European Commission — FP7

Reducing Internet Transport Latency

Project acronym: **RITE**

Project number: 317700

Work package: Network and interaction

Deliverable number and name:

D2.1: Network systems analysis and preliminary development report

Title: Network systems analysis and preliminary development report
Work Package: WP2

Version: 1
Date: November 1. 2013
Pages: 30

Author:
David Ros

Co-Author(s):
Iffat Ahmed, Amadou Bagayoko, Anna Brunstrom, Gorry Fairhurst, Carsten Griwodz, David Hayes, Naeem Khademi, Andreas Petlund, Ing-Jyh Tsang, Michael Welzl

To:
Rüdiger Martin
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:
Report describing the outcome of the network systems analysis, corresponding to Task 2.1. The most promising avenues for further investigation and prototype development are identified. The report also includes initial results from simulations and prototype developments in Task 2.2 and Task 2.3.

Abbreviations

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

ABR Available Bit Rate

DSL Digital subscriber line

ADU Application Data Unit

AF Assured Forwarding

API Application Programming Interface

AQM Active Queue Management

ATM Asynchronous Transfer Mode

BE Best Effort

BER Bit Error Rate

BM Burst Mode

BNG Broadband Network Gateway

BRAS Broadband Remote Application Server

CAIDA Cooperative Association for Internet Data Analysis

CBR Constant Bit Rate

CC Congestion Control

CDN Content Delivery Network

CN Core Network

CMTS Cable Modem Termination System

CoS Class of Service

CPE Customer Premise Equipment

DASH Dynamic Adaptive Streaming over HTTP

DBA Dynamic Bandwidth Allocation

DBG Downstream Bonding Group

DiffServ Differentiated Services

DMT Discrete Multitone

DOCSIS Data Over Cable Service Interface Specification

DCF Distributed Coordination Function

DS Differentiated Services (DiffServ)

DSCP Differentiated Services Code Point

EF Expedited Forwarding

EFM Ethernet in the First Mile

ETP Experimental Transport Processing

FDD Frequency Division Duplex

FEC Forward Error Correction

FIFO First-In First-Out

FTP File Transfer Protocol

GB Gigabyte

HAS HTTP Adaptive Streaming

HFC Hybrid Fiber-Coaxial

HIS High Speed Internet

HTTP HyperText Transfer Protocol

IETF Internet Engineering Task Force

INP Impulse Noise Protection

IntServ Integrated Services

IP Internet Protocol

IPoE IP over Ethernet

IPTV IP Television

ISAM Intelligent Services Access Manager

ISDN Integrated Services Digital Network

ISP Internet Service Provider

LAN Local Area Network

MAC Medium Access Control

MB Megabyte

MCM Multi-Carrier Modulation

MD5 Message Digest 5

MPEG Motion Pictures Expert Group

MPLS Multi-Protocol Label Switching

MSS Maximum Segment Size

MTU Maximum Transfer Unit

NC Network Control

NRT Non Real-Time

NS-2 Network Simulator 2

OFDMA Orthogonal Frequency Division Multiple Access

OLT Optical Line Terminal

ONU Optical Network Unit

P2P Peer-to-Peer

PCAP Packet Capture

PE Provider Edge

PHB Per Hop Behaviour

PI Proportional Integral control

PMD Physical Media Dependent

PMTU Path MTU

POTS Plain Old Telephony Service

PPP Point-to-Point Protocol

PPPoE PPP over Ethernet

PTMP Point-to-MultiPoint

PTP Point-to-Point

QAM Quadrature Amplitude Modulation

QoS Quality of Service

QPSK Quadrature Phase Shift Keying

RED Random Early Drop

REIN Repetitive Electrical Impulse Noise

RITE Reducing Internet Transport Latency End-to-End

RAN Radio Access Network

RS Reed Solomon

RT Real-Time

RTT Round Trip Time

PCF Point Coordination Function

PEIN Prolonged Electrical Impulsive Noise

SAP Service Access Point

SCDMA Synchronous Code Division Multiple Access

SCM Single-Carrier Modulation

SFQ Stochastic Fair Queuing

SHINE Singular High Impulse Noise Environment

SLA Service Level Agreement

STB Set Top Box

TCP Transmission Control Protocol

TD Tail Drop

TDD Time Division Duplex

TDMA Time Division Multiple Access

ToS IP Type of Service

UDP User Datagram Protocol

VBR Variable Bit Rate

VoD Video on Demand

VoIP Voice over IP

VLAN Virtual LAN

VPLS Virtual Private LAN Service

VPRN Virtual Private Routed Network

WAN Wide Area Network

WDM Wavelength Division Multiplexing

WFQ Weighted Fair Queueing

WG Working Group

WRED Weighted Random Early Drop

WRR Weighted Round Robin

WP Work Package

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

1 Introduction

Excessive buffering in the network is one of the main contributors to high end-to-end latency. Efforts such as the Bufferbloat projects [1] have uncovered, and increased awareness of, the presence of oversized buffers along many Internet paths. Given the way in which TCP congestion control (CC) works, and given that many gateways use simplistic queue-management policies, such buffers often end up being fully filled and yield large queuing delays. This problem is compounded by the way in which flows, both with and without CC, interact in network buffers. On the other hand, buffering *is* necessary in packet-switched networks to cope with temporary bursts of traffic. In spite of a large volume of research work in the area, finding both the “right” buffer size and the “right” queue management method is still a non-trivial problem with many open questions.

In principle, a long-lived, greedy flow[1] using TCP will *always* manage to fill a bottleneck's buffer—that is, unless some kind of proactive mechanism is adopted to anticipate and avoid buffer saturation as much as possible. Active Queue Management (AQM) is the generic term for a set of algorithms that aim at preventing buffers from being persistently full. Coupled to a suitable form of AQM, Explicit Congestion Notification (ECN) is a standard protocol mechanism to further improve congestion avoidance, by carrying early congestion signals (marks) to senders. In theory, both AQM and ECN may help in improving user-experienced performance for many applications—and, especially, in improving latency—, but their adoption and deployment have been limited, for reasons ranging from configuration issues (AQM) to lack of incentives (ECN).

1.1 Analysis breakdown and document structure

This document presents a summary of the preliminary analysis phase in Work Package 2 of the RITE project. The main focus of this report is on the issues arising from the interaction between application flows and network buffers.

Following are key contributions of this document:

- We have performed a thorough survey of the state-of-the-art for the topics under investigation. This survey will be published as a journal paper, but some key findings related to Work Package 2 are summarised in this report.
- We have done preliminary work for the most promising avenues of research resulting in a detailed map of the topics that should be prioritised for the rest of the RITE project.

More specifically, this document presents:

- An analysis of root causes of delay in broadband access and aggregation networks (§ 2) and their possible impact on transport latency, including an experimental assessment of excess buffering in cellular networks (§ 3.2), to identify where the efforts of the RITE project will have the greatest impact.
- A summary of RITE's ongoing efforts at producing a set of recommendations on network buffering and Active Queue Management, to be published through the IETF (§ 2.3).
- An analysis of the impact that latency-insensitive, congestion-controlled flows may have on latency-sensitive ones, due to their interaction in network buffers. The former include both standard TCP flows (§ 3.1) and flows using LEDBAT congestion control (§ 3.3), which is supposed to do no harm in terms of delay.
- An assessment of the performance of several key AQM mechanisms designed for low latency, using real-world implementations (§ 4.1). To the best of our knowledge, this is the first such study that has

[1] That is, a flow whose rate is limited neither by the application, nor by flow control at the receiving end, and that is always able to fully utilise any available capacity.

been performed of algorithms such as CoDel and PIE. Related to this, this document summarises an ongoing, systematic effort at designing AQM test scenarios, coupled with an exhaustive simulation study; the final goal of this activity is producing a set of evaluation guidelines that could feed the work of the AQM Working Group at the IETF.

- A proposal for improving ECN so as to incentivize its usage on the Internet (§ 4.2.1). Given the renewed interest on AQM in the IETF community, we believe that the time is right for pushing for wide ECN deployment, but for this to happen some issues with ECN have to be fixed.

This deliverable is closely related to deliverable 1.1 ("End-system analysis and preliminary development report") that is submitted at the same time. Whereas this deliverable addresses work package 2 and has a focus on network mechanisms and interactions between end-hosts and the network, deliverable 1.1 centers on end-host mechanisms. Some of the subprojects include some elements that belong to work package 2 and other elements belonging to work package 1. To avoid redundancy, we have chosen to describe each subproject in only one of the deliverables. We have tried to make it clear from the text how different elements relate to the research areas of the different work packages. The documents should therefore be read in conjunction. Deliverable 3.1 ("Traffic pattern analysis and data set acquisition report") [2] may also be a useful reference when it comes to questions of trace usage and testbeds for RITE subprojects.

The remainder of this report is organised as follows. Section 2 looks at root causes of delay in access networks, and in particular at issues related to the management and dimensioning of buffers. Section 3 focuses on how the performance of delay-sensitive flows may be degraded, due to their coexistence in buffers with flows that use commonly-deployed congestion control schemes such as TCP and LEDBAT. AQM mechanisms, and problems related to ECN, are discussed in Section 4. A summary of the main results of our analysis is given in Section 5. Finally, Section 6 concludes this report.

2 Analysis of the causes of delay in broadband access and aggregation networks

The broadband access, or last mile as it is known by the telecommunications industry, encompasses the network infrastructure responsible for delivering connectivity to the end users. The last mile interlinks the home user(s) and aggregation network, which lies within the operators infrastructure, interconnecting their core and peering points, hence in a simplistic way forming the Internet. This section analyses two elements of any network equipment that can act as sources of delay, i.e. the transmission technology and the mechanisms to deal with network congestion and traffic prioritisation.The access network can be divided according to the physical layer technology it employs, i.e. fiber, copper, cable, or wireless (3G/4G). Each of these technologies leads to a different delay due to the fundamental characteristics of the physical medium. These sources of delay are also present in the home or aggregation network, e.g. WiFi or microwave transmission. Mechanisms to minimize transmission propagation delay will be discussed in section 2.1. In addition, section 2.2 details the most common mechanisms to address network congestion and traffic management, which are implemented and deployed in different kinds of network equipments. This section will mostly focus on known and/or deployed technologies, which are commonly employed in current network infrastructures.

2.1 Transmission delay due to physical and link layer mechanisms

Digital data transmission is fundamentally limited by two elements due the physical medium, the propagation delay and the bit error rate (BER). Propagation delay is intrinsic to the speed electromagnetic waves travel in the medium. For instance, electromagnetic waves travel at about 30 cm per nanosecond in the air, which is slightly slower than the speed of light in vacuum. In guided transmission media the propagation speed is slower, around 20 cm per nanosecond, and slower in fiber than in copper. There are several options to optimise the medium or the utilisation of the medium, such as use of hollow fiber or straighter cable paths. However, in practice once a medium is chosen there is not much that can be done to reduce the propagation delay except replacing the medium.

Bit errors are altered bits over the transmission channel caused by noise, attenuation, distortion, interference, fading/shadowing or bit synchronization issues. In general, perturbations due to noise are much more prominent in electromagnetic transmission than in photonic (fiber) transmission, since photons don't interact with each other. Several mechanisms at the link layer are designed to address bit errors, with respect to both, error detection and possible correction. Protocols at link layers vary according to medium and technology used, as such different protocols have different performance and associated intrinsic transport delay. The remainder of this section will address the different technologies, focusing on two aspects of latency irrespective of the technology: the intrinsic delay due to the link protocols and the inbuilt mechanisms to address bit error rate, as a way to decrease packet loss that eventually leads to end-to-end transport delay.

2.1.1 Digital subscriber line (DSL)

DSL is one of the most common access technologies deployed in the world. The evolution and improvement in the standard for copper access from ADSL (ITU-T G.992.1, G.992.2) [3, 4], ADSL2 (ITU-T G.992.3, G.992.4) [5, 6], ADSL2+ (ITU-T G.992.5) [7] to VDSL (ITU-T G.993.1) [8], VDSL2 (ITU-T G.993.2) [9] and Vectoring (ITU-T G.993.5) [10] has allowed a steady increase of bandwidth at the last mile. DSL is a point-to-point (PTP) technology that is still evolving; for example at the transmission level, frequency division duplex (FDD) has been used, but in the upcoming standard G.fast (ITU-T G.9700) time division duplex (TDD) has been adopted. In the past, both single-carrier modulation, mostly in the form of quadrature amplitude modulation (QAM), and multi-carrier modulation, mostly in the form of discrete multitone (DMT), have been used in the various DSL flavours, but at present DMT is the most adopted due to its performance. In the same token, data link layer has changed from ATM to Ethernet, specifically the IEEE 802.3ah Ethernet in the first mile (EFM) standard [11] has been adopted since the G.SHDSL.bis (ITU-T G.992.1) [3] and VDSL (ITU-T G.993.1) [8] specifications. The transmission and data link techniques have driven the increase of throughput in copper access. Complementarily mechanisms to deal with noise and bit errors have enabled not only higher bandwidth, but have also had a strong impact on latency as there is often a tradeoff between error correction capabilities and added delay.

Two important types of noise considered on copper transmission are crosstalk and impulse noise. Mechanisms to deal with crosstalk have lead to dramatic improvements in bandwidth, albeit at shorter distances, culminating in the latest Vectoring technology. Moreover mechanisms to reduce the impact of impulse noise have direct effects on latency, as any decrease in packet loss improves the end-to-end data transport, specially when using TCP as the transport protocol. There are three categories of impulse noise, which can be divided by time duration [12]: Repetitive electrical impulse noise (REIN) has a duration of less then 1ms; prolonged electrical impulsive noise (PEIN) with a duration between 1 ms and 10 ms; and singular high impulse noise environment (SHINE) with a duration larger than 10 ms. REIN is noise produced by regular radio interference causing destructive interference on the DSL modulated signal. At the data link level this is reflected as CRC errors leading to frequent packet drops. PEIN is longer duration noise, which will also manifest itself as CRC errors with higher rates of packet drop than REIN. SHINE is short peak interference that causes burst transmission errors.

There are various strategies to address the different types of impulse noise. Forward error correction (FEC) is best suited for REIN or PEIN, as it incurs a fixed overhead. FEC is usually implemented using Reed-Solomon coupling with interleaving (I-FEC), creating a robust scheme applied to the DMT symbols. Figure 2.1 depicts how this scheme works. It is straightforward to notice that there is an intrinsic trade off between choosing I-FEC parameters which would create a robuster scheme with higher overhead, and lower overhead but weaker scheme. These choices have a direct impact on the delay created at the DSL transmission layer. A typical profile configuration in DSL deployment are for example: a default profile with overhead protection parameters that give 8 ms delay in downstream and no protection, i.e. fast mode, in upstream; and a repair profile with overhead protection yielding 16 ms in downstream and 4 ms delay in upstream.

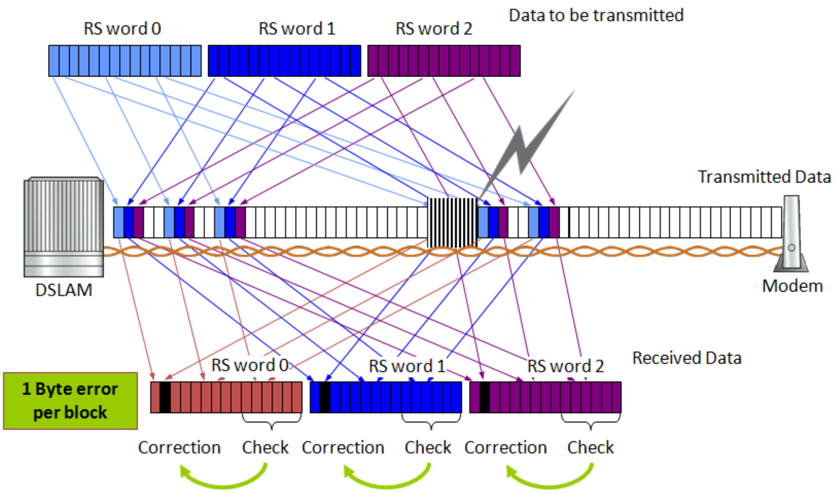


Figure 2.1: Schematic Illustration of Interleaving-FEC

Another impulse noise protection method is to use retransmission at the physical layer, also known as Automatic Repeat Request (ARQ). In DSL this has being standardized separately by the G.INP (ITU-T G.998.4) [13], which is on top of ADSL2, ADSL2+, VDSL2 and Vectoring standards. ARQ is more suited to SHINE type of noise, since a higher level of errors would require many more retransmissions. The advantage is that the overhead of retransmitted packets is only generated when an impulse noise has occurred. In general, it will impose lower DSL transmission delays, in the order of 4 ms. ARQ can also handle REIN, but as indicated this may lead to higher retransmission overhead and delays.

2.1.2 Cable

Cable networks offer video and IP service using Hybrid Fiber-Coaxial (HFC) networks, whereas the last mile is based on coaxial cable to the house. Cable transmission is standardized by Data Over Cable Service Interface Specification (DOCSIS), with the most recent release being DOCSIS 3.0 [14] and ratified by ITU-T Recommendation J.222 [15, 16, 17]. Like DSL transmission, it also uses QAM, more specifically Quadrature Phase Shift Keying (QPSK) as modulation scheme, thus the DOCSIS standard also applies the I-FEC impulse noise protection technique. However, unlike DSL, cable is intrinsically a point-to-multipoint (PTMP) technology, thus part of the latency in a cable system is inbuilt by the design of the physical layer so as to optimize signal transmission in a shared medium between the cable modem termination system (CMTS) and the cable modems (CMs) at the customer premises. The physical media dependent sublayer uses both time division multiple access (TDMA) and synchronous code division multiple access (SCDMA) modes. DOCSIS 3.0 introduces channel bonding, which is a logical process that combines the data packets received on multiple independent channels into one higher-speed data stream. Channel bonding can be implemented independently on upstream channels or downstream channels. The CMTS can dynamically balance the data across the downstream bonding group (DBG). Each outgoing packet from the CMTS is tagged with a sequence number. Thus, packets can be sent across different downstream channels and can have different time delays in arriving, which requires re-synchronization by the modem. On the upstream there is a mechanism to allocate mini-slots to a CM, which is communicated downstream as a allocation map (MAP). Using TDMA means that upstream transmissions are burst in nature. A given RF channel is shared by multiple CMs via the dynamic assignment of time slots. SCDMA makes sure that multiple CMs can transmit simultaneously on the same RF channel and during the same TDMA time slot, as the signals are separated by different orthogonal codes. The Request-and-Grant cycle between the CMTS and the CM can only take advantage of every other MAP at most, depending upon the round trip time (RTT), the length of the MAP, and the MAP advance time. This back-and-forth communication, which is intrinsic to the protocol, produces an intrinsic latency. In addition, the modem may miss a MAP allocation, when it is waiting for a grant from its last request. Missing mini-slots allocation results in delays of a couple of milliseconds, which will ultimately impact the end-to-end data transport.

2.1.3 Fiber

Fiber access is considered the ultimate broadband access service technology. Photonic transmission can achieve very high bandwidth, since it does not suffer from interference or noise, but it is not immune to attenuation, dispersion or bit synchronization issues. This technology has long been deployed in metro and core interlinks, with well defined Medium Access Control (MAC) layer to deal with transmission errors, whereas FEC is still the fundamental underlying technology used for error correction. In the last mile, there are both PTP and PTMP fiber access technologies. PTP fiber basically uses IEEE 802.3 [18] technologies such as 100BASE (Fast Ethernet) or 1000BASE (gigabit Ethernet or GbE). Alternatively, PTMP fiber is based on passive optical network (PON) technology. There are competing standards on PONs, by IEEE as part of the IEEE 802.3 standard, i.e. the 1G-EPON, 10G-EPON (IEEE 802.3av) [19] and by ITU-T GPON (ITU-T G.984) [20, 21, 22, 23], 10G-PON (ITU-T G.987) [24, 25, 26]. In both systems, they assume a shared medium and used time division multiplexing (TDM), time division multiple access (TDMA) and wavelength division multiplexing (WDM) technologies to achieve high throughput transmission. In any TDMA system, independently of the medium, i.e. cable, fiber or wireless, protocols are needed between the physical and data link layer to allocate time slots for accessing the shared medium. Such mechanisms intrinsically increase the complexity of the system and incur delays. In GPON (ITU-T G.984) the downstream shares the medium using a TDM mechanism, i.e. the optical line terminal (OLT) broadcasts the information in the fiber, but only the optical network unit (ONU) that is allowed to read the data has access to it, which is protected by an encryption mechanism. In the upstream, the different distances between the ONUs and OLT require a burst mode (BM) transmission, whereas each ONU will send their signal with a certain power level, and the OLT has to adjust to the power levels of each of the TDMA slots allocated to the ONU to transmit. The standard determines a mechanism to dynamically allocate time slots for the upstream, which can be done according to service requirements. This dynamic bandwidth allocation (DBA) can incur a delay of up to 30ms when not efficiently done [27]. The OLT is responsible of creating and informing the the ONUs of the bandwidth allocation map. This is part of the complexity incurred by a shared medium mechanism, which needs to be properly designed for avoiding unnecessary latency.

2.1.4 Wireless

A wireless link is a shared-medium environment, where the MAC layer provides the fundamental mechanisms to arbitrate access to the physical medium. Media access can be achieved either by contention or by request/allocation. Both approaches introduce a delay that typically increases as the link becomes loaded and systems compete for limited resources. This section discusses channel access delay in two different wireless networks, i.e., WiFi and 4G LTE networks. The queuing delay caused by network congestion or buffering will be dealt with in Section 2.2.

The IEEE 802.11 MAC protocol [28] has gained widespread popularity and become the *de facto* layer-2 standard for wireless local area networks (WLAN). It specifies two channel access mechanisms: point coordination function (PCF) and distributed coordination function (DCF). The PCF mode uses centralized scheduling and polling for medium access, while DCF employs a random access scheme based on the carrier-sense multiple-access with collision avoidance (CSMA/CA) technique. Compared to the DCF mode, the contention-free scheme in PCF is able to provide strict quality of service and delay guarantees, hence, it is expected to better support real-time applications (e.g., voice, interactive video). Some studies [29] [30] have analyzed the expected latency of a node in the PCF mode as a function of packet sizes, polling rates, channel rates and the power management strategy that a node may employ (e.g., active or power save mode). In DCF, delays caused by both channel access (due to other nodes transmitting) and by collisions (due to random accessing) need to be considered [31] [32] [33]. However, it has been shown in [34] that although DCF provides satisfactory performance in scenarios with a limited number of users, it fails to provide the requirements of delay sensitive applications in more crowded scenarios.

Developed by 3GPP, the 4G LTE network, based on OFDMA (Orthogonal Frequency Division Multiple

Access), is able to provide higher peak data rates and lower latency compared to other 3GPP technologies, e.g., EDGE, HSDPA, HSDPA+. The LTE mobile network addresses latency from two aspects: the control plane (C-Plane) and the user plane (U-Plane). The C-Plane deals with signaling and control functions, and the U-plane deals with actual user data transmission. The U-Plane latency is defined as the one-way transmit time between a packet being available at the IP layer in the user terminal/base station, and the availability of this packet at the IP layer in the base station/user terminal.

The C-Plane latency takes into account RAN (Radio Access Network) and CN (Core Network) latencies in unloaded conditions. Compared to Release 8, the overall C-Plane latency of LTE-Advanced in Release 11 [35] decreases significantly. The target of transition time from IDLE to CONNECT mode is expected to be less than 50 ms, including the establishment of the U-Plane (excluding the transfer latency on the S1 interface). A transition time of less than 10 ms should be achieved from the DORMANT state to the ACTIVE state in the CONNECT mode. LTE-Advanced systems are able to achieve a U-Plane latency of less than 10 ms in unloaded conditions (i.e., a single user with a single data stream without scheduling delay) for small IP packets for both downlink and uplink.

Figure 2.2 presents the latency for different commercially deployed mobile technologies, measured between the subscriber unit and a node immediately external to the wireless network, i.e., without considering Internet end-to-end latency. The latency requirements in the above analysis and in Fig. 2.2 serve as a lower bound in LTE networks. Delay values in loaded scenarios might be significantly higher, particularly for low priority traffic, due to dominant queuing and scheduling delays.

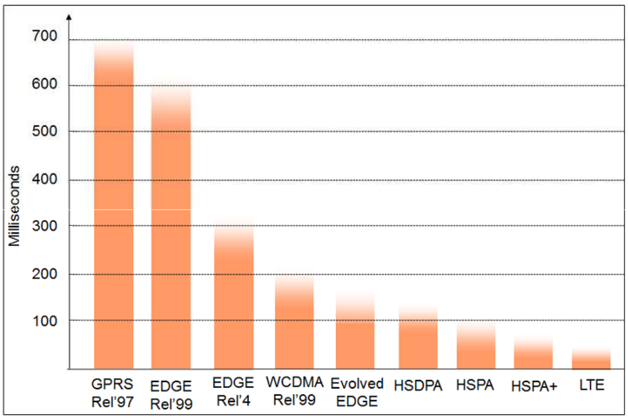


Figure 2.2: Latency of different technologies (Source: 4G Americas member companies).

2.2 Traffic prioritisation and queue management

An essential component of any network equipment is how it handles quality of service (QoS). The standard mechanism for class prioritisation is Diffserv [36, 37], which separates the traffic flows according to classes. In practice, packets of different classes will be serviced in different queues. Both the scheduler model and the class queue management model have an impact in terms of latency. In section 4.3.2 of [2] we have seen that packets follow the classical prioritisation, marking/re-marking, policing, queuing, scheduling and shaping approach. For some access equipment a four-queue model is common, whereas a full eight-queue model is more common in high throughput access nodes such as in PON systems. In residential networks, a common approach is to use four classes of service, i.e. voice, video (IPTV), controlled load (network management traffic) and best effort, in which a packet's p-bit marking is mapped to one of the four queues. Some network operators use even just two classes of service, while others might use more classes. In any case, scheduling assumes a strict priority for the higher priority queues, whereas the lower priorities are scheduled using weighted round robin (WRR) or weighted fair queuing (WFQ). In practice, this implies that traffic of high priority will have an impact on the latency suffered by traffic of lower priorities. For example, an IPTV video service might have impact on the end-user best effort traffic, specially if the video class is serving multiple high definition streams and the last mile physical link has a limited bandwidth. Putting it in another way, the egress rate of the best effort queue is impacted by the higher priority scheduling, causing the best effort queue to build up, creating a latency due to priority scheduling.

The standard mechanisms of queue management are widely implemented and deployed in the access and aggregation networks. In general, tail drop (TD), random early drop (RED), weighted-RED (WRED) and other RED variants are used, whereas the exact configuration as of queue size, weights and drop probability depends on the queue service classes. Moreover, this information is not made public by operators and configurations are optimised according to each operator's network infrastructure and service offerings. For example, it is common that IPTV service queues are dimensioned with a substantial buffer length as compared with those used for best effort service. IPTV traffic consists mostly of multicast video packets. It is more important in terms of QoS that packets are not dropped as opposed to incur more delay, even for live broadcasted events. Queue management strategies vary according to network node, home device, access network, aggregation and core equipment. Each of these nodes have different queue configurations as they have different requirements, due for example to equipment hardware capabilities, level of traffic aggregation and available bandwidth. The overall priority class and queue management that a packet is subject to throughout the network have a strong impact on the end-to-end delay experienced by a flow.

2.3 Recommendations on network buffering and Active Queue Management

RITE partners B. Briscoe (BT) and G. Fairhurst (University of Aberdeen) have worked with input provided by J. Gettys (on the RITE steering board) to write an IETF Individual draft entitled "Advice on network buffering" [38]. This was motivated by the need to update the advice given in section 13 of RFC 3819 [39], which gave guidance to subnet designers on the use and sizing of buffers. At the time when RFC3819 was written, much networking equipment had insufficient buffering to accommodate the size of bursts produced by TCP stacks, it therefore motivated the need for larger buffers as well as suggesting use of queue management. Subsequent research has altered understanding of buffer sizing and led to updated recommendations on queue management.

The "Advice on network buffering" draft was published in March 2013 and proposed to significantly revise the previous recommendations on buffering with advice that would apply to all packet buffers, whether in network equipment, end hosts or middleboxes (such as firewalls or NATs). It was presented at the IRTF Internet Congestion Control Research Group (ICCRG[2]) (IETF-86, Orlando) as an input to the process of understanding the need for standards in this area.

G. Fairhurst joined F. Baker to edit "IETF Recommendations Regarding Active Queue Management", which has been adopted to be published as a Best Common Practice (BCP) document [40] within the IETF Active Queue Management (AQM) working group[3]. Earlier research, presented in [41] has been widely cited (over 28 RFCs reference this[4]), but there was a need to update the recommendations after fifteen years of experience and new research. This draft has been presented at the IRTF ICCRG (IETF-86, Orlando) and was also one of the documents at the Birds of a Feather (BoF) that formed the AQM working group at IETF-87 (Berlin), and will become the first WG document in October 2013, and will be presented at IETF-88 (Vancouver). It is scheduled for publication in January 2014.

The document seeks to define recommendations to the Internet community concerning measures to improve and preserve Internet performance, specifically with a focus on reducing the Internet latency due to buffering within routers. The goal is to present a strong recommendation for testing, standardization and widespread deployment of AQM in network devices, to improve the performance of today's Internet. It will be the first IETF document to provide such advice, and as such will be a useful tool in encouraging network operators to deploy mechanisms that can control network queuing latency.

Although the document does not recommend any particular AQM method, it does make BCP recommendations that affect the choice of procedure that will be standardised by the IETF. The document urges

[2] http://irtf.org/iccrg

[3] http://tools.ietf.org/wg/aqm/charters

[4] http://www.arkko.com/tools/allstats/citations-rfc2309.html

a concerted effort of research, and measurement, to define suitable mechanisms. It also motivates the use of Explicit Congestion Notification (ECN) as a method to reduce latency and provide more feedback information to senders. There is additional work that is needed here (e.g., [42]), but the opportunity to deploy and use ECN has never been greater than when widespread deployment of AQM is being sought.

The "IETF Recommendations Regarding Active Queue Management" also express concerns about the potential future congestive collapse of the Internet due to flows that are unresponsive, or not sufficiently responsive, to congestion indications. It acknowledges that there is no current consensus solution to controlling congestion caused by such aggressive flows (known in RITE as "collateral damage"); significant research and engineering will be required before any solution will be available. It is imperative that this work be energetically pursued, to ensure the future stability of the Internet.

The research topics identified in the draft are core research activities within the RITE project, and the project therefore plans not only to "encourage" research, but also to contribute findings and techniques that can be used to build the next generation of standards.

3 Interaction of deployed congestion-control algorithms with latency-sensitive flows in network buffers

The extent of buffer buildup is determined by the rate of incoming packets versus the rate of outgoing packets. Standard TCP congestion control probes the available bandwidth by injecting packets into the network until there is packet loss, which for tail-drop queuing happens when buffers are full. The way buffers fill up is thus highly dependent on the transport protocol behavior. In this section we explore the interaction between deployed congestion-control algorithms and latency-sensitive flows in shared network buffers.

3.1 Multimedia-unfriendly TCP congestion control and home gateway queue management

A previous study coauthored by RITE partners [43] highlights issues multimedia traffic can have when sharing with TCP traffic through Home Gateway (HG) access equipment. HG access equipment is widely used and has particular characteristics:

- it is often the bottleneck in communications,
- only a small number of concurrent flows share a limited capacity,
- it often has large buffers (see section 2).

TCP tries to fill the bottleneck buffer which causes the multimedia traffic to experience high latency and loss. [43] find that more aggressive versions of TCP, such as CUBIC, exacerbate the problem, as do minor differences in the buffer management packet dropping policies.

Currently, the type and details of buffer management mechanisms that are deployed at the congestion points of networks are not always publicly known. This work aims to investigate this using different network measurement techniques. Current progress on this and future plans are described in section 3.1.1.

In addition we plan to investigate the impact of TCP on multimedia flows in the mobile network environment. Mobile communication has the same characteristics as wired HG access equipment outlined above, and specific issues to do with mobility and the wireless interface. Section 3.2 outlines our current progress and future plans.

With the knowledge gained from the work outlined in sections 3.1.1 and 3.2, we hope to be able to model the network characteristics that we observe and recommend network buffer sizes and buffer management techniques that will improve the latency of multimedia flows and how TCP interacts with them.

3.1.1 Detecting the queuing/dropping scheme at the bottleneck

We have seen in the previous sections that the bottleneck queue drop scheme can have a significant impact on the performance of multimedia traffic. To find out what type of drop/-queue scheme is deployed in HG bottlenecks, we are developing an active measurement tool. Using our knowledge of loss and latency patterns in different queue environments, we aim to be able to detect the drop-/queue scheme on the bottleneck of the connection that is measured.

For our current experimental setup, we utilise an experimental test-bed including 2 hosts and a router. Generating one thin-stream and one greedy stream from the sender to the receiver, we log all the packets going in and out to them (Figure 3.1). At the end of a test, we have two different log files on the sender and the receiver. Then, we transfer the receiver log file to the sender machine in order to be able to analyse both log files and use statistical analysis to generate the results.

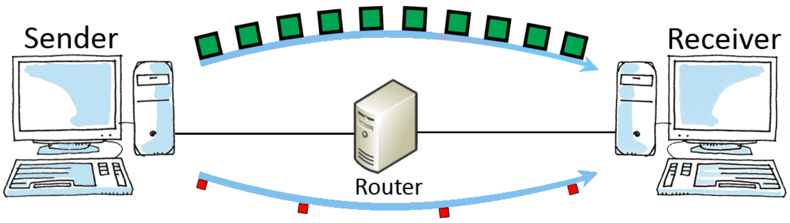


Figure 3.1: Test setup for the initial phase

The intended end-product is a tool that can determine the buffer management strategy that is used on the bottleneck router within a certain probability, based on executing one or more black-box tests between a sender and a receiver machine. For the initial phase, we will limit the scope to detect the drop schemes described in [43]. For more advanced analysis, we hope to extend the tool for the detection of more advanced drop schemes like AQMs.

3.2 Excess buffering in cellular access networks

Cellular networks are becoming an increasingly important Internet access technology. To accommodate varying data rates over time-varying wireless channels they are also normally provisioned with large buffers [44, 45]. The fact that cellular networks typically employ individual buffer space for each user [45, 46] in combination with a low level of user multitasking over cellular connections has in the past limited the impact of these buffers on user performance. However, with the emergence of more and more powerful smartphones, as well as the increasing use of cellular broadband connections for residential Internet access, multitasking over cellular connections is today becoming common. This makes bufferbloat in cellular networks an increasingly important problem. The recent study by Jiang et. al. [47] also confirm that bufferbloat can lead to round trip times (RTTs) on the order of seconds for cellular networks.

As discussed above, the way buffers fill up are highly dependent on the transport protocol behavior and varies between different TCP congestion control algorithms. To examine the interaction between TCP congestion control and bufferbloat in 3G/4G cellular networks we have performed a measurement study in the 3G (UMTS), 3.5G (HSPA+) and 4G (LTE) networks of one of the leading commercial providers in Sweden, involving more than 1800 individual measurements. In our measurements we study how the response time of a Web transfer is affected by varying levels of competing background traffic and how the congestion control algorithms used affect performance. Three congestion control algorithms are considered: TCP NewReno, TCP CUBIC [48] and TCP Westwood+ [49].

Our results indicate that the 3G and 3.5G networks suffer from severe bufferbloat. When background traffic is introduced the Web response time sometimes increases by more than 500%. Furthermore, the congestion control algorithm used for the background flow has a significant impact on the Web response time. The more aggressive congestion control used by CUBIC roughly doubles the Web response time in comparison to Westwood+. For low bandwidths (i.e. 3G) the congestion control version used by the Web flow also has a significant impact on performance. In the studied 4G network, bufferbloat is less of a problem. Full details on our results are available in [50]. Below, we briefly illustrate our results for the 3.5G network.

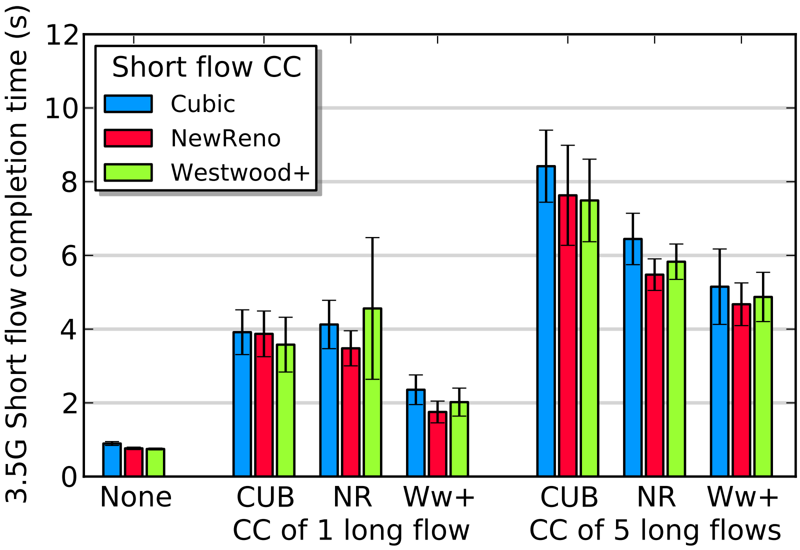


Figure 3.2: The average Web response times of short flows for different background loads, over 3.5G.

The impact of varying levels of background traffic on the completion time of a Web flow is illustrated in Figure 3.2, for the different congestion control schemes. The average Web response time and the 95% confidence interval of 30 repeated short flows are displayed for each configuration. Leftmost in the figure, the first three bars show the average Web response times without any background traffic. Second, the middle group of nine bars shows the Web response times of the short flows with one background flow, and all combinations of the three congestion control algorithms. Third, the right group of nine bars also shows the Web response times but for five background flows, for all combinations of congestion controls.

The graphs in the figure illustrate that the 3.5G network in our measurements is indeed prone to excessive buffering. The introduction of more background flows has a severe impact on the short flows and results in higher Web response times. Additionally, the congestion control algorithm used for the background flows clearly affects the Web response time, where CUBIC background flows result in significantly higher Web response times compared to Westwood+ background flows. Varying the congestion control for the short flow does, however, not result in any significant differences in response time for the 3.5G network in our measurements.

We are currently performing additional measurements, considering also the impact of various types of competing traffic on VoIP performance.

3.3 Issues with delay-based congestion control

Delay-based congestion control (DBCC) uses end-to-end delays as a signal to adapt a sender's transmission rate. DBCC attempts at reducing latency by controlling the size of queues in network packet buffers.

Most DBCC mechanisms work by measuring either round-trip times (RTT) or one-way delays (OWD) at senders, with the goal of obtaining an estimation of the *queuing* delay T_q. Nevertheless, such measurements include also serialization delays, propagation delays, and so on. To extract a measure of T_q from either RTT or OWD, the common assumption is that contributions to delay other than queuing are fixed, at least over some time interval. Examples of such congestion controllers include Vegas [51], FAST TCP [52], Compound TCP [53] and LEDBAT [54]. The latter two have been deployed in the Internet, but only LEDBAT seems to have been widely adopted[5].

In spite of DBCC's promises of combining low latency with low packet loss, such congestion controllers may typically suffer from one or more of the following issues [55, 56, 57].

- *Incorrect estimates of minimum RTT or minimum OWD:* A way of estimating T_q from RTTs or OWDs is to measure the *minimum* or *base delay value* (either unidirectional or bidirectional, depending on whether OWD or RTT is used), then considering that any excess delay is due solely to queuing. However, a new delay-based flow arriving at a busy queue may mistake the queuing delay due to *other*, already-present flows for the minimum delay; this leads to the so-called *latecomer advantage*. Similarly, under some circumstances a delay-based flow may mistake its *own* induced queuing delay for the base delay, leading to standing queues increasing over time (§ 3.3.1).

  Preliminary previous work by RITE partners [58] has shown that it is possible to avoid errors in base delay estimates by using the change in delay, or the *delay gradient*, as a measure instead. Such work will be further pursued by RITE in the incoming months, including looking at alternate delay-based congestion signals.

  It is also worth noting that if the network indicated which packets were *not* queued—e.g., as indicated by the *absence* of an early ECN-type mark that is present only if the packet needs to be queued—, base delays could be measured with better accuracy. We therefore plan to explore such approaches in RITE.

- *Noisy measurements:* Delay measurements are not always well correlated with congestion. Sampling a queue, especially one under load, can yield a noisy delay measurement [59]. Nonetheless, as argued by McCullagh and Leith [59] what matters most is *aggregate* behaviour, so path delay can indeed be used as input to a congestion controller. Another source of noise when estimating delays is TCP's delayed acknowledgements (ACKs) mechanism, but this effect can easily be compensated for by considering only immediate ACKs. Finally, some MAC layers may inherently add delay jitter due to link-local retransmissions.

  Filtering is used by some DBCC algorithms as a mean of removing noise from delay samples; typically, this is implemented as a simple averaging (low-pass) filter. The flip side of such filtering is that it degrades the responsiveness of the congestion control system to rapid changes. Proposals such as [60, 61, 62, 58] use instead a probabilistic congestion signal, i.e., triggering a congestion-control action depends on a weighted probability proportional to the measured delay.

  One avenue of research worth exploring is to consider the "noise" of the delay measurements as the signal used to infer congestion, rather than something to be filtered out. We expect that this may be especially useful when buffers are small.

- *Coexistence with loss-based congestion control:* In general, DBCC flows may not coexist well with those using loss-based congestion control (LBCC). Delay-based controls react earlier to congestion, and try to keep delays, and thus queues along the path, small. Loss-based mechanisms such as TCP's try to fill network queues, probing for available bandwidth, until there is loss. Hence, flows using LBCC tend to "push aside" flows using DBCC, potentially leading to starvation of the latter.

  DBCC proposals such as [61, 62, 58] cope with this problem by adopting a *bimodal* behaviour. They try to detect whether loss-based flows share the bottleneck buffer with them. By default, a "bimodal" DBCC flow reacts to delay as a congestion signal and strives to keep the bottleneck buffer lightly loaded. If even a single *long-lived* LBCC flow goes through the same bottleneck (and so, it tends to fill the buffer), the DBCC flow switches to TCP-like congestion control. On the other hand, if competing LBCC flows are *short-lived*—and so, likely to be *latency-sensitive*—, then in general they are not able to drive buffers full, so they should see low queuing delay *even if they share the bottleneck with long-lived (DBCC) flows.*

- *Directional congestion:* Using RTTs as a metric of path delay is often simpler than using OWDs, but RTTs do not allow to distinguish between congestion on the "forward" path and on the "backward"

[5] Compound has been part of Windows systems since Vista, but it seems to be disabled by default.

path. Moreover, congestion in the return path adds up as noise to the forward delay estimation. The tradeoff between the use of RTTs versus OWDs is a practical one; e.g., measuring the OWD with TCP would require changes in both senders and receivers.

3.3.1 Impact of LEDBAT congestion control on short flows

Delay-based congestion control can be used to achieve a *less-than-best-effort* or *scavenger* behaviour. Indeed, queuing delay can be seen as an early congestion signal, so a congestion controller may be designed to react and back off promptly and conservatively when delay increases [57]. This is the approach taken by LEDBAT (Low Extra Delay Background Transport) [54]. LEDBAT is an IETF experimental congestion control algorithm, currently deployed in peer-to-peer application software[6] as well as in some operating systems[7].

LEDBAT works as follows. The sender timestamps data packets with values taken from a local clock. The receiver computes the difference between the sender timestamp and its own local clock, then sends this difference back to the sender in its acknowledgement packets (ACKs). One-way queuing delays from sender to receiver are constantly monitored by the sender, by subtracting from every new sample carried by an ACK the *minimum* clock difference measured so far[8]. Then, a congestion window is increased or decreased in proportion to the difference between the observed queuing delay and a fixed *target* value (recommended by RFC 6817 to be $\leq$ 100 ms). In order to make LEDBAT behave as a scavenger protocol, a few rules are applied at the sender so that (in theory) a LEDBAT flow does not grab more bandwidth than a competing TCP flow. Window increases are never larger than those a TCP sender would apply under similar conditions; also, the closer the queuing delay gets to the target, the smaller the window increase. Finally, packet loss results in a window reduction just like TCP's.

In spite of fairly abundant literature on the subject of LEDBAT performance, most published work has focused on (inter- and intra-protocol) fairness in terms of throughput (see [57] and references therein). Little effort has been devoted to assessing how LEDBAT behaves with respect to *latency* [63, 64, 65], with mostly anecdotal evidence to back up claims of LEDBAT being innocuous to other flows, in terms of added latency.

We have thus studied the impact that LEDBAT flows may have on *latency-sensitive* flows [66]. Our work is based mainly on simulations, but some limited-scale experiments with a simple testbed confirm some of our main conclusions.

We have focused on two potential issues with the current LEDBAT specification. First, it assumes that a sender can accurately measure the base delay—i.e., the lowest possible end-to-end delay, without queuing. A delay-history mechanism is embedded in the algorithm, so that e.g. if the base delay changes due to rerouting, old invalid values are eventually discarded. For this, the minimum delay is measured over one-minute intervals, then the base delay is taken as the smallest of N such minimum values; RFC 6817 recommends setting N =10. A sliding window of N minimum values is kept, so the system's memory goes back for N minutes.

Second, a LEDBAT sender is supposed to detect the presence of other flows in the bottleneck by the amount of queuing they cause, and also by way of packet loss experienced by the LEDBAT flow. The design of the congestion controller is such that LEDBAT should behave less aggressively than a TCP flow, so a LEDBAT flow should "quickly get out of the way" of competing TCP flows. As we will see below, this claim may not always hold.

To illustrate the first issue, consider a long-lived LEDBAT flow with *no* competing flows, so the bottleneck buffer is empty. Once it fills the bottleneck queue up to the target value T, the sender will keep a steady, standing queue. Later, after N minutes, all minimum delay measurements will yield a value $\approx T$. Because of the history mechanism, the *actual* base delay is "forgotten" and the sender takes its own

[6] See e.g. https://github.com/bittorrent/libutp.

[7] For instance, recent versions of Apple's MacOS X implement a close variant of the mechanism described in RFC 6817; see http://opensource.apple.com/source/xnu/xnu-2050.24.15/bsd/netinet/tcp_ledbat.c.

[8] Measuring the OWD would in principle require some form of clock synchronization between sender and receiver. By subtracting the minimum, the LEDBAT sender corrects for any offset that there may exist between the two clocks, avoiding the need for synchronization. See e.g. [57] for a more detailed explanation.

induced queuing delay as part of the base delay. The specification argues that randomness inherent to e.g. operating system delays would ensure the queue oscillates enough, so that the sender eventually measures the true base delay. Our results strongly suggest this is not necessarily the case. Figures 3.3 and 3.4 show that (a) delays may well keep on growing until buffers are saturated, (b) sources of randomness would need to add very large amounts of noise in measurements to be effective. This issue may be more severe in scenarios with "bloated" network buffers, since queuing delay may grow well beyond its target value before losses happen.

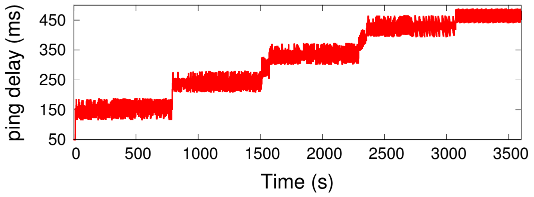


Figure 3.3: End-to-end delays, measured with the ping tool on a simple testbed with one long-lived LEDBAT flow. The LEDBAT flow was generated using libutp's *utp_test* program.

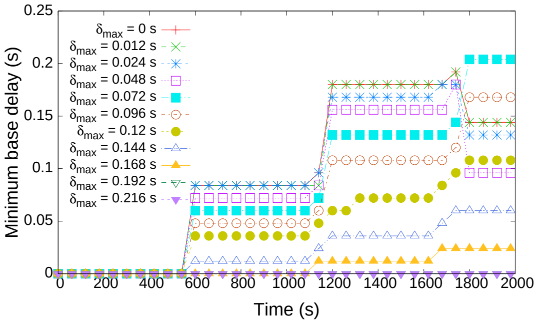


Figure 3.4: Evolution of the minimum base delay over time, obtained via ns2 simulation. Random jitter, uniformly distributed in $[0, \delta_{\max}]$ ms, is added to inter-packet transmission times. Note how maximum jitter has to be $\approx$ 2 T (in this case, T=100 ms), to allow the queue to drain so the base delay stops growing.

A simple palliative to this problem could be to enforce long enough "transmission pauses" at the sender before updating the base delay value, as well as using smaller values of T. Arguably, a more promising approach would consist of applying to LEDBAT's design some of the solutions outlined in § 3.3.

The second issue is summarized in Figure 3.5, obtained via ns2 simulations. Flow completion times were measured for TCP flows of varying sizes, when competing with a long-lived LEDBAT flow. It can be seen how LEDBAT only yields to a competing flow when the latter induces a large (additional) queuing delay for LEDBAT to detect it; otherwise, the LEDBAT sender does not react noticeably. For example, with TCP flows as large as 50 MSS (75,000 bytes, in this case), the LEDBAT flow reduces its sending rate by less than 5%. For short, latency-sensitive flows this means that their packets always experience a queuing delay of T. Moreover, the recommended maximum value of 100 ms for T is deceptive. For interactive applications like web browsing, flow completion involves transferring several objects in sequence [67], so delays add up to give total latency figures that are much higher than T, as shown in Figure 3.6.

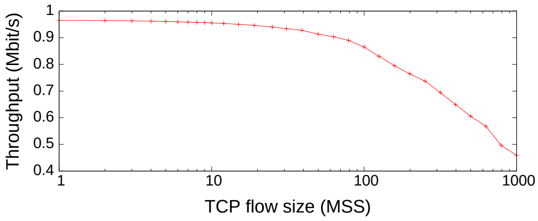


Figure 3.5: Average throughput of a LEDBAT flow competing with a TCP flow of varying size. The LEDBAT flow is insensitive to the presence of TCP flows which size is not large enough to induce enough added delay in the bottleneck queue.

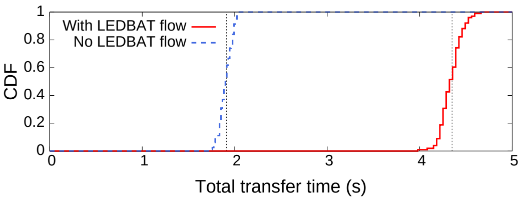


Figure 3.6: CDF of the total transfer time for web-like TCP flows, with and without a competing LEDBAT flow. The dotted vertical lines indicate average transfer times

3.3.2 Delay-gradient based Congestion Control, considering all congestion signals

Recently, a delay-based CC algorithm called CAIA Delay Gradient (CDG) has been proposed in [58], now a RITE participant through UiO. CDG addresses some issues that plague many delay-based CC proposals (e.g., the use of delay thresholds to detect congestion, or the noise in the RTT signal). CDG modifies the TCP sender to use the *delay gradient* to determine whether there is congestion. CDG aims at having an average back off probability which is independent of RTTs, as well as backing off in the face of congestion losses while tolerating non-congestion related ones. Finally, CDG strives for coexisting well with flows using loss-based CC.

CDG uses maximum RTT (τ_{max}) and minimum RTT (τ_{min}) during measured RTT interval[9]. Based on these two measures for every RTT interval, the change in RTT measurements is kept as a gradient which is a less noisy measurement as compared to per-packet RTT measurements. Thus, according to CDG, the gradient for the n^{th} RTT interval is:

$$g_{min,n} = \tau_{min,n} - \tau_{min,n-1} \tag{1}$$

$$g_{max,n} = \tau_{max,n} - \tau_{max,n-1} \tag{2}$$

CDG applies moving average smoothing by using the following equation:

$$\bar{g}_n = \sum_{i=n-a}^{n} \frac{g_i}{a} = \bar{g}_{n-1} + \frac{g_n - g_{n-a}}{a} \tag{3}$$

where a is the number of samples in the moving average window and $g_i = g_{x,i}$ for calculating $\bar{g}_{x,n}$, with $x \in \{min, max\}$. As explained in [58], the congestion window w for the n^{th} RTT interval is updated as follows:

$$w_{n+1} = \begin{cases} w_n\beta & \text{if } X < P[\text{backoff}] \text{ and } \bar{g}_n > 0 \\ w_n + 1 & \text{otherwise} \end{cases} \tag{4}$$

w is updated every RTT by using Eq. 4. β is a multiplicative decrease factor and X is drawn from a uniform random distribution in $[0, 1]$. $P[\text{backoff}] = 1 - e^{-(\bar{g}_n/G)}$, where G is a constant scaling factor.

We plan to further analyze and extend the prior experimental study of CDG, of which an implementation is available as a part of FreeBSD v9.x. Our motivation is to design and evaluate a better loss detection mechanism keeping in view the smoothness of gradient information. That is, the RTT signal is noisy and the gradient reduces the noise somewhat. We want thus to investigate if we can further reduce noise in the gradient signal, specially when time-sensitive flows are considered. Moreover, we plan to deepen the investigation on how does CDG perform in coexistence with other standard TCP flows. Finally, we would like to explore how to leverage ECN information in the framework of a delay-based CC such as CDG.

4 AQM techniques for low-latency requirements

An Active Queue Management (AQM) mechanism proactively marks or drops packets as signals to endpoints (or other routers or switches along the path), in order to manage the queues in such a way as to achieve certain queue loss and latency characteristics. AQM algorithms have two main goals: (a) to allow a buffer to absorb packet bursts while not letting long standing queues to form in the buffer; (b) to avoid loss synchronisation between flows competing in a bottleneck buffer. AQM generally works in combination with scheduling, traffic shaping, and/or transport layer congestion control. Adams [68] provides an extensive survey of techniques, starting from Random Early Detection (RED) [69] introduced in 1993.

[9] Time interval is the same as used in TCP Vegas [51]

AQM has once again become an intensive area of research as a tool for helping to combat latency caused by large buffers along the end-to-end path, particularly for buffers in slow access links at the edges of the Internet, where the number of flows sharing the link is typically small. Two recent proposals, Proportional Integral controller Enhanced (PIE) [70] and Controlled Delay (CoDel) [71] (and more recently FQ_CoDel, a fair queueing version of CoDel) aim to manage large buffers to have a small average queueing delay. Both of these schemes, however, allow very high latency during transient congestion episodes. In these schemes the cost of dropping packets during transient congestion episodes is considered to be higher than the cost of the latency that keeping them induces. As latency becomes more and more critical for applications, this may no longer be applicable.

4.1 Evaluation and improvement of AQM algorithms designed for low latency

CoDel and PIE have been the subject of much recent discussion in fora such as the IETF and the IRTF. Interest in these algorithms is due to their promise of keeping latency in check, while requiring no manual parameter tuning. However, to the best of our knowledge they have not been put yet through exhaustive independent testing, especially with real-world implementations, and some of the available studies have not been subject to peer review. Results reported in [72] assess the performance of CoDel, but only on DOCSIS 3.0 modems and rely solely on ns-2 simulation results; [73] complements such study by considering PIE and FQ_CoDel as well. [74] conducts a preliminary investigation on the interaction of different AQMs and low-priority "scavenger" congestion control (such as LEDBAT) using real-life tests.

Therefore, one objective of the RITE project is to help in covering some important gaps in existing studies, by performing a thorough independent assessment of key AQM algorithms, both by simulation and by real-world testing. In particular, for testbed experiments we have considered the case of IEEE 802.11 links, where AQM implementation efforts such as those by the Bufferbloat project[10] have focused on, but for which there is a dearth of available results. We have taken ARED [75], an adaptive version of the RED algorithm, as a baseline for performance comparison purposes. ARED has been chosen because it shares with both CoDel and PIE the goal of "knob-less" operation. Given that one of the major contributions of the new mechanisms appears to be their parameterless operation, we have set to try to answer the question: How much better are CoDel, FQ_CoDel and PIE with respect to ARED? Are they truly insensitive to the values of some "magic numbers" in the algorithms, or are these factors actually just parameters in disguise? Some of the first results of this ongoing effort are shown in § 4.1.1 for testbed experiments.

Moreover, as part of this effort, we plan as well to contribute to the work of the newly-created AQM Working Group (WG) at the IETF. A starting point for this contribution will be the submission of a draft describing AQM evaluation guidelines; such a document is one of the current milestones in the Working Group's charter. A presentation of our work in this direction is already scheduled for the first meeting of the AQM WG at the Vancouver IETF 88th meeting, in early November 2013.

Finally, we expect our evaluation results will feed work on either new AQM proposals, or modifications to existing ones to improve their performance. For example, algorithms such as ARED were designed with the implicit assumption that the link speed is fixed. Adapting ARED to explicitly take into account link speed variations—as may happen in wireless links, say—is a possible avenue of research to be pursued in RITE.

4.1.1 Preliminary experimental assessment

Our experimental evaluation of low-latency AQM mechanisms centers on access links scenarios, both wired and wireless; these links act as bottlenecks, and use of AQM in such bottleneck buffers would be beneficial. Edge network scenarios with a low level of statistical multiplexing are emulated by means of simple testbed setups (dumbbell for wired, and the one depicted in Figure 4.1 for wireless). For simplicity, wired bottleneck links are Ethernet links running at 10 Mb/s. For wireless links we use

[10] http://bufferbloat.net/

standard 802.11b/g hardware and drivers.

Figure 4.1: Wireless testbed for AQM evaluation.

We highlight below a few key results of our experiments; a paper summarising our findings has recently been submitted to a leading conference. Bottom and top of whisker-box plots show 10th and 90th percentiles, respectively.

4.1.1.1 Parameter sensitivity Both CoDel and PIE claim that the default values for their parameters are robust enough to cover a fairly wide range of operating conditions, in terms of RTTs. Thus, we have assessed how the AQMs under consideration behave in wired scenarios with different numbers of flows and RTT values, and also for settings of their *target delay* and *update interval* parameters different from the default ones.

Figure 4.2 shows per-packet queuing delays under light, moderate and heavy congestion, when 4 sender/receiver pairs share the bottleneck. Three base RTT values are used (5, 100 and 500 ms), covering very different delay scenarios. Interestingly, ARED seems to outperform the other two AQMs in many instances, in terms of keeping delay under tighter bounds, with a reasonable penalty in terms of goodput in many cases (Figure 4.3).

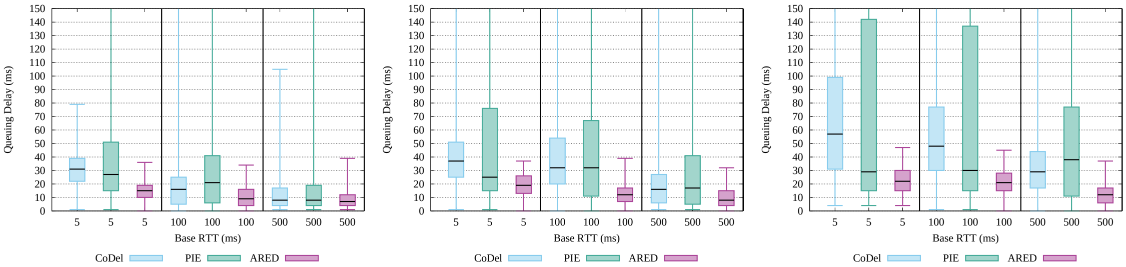
The behavior of the three AQM algorithms with different target delay values is illustrated in Figure 4.4. Here the plots show total per-packet RTTs; queuing delays can be obtained by subtracting the base RTT (100 ms) from plotted values. It can be noticed that, again, ARED achieves delays which are both closer to the desired target and within tighter bounds than either CoDel or PIE.

4.1.1.2 Wireless performance Our experiments have focused on emulating a public WiFi network, where the provided Internet access bandwidth is larger than the capacity of the wireless network; the access point becomes a bottleneck for downlink traffic.

Figure 4.5 presents results for the case where data traffic flows only in the downlink direction. We see that AQMs can control latency well on the wireless interface in such a case. This is because in scenarios with data traffic only in the downlink, few stations are actively trying to access the channel while the AP is in charge of transmitting all data packets, resulting in the WiFi rate adaptation mechanism to rarely trigger the use of lower bit-rates when the channel condition is good and stable (downlink and uplink Wi-Fi performance has already been thoroughly investigated in previous works by RITE partners [76, 77]).

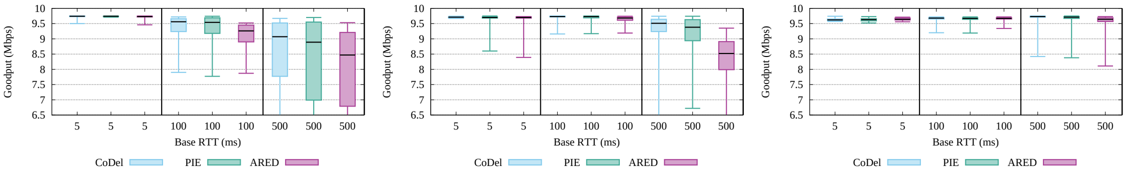
However, the presence of uplink traffic can lead to significantly high latencies. Here, the channel access latency becomes the major source of delay. Figure 4.6 shows the uplink performance in a mixed traffic scenario with an equal number of uploading and downloading flows.

A detailed investigation of the packet traces revealed the origin of this significant increase of delay with CoDel and ARED: the downlink ACKs that are caused by the uplink traffic share the bottleneck with the data packets at the AQM queue (AP's wireless interface). The AQM queue is then backlogged with these data packets and ACKs, and frequent ACK drops by an AQM lead to excessive end-to-end delays for



(a) Light (b) Moderate (c) Heavy

Figure 4.2: Per-packet queuing delay, for different base RTTs and with light, moderate and heavy congestion levels (1, 4 and 16 long-lived TCP flows per sender/receiver pair). Target delay is 5 ms.



(a) Light (b) Moderate (c) Heavy

Figure 4.3: TCP goodput at the bottleneck link (measured over 5-sec intervals), for the case depicted in Figure 4.2.

the uplink traffic, in addition to an already high channel access delay that prolongs the ACK departures from the AQM queue. This causes the AQM to react further to the accumulated ACKs in the buffer, making TCP's feedback loop exceedingly long.

The RTT that is measured by the TCP uplink sender gets longer when ACKs at the head of the queue are dropped (CoDel). Until a timeout occurs (which, by itself, is affected by the duration and variation of the measured RTT), uplink data packets will be transmitted on the channel without the sender being notified about congestion until the AP finally gets the chance to deliver one of its ACKs (note that the AP has the same chance to access the channel as all other stations). In this regard PIE works best, as it is the AQM that least aggressively reacts to queue growth in our evaluations, as opposed to ARED which drops rather aggressively.

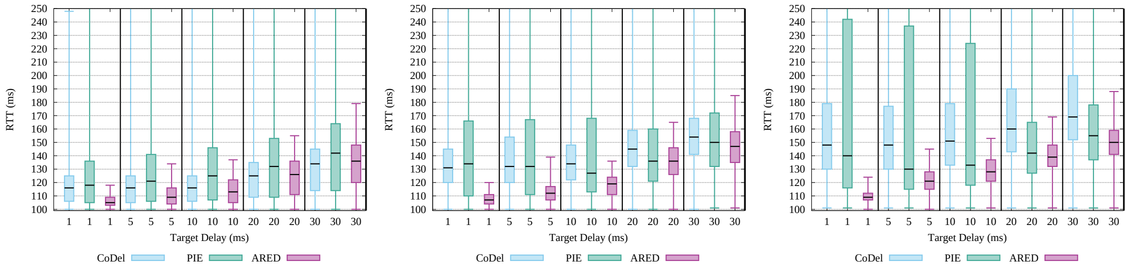
4.1.2 Simulation study and evaluation guidelines

Tests with real-world implementations are necessary to validate any proposal, and to uncover issues that may not be apparent otherwise. On the other hand, such tests are subject to practical constraints, and their results may be bound to the hardware and software used. This is why simulation is useful and allows to complement any real-world experiments.

Several RITE partners are thus currently carrying out a joint, extensive simulation study of AQM algorithms such as CoDel, PIE and ARED, together with versions of those coupled with stochastic fair queuing. This study aims at exploring a large parameter space that would be hard to cover with simple testbeds. Some early results have been already published in [78]. A second goal of the study is to feed the development of an evaluation-guidelines document, as described before.

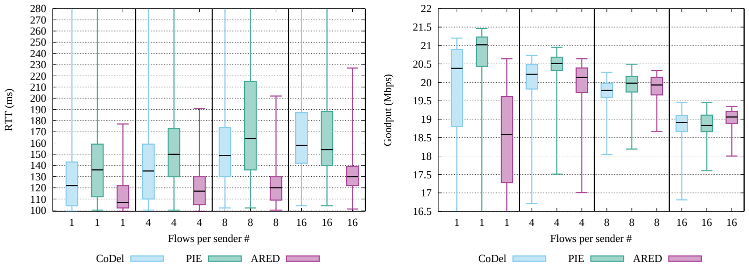
For now, we have chosen the well-known ns2 simulator due to the availability for this tool of simulation models of the AQMs under study. As performance metrics, our main focus is on queuing delays, buffer occupancy, link utilisation, throughput and packet drop rate, and tradeoffs between some of these parameters; initial results reported in [78] consider as well pseudo-subjective metrics like peak signal-to-noise ratio and mean opinion scores for video and voice, respectively.

We have identified, and are currently investigating, a spectrum of evaluation scenarios as follows:



(a) Light (b) Moderate (c) Heavy

Figure 4.4: Per-packet RTTs, for different target delays and with light, moderate and heavy congestion levels (1, 4 and 16 long-lived TCP flows per sender/receiver pair). Base RTT is 100 ms.



(a) Per-packet RTT (b) Goodput

Figure 4.5: 802.11 downlink traffic scenario for 4 sender/receiver pairs, base RTT of 100 ms and target delay of 5 ms. TCP goodput is measured over 5-sec intervals.

Sensitivity study of AQM parameters: Our first goal is to assess to what extent the performance of proposed AQMs is dependent on the setting of their main parameters: target delay and update interval.

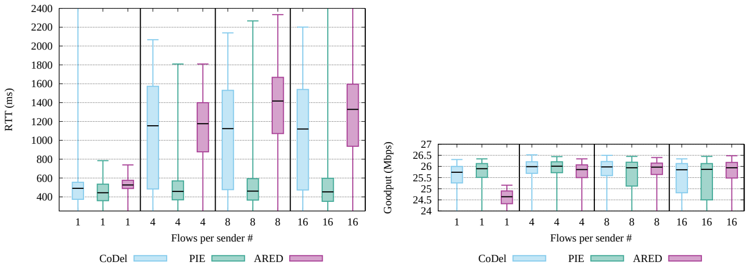
Impact of buffer size on AQM performance: We wish to evaluate here the ability of AQMs to absorb and deal with traffic bursts, and how the buffer size may influence the behaviour of AQMs. Besides using a range of values proportional to the bandwidth-delay product of the path, we take into account buffer sizes found in actual home gateways, as reported in [79]. Loss synchronisation between flows—which algorithms such as RED were supposed to alleviate—has to be taken into account and assessed.

RTT sensitivity and fairness: We would like to see, when using default values for AQM parameters, how the mechanisms will fare in scenarios where TCP flows with different RTTs are sharing the bottleneck.

Flow isolation via fair queuing: Proposals such as FQ_CoDel already combine an AQM algorithm with some form of stochastic fair queuing (SFQ). It is important therefore to analyse e.g. whether improvements in performance are due to SFQ, to AQM, or to the combination of both.

Fluctuating link speeds: Wireless technologies, as well as several wired ones, do not offer a constant link speed; instead, the bandwidth an end-host "sees" may fluctuate over time due to many different causes (link quality, allocation of resources by a centralised entity, etc.). This may be particularly critical for algorithms, like PIE and ARED, that do not directly measure queue sojourn times.

Realistic traffic profiles: Simple cases with a few long-lived TCP flows may be helpful in identifying issues and pathological behaviour of AQMs, however, their performance in the presence of e.g. web-like (short, latency-sensitive) TCP flows has to assessed as well. Similarly, it is important to run tests using different packet sizes, especially since buffers (and AQMs) may be byte-based or packet-based.



(a) Per-packet RTT

(b) Goodput

Figure 4.6: 802.11 mixed downlink+uplink traffic scenario for 4 sender/receiver pairs, base RTT of 100 ms and target delay of 5 ms. Results shown correspond to *uplink* flows.

Datacentre-like scenarios: We are investigating how the AQMs of interest perform in high-speed links with very short RTTs. The main question we try to address here is: Is there something structural in such AQMs that make them unsuitable for high-speed, low-delay links, or is it just a matter of adequate parameter setting?

Rural Broadband-like scenarios: Conversely, we are looking into situations, typical of rural broadband access, where link speeds are low and RTTs large.

4.2 AQM and Explicit Congestion Notification (ECN)

Theoretically, AQM could be much better when an explicit signal, such as Explicit Congestion Notification (ECN), is present. This avoids the requirement that a transport/application has to experience loss to discover congestion. ECN allows routers to mark packets earlier than the point where packets would normally be dropped by AQM, e.g., right at the point where capacity is first exceeded. Responding to ECN marks could cause senders to back-off earlier and reduce queuing delay.

So why do we not see wide-spread deployment of ECN? ECN relies both on ISPs enabling AQM to mark packets and on enabling the user stack to process ECN markings. Most mainline operating systems now implement ECN, but while being enabled at the server-side, this is typically disabled in clients. Awareness of the issues incurred by latency helps provide an important incentive for users to enable ECN, if network operators deploy ECN-enabled routers.

We assert that now is the time to consider seriously an ECN roadmap to deployment, because this can directly leverage the benefits of AQM for latency-sensitive applications. In doing so, we need to plot an incentive-compatible way forward to affirm user and network benefits from ECN. As in AQM, ECN needs to seek solutions that maximize performance benefits across different kinds of applications, without incurring penalties in explicit configuration.

4.2.1 Early ECN Marking: a proposal for building a better ECN

The incentives for ECN deployment will be assisted by defining a marking method that potentially improves and never degrades the performance experienced by users, while reducing the queuing delay (latency) in routers.

How and how early should AQM mark? RFC 3168 [80] says that AQM should only mark packets as Congestion-Experienced (CE) when a router would drop them in the absence of ECN. This ensures compatibility between ECN-capable and non-ECN-capable flows: if an AQM router would mark ECN-capable traffic earlier than it would drop non-ECN-capable traffic, the ECN-capable flows would receive lower throughput than a non-ECN-capable flows. This present requirement does not encourage use of ECN.

We assert that routers should be encouraged to support an earlier ECN marking, in a method we call Early ECN Marking (EEM). This can provide significant benefit, with obvious performance improvement for applications that are both loss and latency-sensitive such as video. If early marking (based on low latency thresholds) is not the default method – assuming that some flows want a behaviour closer to current Internet trade-offs, then we need a way to introduce an early marking method.

We suggest considering exploring a change to existing standards with the following requirements:

1. Routers become able to identify packets from an "new" ECN-capable transport and early-mark only such packets.
2. If both types of traffic share the same queue, "new" ECN-capable traffic should exhibit a more aggressive form of transport congestion control to compensate for the presumed early ECN-marking. For instance, in response to a "new" CE signal, instead of halving the sending rate, a new ECN-capable TCP could reduce it by a smaller amount.

Item 2) is an open transport area research question. *Item 1)* could be addressed by re-defining the ECT(1) codepoint. ECT(1) was introduced in RFC 3168 [80], with one use to support the ECN nonce. This nonce is only useful if it is deployed in at both an ECN sender and receiver, and has not seen widepread deployment. ECT(1) has also been used to provide alternate ECN markings. We suggest to use the ECT(1) codepoint to let a "new" ECN-capable sender signal an EEM-capable router (on a side note, it is possible to realise a weaker variant of the nonce with the mechanism presented here, but we skip these details for the sake of brevity). The CE mark remains the same as with standard-ECN.

4.2.1.1 Backward compatibility Methods have been proposed that provide flow-isolation - reducing the impact of flows sharing a common queue with other flows that have different application goals. Diffserv provides one architecture for doing this, there are also AQM-variants that attempt to isolate flows. Combining this with AQM and ECN could prevent bulk flows impacting latency (or loss) sensitive flows. Mechanisms are also needed to isolate traffic from flows that do not appropriately respond to CE-markings or loss, to prevent increasing the latency for other flows.

However, several questions arise when considering EEM in a network with a mix of routers and end hosts that may or may not support EEM and cannot separate the traffic flows:

- Should we keep separate virtual queues for standard-ECN and EEM flows?
- If standard-ECN and EEM flows share an EEM-enabled router queue, how should we tune the EEM sender's congestion response to be compatible with standard-ECN flows?
- If standard-ECN and EEM flows share the same queue in a traditional ECN router, EEM flows may be marked the same as standard-ECN flows and then push these flows aside - but is this bad?, or can it motivate deployment? e.g., is it worse than current behavior with CUBIC vs. standard-TCP, or would it just give users the right incentives to upgrade their systems?

Table 4.1 provides an overview of all combinations of "old" and "new" hosts and routers, assuming that traffic from "old" and "new" end systems share a common router queue. The following terminology is used:

Sender:

1. Old (no ECN): TCP flow with ECN disabled
2. Old (ECT(0)): TCP flow with standard ECN enabled
3. New (ECT(1)): TCP flow with Early ECN Marking (EEM) enabled

Router:

1. No AQM: FIFO/DropTail

Router Type	No AQM	Old AQM ECT(0)	New AQM ECT(1) +ECT(0)
Sender Type			
Old (no ECN) + Old (no ECN)	Default behavior	Default AQM behavior	Default AQM behavior
Old (no ECN) + Old (ECT(0))	Default behavior	Default AQM behavior and saves some packets for ECN flow	Default AQM behavior and saves some packets for ECN flow
Old (no ECN) + New (ECT(1))	Default behavior	Default AQM behavior and, if router supports ECT(1): saves some packets for ECN flow, ECT(1) flow is more aggressive than the other flow	Default AQM behavior for old flow, early marking for new ECT(1) flow, which may need to be aggressive enough to compensate
Old (ECT(0)) + Old (ECT(0))	Default behavior	Default AQM behavior and saves some packets for both flows	Default AQM behavior and saves some packets for both flows supposedly almost equally
Old (ECT(0)) + New (ECT(1))	Default behavior	Default AQM behavior and, if router supports ECT(1): saves some packets for ECN flow, ECT(1) flow is more aggressive than the other flow	Default AQM behavior for old flow, early marking for new ECT(1) flow, which may need to be aggressive enough to compensate
New (ECT(1)) + New (ECT(1))	Default behavior	If router supports ECT(1): new, aggressive end system reaction for both flows. Else, default behavior.	EEM behavior

Table 4.1: Summary of possible deployment combinations for 2 TCP flows.

2. Old AQM ECT(0): Any AQM with standard CE marking on ECT(0) packets
3. New AQM ECT(1)+ECT(0): A hypothetical AQM which does EEM-marking on ECT(1) packets and standard CE marking on ECT(0) packets. When exactly these markings should happen is an open research question.

5 Summary of analysis

The analysis phase of RITE has covered a broad range of different approaches to the problem of Internet transport latency. This section shows how the different elements of analysis and preliminary experiments complement each other, in the topics covered by Work Package 2, and provides a firm foundation for the planned future tasks of the project.

The bulk of our efforts here has been on issues related to network buffers, their interaction with end-to-end protocols like TCP and, conversely, how end-to-end mechanisms interact in such buffers; also, on how network signals (such as ECN) can be better exploited by end-system mechanisms such as delay-based congestion control. The status of the main topics covered so far in WP2 and their dependencies is summarised in Table 5.1; when relevant, this table points as well to cross-dependencies with WP1 tasks. We can see from Table 5.2 that all the objectives of WP2 are addressed by the subprojects currently being investigated in RITE.

One important outcome of this analysis phase is a survey paper of latency-reducing techniques that will

Task objectives	Subprojects	Maturity level	Depends on	Feeds task
Network delays analysis	Development of recommendations on buffering and AQM	Analysed, work to be continued	Fed by results from Tasks 2.2 and 2.3	2.4
Interaction of CC algorithms with latency-sensitive flows in network buffers	Determination of types of queueing / scheduling / packet drop mechanisms at congestion points, Mobile network multimedia-unfriendly tests	Analysed, Preliminary results	Partly dependent on techniques and protocol improvements developed in Task 1.2 (deliverable 1.1)	1.2, 1.4, 2.2, 2.3, 2.4
Delay-based CC issues	Delay-gradient CC considering all congestion signals, Impact of LEDBAT on short flows	Analysed, Preliminary results	Partly dependent on techniques and protocol improvements developed in Task 1.2 (deliverable 1.1), and on Early ECN Marking	1.2, 1.4, 2.2, 2.3, 2.4
AQM algorithms designed for low latency	Evaluation and improvement of low-latency AQMs, AQM evaluation guidelines	Analysed, preliminary work started	Partly dependent on Network delays analysis	2.2, 2.3, 2.4
ECN-based approaches for low latency	Early ECN Marking	Analysed, preliminary work planned to start in Fall 2013	Partly dependent on AQM algorithms designed for low latency	2.3, 2.4

Table 5.1: Table summarising the status of each of the main topics for Task 2.1 and their dependencies.

be submitted to a top-tier journal in the Fall of 2013. The survey was outlined in a two-page position paper [81], presented at the ISOC/RITE workshop on Internet latency that was held in September 2013[11]. This survey aims not only to list the state-of the art in Internet latency reduction techniques, but also to quantify the possible latency gain by applying different kinds of techniques. Such a tool has been of great value to the consortium when prioritising between topics, and for choosing what topics to focus on. In particular, one important goal of the analysis and survey work is to highlight tradeoffs between gains in latency reduction and ease of deployment, as illustrated in Figure 5.1.

Table 5.3 shows the involvement of each RITE partner in each subproject described in this document. All partners have collaborated on the latency analysis and on identifying the most promising topics based on this analysis. The partners listed in table 5.3 are the partners who have been involved in the activities during the reporting period. There will probably be other constellations of involved partners as work in the project progresses and as more subprojects move into a experimental/testbed stage.

6 Conclusion

This report has offered an overview of the results of the analysis phase of Work Package 2 in the RITE project. We have described the initial work done for the most mature topics, as well as the planned work for topics in a less mature stage.

The analysis phase has provided the needed insight to make the best choices between the possible future paths of investigation in the RITE project. The subtopics chosen for continued exploration are promising and provide the needed basis for the coming tasks according to the Description of Work.

[11] http://www.internetsociety.org/latency2013

Objective	Subprojects addressing objective
Explore latency issues for future infrastructures	All subprojects
Acquire a deep understanding of the latency/throughput tradeoff from a network perspective	Determination of types of queueing / scheduling / packet drop mechanisms at congestion points, Mobile network multimedia-unfriendly tests, Impact of LEDBAT on short flows
Make the network routers more suitable for low latency traffic	Evaluation and improvement of low-latency AQMs, Development of recommendations on buffering and AQM
Explore the use of alternative queue management for low latency, bearing in mind scalability issues and the massive amount of flows in the next generation networks	Evaluation and improvement of low-latency AQMs, Early ECN Marking
Investigate Internet deployment of combined network/end system solutions	Early ECN Marking
Develop methods to make home gateways and wireless access points appropriate for low-latency traffic	Determination of types of queueing / scheduling / packet drop mechanisms at congestion points, Evaluation and improvement of low-latency AQMs
Develop robust methods for end-system/home gateway interaction to enable low-latency communication	Evaluation and improvement of low-latency AQMs, Early ECN Marking, Considering all congestion signals

Table 5.2: The objectives of Work Package 2 and subprojects addressing the objectives.

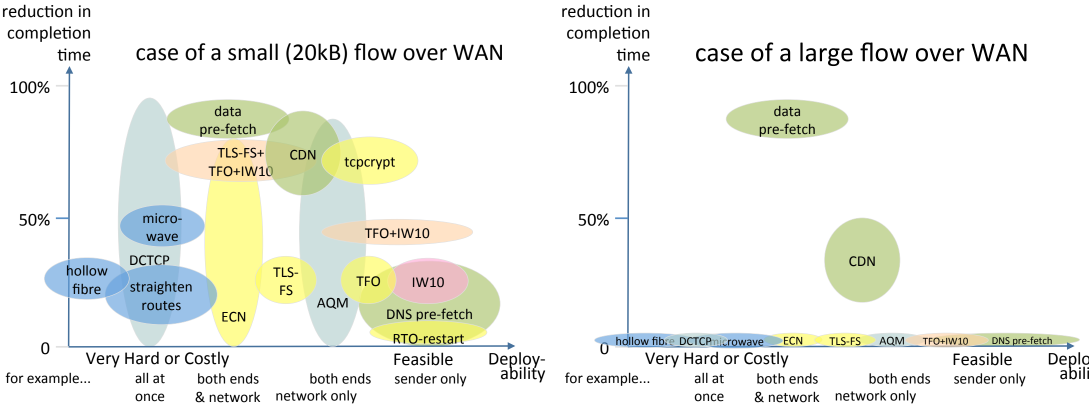


Figure 5.1: Examples of tradeoffs between deployability and potential for latency reduction, for several techniques (taken from [81]).

References

[1] "The Bufferbloat projects," Oct. 2013. [Online]. Available: http://www.bufferbloat.net/

[2] K. De Schepper, A. Petlund, and E. Andersen, "D3.1 traffic pattern analysis and data set acquisition report," RITE - Reducing Internet Transport Latency, European Commission - FP7, Jul 2013.

[3] ITU-T, "ITU-T Recommendation G.992.1: Asymmetrical digital subscriber line (ADSL) transceivers," Geneva, Sep 1999.

[4] ——, "ITU-T Recommendation G.992.2: Splitterless asymmetric digital subscriber line (ADSL) transceivers," Geneva, Sep 1999.

[5] ——, "ITU-T Recommendation G.992.3: Asymmetric digital subscriber line transceivers 2 (ADSL2)," Geneva, Apr 2009.

[6] ——, "ITU-T Recommendation G.992.4: Splitterless asymmetric digital subscriber line transceivers 2 (splitterless ADSL2)," Geneva, Sep 2002.

RITE subproject	Collaborating partners
Development of recommendations on buffering and AQM (§ 2.3)	BT, UoA
Determination of types of queueing / scheduling / packet drop mechanisms at congestion points (§ 3.1.1)	BT, KaU, SRL, UiO
Mobile network multimedia-unfriendly tests (§ 3.2)	KaU, SRL
Delay-gradient CC, considering all congestion signals (§ 3.3.2)	SRL, UiO
Impact of LEDBAT on short flows (§ 3.3.1)	IMT, UiO
Evaluation and improvement of low-latency AQMs (§ 4.1)	BT, IMT, UiO, UoA, ALU
AQM evaluation guidelines (§ 4.1.2)	IMT, UiO, UoA, ALU
Early ECN Marking (§ 4.2.1)	BT, IMT, UiO, UoA

Table 5.3: RITE subprojects in Work Package 2 and the partners collaborating on the respective subproject.

[7] ——, “ITU-T Recommendation G.992.5: Asymmetric Digital SubscriberLine (ADSL) transceivers – Extended bandwidth ADSL2 (ADSL2+),” Geneva, Jan 2009.

[8] ——, “ITU-T Recommendation G.993.1: Very high speed digital subscriber line transceivers (VDSL),” Geneva, Jun 2004.

[9] ——, “ITU-T Recommendation G.993.2: Very high speed digital subscriber line transceivers 2 (VDSL2),” Geneva, Dec 2011.

[10] ——, “ITU-T Recommendation G.993.5: Self-FEXT cancellation (vectoring) for use with VDSL2 transceivers,” Geneva, Apr 2010.

[11] IEEE, “IEEE 802.3ah Ethernet in the First Mile Task Force,” Jun 2010.

[12] Z. Li, X. Zhu, A. Begen, and B. Girod, “Forward and retransmitted Systematic Lossy Error Protection for IPTV video multicast,” in *Proceedings of the 17th International Packet Video Workshop, 2009*, 2009, pp. 1–9.

[13] ITU-T, “ITU-T Recommendation G.998.4: Improved impulse noise protection for DSL transceivers,” Geneva, Jun 2010.

[14] CableLabs, “DOCSIS 3.0 Specifications,” Aug 2006. [Online]. Available: http://www.cablelabs.com/cablemodem/specifications/specifications30.html

[15] ITU-T, “ITU-T Recommendation J.222.0: Third-generation transmission systems for interactive cable television services - IP cable modems: Overview,” Geneva, Dec 2007.

[16] ——, “ITU-T Recommendation J.222.1: Third-generation transmission systems for interactive cable television services - IP cable modems: Physical layer specification,” Geneva, Jul 2007.

[17] ——, “ITU-T Recommendation J.222.2: Third-generation transmission systems for interactive cable television services - IP cable modems: MAC and Upper Layer protocols,” Geneva, Jul 2007.

[18] IEEE, “IEEE 802.3 - 2012 LAN/MAN CSMA/CD Access Method,” 2012.

[19] ——, “IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 3: CSMA/CD Access Method and Physical Layer Specifications Amendment 1: Physical Layer Specifications and Management Parameters for 10 Gb/s Passive Optical Networks,” IEEE Std 802.3av-2009 (Amendment to IEEE Std 802.3-2008), pp. 1–227, 2009.

[20] ITU-T, “ITU-T Recommendation G.984.1: Gigabit-capable passive optical networks (GPON): General characteristics,” Geneva, Mar 2008.

[21] ——, “ITU-T Recommendation G.984.2: Gigabit-capable Passive Optical Networks (G-PON): Physical Media Dependent (PMD) layer specification,” Geneva, Mar 2003.

[22] ——, “ITU-T Recommendation G.984.3: Gigabit-capable Passive Optical Networks (G-PON): Transmission convergence layer specification,” Geneva, Mar 2008.

[23] ——, “ITU-T Recommendation G.984.4: Gigabit-capable passive optical networks (G-PON): ONT management and control interface specification,” Geneva, Feb 2008.

[24] ——, “ITU-T Recommendation G.987.1: 10-Gigabit-capable passive optical networks (XG-PON): General requirements,” Geneva, Jan 2010.

[25] ——, “ITU-T Recommendation G.987.2: 10-Gigabit-capable passive optical networks (XG-PON): Physical media dependent (PMD) layer specification,” Geneva, Oct 2010.

[26] ——, “ITU-T Recommendation G.987.3: 10-Gigabit-capable passive optical networks (XG-PON): Transmission convergence (TC) layer specification,” Geneva, Oct 2010.

[27] “The Importance of Dynamic Bandwidth Allocation in GPON Networks,” White Paper, PMC-Sierra, Sep 2008.

[28] “IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999), pp. 1–1076, 2007.

[29] B. Sikdar, “An Analytic Model for the Delay in IEEE 802.11 PCF MAC-Based Wireless Networks,” *IEEE Transactions on Wireless Communications*, vol. 6, no. 4, pp. 1542–1550, 2007.

[30] L. Feng, J. Li, and X. Lin, “A New Delay Analysis for IEEE 802.11 PCF,” *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2013.

[31] Y. Xiao and J. Rosdahl, “Throughput and delay limits of IEEE 802.11,” *IEEE Communications Letters*, vol. 6, no. 8, pp. 355–357, 2002.

[32] A. Zanella and F. De Pellegrini, “Statistical characterization of the service time in saturated IEEE 802.11 networks,” *IEEE Communications Letters*, vol. 9, no. 3, pp. 225–227, 2005.

[33] O. Tickoo and B. Sikdar, “Modeling Queueing and Channel Access Delay in Unsaturated IEEE 802.11 Random Access MAC Based Wireless Networks,” *IEEE/ACM Transactions on Networking*, vol. 16, no. 4, pp. 878–891, 2008.

[34] ——, “Queueing Analysis and Delay Mitigation in IEEE 802.11 Random Access MAC Based Wireless Networks,” in *in Proceedings of IEEE INFOCOM*. IEEE, 2004, pp. 1404–1413.

[35] “3GPP Technical Specification, TS 36.913, Requirements for further advancements for Evolved Universal Terrestrial Radio Access (E-UTRA) (LTE-Advanced) (Release 11),” Technical Specification Group Radio Access Network, 2012.

[36] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An Architecture for Differentiated Services,” RFC 2475, Internet Engineering Task Force, Dec 1998, updated by RFC 3260. [Online]. Available: https://tools.ietf.org/html/rfc2475

[37] D. Grossman, “New Terminology and Clarifications for Diffserv,” RFC 3260, Internet Engineering Task Force, Apr 2002. [Online]. Available: https://tools.ietf.org/html/rfc3260

[38] G. Fairhusrt and B. Briscoe, “Advice on network buffering,” Internet Draft draft-fairhurst-tsvwg-buffers, work in progress, Mar. 2013. [Online]. Available: http://tools.ietf.org/html/draft-fairhurst-tsvwg-buffers

[39] P. Karn, C. Bormann, G. Fairhurst, D. Grossman, R. Ludwig, J. Mahdavi, G. Montenegro, J. Touch, and L. Wood, “Advice for Internet Subnetwork Designers,” RFC 3819 (Best Current Practice), Internet Engineering Task Force, Jul. 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3819.txt

[40] F. Baker and G. Fairhurst, “Ietf recommendations regarding active queue management,” Internet Draft, work in progress, Oct. 2013.

[41] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet," RFC 2309 (Informational), Internet Engineering Task Force, Apr. 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2309.txt

[42] M. Welzl, G. Fairhurst, and N. Khademi, "ECN & early ECN marking," in *Internet Society Workshop on Reducing Internet Latency*, London, Sep. 2013. [Online]. Available: http://www.internetsociety.org/latency2013

[43] L. Stewart, D. A. Hayes, G. Armitage, M. Welzl, and A. Petlund, "Multimedia-unfriendly TCP congestion control and home gateway queue management," in *Proceedings of the second annual ACM conference on Multimedia systems*, ser. MMSys '11. New York, NY, USA: ACM, 2011, pp. 35–44.

[44] R. Chakravorty, J. Cartwright, and I. Pratt, "Practical experience with TCP over GPRS," in *Proceedings of IEEE GLOBECOM*, Taiwan, Nov. 2002.

[45] X. Liu, A. Sridharan, S. Machiraju, M. Seshadri, and H. Zang, "Experiences in a 3G Network: Interplay Between the Wireless Channel and Applications," in *Proc. ACM MOBICOM*, USA, 2008.

[46] M. Sågfors, R. Ludwig, M. Meyer, and J. Peisa, "Queue management for TCP traffic over 3G links," *Proc. IEEE WCNC*, 2003.

[47] H. Jiang, Y. Wang, K. Lee, and I. Rhee, "Tackling Bufferbloat in 3G/4G Networks," in *Proc. ACM IMC*, 2012.

[48] I. Rhee and L. Xu, "CUBIC: A new TCP-friendly high-speed TCP variant," in *Proc. Protocols for Fast Long-distance Networks*, 2005.

[49] L. Grieco and S. Mascolo, "Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control," *ACM CCR*, vol. 34, no. 2, 2004.

[50] S. Alfredsson, G. D. Giudice, J. Garcia, A. Brunstrom, L. D. Cicco, and S. Mascolo, "Impact of TCP Congestion Control on Bufferbloat in Cellular Networks," in *Proc. IEEE WoWMoM*, 2013.

[51] L. Brakmo and L. Peterson, "TCP Vegas: end to end congestion avoidance on a global Internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, pp. 1465–1480, 1995.

[52] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, "FAST TCP: Motivation, Architecture, Algorithms, Performance," *IEEE/ACM Transactions on Networking*, vol. 14, no. 6, pp. 1246–1259, Dec. 2006.

[53] K. Tan, J. Song, Q. Zhang, and M. Sridharan, "A Compound TCP approach for high-speed and long distance networks," in *Proceedings of IEEE INFOCOM 2006*, Barcelona, Spain, Apr. 2006.

[54] S. Shalunov, G. Hazel, J. Iyengar, and M. Kuehlewind, "Low Extra Delay Background Transport (LEDBAT)," RFC 6817 (Experimental), Internet Engineering Task Force, Dec. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6817.txt

[55] D. Hayes and D. Ros, "Delay-based congestion control for low latency," in *Internet Society Workshop on Reducing Internet Latency*, London, Sep. 2013. [Online]. Available: http://www.internetsociety.org/latency2013

[56] M. Welzl and D. Ros, "A Survey of Lower-than-Best-Effort Transport Protocols," RFC 6297 (Informational), Internet Engineering Task Force, Jun. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6297.txt

[57] D. Ros and M. Welzl, "Less-than-best-effort service: A survey of end-to-end approaches," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 2, pp. 898–908, May 2013.

[58] D. A. Hayes and G. Armitage, "Revisiting TCP congestion control using delay gradients," in *Proceedings of IFIP Networking*. Springer-Verlag, 2011, pp. 328–341.

[59] G. McCullagh and D. Leith, "Delay-based congestion control: Sampling and correlation issues revisited," Hamilton Institute, National University of Ireland Maynooth, Tech. Rep., 2008.

[60] S. Bhandarkar, A. Narasimha Reddy, Y. Zhang, and D. Loguinov, "Emulating AQM from end hosts," in *Proceedings of ACM SIGCOMM*, Kyoto, Aug. 2007.

[61] L. Budzisz, R. Stanojević, A. Schlote, F. Baker, and R. Shorten, "On the fair coexistence of loss- and delay-based TCP," *IEEE/ACM Transactions on Networking*, vol. 19, no. 6, pp. 1811–1824, Dec. 2011.

[62] D. Hayes and G. Armitage, "Improved coexistence and loss tolerance for delay based TCP congestion control," in *Proceedings of IEEE LCN*, 2010, pp. 24–31.

[63] J. Schneider, J. Wagner, R. Winter, and H.-J. Kolbe, "Out of my Way – Evaluating Low Extra Delay Background Transport in an ADSL Access Network," in *Proceedings of ITC 22*, Amsterdam, Sep. 2010.

[64] A. Abu and S. Gordon, "Impact of delay variability on LEDBAT performance," in *Proceedings of IEEE AINA*, Singapore, Mar. 2011.

[65] R. Jesup, "Issues with LEDBAT in wide deployment," in *84th IETF meeting*, Vancouver (BC), Canada, Jul. 2012.

[66] D. Ros and M. Welzl, "Assessing LEDBAT's Delay Impact," *IEEE Communications Letters*, vol. 17, no. 5, pp. 1044–1047, May 2013.

[67] D. Wischik, "Short messages," Royal Society workshop on Networks: modelling and control, Sep. 2007.

[68] R. Adams, "Active Queue Management: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, pp. 1425–1476, 2013.

[69] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[70] 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," in *Proceedings of the 14th IEEE Conference on High Performance Switching and Routing*, Taipei, Jul. 2013.

[71] K. Nichols and V. Jacobson, "Controlling Queue Delay," *ACM Queue*, vol. 10, no. 5, May 2012.

[72] G. White and J. Padden, "Preliminary Study of CoDel AQM in a DOCSIS Network," Technical Report, CableLabs, Technical Report, Nov. 2012.

[73] G. White, "A Simulation Study of CoDel, SFQ-CoDel and PIE in DOCSIS 3.0 Networks," Technical Report, CableLabs, Technical Report, Apr. 2013.

[74] Y. Gong, D. Rossi, C. Testa, S. Valenti, and D. Täht, "Fighting the Bufferbloat: On the Coexistence of AQM and Low Priority Congestion Control," in *IEEE INFOCOM Workshop on Traffic Monitoring and Analysis (TMA'13)*, 2013.

[75] S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: An algorithm for increasing the robustness of RED's active queue management," ICIR, Technical report, Aug. 2001. [Online]. Available: http://www.icir.org/floyd/papers/adaptiveRed.pdf

[76] N. Khademi, M. Welzl, and S. Gjessing, "Experimental Evaluation of TCP Performance in Multi-rate 802.11 WLANs," in *Proceedings of IEEE WoWMoM*, 2012, pp. 1–9.

[77] N. Khademi and M. Othman, "Size-based and direction-based TCP fairness issues in IEEE 802.11 WLANs," *EURASIP J. Wirel. Commun. Netw.*, pp. 49:1–49:13, Apr. 2010. [Online]. Available: http://dx.doi.org/10.1155/2010/818190

[78] E. Grigorescu, C. Kulatunga, and G. Fairhurst, “Evaluation of the impact of packet drops due to AQM in capacity limited paths,” in *Capacity Sharing Workshop (CSWS’13)*. Göttingen: IEEE, Oct. 2013.

[79] L. DiCioccio, R. Teixeira, M. May, and C. Kreibich, “Probe and pray: Using UPnP for home network measurements,” in *Proceedings of the 13th Passive and Active Measurement Conference (PAM)*, ser. Lecture Notes in Computer Science, no. 7192. Vienna: Springer, Mar. 2012, pp. 96–105.

[80] K. Ramakrishnan, S. Floyd, and D. Black, “The Addition of Explicit Congestion Notification (ECN) to IP,” RFC 3168 (Proposed Standard), Internet Engineering Task Force, Sep. 2001, updated by RFCs 4301, 6040. [Online]. Available: http://www.ietf.org/rfc/rfc3168.txt

[81] 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