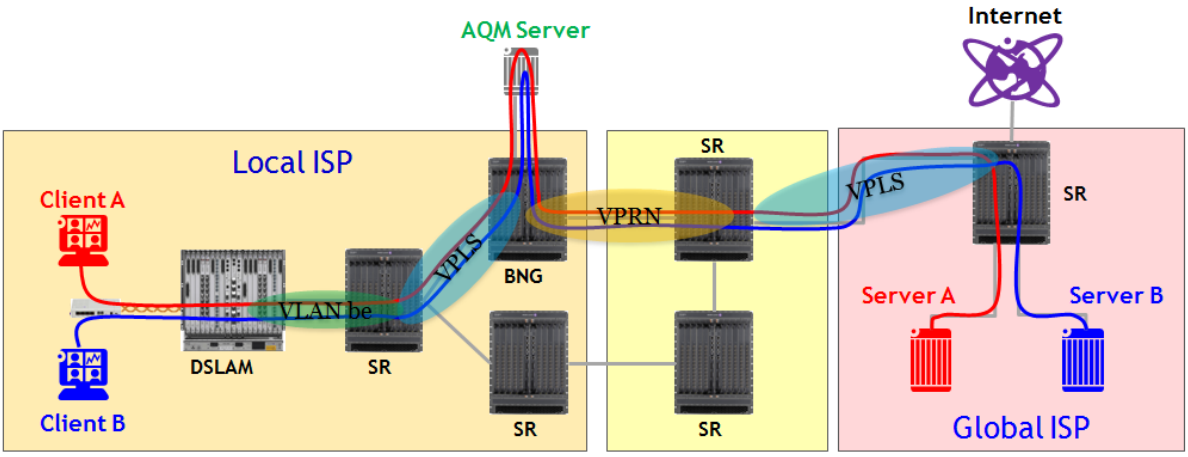


Figure 4.2: Testbed configuration

4.2.1 Network Architecture, Elements and Configurations

The testbed consists of a classical residential service delivery network composed of xDSL DSLAM (DSL Access Multiplexers), BNG (Broadband Network Gateway), Service Routers (SR) and application servers. Figure 4.2 shows the details of the testbed, two residential users are connected by VDSL to a DSLAM, which is connected to the BNG through a layer 2 aggregation network, representing a local ISP or access wholesaler. Traffic is routed to another network representing a global ISP that hosts the application servers and offers breakout to the Internet. The client computers in the home network and the application servers at the global ISP are Linux machines, which can be configured to use any available TCP variant and start applications and test traffic. The two client-server pairs (A and B) are

respectively configured with matching TCP variants and applications. Figure 4.3 shows pictures of the actual testbed setup at ALU lab facility in Antwerp

Figure 4.3: Pictures of the actual testbed setup at ALU lab facility in Antwerp

Within a BNG, per-customer queues form the leaves of a hierarchical scheduling tree. In a production access network, the BNG is deliberately arranged as the bottleneck for the per customer queues. It aggregates all the different service traffic of a user, which it shapes according to the subscribed bandwidth, usually set at configuration to just under the maximum xDSL line speed. Therefore, in the downstream direction of our tests it is reasonable to arrange the BNG as the only bottleneck, where packets are dropped/marked when congestion occurs. A linux server (AQM server) is used to create this bottleneck and to configure the different AQMs that we evaluated. Traffic from the client-server pairs is routed from the BNG through this linux box as to simulate the function proposed for the BNG. This server also controls the experiments and captures and analyses the traffic.

The DSLAM aggregates all the subscribers traffic, which can originate from different BNGs. The downstream ingress traffic at the network termination is queued and scheduled to the respective subscriber line terminator card. This is another possible bottleneck point that could benefit from the proposed Coupled Dual Queue AQM. On the upstream traffic, a good AQM is critical at the output of the residential gateway into the DSL line, which most likely is the bottleneck in the whole access loop Many studies have rightly focused on the upstream, e.g. [29]. However, even though the downstream bottleneck is faster, it can be more important given many customers download more often than they upload.

The testbed was built to reflect a generic residential service network, where different parameters and characteristics can be adjusted. The following setup was used for the evaluations: Two client computers were connected to a modem using 100 Mbps Fast Ethernet; the xDSL line was configured at 48 Mbps downstream and 12 Mbps upstream; the links between network elements consisted of at least 1GigE connections. The AQM server at the BNG creates a 40 Mbps bottleneck, before the configured AQM for the downstream traffic. No bottlenecks are present for the upstream traffic. All Linux servers ran Ubuntu 14.04 (kernel 3.18.9). This Linux kernel version is used for the implementation of DCTCP, Cubic, ECN, RED, PIE and FQ-Codel For the experiments each pair of application client and server was configured with a specified TCP variant, while the AQM server was configured with a specified AQM. Similar downstream load is generated between both pairs. The base RTT (RTT_0) between the clients and servers is 7 ms, which fundamentally originates from the xDSL interleaved FEC configuration.

4.2.2 Panoramic Interactive Video Application

Interactive video applications require a high throughput and low latency service delivery network. The quality of experience of such services can be easily degraded by coexistence with competing greedy traffic such as from file downloads.

Latency has a direct impact on the quality of experience of an interactive video service application, which depends not only on the display quality of the video, but also on the response time each user experiences between the commands (gestures or mouse movements) to zoom or pan the image, and the image reaction to these requests.

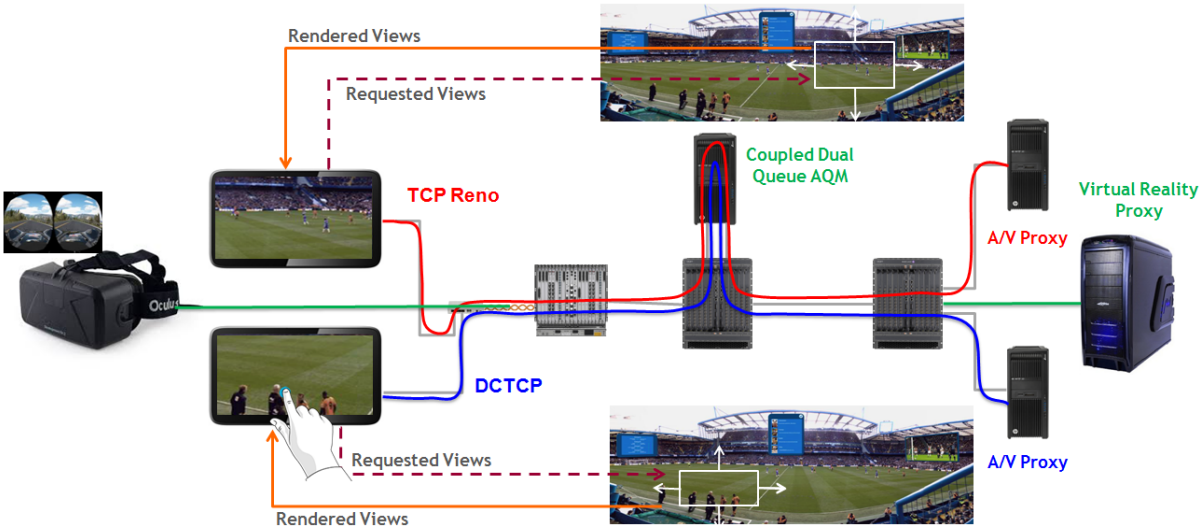


Figure 4.4: Panoramic interactive video application

Figure 4.4 depicts the panoramic interactive video application deployed over the testbed. Two A/V Proxies were installed in the global ISP's network, and two client laptops are installed at the home network. One pair of client laptop – A/V Proxy was configured using DCTCP, while the second pair used standard TCP Reno. Traffic between both end-to-end systems share the same network path. One PIV application benefited from a low latency service using DCTCP, while the other remained on the classic service as their traffic traversed the Coupled Dual Queue AQM mechanism served by the (low latency low loss and scalable) *L4S* and *Classic* queue, respectively.

The application is built so that at the client laptop or tablet the user can zoom or pan the panoramic video. The A/V Proxy encodes and streams an high definition (HD) sub-window of the larger scene tailored for the client on-the-fly. The complete video application includes a live video ingress system composed of several cameras to produce 7K panoramic video images. However, for practical reason our setup relies on a prerecorded 7K video of a sport event. Nonetheless, the client side of the A/V Proxy is no different from a production system. The application is tuned for interactive video, adapting its frame rates to best use TCP's instantaneous rate. At the server, it controls the number of in-flight frames, encoding and sending new frames to the TCP socket depending on acknowledged frames and the maximum number of in-flight frames threshold. When network congestion occurs, such mechanism tries to balance between sending the most recent frame and the rate it can send them; in order to deliver the most responsive video frames in respect to the interactive commands.

In addition to the PIV we integrated a virtual reality (VR) interactive video application; the whole application back-end built based on the PIV application. However, in this case the client device, server video processing application and related software was adapted to interface with the Oculus Rift VR goggle. The content was based on a video recorded with 360 degree camera view. Any slight head movement required the server to change the video camera view accordingly. This setup demonstrated a network virtual reality application, whereas the sensory interface makes service experience very dependant on delay. This application clearly shows that network latency needs to be addressed for such type of service

to become successful and widely adopted by end users.

In the testbed, one client/server pair was configured using DCTCP, while the other used TCP Reno. All the traffic passed through the same bottleneck, going through the Linux server connected to the BNG, where the coupled DualQ AQM was operating. While the PIV application was running, several TCP flows were initiated as background traffic. These were both DCTCP and TCP Reno flows, representing long file download.

Under the same competing traffic conditions, the *L4S* service (i.e. DCTCP) maintained a high quality of experience for the interactive video session, while the interactive video over the *Classic* service (i.e. TCP Reno) showed a noticeably sluggish reaction to end user commands. Moreover, there was a balance on the throughput amongst the long TCP flows, independently if it was a DCTCP or TCP Reno. These experiments showed that the proposed mechanism introduces an evolutionary path to support true ultra-low latency applications for all Internet users without special management or configuration of network devices.

4.2.3 Graphical User Interface

The Graphical User Interface (GUI) was developed for evaluation and live demonstration of how different types of network traffic behave under chosen combinations of AQM, link capacity and RTT. The GUI consists of two main parts - Client blocks (left half of the window, two clients - one for each congestion control) and AQM/Link block (right half of the window), which are described in the respective paragraphs below. All the measurements are taken each second and displayed immediately.

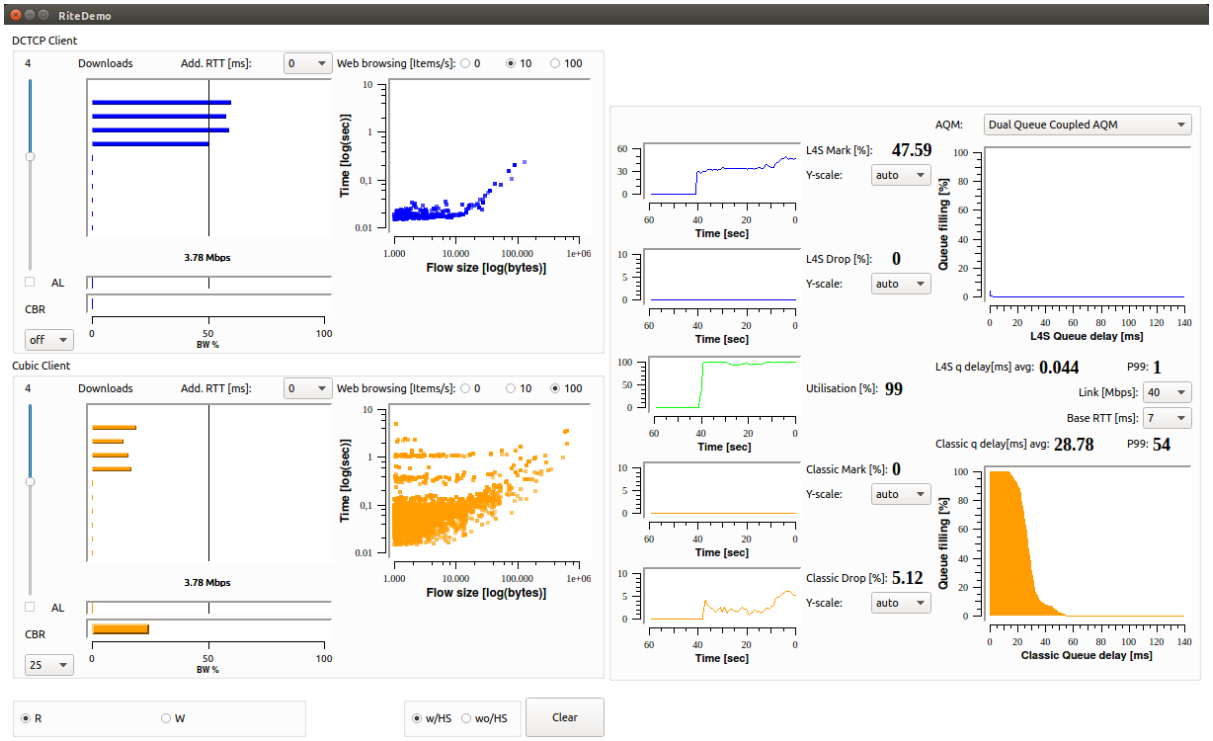


Figure 4.5: Graphical User Interface

4.2.3.1 Client block Each client block consists of long-running flows section (on the left) and short-running flows, or web traffic (on the right). Long-running flows include the chosen number of file download flows (emulated with Session Control Protocol (SCP) [30]), selected with the slider on the right, Application-Limited (AL) flow, generated by the PIV application, and Constant BitRate flow (CBR), emulated with iperf [31].

The bar plot next to the slider displays throughput for each file download flow, while a similar bar plot below file downloads shows the throughput for AL flow. The throughput for file downloads and AL flow is shown as a percentage of "fair rate", which is calculated as link capacity excluding the bandwidth taken by CBR and AL flows, divided by the number of downloads. If all file downloads get a fair share of the bandwidth, their rate will be close to 100%. Alternatively, it is possible to show TCP window instead of throughput, toggled by a radio button R/W, where R is rate, and W is window (located under the client blocks). The AL flows might use less bandwidth than the file downloads at times (due to being application limited), but for convenience, their rate is also shown as the percentage of the calculated fair share of the bandwidth. The CBR flow can be turned on by selecting the desired rate, which is represented as the percentage of the link capacity.

The short-running flows emulate web browsing consisting of 10 or 100 flows per second (selected with the radio button). The plot below the radio buttons shows completion time for each of the flows in seconds. Both X and Y axes of the plot are on a log scale. The completion time is displayed as a measurement that includes TCP handshake by default, which can be switched to the measurement excluding the handshake (toggled by the radio buttons below both clients blocks: "w/HS", meaning ' 'with HandShake", and "wo/HS", meaning "without HandShake"). The "Clear" button below the client blocks clears the completion time plots, removing the old data.

4.2.3.2 Link/AQM block Link/AQM block allows to select an AQM, link capacity, and base RTT from the respective combo-boxes. The block consists of 2 parts - mark/drop probability and link utilization (on the left) and queue delay (on the right). All the measurements are displayed separately for each traffic type (ECN-capable traffic/*L4S* queue, not ECN-capable traffic/*Classic* queue). Mark/drop probability and utilization measurements are shown as both per 1-second sample (on the right) and as a 60-second history (on the right). The X-scale for the history plots is adjusted automatically by default, but can also be changed to a fixed value, selected from the respective combo-boxes. Queue delay is shown in milliseconds per sample, as an inverted PDF in the plots and average/P99 value per sample.

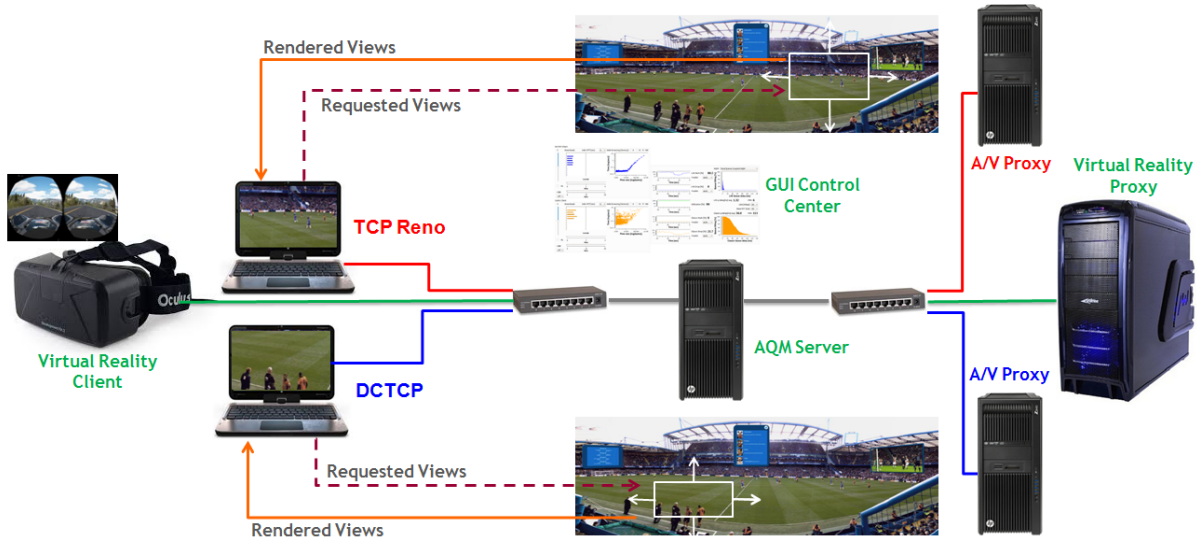


Figure 4.6: Demo setup used to demonstrate the panoramic interactive video application

4.2.4 Demo Setup

The interactive video demo setup was assembled as a compact version of the main test bed. The objective was to create a small easy to transport demonstrator to show and disseminate RITE work at major events such as conferences, fairs or IETF meetings. This demo setup is basically a stripped-down version of the main testbed without the residential service network. Figure 4.6 depicts the elements of this setup, the two client-server pairs composing the interactive video application respectively configured with matching

TCP variants. In addition to host the panoramic interactive video application, those pairs were used to generate background traffic, which could be either long standing TCP file download and/or short flows simulating dynamic web traffic. Moreover, the virtual reality interactive video application was also integrated in the system, whereas the end used could feel the effect of delay in a more immersive sensory way than with the touchscreens. The AQM server was used as the central node, interlinking the client and server application. It was also used to configure different network conditions, such as different RTT delay and throughput bottleneck, which induced packets to be dropped/marked when congestion occurs. In addition, we could configure and test different AQMs, e.g. DualQ, RED, PIE and FQ-Codel.

The demo setup with the panoramic video application was used at the Bits-N-Bites session at IETF-93 in Prague. Bits-N-Bites is an exhibition held in conjunction with the IETF focusing on demonstrations related to Internet standards and technology. The Prague gathering included nearly 1400 engineers, computer scientists and academics at the forefront of developing standards for the continuing expansion and robustness of the Internet. The demonstration was a great success and generated a queue of interested people throughout the session. The technology garnered interest from several major equipment vendors, ISPs and operating system developers. Some pictures of the event can be seen in the image collage in Figure 4.7

Figure 4.7: RITE demonstration at IETF 93 Bits-N-Bites

4.3 Experiments and Results

4.3.1 Test framework for Steady State and Dynamic Load

To be able to generate reproducible test results, a configurable and automated framework was developed. The framework supports documenting and executing test cases, capturing and processing measurement data, and finally plotting results in different relevant representations.

Initially test cases for long term throughput equivalence (sometimes referred to as TCP-fairness) were developed. Each experiment (lasting 250 s) was performed between client-server pairs A and B, with a specified TCP variant configured on each client-server pair and a specified AQM on the AQM server. For all long-running flow experiments, each client started 0 to 10 file downloads on its matching server, resulting in 120 combinations (11x11, skipping combination 0-0).

After finding the useful configurations showing promising results for long term throughput equivalence, test cases for the dynamic evaluation were added. For the dynamic behaviour experiments, load was gen-

erated using an exponential arrival process with an average arrival rate of 0, 10 and 100 requests/second, optionally combined with 1 long flow. The downloaded sizes were Pareto distributed ($\alpha = 0,9$) with a minimum size of 1KB. The client issued a request by opening a new TCP connection and received data sent by the server until the connection was closed by the server. The complete completion time was measured by the client from when it opened the TCP connection to when it received the close notification from the server. The experiment again lasted for 250 s allowing measurement of a completion time profile for every TCP variant on every AQM. 5 load profiles in increasing level of load were tested on each client-server pair (10 r/s; 100 r/s; 0 r/s + 1 long; 10 r/s + 1 long; 100 r/s + 1 long), resulting in 25 test combinations per experiment.

To evaluate the support for non-responding / non-congestion-controlled traffic the test cases can also be executed with an additional CBR UDP flows of 10 or 20 Mbps, with or without ECN capability.

The test script run autonomously from the AQM server, executing each experiment one after the other. For each experiment the congestion controls on clients and servers were configured using different AQMs. Following a predefined configuration, the scripts start the relevant load generators and the measurement tools. After 250 seconds the load and measurement tools are stopped, results were gathered and processed and the plots generated.

An overview of experiments that were executed are listed in Table 4.1.

To allow a thorough analysis of the results, plots were generated preserving detailed information. These plots were combined in overview matrices to allow trends to be detected. Plot matrix's were produced to show flow and class throughput, queue delay, marking and dropping probability, and for the dynamic load, and flow completion time plots.

Figures 4.8 & 4.9 & 4.10 show a relevant subset of the experiments.

Plotted values of flow or class throughput and marking or dropping probabilities are always measured over each second. Flow completion time plots show a dot per request. A reference line shows the completion time that a perfect lone download would have achieved at full line rate after a 2-RTT handshake. The Queue delay CDF plots the sojourn time of every packet for each traffic class. For flow throughput, the Y-axis range was adapted to locate the expected throughput of the N dominant flows across the middle of the graph ($80/N$ Mbps). For example, if 4 Cubic flows compete over the 40 Mbps bottleneck, the number of dominant flows is 4, the expected throughput is 10 Mbps and the scale 20 Mbps. If Cubic flows are starved when 5 Cubic and 2 DCTCP flows compete, then only $N = 2$ flows are dominant, and the upper limit of Y will be 40Mbps. This results in overall comparable plots, scaling the visualisation optimally.

The row and column labels indicate the TCP variant R:Reno, C:Cubic, E:ECN-Cubic, D:DCTCP) followed by the number of long flows and optionally the load level of the dynamic load, on the respective client-server pairs. The left matrix label shows the experiment number, AQM used and the X and Y axis ranges.

4.3.2 Long term throughput equivalence

To demonstrate the starvation problem that the DualQ aims to solve, the first evaluation uses DCTCP and Cubic over RED, PIE, FQ-Codel and DualQ AQMs.

The results are plotted in Figures 4.8 & 4.9. As expected, the DCTCP flows take most of the available bandwidth from Cubic on the single queue AQMs (RED and PIE).

The queuing delay in the RED AQM is very high, but with moderate dropping and marking probability. PIE, on the other hand, controls the delay almost perfectly to the 20 ms target. The DCTCP traffic, even with little variance from the equal throughput rate, is rather unstable. The individual DCTCP flows show an excessive variation or even oscillation, while the Cubic flows starve, clearly due to PIE's built in policy to drop rather than mark ECN capable packets above a 10% marking probability, and DCTCP's incorrect response to drops. Ideally, DCTCP would behave as Reno at drop, so it could support gradual deployment on current Internet. Later tests with the Windows DCTCP show the correct behaviour for drop. Once PIE dropping probability exceeds 10%, there seems to be no way back, as drops make

Table 4.1: Overview of experiments executed

AQM	Parameters	CC A-pair	CC B-pair	Reason
RED	20ms Q target, max 70ms	Cubic	Reno	Cubic fairness to Reno on a classic AQM
DualQ	20ms Q target	Cubic	Reno	Cubic fairness to Reno on a Curvy-RED AQM
RED	20ms Q target, max 70ms, ECN	DCTCP	Cubic	DCTCP and Cubic don't work together on a classic AQM with ECN support
RED	20ms Q target, max 70ms, ECN	DCTCP	Reno	DCTCP and Cubic don't work together on a classic AQM with ECN support
PIE	Defaults (20ms Q target), ECN	DCTCP	Cubic	DCTCP and Cubic don't work together on a classic AQM with ECN support
RED	20ms Q target, max 70ms	DCTCP	Cubic	DCTCP does not fall back to Reno on drop
PIE	Defaults (20ms Q target)	DCTCP	Cubic	Showing additional issues with DCTCP and PIE overload-fallback to drop
FQ-Codel	Defaults (5ms Q target, ECN)	DCTCP	Cubic	DCTCP and Cubic do work together on an FQ-Codel AQM
DualQ	Many slope combinations	DCTCP	Cubic	DCTCP and Cubic do work together on a DualQ AQM, find slope boundaries, verify fairness theory
DualQ	Many smoothing combinations	DCTCP	Cubic	Find smoothing impact
DualQ	More flows (per 2 and per 5)	DCTCP	Cubic	Find load boundaries
DualQ	20ms Q target, also square for marking	ECN-Cubic	Cubic	Using Cubic with ECN as *L4S* traffic on a DualQ applying also squared marking probability
DualQ	Many slope combinations	DCTCP	Reno	DCTCP and Cubic do work together on a DualQ AQM, find slope boundaries, verify fairness theory
DualQ	Many smoothing combinations	DCTCP	Reno	Find smoothing impact
DualQ	More flows (per 2 and per 5)	DCTCP	Reno	Find load boundaries
RED	20ms Q target, max 70ms, ECN	ECN-Cubic	Cubic	Comparing Cubic with ECN and Cubic without ECN on a RED AQM
PIE	Defaults (20ms Q target), ECN	ECN-Cubic	Cubic	Comparing Cubic with ECN and Cubic without ECN on a PIE AQM
FQ-Codel	Defaults (5ms Q target, ECN)	ECN-Cubic	Cubic	Comparing Cubic with ECN and Cubic without ECN on an FQ-Codel AQM
FQ-Codel	With extra (ECN) CBR flows of 10 and 20Mbps	DCTCP	Cubic	4 extra combinations to test the support for application limited CBR traffic
DualQ	With extra (ECN) CBR flows of 10 and 20Mbps	DCTCP	Cubic	4 extra combinations to test the support for application limited CBR traffic
All		All	Same	Verification for equal behaviour on the 2 identical client-server pairs to exclude bias in test results

DCTCP even more aggressive, which can be seen in the marking and droping probability plots (also, but not shown, the marking probability goes to 0 and dropping probability is above 0 for the DCTCP flows).

FQ-Codel seems to handle DCTCP quite well, providing every flow with an almost perfectly stable and equal rate, except when the statistical buffer assignment fails to use a unique queue per flow. If flows of the same class land in the same queue, the throughput deviation from the equal rate is only 50%. If flows of different classes are assigned to the same queue, the Cubic flow starves (as in Figure 4.8 D:8-C:7). This behaviour results in sporadic and hard to reproduce random failure of applications, with potential frustration for users and service support. From a queuing delay perspective, unlike PIE, FQ-Codel is not able to keep the DCTCP flows at its (smaller) target delay, but delay is still low. However, as predicted, the drop probability of FQ-Codel rises quickly with load.

Our DualQ Coupled AQM is able to guarantee equal throughput between DCTCP and Cubic flows. It deviates slightly due to the Classic queue size, which grows at higher loads, resulting in less throughput for the Classic flows, and is smaller at lower loads, resulting in a higher throughput for the Classic flows. The queuing delay for the *L4S* traffic is stunningly low—so low that the CDF plots are nearly perfect step functions. In the D:10-C:10 combination, the marking probability approaches 100%. Combinations

with a larger number of flows revealed a previously unnoticed limit to TCP's ability to scale to low queuing delays, which needs to be fixed (at least in DCTCP). We reported on this in D2.3 Section 3.7.1 "Improvements to prevent harm to other traffic". Essentially, DCTCP or TCP will override any AQM and increase queuing delay to keep at least 2 segments in flight. For Classic traffic, compared to FQ-Codel, loss levels are kept to reasonable levels by relaxing the delay constraint somewhat (see D2.3 Section 2.4.1 "AQM: Fixed Delay Target Considered Harmful").

Further experiments adding a 10 or 20 Mbps unresponsive UDP CBR flow also showed an important difference between FQ-Codel and DualQ. FQ-Codel assumes that capping the CBR flow rate to an equal share is always the correct policy (additionally creating a large tail-drop queue if this rate was not appropriate). Whereas the DualQ AQM allows applications to determine their flow rate and the responsive flows to share the remaining 30 or 20 Mbps evenly. For instance, DualQ would support unresponsive multicast TV up to the link capacity whereas FQ-Codel would divide this by the current number of large flows. Further FQ-Codel treats a VPN tunnel containing many flows as one, and can prevent background/delay-based flows from yielding their throughput to others. It is worth noting that unfortunately in our FQ-Codel D:1-C:1 combination, the 20 Mbps flow was classified into the same queue as the CBR flow, resulting in the starvation of the DCTCP flow. FQ-Codel's queue collisions were more frequent than expected.

4.3.3 Evaluation under dynamic load

Figure 4.10 shows the results for 6 dynamic experiments. We used the (ECN) Cubic experiments as a benchmark to distinguish how much improvement was due to ECN, DCTCP or DualQ. RED and PIE show similar results. FQ-Codel approximates the perfectly equal share for completion times. The small queuing delay (5 ms) just accommodates Cubic's needs preventing underutilization.

The completion time results for Cubic show 2 levels of timeouts: at 1 s (due to lost SYN) and 300 ms (due to lost FIN). Using ECN-Cubic does not reduce the number of lost SYN/ACK/FIN packets, since they don't carry the ECN capability in the IP header. For the SYN and ACK this is in compliance with the ECN standard, for the FINs see below. Interpreting the results, we found an anomaly in the Linux implementation. Flows with 1 or 2 packets of data (below 3KB of data) keep experiencing lost packet timeouts of 200 ms. The reason is that 2 packets are sent without delay and if the last FIN packet is lost, a later close connection call resulted in a separate FIN packet sent without ECN would just come after the RTO of 200 ms as configured in Linux. The current DCTCP implementation uses ECN flags in the IP header of all packets.

In Experiment 4 (PIE), the dots at 300 ms (due to drop of final packets) indicate that burst allowance is not effective if other long flows are present. The results for RED were almost identical. Using ECN clearly has an advantage, resulting in shorter completion times as drop is partially avoided. FQ-Codel's burst allowance is effective as it works per flow, and ECN provides no significant improvement (Experiment 5). Sporadic occurrences can be attributed to queue collision with an ongoing flow. Also, lower completion times for ECN-Cubic are due to fewer retransmissions of other dropped packets. ECN has no significant impact on queuing delay.

In Experiment 6 we configured the DualQ AQM with parameters adapted to couple ECN-Cubic with DCTCP, for that we halved the *L4S* slope, additionally applying a squared probability to the ECN-Cubic. As a result, we see significant improvement for the completion times, similar to FQ-Codel. Also, we see near zero queuing delay for the *L4S* traffic, but with a significant reduction in utilisation when competing with many short flows.

Comparing Experiment 5 to 7, Cubic has more completion time outliers when a long running DCTCP flow is active, probably due to FQ-Codel's queue collisions. Additionally, long running DCTCP flows creates a much larger queue, explaining the full utilisation.

Comparing Experiments 5, 7 with 6, 8, we can again conclude that our DualQ AQM approximates the qualities of the FQ-Codel AQM without the need for flow identification and more complex processing. The main advantage is DualQ's lower queuing delay for *L4S* traffic. Compared to Cubic, DCTCP improves utilisation as it reduces the throughput more appropriately on congestion signals—but it can

only regain the available capacity incrementally when a short flow ends. One issue with DCTCP also becomes apparent for flows bigger than the initial windows size of 10. As the marking probability is much higher, slow start is consistently prematurely interrupted. A good result is that no slow start overshoot is detected (zero Q delay), but it leads to unnecessarily longer completion times, with the interruption of slow start. The outcome suggests that a gradual slow start exit scheme may remedy this short coming and it is possible to be designed. Again, Dual Q queuing delay is nearly zero for DCTP traffic, and even for Classic traffic its delay is nearly as low as FQ-Codel.

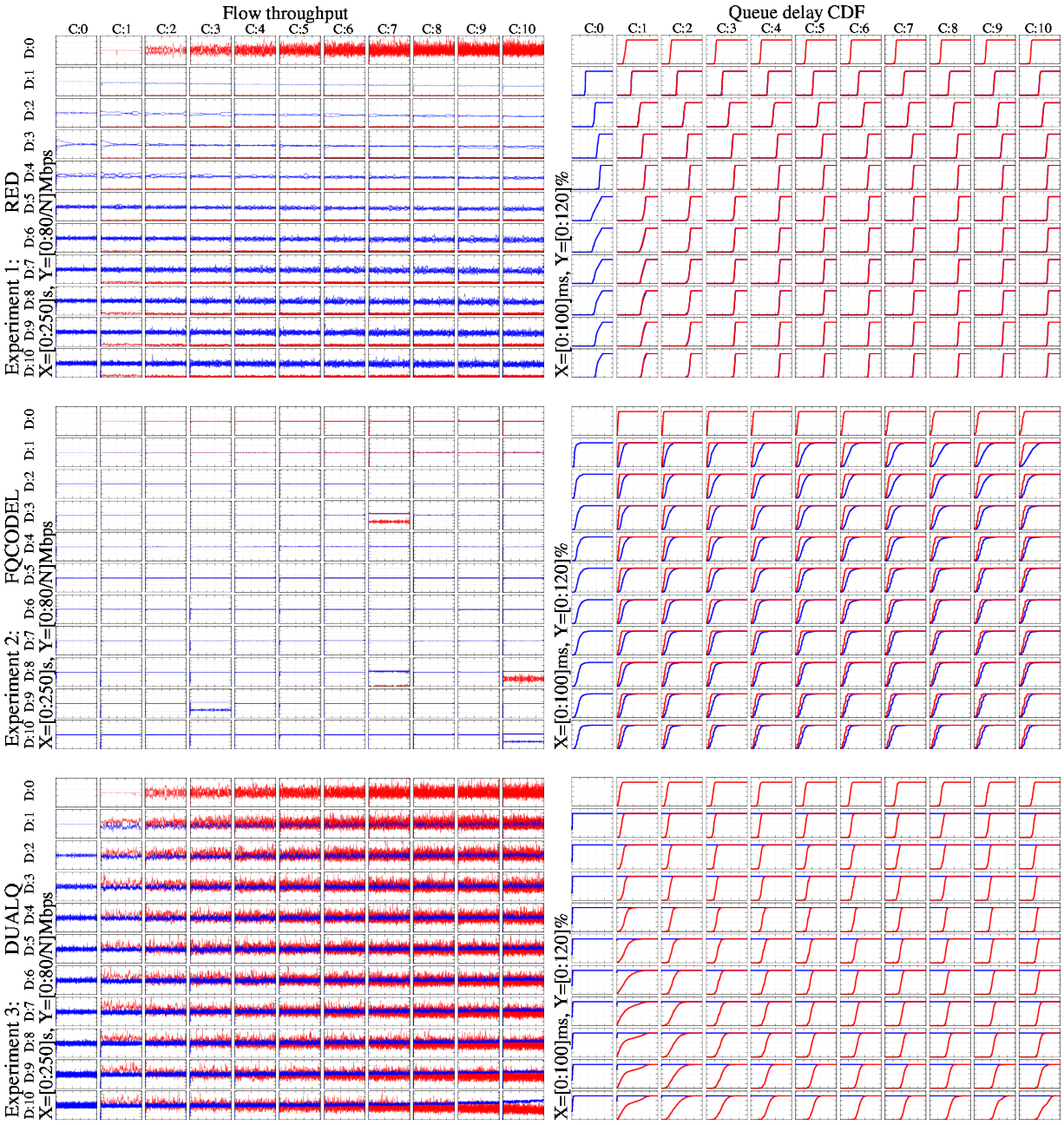


Figure 4.8: AQM comparison for long flows
Showing the coexistence problem (Exp 1) and potential solutions (Exps 2 & 3).
D: Number of DCTCP flows (blue), C: Number of Cubic flows (red), N: the number of expected dominant flows.

Note: With Dual Q, the queue delay CDFs for DCTCP (blue) are hard to see, because they are all nearly perfect step-functions.

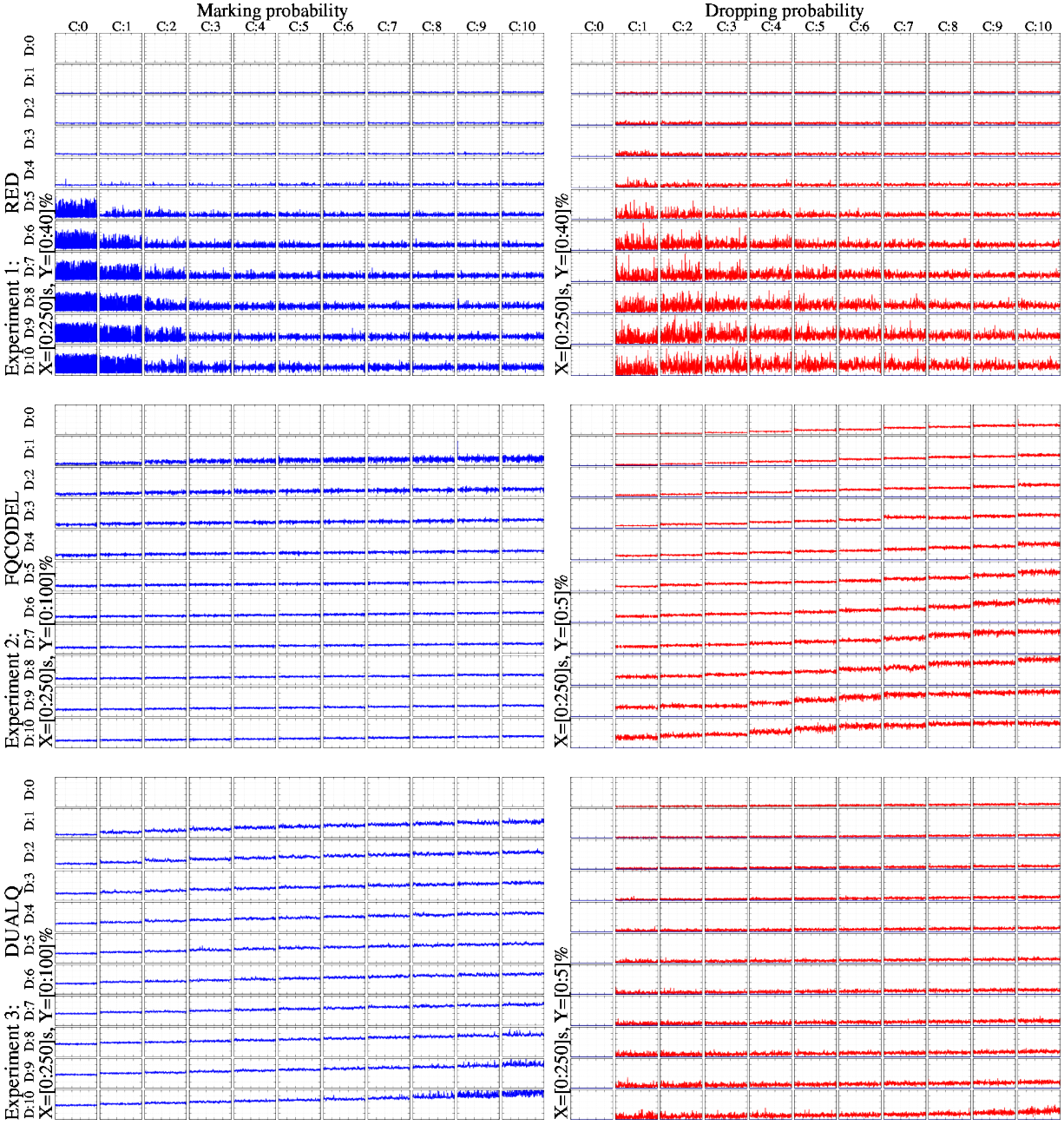


Figure 4.9: AQM comparison for long flows (cont.)
D: Number of DCTCP flows (blue), C: Number of Cubic flows (red).

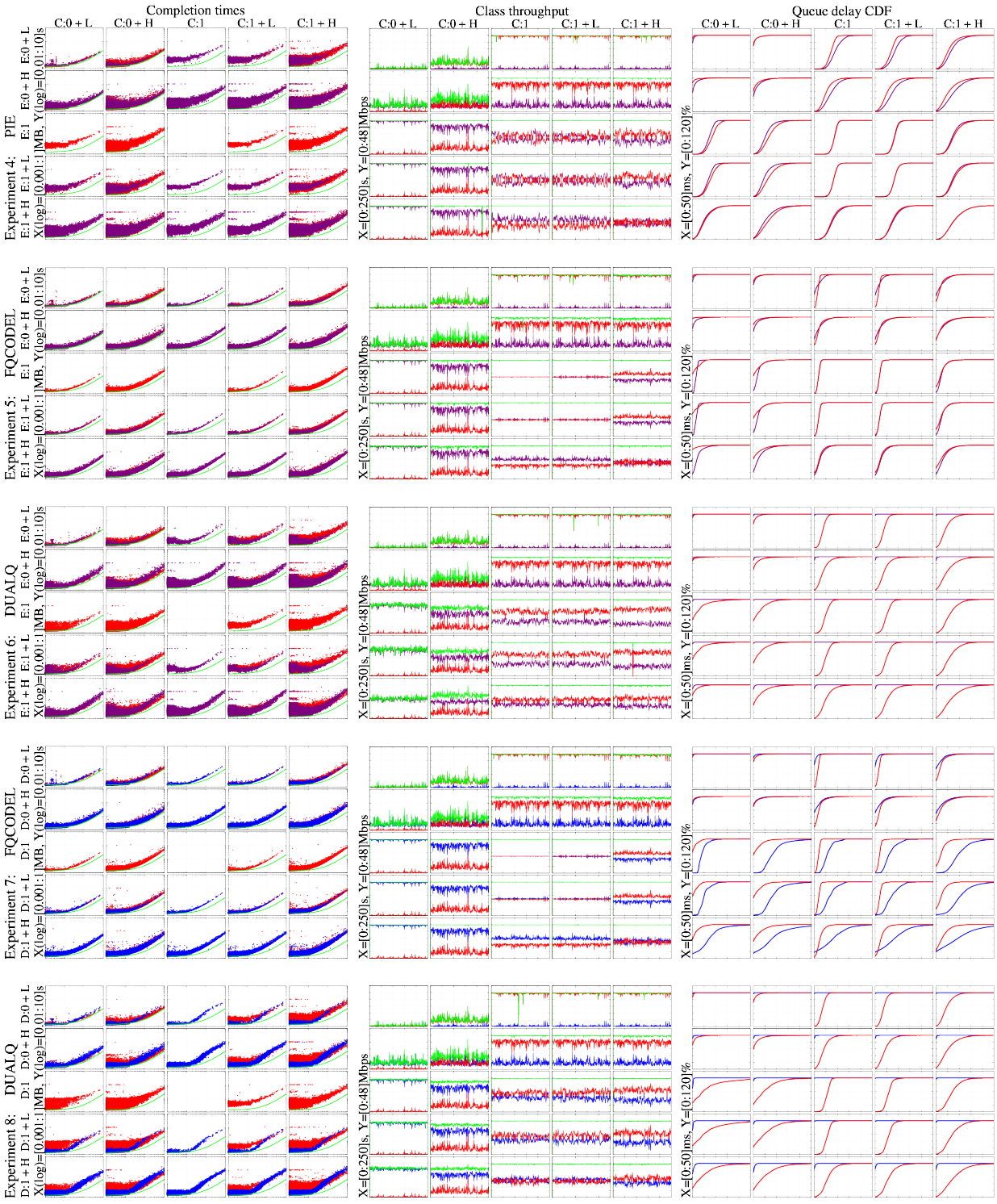


Figure 4.10: Dynamic load for different AQMs for ECN-Cubic & Cubic (Exps 4–6) or DCTCP & Cubic (Exps 7 & 8).
E: Number of ECN-Cubic flows (purple), D: Number of DCTCP flows (blue), C: Number of Cubic flows (red), Green = total throughput, showing utilisation. L: 100 ms exponential load, H: 10 ms exponential load.
Note: With Dual Q, the queue delay CDFs for DCTCP (blue) are hard to see, because they are all nearly perfect step-functions.

4.4 Conclusion

We have fulfilled the interactive video testbed objects defined by the RITE description of work:

- The testbed was built using carrier grade equipment assembled in the same lab environment as customer solutions and deployment scenarios are tested and validated.
- It consisted of a classical residential service delivery network composed of xDSL DSLAM (DSL Access Multiplexers), BNG (Broadband Network Gateway), Service Routers (SR) and application servers.
- We have used the testbed to evaluate the proposed DualQ AQM mechanism and other AQMs in a realistic setting and run repeatable experiments in a controlled environment. In addition, we used several different flavours of TCP transport protocol.
- The testbed is generic enough to be used for testing and validating any novel mechanism, be it end system, network-based or mechanism that combines interaction between both end systems and network elements.
- Two interactive video application were deployed on the testbed, a panoramic interactive video application where user can zoom or pan a panoramic video served from the cloud. And a virtual reality (VR) interactive video application using a Oculus Rift device, were any slight head movement required the remote server to change the video camera view accordingly from video recorded with 360 degree camera view.
- A GUI was developed for evaluation and live demonstration of how different types of network traffic behave under chosen combinations of AQM, link capacity and RTT. In addition to allow easy configuration of the setup, the GUI displayed several measurements in real-time.
- Evaluations of proposed mechanisms by RITE were done using a framework which supported documenting and executing test cases, capturing and processing measurement data, and finally plotting results in different relevant representations.
- Exhaustive tests were carried out on a very comprehensive list of test cases. Tests were automatized and configured to use a variety of different network conditions, different AQMs and end-systems protocols, such as UDP and several different TCP favours.
- We have successfully implemented and evaluated mechanisms that improved latency. This was demonstrated using end-used devices and can be verified in the results from the extensive evaluations and those reported previously.

5 Deployment and evaluation of RITE mechanisms over UoA emulated testbed

In addition to the fully-fledged industrial testbeds described in this report, RITE also employed a set of laboratory testbeds. Smaller network network testbeds are useful for initial prototyping and ease of portability. This section describes a testbed developed at UoA based on virtual machines (VMs) that was used to carry out integration tests for mechanisms developed in WP1.

5.1 Emulated Testbed description

The Alcatel-Lucent low-latency testbed was replicated at UoA as an emulated testbed with the goal of jointly evaluating RITE mechanisms developed in different parts of the project. The emulation was based on GNS3 (Graphical Network Simulator-3) [32], a software emulator featuring Cisco's Dynamips software that is able to emulate a combination of virtual devices based on Cisco's IOS, virtual machines (VMs) and real machines. Fig. 5.1 shows the topology of the network simulated at UoA. It consists of five Linux VMs equipped with the RITE integrated kernel (see D1.3 Appendix) and two emulated Ethernet switches.

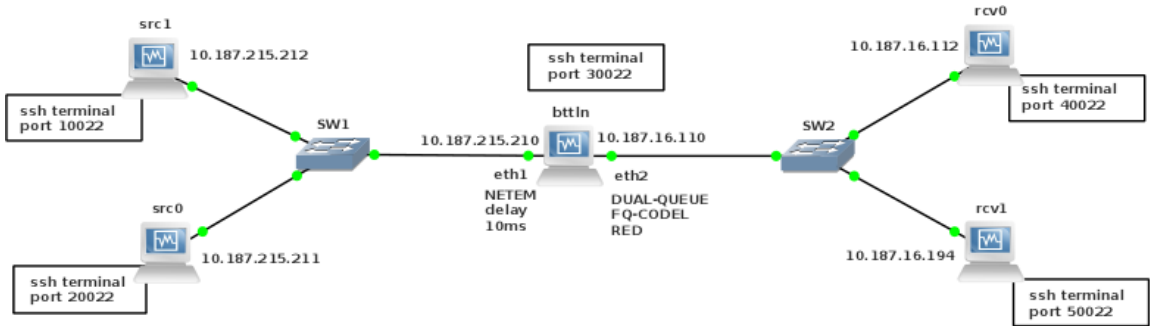


Figure 5.1: Topology of the network emulated in GNS3.

The virtual hosts src0 and src1 act as traffic sources, rcv0 and rcv1 as receivers, and bttln as a link emulator. Since the connection between src0 and rcv0 and between src1 and rcv1 can be configured with a different transport protocol, this testbed allows simulating scenarios with heterogeneous competing flows. This emulated network path represents the same path and used the same AQM parameter configuration as used in the Alcatel-Lucent low-latency testbed (Section 3). Bttln can limit the capacity towards the receivers at a certain bit-rate and introduce an artificial delay so that the RTT of the source-receiver connections can be varied. Also, bttln can enforce various AQM schemes towards the receivers, including Dual-Queue, FQ-Codel and RED. Although the mechanisms and configuration parameters were consistent with the ALU low-latency testbed, the aim of these experiments was not to evaluate changes to the AQM methods, but rather to verify the correctness of the end-to-end mechanisms when faced with the particular characteristics of these new RITE mechanisms.

5.2 Performance of ABE with Dual-Queue

To evaluate the performance of ABE with Dual-Queue we ran three bulk TCP/Reno data transfers as background traffic between src1 and rcv1 and a bursty source between src0 and rcv0 as foreground traffic. The bursty source produced data in bursts inserting a 0.5 s idle period between the end of the transmission of a burst and the beginning of the next burst. The tests used synthetic application-limited burst traffic, rather than a detailed traffic model. The use of synthetic traffic minimises the impact of the complexities of actual application-limited traffic, important for a verification test.

The four graphs in Figure 5.2 compare the CDF of the burst transfer latency when the foreground source is ABE with $\beta = 0.7$, ABE with $\beta = 0.85$, and non-ECN-capable. These two values were chosen within

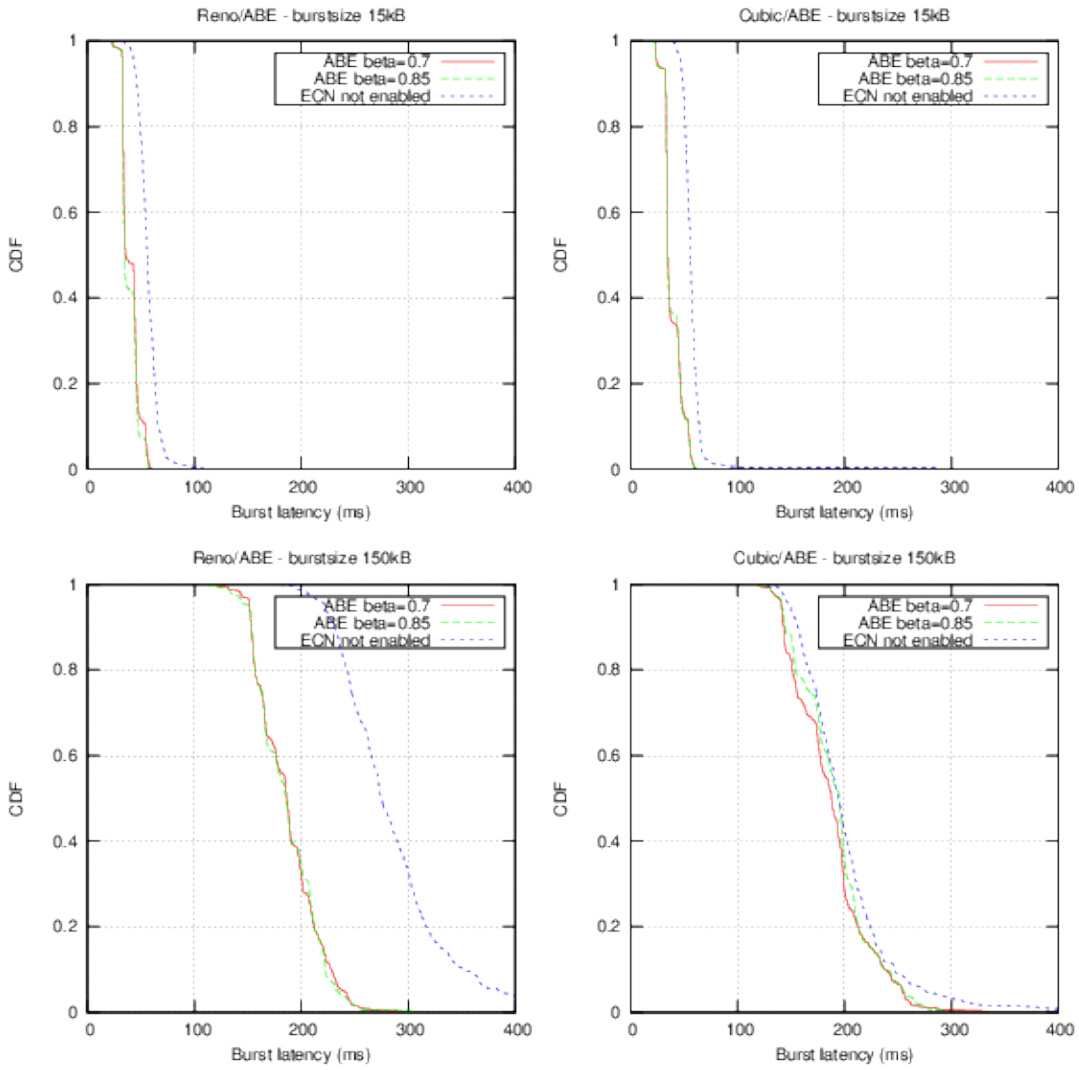


Figure 5.2: CDF of transmission latency using ABE with $\beta = 0.75, 0.85$ and non-ECT. The top two graphs refer to a burst size of 15 kB, while the two bottom graphs to 150 kB. The left column graphs consider Reno foreground traffic, while the right graphs consider Cubic foreground traffic.

the interval of β [0.7,0.85] that was acceptable for use, based on analysis of ABE in [33]. When ABE was used, ECN was enabled also for the background traffic so that foreground and background traffic shared the same queue at the bottleneck. In particular, non-ECN-capable flows are subject to a dropping policy that was quadratic with respect to the average queue length, whereas ECN-capable flows are subject to a marking policy linear with respect to the average queue length.

The four graphs represent the burst latency for a burst size set to 15 kB and for a burst size set to 150 kB (the top and bottom row respectively) and when the foreground TCP flow is Reno/ABE or Cubic/ABE (the left and right column respectively). The diagrams illustrate the significant latency reductions that can be achieved enabling ECN even though ABE was not specifically designed to be used with the RITE Dual-Queue scheme developed in WP2. Reno/ABE and Cubic/ABE benefit from the low-threshold marking indications provided by Dual-Queue[3], which allows TCP to early back-off from full-queue conditions. The actual benefit is likely to depend on the foreground and background traffic, and the bottleneck configuration. Although the experiments do not provide a comprehensive analysis, they do indicate the ECN mechanism operates correctly and can can prevent most of the retransmissions

[3]Dual-Queue started marking packets with an average queue size of five packets.

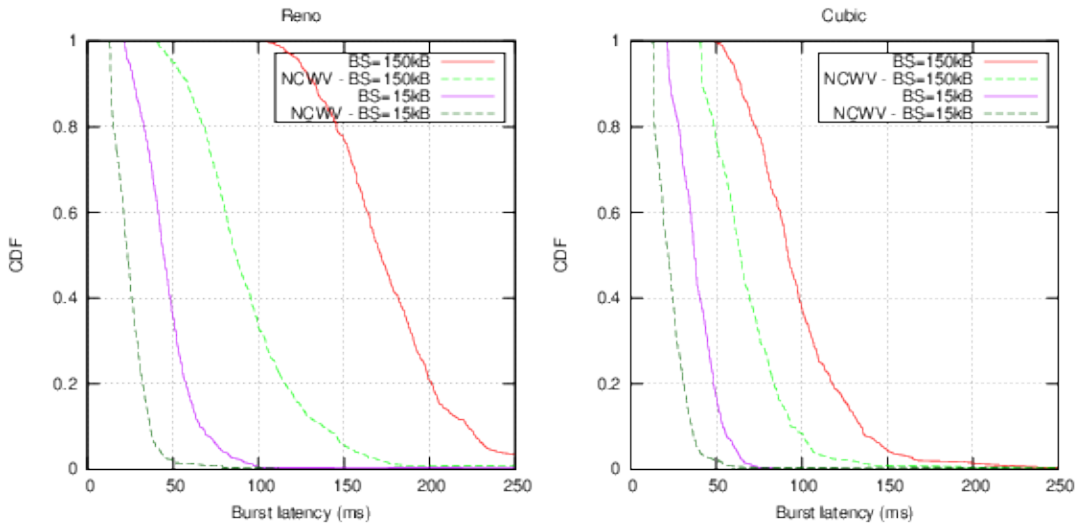


Figure 5.3: Distribution of burst latency with/without new-CWV for bursts of size 15 kB and 150 kB.

and timeouts that would be expected due to tail loss. This is expected to reduce head of line blocking for the application, reducing the experienced latency.

5.3 Performance of new-CWV with curvy-RED

The GNS3-based testbed was used also to evaluate the performance of new-CWV TCP modification with curvy RED under the same setup that was described in the previous section. Three Reno bulk TCP flows generated a background traffic. The foreground traffic was a persistent ON/OFF connection with bursts separated by 0.5 seconds.

The figure 5.3 shows the distribution of burst latency when new-CWV is enabled/disabled with a burst length of 15 kB and 150 kB (consistent with the traffic model used in section 5.2. The left graph refers to a Cubic congestion control behaviour, the right to a Reno congestion control. Although new-CWV was not designed with curvy-RED in mind, its advantages in terms of latency over the legacy TCP are evident.

This result shows an unexpected performance problem that resulted from the implementation of the Cubic algorithm in Linux. The latency for burst traffic using Cubic was always lower than that using Reno congestion control. This is because when Cubic is used with a persistent bursty connection, it does not reset the target cwnd profile at each idle period. Thus, the cwnd target keeps increasing with time and TCP slow-starts from the TCP restart window (RW) to the target every time the connection starts again This behaviour is safe when the idle period is relatively small (as in this case), but with longer idle periods the target value of Cubic may become too large, which can cause substantial capacity overshoot resulting in packet loss. This behaviour has also been pointed out as erroneous in a presentation by Google at the TCPM WG meeting in Yokohama [34] and will be corrected in future realeases of the Linux kernel.

5.4 Discussion

These tests combine RITE-developed transport techniques that seek to reduce network latency, including new AQM schemes (WP2) and new congestion-control mechanisms (WP1) These techniques were developed with different methodologies and sometimes following separate design paths. Thus, a set of integration tests were needed to evaluate how they perform when deployed together. These tests sought to verify that the techniques functioned correctly in an integrated system, and that the combined system still could offer the benefits claimed by independent analysis of each method. The experiments did

not seek to evaluate a wide range of parameters and traffic and hence may not be used to fine-tune or optimise the proposed solutions.

These tests were particularly important to evaluate operation of the new TCP modifications, such as ABE and new-CWV, when running over advanced AQM techniques, such as Dual-Queue. Indeed, these TCP modifications were designed to be compatible with the standard TCP congestion control, while Dual-Queue assumes different congestion control characteristics. Although these two methods have not been co-designed, the end-to-end methods have been designed to be robust, and hence we do not anticipate any unexpected performance degradation when deployed with such new network mechanisms.

Results confirm that the performance of the new TCP modifications, while inferior with respect to the case with more ECN proactive congestion control behaviour (e.g. DCTCP), still show improvements with respect to the current standard. This suggests that the deployment of the new AQM does not need to be constrained by the proposed TCP evolution and vice versa.

5.5 Conclusion

The emulated testbed was used to verify that the techniques correctly functioned in an integrated system for application-limited traffic. This used an integrated kernel that included the RITE TCP mechanisms, and evaluated this using selected tests over the DualQ AQM network method. A benefit was observed usingi new-CWV and ABE. The test cases showed a reduction in the application-layer latency and no tests reported that the methods increased latency. This supports the recommendations to deploy the TCP mechanisms in the Internet.

6 Conclusions

This report addresses core objectives assigned for WP3 and defined by the RITE - Description of Work (DoW) and subsequent web use case amendment. D3.3 details the deployment of RITE mechanisms in use-case trial testbeds, summarising the effect of the developed mechanisms in each of the use-case trials in the industry testbeds. In particular we fulfill the deployment and evaluation of RITE mechanisms in use case trials, which are carried out in tasks 3.3, 3.4 and 3.5, whereas:

- *Task 3.3* addresses use-case trials for online gaming in a Megapop game environment. The Megapop game environment offers a unique opportunity to test RITE mechanisms with real users, while the tested RITE mechanisms are restricted to end system solutions
- *Task 3.4* addresses use case trials for web applications in a BT low-latency testbed. It leveraged Alcatel-Lucent testbed in a single integrated enviroment for these two use-cases.
- *Task 3.5* addresses use-case trials for interactive video in an Alcatel-Lucent testbed. It was built using carrier grade equipment assembled in the same lab environment that customer solutions and deployment scenarios are tested and validated.

In addition to the industry testbed, listed above a virtual testbed was developed at UoA to perform integration tests for key RITE techniques and was used to emulate a network topology similar to the one used by ALU/BT.

The following subsection summarise each of the testbed contribution and fulfillment of the workpackge objectives.

6.1 Online gaming in a Megapop game environment

Megapop has made available a test server that mirrors their production servers in all aspects. The server is used for internal Megapop prototyping and development testing and for the RITE experiments. The experiments run on this testbed addresses the online game server use-case and evaluated the end-host mechanisms that were developed within the project for this use-case. In summary, we have done the following:

- Deployed a customized kernel including the RITE mechanisms on the Megapop test server deployed in Microsoft Azure Cloud.
- Made a traffic pattern analysis of the “Trolls vs. Vikings” game traffic as this was not available in the analysis phase of RITE.
- Evaluated the RITE mechanisms that could have an effect for the “Trolls vs. Vikings” game traffic.
- Made a game server emulator based on the specifications from Megapop on a new realtime game and deployed the emulated game server on the Megapop testbed.
- Evaluated the relevant RITE mechanisms for the emulated real-time game.
- Uncovered interactions between TLP and the Nagle Algorithm as well as between RDB and the Nagle algorithm that will provide input for corrective Linux patches to be submitted to the kernel.
- The results from the evaluation shows that we are able to reduce the retransmission delays significantly by deploying the RITE mechanisms for the gaming scenario. Most of the gain comes from the RDB mechanism at the cost of extra bandwidth.

6.2 Web application low-latency testbed

This testbed address the web application use-case, it was developed and used to assess RITE's innovative mechanisms for broadband applications. BT used the ALU testbed which consists of a classical residential service delivery network composed of xDSL DSLAM (DSL Access Multiplexers), BNG (Broadband Network Gateway), Service Routers (SR) and application servers. More specifically, the testbed tested RITE's DualQ AQM (Active Queue Management) with various mixes of video, web and file downloads. We also assessed the performance with existing AQMs, such as RED since BT's equipment already has RED implemented, but not turned on, it is important to understand whether RED is 'good enough'. This work meets the requirements of RITE's Description of Work (which was amended to be a broadband testbed from the original specialised financial application).

In summary:

- We have shown that RITE's DualQ AQM mechanism in combination with Scalable TCP (DCTCP) delivers a much better quality of experience for video applications using HAS (HTTP Adaptive Streaming) than Classic TCP. The importance is that HAS traffic now dominates the Internet; Netflix accounts for about 35% and YouTube 15% of traffic at peak time [27].
- The DualQ AQM also substantially improves the latency /download time of other typical applications like web browsing and file downloads.
- We showed that DualQ AQM also allows applications that use Scalable TCP (DCTCP) and 'legacy' applications that use Classic TCP to coexist. There is no need to segregate bandwidth, as a flow of either type of application gets a fair share of the bandwidth. The particular importance of this is that it helps deployability: a smooth transition of transport protocols from legacy classis to scalable TCP.
- The DualQ AQM should enable an operator's Quality of Service Model to be substantially rationalised by delivering a better QoS with minimum management requirements. There are some details that need further investigation, including discussion with BT's market facing units for example, whether there needs to be a separate elevated CVLAN, the use of shaping vs Weighted Round Robin and the handling of 'redside' services like WiFi FON.
- We have also compared the DualQ AQM with the existing AQM, RED, and with two emerging AQMs, PIE and FQ-Codel. None gave a comparable performance.

6.3 Alcatel-Lucent interactive video testbed

Alcatel-Lucent designed and built a testbed in the same way as it does for actual customers and internal test and development. It was done in the same lab environment used to test and validate customer solutions and different deployment scenarios. The testbed focused on interactive video application, where we not only took into account network latency, but also the interactive video application itself, i.e. latency due to how video is encoded, packetized and sent to the transport layers. We have fulfilled the interactive video testbed objects defined by the RITE description of work:

- The testbed was built using carrier grade equipment assembled in the same lab environment as customer solutions and deployment scenarios are tested and validated.
- It consisted of a classical residential service delivery network composed of xDSL DSLAM (DSL Access Multiplexers), BNG (Broadband Network Gateway), Service Routers (SR) and application servers.
- We have used the testbed to evaluate the proposed DualQ AQM mechanism and other AQMs in a realistic setting and run repeatable experiments in a controlled environment. In addition, we used several different flavours of TCP transport protocol.

- The testbed is generic enough to be used for testing and validating any novel mechanism, be it end system, network-based or mechanism that combines interaction between both end systems and network elements.
- Two interactive video application were deployed on the testbed, a panoramic interactive video application where user can zoom or pan a panoramic video served from the cloud. And a virtual reality (VR) interactive video application using a Oculus Rift device, were any slight head movement required the remote server to change the video camera view accordingly from video recorded with 360 degree camera view.
- A GUI was developed for evaluation and live demonstration of how different types of network traffic behave under chosen combinations of AQM, link capacity and RTT. In addition to allow easy configuration of the setup, the GUI displayed several measurements in real-time.
- Evaluations of proposed mechanisms by RITE were done using a framework which supported documenting and executing test cases, capturing and processing measurement data, and finally plotting results in different relevant representations.
- Exhaustive tests were carried out on a very comprehensive list of test cases. Tests were automatized and configured to use a variety of different network conditions, different AQMs and end-systems protocols, such as UDP and several different TCP favours.
- We have successfully implemented and evaluated mechanisms that improved latency. This was demonstrated using end-used devices and can be verified in the results from the extensive evaluations and those reported previously.

6.4 Emulated Testbed

In addition to the industry testbed, a virtual testbed was developed at UoA to perform integration tests for key RITE techniques. This testbed was based on GNS3, a graphical environment featuring Cisco's Dynamips emulation engine. It was used to emulate a network topology similar to the one used by ALU/BT. The test cases demonstrated the correct function of implementations of key RITE method and showed a reduction in the application-layer latency when using CWV and ABE in combination with the Dual-Queue AQM scheme. No tests reported an increase in latency using the RITE methods. These results support the recommendations to deploy the TCP mechanisms in the wider Internet.

References

[1] RITE Project, "D1.3 - Report on Prototype Development and Evaluation of End-System, Application Layer- and API Mechanisms."

[2] RITE Project, "D2.3 Report on Prototype Development and Evaluation of Network and Interaction Techniques."

[3] P. Hurtig, A. Brunstrom, A. Petlund, and M. Welzl, "TCP and SCTP RTO restart," Internet Draft draft-ietf-tcpm-rtorestart, work in progress, September 2013. [Online]. Available: http://tools.ietf.org/html/draft-ietf-tcpm-rtorestart

[4] N. Dukkipati, N. Cardwell, Y. Cheng, and M. Mathis, "Tail Loss Probe (TLP): An algorithm for fast recovery of tail losses," Internet Draft draft-dukkipati-tcpm-tcp-loss-probe, work in progress, February 2013. [Online]. Available: http://tools.ietf.org/html/draft-dukkipati-tcpm-tcp-loss-probe

[5] G. Fairhurst, A. Sathiaseelan, and R. Secchi, "Updating TCP to support rate-limited traffic," Internet Draft draft-ietf-tcpm-newcwv-02, work in progress, Jul. 2013.

[6] K. R. Evensen, A. Petlund, C. Griwodz, and P. Halvorsen, "Redundant bundling in tcp to reduce perceived latency for time-dependent thin streams," *IEEE Communications Letters*, vol. 12, no. 4, pp. 334 – 336, 4 2008.

[7] N. Khademi, M. Welzl, G. Armitage, C. Kulatunga, D. Ros, G. Fairhurst, S. Gjessing, and S. Zander, "Alternative Backoff: Achieving low latency and high throughput with ECN and AQM," in *Reducing Latency in Internet Access Links with Mechanisms in Endpoints and within the Network*. University of Oslo, 2015, ch. Paper V, PhD Thesis.

[8] Microsoft, "Microsoft smooth streaming." [Online]. Available: http://www.microsoft.com/silverlight/smoothstreaming/

[9] Microsoft, "Microsoft smooth streaming technical overview." [Online]. Available: http://www.iis.net/learn/media/on-demand-smooth-streaming/smooth-streaming-technical-overview

[10] K. De Schepper, O. Bondarenko, I.-J. Tsang, and B. Briscoe, "'Data Centre to the Home': Ultra-Low Latency for All," in *Under submission*, Jul. 2015.

[11] K. De Schepper, B. Briscoe, O. Bondarenko, and I.-J. Tsang, "DualQ Coupled AQM for Low Latency, Low Loss and Scalable Throughput," Internet Engineering Task Force, Internet Draft draft-briscoe-aqm-dualq-coupled-00, Aug. 2015, (Work in Progress). [Online]. Available: http://datatracker.ietf.org/doc/draft-briscoe-aqm-dualq-coupled

[12] M. Rajiullah, P. Hurtig, A. Brunstrom, A. Petlund, and M. Welzl, "An evaluation of tail loss recovery mechanisms for tcp," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 1, pp. 5–11, Jan. 2015. [Online]. Available: http://doi.acm.org/10.1145/2717646.2717648

[13] B. R. Opstad, J. Markussen, I. Ahmed, A. Petlund, C. Griwodz, and P. Halvorsen, "Latency and Fairness Trade-Off for Thin Streams using Redundant Data Bundling in TCP," in *Proceedings of IEEE LCN*, Clearwater Beach, 2015.

[14] A. Petlund, "Improving latency for interactive, thin-stream applications over reliable transport," Ph.D. dissertation, University of Oslo, Dec. 2009.

[15] M. Allman, K. Avrachenkov, U. Ayesta, J. Blanton, and P. Hurtig, "Early retransmit for TCP and Stream Control Transmission Protocol (SCTP)," RFC 5827 (Experimental), Internet Engineering Task Force, April 2010. [Online]. Available: http://www.ietf.org/rfc/rfc5827.txt

[16] S. Hemminger, "Network Emulation with NetEm," in *Linux Conference*, Australia, 2005, pp. 18–23.

[17] B. Trammell, M. Kühlewind, D. Boppart, I. Learmonth, G. Fairhurst, and R. Scheffenegger, "Enabling Internet-wide Deployment of Explicit Congestion Notification," in *Passive and Active Measurement Conference (PAM)*, New York, New York, USA, Mar. 2015.

[18] BT Wholesale, "Wholesale Broadband Connect - The Future Of Broadband." [Online]. Available: https://www.btwholesale.com/shared/document/Products/Broadband/Wholesale_Broadband_Connect/WBC_Product_Outline_Issue_5_1.pdf

[19] Cisco Systems, "Cisco Visual Networking Index: Forecast and Methodology, 2014-2019." [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html

[20] Bell Labs, "Bell Labs forecasts a 560 percent increase in data traffic on metro networks by 2017." [Online]. Available: https://www.alcatel-lucent.com/press/2013/002957

[21] C. Hollot, V. Misra, D. Towsley, and W.-B. Gong, "A control theoretic analysis of RED," in *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computerand Communications Society (IEEE INFOCOM 2001)*, vol. 3. IEEE, 2001, pp. 1510–1519.

[22] Y. Zhang and L. Qiu, "Understanding the end-to-end performance impact of RED in a heterogeneous environment," Cornell University, Tech. Rep., 2000.

[23] R. Pan, P. Natarajan, C. Piglione, M. S. Prabhu, V. Subramanian, F. Baker, and B. VerSteeg, "PIE: A lightweight control scheme to address the bufferbloat problem," in *Proceedings of the IEEE 14th International Conference on High Performance Switching and Routing (HPSR)*, 2013, pp. 148–155.

[24] 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 — further studies," Presentation at the ICCRG meeting, 86th IETF, Orlando, Mar. 2013. [Online]. Available: http://www.ietf.org/proceedings/86/slides/slides-86-iccrg-5.pdf

[25] N. Khademi, D. Ros, and M. Welzl, "The new AQM kids on the block: An experimental evaluation of CoDel and PIE," in *17th IEEE Global Internet Symposium (IEEE INFOCOM 2014 Workshop)*, Toronto, April 2014, pp. 85–90.

[26] T. Hoeiland-Joergensen, P. McKenney, D. Täht, J. Gettys, and E. Dumazet, "Flowqueue-codel," Internet Draft draft-hoeiland-joergensen-aqm-fq-codel, work in progress, Jun. 2014. [Online]. Available: http://tools.ietf.org/html/draft-hoeiland-joergensen-aqm-fq-codel

[27] Quartz, "Netflix now accounts for nearly 37% of peak web traffic in North America." [Online]. Available: http://qz.com/414271/netflix-now-accounts-for-nearly-37-of-peak-web-traffic-in-north-america/

[28] B. Briscoe, A. Brunström, D. Ros, D. Hayes, A. Petlund, I.-J. Tsang, S. Gjessing, and G. Fairhurst, "A survey of latency reducing techniques and their merits," in *Internet Society Workshop on Reducing Internet Latency*, London, Sep. 2013. [Online]. Available: http://www.internetsociety.org/latency2013

[29] G. White, "Active Queue Management Algorithms for DOCSIS 3.0; A Simulation Study of CoDel, SFQ-CoDel and PIE in DOCSIS 3.0 Networks," CableLabs, Technical Report, Apr. 2013. [Online]. Available: http://www.cablelabs.com/downloads/pubs/Active_Queue_Management_Algorithms_DOCSIS_3_0.pdf

[30] W3C, "Session Control Protocol (SCP)." [Online]. Available: http://www.w3.org/Protocols/HTTP-NG/http-ng-scp.html

[31] iPerf , "iPerf - The network bandwidth measurement tool ." [Online]. Available: https://iperf.fr/

[32] Cisco Systems, "Graphical Network Simulator-3." [Online]. Available: http://www.gns3.com

[33] N. Khademi, M. Welzl, and G. Fairhurst, "IETF tcp alternative backoff with ecn (abe)," Internet Draft draft-khademi-alternativebackoff-ecn-01, work in progress, Sep. 2015. [Online]. Available: https://tools.ietf.org/html/draft-khademi-alternativebackoff-ecn-01

[34] J. Iyengar, "Cubic quiescence: Not so inactive," Presentation at the IETF TCPM meeting, 94th IETF, Yokohama, Nov. 2015. [Online]. Available: https://www.ietf.org/proceedings/94/slides/slides-94-tcpm-8.pdf