Arne Lie

Enhancing Rate Adaptive IP Streaming Media Performance with the use of Active Queue Management

Doctoral thesis for the degree of doktor ingeniør

Trondheim, April 2008

Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Telematics

NTNU

NTNU
Norwegian University of Science and Technology

Doctoral thesis for the degree of doktor ingeniør

Faculty of Information Technology, Mathematics and Electrical Engineering (IME)
Department of Telematics

ISBN 978-82-471-7457-9 (printed version)
ISBN 978-82-471-7460-9 (electronic version)
ISSN 1503-8181

Doctoral theses at NTNU, 2008:74

Printed by NTNU-trykk

Abstract

The Internet is today a world wide packet switching arena constituting enormous possibilities of new services and business creation. E.g., there is a clear tendency that more and more real-time services are making the jump from dedicated circuit-switched and broadcasting networks into packet switching. Examples are telephony, videoconferencing, and television. The Internet today is thus hosting a large set of different services, including the delay tolerant Web-surfing traffic, but also the non-delay tolerant real-time services. An additional challenge with most real-time traffic is that its traffic pattern do not adapt to the varying traffic load as Web-traffic do. Still, these new services work well, as long as the packet switching capacity is sufficient. Problems arise when the growth of real-time service usage is larger than the capacity increase. During peak hours, users will then start to experience media services fall-out and excessive communication delay.

The reason is that the Internet as we know it today was not built to handle such services at all. In motor traffic, as a comparison, queues build up when the traffic load is larger than the road and crossover capacity. The Internet behaves in a similar fashion: information is sent in packets that can be compared to cars. If too many packets are heading the same direction, queues of packets build up in the Internet routers, causing extra delay during such peak hours. In one way the Internet is more fearful than motor traffic: if queues get too long, new arriving packets are simply dropped, i.e. they just vanish. Luckily, there is no direct parallel to this phenomenon in the motor traffic comparison realm! To assist the queuing problems in motor traffic, special traffic lanes can be defined to allow e.g. only buses, taxis, and cars where the driver has at least one passenger, to drive in that lane. Thus, these road-users will experience less delay in peak hours than the rest of the population. The Internet is tried “healed” with some comparable means. E.g. with the use of IntServ or DiffServ Quality of Service, packets belonging to high priority applications are treated in a preferential fashion. But what happens if too many applications start to use these “special-lanes”? What if the total capacity is over-loaded over a significant time period? The answer to fix the problem is simple: the aggregate traffic generation must

slow down! In motor traffic, this means that each car carries more people (i.e. fewer cars in total), or, equivalently, that big cars are exchanged by smaller cars, thus producing smaller queues. In the multimedia real-time packet switching realm, the equivalent solution is that the same content must be compressed more efficiently, thus producing fewer and/or smaller packets.

This thesis proposes *a solution for live interactive real-time streaming media where a tight interaction between the media sources and the network is very essential*. A novel router architecture, "P-AQM", for packet switched networks is its core component. The primary P-AQM design objective is native support for rate adaptive real-time multimedia flows, addressing low queuing delay and low packet losses even at high traffic load to assist conversational media flows. The second objective is bandwidth fairness among the media flows, but also fairness to elastic (TCP) flows. These two design objectives are achieved due to the aforementioned interaction between the network routers and the traffic sources: the routers signal the traffic congestion level, while the media and TCP sources apply rate adaption. TCP has built-in congestion control mechanism (e.g. Tahoe or Reno) that reacts on packet drops or packet ECN tags performed by the router. Real-time media using the UDP protocol has no standardized congestion control mechanism. While DCCP/TFRC has become a compelling IETF standard during the last years, the work of this thesis has chosen another solution for media rate control that bypasses the TFRC performance. Using the traffic congestion level signals from P-AQM routers, the media rate control can be done much more precise, react faster to traffic load changes, and obtain intra-flow global max-min fairness. The cost of these improvements is gradual deployment of the new P-AQM packet switching routers, and some added signaling traffic.

The P-AQM design is following classical control theory principles, and has been developed and improved using a combination of analytical and simulation tools. As a side effect to the need for true decodable rate adaptive video traffic, a simulation framework and tool-set, "Evalvid-RA", was developed to generate such traffic. Evalvid-RA can also assist other researchers in improving their own work, e.g. applying rate adaptive video codecs over the DCCP/TFRC protocol.

Preface

This dissertation is submitted in partial fulfillment of the requirements for the degree of *doktor ingeniør* at the Department of Telematics, Norwegian University of Science and Technology (NTNU). The studies have been carried out in the period from May 2001 to December 2007, under the supervision of Professor Leif Arne Rønningen. The work includes the equivalent of a year of full-time course studies. In the period May 2001 until May 2005 the work was funded by a scholarship from the Research Council of Norway, via the IKT2010 project *Universal Multimedia Access (UMA)*, and SINTEF ICT. With still more unexplored ideas as of spring 2005, the remaining work has been finalized in spare time.

Production note: this thesis document was created with Adobe FrameMaker 7.2, enhanced with CiteMaker 1.2 for Bibtex support, and with LaImport.dll of Nigel Horspool for Latex to FrameMaker conversion of Paper E. Simulation results are obtained with Demos/Simula and ns-2.28. Result post-processing and plots are made with Matlab 7.

Acknowledgements

A lot of people have had important contributions on content and motivation of my thesis. Leif Arne Rønningen, my main supervisor, whose stubbornness and initial ideas and motivation convinced me to start the thesis work in the first place. Andrew Perkis, my second supervisor, whose ISO MPEG and JPEG committee engagement called by first excitement for multimedia research topics. My fellow Ph.D. students within the Universal Multimedia Access (UMA) project, Jijun Zhang and Svein Høier; together we created the first collaboration and discussion team, ranging from motion picture film form and film cues, to new possibilities of MPEG-4 arbitrary shaped objects and MPEG-21 content framing and adaptation. In these discussions also Johan Magnus Elvemo and Aud Sissel Hoel contributed with vital knowledge. Bjarne Kjøsnes, whose engagement in Midgard

Media Lab and related activities has been an always present source of vitality and inspiration. Ole Morten Aamo, who by destiny became my chosen cybernetics expert and co-author when I needed one. Ph.D. student fellows Odd Inge Hillestad and Stian Johansen with their contributions on video and audio compression standards and research. Jirka Klaue, whom I have never met in person, but still feel to know very well, through our collaboration using e-mails and SVN tools, in our work on the Evalvid-RA tool and paper.

The MPEG-4 company Envivio that I visited in Rennes, France, and San Francisco, USA. I met several inspiring persons, among them Guillaume Cohen, Cyrille Berson, Gall Le Garrec, Tim Boucher, Yuval Fischer, and last but not least Zia Rahim. During my San Francisco stay autumn 2004 I also had the pleasure of meeting Sally Floyd at her office in Berkeley. Floyd gave me several important feedbacks on my work up to that point, and she made me aware of other related research, such as the XCP work by Dina Katabi.

Of other researchers that I have learned a lot reading their papers, but also discussing with them using e-mail, and thus have colored this work in different ways, should be mentioned Nandita Dukkipati (Rate Control Protocol), Paul Hurley (Alternative Best Effort), James Roberts (France Telecom) whom I also had the pleasure of meeting twice, Maher Hamdi (SVBR), Bartek Wydrowski (MaxNet), and Ke Chih-Heng (NCKU Taiwan), whose work on interfacing Jirka's EvalVid to ns-2 made my work with the rate adaptive version of the ns-2 framework so much more manageable. I also want to thank the anonymous reviewers that have read my paper contributions for their creative suggestions on improvements.

I have to thank all the people at the NTNU Department of Telematics, and especially roommate Richard Sanders, Pål Sæther, Asbjørn Karstensen, Jarle Kotsbak, Harald Øverby, Andreas Kimsås, Mazen Malek Shiaa, Poul Heegaard, Kjersti Moldeklev, Bjarne Helvik, Norvald Stol, Yuming Jiang, Steinar Andresen, and Randi Flønes. At NTNU Q2S I would like to thank Erling Austreim, Eren Gürses, Otto Wittner, and Peder Emstad. And, I am very grateful to my employer SINTEF ICT and my research director Erik Kampenhøy, who together with the The Research Council of Norway made this thesis project possible.

I wish to thank my mother and my father, for always being encouraging and loving during my whole life. And last, but not least, to my wife Heidi and our daughters Ida and Maja, whose patience and understanding have given me the opportunity of finishing this work, well behind initial time schedule.

List of Papers

The papers published during the thesis work are listed below. Bold numbers mark papers included in Part II of this thesis. A graphical overview is given in Figure 5.5 at page 62.

[1] Leif Arne Rønningen, Arne Lie, "Transient Behaviour of an Adaptive Traffic Control Scheme", presented at *EUNICE'02*, Trondheim, Sept. 2002. ([RL02b])

[2] Leif Arne Rønningen, Arne Lie, "Performance Control Of High-Capacity IP Networks For Collaborative Virtual Environments", In *IBC 2002 Conference Proceedings*, Amsterdam, Netherlands, 12–15 Sept. 2002. (**Paper A**, [RL02a])

[3] Arne Lie and Leif Arne Rønningen, "Distributed Multimedia Plays with QoS guaranties over IP", In Proc. of *IEEE Wedelmusic'03*, ISBN 0-7695-1935-0, Leeds UK, 14–17 Sept. 2003. (Appendix A, [LR03])

[4] Arne Lie, Ole Morten Aamo, Leif Arne Rønningen, "On the use of classical control system based AQM for rate adaptive streaming media", In Proceedings of *17th Nordic Teletraffic Seminar*, ISBN 82-423-0595-1, Fornebu Norway, August 2004. (**Paper B**, [LAR04a])

[5] Arne Lie, Ole Morten Aamo, Leif Arne Rønningen, "Optimization of Active Queue Management based on Proportional Control System", In Proceedings of *IASTED Communications, Internet, and Information Technology (CIIT'04)*, ISBN 0-88986-445-4, Virgin Islands, Nov. 2004. (**Paper C**, [LAR04b])

[6] Arne Lie, Ole Morten Aamo, Leif Arne Rønningen, "A performance comparison study of DCCP and a method with non-binary congestion metrics for streaming media rate control", In *Proceedings of the 19th International Teletraffic Congress (ITC'19)*, ISBN 7-5635-1141-5, Beijing University Post and Telecommunications Press, Beijing China, 29. August – 2. September, 2005. (**Paper D**, [LAR05])

[7] Arne Lie, Jirka Klaue, "Evalvid-RA: Trace Driven Simulation of Rate Adaptive MPEG-4 VBR Video", *ACM/Springer Multimedia Systems Journal*, online Nov. 2007, in print 2008. (**Paper E**, [LK07])

[8] Arne Lie, "P-AQM: low-delay max-min fairness streaming of scalable real-time CBR and VBR media", In Proceedings of *IASTED EuroIMSA'08 conference*, Innsbruck, Austria, 17–19 March 2008. (**Paper F**, [Lie08])

Contents

List of Tables

Paper A

Paper B

Paper E

List of Figures

Paper A

Paper B

Paper C

Paper D

Paper E

Paper F

List of Acronyms and Glossary

Below follows a list of the acronyms and abbreviations used in this thesis. Acronyms marked with a star (*) are abbreviations for inventions of this thesis.

1080i	See HDTV
3G	Third Generation cellular networks, like UMTS and CDMA2000.
720p	See HDTV
AAC	Advanced Audio Coding
AAC-LD	AAC Low Delay Profile (20ms algorithmic delay)
ABE	Alternative Best Effort
ABR	Available Bit Rate
AC	Admission Control
ACF	Auto Correlation Function
ACK	Acknowledge packet
AIMD	Additive Increase Multiplicative Decrease
AQM	Active Queue Management
ARIMA	Integrated Auto-Regressive Moving Average — used for modelling data with both a non-stationary trend and a zero-mean stationary component
ASP	Advanced Simple Profile (an MPEG-4 profile)
ATM	Asynchronous Transfer Mode
AVC	Advanced Video Coding. ISO/MPEG video codec, successor of MPEG-1, MPEG-2 and MPEG-4 Part 2. Developed jointly with ITU, also known as MPEG-4 Part 10, MPEG-4 AVC, and ITU-T H.264.
AVQ	Adaptive Virtual Queue
BDP	Bandwidth Delay Product
BEI	Best Effort Internet

BIC	Binary Increase Congestion control
B-ISDN	Broadband Integrated Services Digital Network
CBR	Constant Bit Rate
CBT	Class-Based Threshold
CDF	Cumulative Distribution Function
CDN	Content Delivery Network
CDMA	Code Division Multiple Access
CHOKe	CHOose and Keep for responsive flows, CHOose and Kill for unresponsive flows
CIF	Common Intermediate Format (352 x 288 pixel video frame size)
CSFQ	Core Stateless Fair Queuing
DCCP	Datagram Congestion Control Protocol
DCT	Discrete Cosine Transform
DIA	Digital Item Adaptation (in MPEG-21)
DiffServ	Differentiated Services
DNS	Domain Name Service
ECF*	Explicit Congestion Feedback
ECN	Explicit Congestion Notification
ERF*	Explicit Rate Feedback
Evalvid-RA*	Evaluation of video - Rate Adaptive version
F-ARIMA	Fractional ARIMA
ffmpeg	“fast forward MPEG”, a collection of software libraries that can record, convert and stream digital audio and video in numerous formats, among others MPEG-4.
FGN	Fractional Gaussian Noise
FGS	Fine Granularity Scalability
FIFO	First In First Out
fps	frames per second
FRED	Flow Random Early Detection
FTP	File Transfer Protocol
FQ	Fair Queuing
GFC-2	Generic Fairness Configuration version 2

GOP	Group of pictures (units are: number of frames per GOP or equivalent number of seconds per GOP, and number of bytes per GOP)
H.264	ITU-T video codec, successor of H.261 and H.263. Jointly developed with ISO/MPEG, also named AVC and MPEG-4 Part 10.
HDTV	High Definition TV. In Europe: 720p25: 1280 x 720 @ 25 fps 720p50: 1280 x 720 @ 50 fps 1080i: 1920 x 1080 @ 50 interlaced fields per second
HE-AAC	High Efficiency AAC
HTML	Hypertext Markup Language
HTTP	Hypertext Transport Protocol
ICMP (SQ)	Internet Control Message Protocol (Source Quench)
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IntServ	Integrated Services
IP	Internet Protocol
IPTV	SDTV quality streaming over IP
IR	Interlacing Resilience
ISO	International Standardisation Organization
ITU-T	Telecommunication Standardization Sector of the International Telecommunications Union, formerly known as CCITT.
JPEG	Joint Photographic Experts Group
LAN	Local Area Network
LB	Leaky Bucket
LDA+	Loss Delay based Adaptation
LRD	Long Range Dependence
M/M/c	Markovian input / Markovian service / c services / infinite waiting queue — n.e.d. distribution for the interarrival times of new customers, n.e.d. distribution for service times
M/M/c/c	Same as M/M/c, but no waiting queue (total of customers is c, and c servers, gives c-c=0 waiting positions)
MBAC	Measurement Based Admission Control
MMS	“Microsoft Media Server” — Microsoft's proprietary network streaming protocol, corresponds in functionality to IETF’s RTSP

MOS	Mean Opinion Score
MPEG	Moving Picture Expert Group. The video group within ISO
MPEG-1	First generation video codec from MPEG group, targeted “VHS”-quality video stored on CD.
MPEG-2	Second generation video codec from MPEG group, targeting digital TV (interlaced video).
MPEG-4	Third generation video codec from MPEG group, targeting low- and high-rate video and object segmented video.
MPEG-21	The multimedia framework (digital rights, transactions, content adaptation)
MPLS	Multi-protocol Label Switching
n.e.d.	negative exponential distribution
OC	Optical Carrier
OS	Operating System
OSI	Open Systems Interconnection
P-AQM*	Proportional Active Queue Management
PI	AQM based on Proportional-Integral control logic
PRNG	Pseudo Random Generator
PSNR	Peak Signal-to-Noise Ratio
PTP CADPC	Performance Transparency Protocol Congestion Avoidance with Distributed Proportional Control
QCIF	Quarter Common Intermediate Format (176 x 144 pixel video frame size)
QoS	Quality of Service
RAP	Rate Adaption Protocol
RCP	Rate Control Protocol
RED	Random Early Discard or Random Early Dropping or Random Early Detection
RED-PD	RED with Preferential Dropping
REM	Two different meanings: (i) Rate Envelope Multiplexing and (ii) Random Exponential Marking
RL-QoS*	Rønningen-Lie QoS
RR	Round Robin
RS	Rate Sharing
RSVP	Resource Reservation protocol

RTCP	Real-Time Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real-Time Streaming Protocol
RTT	Round Trip Time
SACK	Selective ACK
SCTP	Stream Control Transmission Protocol
SDTV	Standard Definition TV (PAL: 720 x 576 pixel video frame size @ 25 fps) (NTSC: 720 x 480 pixel video frame size @ 29.97 fps)
SMG	Statistical Multiplexing Gain
SMTP	Simple Mail Transfer Protocol
SNR	Signal-to-Noise Ratio
SRED	Stabilized RED
SSH	Secure Shell
SSIM	Structural SIMilarity
Streaming Server	A server that communicates over RTSP and/or MMS, as opposed to a Web Server that communicates over HTTP
SVC	Scalable Video Coding (in H.264/AVC)
TCP	Transmission Control Protocol
TEAR	TCP Emulation at Receivers
TFRC	TCP Friendly Rate Control
TOS	Type Of Service (field in IP header)
UBR	Unspecified bit Rate
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
VBR	Variable Bit Rate
VBR-nrt	VBR non-real-time (in ATM)
VBR-rt	VBR real-time (in ATM)
VBV	Video Buffer Verifier
VC	Virtual Circuit
VGA	Video Graphics Array (standard computer screen resolution of 640 x 480 pixels)
VoD	Video on demand

VoIP	Voice over IP
VPN	Virtual Private Network
WebTV	TV quality streaming over the Web
WLAN	Wireless LAN
WMA	(Microsoft) Windows Media Audio
WMV	(Microsoft) Windows Media Video
WWW	World Wide Web
XCP	eXplicit Congestion control Protocol
XML	eXtensible Markup Language
YUV	Defines a color space in terms of one luma (Y) and two chrominance components (U and V)

Part I — Introduction

Get your facts first, and then you can distort them as much as you please.

Mark Twain — US humorist, novelist, short story author, & wit (1835–1910)

1

Background and motivation

Your motivation? Your motivation is your pay packet on Friday. Now get on with it.

Noel Coward — English actor, dramatist, & songwriter (1899–1973)

The Internet became first widely known among the public after the invention of the World Wide Web (WWW), including the Hypertext Transfer Protocol (HTTP) and Hypertext Markup Language (HTML), originally suggested by the CERN ("Conseil Européen pour la Recherche Nucléaire") scientist Tim Berners-Lee, in order to improve the ability for co-operation among the nuclear researchers [BL89]. HTTP [BLFF96, FGM+99] became de facto standard protocol for how to access online documentation world-wide, starting with modest backbone network load numbers of 0.1% and 1.0% in March and September 1993 [Cre01], respectively, following an exponentially growth to 80–90% of total traffic load only a few years later. The scope of success both within the academic as well as in the commercial communities did surprise all, including the WWW inventors, and completely changed the methods as well as the possibilities for human interaction and co-operation.

HTTP is an application level protocol, and relies on the usage of a transport protocol providing a reliable channel between the communicating parties. The Transport Control Protocol (TCP, [CDS74, Pos81]) was the natural choice. The TCP set-up phase establishes a connection-oriented reliable ses-

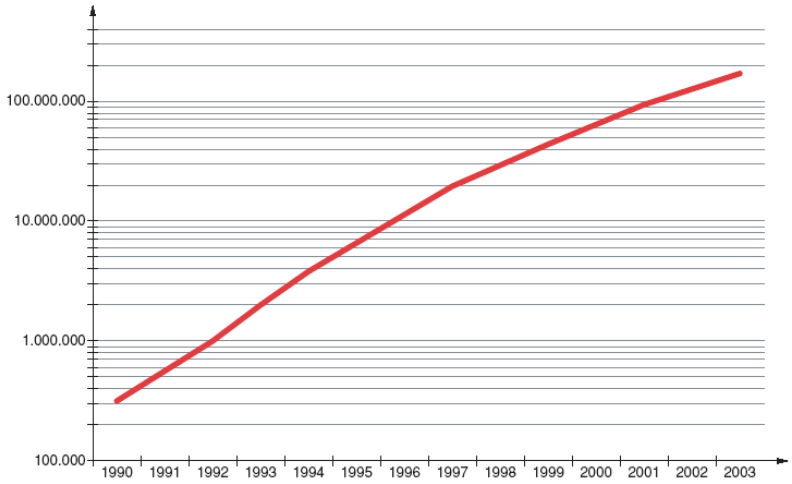

Figure 1.1: The exponentially growth of number of computers connected to the Internet in the period 1990–2003 [UNI03].

sion between the Web browser client and the HTTP server, for transparent delivery of online documents, formatted in HTML [RHJ99] or any other document formatting system. Since each Internet network *link* generally is reliable due to layer 2 mechanisms, the frames embedding the IP packets do get delivered correctly to the next hop router. However, IP packets arriving a full router queue will be dropped. This is why the Internet is often called "best effort". It is up to the end systems, and not the network, to detect packet losses, and decide if lost packets should be retransmitted. The TCP protocol includes such capabilities, thus it is called *reliable*. The File Transfer Protocol (FTP) also uses TCP, as well as e-mail SMTP and other applications. Typically ~90% of total traffic on Internet backbone links is TCP traffic [DJD05].

However, things are about to change, only 15 years after the birth of WWW. A new application type is starting to dominate the Internet more and more. It is called *streaming media*.

1.1 The challenges of continuous real-time streaming media

The term digital streaming media includes in general media content like video and audio and covers (but is not limited to) digital cinema, digital-TV, IPTV & WebTV, Internet Radio, Voice over IP, and Internet video streaming (live or pre-stored Video on Demand — VoD). This heterogeneous set of media sources, compressed in various formats such as ISO MPEG and Microsoft WMV, is streamed to different type of terminals, over a vast variety of network types, including wired modems and ADSL over telephone networks, 10–1000Mbit/s IEEE 802.3 Ethernet, IEEE 1394 FireWire, gigabit fibre optical networks, but also wireless IEEE 802.11 (WLAN), 802.15 (Bluetooth, ZigBee), 802.16 (WiMax) and ITU cellular systems like UMTS and CDMA2000. Actually, 3G/UMTS services are currently being deployed where videoconferencing can be established between 3G and Internet clients. The traffic volume of Internet streaming media is reported rapidly increasing [KW02, DPR02].[1] The deployment of new audiovisual 3G services will certainly not slow down that tendency, rather on the contrary. Although this thesis will focus on the carriage of digital streaming media over the wired Internet, the provided solutions should also benefit wireless packet switching.

The heterogeneous set of media formats, terminals and networks creates a large matrix of different combinations, all made possible by the single common factor: the Internet Protocol (IP). Unlike the circuit switched telephone network, where an established connection is granting some fixed capacity and latency, the Internet is a packet switched network where information flows in datagrams (i.e. finite but variable sized packets with source and destination information headers). These are statistically multiplexed in order

1. A comprehensive commercial forecast report can be purchased from http://www.insight-corp.com/reports/streaming.asp

to share common resources such as queues and connection links. Varying traffic load creates dynamics in the number of flows sharing common resources, thus the bandwidth granted to each flow is an uncontrolled variable. There are generally no restrictions in the number of simultaneous flows; nobody gets the "busy signal".[2] To avoid traffic overload, the TCP protocol has built in end-to-end congestion control mechanisms, which will be described in more detail later. First, let's focus on the difference between *elastic* and *real-time* content.

Elastic content is content which do not hold real-time delivery requirements. Whether e.g. an HTML document is completely delivered in one or three seconds after the user requests it does not matter anything for the value of the content: the information of the document is still durable. Thus it is elastic in the sense that its delivery time can be stretched without altering the information value (but the waiting time do have an upper bound due to user impatience [YdV01]).

Real-time content however *does* hold real-time requirements, as the name implies. This is so because it can be described as a continuous flow of *information change*, in contrast to static content like HTML documents. I.e., documents are elastic content, while streaming media is real-time content. Whether the real-time content can tolerate packet loss or not makes a sub-classification: tolerant and intolerant. An example of real-time intolerant content is control signals for robots. Audiovisual content, however, even in compressed form, can generally tolerate some degree of packet loss without displaying too noticeable artifacts.

Table 1.1: The different requirements of elastic and real-time tolerant content

requirements	Elastic content	Real-time tolerant content
latency	no	yes
unidirectional	–	<10 s
voice conversation	–	<150 ms [Int96]
musical collaboration	–	5–20 ms [CG04]
transparent channel	yes	no
packet loss ratio	0% (through retransmissions)	<5 % (no retransmissions) MPEG-4, [HWZ+99]
bandwidth	no	yes
average	–	>average stream bit rate
spare	–	>stream bit rate peaks

2. the only "exception" being if using dial-up modems, since an ISP has a fixed number of modems available in a modem pool. But in reality, this again is a restriction of the circuit switched network part of the communication channel, and not the Internet.

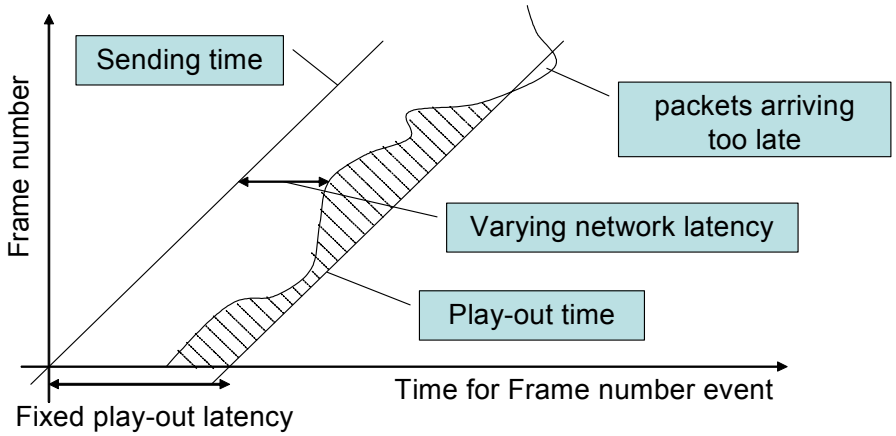

Figure 1.2: Depiction of the latency requirements of streaming media.

Table 1.1 gives a short overview of the main differences between elastic and real-time content. For *unidirectional* (non-conversational) streaming, like WebTV broadcasting and Video on demand (VoD) services, there is in reality no hard latency deadline for the information, since there is no interaction involved like in a voice conversation. However, in order to experience the feeling of "instant availability" when starting a new service, e.g. zapping WebTV channels, the initial delay should be upward limited to just a few seconds. However, the delay variance, or jitter, is of more concern, as shown in Figure 1.2. For *conversational* media, like Internet Telephony (Voice over IP, VoIP) and videoconferencing, the same requirements exist as for general telephony, which claims 150 ms one-way latency at maximum [Int96]. Research has also revealed that for musical collaboration, i.e. musicians that practice and perform live music over the Internet, the overall latency must resemble the audio latency experienced at concert stages, i.e. five to twenty milliseconds [CG04]. Actually, even for VoIP applications, the Distributed Open Signaling Architecture recommended 10ms queuing delay as maximum for US coast-to-coast VoIP [GGK+99].

Figure 1.2 gives an assembly of the most important metrics of streaming media. The *y*-axis gives the sequential frame numbers (for 25 fps video there is one frame per 40 ms), and the *x*-axis is the time given for different frame *events*. In the figure, three frame events are given: sending time (left straight line), receiving time (middle curved line), and play-out time (right straight time). In order to balance the variable network latency, the receiving terminal includes a "receiver buffer".[3] The shaded area depicts the amount of information packets (here measured in time) contained in this buffer. If the network latency grows too much, the buffer is emptied, and play-out rendering is prohibited (as exemplified in the upper part of the figure). The conflicting requirements are to have a small buffer in order to have overall low end-to-end latency, and at the same time limit the

3. Later in this thesis a "sender buffer" will also be discussed.

occurrences of buffer underruns to less than e.g. 2% of the time. For VoD services, one can implement the trick of having a small initial buffer latency to enable the "instant on" perception. However, during the first phase, the streaming bit rate is higher than the consumption bit rate (if available bandwidth is larger than media bit rate), thus increasing the amount of information packets contained in the buffer. Apple QuickTime™ buffers typically 10 seconds of data in its buffer (www.apple.com/quicktime), and after having done that, the streaming rate is controlled to equal the consumption rate. In this way, the streaming service can tolerate longer bursts of increased latency, than if it kept only the initial buffer of one second of media content. This possibility is however not available for live services, since the streaming server can not transmit information not yet created. Live sources for one-way streaming such as WebTV can instead slow down the rendering time in some period (see Chapter 4.3.1 on page 41).

As stated earlier, packets might be dropped, as well as being delayed due to queuing. Typically, for real-time services there is no time for retransmission of dropped packets. In fact, the UDP protocol does not include any mechanisms for retransmission. However, some vendors implement proprietary solutions, often named "reliable UDP" on top of the IP stack to do exactly this. If the client media player has buffered 10 seconds of media, but detects that one packet scheduled for rendering in eight seconds is still missing, there is plenty of time to do retransmission. E.g. QuickTime uses RTCP APP (Application Defined) packets for acknowledgement of received packets, and missing acknowledgements may trigger retransmissions [App03, SCFJ03]. Actually a "reliable UDP" based on [VHS84, PH90] has been proposed to IETF, but has not yet received finalization [BK99]. *This thesis focus on live interactive content like voice and videoconferencing with short delays, and retransmission is therefore not an issue.*

The last issue included in this introduction to main streaming media challenges, is about the media sources themselves. Continuous media like video and audio is normally compressed in order to achieve high bandwidth efficiency. As an audio example, uncompressed CD quality audio has a bit rate of 1.41 Mbit/s (44.1 kHz sampling, two channels, and 16 bits sample resolution gives $44100 \times 2 \times 16 = 1411200$ bit/s). The popular MP3 compression format can obtain near-CD-quality audio at a CBR rate of 128 kbit/s (rule-of-thumb), while even more recent technologies can achieve near-CD-quality at 80 kbit/s (AAC), 64 kbit/s (WMA) and 48 kbit/s (AAC-HE) [Sto03]. However, the actual perceptual quality depends on the complexity of the audio. Generally, classical music needs a higher bit rate (i.e. more information) in order to prevent quality loss, compared to e.g. pop music. As a video example, standard definition European TV (PAL) digitized (SDTV) need 124 Mbit/s (720 horizontal pels, 576 vertical pels, 12 bits/pixel (assuming YUV 4:2:0) and 25 full frames per second gives 720 x 576 x 12 x 25 = 124.416 Mbit/s). State-of-the-art codecs like MPEG-4 ASP can compress at a rate of 1.5 Mbit/s without loosing too much noticeable details [Vat05], H.264 (MPEG-4 AVC)

might give similar results at a rate of 1.0 Mbit/s [Vat06]. Again, the actual needed bit rate to obtain a certain quality depends on the content: fine detailed and/or high motion scenes need more bits than coarser and slow moving scenes. Thus, audiovisual encoding produces variable rate traffic.

However, with the use of a *rate controller*, the codecs can be run in different modes [SR01]:

- VBR open-loop (constant quality, no bit rate control),
- VBR constrained (steady quality, average bit rate control),
- and CBR (variable quality, constant bit rate).

The VBR open-loop mode provides the largest potential for *statistical multiplexing gain* (SMG), while the CBR mode will have zero SMG. SMG is a measure describing the potential of serving more VBR flows than CBR flows given a total capacity and quality per flow. Uncorrelated VBR flows will tend to fill and exploit each others rate variability, while CBR, in the absence of such possibilities, also must have high enough rate to produce adequate quality of the most complex frames. Thus, there is a desire of using VBR open-loop with two motivations: constant perceptual quality, and high SMG, i.e. the largest possible number of simultaneous audiovisual sessions given a requested quality. However, from a network point of view, these VBR open-loop sources are the most difficult to transmit. The reason for the latter is the possibility of simultaneous occurrence of high bit rates of multiple uncorrelated flows, and the uncontrollable length of such high bit rate scenes. This can lead to excess delays due to queue build-up at routers and switches, and possible significant packet loss events. CBR sources, on the other hand, are easier for the network to transport, since they do not behave unpredictable. The *VBR constrained rate control* provides thus a very inviting compromise, in that it ensures that the rate variability is reduced and that the rate control targets a specified average rate. This average rate is calculated typically over several GOP periods. VBR constrained rate control can also gain SMG, although not so much as VBR open-loop. Such SMG will be demonstrated in Paper E.

To summarize this first subchapter: the main challenges in carrying streaming media over a packet switched network is to support variable bit rate sources with some latency and bandwidth guarantee. The question is if this is feasible within a congested Best Effort Internet. As long as the traffic load cannot be controlled (as with the telephone network), a first intuitive answer is “no”. To answer these challenges the IETF set forth new concepts that should try to mix the best qualities of the Internet and the circuit switched telephone network. The new buzzword was: *QoS*.

1.2 QoS — Quality of Service

During the 1980's and 1990's there was a significant focus on the research leading up to ATM (Asynchronous Transfer Mode) network standard, initiated by the ITU-T and promoted by the ATM Forum. It was believed that ATM, as a standard for both layer two and three of the OSI protocol stack, would become a key component of future broadband services named B-ISDN, and provide access end-to-end for the users. The traditional telecommunication networks provide reliable but rigid systems. The goal was to obtain both the reliability of the telephone circuit switched networks, and the flexibility of packet switched networks. Two important tools to reach this goal were fixed sized packets named *cells* (53 bytes), and *virtual circuits* (VC). A lot of research efforts were put into the goal of supporting native Quality of Service inside the network, as contrasted by the best-effort IP (Internet Protocol) technology. Due to the native QoS support, ATM would be far better in providing services for video streaming, which was believed would become the dominating traffic class of ATM networks [HRR97, LSS01]. In order to meet the QoS requirements, the basic idea was to establish resource reservation at session startup, and admit connections only if the resource requests could be met without ruin the QoS parameters for already established connections. Thus, Admission Control and calculation of Effective Bandwidth were two key components of ATM.

Due to the focus of flexibility *and* QoS put into the telecom ATM baby, the IETF answered with its "Integrated Services", or IntServ, initiative. Thus, where the telecom community was moving from the QoS solid and rigid circuit switched technology towards the much more flexible but QoS oriented ATM, the Internet community moved in the opposite direction, but targeting the same goal: the flexibility of statistical multiplexing while at the same time providing QoS support. The best effort Internet Protocol should provide QoS through a set of upper layer technologies. "Integrated Services" reflected that the Internet should support a range of different qualities integrated into one network, from the unreliable best-effort service in one end, towards absolute QoS support in the other end. There should be no need for deploying more than one network to support these various services. In the support of QoS, the focus was now on how also Internet sources could be characterized by a limited parameter set, and to provide Admission Control (aka the telephone system caller blocking probability) to control that the Quality of Service stochastic or deterministic guarantees could be met. The idea was that better quality should be priced higher than lower quality. Indeed, the Internet and telecommunication communities have worked out an agreement in how to map the different service classes between IntServ and ATM [Ber98a, Ber98b, GB98, CBB+98]. However, the Internet had always lacked a credit system, which the telephone systems always have had. IntServ failed in deployment both due to this fault, as well as the problems with service scalability.[4] Put

4. in fact, each flow should be set-up and monitored by the RSVP protocol, to provide per-flow QoS

shortly, IntServ became way too complex to be implemented worldwide. Also ATM failed as an end-to-end network architecture, but does exist in some networks with IP on top.

Due to the complexity of resource reservations of IntServ, a much more scalable QoS scheme was promoted by IETF through the DiffServ initiative. Through relaxing the possible QoS guarantees, DiffServ became much less complex. While IntServ could provide end-to-end service guarantees, DiffServ was limited to hop-by-hop behavior. An ingress router marks each flow (using the IP TOS field) as belonging to one specific *service class.* This stateless concept is much more scalable, also because the traffic flows are aggregated into a few number of such classes. It is then up to each router and switch on each hop from source to destination to decide how this class is treated relative to the other defined classes. Thus, DiffServ is often labeled as "relative QoS" or "prioritize based QoS", while IntServ is often labeled as "absolute QoS" or "reservation based QoS". While DiffServ proves to be a much more scalable solution on the price of relaxed QoS guarantees, the credit facility still remains an unresolved issue.

Due to the postponement of deploying QoS models into the Internet, and also the failure in providing ATM services "to the desktop", the research focus has again shifted to best-effort Internet (BEI). BEI can be compared to the UBR (Unspecified bit Rate) class of ATM. This also means that there is a focus shift from proactive (or preventive) traffic control to reactive (congestion controlled) traffic control. *This thesis focus in fact on the latter, with enhanced traffic load measurements provided by special designed Active Queue Management routers named P-AQM, which is the main contribution of this work.* More details on this AQM will be given later in Part I, starting at Chapter 5.2, and of course in the papers listed in Part II.

Before ending this subchapter, it must also be mentioned that MPLS (Multiprotocol Label Switching) [RVC01] has gained much attention during the last years as a commercial underlying QoS mechanism [GB05] for business-to-business communications, such as the carrying of videoconferencing real-time media. It is regarded as a "Layer 2.5" technique that, as ATM, uses connection-oriented services provided by the RSVP protocol [Wro97, ABG+01], or the alternative LDP (Label Distribution Protocol). RSVP and LDP are used to set up the label switched path through the network. The MPLS 32-bit header carries a 20 bit label and a 3 bit QoS priority value. Thus, the key elements are to set up an efficient routing path as in IntServ, and use relative QoS metrics as in DiffServ. MPLS VPN sessions are typically set up by a service provider such as AT&T and TelePacific in the US. The MPLS strategy of providing real-time media QoS is out of the scope for this thesis.

Finally, a smart technique for pre-stored media must be mentioned: mirroring. CDN (Content Delivery Network) has become a large industry for companies such as Akamai and others using *overlay networks*. The simple idea is it mirror large amount of content to

world-wide distributed servers. The users are unaware of their location, because of an invisible DNS name to IP address mapping, that makes sure that the end user is redirected to the server having the best network connection path towards the user. Since this thesis focuses on live media, CDN networks are also out of the scope.

1.3 The challenge of mixing elastic and real-time media

The main difference between elastic and real-time media is the delay requirements, and that elastic applications need a transparent layer 4 channel while real-time media do not. Hurley [HBTK01] showed that a packet scheduler could be designed so that these two different preferences could be supported in the router itself (ABE) by a simple algorithm.[5] The key idea was that, based on the competition of resources in a best-effort network, the network resources should be granted fairly, i.e. the sources had to make a choice: If the application is a real-time constrained application, it should mark its packets "green", while if it is an elastic application it should mark its packets "blue". An ABE router will schedule the green packets with small delay, on the cost of somewhat higher probability of packet loss or packet ECN marks, using a sophisticated two-queue scheduler. By balancing delay and throughput, this solution can be said to still belong to the best effort regime, in that there is still a "flat cost", i.e. only one cost class.

Put another way, elastic applications can "buy" transparency on the cost of added delay (larger buffer sizes and/or more retransmissions), while real-time media can "buy" shorter delay on the cost of higher loss probability. Since lost TCP packets result in retransmission, which in fact is the same as adding extra traffic into the already congested network (we do not consider packet loss due to wireless communication, or error-prone wired lines here), larger buffers for TCP flows can be a good solution. In fact, legacy FIFO routers are most often configured with BDP (defined below) sized queue buffer sizes (more on buffer sizing in Chapter 6.1 on page 71), so that a single TCP flow is able to maintain high link utilization [VS94]. The reason is that the TCP congestion control algorithm increases the flows throughput by one extra packet per RTT until there is a packet drop (or mark) event, in which the TCP congestion window is halved. The time it takes to drain a BDP sized queue is BDP divided on the outbound link capacity C,

$$\frac{\text{BDP}}{C} = \frac{C \times RTT}{C} = RTT \text{ (s)}, \tag{1.1}$$

i.e. one RTT, where RTT is the RTT used when defining the BDP for that router. In Figure 1.3 the RTT of the flow matches exactly with the RTT used in the BDP, thus the queue is completely drained before immediate increase. The advertised window size from

5. However, there is no explicit mechanisms in that solution as how to assist rate adaptation in the media sources.

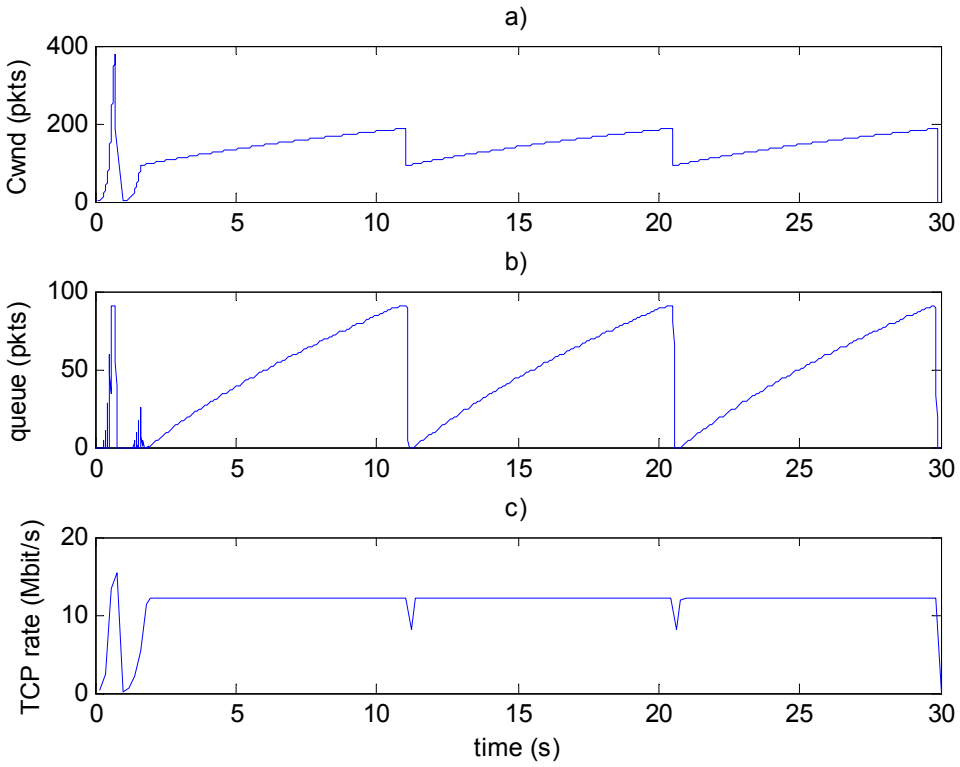

Figure 1.3: TCP congestion window (a), router queue configured to BDP (b), and resulting TCP sending rate (c) for single TCP flow over 12Mbit/s link with 65ms RTT (propagation delay). TCP New Reno simulated in ns-2.

the receiving end must be higher than the local congestion window to enable this source to "fill the pipe". The characteristic saw-tooth shape is unmistakable, and the link capacity is kept very close to 100%.[6] Using a shorter buffer to limit the delay will give significant drop in link utilization. For this reason, legacy FIFO routers are often configured with an RTT = 200 ms in the BDP to enable also long distance flows to reach high link utilization. Such buffer sizes are however unnecessary large when multiple greedy flows share the same link, except when packet drop events are synchronized. The latter will be the case when the flows have identical RTT, because their congestion windows will increase at the same speed, and FIFO routers drop packets on full buffer.

If the packet drops were more randomized, even at similar RTTs, the buffer size can be reduced. RED routers have exactly this property [FJ97]. However, reducing the buffer size with more than factor two is difficult without risking traffic load "oscillations",

6. Due to the BDP sized queue and single TCP flow, the TCP *rate* does not have the typical saw-tooth shape in this example. This is due to that the queue is almost never drained, and that the RTT is doubled from queue empty until queue is full. The received ACK packets clocks the AIMD, and the ACK packets are clocked by the receiving TCP packets, which are rate limited to 12Mbit/s.

resulting in reduced link utilization [OLW99, MBDL99]. RED is the best known AQM router commercially available, and as already mentioned, the main solution to the rate control challenge of streaming media of this thesis is based on AQM supported congestion control.

Even 100 ms network delay may be too much for interactive real-time traffic, in that there might be multiple congested links, and that there is additional delay at media sender (codec, rate controller, optional sender buffer) and receiver (receiver buffer). To obtain sufficiently low delay for the real-time traffic, its IP packets can not be scheduled through the same queue as the TCP packets. The question is then: is it possible to design a solution without using QoS techniques such as IntServ or DiffServ? ABE, as already discussed, gives a solution for best effort Internet that lowers the delay for the media ("green") packets. But there is nothing in ABE that can quantify *how much* lower the delay will be, nor how the media source rate controller shall balance its throughput to maintain low delay at an acceptable packet drop ratio.

Obviously, the solution needs to make sure that the UDP packets (or "green" packets or whatever transport packet is used) are not backlogged in too long queues at routers and switches.

Proposition 1*: Media IP packets will experience low queuing delay and low packet loss ratio even at link utilization close to 100% if*

1. *network routers/switches are configured with small maximum queue sizes*
2. *the network traffic load feedback is precise enough*
3. *there is media rate control reacting on network traffic load feedback, which design is based on seeing the source-network-destination path as an integrated system.*

This thesis will prove this proposition by designing such a system, based on an AQM type of router with primary support of real-time flows. The design is built on classical control theory analysis since the network with the feedback signaling in fact defines a control loop. Elastic TCP traffic and variants have been analyzed in such manners by several researchers [KL03, HMTG01]. To the best of my knowledge, such a proposition is not claimed up to now for media traffic. One major reason why Proposition 1 can be fulfilled is the following:

Lemma 1*: The aggregate of VBR constrained media flows does not exhibit long range dependency (LRD).*

This lemma was proved by Hamdi et al. [HRR97]. This thesis must also show that this holds for rate adaptive constrained VBR.

1.4 Summary — This Thesis Challenge

To conclude this introductory chapter, the follow paragraph summarizes the challenge for *live interactive streaming media communication* in which this thesis will focus, analyze, and propose solutions for.

> *The challenge is to control the streaming media transmission rate so that the total system of aggregated flows allows for high link utilization, and at the same time gives acceptable low drop probability and low delay with the use of short buffers. This thesis will investigate the communion of streaming media traffic with elastic traffic in a best-effort network. Parts of the solutions should be adaptable for QoS enabled networks as well. The streaming media traffic must be characterized, and solutions must be provided for how to control the media, both at the sources and in the network, in a scalable manner.*

1.5 Outline of thesis

Part I gives an introduction to the fields of research relevant to this thesis. It is rather comprehensive since the research focus is somewhat multi-disciplinary. Part II constitute the contributions of this thesis, i.e. the papers. Part III is the Appendix, listing one additional paper (Appendix A), some unpublished research related to video error resilience (Appendix B), and pseudo code of the main contributions of this thesis, i.e. the network AQM algorithm and the video rate control at the sources (Appendix C).

Part I is organized in the following way: After this introductory Chapter 1 follows Chapter 2 where the differences and similarities of elastic and real-time traffic characteristics are explained, and how the choice of rate controller colors the video traffic. The LRD issue is explained here. The possible methods for media scaling are covered in Chapter 3, while Chapter 4 outlines the possible solutions for how the network design can influence the media scaling. The last part of Chapter 4 identifies the *requirements* that the envisaged system should hold. Chapter 5 presents the listed papers in Part II, after an introduction to the main fields of research of this thesis, which is AQM assisted congestion control. Part I is closed by Chapter 6, giving some additional discussion of the proposed architecture and open issues, and some more details and ideas for future work not mentioned in the papers. Part I has its own Bibliography found in Chapter 7 on page 79.

2

Audiovisual traffic characteristics

Traffic signals in New York are just rough guidelines.

David Letterman — US comedian & television host (1947–)

Generally, practical traffic sources have variable bit rate, and the variance might be noticeable even when averaging over different time scale orders. This applies both to elastic data sources [LWTW93], and continuous real-time audiovisual sources [GW94]. Thus, the aggregate traffic is also very dynamic, and network design must take this into account. Bufferless multiplexing [RMV96] is ideal when small delay jitter is of great importance. However, the traffic load in such networks must be kept low in order to avoid packet loss. In the other extreme, if high link utilization and robustness to packet drops are of great concern, large buffers at routers and switches should be applied. However, if traffic bursts occur over many time scales, even extreme large buffers might be overflooded. Thus, in network design, there must be a balanced budget of link capacity and buffer sizes based on assumed traffic characteristics.

The data sources are often divided into two classes when it comes to session *duration*: short-lived and long-lived. An example of short lived traffic is Web browsing data traffic, where HTML pages of often small sizes are downloaded from Web servers to client browsers. Long-lived data traffic can be exemplified by the download of large data files using HTTP, FTP or SSH clients. Even in the latter case traffic is not smooth. This is due to the flow control and congestion control mechanisms of the TCP protocol, as was demonstrated in Chapter 1.3.

As already pointed out, audiovisual traffic is variable by nature [GW94]. This is due to the compression algorithms that strive to remove redundancy and irrelevance. Since the imaginary and aural information entropy changes over time, the compression algorithms can compress the content at variable bit rates, given a target quality. The harder the media is compressed, the lower is the average bit rate, and the lower is the resulting quality. In order to combat variable bit rates, the compression algorithm can make use of a *rate con-*

troller, which strives to keep a constant bit rate on the cost of variable quality and increased algorithmically delay.

The latter will be detailed more in Chapter 2.2. A more general discussion is given first in Chapter 2.1.

2.1 Poisson vs. self-similar

The Poisson modelling has been widely and successfully adopted for many telecommunication tasks, and was pioneered by Agner Krarup Erlang in 1909 [Erl09]. E.g., the time between telephone call establishment and service time per call session is well modelled by the negative exponential distribution, which is the time between events driven by a Poisson source [GH98]. In 1917 Erlang published his famous simple equations for call blocking and waiting probabilities, know as Erlang-B and Erlang-C formulas [Erl17], respectively. However, other results have had more impact on computer packet switched networks, such as the work of Pollaczek and Khintchine, who found that the expected waiting time in a one-service system with infinite buffer size is given by

$$E[W] = \frac{\lambda E[S^2]}{2(1-\rho)} \tag{2.1}$$

where $\rho = \lambda/\mu$ is the work load where λ is the input traffic intensity and μ^{-1} is the service time. $E[S^2]$ denotes the second moment of the service time S. Since the mean service time is included in the work load ρ, this equation, termed the *Pollaczek-Khintchine formula*, is valid and useful for any M/G/1 system where the work load and second moment of service time is known. If packet sizes are fixed to a deterministic value as in ATM, the second moment equals the square of the mean service time, which simplifies the needed information to utilize (2.1). Notice however that for (2.1) to be valid, the inter-arrival time between packets entering the system must be negative exponential distributed, i.e. the number of packets per time unit must be Poisson distributed.

Since the late 1980's there has been much debate on whether or not the classical teletraffic theory can be applied successfully on packet switched networks or not. In pioneering work performed by Leland et al. [LWTW93], it was discovered that the inter-arrival time between packets measured in 10Mbit/s LANs did not follow n.e.d. Analysis showed that when averaging and displaying the number of packets over different time-scales, visual *self-similarities* appeared. (A stochastic process is termed self-similar if its statistical properties are exactly the same at all time scales, up to some scaling factor.) If the number of packets per time unit had been Poisson distributed, averaging over increasing time units would produce smoother and smoother traffic. The measured time series $X = (X_t, t = 0, 1, 2, \ldots)$ however showed burstiness over four magnitudes of scales (see

Figure 2.1). In addition, its calculated auto correlation function (ACF) $r(k)$ was decaying very slowly, with a shape following the model

$$r(k) \sim ak^{-\beta}, k \to \infty \tag{2.2}$$

where $0 < \beta < 1$ and a is a positive constant, which is in contrast to the Poisson ACF $\sim e^{-\beta k}$ for large lags *k*. Actually, this phenomenon is called *long range dependence* (LRD), and a self-similar process is LRD as soon as it has any positive correlations, because these will remain the same through all its time scales [RMV96]. The aggregation process to create the scaled versions can be expressed as

$$X^{(m)}(n) = \frac{1}{m} \sum_{i=(n-1)m+1}^{nm} X_i \quad , n = 1, 2, 3, ..., \frac{N}{m} \tag{2.3}$$

where m denotes the aggregation scale, i.e. the averaging is performed over non-overlapping blocks of size m. The corresponding ACF can be denoted $r^{(m)}$.

Mathematically, a discrete-time covariance-stationary zero-mean stochastic process[7] $X = (X_t, t = 0, 1, 2, \ldots)$ is *exactly* self-similar with scaling parameter $H \in (0.5,1)$ and $H = 1 - \beta/2$, if, for *all* levels of aggregation $m \geq 1$ the $X^{(m)}$ series will have the same ACF structure as X, i.e. $r^{(m)}(k) = r(k)$. An example of exactly self-similar process is fractional Gaussian noise (FGN) with Hurst parameter $H \in (0.5,1)$.

A time series is said to be *asymptotically* self-similar if $r^{(m)}(k)$ agrees asymptotically with $r(k)$ for large m and large lags k. A fractional ARIMA, i.e. F-ARIMA(*p*,*d*,*q*), process with $0 < d < 0.5$ is an example of such a process.

The measurements showed a very slowly decaying ACF in accordance with (2.2), indicating LRD. Different parameter estimation techniques have shown, in [LWTW93, BSTW95] and other work, that the Hurst parameter normally lies in the region 0.6 to 0.85. Larger values means stronger LRD. The reasons for both self-similarities and LRD are described as due to the fact that there are several layers of communication activity which causes traffic variations at different time scales. The marginal distribution of flow sizes (i.e. the number of bytes transmitted per flow session) and activity vs. idleness periods is shown to be *heavy-tailed*, which means that it could reach very large values. Heavy-tailed distributions can e.g. be approximated using the Pareto distribution with shape parameter $1 < \alpha < 2$, which shows that it is not necessary to have a hierarchy of activities in order to generate a self-similar process. If many sources send small bursts of data with heavy-tailed idle periods in between, the resulting aggregated traffic will show self-similarity.

7. a stochastic process with zero mean, finite variance $\sigma^2 = E[(X_t - \mu)^2]$ and an ACF $r(k)$ that is only dependant on lag *k*.

Figure 2.1 shows that at a range of five magnitudes, averaging the number of packets does not converge towards smooth traffic, as it would have done if it was Poisson generated. However, the measured burstiness is starting to decay at time unit 100 s, indicating that the real time series is self-similar over a large but *limited* range of scales, which is in contrast to the *unlimited* range of scales for the mathematical models of FGN and F-ARIMA. The importance of the observations is however that traffic bursts can arise over both short and long periods. Long traffic bursts makes buffer dimensioning in routers and switches very difficult.

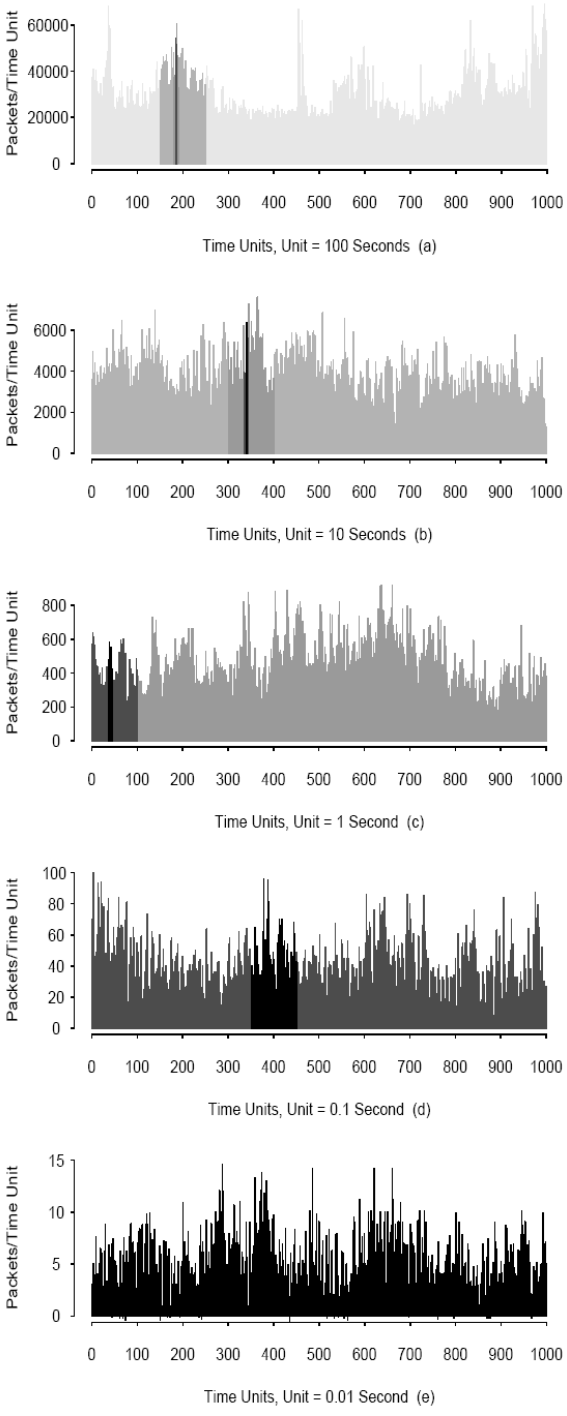

Figure 2.1: The visual proof of self-similarities, copied from [LWTW93].

What was left of Poisson modelling after this discovery was that it could still be argued that *session and flow arrivals* could be modelled using the old model. However, since the flow sizes were computer specific, and not human specific, the self-similarity characteristics made the clear distinction between a computer network and a telephone network carrying human voice. Some researchers claimed that the teletraffic theory was used blindly in computer network dimensioning and research, undermining the need for updated traffic measurements and modeling [PF95, WP98]. It has also been discovered that for small time scales, typically below 100 ms, irregularities are found that matches *multifractal* scaling behavior, that can be efficiently analyzed using wavelet-based techniques [Kan99, RCRB99].

However, in the past few years there have been some publications advocating “the comeback” of Poisson modeling. In 2004 [KMFB04], Karagiannis et al. have compared the

measurements from late 1980's and mid 1990's to new measurements on fiber-optical Internet core links, more exactly OC-48 (2.488 Gbit/s capacity). The findings differ from the over 15 year old LAN measurements in that the current network traffic can be well represented by the Poisson model for sub-second time scales. The authors claim that when more and more flows are multiplexed into higher and higher core network capacity links, future traffic will follow the same trend. This is actually good news, since Poisson modeling makes both network modeling and network dimensioning much easier, as pointed out earlier. Above sub-second time units the authors claim that the observed traffic takes on a distinctive form of nonstationary behavior. This nonstationarity consists of short intervals of fixed Poisson intensity λ_i. The duration of these intervals *i* was found also to be exponentially distributed. Further the authors make a note on that their conclusions are in agreement with large-scale aggregation of renewal processes that, according to Palm-Khintchines theorem, will approximate to Poisson behavior when the number of multiplexed sources are high enough [Cin72].

Nevertheless, the LRD characteristics of LAN networks call for a decoupling of elastic and real-time traffic. Future networks will get higher and higher capacities, making the claims of [KMFB04] more and more valid.

2.2 Media Source Characterization

Decoupled from TCP traffic, self-similar or not, the media traffic aggregate must also have known characteristics. This subchapter looks more into these details.

2.2.1 Basic characteristics

Audiovisual content differ from elastic content in their intrinsic bit rate that must be served by the network in order for continuously play-back rendering. In addition, media content has a distinct duration, over which the entire content should be transmitted, received and consumed. However, end user interruption can alter the session durations, such as through the use of interaction interfaces (*pause*, *stop*, *fast forward*, etc. of VoD content, and *stop* or *pause and store* live content), but the intrinsic bit rate must still be served. In the following some typical bit rate requirement for different audiovisual content will be given. Chapter 3 will discuss the possibilities for streaming media to diverge from its intrinsic bit rate through *rate adaptation*.

Stereo audio with near-CD quality require, with current state-of-the-art codecs, 48–128kbit/s average bit rate (AAC-HE@48kbit/s, WMA9 and Ogg Vorbis@64kbit/s, AAC@80kbit/s, MP3@128kbit/s [Sto03]). Actually, 5.1 multichannel audio can also be compressed with high quality at 128 kbit/s total bit rate, and also 48 kbit/s is possible with some reduced quality [HFD+04]. Voice over IP (VoIP) applications typically requires 16–

64 kbit/s; these codecs generally need very short algorithmic delay: AAC-LD (full 20 kHz audio bandwidth) gives 20 ms delay, while [KSW+04] has shown that 2 ms delay is possible also for audio compression at comparable bit rates at lower audio bandwidths, typically 4–7 kHz.

So what about the bandwidth requirements for the video part? In selecting a balanced perceptual quality, possible combinations could be (some of the numbers collected from [Vat05, Vat06]):

- For near-CD audio quality in companion with SDTV or HDTV video at 25 fps:
  - SDTV encoded with MPEG-4 ASP will have high quality at an average of ~1.5 Mbit/s
  - SDTV encoded with H.264 will have high quality at an average of ~1.0 Mbit/s
  - HDTV encoded with MPEG-4 ASP will have high quality at an average of 14 Mbit/s
  - HDTV encoded with H.264 will have high quality at an average of 7–8 Mbit/s [App05]
- VoIP (speech) audio quality in companion with QCIF or CIF video at 25 fps or lower. Typical application is low-end videoconferencing.
  - QCIF@10fps at 80 kbit/s with MPEG-4 Simple Profile has decent quality
  - CIF@25fps at 300 kbit/s with MPEG-4 Simple Profile has good quality

Since the video streams generally requires significant more bandwidth than audio, the focus of this thesis is on the video part. Note however that audio-only applications such as “clean” VoIP and Internet Radio will exist, while video-only applications hardly will exist. The emerge of 3D video [ATS07, SK05, YCA+06, YYNB06] will make the video part of occupied bandwidth become even greater. 3D audio will demand higher bit rates than 2D audio, although there has been tremendous progress in coding efficiency the last years [BSP01, HFD+04]. It can be argued that the time critical part of a video conference or musical collaboration is the audio, while the video part can tolerate some more delay. In this thesis it is assumed that the video should have the same delay requirements as audio, and thus creating perfect “lip synchronization” as an inherent property.

2.2.2 CBR and VBR open-loop

During the late 1980’s and throughout the 1990’s there was a significant research into how to model VBR video streaming [GW94, HL96, And97, KM98, LOR98]. Most of this research was motivated from the need for ATM services to specify parameters for admission control of the VBR-rt and VBR-nrt application classes. The applications belonging

to these classes should not apply any rate adaptation (as contrasted by the ABR class), and new flows should be admitted only if they could be granted QoS guarantees without reducing the QoS of already existing flows. The source description is based on a leaky-bucket parameters consisting of leak rate r and bucket size b, defining the leaky bucket parameter set LB(r, b). The leak rate decides the average rate, while the bucket size decides the maximum burst size, i.e. the larger the value of b, the more variability is allowed for the source. As an example, an ideal CBR source of rate r can be characterized as LB(r, 0), while a VBR source where the largest burst size can equal the average rate spent over 2 seconds is given as LB(r, $2r$). A peak-rate maximum p might also be specified in addition. An open-loop VBR source, i.e. a compression algorithm working with constant quantization parameters, has no control of the average bit rate at encoding time (and hence neither the peak rate), because the bit rate variation is entirely given by the entropy of the content to be compressed. Thus, as an example, a video codec compressing a conference scene with little or no motion and a steady background, will produce significant lower average bit rate compared to when compressing a feature film of rich and variable high-frequency content and a lot of motion (assuming the same codec parameter set). Pre-stored open-loop encoded media files can however be analyzed and a LB(r, b) parameter set can be calculated so that r corresponds to the average bit rate of the total media file, and b corresponds to the maximum peak size, supporting AC control of new flow session arrivals. However, such a policy has its price of low link utilization if a significant number of the connections are open-loop VBR encoded with significant burst peaks, especially when those maximum peaks are rare events inside the content.

Since an audio and video encoder can be made producing CBR output, why not choose CBR mode always? From a network point of view, this is the easiest sources to deal with. But, as stated before, CBR offers constant bit rate on the cost of variable quality. Further, CBR mode was originally developed to target the old fixed capacity telecom lines. Packet networks offer flexibility, and multiplexing large numbers of non-synchronized VBR sources open the possibility for statistical multiplexing gain (SMG), which is lost if using CBR. On the positive side of CBR, as long as the sources are not synchronized, the aggregate CBR "burst periods" are marginal and thus producing limited excess bit rates compared to the average rate.[8] Thus, peak rates are small compared to the total link capacities, and the sources can be efficiently multiplexed with high link utilizations without using too large multiplex buffers. VBR open-loop video streaming however, especially of feature film, has characteristics that make things much worse: it produces burst periods at different time scales [GW94, BSTW95]. Actually, at GOP level, and thus also at frame and packet level, it produces LRD within each VBR flow! Increased buffer sizes to cope with LRD traffic may be a complete waste since worst case traffic burst peaks are unpredictable. The only way to provide QoS guarantees for aggregate traffic dominated by VBR open loop, is to have a priori knowledge of average and peak rates, thus ruling out

8. Following [Cin72], many CBR flows will together approach Poisson.

live content. Since VBR open-loop can have very long periods of high bit rates ("long" peaks), its average rate is colored significantly by these periods. Thus, to account for the probability of simultaneous occurrence of such periods in two or more flows transmitting packets through a common link, the utilization has to be kept low, especially if providing *deterministic QoS guarantees*. To enhance the link utilization, one can instead offer

1. *Statistical* QoS guarantees, or
2. using *constrained VBR* instead of open-loop VBR.

The former subject will be discussed in Chapter 4.1, while the latter will be elaborated in the next subchapter.

2.2.3 Constrained VBR and LRD suppression

In CBR encoding, the rate controller's goal is to produce a fixed number of information bits per time unit, giving a target bit rate r. This time unit is relatively small, typically 20 or 40ms for audio codecs, and one GOP interval for video codecs. A GOP interval for networked media should typically be in the range of 0.5–1.0s, while media stored on reliable storage media such as compact discs can have significantly longer GOPs. The reason for short GOPs in networked media is to limit the damage period in case of packet losses. The CBR transmitter outputs the data at a constant rate, which necessarily will delay some of the frames. The receiver playout time must take this into account in order to avoid frame starvation.

Most modern media codecs offer also VBR rate control. Basically, the VBR rate controller ensures a stable average rate over an expanded time unit, but the frames are generally submitted directly into the network without any delay. Thus, the instantaneous bit rate can vary significantly, but the bit rate averaged over the expanded time unit will be close to the target average rate r. A common method of producing such rate control is by the use of the leaky bucket algorithm.

ISO MPEG has defined a video buffer verifier (VBV) based CBR rate controller, taking into account both sender and receiver buffer, and eventually transmission medium delay, based on e.g. work by Reibman et al. [RH92]. Its variants are nicely explained in Sun and Reibman [SR01] chapter 9. The VBV algorithm controls the rate so that both receiver buffer overflow and underflow are avoided, given a constant transmission rate. A much simpler rate controller for constrained VBR was proposed by Hamdi et al. [HRR97], where the leaky bucket at sender only is used as a virtual buffer, in that the compressed media is not delayed since it does not pass through it, as shown in Figure 2.2. As shown in Paper E of Part II, a rate adaptive LB$\{r(k),b(k)\}$ simulation tool was made based on Hamdi's rate controller. In this subchapter, however, some interesting and important properties will be presented by some video coding examples.

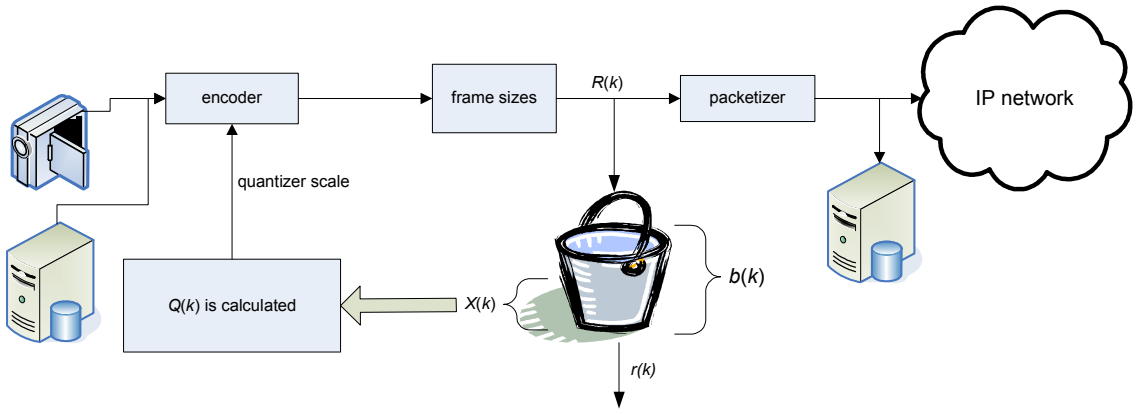

Figure 2.2: A VBR rate controlled encoder of the leaky bucket type LB(r,b), working on either live or pre-stored media, and outputting packetized data either to IP network or media storage server.

In Lemma 1 on page 13 it is stated that if Proposition 1 on the same page should be attainable, the aggregated VBR flows should not exhibit any significant LRD characteristics. Paper E examines this issue in detail. Still, the rest of this subchapter will enlighten this topic even further by some more examples, both to support Paper E into more detail, and to show that this property is not a local phenomenon of the Hamdi VBR rate controller.

For the following examples a concatenated video sequence of the official ISO MPEG test sequences News, Football, Akiyo, Stefan, and Paris in 30fps CIF resolution was created (see Figure 2.3, i.e. the same sequence used in Paper E and F), resulting in a sequence length of about one minute. The ffmpeg MPEG-4 codec was used to VBR open-loop compress this sequence using a static quantizer scale Q=2 (best possible), GOP size of 12 frames, and allowing only I-frames and P-frames. The resulting compressed video stream got the frame sizes shown in Figure 2.4.[9] The I-frames are clearly visible as peaks, also indicating the start of each GOP period. Also noticeable is that the sport scenes (Football and Stefan) have the highest bit rate. Akiyo, which is a very static scene with the female news reporter in front of a static background, has the lowest bit rate. Since the quantizer scale is set static at the highest level possible, the quality is constant and very high. Clearly, this compressed video sequence will produce a very variable traffic load, both short term (I-frame peaks), but also long term (the scene changes). Actually, it is the scene changes that produce the LRD effect of VBR open loop video [GW94]. Obviously, if the content is live and continuous, it is impossible to know a priori what the average rate of a session will be, because it will depend on the entropy of the media to be compressed.

The same video sequence was now compressed and supervised by the Evalvid-RA VBR tool-set (Paper E contribution), resulting in the frame sizes shown in Figure 2.5. Clearly, the traffic has been given a controlled average rate, and the scene change effects are

9. The compressed content was analyzed with the mp4.exe tool from EvalVid 1.2 toolset [KRW03]

almost completely eliminated. The latter is an important property, since it will remove LRD. This is confirmed by calculating the autocorrelation of the same sequence at GOP size scale, resulting in the graph of Figure 2.6.

Paper E also uses longer video clips (approximately 7 minutes) from the documentary "The Inconvenient Truth" (CIF @ 25 fps, I and P frames) and the feature film "The Matrix" (CIF @ 29.97 fps, I, P, and B frames). As to provide further evidence of the generality of the claims made above, the ffmpeg MPEG-4 encoder itself is tested here both in VBR open loop and in *ffmpeg constrained mode* to encode these longer video clips. The

Figure 2.3: First frame of "News" (top left), "Football" (top right), "Stefan" (bottom left), "Paris" (bottom right). "Akiyo" is a sequence with the female reporter in "News".

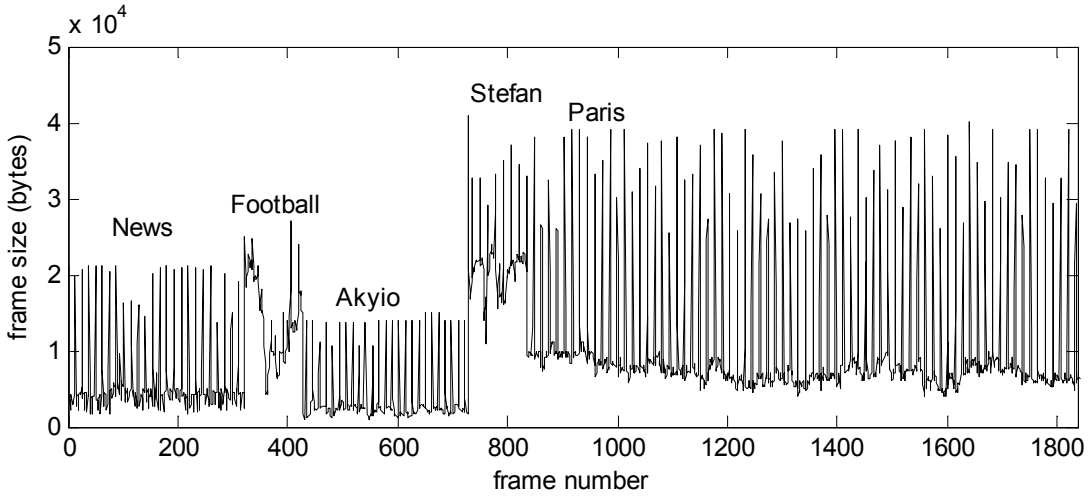

Figure 2.4: Frame size variations of the concatenated video sequence.

resulting autocorrelation plots are given in Figure 2.7. Both examples use GOP size of 12 as before, but note that The Matrix sample now is compressed also with B-frames. The ffmpeg parameters used in these examples were (for repeatability):

- Figure 2.7 a)
  - Open loop:

```
ffmpeg -s cif -r 25 -i Inconv1.yuv -vcodec mpeg4 -4mv -s cif -g 12
-sgop -sc_threshold 20000 -r 25 -qscale 4 -y test_aO.m4v
```

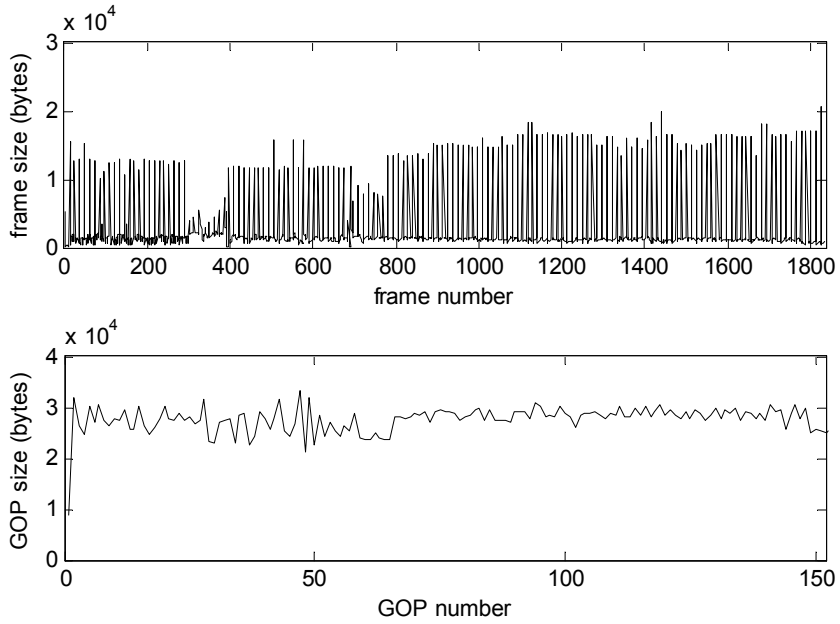

Figure 2.5: Frame size (top) and GOP size (bottom) of the test sequence when applying Evalvid-RA's RA-VBR rate control with 600kbit/s fixed average rate and b=360kbit. The GOP period is 400ms.

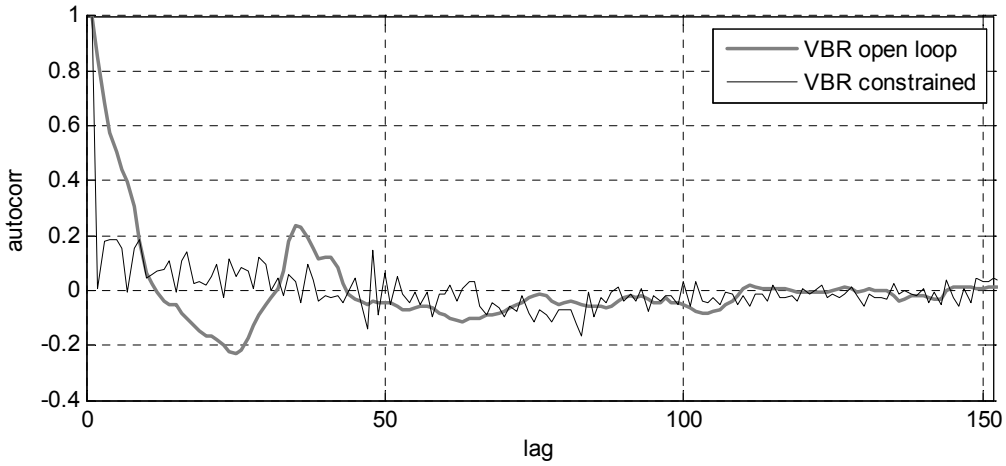

Figure 2.6: The autocorrelation at GOP size scale of the same video sequence as in Figure 2.5 shows a slow decay of the VBR open loop, and a much faster decay of VBR constrained.

- Constrained:

```
ffmpeg -s cif -r 25 -i Inconv1.yuv -vcodec mpeg4 -4mv -s cif -g 12
-sgop -sc_threshold 20000 -r 25 -b 400 -maxrate 490 -bufsize 20 -y
test_aC.m4v
```

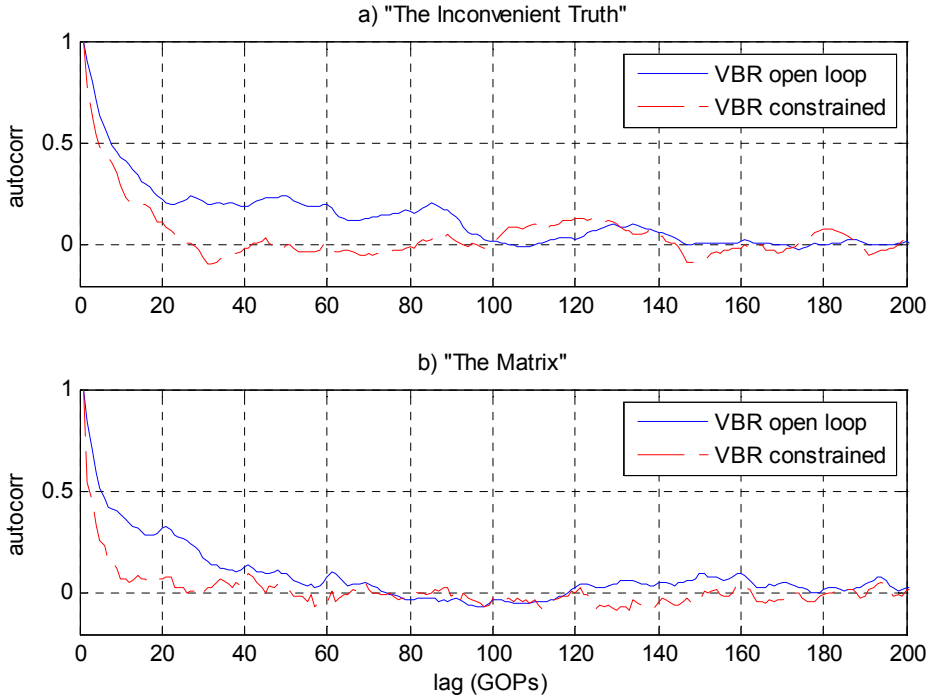

Figure 2.7: The autocorrelation of GOP sizes of clips from a) The Inconvenient Truth and b) The Matrix, encoded with fixed quantizer scale 4 (open loop) and target bit rate 400 kbit/s (constrained), respectively, using ffmpeg.

- Figure 2.7 b)
  - Open loop:
    ```
    ffmpeg -s cif -r 29.97 -i matrix_cut1_sm.yuv -vcodec mpeg4 -4mv -s
    cif -g 12 -sgop -sc_threshold 20000 -r 29.97 -bf 2 -qscale 4 -y
    test_bO.m4v
    ```
  - Constrained:
    ```
    ffmpeg -s cif -r 29.97 -i matrix_cut1_sm.yuv -vcodec mpeg4 -4mv -s
    cif -g 12 -sgop -sc_threshold 20000 -r 29.97 -bf 2 -b 400 -maxrate
    490 -bufsize 20 -y test_bC.m4v
    ```

We see that in both examples, the VBR constrained version has an ACF with much faster decay towards zero than the open loop variant. The decay is however not so fast as for the concatenated MPEG video sequence: the reason for this is that these two feature films have very variable action, resulting in periods where the bit rate at Q=2 is in fact lower than the target bit rate. Still, keep in mind that these results show single flow ACF, so the aggregate of many of these will lean more against Poisson behavior [Cin72]. P-AQM handles also these sequences well, as the PSNR plots of Paper E subchapter 6.5 show. Thus it is verified that this property is present in ffmpeg compressed media also, and is thus not a unique quality of the Evalvid-RA VBR rate controller alone. Paper E and F will show that this property is present also for *adaptive rate control* so that the aggregate of VBR and elastic flows can be dynamically controlled to ensure high link utilization and low delay and packet loss for the VBR flows.

3

Audiovisual rate adaptation possibilities

It is not the biggest nor the fastest of the species that survive,
but the one that adapts to its environment

It is not the strongest of the species that survive, nor the most intelligent,
but the one most responsive to change

Charles Darwin (Theory of Evolution, 1809–1882)

3.1 New need for rate adaptation

As already discussed, real-time media traffic has end-to-end constraints. Different QoS models have also been discussed, and it becomes clear that all methods will eventually fail if the traffic load is not adjusted to match the available bandwidth resources. This happens if the arrival rate of new sessions is larger than the departure rate over a sufficient time window. Traditionally, media applications have not been regarded as "good network citizens" (perhaps with the exception of low-rate self-limiting applications like VoIP), in that its data traffic per flow has often exceeded the fair rate compared to TCP flows with a significant margin. This has been the main motivator for network administrators to deploy firewalls that block unknown UDP flows from entering LAN segments.

The question then is by which means the media should scale? The goal is similar to the goal of an off-line media encoder given a specific rate constraint: the media should be compressed to best possible quality given a new average bit rate constraint. Inevitable, when congestion calls for lowering the average bit rate, this action results also in lowering the end user perceived QoS. However, the avoidance of doing so will produce even lower QoS because of significant packet drops due to *congestion collapse*. Ironically, up to now, *the avoidance* of performing media rate adaptation has actually given *the end user* the best QoS. This is because it has been the elastic applications, and perhaps a small number of media applications, that have paid the "congestion bill" alone. Firewalls and new router

design are expected to stop or penalize ill-behaving media flows, to force an incentive for adaptation deployment.

The means by which the content is scaled, and the effect it has on the perceived quality, is however a challenging area of research. The reason is that it depends on both the type of content and end user preferences. It turns out that media can be scaled in several different ways. The next subchapter will look into the different methods. It is however outside the scope of this thesis to elaborate on which methods should be used when, or which methods should be simultaneously combined, in order to optimize the perceived quality given a certain set of constraints (e.g. [Joh08] looks into these issues). The set of methods will be explained, and arguments given why this thesis research have focused on one particular method.

3.2 Media content compression and its quality measures

Media compression techniques generally include three main steps:

1. Transform coding: temporal content is transformed to the frequency domain by the use of e.g. DCT (e.g. MPEG) or Wavelet (e.g. JPEG2000). The transformed signal is better suited for the next step, which is
2. Quantization: the transformed signal values are quantized down to a fixed number of bits per value. This can be regarded as the removal of *irrelevant* information. Typically, many of the high-frequency component values are rounded to zero.
3. Entropy coding: the set of quantized values are organized (e.g. zig-zag sorted in video 8x8 pixel DCT transform) and entropy encoded (e.g. a mix of run length and Huffman coding), i.e. the *redundancy* is removed.

In the decoder, the inverse operations are performed. A video codec typically also includes compression along the temporal axis. This is accomplished by calculating the difference between two sequential video frames, named P_i, but where the oldest frame is motion compensated to match the current frame in best possible way (predictive motion compensation coding). The “error” signal P_i is compressed as a normal frame. The combination of transform coding and predictive coding is named *hybrid transform coding*, which forms the basis of all MPEG codecs (MPEG-1, MPEG-2, MPEG-4, AVC). Hybrid coding is using P- and B-frames in addition to I-frames. P-frames are forward predicted, while B-frames are bidirectional predicted (both from previous frame, but also from future frame, which causes algorithmic delay, and is therefore rare in live interactive communications).

It is the quantization step that introduces quality loss.[10] Thus, the decoder is unable to reconstruct the digital content back to its original bit-exact form. To measure the recon-

10. loss-less coding, which yield much less compression efficiency, is not considered here.

structed quality relative to its original quality, different measures exist. Since the erroneous audiovisual content is to be consumed by humans, it is the human perceptual quality that is of importance. This can be measured by human observers comparing original and compressed content "side-by-side". This is a common way of improving coding efficiency in new compression algorithms, and is thus named "subjective testing". Perceived quality is e.g. given as mean opinion score (MOS) defined by ITU, ranging from 5 (excellent) to 1 (bad). Subjective testing is however time and resource demanding, and "objective testing" is often chosen as a good alternative. Objective tests can be divided into two categories: *pixel-based* (e.g. PSNR) and *psycho-visual* metrics. PSNR (peak signal-to-noise ratio) values are calculated in video systems by comparing the luminance original frame $Y_O(n)$ to the compressed/decompressed frame $Y_D(n)$ calculating

$$PSNR(n)_{\mathrm{dB}} = 20\log_{10}\left(\frac{V_{\mathrm{peak}}}{\sqrt{\frac{1}{N_{col}N_{rows}}\sum_{i=0}^{N_{col}}\sum_{j=0}^{N_{rows}}[Y_O(n,i,j)-Y_C(n,i,j)]^2}}\right) \quad (3.1)$$

where $V_{\mathrm{peak}} = 2^k - 1$, k being the number of bits per pixel in the luminance component, i.e. 8 bits give $V_{\mathrm{peak}} = 255$. If the quality is high, i.e. the frames are very similar, the PSNR values are high. As a rule of thumb, values above 30dB are regarded as good, and above 40dB as very good. However, absolute borders between bad/good/very good/excellent can not be made because the values vary by content. The psycho-visual metrics, which are based on the human visual system, don't have these limitations, and therefore outperform PSNR metrics in most cases [WP02, Win05]. Sadly, this comes on the cost of considerably complexity increase. An example of such software is VQM [ITS]. Due to the latter reason, PSNR metrics are still used, especially when not absolute metrics but rather relative metrics are the primary interesting quality measure. SSIM is a good alternative to PSNR since it improves the confidence of the quality metric at a small complexity cost. (However, this thesis uses PSNR due to the usage of the available tools of Evalvid, see Paper E for details.) Since it is the network and rate control framework that is in focus here, and not the compression algorithms themselves, relative metrics can be regarded as adequate.

3.3 Media coding and scaling technologies

The resulting compressed media bit rate is decided by more parameters than those dealing with the core compression algorithm presented in the previous subchapter. Actually, a media encoder can scale the entropy along three different main axes:

1. Temporal axis, i.e. vary the number of frames per second (video) or sampling rate (audio). E.g. dropping each other frame can reduce the bit rate by almost 50%.[11]

2. Spatial axis, i.e. the frame size or frame resolution (video) or sampling accuracy (audio). E.g. dropping from CIF (352x288 pixels) to QCIF (176x144) produces a potential of 75% bit rate drop.
3. Quantization “axis”, i.e. how much detailed information is left after performing the quantization part of the compression. This is often also called “SNR scalability”, and can be adjusted to very fine steps.

The reminder of this subchapter will now focus on some video coding issues and details, although some of them also applies to audio coding.

Step 3 is nothing else than the quantization part of the transform coder. E.g. in MPEG-4 video coding, a *quantizer scale* parameter Q is defined, having values 1–31 [ISO99]. For each value, a quantizer matrix is defined, identifying the actual quantizer value for each of the 64 DCT coefficients in a 8x8 DCT block. Varying this quantizer scale per frame or even per macro block (a 2x2 DCT block, i.e. 128x128 pixels, used as the motion prediction calculation block), opens for very fine rate adjustments.

A video codec rate controller will therefore normally only work along the quantization axis, targeting one fixed average bit rate (CBR or VBR). A quantization scale of 2 gives twice the bit rate compared to $Q = 4$, which gives twice the bit rate compared to $Q = 8$, and so on. The range 1–31 gives approximately a compression range dynamic of 1:16 (lowest bit rate is about 6.5% of the maximum). Some encoders also include the possibility of *frame discard*, which e.g. is used in situations where using the most coarse quantization simply is not sufficient in obtaining the bit rate constraint. This implies that frame discard is most helpful in CBR coding. VBR rate control on the other hand can tolerate such variations. To give an example: in Figure 2.4, at about frame number 800, the content has a high traffic peak. A CBR rate controller might turn to frame discard in order to have the rate constraint fulfilled in that period.

When taking the step into rate *adaptive* control, a couple of important aspects must first be illuminated:

- Rate controllers having access to network and display terminal feedback at encoding time is name “live adaptive encoders”
- Off-line encoders might encode with special tools that open the possibility of streaming media at different rates, based on the very same content file. This is named “scalable content”.

Not surprisingly, being online or offline at encoding time gives different approaches and answers solving the rate adaptation challenge. A common challenge is how to achieve a

11. In I-frame only coding, the bit rate will be reduced by 50%. However, for hybrid coding, such as MPEG, the reduction will be less because the remaining P- and B-frames will increase in size to compensate for the missing temporal information.

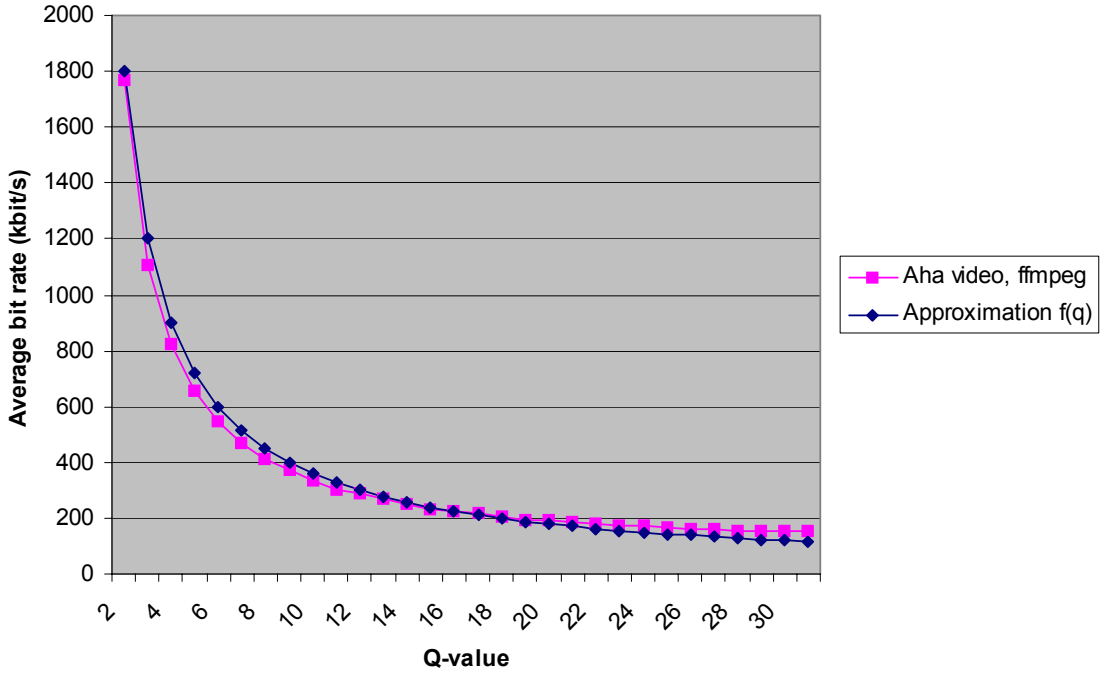

Figure 3.1: The quantization scale parameter Q fixed to values 2–31 to give 30 different qualities of a CIF@25fps Aha music video. As the comparison shows, the rate curve is proportional to 1/Q.

coding efficiency as close to non-scalable state-of-art efficiency as possible. Actually, live adaptive encoders achieve this most easily, because the only modification necessary is that their target bit rate must be a dynamic parameter, controlled and modified by the network feedback information. Therefore, the most used means in online rate adaptation is SNR scalability, eventually combined with frame discard. However, one limitation of live encoders is that multipass encoding is prohibited. Therefore, live encoding, being rate adaptive or not, can never achieve the same maximum efficiency as offline multipass encoders.[12]

Offline encoding of pre-stored media is quite another story. It has proved to be difficult to create *scalable* content without loosing some coding efficiency. One simple solution is *multirate coding*, which is actually nothing else than encoding the same content multiple times, and eventually putting all variants into one single file. Each rate variant can therefore have maximum efficiency. However, these achievements come at an obvious added storage requirement cost. Another solution is the ISO MPEG-2 & MPEG-4 technologies where one *base layer* is combined with one or more *enhancement layers*. This solution has much better storage efficiency, but this comes on the cost of reduced coding effi-

12. Multipass encoding can be explained as a coding process where the temporal redundancy is explored to its full potential, because the first pass gives the rate controller of pass two a priori information of the whole sequence, so that the quantization parameter can be fine tuned per macro block level in order to fulfill the rate constraint and produce optimized quality.

ciency. The base layer has typically low frame rate and frame size. The enhancement layers may add higher frame rate and resolution, as well as more bits per pixel (SNR scalability): the more enhancement layers that can be sent and received the better the quality, on the cost of higher bit rate. A major drawback with both multirate and enhancement layer coding is the somewhat coarse rate adaptation: the rate adaptation engine has to choose among typical only 3–4 different qualities (as a comparison, imagine that a TCP session can only work at 4 different bit rates). In addition, enhancement layers must be fully received before they can be decoded, i.e. per frame basis. This was the motivation behind MPEG-4 FGS (Fine Granularity Scalability). The MPEG-4 FGS supports temporal as well as SNR scalability. The bit stream is divided into one base layer and one FGS enhancement layer. In FGS, the rate adaptation can be made at very fine levels, and its character can be compared to the download of a progressive JPEG image: the more information received the better is the image quality. I.e., the more information received from the FGS layer, the higher is the resulting video quality. An FGS compliant streaming server can therefore at a fine granularity stream only the part of the FGS layer that is accepted by the current network conditions. Also, the FGS storage efficiency is very high. Not surprisingly, however, the FGS gain of scalability and storage efficiency come at the cost of significant reduced coding efficiency and high implementation complexity.

Finally, *transcoding* is a way of adapting pre-stored media to current bandwidth constraints. Transcoding is in its simplest form decoding of a media file, followed by a new encoding phase, to target a different bit rate (normally lower). In addition, transcoding supports the transfer to another coding and/or file format. Live transcoding requires a lot of CPU resources, and quality loss will take place both in its first (original) and second (transcoding) encoding phase.

In sum, as coding *efficiency* has been the most important market parameter, the only commercially successful scalable solution up to now has been multirate coding. Examples are Real SureStream and Microsoft Intelligent Streaming. Envivio delivers MPEG-4 compliant multirate solution. Also, most research focus has been on creating scalable *pre-stored* media formats, since this is the most challenging task, balancing coding and storage efficiency with high flexibility, at a lowest possible complexity. The brand new H.264 SVC might succeed in market acceptance. Still, rate adaption is perhaps even more important for live and interactive media, due to the strong latency constraint. Chapter 4 will explore how the network can cooperate with the media rate controllers to obtain just that.

3.4 Emerging new technologies, future speculation

As a last subchapter before diving into the adaptive rate control, some new and emerging technologies and framework should be listed.

Normally, when requesting a video stream, an end user anticipates to receive that video stream. But what happens if the Internet get too congested during that session? In peak hours, substantial packet loss might occur, resulting in corrupted video frames and perhaps also audio drop-outs, and the end user's expectations are far from being fulfilled. This can also happen when using adaptive techniques, because the available bandwidth is below a certain base layer or highest quantization scale. In such situations, the user would have been more satisfied with a "slide show", i.e. a good still image once per 2 seconds or so, with the audio quality preserved. If it is a news summary, *text* would be much better than corrupted video and audio, because the end-users primary focus is the news *information*, and not its *form*. This adds a new adaptation axis to the three already listed: *modality conversion* [AD05].

An ISO MPEG initiative that both take rate adaption, with possible modality change as well as many other aspects of media consumption into account, such as intellectual property rights, billing, security, and easy retrieval of media meta data, is MPEG-21. Supported by MPEG-7, MPEG-21 defines a new XML language, DIDL (Digital Item Declaration Language), which can be used to define complete "packages" of media content, with its adaptation variants, delivery constraints, rights, and more. The DIA (Digital Item Adaptation) defines a set of XML Schemas supporting a flexible adaptation engine [VT05]. Still, MPEG-21 is only a framework, and the tools performing the different functionality such as rate adaptation, must be found elsewhere.

Current state-of-the-art standardized codec is named H.264 (MPEG-4 AVC). A lot of research in scalable coding techniques still goes on within ISO MPEG and ITU-T. SVC (Scalable Video Coding) is a new scalable profile of H.264, just finalized as a standard. It is expected that the results will show much improved efficiency performance compared to earlier solutions. Actually, Iqbal et al. have proposed a rate adaptation scheme for H.264 video based on the MPEG-21 DIA framework [ISS07].

Lastly, object based encoding should not be forgotten. The MPEG-4 standard has had this feature among its main assets since 1999. Since object segmentation is still regarded as a complex task (except when using TV-studio "blue wall"), the motivation of using encoding with arbitrary shaped objects has been limited. This can however change in the future. In streaming feature film, film cues could guide rate adaptation engines so that e.g. objects not so important for the story telling could be compressed harder than important objects [Lie02].

4

Controlling streaming media

We cannot control the evil tongues of others;
but a good life enables us to disregard them.

Cato the Elder — Roman orator & politician (234 BC - 149 BC)

This Chapter will outline the background and motivation for the chosen network architecture. Some historical architecture schemes are listed initially, leading up to the chosen strategy of this thesis. The main goal is to establish and present the different network traffic research areas and how they relate to each other, with pros and cons. Both QoS and flat best effort networks are covered. Finally, media rate control requirements are defined, and existing proposals are evaluated.

4.1 Statistical QoS guarantees

Deterministic QoS guarantees is the name of a service model promising that e.g. packet latency and packet loss bounds are *never* exceeded. Generally, to maintain such QoS requirements, the link utilization has to be kept low by a reliable Admission Control system. The sources traffic is policed, and packet drops are executed as soon as a source tries to transmit more packets (or ATM cells) than agreed by its traffic model. The number of allowed simultaneously sources is dependent on the requested persistent rate and rate variability. Network architectures providing such policing are e.g. ATM VBR-rt and Internet IntServ, and to a certain extent DiffServ with Assured Forwarding class. Knightly et al. reports that with advanced techniques the utilization of a VBR open loop aggregate with 50 ms network delay is in the range 18–37%, depending on the type of media content [KWLZ95].

Statistical QoS guarantees, on the other hand, is a service that promises that the QoS bounds will be maintained in some p fraction of the time. Alternatively, that the *probability* a certain packet drop ratio or network delay will be higher than a certain bound is a small but non-zero value. Statistical guarantees can generally be maintained at a higher link utilization than deterministic.

There are generally two main options for statistical multiplexing network design according to [RMV96]:

1. Rate Envelope Multiplexing (REM)

2. Rate Sharing (RS)

In REM, all flows are assumed with a well defined continuous rate $\lambda_i(t)$, which varies slowly except at the start and end of such sessions. REM ensures that for a set S of flows sharing some capacity C, the probability of aggregate traffic exceeding the capacity should be less than some small fraction ε, i.e.

$$Pr\left[\sum_{i \in S} \lambda_i(t) > C\right] < \varepsilon . \tag{4.1}$$

REM is assumed in a hypothetical bufferless system, such that queuing delay is zero and loss is limited by applying Admission Control (AC). The performance of the REM model is discussed thoroughly in [RMV96], and the application of video transport is outlined in [RH98]. The latter paper also argues how variable VBR open loop sources may utilize REM designed networks.

In practice, a REM designed network must use *some* buffering to cope with multiple interfaces routing packets simultaneously to the same outbound link (in ATM this is called "cell scale buffering", and is in the order of 100 ATM cells according to [RH98]). Due to the short buffers, the traffic characteristics are maintained throughout the network. The short buffers take care of the delay guarantees. The packet (cell) loss rate is controlled by the AC, controlling the access blocking of new sessions. The AC is dependent on that it is possible for the sources to characterize their video traffic. It is argued in [RH98] that REM packet loss performance is only dependent on the *stationary* statistics of the sources. But it is also argued that such characterization is almost impossible for VBR open loop sources, at least for live sources. For pre-stored open loop media files, it is possible to define a leaky bucket pair $LB(r, b)$ that will ensure that a source buffer b is never exceeded, as long as r is at least higher than the media files average rate (the lower r the higher b). For network dimensioning, also a peak rate p should be defined. If assuming N identical but statistical independent sources with parameters r and p, and the sources always transmits at their peak rate when they are active, their probability of being active is $\alpha = r/p$ and the number of active sources follows the binomial distribution. [RH98] shows that the cell loss rate CLR can be expressed as

$$CLR(N, C, r, p) = \frac{\sum_{n=\lceil C/p \rceil}^{N} (np - C)\binom{N}{n}\alpha^n(1-\alpha)^{(N-n)}}{Nr} \tag{4.2}$$

because there is packet (cell) loss only when the product of peak rate and number of active sources exceeds the link capacity C. Keeping all factors fixed except N, the CLR increases with increasing N. The number of active flows admitted should then be limited to the supremum of N constrained by some upper limit of CLR, i.e. ε as given by (4.1).

REM is generally more efficient (higher link utilization) when the sources peak rate p is small compared to the links total capacity. Consequently, *VBR constrained* sources will define an aggregate allowing a higher number of flows into the network, compared to VBR open loop, given a comparable average quality and packet loss statistics constraint. Alternatively, VBR open loop encoders can have their bursty traffic smoothed by sender buffers, thus trading delay for higher utilization. REM designed networks are suitable for low delay video like videoconferencing, due to the controlled low network delay.

RS designed networks generally allows higher link utilization than REM. This is so because in RS design, the network routers are equipped with well sized buffers (thousands of ATM cells according to [RH98], which could be mapped to hundreds of IP packets, i.e. a typically Internet router of today) to cope with source traffic variance. Consequently, RS designed networks have their performance influenced also by the sources correlation: long peaks may create buffer overflow even in well designed routers. As already mentioned in Chapter 2.2, Beran et al. [BSTW95] and Garrett and Willinger [GW94] showed that open loop VBR video has long range dependency, i.e. its variability is visible at multiple time scales. Thus, it is very difficult to utilize the increased robustness provided by the well sized buffers when such sources are admitted into the network. (On the other hand, the network delay of REM designed networks are independent of any LRD characteristics.) Consequently, VBR constrained video should be used to utilize the full RS potential, because (i) it still has some SMG potential (see Paper E), (ii) the RS buffers allows bit rate dynamics even at high average load, and (iii) packet loss is limited because an average rate is defined. [RH98] argues that REM and RS link utilization performances become not very different when the sources are equipped with sender buffers in the REM case, on the cost of increased delay. Still the REM delay is marginally smaller than the RS delay.

Statistical QoS guarantees is still an ongoing research area, also in the context of wireless LANs [ZWF06] and DiffServ networks [WXBZ03]. Network calculus has become a vital tool for performance analysis [Jia06]. The details of these topics are however out of scope of this thesis. Nevertheless, the included material of REM and RS design provides a preliminary conclusion that low delay and high link utilization are conflicting requirements, of which both differentiation and source rate control both can provide solutions for. This thesis focus on the latter solution space.

4.2 Proactive vs. reactive control

The previous subchapter gave an overview of the main ideas from the network design and dimensioning research of ITU and ATM Forum of the 1990's. As already covered in Chapter 1.2, this work also influenced the QoS research of IETF (mainly the reservation based and AC controlled IntServ, but also DiffServ with prioritized classes). To repeat, most of these regimes are found to scale very badly. In addition, as Roberts & Hamdi and others were emphasizing, one of the main problems was to adequately model a source traffic specification, e.g. with mean rate, peak rate, and some burst buffer size. If many sources were too optimistic modeled, the aggregate of these would ask for far more bandwidth than admitted, and significant packet drops (either by AC policing or overflooded network buffers) would result. On the other hand, if many sources were too conservative modeled, the aggregate would by far utilize available bandwidth, resulting in poor exploitation of invested link capacities.

One remedy to the incorrect source descriptions is named Measurement Based Admission Control (MBAC) [JSD97, GT99, FOBR01]. In MBAC, there is no need for a detailed a priori source description. A peak rate can be sufficient. This is so because in MBAC, the already admitted traffic is measured. Based on this measurement, an available rate is estimated. MBAC has been an ongoing research area for a decade, and improvements to the initial ideas have been many, e.g. [JEN+05]. MBAC is outside the scope of this thesis.

In Proposition 1 on page 13, the goal of this thesis says "close to 100% link utilization". To reach such a goal, both elastic traffic and real-time traffic must include congestion control, and some buffering must be applied to hold traffic bursts. Thus, the targeted architecture is modeled more like RS than REM. However, well sized buffers to tackle elastic traffic characteristics will break the latency budget, as outlined in Chapter 1.3.

Conjecture 1: *Elastic and real-time traffic must be decoupled in the network, e.g. by differential scheduling based on classes, to ensure high link utilization, and at the same time low latency for the real-time traffic. This can be accomplished without offering more bandwidth to the low-latency traffic, if both type of traffic deploy rate control.*

ABE, as already covered, is one approach, to help the latency requirements. However, it does not discuss the question of rate control. In this thesis, a two queue scheduler will be part of the final router design. One queue is well sized as in rate sharing, and will buffer elastic traffic. The other queue is smaller, and will buffer real-time traffic. Its size is larger than in REM, but smaller than in RS.

Conjecture 2: *To avoid significant packet loss in the real-time queue, the media source rate controllers must be adaptive, and react on precise network feedback information.*

The selected regime does not rule out Admission Control. However, new flows can be allowed into the network even if the utilization is already close to 100%. This is so because the other flows adapt to lower rates to make room for the new flow. If the demand for new flows exceeds the session completion rate for some period, the adapted rates might be so low that the end users QoS requirements are not met. This happens when e.g. the video is forced to use very high quantization scale values for some significant time, or the elastic flows completion time exceeds some impatience limit. Network feedback, congestion control, and admission control can be combined to avoid such situations. A user must then be prepared to experience service blocking even when surfing open best effort services. This thesis do not incorporate a specific AC design in its solution.

4.3 Live interactive streaming media requirements

4.3.1 Delay and delay jitter

The end-to-end delay requirement is dependent on the application. As indicated in Chapter 1.1, VoD service may have several seconds, and thus network delay jitter will be absorbed effectively by a well sized receiver buffer. Since there is also time for retransmission of lost packets, such services often use TCP instead of UDP. Progressive download is an alternative for limited sized video clips, such as movie trailers. Feature films should not use progressive download, if not opting for storing several gigabytes for later offline viewing.

WebTV should have a limited delay at session start-up, to facilitate fast channel zapping. Assuming WebTV is a multicasting service, the receiver buffer playout time can be adjusted after startup by slowing down the rendering time of video and audio (audio should be pitch adjusted) during some time period, until the receiver buffer has grown to a “jitter secure size”. In this way, the jitter vulnerability at initial channel setup can be slowly reduced towards a jitter robust state [PST05].

When it comes to interactive applications like VoIP and videoconferencing, there exist no such remedies. Both codec delay and network delay must be kept to a minimum. For human conversation, the rule of thumb is 150ms one-way delay [Int96]. For musical collaboration, 5–20ms is more likely the limit [CG04]. The latter puts strong focus on an integrated codec-packetizing-submit design and implementation (IP stack handling, real-time OS, and so forth), and a limitation on maximum distance due to propagation delay. “Normal” videoconferencing use 25 or 30fps CIF or VGA sized video. This means that there is 33–40ms time separation between each frame. If the jitter was small compared to this time separation, no receiver buffering would be required.

4.3.2 Packet loss

Media decoders generally tolerate some information loss at decoding/rendering time. Rule of thumb for MPEG-4 says that 2–5% packet loss is acceptable, except if error events are too bursty. An important application requirement is that the decoder itself must not stop working, even if the needed next information packet is not received in time. A well designed decoder will try to conceal missing information, by one or more advanced tools. If a video key frame (I-frame) was fragmented into several packets, only the first packet will normally contain the frame heading information, such as quantization scale used, etc. Such inband side information is crucial for the decoding process, and if missing, any further decoding of such a frame can be a total waste (a decoder might guess parameter values). The error resilience can be strengthened by repeating the header information in all packets, at the cost of reduced compression efficiency. Each new frame should have special markers that enable the decoder to resynchronize and understand when a new frame has started. Missing packets leading to frame discard at receiver can be dealt with by duplicating the previous frame. This will however be perceived as a freeze of movement. More advanced error resilience tools can be used, both for e.g. MPEG-4, MJPEG2000, and the state-of-the art H.264/AVC [Bra99, FXZH00, KKVS04, TABM05].

One possible error resilience tool not very much considered is *interlacing resilience* (IR) [HR03]. As long as packet losses are randomized (which they would if congestion takes place at an Active Queue Management router, to be discussed in Chapter 5.2), a strong robustness can be achieved with splitting each frame into two “interlaced” frames (both captured at same time, or half frame interval offset as in television systems). Simple pixel interpolation can be used for the part of the frame missing decodable information, assuming that the other “interlaced” frame is not missing. Even better robustness is achieved if splitting the image into four sub-frames, possibly at the cost of even more decreased compression efficiency. This method has been investigated to some extent during this thesis period, and some unpublished results are shown in Appendix B on page 235.

4.4 Congestion control

To balance latency and loss at low levels while keeping a high link utilization (close to traffic overload), a robust and stable congestion control regime is needed. This subchapter lists the available and proposed control schemes for both elastic and streaming traffic.

4.4.1 Elastic traffic congestion control

The TCP congestion control algorithm has been refined several times. Basically, the refinements are all based on “binary” network feedback, i.e. packet drops or (ECN) marks signaled as acknowledgement and the lack of acknowledgement. Only one variant (TCP

Vegas) uses RTT estimation as a non-binary metric to signal the amount of network traffic contention. In Table 4.1 the main contributions of the TCP development the last decades

Table 4.1: The TCP development history

Year	RFC	TCP Name	Title
1974	675	original TCP	“Specification of Internet Transmission Control Program”
1981 1988	793 + [Jac88]	Tahoe	“Transmission Control Protocol” “Congestion Avoidance and Control”
1996	2018	SACK	“TCP Selective Acknowledgment Options”
(1997) 1999	(2001) 2581	Reno	(“TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms”) “TCP Congestion Control”
(1999) 2004	(2582) 3782	NewReno	“The NewReno Modification to TCP's Fast Recovery Algorithm”
1994	[BOP94]	Vegas	“TCP Vegas: New techniques for congestion detection and avoidance”
(1999) 2001 2003	(2481) 3168 3540	ECN	“The Addition of Explicit Congestion Notification (ECN) to IP” “Robust Explicit Congestion Notification (ECN) Signaling with Nonces”

are listed. TCP Tahoe (RFC 793 + [Jac88]) adds Fast Retransmit (after 3 duplicated ACK packets), while Reno added Fast Recovery (Fast Retransmit coupled with halving the congestion window before additive increase, instead of first running slow start). NewReno has more intelligent slow-start at multiple packet loss. Vegas is the only technique labeled “congestion avoidance” instead of “congestion control”, since it is able to slow down the TCP rate before a packet loss occurs. The SACK option improves the efficiency in that acknowledgement can be given to certain packets, and not cumulative as is normal. The self-clocking behavior of TCP, i.e. controlling new packet submission time by ACK packet reception events, assists in limiting traffic burstiness, both in slow-start and AIMD state.

ECN (Explicit Congestion Notification) is the only technique listed in Table 4.1 that requires network support, and can be used as option for any TCP flavor except Vegas. With ECN, two bits of the TOS field of the IP header are used to flag a congestion state. If both bits are zeroed by the sender, the end points do not support ECN. If one of the bits is set, a congested router will set both bits to one. The best known router architecture that supports ECN packet tagging is RED (Random Early Detection) [FJ97], which will be explained in Chapter 5.2 on page 54.

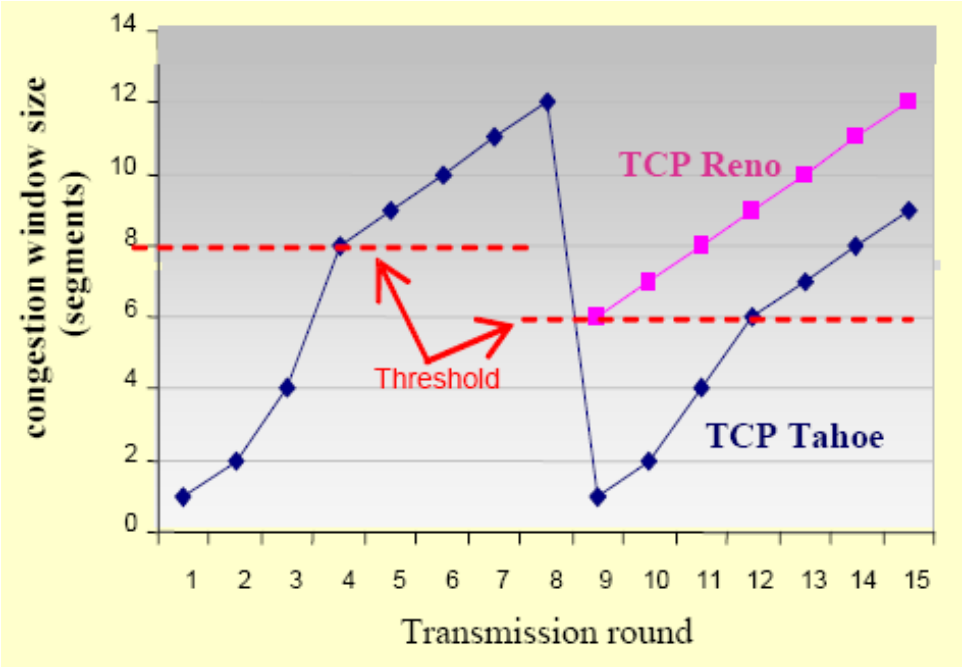

Figure 4.1: The TCP Reno Fast Recovery helps increasing the throughput (figure copied from Chen-Nee Chuah, Univ. of California, Davis).

During the last years, even more modifications and additions to the TCP flow and congestion control algorithms have been developed. The main focus has been how to increase the efficiency of TCP over links with a high BDP. The increase rate is controlled by the RTT. A 200 ms RTT link of 1 Gbit/s will take a single flow only 2.8 s to reach full capacity during the initial exponential slow-start. But then the congestion window is halved, and (at least in Reno) the AIMD operation state is started, adding one more packet per RTT: this would take the flow almost half an hour to reach full capacity again (using 1500 byte packets)! Thus, numerous suggestions have been proposed to address this issue. The most common method used today is to establish multiple TCP connections for file transfer, since N TCP flows will be N times more rate aggressive than a single flow. However, this comes on the cost of unfairness against other applications not using this dubious practice. Thus, new TCP architectures are proposed that can utilize large BDP pipes without resulting in unfairness. Among these proposals are High-Speed TCP [Flo03], Scalable TCP [Kel03], Hamilton TCP [SL04], Fast TCP [JWL04], and BIC [XHR04] to name the most known. All of these proposals should work using legacy FIFO routers. Other proposals have even better performance, on the cost of new router architecture, e.g. XCP [KHR02], PTP CADPC [WM02], and RCP [DKZSM05]. As will be shown in the papers in Part II, this thesis proposes a *media aware* network router and related congestion control that have similarities to both XCP and RCP. Some more details of XCP and RCP in the context of streaming media will be given in subchapters 5.2.1 and 5.2.2.

4.4.2 Media congestion control — Requirements

Media sessions are generally long lived sessions. When comparing to long lived (elastic) TCP sessions (e.g. FTP and HTTP file download), three differences strike:

- The media session generally requires a persistent bandwidth, while the TCP session generally only requires an efficient session completion time (within which the throughput rate might vary significantly).
- The media itself might (if VBR constrained encoded) have significant fluctuating sending rate at time units smaller than GOP size, while the elastic applications allow the TCP congestion control to adjust to smooth rates (AIMD, except when halving the congestion window).
- The media flows prefer low end-to-end delay before packet loss, while the TCP sessions prefer low loss before low end-to-end delay.

This can be summed in the first performance requirement for media rate control:

Media rate control Requirement 1*: The media congestion control must support steady bandwidth when averaging the rate over ~0.5 s (i.e. one GOP period of networked media, see subchapter 2.2.3 on page 22) and above time scales, while tolerating significant variations on smaller time scales. This should be feasible simultaneously with low queuing backlog, modest packet loss, and high link utilization.*

When mixing real-time and elastic traffic within the same class, such as the best effort class, the bandwidth resource sharing discipline is important. The general requirement today is, if a new congestion control regime is invented, it should be designed so that it supports *TCP-friendliness*: the acquired throughput on average should equal that of an equivalent TCP session, having the same network path. Further requirements are fast reaction to traffic load changes, so that packet losses are minimized, and available spare bandwidth is utilized, if possible. In addition, the solution should be scalable to large networks, and have a low complexity deployment strategy. This can be listed as the following additional formal requirements:

Media rate control Requirement 2*: The media congestion control must be TCP friendly,*[13] *so that media sources using it will be regarded as well behaving "Internet citizens". This should also implicitly result in intra media flow fairness.*[14]

Media rate control Requirement 3*: The media congestion control must be scalable to large networks, and stable and robust to a large range of RTTs.*

13. If deployed in a separate class than TCP, eventually in a dedicated media network, this requirement can be dropped.
14. Please be aware that an amendment is made to this requirement, see page 49.

Media rate control Requirement 4*: The media congestion control should have fast reaction speed to traffic load fluctuations.*

Media rate control Requirement 5*: The media congestion control should have an easy Internet deployment strategy. i.e. an incremental deployment strategy should be feasible.*

Note that the last two requirements have been given "should" instead of "must". It can be argued that in order for a new scheme to be fulfilled, there is a need for a strong incentive to "follow the new rules". Why should a media service use a new congestion control scheme if it is better off (i.e. can use more bandwidth) without it? One incentive that follows automatically is that the perceived performance of rate adaptive media will degrade *gracefully* at increasing traffic load. This is in contrast to non-adaptive media which sponge on the adaptive sources mechanisms until a sudden service breakdown when packet loss starts to occur. Some additional comments on "ill-behaving" sources are found in Chapter 6.2. As far as the media requirements are concerned, it is assumed, even though perhaps being considered naïve, that new services follow them because of the acceptance that network resources must be shared in a righteous way. Thus, finding additional incentives for following these rules are out of the scope of this thesis.

The last part of this subchapter will be used to list some of the best known media congestion control mechanisms, and how they fulfill the five media rate control requirements listed above. In addition to the selected list of methods, additional proposals could have been included, such as [KMR93, LMR97, Mis95] and the SAVE algorithm as described in [SR01] chapter 10.6. Of quite recent work of interest could be mentioned the work of Jammeh et al. [JFG07] based on fuzzy logic.

TFRC

TFRC [HFPW03] (TCP Friendly Rate Control) has been through an ongoing development during many years, and is today an IETF standard RFC3448. It has also become one of the congestion control methods of DCCP, namely Congestion "Control ID 3" RFC4342 [FKP06]. Within the DCCP framework, it is intended as a unicast congestion control for a connection oriented unreliable media flow. It is connection oriented so that penetration of firewalls is more likely (port numbers can be negotiated). It is unreliable as UDP is to avoid retransmissions (media decoders are error resilient). It is TCP friendly because its rate is calculated by a TCP throughput equation, giving the averaged TCP rate at stable AIMD operation over several "sawtooth periods" [Bou06]. The throughput equation, first published by Padhye et al. [PFTK98], is

$$r_{\mathrm{TCP}} = \frac{M}{t_{\mathrm{RTT}}\sqrt{\frac{2Dl}{3}} + t_{\mathrm{RTO}}\left(3\sqrt{\frac{3Dl}{8}}\right) l(1+32l^2)} \tag{4.3}$$

where M is packet size, D is the number of acknowledged TCP packets per acknowledgement packet (ACK packet, normally 1 if not delayed ACK is used), l is the loss fraction, and t_{RTT} and t_{RTO} is the RTT and time-out estimate, respectively. This function gives the maximum acceptable *packet rate* calculated by parameters that are derived from receiver feedback, and thus characterize the level of network congestion. TFRC avoids TCP's aggressive rate drop (50%) by combining much smoother increase as well as decrease. Thus, in general, the response time to traffic load changes are slow. If however ACK packets are missing in several RTT periods, the sender slows the sender rate and ultimately stops sending. The ACK packets include the sequence number of the last received packet, along with timing information. This enables the sender to compute t_{RTT}, after having filtered such estimates from an exponentially weighted two-tap moving average filter. t_{RTO} is estimated in similar manner from this RTT estimate as in TCP (this parameter is only used in the equation, since no retransmissions take place). The loss fraction l is calculated in the receiver and returned to the sender via the ACK packets. The details on this calculation are not included here, but the important fact is that the value is a result from an averaging function, taken over several *loss intervals* (the time between packet loss epochs). This ensures smooth l values, and thus a smooth rate when in stable operation at stable number of flows.

The TFRC sender (e.g. a video encoder) changes its rate controller so that its packet rate does not exceed r_{TCP}. In practice, this can be done using a sender buffer between encoder (or streaming server) and the TFRC stack, which is drained by a packet rate of r_{TCP}. Since TFRC gives packet rate, and not bit rate, full packets should be transmitted using "byte stuffing" when necessary since "TFRC is designed for applications that use a fixed packet size" [HFPW03].

During the last year, new variants of TFRC are starting to develop. TFRC-SP (TFRC Small Packets) is targeting VoIP applications, to address the challenge of using variable packet sizes. Several Linux kernels and kernel patches are made available with DCCP/TFRC included during 2006/2007.

RAP

RAP [RHE99] (Rate Adaption Protocol) by Rejaie et al. is a AIMD based "TCP-friendly" media congestion control system. It runs over legacy FIFO router network, and includes a media server part and a client player part. The paper says it is best suited for unicast layered pre-coded media. RAP congestion control is loss based, but could become even more efficient if an explicit congestion signal from the Internet was available. RED routers improve the TCP friendliness. The packets are ACK'ed as in TCP. The simulations carried out are missing media packetizing, and use sources following the actual RAP rate perfectly. The RAP rate is generally saw-toothed. Thus it is unknown if Requirement 1 is fulfilled. Also, TCP-friendliness depends on the window size of the TCP end-points. The

fairness ratio is far from 1 in many simulation results. Transient performance is not tested, but seems to be fast to downscale the rate due to 50% multiplicative reduction. Increasing the rate with AIMD seems slower as the step-size is fixed.

LDA+

LDA+ [SW00] (Loss Delay based Adaptation) by Sisalem and Wolisz uses RTP and RTCP protocols to collect delay and loss statistics. LDA+ extends RTCP by adding the estimation of the smallest link capacity on the path in the application part of the control packet. The AIMD philosophy is used in LDA+ as in RAP. LDA+ uses (4.3) to calculate the rate during packet loss events. During no loss, LDA+ uses the minimum link bandwidth estimate to adjust the additive rate increase so that faster convergence to full utilization is achieved compared to RAP. Based on the results of [SW00], it seems like the TCP friendliness is better than in RAP.

TEAR

TEAR [ROY00] (TCP Emulation at Receivers) by Rhee et al. shifts almost all control to the receiver. The TEAR receiver emulates the TCP congestion control at sender, and uses the congestion signals (packet losses and timing) at forward path to calculate a TCP-friendly rate. This rate is periodically sent back to the sender. The authors claim that this period can be up to 10 RTTs without sacrifice too much performance. The TEAR receiver is a complex seven-state machine, mimicing the different TCP states. Besides the state machine the algorithm is of fairly low complexity. TEAR is applicable to multicast, due to its receiver driven architecture. It shows very good fairness results, comparable or better than TFRC. The comprehensive report shows however very little other information than fairness tests with similar number of TCP and TEAR and TFRC flows over FIFO and RED routers. Heterogeneous RTTs, and large RTTs, are not included in the report, thus robustness to network complexity is difficult to evaluate from [ROY00].

Table 4.2 is not intended as a absolute reference comparing the listed media rate control methods, but just as a rough overview of pros and cons. As already stated, TFRC is the method selected by the IETF, which also gets the best scores in this evaluation. In recent years there have been more papers suggestion improvements to these methods, e.g. [JFG06, MSPZW04].

None of the evaluated methods can show both stable operation in “steady state”, combined with fast response in situations with rapid traffic load change. In addition, one can question the strive for TCP friendliness. The reason for this questioning is that media rate control also is to be used by interactive services like VoIP and videoconferencing. Due to TCP’s “proportional fairness” property [KMT98], long flows (large RTT) will be granted lower throughput than short flows (small RTT), even when they are sharing a single com-

Table 4.2: Pros and cons of the investigated media rate control proposals. "?" means "not investigated", "–" means low performance, "0" means average performance, while "+" means good performance.

Req.	TFRC	RAP	LDA+	TEAR	Comment
1 (BW)	+[a]	?	?	?	The score is high if tested positive with media traffic
2 (TCP friendly)	+	–	0	+	High score if fairness index is close to one.
3 (scalable, stable)	+	+	+	+	High score if it does not oscillate at high RTT values, or special parameters need special tuning.
4 (transient)	-	-	0	?	High score if rate reduction is proportional to the traffic overload, and rate increase utilizes spare bandwidth within a few RTTs.
5 (deployment)	+	+	+	+	The score is high if the protocol works on legacy FIFO routers.

a. TFRC is tested with media applications by this author [LK07] and others [BENB06].

mon bottleneck. The question is: is proportional fairness an appropriate fairness criteria for media traffic? If so, end users must accept that the voice quality of inter-continental VoIP calls will have poorer quality than local calls (assuming both is performed in best effort class Internet). In this thesis, global max-min fairness property [BG92] has been the ultimate goal, based on the aforementioned argument, since max-min fairness is RTT independent. (Actually, some TCP variants also strive for RTT independency, e.g. H-TCP [SL04] and XCP [LAW05] and RCP [DKZSM05].) Thus, an amendment to Requirement 2 can be made:

***Media rate control Requirement 2 Amendment**: The media congestion control must be global max-min fair if media services should exhibit RTT independence.*

Of other proposals not included in this overview can be mentioned plain RTP with RTCP feedback, proposed by several, among them [SCFJ03, WHZ+01]. Hsiao and Hwang [HH04] have proposed an extension to XCP to support layered video.

Based on this set of requirements, a visual matrix exemplified in Table 4.3 is used in Chapter 5.4 to make it easier for the reader to understand what focus the different papers included in Part II have. Empty table cell means no focus, lowercase "x" means some focus, uppercase "X" means significant focus, and bold uppercase "**X**" means detailed focus.

Table 4.3: Example of a method focusing on fairness but most on adjustment speed.

1	2	2Amd	3	4	5
	x	x		**X**	

5
Thesis research

From peaceful minds do great ideas flow.

Neale Donald Walsch — US spiritual writer (1947–)

5.1 Research goals and constraints

5.1.1 Interactive communication: low delay even at high load

This thesis work started with a clear goal: conversational (i.e. interactive) media over IP! To stress this further, even music collaboration with extreme low-latency requirements was examined as a case (Paper A). Thus, the whole chain of codec, rate control, and network buffer dimensioning should add minimal values to the total end-to-end latency budget.

Low network queuing delay is not a challenge until the traffic load is close to a links capacity. As stated earlier in Chapter 4.1, buffer sizing should be larger than in REM design, and smaller than in typical RS design. Another important revelation is that link capacities continues to grow. Instead of using this as a motivation for over-provisioning, it can be a motivation for high link utilization. Actually, as link capacities continue to grow, and input traffic characteristics become more and more Poisson [KMFB04, LK07], a certain offered load to a link will give less and less queuing delay. This is evident when looking at the delay profile for an M/D/1 queue, which have the Laplace transform of its waiting time equal to (J. F. Hayes)

$$W^*(s) = \frac{s(1-\rho)e^{-ms}}{s-\lambda-\lambda e^{-ms}} \tag{5.1}$$

where $m = 1/\mu$ is the deterministic packet service time (fixed packet sizes assumed), λ is the packet arrival intensity, ρ is the applied work load λ/μ, and s is the Laplace transform operator. The expected waiting time can now be calculated as

$$E[W_{\mathrm{M/D/1}}] = (-1)\frac{d}{ds}W^*(s)\Big|_{s\to 0} = \frac{m(2-\rho)}{2(1-\rho)}. \tag{5.2}$$

If the Inverse Laplace function of (5.1) had existed, this would have been a correct expression for the *distribution* function of the M/D/1 waiting time. However, [RMV96] managed to find a closed expression for the M/D/1 waiting time CDF saying

$$W_{q,\,\mathrm{M/D/1}}(t,\rho) = (1-\rho)\sum_{k=0}^{\lfloor t \rfloor}\frac{[\rho(k-t)]^k}{k!}e^{-\rho(k-t)} \tag{5.3}$$

where the service time m is normalized to one. This function is plotted in Figure 5.1 together with the corresponding CDF for M/M/1, which is [GH98]

$$W_q(t) = 1-\rho e^{-\mu(1-\rho)t}, \qquad (t \geq 0). \tag{5.4}$$

It is seen both visually, and easily verified by dividing the equations on each other, that the M/D/1 queue has only half the waiting time of an M/M/1. However, the most important observation is the general scaling of time units: If the service time m is 1 second, then 99% of the packets will experience a waiting time less than 12 s. If the service time is 1 µs, then the corresponding result is 12 µs. If defining a 99% requirement of 1 ms queuing delay, the corresponding traffic load at different service times (i.e. link capacities) are as

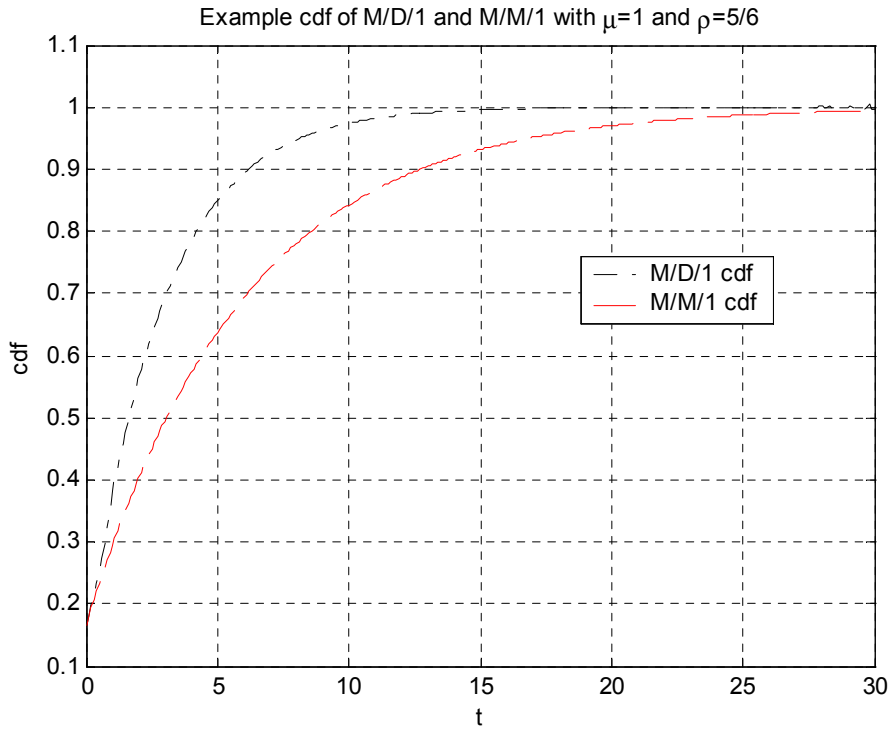

Figure 5.1: The CDF of the waiting time distribution of M/D/1 and M/M/1 when offered load is $\rho = 5/6$.

Table 5.1: Possible M/D/1 traffic loads at different link capacities where 99% of the packets will experience a waiting time of 1ms or less and 10ms or less, assuming 1000 byte packets.

Link capacity	1Mbit/s	10Mbit/s	100Mbit/s	1Gbit/s	10Gbit/s	100Gbit/s
Traffic load, *1ms waiting time*	0.012	0.18	0.83	>0.99	>0.99	>0.99
Traffic load, *10ms waiting time*	0.18	0.83	>0.99	>0.99	>0.99	>0.99

given in Table 5.1 second row. As seen in row three, if allowing 10ms delay, 0.99 traffic load can be offered at 10 times lower capacity. Higher link capacities are thus a motivation for utilizing available bandwidth even more, as long as rate control is assured. This analytical observation will be verified by results both in Paper A and Paper F in Part II.

5.1.2 Fairness

At start of this thesis work, it was envisioned that the media traffic should belong to a Diff-Serv type of real-time class. Within this class, intra-fairness should govern. RTT-independent max-min fairness was chosen due to the already mentioned motivation of enabling the similar service quality at both short and long distance communication.

Later in the research period, the router architecture was modified to also include low-latency TCP-friendly streaming within flat best effort class type of network, first motivated by reading the paper about ABE [HBTK01], and later by the awareness of the DCCP/TFRC standardization process [FKP06].

5.1.3 Scalability and deployment

As pointed out in Chapter 4.4.2 where the rate control requirements were outlined, the solution should be scalable to increasing size of link capacities, number of network links, and number of network flows. A general rule of thumb is then to avoid per flow state dependencies inside the network, and have limited signaling traffic. The easiest deployment strategy is if the solution can run over legacy FIFO equipped network. The work on TFRC and DCCP were well established by the IETF in the same time period as this thesis research was carried out. It was decided to go for an alternative solution, which included added network intelligence. A new router architecture was created with special support of the media flows requirements. This makes the Internet deployment issue more challenging. However, with the scalability issues included in the requirement list, the solution should be feasible for large networks, being Internet segments, or dedicated media networks to support VoD and WebTV/IPTV services. The router architecture focuses on

congestion control support, and belongs thus to the *Active Queue Management* (AQM) research arena.

5.2 Active Queue Management

5.2.1 AQM for elastic flows

In order to provide the adaptive media rate controllers with fair and correct rate feedback information, the aggregate traffic entering the network routers must be monitored accurately and adequately. A router that perform such tasks to assist more efficient congestion control than plain FIFO packet tail dropping is said to provide *Active Queue Management*. Many AQM architectures have been proposed during the last decade to control elastic data traffic (i.e. not real-time traffic). An early proposal of such a system is found described in [Rø84]. The first well known AQM architecture in the literature is RED [FJ97], which was developed to increase the efficiency and throughput of elastic TCP traffic. RED calculates a tagging (or dropping) probability that corresponds to the congested state (see Figure 5.2). The congested state is decided based on the averaged queue length. Thus, a congested router will not tag (or drop in case ECN is not supported by the end nodes) all packets during a congested period, but uses instead randomized tagging so that the amount of packets tagged corresponds to the congestion level. This operation also helps avoiding synchronized TCP congestion reactions since bursty packet losses are eluded, i.e. the TCP flows do not lose packets simultaneously. With other words, the queue backlog can be more stable leading to increased link utilization. However, the main motivation of deploying RED and ECN is that the number of TCP retransmissions can be significantly reduced, since much fewer packets are dropped (dropping occur only at full router).[15] Again, this increases the throughput of the network (faster session completion time). In addition, when working at queue equilibrium it will still have buffer space available to support low packet drop ratios for bursty sources. Generally, AQM also enables lower queue sizes, which reduces delay and delay jitter. Le Boudec [Bou06] argues that AQM is *not* about detecting misbehaving flows and protecting adaptive flows from being starved. Others, such as Lin and Morris [LM97] argue that it should.

The main challenge with many AQM routers, including RED, is that the congestion control algorithm has parameters that can be difficult to set once and for all. This may result in an oscillating behavior, i.e. the queue size does not converge towards a stable level, but vary heavily between large backlog and completely drained queue. The latter limits the high link utilization that was envisioned. The queue oscillations are unwanted, especially for real-time traffic, since this increases the delay jitter.

15. ECN should then also be regarded as congestion avoidance technique, as TCP Vegas is.

This has resulted both in research on RED improvements and the proposals of other AQM regimes. In fact, an online paper from Bitorika et al. [BRHG04] claims that in 2004 there were at least 50 AQM proposals published in the period 1999–2004. The current "state-of-the-art" RED technology is named "gentle adaptive RED" where the static parameters have well-defined agreed values [Flo] while the algorithm adapts the dynamic parameters after a set-up period. Of other derived RED proposals can be mentioned FRED [LM97], SRED [OLW99], and RED-PD [MF01]. Completely different architectures can be found in CSFQ [SSZ98], BLUE [FKSS02], GREEN [WZ02], REM [ALLY01], CHOKe [PPP00], AVQ [KS01], and PI [HMTG01]. Some of them (FRED, RED-PD, CSFQ, Stochastic Fair BLUE, CHOKe, and to some extent SRED) also incorporate methods for fair bandwidth sharing and handling of unresponsive flows. Most algorithms monitor current queue backlog or an exponentially weighted moving average of the queue backlog (RED, RED-PD, PI, FRED, SRED), some monitor the input rate (AVQ, BLUE, CSFQ, GREEN), while some do both (well, at least REM does).

All of these were designed to support TCP congestion control into improved performance. However, it can be argued that many of these should also support end-to-end regimes for media traffic control such as DCCP/TFRC. Actually, Paper E shows that TFRC can gain about 4dB PSNR performance per video flow when using gentle adaptive RED instead of FIFO routers, and additionally 4dB if the RED network also supports ECN. Most of this gain is obtained by the shorter and more stable average queue backlog supported by the RED routers (those tests had end-to-end latency requirements). Since many of the AQM algorithms proposed after RED have claimed superior performance, it is reasonable to propose "why not test TFRC using every invented AQM in turn, then compare, and proclaim a winner!". Indeed, such an examination would have been interesting. However, when looking into many of the papers comparing performance of different AQM scheme,

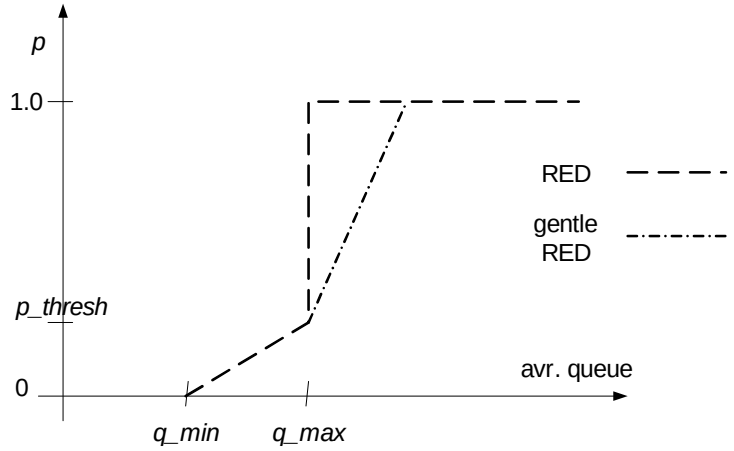

Figure 5.2: RED maps an ECN tagging (or packet drop) probability p to the current averaged queue backlog. If the queue is less than q_min, there is no tagging. Between q_min and q_max the probability increases linearly towards p_thresh. Above that level, RED marks all packets, or increases fast to all as in gentle RED.

the results are to some extent ambiguous. The reason is most likely to the aforementioned problem of selecting AQM parameter values [MBDL99]. Since gentle adaptive RED, part of the ns-2 library, shows significant stability improvement compared to the old "static" RED, the author believe gentle adaptive RED would perform very well in such a competition. Nevertheless, the results obtained when testing TFRC over RED routers (Papers D, E, and F) shows that the current performance of such end-to-end regimes is not satisfactory for low-delay real-time communication in resource constrained networks. This motivates for alternative solutions.

Before looking into this thesis proposed solution, some new interesting and radical new protocols for elastic flows, MaxNet [WAZ03], XCP [KHR02], and RCP [DKZSM05], must be mentioned. These methods incorporate novel AQM router algorithms, designed together with a novel non-TCP-compliant host congestion control algorithm. All of these have an AQM algorithm monitoring both input rate and queue backlog. They all calculate a link state expressed as a multi-bit number, signaled periodically via packet header tagging. Simulations have shown that if all flows follow these feedback signals, link utilization would be close to 100% and with very small queue backlogs. RCP is the latest innovation of these three, and claims superior performance, especially in fast session completion times and providing global max-min fairness. XCP's max-min fairness performance is questioned by Low et al. [LAW05], but XCP is also the only one with a draft IETF recommendation. Anyhow: could these methods support rate adaptive media as well? Low queue backlog is one of the most important requirements (see Requirement 1 page 45). However, the VBR constrained media sources will have significant variance, which might ruin the stability of these methods. Nevertheless, they include solutions that have a clear resemblance to the main contribution of this thesis, the "P-AQM".

5.2.2 P-AQM — AQM for rate adaptive real-time flows

To the best of the author's knowledge, P-AQM with "ECF" and "ERF" feedback system, presented in this thesis, is a novel AQM proposal special designed to support *rate adaptive media traffic* in IP networks in general, and live interactive flows in particular. There are *some* proposals available that supports related technology, such as Hsiao et al. [HH04], Parris et al. [PJS99], and Chung et al. [CC00a, CC00b]. Hsiao has in fact tested XCP with "layered video" in ns-2. However, the video is highly synthetic and any rate variability is non-existent. Parris et al. and Chung et al. (CBT and Dynamic CBT) have their focus mostly on how to tackle unresponsive UDP flows, but in their last paper also a five level layered video approach is supported.

The P-AQM on the other hand is "tailor-made" for adaptive media flows, but with added support for elastic flows. Figure 5.3 shows the final design of P-AQM with a two-queue scheduler ensuring that the UDP (real-time) congestion control can be designed indepen-

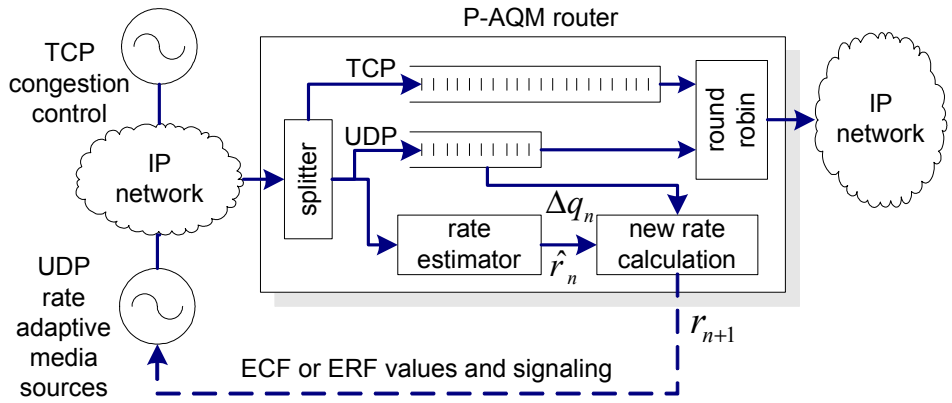

Figure 5.3: The P-AQM router decouples elastic and real-time traffic. Its traffic load metrics of rate adaptive media traffic is signaled back to the source via end-to-end signals (ERF) or direct ICMP signaling (ECF).

dently of TCP (elastic) congestion control.[16] Thus, RTT independent TCP-friendliness is achieved through the outbound capacities given to these two queues. A flow number estimator (not depicted in the figure) gives weights to the round robin scheduler, which is in charge of this throughput balance. This weight is calculated once per feedback period. The TCP queue buffer is dimensioned like a legacy FIFO or RED router, supporting either packet dropping or ECN marking, using the Paper D P-AQM inner-loop algorithm. The smaller UDP queue runs two independent AQM algorithms: the inner loop algorithm (Paper D), and the outer loop algorithm (Paper F). The new rate calculation metric is based on both current input bit rate and current queue backlog (see final algorithm in Paper F). Fairness between the media flows is obtained through the use of periodic multi-bit feedback metrics (see next subchapter). It is the outer loop that has clear similarities with XCP and RCP. E.g., the RCP rate equation is (by using the same definitions as in [DKZSM05])

$$R(t) = R(t-T)\left[1 + \frac{\frac{T}{d_0}\left(\alpha(C - y(t)) - \beta\frac{q(t)}{d_0}\right)}{C}\right] \tag{5.5}$$

where R is the estimation of fair rate (to be signaled to the sources), T is the period update interval (typically several per RTT), d_0 is the RTT average of the flows passing through the router, C is the outbound link capacity, $y(t)$ is the measured input rate during last update interval, $q(t)$ is instantaneous queue backlog, and α and β are stability constants. The P-AQM variant with closest resemblance to RCP is P-AQM+ERF (from Paper F) which has rate equation (using a mapping into the same variable definitions as RCP above)

16. See Chapter 6.2 on page 74 for how P-AQM handles other protocols.

$$R(t) = R(t-d) + \frac{\alpha(C - y(t)) + \beta\frac{\Delta q_n}{d}}{\hat{N}_n} \tag{5.6}$$

where d is the update period (and should be larger than average flow RTT and/or video source GOP period), Δq_n is instantaneous queue backlog compared to a wanted queue backlog equilibrium, while $\hat{N}_n = C/R$. The main differences between these two are that the periodic signaling is performed more seldom, typically in the order of twice per second, and that a residual but small queue backlog is wanted. A somewhat simplified pseudo code of P-AQM is shown in Appendix C.

While ECN [RFB01] has introduced the possibility of signaling congestion without dropping packets for TCP sessions, the exact level of congestion is very hard to predict. Xia et al. argue in their paper "One more bit is enough" [XSSK05] that if the two ECN bits are redefined to signal a three-level congestion state, a much better control would be possible. Paper F gives the reasoning why their VCP architecture "collapses" when there are multiple bottlenecks, i.e. it shows that long distance flows are choked by short distance flows! In P-AQM therefore, but also in XCP, RCP, and MaxNet, the congestion level is signaled as a multibit number (e.g. 32 bit floating point). In addition, the P-AQM signaling is periodic at typical GOP time scales (~0.5–1.0 second), using either separate signals from each router (the ECF solution), or using packet header field updating at each downstream P-AQM router (the ERF solution). More details will be given in Chapter 5.4, and of course in the papers, especially Paper D and F.

A final note: In the name *P-AQM* the "P" stands for *Proportional* (AQM). It was first used in Paper D when naming the "inner loop" invention of Paper B and C. Since the "outer loop" of P-AQM performs *max-min fair* rate control (see Paper F), which fairness goal deviates from the established *proportional fairness* of TCP [KMT98], the chosen name might seem a bit misleading. Recognizing that the naming has some improvement potential, it was nevertheless decided to keep it also for Paper E and F.[17]

5.3 Research methodology

The design of the final AQM architecture and feedback structure is a result of incremental improvements presented throughout the papers. Each paper has an initial idea, a design concept or design hypothesis. In the first papers (see Figure 5.5) these concepts were tested through software implementation and simulated in selected network architectures

17. After Paper D it was discovered that there existed two other AQMs designed for elastic flows named *PAQM* (without the dash). In [GHH02] the "P" stands for *Predictive*, and in [RRQ04] the "P" stands for *Pro*. Both tries to utilize correlation in the traffic patterns to predict future traffic load and therefore create better control of elastic TCP traffic.

in Demos/Simula. The simulations were conducted in order to reveal performance (e.g. the transient traffic load behavior, or the steady state delay and packet loss). Later the design was funded on control theoretical design principles, and the simulation platform was moved from Demos to ns-2. Ns-2 has validated TCP models, and is therefore a compelling simulation platform to verify TCP-friendliness. Simple dumbbell architectures (one flow per input link, one common bottleneck link) are used in several of the papers to test e.g. bandwidth sharing fairness. However, also more complex architectures like parking lot topologies (some flows transversing many bottlenecks, with other flows only utilizing one bottleneck, so-called cross-traffic) are also used in several of the first papers. Paper F also includes the GFC-2 network, which is a complex network structure originally invented to test bandwidth fairness of ATM networks [Sim94]. To better test the networks with realistic video traffic, the Evalvid-RA tool-set was developed.

It should be noted that the research work generally has put more focus on simulation tools than on mathematical analysis. In Paper B and C the control theoretical analysis of the "inner loop" was presented and used as basis for the core P-AQM design, and backed up by simulation verification. The design and stability analysis was performed using Lyapunov tools, which actually seems not so uncommon in AQM design these days, see e.g. [LAG07, ZqYyM+07]. In Paper F, a complete control theoretical analysis was established of the "outer loop" in order to set sound scaling parameter values to ensure stable operation in a large range of variable network conditions. For this analysis classical Bode/Nyquist control theory was applied to a fluid flow approximation of the packet network system, which helped finding the stability region for the scaling parameters α and β. One can assume that such an approximation will be more correct in higher capacity networks than in low-capacity networks, while keeping maximum packet size at 1500 bytes (the relative granularity will decrease at increasing capacities). However, this analysis produced results with good resemblance to simulation results already at 16 Mbit/s capacity links. Nevertheless, the Nyquist stability analysis showed appropriate, and is in fact used by many other researchers in this field, e.g. [HMTG01, KHR02, DKZSM05]. Traditional traffic queuing analysis is only used in a limited extent in this thesis. Paper C verified the static traffic load behavior, i.e. queue occupancy, of the "inner loop" of P-AQM to the classical M/D/1 model. There has been no attempts to find a model for the waiting time through a P-AQM router in the closed "outer loop" system. Due to the findings of LRD absence in the VBR streams, a qualified assumption is that the aggregated input traffic can be Poisson modeled. Thus, the M/D/1 model should therefore be a good approximation also for the rate adaptive system with one bottleneck queue.

Both Demos and ns-2 include sophisticated pseudo random generators (PRNG). More precisely, since version 2.1b9 the ns-2 PRNG has been of good quality, because its old PRNG with shortcomings [HE02] was then replaced by L'Ecuyer's "MRG32k3a". Still, the new PRNG can lead to data correlation if the "harmful" seeds are chosen [UR07], so

specific guides must be followed to avoid this. The simulations performed during this thesis work include the following stochastic processes using PRNG: Poisson distributed traffic generator, long-lived flow (UDP, TCP, TFRC) start-time generator (uniformly within some time range, typically the first 0–5 seconds of the simulation), and Web-traffic generator (Poisson distributed session intervals, Pareto distributed flow sizes).

Confidence intervals (CI) of steady state delay conditions are calculated based on *replication method* using independent PRNGs in Paper F. *Sectioning* was not found adequate since the video flow generation of Evalvid-RA is using independent video start-times but with identical GOP sizes, which can lead to "best-case" smooth traffic (the flows key-frames are submitted at non-synchronized time instances during a GOP interval) or "worst-case" bursty traffic (the key-frames are highly synchronized each GOP interval). Thus, this situation created the need for "Monte Carlo" type of simulations. The replication method was thus a natural choice to reveal both smooth and bursty traffic performance results, because the traffic type is decided by the video flow start seed and kept during the complete simulation. The transient part of the simulations in Paper F was removed before performing the CI calculations. Future Evalvid-RA enhancements should include the possibility of selecting independent and different frame rates, or even multiple GOP sizes.

The other papers do not include CI calculations. It could be argued that this should have been performed, e.g. in Paper A Table 1, Paper B Table 1, Paper D Figure 8, and Paper E Figure 7, to increase the result confidence value. It must be noted that in comparative analysis of different systems, these single runs have been performed over long simulation intervals (e.g. typically several 1000 samples per calculated delay mean and variance), and the different compared systems have been started with fixed PRNG seeds to ensure identical conditions. Most of the simulations, before the Evalvid-RA creation, have been performed with rate adaptive CBR traffic (fixed packet size and adaptive packet rate) and rate adaptive Poisson traffic (fixed packet size and n.e.d. inter-arrival times with intensity adapted by the feedback). The input traffic does thus not exhibit any dependencies, even the rate adapted video traffic in Paper E and F was shown to hold very limited dependencies. In addition, the high number of input sources used in many of the simulations creates an aggregated Poisson traffic characterization [Cin72]. All this opens for relaxed statistical analysis. Nevertheless, CI calculations should always be considered. Actually, a survey conducted by the CI supporter Pawlikowski [Paw03] concludes that about 70% of the telecommunication papers during the last decade is *not trustworthy*, among other things due to missing CI calculations!

The simulations are used to verify the mathematical analysis (e.g. Paper B "inner loop", and Paper F "outer loop"). The RL-QoS and P-AQM architecture is not implemented in real routers (e.g. Linux PCs are an attractive router platform), and the new inventions are thus not yet validated in real networks. The other models used in the ns-2 simulations,

such as the TCP New Reno congestion control algorithm, are verified and validated elsewhere.

5.4 Contributions

The papers are listed in their creation order, and thus give the development line from the initial ideas towards the current final design at the end of this thesis (see Figure 5.5). The complete list of published papers are given in page vii. Papers A–C were not concerned with so many general Internet "rules", but were merely ideas about completely different and disengaged packet switched networks. During 2004 the interest of making an Internet best-effort compliant solution was wakened. The work on the router architecture was continued, acknowledging the deployment strategy challenges compared to pure end-to-end regimes using legacy FIFO routers. Several papers compare the P-AQM performance to DCCP/TFRC (Paper D, E and F), to expose the differences and the performance gains that added network intelligence could achieve. However, to serve *both* strategies (P-AQM and TFRC), the Evalvid-RA simulation framework was established during 2005–2006, and presented in Paper E. Evalvid-RA implements MPEG-4 video rate adaptation by quantizer scale adjustments at simulation time. This creates the possibility of running simulations with true video traffic, enabling a deeper understanding of the influence and interplay of video rate controller and network feedback, sender and receiver buffering, and the different congestion control methods. As such, the author hopes that this tool can contribute to current and future rate adaptive media research, regardless of the chosen control strategy. The Evalvid-RA software package has been available for free download since September 2006 at `www.item.ntnu.no/~arnelie/Evalvid-RA.htm`. The final contribution, Paper F, includes a complete control theoretical stability analysis

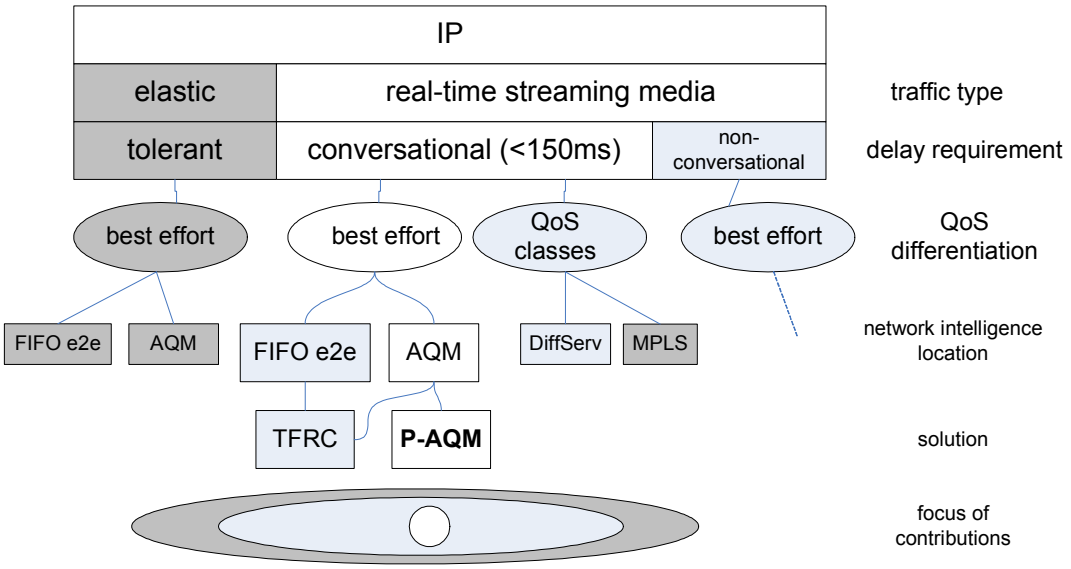

Figure 5.4: The focus of research and contributions. White background is main focus.

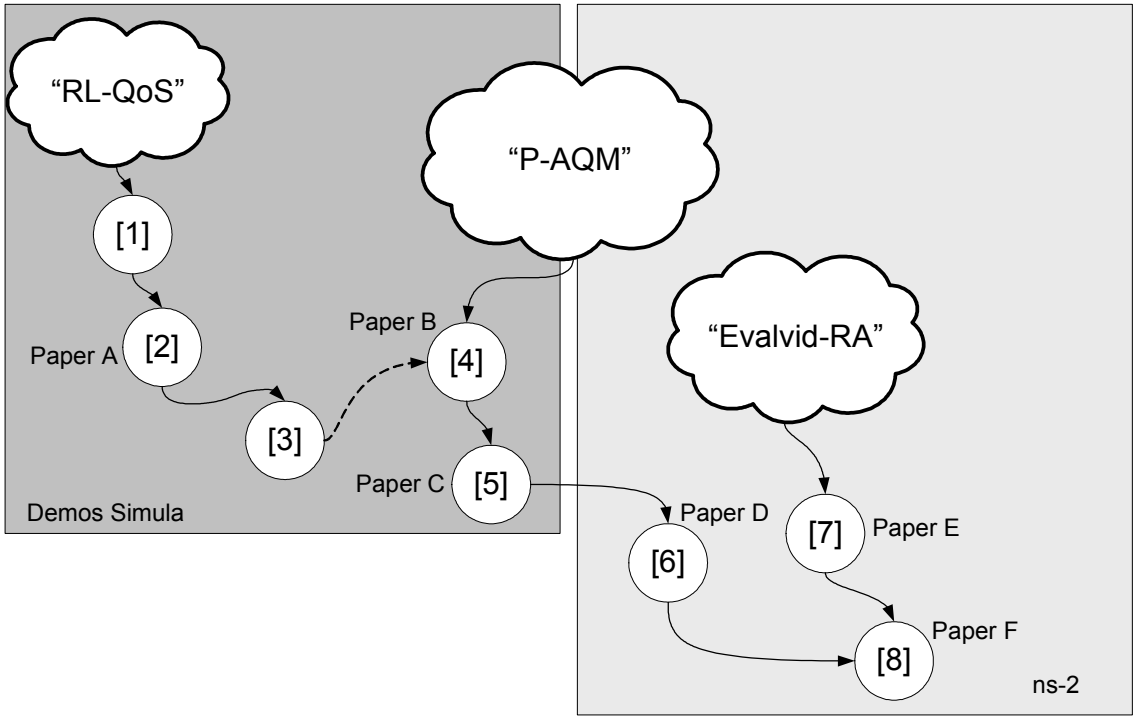

Figure 5.5: The different main inventions, and the related paper numbers (see the overview at page vii). The first five papers used Demos Simula as simulation environment. The last three papers used ns-2. RL-QoS architecture was replaced by P-AQM in Paper B, and moved to the ns-2 platform in Paper D. Evalvid-RA, the framework for real rate adaptive controlled MPEG-4 video, is thoroughly presented in Paper E, and used as an important tool in Paper F.

of the source-network-bottleneck-feedback rate adaptation loop. As a result of this analysis, it presents a modification of the Paper D version of P-AQM+ECF. In addition, it presents the RCP inspired P-AQM+ERF architecture. Finally, it argues and demonstrates control principles that lead to low-complexity and robust global max-min fair bandwidth sharing properties for rate adaptive media.

5.4.1 Paper A

Title: Performance control of high-capacity IP networks for collaborative virtual environments

Authors: Leif Arne Rønningen and Arne Lie

Published in IBC 2002 Conference Proceedings, 12–15 Sept., Amsterdam, 2002.

In this paper [RL02a], which continued the work from [RL02b], the main focus was to show that increased link capacities can be exploited to high link utilization and still low queuing delay and packet loss statistics (as pointed out in Chapter 5.1.1). Musical collab-

*Table 5.2: Overview of paper focus and contributions, in terms of the requirements listed in Chapter 4.4.2. Empty table cell means no focus, lowercase "x" means some focus, uppercase "X" means significant focus, and bold uppercase "**X**" means detailed focus.*

Paper	System Requirements (from Chapter 4.4.2 on page 45–50)					
	1	**2**	**2Amd**	**3**	**4**	**5**
	media BW share	TCP friendly	max-min fair	scalable and robust	transient robust	deployment
A	x				x	
B	x				X	
C	x				**X**	
D	X	x	x	X	x	
E	**X**	X	x	x	x	x
F	X	X	**X**	**X**	x	x

oration is mentioned as an application with extreme delay requirements (about 10ms). The initial router algorithm (later named "RL-QoS") is presented in the paper. It is designed to monitor instantaneous queue lengths and queue length changes each millisecond. The algorithm calculated a "scaling" parameter which should be signaled to all neighboring routers each 5 ms. Each router would perform statistical packet dropping based on these scaling values and its own local scaling metric. Thus the network traffic scaling would react very fast because such packet dropping decisions are taken inside the network. Eventually the scaling messages would reach the sources, which in turn should apply adaptive rate control and thereby avoid further packet losses. The media streaming traffic should be submitted in RTP/UDP/IP packets in a dedicated DiffServ class, while the signaling traffic was envisioned in a negligible delay "expedited forwarding" type of class. The simulation part of the paper shows "proof of concept" using a network example consisting of three congested 1Gbit/s links. The results showed a total queuing delay of less than 10ms and less than 0.3% packet loss ratio in the core network which was utilized almost 100%. More significant packet loss ratios during transient periods was envisioned concealed with interlaced error resilience (see Appendix B for additional information).

The main contributions of this paper was the proof of concept of the RL-QoS AQM architecture given by Demos simulation modeling and results. Our findings were that low delay and marginal packets loss could be achieved at link utilization close to 100%, and that very fast network reaction time to sudden traffic load changes was possible with direct router to router signaling.

5.4.2 Paper B

On the use of classical control system based AQM for rate adaptive streaming media

Authors: Arne Lie, Ole Morten Aamo, Leif Arne Rønningen

Published at the 17th Nordic Teletraffic Seminar, Fornebu, Norway, Aug. 2004.

Follow-up work to Paper A revealed that the router to router signaling, i.e. traffic "back-pressure", was difficult to control in a fair and stable manner. In addition, such a signaling regime was very hard to investigate analytically. Based on these findings it was decided to start over with a new design, based on classical control theory and signaling directly towards the sources.

Paper B [LAR04a] was thus the first paper to present the new P-AQM algorithm design,[18] and to show how the new packet drop algorithm scaled incoming traffic burst peaks such that the queue backlog at overload stabilized at a predefined level. In contrast to RL-QoS, P-AQM's core architecture was, as already mentioned, designed based on classical control theory (continuous-time Lyapunov analysis). Note that the control loop in focus was only the local *inner loop* consisting of input traffic load, queue backlog, queue drainage capacity, and the calculated "valve" opening (u-value) to control the "survival traffic" so that the queue backlog was held at the wanted level N^* . Thus, the analysis did not include the feedback loop (including the media source): this "outer loop" was not mathematical analyzed before Paper F, but simulated in almost all papers (B and D–F). The local loop was tested with both constant and "saw-toothed" shaped peak traffic, and with both Poisson shaped and constant bit rate. The "ECF-report" was defined to be the periodic reports (one per 40 ms, i.e. a typical video frame interval) to signal a running averaged value of this u-value. This value should be used by the sources for media rate adaptation. In turn, the new aggregate rate should arrive close to 100% link utilization and gain almost zero packet loss and controlled queue delay.

The simulation section of the paper includes both rate adaptive Poisson and CBR sources (adaptive in the sense of packet submission intensity at constant packet size of 1500 bytes). These sources fed a network consisting of a mix of long flows and short cross traffic. The *randomized* packet dropping taking place at the "valve" is important for supporting the error resilient tools at decoding time (bursty packet losses are generally more damaging to media decoding than randomized packet losses).

The main contributions of Paper B were (i) a list of requirements for network supported media rate adaptation, (ii) the design presentation of P-AQM's "inner loop" based on Lyapunov analysis, (iii) and the description of the first version of the "outer loop" ECF

18. although the P-AQM name was not given before Paper D.

feedback signaling. Our findings via simulations were that the inner loop performed as envisioned, and that the ECF system managed to keep the sources at a rate balancing packet drop statistics at 0.3% (only CBR traffic) and up to over 3% (only Poisson traffic) at link utilizations in the 97–99% range.

5.4.3 Paper C

Title: Optimization of Active Queue Management based on Proportional Control System

Authors: Arne Lie, Ole Morten Aamo, Leif Arne Rønningen

Published in Proc. of IASTED Communications, Internet, and Information Technology (CIIT), Virgin Islands, Nov. 2004.

This was the second paper publishing the P-AQM architecture [LAR04b]. It was discovered during the work on Paper B that the overall packet loss was dependent on the *Proportional*[19] *gain* constant K_I value: too low value caused a slow reaction to traffic load changes, and thus would cause unnecessary high packet loss in transient periods due to queue buffer space overrun. However, a too aggressive high K_I value would cause high packet losses due to oscillations in the u-value (both locally and for the ECF-report feedback values controlling the rate adaptive sources). Thus, an optimum gain setting between these extreme points should be possible to find. This gain setting was found by simulations of a scenario with one P-AQM node and fixed source rates. Comparison to ideal M/D/1 behavior verified that when the sources were non-adaptive, the average queue size was indeed given by M/D/1 traffic analysis waiting time equation. Different gain settings resulted in different packet loss values at traffic load values in the interesting 95–105% utilization range.

The main contribution of Paper C was thus the search for the optimum proportional gain setting K_I. Our findings were that a gain setting at 100% traffic load was $K_I = 90 \times 10^{-7}$, giving 0.57% drop at Poisson traffic input. However, further research showed that a lower value, $K_I = 15 \times 10^{-7}$, giving 0.69% drop at corresponding traffic load input, was preferred because of more smooth and robust performance at a larger range of traffic load inputs.

19. which actually is the *P* in P-AQM.

5.4.4 Paper D

Title: A performance comparison study of DCCP and a method with non-binary congestion metrics for streaming media rate control

Authors: Arne Lie, Ole Morten Aamo, Leif Arne Rønningen

Published in Proceedings of the 19th International Teletraffic Congress (ITC'19), Beijing, China, 29 August – 2 September, 2005.

Paper D [LAR05] was the first paper showing the complete architecture with the two-queue scheduler decoupling elastic from real-time traffic. It was also the first paper after the move from Simula/Demos to ns-2 simulation platform, where the library of TCP and other protocols made it possible to test P-AQM in a more realistic network setting. The two-queue scheduler design resulted from the initial TCP tests showing that a common small queue for both elastic and real-time traffic gave too low link utilization when dominated by TCP flows (see Chapter 1.3 and 4.4). A larger common queue alternative would ruin the real-time delay requirements of the media flows. Instead of "re-inventing" class-based differentiating, it was decided to create two separate queues, and to base their drainage capacity on estimates of the number of long-lived flows. It was envisaged that this architecture would support common best-effort class, and that TCP-friendliness could be handled inside the router instead at the sources. The "TCP-queue" was equipped with the same AQM "inner loop" algorithm as described in Paper B and C, but also furnished with ECN marking as an alternative to packet dropping. Perhaps as important, it was decided to decouple the local inner loop control (as presented in Paper B and optimized in Paper C) from the outer loop, consisting of the ECF-reports.[20] The reason for this was the realization that the ultimate goal was to balance the input rate from the sources so that at steady-state, there should be *zero packet loss* at the routers, even if the input traffic had some variance. Thus, two different queue equilibrium values were identified: the N^* for the inner loop, and the smaller N_u^* for the outer loop. The inner loop will deal with situations where the input traffic load is increasing too fast, resulting in a maximum queue backlog given by N^* as explained in Paper B and C.

The simulation section showed both transient and steady state performance, and P-AQM was compared to DCCP/TFRC and DCCP/TCP-like performance. Rate adaptive CBR and Poisson models were used as sources (still with fixed packet sizes) for the P-AQM tests, while FTP was used for the DCCP tests. The network was a simple dumbbell scenario. Delay and delay jitter was shown as function of variable RTT. In addition, bandwidth sharing intra protocol fairness was shown as function of flow number.

20. The ECF-reports were decided sent to the sources using 64 byte ICMP SQ packets, after discussions with Sally Floyd autumn 2004 in Berkeley, CA.

The main contribution of this paper was the presentation of the two-queue scheduler solution for P-AQM to support in-router TCP-friendliness and low-delay management of media flows, simultaneously with high link utilization of both elastic and real-time traffic. Further, the ECF feedback was enhanced since Paper C in that the Additive Increase part was no longer static, but adaptive to the observed spare capacity. Our findings were that P-AQM did provide a stable steady state bandwidth share, while TFRC did not. P-AQM showed fast response to changes in the number of flows and robust performance in varying RTT, while DCCP had increasing bandwidth unfairness with increasing RTT.

5.4.5 Paper E

Title: Evalvid-RA: Trace Driven Simulation of Rate Adaptive MPEG-4 VBR Video

Authors: Arne Lie, Jirka Klaue

Published in ACM/Springer Multimedia Systems Journal, online 13 Nov. 2007.

During the work of this thesis papers, the lack of a correct source model for rate adaptive media used within the simulations was recognized as a shortage of simulation results validity. Even though many acknowledge that a large number of flows create Poisson like traffic characteristics [KMFB04], there are small scale variations like I-, P- B-frame variations, packet size modulation, and packet bursts per frame, in addition to average packet rate that will influence the overall performance. Both small scale packetizing effects, GOP variations, and rate controllers delay in adapting to the requested rate can not be modelled with “synthetic” traffic. In addition, the perceived quality could not be evaluated, since there was no true media to be consumed. This motivated the design of “Evalvid-RA”. Early 2005 it was discovered that Klaue et al. [KRW03] had developed an architecture in 2003, EvalVid, for MPEG-4 trace file construction from real (non adaptive) media files, enabling easy creation of streaming traffic generation in real networks. Ke [Ke04] had interfaced EvalVid to ns-2 in 2004, so that similar results could be simulated instead. Combined with the simple VBR rate controller of Hamdi et al. [HRR97], the hypothesis evolving from intuition was the following:

- If the Hamdi VBR rate controller would work in a modified version including *rate adaptive leaky bucket rate*, and
- GOP sized rate adjustment periods[21] were sufficient for the P-AQM system,
- then a pre-processing phase of multiple fixed quantizer scale encodings of the same media file would create a set of trace files that would *support the construction of a rate controller running at simulation time*.

21. which is typically one decade larger periods compared to the periods of 40 ms used in Paper B and C (in Paper D they are dynamically adjusted to be always larger than RTT).

This was a necessary condition, since the alternative of real codecs running at simulation time would be way too CPU demanding. In fact, if this would work out, a *large* set of rate adaptive media sources could be simulated over complex ns-2 network structures on a single computer.

The creation of Evalvid-RA builds on modifications to the original software of Klaue and Ke. The main modification to EvalVid was that the re-assembly post process program had to take into account that multiple MPEG-4 source files (up to 30) would contribute to the single received file. The main modification to the ns-2 interface was the corresponding multiple trace file inputs, and the associated VBR rate controller based on SVBR by Hamdi et al.

This paper [LK07] presents and explains the architecture of Evalvid-RA, and shows simulation results proving it works as anticipated. To make the toolset even more attractive, an interface to TFRC congestion control was implemented in addition to the P-AQM interface. The paper presents results for both of them.

The main contribution if this paper is to show the novel simulation architecture for quantization scale rate adaptive video system, its performance, and also show some simulation cases using this toolset. These cases show e.g. that interfacing a VBR rate controller to TFRC is a challenge: the TFRC sender buffering delay might become significant because of conflicting behavior of the variable VBR rate and the smooth sending rate of TFRC. P-AQM on the other side transmits packets without any sender buffer, and thus eliminates any such delay. It is shown that the aggregate of VBR flows submitted to the network without any sender buffer creates traffic characteristics almost free of any LRD. Thus the paper proves that Lemma 1 of this thesis also holds for rate adaptive media. This is the main reason why it is possible to achieve low packet delay and loss at high link utilization for rate adaptive media flows. Results are given as PSNR and MOS scores. The final results show also that the VBR rate control gives statistical multiplexing gain when used with P-AQM.

5.4.6 Paper F

Title: P-AQM: low-delay max-min fairness streaming of scalable real-time CBR and VBR media

Author: Arne Lie

Published in Proc. of IASTED EuroIMSA, Innsbruck, 17–19 March 2008.

The last paper [Lie08] has the following contributions: (i) it gives the control theoretical stability criterion for the complete outer loop feedback chain (Paper B and C did only provide stability analysis of the “inner” local loop), (ii) introduces the “ERF” signaling

model, which have several advantages to ECF, and (iii) gives a thorough explanation of why P-AQM provides global max-min fairness, proved by GFC-2 [Sim94] network simulations. The stability analysis is based on fluid flow approximation and the usage of classical Bode and Nyquist stability criteria. Extensive ns-2 simulations with rate adaptive CBR and VBR media flows are included, and mixed with TCP flows where found appropriate. The end-to-end performance as a function of RTT, number of flows, and link capacity is shown. The P-AQM max-min tests in GFC-2 network scenario are compared to corresponding TFRC tests, showing that the added network intelligence gives significant improvement in end-to-end delay control and bandwidth sharing.

All VBR media flows are created and controlled with the Evalvid-RA toolset (in this paper Evalvid-RA is used only to create correct traffic characteristics, it is not used to calculate received PSNR of the media flows as in Paper E). The paper is thus the final contribution, showing the suitability of P-AQM of proving a practical architecture for fair bandwidth sharing in a common best effort class for both elastic and real-time constrained flows. The link utilization is generally close to 100%, and packet drops occur only at epochs having significant change in the number of active flows. The claim of decreased network delay at higher link bandwidths, as envisioned in Chapter 5.1.1, was verified to hold by simulation experiments.

6

Concluding remarks and future work

I never think of the future — it comes soon enough.

Albert Einstein — US (German-born) physicist (1879–1955)

6.1 The P-AQM solution — discussion and open issues

6.1.1 P-AQM benefits

The included papers in this thesis have shown that the P-AQM added network intelligence gives improved control of rate adaptation, network latency, and bandwidth fair sharing, compared to pure end-to-end regimes like TFRC. However, even TFRC is shown to benefit from the minor added network intelligence of ECN (Paper E). The steady aggregated media rates of CBR and/or VBR rate controlled media shows no or limited LRD (Paper E), and thus high link utilization with low router queue backlog is possible. The five requirements set forth in Chapter 4.4.2 is fulfilled.

As link capacities increase in the future, these delays will continue to decrease, even at full link utilization. However, this might not be the case for legacy FIFO routers, since they are often dimensioned with at least 200ms or so queuing capacity (BDP dimensioned, assuming 200ms average RTT of its flows), regardless of capacity [VS94]. Others have recently argued for a significant reduction in the buffer sizing [AKM04, AR06]. Others again is pointing out that networks using small buffers could obtain high link utilization, but not without the cost of high loss rates, and thus a high degree of retransmissions of TCP traffic [DJD05]. Thus, the design rules are still a bit ambiguous, and the incentive of using dedicated decoupled smaller buffers for real-time traffic, as in the proposed P-AQM architecture, should still hold.

It must be noted that, unlike e.g. the ATM ABR approaches [LMR97] and SAVE [SR01], the P-AQM system does not support an initial rate request from the sources. This is deliberately. E.g., if a 100 Mbit/s link is shared by 100 VBR and 100 TCP long lived flows, and

all flows are bottlenecked by this link, the P-AQM will grant all VBR and TCP flows with 500 kbit/s each (i.e. the TCP queue is granted 50 Mbit/s output capacity, and the TCP congestion control mechanism will then try to utilize this capacity at its best). The VBR flows can however, unlike long-lived TCP flows, be *self-limited*, meaning that their maximum output bit rate is below their fair network share. If half of the VBR flows are high-quality 100 kbit/s VoIP flows, and the rest is high-quality 2 Mbit/s video flows, the VoIP will be granted its maximum value (100 kbit/s) while the video flows will get 900 kbit/s each (50 x 100 kbit/s + 50 x 900 kbit/s = 50 Mbit/s). This is in agreement with the best-effort Internet principles, where all flows have equal right for bandwidth resources. If the video flows in this example require less than 900 kbit/s on average, the TCP flows would achieve more than its fair share of 50 Mbit/s as in this example (the Round Robin scheduler will transfer excess bandwidth to the other queue if the queue destined for transmission is emptied). To conclude, there is no need for the media flows to require a specific bit rate: the network will either tell the source after session start-up that its fair share is above its own upper requirement, which is fine, or that it is below, forcing the media sources to apply rate adaptation mechanisms.

6.1.2 Future tests

In the final two papers of this thesis (Paper E and F), the P-AQM design was tested with a rate controller using the video quantizer scale as measure for rate adaptation. This was not accidental: the design of P-AQM, and also RL-QoS, was based on the assumption of a possible fine grained rate control. If FGS rate control of pre-stored video should ever become commercial available, even finer rate steps would be possible. What is not tested in this thesis is P-AQMs performance with multirate encoding and traditionally MPEG layered encoding. These solutions will provide much coarser rate control steps, making it much more difficult to obtain high link utilization without experience significant packet loss. The local source rate controllers of such regimes should be much more conservative and only step up their rate if having a margin to the feedback rate seen over a significant time period, e.g. several seconds. It is envisioned that such a “hysteresis” should enable a robust control of multirate and layered rate adaptive sources within the P-AQM regime.

6.1.3 Dead ends

The lack of VBR traffic correlation at large time scales ended the initial ideas of enhancing the P-AQM further with *forecasting capabilities*, e.g. using Kalman filtering. It was also discovered that Gao et al. [GHH02] already had proposed a predictive AQM in 2002 utilizing elastic TCP traffic self-similarities.

It was envisioned in Paper A that feedback signaling once per 5 ms was practical, as this would also enable very fast RL-QoS network reaction responses. Later, on the P-AQM

work, once per 40 ms as lower limit was suggested (Paper B–D), though it was realized that rate change should happen at time scales equal or above the RTT of the flows. However, when introducing real video traffic in Paper E and F, it became clear that the rate update frequency should also be correlated to the GOP sizes of the media traffic. More on this in the following section.

6.1.4 Open issues and implementation limitations

Another less explored issue is P-AQM's robustness to varying video GOP sizes. It is a known fact that long GOP sizes increases the coding efficiency at the cost of increasing packet loss vulnerability. The general rule is therefore to use long GOP sizes only when transmitting media over reliable channels. Even though the P-AQM design can ensure low packet drop ratios, error prone wireless links on the path can be a challenge. Also, during the work on Paper E and F it became clear that the network feedback period should at least be as high as the average GOP size of the video sources contributing to the traffic aggregate. If this was not the case, the P-AQM queue could start to oscillate. This vulnerability was more evident in small aggregates than in larger aggregates. Within the open Internet, it would be out of the question to make end user *requirements* constraining the maximum GOP size allowed. Still, one could *recommend* to use GOP sizes for streaming media in the order of half a second to minimize the artifacts due to packet loss, and to stabilize its bandwidth occupancy. Future work should anyway explore the remedies and workarounds, to increase the P-AQM's robustness to GOP size variations. One solution could be to accept short GOP streams directly into the network, while long GOP streams must pass through a transmit traffic shaping buffer. This should create an incentive for interactive media deployment to use short GOPs.

It should also be mentioned that the last P-AQM variant, the P-AQM+ERF published in Paper F, is currently implemented in ns-2 using the same explicit signaling system as P-AQM+ECF. I.e. it uses ICMP Source Quench packets to directly signal the congestion state. However, the very same performance would result from updating this congestion state "in-band" in the packet headers at downstream P-AQM routers. In both systems, it is the lowest resulting bandwidth that will guide the rate control algorithm at the source. The only difference would be that in the ICMP SQ case, the feedback signal will arrive somewhat earlier. Using one ICMP SQ packet per congestion period is however a fragile operation: if that packet is lost, the congestion signal update will have to wait until the next period. In the in-band solution, all packets per feedback period will be tagged with the same congestion level (i.e. the ERF absolute bandwidth), which should make this method more robust. E.g. in a two-way communication session, this feedback signal can be "piggy-backed" in the reverse flow packets. Another limitation in the current implementation of P-AQM+ERF is that the possibility of probing the network with one packet before starting the video flow is not yet utilized. This possibility would be available if

using the “in-band” signaling solution, where the video source waits for the first acknowledge packet from the destination before deciding the initial video rate. In the simulations performed in Paper F, all video flows start with 1.0 Mbit/s initial rate, which causes the congested router to significant packet losses before the first feedback signal is received at the sources. This limitation does not have any effect on the steady state performance, though.

6.1.5 The fair queuing round robin scheduler

In P-AQM, the rate adaptive media flows (as well-behaved UDP flows) are granted shorter delay, but without being granted more bandwidth, compared to the elastic flows. Actually, the bandwidth sharing between the TCP queue and the UDP queue is divided based on an estimate of the number of long-lived flows ratio. This scheduling is in the responsibility of the *fair queuing round robin* scheduler of the P-AQM architecture, see Figure 5.3. This scheduler should weight the outgoing capacity granted per queue based on packet sizes, i.e. counted in bytes and not packets. This work has not yet been finalized, as the current version is still packet oriented. The correct balancing has been created by a work-around that calculates the packet size averages over several seconds. Future work should thus include the implementation of a true byte oriented round robin scheduler.

The estimation of the number of flows is another issue of improvements. There exist some approaches of low-complexity flow number estimation, such as the zombie method of SRED (Stabilized RED) [OLW99]. Actually, this method was tried out in an early P-AQM prototype, but (ironically) found difficult to stabilize. Thus, the method chosen was the more direct one of keeping track of active flow tuples (IP address source and destination, port numbers and protocol type) in a state table, and flagging flows as inactive after some seconds of no activity. The advent of P-AQM+ERF opens the possibility of exchanging the UDP flow counting with its implicit estimation method.

6.2 Deployment issues

Two of the challenges for interactive real-time services over best effort Internet are end-to-end delay control and fair long distance bandwidth occupancy. The longer distance, the more link hops, the more router queue delays, and the longer the RTT. Due to the inherent proportional fairness of TCP and TFRC congestion control, i.e. its RTT dependency, long TCP and TFRC flows generally get smaller bandwidth shares than short flows. Long flows also have greater probability of seeing multiple bottlenecks. The future Internet, with significant increased access and core link capacities, will make over-provisioning more probable, and thus both capacity and delay problems would diminish. However, as links capacities continue to grow, the advent of new capacity hungry applications seems to find their way into the arena, consuming the excess bandwidth capacity being present

only for a limited time. The introduction of AQM supporting router regimes, such as P-AQM, would enable a platform where *graceful degradation of perceived quality* would be supported, avoiding the sudden media application drop-outs commonly experienced today.

Still, there are a couple of questions to be asked and answered before seeing such an intelligent Internet: (i) how should it be deployed, and (ii) how to ensure it is scalable? P-AQM routers could be deployed gradually because they do not change any mechanism already present with legacy FIFO or RED routers. The TCP flows would be treated as in a RED router: congestion is signaled by packet dropping or packet ECN marking. The TCP queue should also carry all other packets except RTP, UDP and other traffic having a media payload, which should be destined the UDP queue. The use of the TOS field could of course assist in this operation. This means that protocols such as DCCP/TFRC, DCCP/TPC-like, and SCTP should be routed through the TCP queue, where it would benefit from ECN tagging if the flows allow it. TFRC flows would in this case experience similar treatment as in legacy FIFO or RED routers. TFRC over P-AQM routers is briefly tested in ns-2, verifying the latter statements (unpublished work). In the P-AQM+ECF regime, the UDP queue should also carry P-AQM's ICMP SQ and echo packets. It might be necessary to bring an incentive to convince streaming media implementors to apply rate adaption. Choking ill-behaving high-rate flows can be one such remedy: see last paragraph in this subchapter in how P-AQM can assist here. Nevertheless, the acceptance of the rate adaptive P-AQM tools could happen gradually.

On the question of scalability, the main concern is the signaling traffic. The P-AQM+ECF regime submits ICMP SQ packets periodically from each P-AQM router. The minimum signaling event frequency is typically twice per second. Even if the ICMP SQ packets are small (64 bytes), they would create some traffic: if sent each 500 ms (which also is the typical time scale of network media GOPs), and the average flow travels through five P-AQM routers, each media flow would generate a feedback signaling traffic of $(64 \times 8 \times 5)/0.5 = 5120$ bit/s. This might seem as a significant number for low-rate VoIP connections, while it will constitute a marginal increase in bandwidth consumption for videoconferencing and VoD applications. The P-AQM+ERF regime with end-to-end signaling only will on the other hand produce lower signaling traffic (and independent of the number of hops), and at the same time reducing the complexity of the P-AQM router itself. The latter is due to that there will be no need for the router to keep track of each flows source address, in order to reach them with the ICMP SQ and echo packets. P-AQM+ERF also avoids the potential problem of Firewalls preventing ICMP traffic.

It could be questioned if P-AQM+ECF and ERF aware media packets should be tagged by the sources for the router to recognize them unambiguously, e.g. using the IP TOS field. Tagging would also assist the routers in identifying non-responsive media traffic. Although ill-behaving sources were not tested in any of the published papers, the topic

was addressed during the thesis period. Actually, the P-AQM source code, available as part of the Evalvid-RA package at `www.item.ntnu.no/~arnelie/Evalvid-RA.htm`, has an unpublished method implemented to find such sources, based on the *strike* method of FRED [LM97]. However, as [Bou06] and others claim, such functionality should perhaps not be a task of AQMs, but rather of Firewalls or other service filtering nodes. Others might disagree, as already discussed in subchapter 5.2.1. In fact, AQM regimes such as RED and legacy FIFO routers are very vulnerable to ill-behaving sources, which can grab all available capacity. AQMs such as CHOKe [PPP00] is special designed to starve ill-behaving sources. In P-AQM, *without* the strike method implemented, a typical few but high-bit-rate UDP based ill-behaving sources could grab all available UDP-queue capacity. However, due to the *round robin scheduler* (see Figure 5.3), the TCP queue will still be granted its fair share of capacity, due to the flow count bandwidth share.

6.3 What about multicasting?

The thesis work has addressed *unicast streaming* only, and two-way interactive communication in particular. Rate control of multicast streaming has been a research topic of its own during the last decade. Of the most promising candidates are *layered multicast* [MJV96, VCR98, Joh99, Joh02], in where layered video is submitted in separate multicast groups, and in which the multicast receivers join only the groups they currently have the resources to successfully receive and consume. If conditions change to the worse, they simply leave the group submitting the highest layer.

P-AQM routers can of course be part of such a layered multicast streaming network. The P-AQM+ECF approach with feedback signaling from each router could however also support the multicast tree directly, if the P-AQM router is serving as the multicast tree root (i.e. the first split of packets). This could e.g. be in a WebTV/IPTV distribution chain, where several co-located adaptive sources feed traffic towards an Internet P-AQM ingress router.

6.4 More error resilience?

Error resilient media decoding is a complete research area of its own [FXZH00, LZYZ05, TABM05, HR03]. Not surprisingly, its capabilities are highly dependent on the way media is encoded and packetized, and the statistics of the packet loss process. In general, the more robust the media is prepared for packet loss, the lower is the resulting coding efficiency. This means that an error free channel efficiency will be best when the media is encoded *without* error resilience tools, while an error prone channel efficiency will be best when the media is encoded *with* error resilience tools. The H.264/AVC is the current state-of-the-art video codec in coding efficiency, but also in available error resilient tools. In addition, the new SVC (H.264/AVC Annex G) added to this standard makes it a compel-

ling video standard candidate for both WLAN and 3G and other types of wireless channels: decoding is robust both in situations where packets are lost due to bad wireless signal quality, and also due to packet loss as a consequence of varying capacity. As a consequence, SVC combined with the AVC error resilient tools can deliver high perceived quality also in the presence of high percentage of packet loss due to congestion in *wired networks:* it is robust to sudden severe packet loss, and can then adapt to available capacity and avoid future packet losses. Using the somewhat complex flexible macro block ordering, packet loss up to 30% is reported (AVC) to give PSNR values above 30dB, a 8 dB loss compared to 0% loss, which again had only 1dB penalty compared to encoding without these error resilient tools [TABM05]. Appendix B gives the ideas behind "2D Interlaced" video which may gain similar error resilience with a completely different low-complexity algorithm, but presumably with lower coding efficiency.

To conclude, in the future, the media flows can be transported with significant packet loss without loosing too much fidelity. Correspondingly, AQM routers operating point can accept some more packet loss at the gain of somewhat more aggressive link utilization. Nevertheless, the need for rate adaptation will be the best option for graceful degradation and minimization of the performance loss.

6.5 Future Internet

This thesis has developed the new router design P-AQM to support low-latency and bandwidth fair media rate control within the best effort Internet paradigm. Its performance has been compared to the pure end-to-end designed TFRC that runs over legacy FIFO routers. As the results show in Part II, P-AQM has clear performance benefits. This comes at the cost of added router and signaling complexity. Throughout the work, however, this added complexity has been sought held at a minimum level, to meet scalability requirements. Many have argued that the success of the Internet has been *simplicity* and *scalability.* At the same time, many also argue that the successful Internet architecture must soon be modified to meet future services requirements. "FIRE" — Future Internet Research and Experimentation expert group, writes in 2007 at
`http://cordis.europa.eu/fp7/ict/fire/`:

> *"Many networking researchers around the world have identified emerging limitations of the current Internet architecture and agree that it is time for research to take a long term view and to reconsider the basic architecture of the Internet, to see if any improvement can be identified, even if it does not appear to be backward-compatible at a first glance."*

In fact, EU Frame Program Seven (FP7) is challenging researchers in 2007 to come up with "new communication and networking paradigms". In light of this focus shift, a thesis addressing added network intelligence does not seem as unorthodox as it might have been

perceived only a few years back. It is in the author's hopes and wishes that at least parts of the architectures and tools invented during this thesis work could support, at least motivate, the ongoing development process, whatever the future multimedia tolerant Internet and service dedicated networks eventually will look like.

7

Bibliography Part I

[ABG+01] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow. RSVP-TE: Extensions to RSVP for LSP Tunnels. Technical report, IETF RFC3209, Dec 2001.

[AD05] Mariam Kimiaei Asadi and Jean-Claude Dufourd. Resource Conversion in MPEG-21 DIA. In *Proc. of Joint Conference on Information Sciences (JCIS)*, Utah, USA, July 2005.

[AKM04] G. Appenzeller, I. Keslassy, and N. McKeown. Sizing Router Buffers. In *Proc. of ACM Sigcomm*, Oct 2004.

[ALLY01] S. Athuraliya, V. H. Li, S. H. Low, and Q. Yin. REM: Active Queue Management. *IEEE Network*, 15(3), May-June 2001.

[And97] R. Andreassen. *Traffic performance studies of MPEG variable bitrate video over ATM*. PhD thesis, Norwegian University of Science and Technology, 1997.

[App03] Apple. Reliable UDP. http://developer.apple.com/documentation/QuickTime/ QTSS/Concepts/chapter_2_section_13.html, August 2003.

[App05] Apple. QuickTime 7 H.264—Stunning video quality from 3G to HD. http://www.apple.com/quicktime/technologies/h264/, April 2005.

[AR06] J. Auge and J. Roberts. Buffer sizing for elastic traffic. In *Next generation Internet Design and Engineering, NGI'06*, April 2006.

[ATS07] N. Ahmed, C. Theobalt, and H.P. Seidel. Spatio-temporal Reflectance Sharing for Relightable 3D Video. In *MIRAGE07*, pages 47–58, 2007.

[BENB06] Horia V. Balan, Lars Eggert, Saverio Niccolini, and Marcus Brunner. An Experimental Evaluation of Voice Qualit over the Datagram Congestion Control protocol. Technical report, NEC Europe, Germany, 2006.

[Ber98a] L. Berger. RSVP over ATM Implementation Guidelines. Technical report, IETF RFC2379, August 1998.

[Ber98b] L. Berger. RSVP over ATM Implementation Requirements. Technical report, IETF RFC2380, August 1998.

[BG92] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, 1992.

[BK99] T. Bova and T. Krivoruchka. Reliable UDP Protocol. Technical report, IETF Internet-Draft, February 1999.

[BL89] Tim Berners-Lee. Information Management: A Proposal. http://www.w3.org/History/1989/proposal.html, March 1989.

[BLFF96] T. Berners-Lee, R. Fielding, and H. Frystyk. Hypertext Transfer Protocol – HTTP/1.0. IETF RFC 1945, 1996.

[BOP94] L. Brakmo, S. O'Malley, and L. Peterson. TCP Vegas: New techniques for congestion detection and avoidance. In *Proceedings of the SIGCOMM '94 Symposium*, pages 24–35, Aug 1994.

[Bou06] Jean-Yves Le Boudec. Rate adaptation, Congestion Control and Fairness: A Tutorial. citeseer.ist.psu.edu/boudec00rate.html, 2006.

[Bra99] J. Brailean. Wireless multimedia utilizing MPEG-4 error resilient tools . In *Wireless Communications and Networking Conference*, pages 104–108, 1999.

[BRHG04] Arkaitz Bitorika, Mathieu Robin, Meriel Huggard, and Ciaran Mc Goldrick. A Comparative Study of Active Queue Management Schemes. citeseer.ist.psu.edu/bitorika04comparative.html, 2004.

[BSP01] S. Brix, T. Sporer, and J. Plogsties. CARROUSO — An european approach to 3D-audio. In *AES 110th Convention*, Amsterdam, May 2001.

[BSTW95] J. Beran, R. Sherman, M.S. Taqqu, and W. Willinger. Long-range dependence in variable-bit-rate video traffic. *IEEE Transactions on Communications*, 43(234):1566–1579, Feb/Mar/Apr 1995.

[CBB+98] E. Crawley, L. Berger, S. Berson, F. Baker, M. Borden, and J. Krawczyk. A Framework for Integrated Services and RSVP over ATM. Technical report, IETF RFC2382, August 1998.

[CC00a] J. Chung and M. Claypool. Dynamic-CBT — Better Performing Active Queue Management for Multimedia Networking. In *Proc. of NOSSDAV*, Chapel Hill, USA, June 2000.

[CC00b] Jae Chung and Mark Claypool. Dynamic-CBT and ChIPS router support for improved multimedia performance on the Internet. In *MULTIMEDIA '00: Proceedings of the eighth ACM international conference on Multimedia*, pages 239–248, New York, NY, USA, 2000. ACM Press.

[CDS74] V. Cerf, Y. Dalal, and C. Sunshine. Specification of internet transmission control program. IETF RFC 675, 1974.

[CG04] C. Chafe and M. Gurevich. Network Time Delay and Ensemble Accuracy: Effects of Latency, Asymmetry. In *Proc. of the 117th AES Conference*, San Francisco, CA, USA, Oct 2004.

[Cin72] E. Cinlar. *Stochastic Point Processes: Statistical Analysis, Theory, and Applications*, chapter Superposition of point processes, pages 549–606. Wiley Interscience, 1972.

[Cre01] Nicole Cremel. A Little History of the World Wide Web From 1960s to 1995. http://ref.web.cern.ch/ref/CERN/CNL/2001/001/www-history/, April 2001.

[DJD05] A. Dhamdhere, H. Jiang, and C. Dovrolis. Buffer sizing for congested Internet links. In *Proc. of INFOCOM 2005*, volume 2, pages 1072–1083, March 2005.

[DKZSM05] Nandita Dukkipati, Masayoshi Kobayashi, Rui Zhang-Shen, and Nick McKeown. Processor Sharing Flows in the Internet. In *Proc. of Thirteenth International Workshop on Quality of Service (IWQoS)*, Passau, Germany, June 2005.

[DPR02] F. Delcoigne, A. Proutière, and G. Régnié. Modelling integration of streaming and data traffic. In *15th ITC Speicialist Seminar — Internet Traffic Engineering and Traffic Management*, Würzburg, Germany, July 2002.

[Erl09] A. K. Erlang. The Theory of Probabilities and Telephone Conversations. *Nyt Tidsskrift for Matematik B*, 20, 1909.

[Erl17] A. K. Erlang. Solution of some Problems in the Theory of Probabilities of Significance in Automatic Telephone Exchanges. *Elektroteknikeren*, 13, 1917.

[FGM+99] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. IETF RFC 2616, 1999.

[FJ97] S. Floyd and V. Jacobson. Random Early Detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4), August 1997.

[FKP06] S. Floyd, E. Kohler, and J. Padhye. Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC). Technical report, IETF RFC4342, March 2006.

[FKSS02] W. Feng, D. Kandlur, D. Saha, and K. Shin. The Blue Queue Management Algorithms. *IEEE/ACM Transactions on Networking*, 10(4), Aug 2002.

[Flo] S. Floyd. Red parameters. http://www.icir.org/floyd/red.html#parameters.

[Flo03] S. Floyd. HighSpeed TCP for Large Congestion Windows. Technical report, IETF RFC 3649 Experimental, Dec 2003.

[FOBR01] S. Ben Fredj, S. Oueslati-Boulahia, and J.W. Roberts. Measurement-based Admission Control for Elastic Traffic. In *17th International Teletraffic Congress*, Salvador, Brazil, December 2001.

[FXZH00] Liang Fan, Wei Xiaohui, Xiao Zimei, and Liu Hongmei. An error resilient codec for wireless video . In *Proc. of ICCT 2000*, volume 2, pages 1165–1168, Beijing, China, 2000.

[GB98] M. Garrett and M. Borden. Interoperation of controlled-load service and guaranteed service with atm. Technical report, IETF RFC2381, August 1998.

[GB05] S. Goyal and U. Bellur. Mapping application QoS to network configurations for MPLS networks. In *IEEE Consumer Communications and Networking Conference, CCNC*, pages 562–564, Jan 2005.

[GGK+99] P. Goyal, A. Greenberg, C. R. Kalmanek, W. T. Mars-Hall, P. Mishra, D. Y. Nortz, and K. K. Ramakrish-Nan. Integration of Call Signalling and Resource Management for IP Telephony. *IEEE Network*, pages 24–32, May-June 1999.

[GH98] Donald Gross and Carl M. Harris. *Queueing Theory.* Wiley Inter-Science, third edition edition, 1998.

[GHH02] Yuan Gao, Guanghui He, and J.C. Hou. On exploiting traffic predictability in active queue management. In *Proc. of IEEE INFOCOM 2002*, volume 3, pages 1630–1639, 2002.

[GT99] M. Grossglauser and D. N. C. Tse. A Framework for Robust Measurement-Based Admission Control. *IEEE/ACM Transaction on Networking*, 7(3), June 1999.

[GW94] M. Garrett and W. Willinger. Analysis, Modeling and Generation of Self-Similar VBR Video Traffic. In *Proc. of ACM Sigcomm*, London, 1994.

[HBTK01] P. Hurley, J.-Y. Le Boudec, P. Thiran, and M. Kara. ABE: providing a low-delay service within best effort. *IEEE Network*, 15(3), May-June 2001.

[HE02] B. Hechenleitner and K. Entacher. On Shortcomings of the ns-2 Random Number Generator. In *Proc. of CNDS 2002 (Communication Networks and Distributed Systems Modeling and Simulation Conference)*, San Antonio, TX, USA, Jan 2002.

[HFD+04] J. Herre, C. Faller, S. Disch, C. Ertel, J. Hilpert, A. Hoelzer, K. Linzmeier, C. Spenger, and P. Kroon. Spatial Audio Coding: Next-Generation Efficient and Compatible Coding of Multichannel Audio. In *AES 117th Convention, Paper 6186*, San Francisco, October 2004.

[HFPW03] M. Handley, S. Floyd, J. Padhye, and J. Widmer. TCP Friendly Rate Control (TFRC): Protocol Specification. http://www.ietf.org/rfc/rfc3448.txt, January 2003. IETF RFC3448.

[HH04] H.-F. Hsiao and J.-N. Hwang. A max-min fairness congestion control for streaming layered video. In *Proc. of IEEE ICASSP '04*, volume 5, pages 17–21, May 2004.

[HL96] D. P. Heyman and T. V. Lakshman. What Are the Implications of Long-Range Dependence for VBR-Video Traffic Engineering? *IEEE/ACM Trans. on Networking*, 4(3):301–317, June 1996.

[HMTG01] C. V. Hollot, V. Misra, D. Towsley, and W.-B. Gong. On Designing Improved Controllers for AQM Routers Supporting TCP Flows. In *Proc. of IEEE Infocom*, 2001.

[HR03] T. Halbach and T. Ramstad. Multidimensional Adaptive Non- Linear Filters for Concealment of Interlaced Video. In *Proc. Norwegian Signal Processing Symposium (NORSIG)*, Bergen, Norway, 2003.

[HRR97] M. Hamdi, J. W. Roberts, and P. Rolin. Rate control for VBR video coders in broad-band networks. *IEEE Journal on Selected Areas in Communications*, 15(6), August 1997.

[HWZ+99] Y. Thomas Hou, Dapeng Wu, Wenwu Zhu, Hung-Ju Lee, Tihao Chiang, and Ya-Qin Zhang. An End-to-End Architecture for MPEG-4 Video Streaming over the Internet. In *Proc. of IEEE Int. Conference on Image Processing*, Oct 1999.

[Int96] One Way Transmission Time. ITU Recommendation G.114, Feb 1996.

[ISO99] ISO/IEC 14496-2, Information technology – Coding of audio-visual objects – Part 2: Visual, 1999.

[ISS07] R. Iqbal, S. Shirmohammadi, and A. El Saddik. A Framework for MPEG-21 DIA Based Adaptation and Perceptual Encryption of H.264 Video. In *Proc. SPIE/ACM Multimedia Computing and Networking Conference*, San Jose, USA, Jan 2007.

[ITS] VQM Software. http://www.its.bldrdoc.gov/n3/video/vqmsoftware.htm.

[Jac88] V. Jacobson. Congestion Avoidance and Control. In *Proceedings of ACM SIGCOMM'88*, Stanford, USA, Aug 1988.

[JEN+05] Y. Jiang, P.J. Emstad, A. Nevin, V. Nicola, and M. Fidler. Measurement-based admission control for a flow-aware network. *Next Generation Internet Networks, 2005*, pages 318–325, April 2005.

[JFG06] E. Jammeh, M. Fleury, and M. Ghanbari. Non-packet-loss-based rate adaptive video over the Internet . *IEEE Electronics Letters*, 42(8), April 2006.

[JFG07] Emmanuel Jammeh, Martin Fleury, and Mohammed Ghanbari. Delay-based Congestion Avoidance for Video Communication with Fuzzy Logic Control. In *Proc. of Packet Video Workshop'07*, Lausanne, Switzerland, Nov 2007.

[Jia06] Yuming Jiang. A Basic Stochastic Network Calculus. In *Proc. of ACM Sigcomm*, Pisa, Italy, Sept 2006.

[Joh99] M. Johanson. Scalable video conferencing using subband transform coding and layered multicast transmissio. In *Proceedings of ICSPAT'99*, October 1999.

[Joh02] M. Johanson. Delay-based Flow Control for Layered Multicast Applications. In *Proceedings of Proceedings of the 12th Packet Video Workshop 2002*, Pittsburg, USA, April 2002.

[Joh08] Stian Johansen. *Rate-Distortion Optimization for Video Communication in Resource Constrained IP Networks*. PhD thesis, NTNU, Faculty of Information Technology, Mathematics and Electrical Engineering, Trondheim, Norway, March 2008.

[JSD97] Sugih Jamin, Scott Shenker, and Peter B. Danzig. Comparison of measurement-based call admission control algorithms for controlled-load service. In *INFOCOM'97*, pages 973–980, Kobe, Japan, April 1997.

[JWL04] C. Jin, D. X. Wei, and S. H. Low. FAST TCP: Motivation, Architecture, Algorithms, Performance. In *Proc. of IEEE Infocom*, 2004.

[Kan99] K. Kant. On aggregate traffic generation with multifractal properties. *Proc. of IEEE GLOBECOM'99*, 2:1179–1183, 1999.

[Ke04] Chih-Heng Ke. How to evaluate MPEG video transmission using the NS2 simulator. http://hpds.ee.ncku.edu.tw/ smallko/ns2/Evalvid_in_NS2.htm, 2004.

[Kel03] Tom Kelly. Scalable TCP: improving performance in highspeed wide area networks. *ACM SIGCOMM Computer Communication Review*, 33(2):83–91, 2003.

[KHR02] D. Katabi, M. Handley, and C. Rohrs. Congestion Control for High Bandwidth-Delay product Networks. In *Proc. of ACM Sigcomm*, 2002.

[KKVS04] W.Y. Kung, H.-S. Kong, A. Vetro, and H. Sun. Error Resilient Methods for Real-Time MPEG-4 Video Streaming. In *International Symposium on Circuits and Systems (ISCAS)*, pages 745–748, May 2004.

[KL03] K. B. Kim and S. H. Low. Analysis and Design of AQM based on State-Space Models for Stabilizing TCP. In *Proc. of American Control Conference*, volume 1, 4–6 June 2003.

[KM98] M. Krunz and A. Makowski. A source Model for VBR Video Traffic Based on M/G/infty Input Processes. In *Proc. of IEEE Infocom*, 1998.

[KMFB04] T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido. A Nonstationary Poisson View of Internet Traffic. In *Proc. of IEEE Infocom*, Hong Kong, March 2004.

[KMR93] Hemant Kanakia, Partho P. Mishra, and Amy Reibman. An adaptive congestion control scheme for real-time packet video transport. In *SIGCOMM '93: Conference proceedings on Communications architectures, protocols and applications*, pages 20–31, New York, NY, USA, 1993. ACM Press.

[KMT98] F. P. Kelly, A. Maulloo, and D. Tan. Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability. *Journal of the Operational Research Society*, 49:237–252, 1998.

[KRW03] Jirka Klaue, Berthold Rathke, and Adam Wolisz. EvalVid - A Framework for Video Transmission and Quality Evaluation. In *Proc. of the 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, Urbana, Illinois, USA, Sept. 2003.

[KS01] S. Kunniyur and R. Srikant. Analysis and design of an adaptive virtual queue. In *Proc. of ACM Sigcomm*, 2001.

[KSW+04] U. Krämer, G. Schuller, S. Wabnik, J. Klier, and J. Hirschfeld. Ultra Low Delay Audio Coding with Constant Bit Rate. In *AES 117th Convention, Paper 6197*, San Francisco, October 2004.

[KW02] T. Kuang and C. L. Williamson. Measurement study of RealMedia streaming traffic. In *Proc. SPIE Internet Performance and Control of Network Systems III*, volume 4865, pages 68–79, July 2002.

[KWLZ95] E. W. Knightly, D. E. Wrege, J. Liebeherr, and H. Zhang. Fundamental limits and tradeoffs of providing deterministic guarantees to VBR video traffic. In *Proceedings of the 1995 ACM SIGMETRICS*, pages 98–107, Ottawa, Canada, 1995.

[LAG07] Yann Labit, Yassine Ariba, and Frederic Gouaisbaut. On designing Lyapunov-Krasovskii based AQM for routers supporting TCP flows. In *Proc. of 46th IEEE Conference on Decision and Control*, pages 3818–3823, Dec 2007.

[LAR04a] A. Lie, O. M. Aamo, and L. A. Rønningen. On the use of classical control system based AQM for rate adaptive streaming media. In *Proc. of Seventeenth Nordic Teletraffic Seminar*, Fornebu, Norway, August 2004.

[LAR04b] A. Lie, O. M. Aamo, and L. A. Rønningen. Optimization of Active Queue Management based on Proportional Control System. In M. H. Hamza, editor, *Proc. of IASTED CIIT*, pages 69–74, St. Thomas, US Virgin Islands, November 2004.

[LAR05] A. Lie, O. M. Aamo, and L. A. Rønningen. A Performance Comparison Study of DCCP and a Method with non-binary Congestion Metrics for Streaming Media Rate Control. In *Proc. of 19th International Teletraffic Congress (ITC'19)*, Beijing, China, Aug–Sept 2005.

[LAW05] Steven H. Low, Lachlan L. H. Andrew, and Bartek P. Wydrowski. Understanding XCP: Equilibrium and Fairness. In *Proc. of IEEE Infocom*, volume 2, pages 1025–1036, 13-17 March 2005.

[Lie02] Arne Lie. On the Importance of Relations between Film Cues and Spectators Perception, and the Possibilities of Selective Compression in New Multimedia Technologies. http://www.item.ntnu.no/ arnelie/papers/Perception-Essay_Lie2002.pdf, June 2002.

[Lie08] Arne Lie. P-AQM: low-delay max-min fairness streaming of scalable real-time CBR and VBR media. In *accepted for publication at IASTED EuroIMSA'08*, Innsbruck, Austria, March 2008.

[LK07] Arne Lie and Jirka Klaue. Evalvid-RA: Trace Driven Simulation of Rate Adaptive MPEG-4 VBR Video. *ACM/Springer Multimedia Systems Journal*, 2007. In MSJ paper version 2008.

[LM97] D. Lin and R. Morris. Dynamics of Random Early Detection. In *Proc. of Sigcomm*, 1997.

[LMR97] T. V. Lakshman, P. P. Mishra, and K. K. Ramakrishnan. Transporting Compressed Video Over ATM Networks with Explicit Rate Feedback Control. In *Proceedings of the INFOCOM'97*, page 38, Washington, DC, USA, 1997. IEEE Computer Society.

[LOR98] T. Lakshman, A. Ortega, and A. Reibman. VBR Video: Trade-offs and potentials. *Proceedings of the IEEE*, 86(5):952–973, May 1998.

[LSS01] D. Liu, E. I. Sára, and W. Sun. Nested Auto-Regressive Processes for MPEG-Encoded Video Traffic Modeling. *IEEE Trans. on Circuits and Systems for Video Technology*, 11(2), February 2001.

[LWTW93] W. E. Leland, W. Willinger, M. S. Taqqu, and D. V. Wilson. On the Self-Similar Nature of Ethernet Traffic. In *Proc. of ACM Sigcomm*, CA, USA, September 1993.

[LZYZ05] Lin Liu, Sanyuan Zhang, Xiuzi Ye, and Yin Zhang. Error resilience schemes of H.264/AVC for 3G conversational video services . In *The Fifth International Conference on Computer and Information Technology, CIT 2005*, pages 657–661, Sept 2005.

[MBDL99] M. May, J. Bolot, C. Diot, and B. Lyles. Reasons Not to Deploy RED. In *Proc. of 7th. International Workshop on Quality of Service (IWQoS'99)*, pages 260–262, London, June 1999.

[MF01] R. Mahajan and S. Floyd. Controlling High-Bandwidth Flows at the Congested Router. In *IEEE 9th International Conference on Network Protocols*, 2001.

[Mis95] P. P. Mishra. Fair Bandwidth Sharing for Video traffic sources using Distributed Feedback Control. In *Proc. of IEEE GLOBECOM*, Singapore, November 1995.

[MJV96] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-Driven Layered Multicast. In *Proc. of Sigcomm*, pages 117–130, USA, August 1996.

[MSPZW04] Vicente E. Mujica, Dorgham Sisalem, Radu Popescu-Zeletin, and Adam Wolisz. TCP-Friendly Congestion Control over Wireless Networks. In *Proc. of European Wireless 2004*, Barcelona, Spain, Feb 2004.

[OLW99] T. J. Ott, T. V. Lakshman, and L. Wong. SRED: Stabilized RED. In *Proc. of IEEE Infocom*, 1999.

[Paw03] Krzysztof Pawlikowski. Do not trust all simulations studies of telecommunication networks. In *ICOIN'03*, Jeju Island, Korea, Feb 2003.

[PF95] V. Paxson and S. Floyd. Wide-Area Traffic: The Failure of Poisson Modeling. *IEEE/ACM Trans. on Networking*, 3(3):226–244, June 1995.

[PFTK98] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Throughput: A simple model and its empirical validation. In *Proc. of ACM Sigcomm*, Vancouver, October 1998.

[PH90] C. Partridge and R. Hinden. Version 2 of the Reliable Data Protocol (RDP). Technical report, IETF RFC1151, April 1990.

[PJS99] M. Parris, K. Jeffay, and F. Smith. Lightweight active router-queue management for multimedia networking. In *Proc. of SPIE Conf. on Multimedia Computing and Networking*, Jan 1999.

[Pos81] J. Postel. Transmission control protocol. IETF RFC 793, 1981.

[PPP00] R. Pan, B. Prabhakar, and K. Psounis. CHOKe, A Stateless Active Queue Management Scheme for Approximating Fair Bandwidth Allocation. In *Proc. of IEEE Infocom*, 2000.

[PST05] Luca Piccarreta, Augusto Sarti, and Stefano Tubaro. An Efficient Video Rendering System for Real-Time Adaptive Playout Based On Physical Motion Field Estimation. In *Proc. of EUSIPCO'05*, Antalya, Turkey, 2005.

[RCRB99] R. H. Riedi, M. S. Crouse, V. J. Ribeiro, and R. G. Baraniuk. A Multifractal Wavelet Model with Application to Network Traffic. *IEEE Transactions on Information Theory*, 45(4):992–1018, 1999.

[RFB01] K. Ramakrishnan, S. Floyd, and D. Black. The Addition of Explicit Congestion Notification (ECN) to IP. Technical report, IETF RFC3168, September 2001.

[RH92] A. R. Reibman and B. G. Haskell. Constraints on Variable Bit-Rate Video for ATM Networks. *IEEE Trans. on Circuits and Systems for Video Technology*, 2(4):361–372, December 1992.

[RH98] J. Roberts and M. Hamdi. Video Transport in ATM Networks. *Interoperable Communication Networks Journal*, pages 121–143, 1998.

[RHE99] R. Rejaie, M. Handley, and D. Estrin. RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet. In *Proc. of IEEE Infocom*, March 1999.

[RHJ99] D. Raggett, A. L. Hors, and I. Jacobs. HTML 4.01 Specification. W3C, Dec 1999.

[RL02a] L. A. Rønningen and A. Lie. Performance Control of High-Capacity IP Networks for Collaborative Virtual Environments. In *Proc. of IBC 2002 Conference Proceedings*, Amsterdam, Holland, September 2002. IBC.

[RL02b] L. A. Rønningen and A. Lie. Transient Behaviour of an Adaptive Traffic Control Scheme. In *Eunice*, Trondheim, Norway, September 2002.

[RMV96] J. Roberts, U. Mocci, and J. Virtamo. *Broadband Network Traffic — Performance Evaluation and Design of Broadband Multiservice Networks, Final Report of Action COST 242*. Springer, 1996.

[Rø84] Leif Arne Rønningen. Dynamisk Flytregulering. Technical Report STF44 A84194, ELAB, Trondheim, Norway, Sept 1984.

[ROY00] I. Rhee, V. Ozdemir, and Y. Yi. TEAR: TCP Emulation at Receivers – Flow Control for Multimedia Streaming. NCSU Technical Report, April 2000.

[RRQ04] Seungwan Ryu, Christopher Rump, and Chunming Qiao. Advances in Active Queue Management (AQM) Based TCP Congestion Control. *Springer Telecommunication Systems*, 25(3-4):317–351, March 2004.

[RVC01] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. Technical report, IETF RFC3031, Jan 2001.

[SCFJ03] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. Technical report, IETF RFC3550, July 2003.

[Sim94] Robert J. Simcoe. Test Configurations for Fairness and other Tests. Technical Report AF-TM 94-0557, ATM Forum, 1994.

[SK05] Aljoscha Smolic and Peter Kauff. Interactive 3D Video Representation and Coding Technologies. *IEEE Special Issue on Advances in Video Coding and Delivery*, 93(1), Jan 2005.

[SL04] R. N. Shorten and D. J. Leith. H-TCP: TCP for high-speed and long-distance networks. In *Proc. in PFLDnet*, Argonne, 2004.

[SR01] Ming-Ting Sun and Amy R. Reibman, editors. *Compressed Video over Networks*. Marcel Dekker, Inc., 2001.

[SSZ98] I. Stoica, S. Schenker, and H. Zhang. Core stateless fair queuing: Achieving approximately bandwidth allocations in high speed network. In *Proc. of ACM SIGCOMM 1998*, pages 118–130, Sept 1998.

[Sto03] G. Stoll. EBU subjective listening tests on low-bitrate audio codecs. Technical report, EBU Project Group B/AIM (Audio In Multimedia), July 2003.

[SW00] D. Sisalem and A. Wolisz. LDA+ TCP-Friendly Adaptation: A Measurement and Comparison Study. In *Proc. of NOSSDAV*, 2000.

[TABM05] N. Thomos, S. Argyropoulos, N. V. Boulgouris, and M.G.Strintzis. Error resilient transmission of H.264/AVC streams using flexible macroblock ordering. In *2nd European Workshop on the Integration of Knowledge, Semantic, and Digital Media Techniques, EWIMT05*, London, UK, Nov 2005.

[UNI03] UNINETT. Digital Brytningstid Uninett 10år. http://forskningsnett.uninett.no/publikasjoner/ digital.brytningstid/digital.brytningstid.pdf, October 2003.

[UR07] Martina Umlauft and Peter Reichl. Experiences with the ns-2 network simulator - explicitly setting seeds considered harmful. In *Proceedings of the 6th Wireless Telecommunications Symposium (WTS 2007)*, Pomona, CA, USA, April 2007.

[Vat05] Dmitriy Vatolin. MPEG-4 Video Codecs Comparison. Technical report, Graphics & Media Lab Video Group, March 2005.

[Vat06] Dmitriy Vatolin. MSU Subjective Comparison of Modern Video Codecs. Technical report, Graphics & Media Lab Video Group, Jan 2006.

[VCR98] L. Vicisano, J. Crowcroft, and L. Rizzo. TCP-like Congestion Control for Layered Multicast Data Transfer. In *Proc. of IEEE Infocom*, volume 3, March 1998.

[VHS84] D. Velten, R. Hinden, and J. Sax. Reliable Data Protocol. Technical report, IETF RFC908, July 1984.

[VS94] C. Villamizar and C. Song. High Performane TCP in ANSNET. *ACM Computer Communications Review*, Oct 1994.

[VT05] Anthony Vetro and Christian Timmerer. Digital Item Adaptation: Overview of Standardization and Research Activities. *IEEE Trans. on Multimedia*, 7(3):418–426, June 2005.

[WAZ03] Bartek Wydrowski, Lachlan L. H. Andrew, and Moshe Zukerman. MaxNet: A Congestion Control Architecture for Scalable Networks. *IEEE Communications Letters*, 7(10):511–513, Oct 2003.

[WHZ+01] D. Wu, Y. T. Hou, W. Zhu, Y.-Q. Zhang, and J. M. Peha. Streaming Video over the Internet: Approaches and Directions. *IEEE Trans. On Circuits and Systems for Video Technology*, 11(3):282–300, 2001.

[Win05] Stefan Winkler. *Digital Video Quality – Vision Models and Metrics*. John Wiley & Sons, 2005.

[WM02] Michael Welzl and Max Mühlhäuser. Scalable High-Speed Congestion Control with Explicit Traffic Signaling. In *PFHSN 2002 (Seventh International Workshop on Protocols For High-Speed Networks - IFIP TC6 WG6.2 / IEEE Comsoc TC on Gigabit Networking)*, Berlin, Germany, April 2002.

[WP98] W. Willinger and V. Paxson. Where mathematics meets the internet. *Notices of the American Mathematical Society*, 45(8):961–970, 1998.

[WP02] Stephen Wolf and Margaret Pinson. Video quality measurement techniques. Technical Report 02-392, U.S. Department of Commerce, NTIA, June 2002.

[Wro97] J. Wroclawski. The Use of RSVP with IETF Integrated Services. Technical report, IETF RFC2210, September 1997.

[WXBZ03] Shengquan Wang, Dong Xuan, Riccardo Bettati, and Wei Zhao. A Study of Providing Statistical QoS in a Differentiated Sevices Network. In *Proceedings of the Second IEEE International Symposium on Network Computing and Applications (NCA'03)*, 2003.

[WZ02] B. Wydrowski and M. Zukerman. GREEN: an active queue management algorithm for a self managed Internet. In *IEEE International Conference on Communications*, volume 4, pages 2368–2372, 2002.

[XHR04] L. Xu, K. Harfoush, and I. Rhee. Binary Increase Congestion Control for Fast, Long Distance Networks. In *Proc. of IEEE Infocom*, 2004.

[XSSK05] Yong Xia, Lakshminarayanan Subramanian, Ion Stoica, and Shivkumar Kalyanaraman. One More Bit Is Enough. In *Proc. of Sigcomm*, Philadelphia, August 2005. ACM.

[YCA+06] Zhenyu Yang, Yi Cui, Zahid Anwar, Robert Bocchino, Nadir Kiyanclar, Klara Nahrstedt, Roy H. Campbell, and William Yurcik. Real-Time 3D Video Compression for Tele-Immersive Environments. In *Proc. of SPIE/ ACM Multimedia Computing and Networking (MMCN'06)*, San Jose, CA, 2006.

[YdV01] Shanchieh Yang and G. de Veciana. Bandwidth sharing: the role of user impatience. *Proc. of IEEE GLOBECOM'01.*, 4:2258–2262, 2001.

[YYNB06] Zhenyu Yang, Bin Yu, Klara Nahrstedt, and Ruzena Bajcsy. A Multistream Adaptation Framework for Bandwidth Management in 3D Teleimmersion. In *Proc. of NOSSDAV'06*, Newport, Rhode Island, 2006.

[ZqYyM+07] Peng Zhang, Cheng qing Ye, Xue ying Ma, Yan hua Chen, and Xin Li. Using Lyapunov function to design optimal controller for AQM routers. *Journal of Zhejiang University - Science A*, 8(1):113–118, Jan 2007.

[ZWF06] Hongqiang Zhai, Jianfeng Wang, and Yuguang Fang. Providing Statistical QoS Guarantee for Voice Over IP in the IEEE 802.11 Wireless LANs. *IEEE Wireless Communications*, pages 36–43, Feb 2006.

Part II — Included papers

The first half of our lives is ruined by our parents,
and the second half by our children.

Clarence Darrow — US defense lawyer (1857–1938)

Paper A

Performance control of high-capacity IP networks for Collaborative Virtual Environments

Leif Arne Rønningen and Arne Lie

Published in
IBC 2002 Conference Proceedings

12–15 Sept., Amsterdam, 2002

Paper A

Performance control of high-capacity IP networks for Collaborative Virtual Environments

L. A. Rønningen, A. Lie

Norwegian University of Science and Technology (NTNU), Norway

Abstract

The support for low latency high throughput IP networks for multimedia streaming is today very limited. This paper shows that in high-capacity Gbps IP networks, RTP/UDP/IP packets of 1500 bytes/packet will achieve extremely low queuing latency (<10ms) and still utilize the network capacity almost 100%. A router must implement a low-complexity algorithm that monitors the queue length and the rate of change of queue length. By sending this information to neighboring routers and hosts, the traffic can be dynamically controlled to achieve the desired balanced throughput and latency. The only traffic needing QoS controlled channels are these low-rate control messages. The video streams are assumed to include error resilient coding in order to cope with packet losses up to 15% of the stream, alternatively to scale down the output data rate with the same factor.

The low latency of this network makes it a very promising network candidate for Collaborative Virtual Environments such as Distributed Multimedia Plays, where distributed musicians and actors can practice and perform live concerts and theatre, provided total latency do not exceed 10ms.

1. Introduction

Basic IP packet switching networks support the “best effort” philosophy, and do not give any Quality of Service (QoS) guarantee. However, extended with schemes like IntServ or DiffServ (see below) the QoS can be controlled. A network providing QoS guarantee is ATM. ATM uses cell (packet) switching, and virtual circuits. MPLS in turn, with its label switching principle, guarantees QoS by offering traffic classes and resource reservation

(RSVP-TE or LDP), and is more flexible than ATM when interconnecting other networks through the multi-protocol encapsulation principle.

1.1 Quality Of Service Control In IP Networks

Distributed Multimedia Plays (DMP) by Rønningen [1], i.e. Collaborative Virtual Environments for musicians, in some cases requires delays as low as 10 ms for audio and 20 ms for video. During a collaborative session (see Figure 1), the requirements may vary, and an adaptive QoS scheme is needed. We will in this paper focus on controlling the end-to-end delays, and at the same time obtain high resource utilization. Coding schemes shall be adaptive and designed so that artifacts from packet drop/loss are minimized. The spatial and temporal resolution as well as the service profile shall be dynamically scalable, based on traffic measurements in the network.

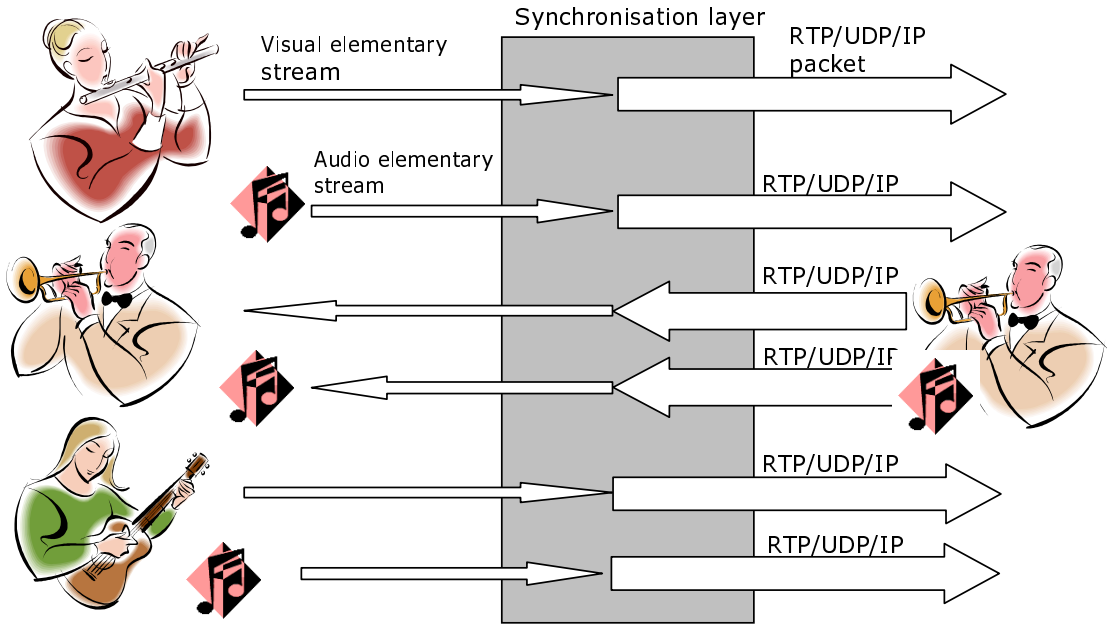

Figure 1 *One host site of Distributed Multimedia Plays showing visual and audio streams. Each elementary stream gets its unique RTP/UDP/IP packet stream. The collaboration between musicians requires a maximum latency of 10ms.*

1.2 IntServ, DiffServ, and traffic shaping

IntServ (Braden et al. [2] and Wroclawski [3]) guarantees the QoS level through per-flow resource reservation (RSVP) and admission control. The drawbacks of IntServ are that it is not very flexible, and that the overhead traffic becomes large in large networks. As an evolution from IntServ, IETF designed the DiffServ (Li et al. [4]) that builds upon priority classes of traffic and decentralized control. Packets entering the network will be marked

according to priority, and routers use this to put the packets into the right outgoing priority queues. Traffic shaping (Rønningen [5]) is a principle used to control traffic streams through controlling the mean value and variance of packet rates. Queuing, dropping and content downscaling are used as control mechanisms.

1.3 How much Traffic Control is needed?

In IP networks, it can be shown that with short packets and high link capacity the utilization can approach 100%, and the absolute delays can be measured in microseconds (see below). This result was recognized by the designers of ATM, where the link capacity is 156 Mbps and the cell length is 53 bytes. The same rationale has been used for satellite TDM multiplexes, where the channel capacity can be 40 Mbps and the transport packets are 188 bytes plus error correction bytes.

The outgoing link and packet queue of an IP router has been simulated. The simulation shows that with increasing link capacity, the utilization of the link can be close to 100% and the delay kept at an acceptable level, which is about 10 ms for Distributed Multimedia Plays [1]. Further, 1Gbps IP networks will only need some sort of admission control and a packet drop mechanism based on the link utilization when it approaches 100%.

2. System description

The system to be studied shall support adaptive multimedia applications like Distributed Multimedia Plays. To reduce the basic data rate generated from a scene, it is assumed that video and sound objects can be tracked and captured in real time. Each object is treated as an entity with its private RTP/UDP/IP traffic stream. There will always be a dependency between streams, but the aim is to reduce this to a minimum. Adaptive compression will be performed within each object, and not between objects. Scene description is according to MPEG-4 (BIFS). The synchronization and multiplexing as defined in MPEG-4 are entirely handled by the RTP/UDP/IP combined header (Kikuchi et al. [6]). Setup and release of services can when needed be handled by the RTSP protocol. A QoS management entity will implement the RTP/UDP/IP protocols, traffic control, transport and routing.

2.1 Proposed Traffic Control And Encoding

The following network characteristics have been chosen: Host link capacity C_h=30Mbps (~HDTV), router link capacity C_r=1Gbps, and IP packet length L=1500 bytes. Thus, one user can never saturate the core network alone.

Three traffic classes are defined:

- TC1 (highest priority): uses reservation, either with IntServ or DiffServ with RSVP. Up to about 10% of the total capacity could be assigned to TC1. The overhead traffic from IntServ would give a small load.
- TC2 (medium priority): This class is for the rate adaptive media flows. The TC2 traffic load triggers feedback "scale-message" (scale-msg), sent to neighbor sources (scale-msg sent in class TC1). The aim is to ask traffic sources to scale down their traffic, and to send "admission denied" to hosts. In addition, when queue lengths exceed a certain number, packets will be dropped. In the host, every other 2, 4, 8, 16,… packets can be dropped. In the routers a large number of streams are merged. It may not be necessary to check *which* packets to drop: just drop the arriving packet when the queue length exceeds a certain value. The following video coding scheme is introduced at the source host: vertical and horizontal "interlace" coding of objects, and interpolation in the receiver when there are lost packets (see section 2.3).
- TC3 (lowest priority): Best effort class, for elastic file transfers, where delay is not critical.

In order to implement the TC2 traffic class, the following algorithm is proposed and must be performed periodic. The prediction of queue-lengths is performed using time series analysis, with exponential smoothing. At interval time points *i* (in the simulations every 1 ms), each router and host do the following:

- measure the queue length x_i
- calculate weighted mean $m = c_i x_i$, (1)
  where c_i are weights, decreasing exponentially backwards in time *i*.
- calculate the growth rate $\Delta = c_i(x_i - x_{i-1})$ (2)
- if the queue length has become too large ($> x_{\max}$), then drop the packets from selected streams according to the rule described above.

After *n* measurements (in simulations *n* =5), routers and hosts

- form the sum $s = \sum c_i x_i + \sum c_i(x_i - x_{i-1})$ (3)
- read incoming scale-msg
- calculate the metric for scale-msg, i.e. the relative number
  $\mathrm{sw}(j) = 1 - \exp[-\{(s(j))/k_1 + (s(j+1))/k_2\}]$ (4)
  where *j* is a router or host and k_1 and k_2 are constants
- if $\mathrm{sw}(j) > k_3$ (to prevent sending scale-msg when there is no need for it), send scale-msg(sw(*j*)) to all neighbor sources of router *j*
- if an edge router: inform hosts if admission is denied for a the next period of length *n* when predicted queue length > $s_{\max}$

2.2 Protocols

For setup and release of channels, the RTSP protocol is applied. For transport of multimedia content like DMP, the RTP/UDP/IP protocols together support the QoS control

parameters needed. The QoS management entity in hosts and routers (hardware) must process the RTP/UDP/IP headers as one header (performance/implementation aspect). This header implements all synchronization, timing and multiplexing similar to that specified by MPEG-4 [6].

The following fields are included in the common header:

- RTP: Version, Payload type, Sequence number, Timestamp, and synchronization source (plus a few more).
- UDP: Source port, Destination port, Length, and Checksum.
- IPv6: Version, Class, Flow label, Payload length, Next header, Hop limit, Source IP address, and Destination IP address.

The Sequence number and the synchronization source number are used to uniquely identify each object stream.

2.3 MPEG-4 Scenes, Objects, and Encoding of Video and Sound

In a fully implemented MPEG-4 Distributed Multimedia Plays, the scenes must have live object texture segmentation (e.g. using blue-screen), each object e.g. being either a musician or a static scene object. The video capture may include 2D interlacing (both in horizontal and vertical directions, in contrast to ordinary even/odd lines), to enable easy interpolation of missing frames in the decoder. A minimum requirement is to add error resilient mechanisms as provided by the MPEG-4 standard to enable graceful degradation of visual and audio streams on packet loss. The benefits of using object segmented scene, rather than conventional rectangular video, is that each stream will be more robust to packet losses, and give a lower total data rate.

Thus, each such object stream (called elementary stream in MPEG-4) will be transmitted as independent RTP/UDP/IP packets (i.e. *not* using the flex-mux tool of MPEG-4 DMIF layer), see Figure 1. In this way, each stream will be almost statistically independent, and the merging of these streams will give a negative exponential distribution of inter-arrival times. This simplifies calculations and simulation modelling.

3. Simulation models and the M/D/1 queue

The theoretical M/D/1 queuing model can be shown to be a good approximation for describing an output link and its queue of a router. This is true when the packet length and the link capacity are constants. The total input stream approaches “Markovian”, that is, the inter-arrival time distribution approaches the negative exponential distribution when a large number of small, independent streams merge.

The simulated network consists of 32 source hosts and three routers each with 16 I/O links, connected to other routers and a number of destination hosts. The 32 source hosts are modelled with two traffic generators. One generator feeds the host queue (named SourceHost1 in Figure 2), and the second feeds the edge router directly, aggregating the total traffic from 31 hosts and a number of routers (RouterGenerator1). Two other traffic generators feed the other two routers, aggregating the traffic from other routers. The three routers are modelled as a series of M/D/1 queues.

In order to use IP packets of 1500 bytes lengths, and to obtain an end-to-end delay of less than 10 ms for a small percentage of the packets, the host links have a capacity of 30Mbps (which is sufficient for HDTV, DV, or 3D MPEG-4 object streams), and the links between the routers in the network have a capacity of 1 Gbps.

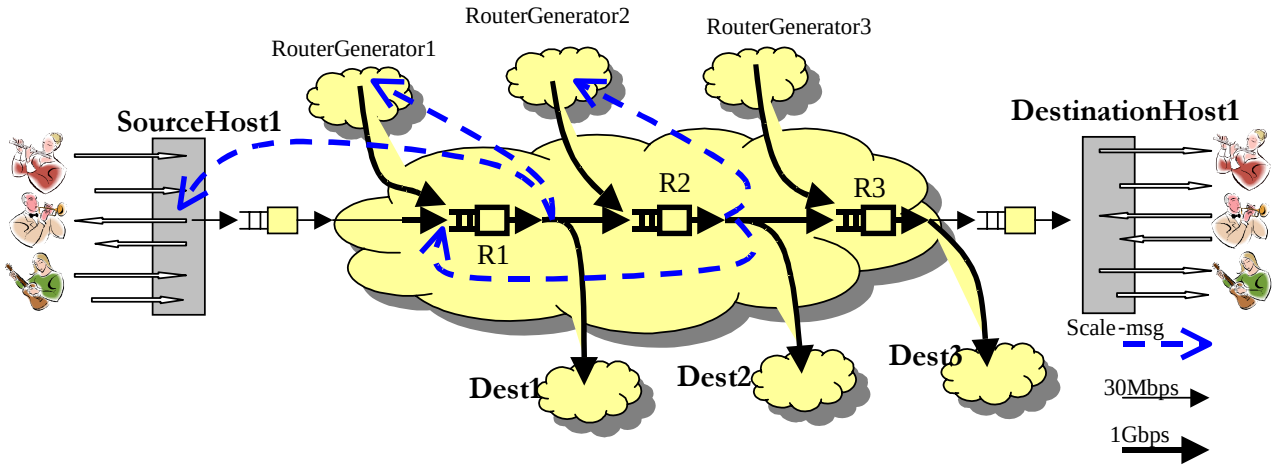

Figure 2 *Traffic model showing the primary traffic and the traffic control messages (scale-msg), the latter showed with dashed lines.*

The SourceHost1 traffic generator generates packets with inter-arrival times drawn from a negative exponential distribution. The traffic generators for R1, R2, and R3 use the same type of generator. All packets from RouterGenerator1 are routed out into the cloud (Dest1) after arrival at R1. The same principle applies for RouterGenerator2 and 3. Packets from SourceHost1 are the only packets that pass through all five queues. Scale-msg is sent from router $R_j, j \in \{1, 2, 3\}$, backwards to routers and hosts that give direct input to router j. It is assumed that an advanced routing algorithm distributes the traffic adaptively on several "best" routes. Since scale-msgs have the highest priority (TC1 traffic class) and are sent periodic, this transfer delay is neglected, and in the simulations global variables are just set to regulate input traffic. An entity Scaler(j) is activated each 1 ms to perform the measurements and prediction calculations (eq. (1) and (2) presented in section 2.1). Each 5 ms a scale-msg(sw(j)) is sent back to the sources as indicated in Figure 2. The traffic generators simulate scaling of both MPEG "profile" and "level" by just terminating packets before they enter any queue, according to the parameters of the scale-msgs.

3.1 Simulation of a Single M/D/1 Queue

With a traditional non-adaptive video/audio encoder it was assumed that about 0.2% of the multimedia content could be missing at the receiver at rendering time without giving noticeable artifacts. This means that 0.2% of the packets can have delays higher than 10 ms. The inter-arrival time of packets to the queue was drawn from a negative exponential distribution (to give the Markovian characteristics). The mean inter-arrival time was adjusted to give the link utilization numbers as shown in Table 1.

Table 1 *Simulated delay and utilization vs. link capacity for a single M/D/1 queue. Packet length = 1500 bytes*

Link capacity C_r	**10 Mbps**	**30 Mbps**	**1 Gbps**	**1 Gbps**	**30 Gbps**
Link utilization	64.8%	87.0%	98.98%	99.98%	99.986%
Mean delay	2.31 ms	1.76 ms	0.52 ms	5.52 ms	0.462 ms
Delay > 10ms	0.223%	0.236%	0.00%	0.186%	0.00%

When C_r = 1 Mbps, all packets are delayed at least 12 ms (transmission latency alone), and this is not usable for DMP audio. With C_r =10 Mbps, about 0.2% of all packets have delays larger than 10 ms at a link utilization of 65%. With C_r = 30 Mbps, about 0.2% of all packets have delays larger than 10 ms at a link utilization of 87%. With C_r = 1 Gbps, about 0.2% of packets have delays larger than 10 ms, at a link utilization of 99.98%! With C = 30 Gbps, the maximum delay was 800 microseconds at a link utilization of 99.986%.

The simulation shows that with increasing link capacity, the utilization of the link can be close to 100% and the delay kept at an acceptable level.

3.2 Simulation of the Network in Figure 2 with Traffic Controls

The traffic control signals as proposed in section 2.1 were implemented in hosts and routers. The SourceHost1 queue rejects new packet arrivals when there are eight packets in queue. The corresponding figure for the routers is 200. (This figures were selected after some experimentation.) The inter-arrival times of packets into host and routers were set high enough to overload the network, i.e. 200 microseconds for the host and 8 microseconds for the routers. It is assumed that the sources are scalable, i.e. that the visual and/or audio bit rates can be adjusted by altering compression efficiency and/or sampling rate. However, the traffic controls simulated behaved nicely, and even if the link utilization for routers became 100% and for hosts 92%, all packets showed a delay of less than 10 ms (see Figure 3).

The traffic stream into R3 consists of 32 sub-streams merged together randomly. E.g., from stream number 20, only 0.3% of the packets was dropped on arrival at destination. For SourceHost1, the total number of packets generated during simulation were 19804, and 2470 were dropped in host queue 1, which is 12.5%. This means that the source should scale down its rate by 12.5% in order to avoid packet losses before entering the core network. 3.75% of the packets transferred from the source host to the destination host used more than 8 ms. All packets in the simulation used less than 10 ms.

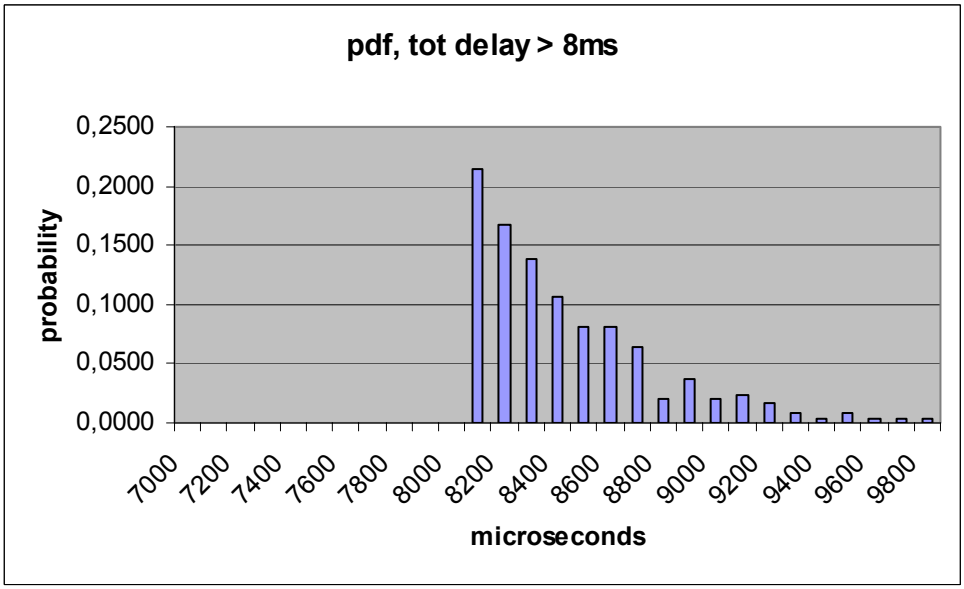

Figure 3 *Simulated probability density function of the packet delay above 8ms of the total system*

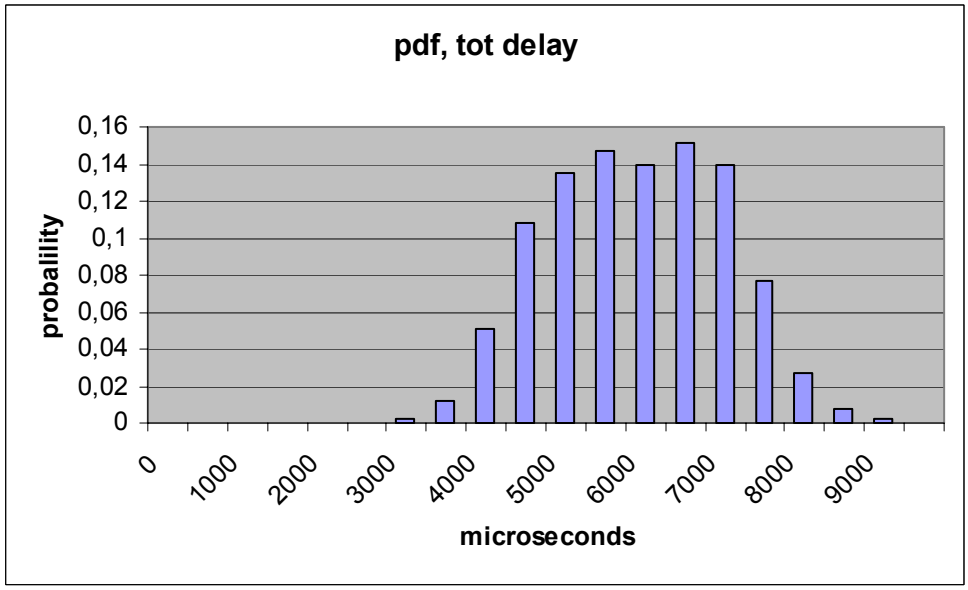

Figure 4 *Simulated probability density function of the packet delay of total system, from input to SourceHost1 to output of DestinationHost1.*

4. Conclusions

The proposed traffic control and network architecture have been modelled and simulated, and performs as expected. The most important results are:

- With the controls and capacities as specified, the link channel utilization between routers can approach 100%, and the time delay can be guaranteed (here less than 10 ms).
- In the source host, 12.5% of the packets had to be dropped due to exceeded maximum queue length. In this case it is absolutely necessary to code in a way that guarantees graceful degradation. Adaptive coding, 2D interlace, error resilience, or other schemes should be investigated. If 2D interlacing is being used, packets are dropped every other, every fourths, every eighths, and so on, to support interpolation in the receiver.
- Regarding routers and dropping of packets, it might be satisfactory just to drop packets when the queue lengths become too large. However, we assume that intelligent sorting and dropping might be necessary, and then routers have to check the whole RTP/UDP/IP header.

References

[1] L. A. Rønningen, February 1999. The Combined Digital Satellite Broadcast and Internet System. Telenor Satellite Services.

[2] R. Braden, D. Clark, S. Shenker, June 1994. Integrated Services in the Internet Architecture: an Overview. IETF RFC1633.

[3] J. Wroclawski, September 1997. The Use of RSVP with IETF Integrated Services. IETF RFC2210.

[4] T. Li, Y. Rekhter, October 1998. A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE). IETF RFC2430.

[5] L. A. Rønningen, 1983. Analysis of a Traffic Shaping Scheme. The 10th International Teletraffic Congress, Montreal 1983 (part of the authors Ph.D. thesis from 1981).

[6] Y. Kikuchi, T. Nomura, S. Fukunaga, Y. Matsui, H. Kimata, November 2000. RTP Payload Format for MPEG-4 Audio/Visual Streams. Status: PROPOSED STANDARD. IETF RFC3016.

Paper B

On the use of classical control system based AQM for rate adaptive streaming media

Arne Lie, Ole Morten Aamo, Leif Arne Rønningen

Published in
17th Nordic Teletraffic Seminar, ISBN 82-423-0595-1

Fornebu Norway, August 2004

Paper B

On the use of classical control system based AQM for rate adaptive streaming media

Arne Lie*, Ole Morten Aamo‡, Leif Arne Rønningen*

*Dept. of Telematics — NTNU,
‡Dept. of Engineering Cybernetics — NTNU

Abstract

This paper presents an Active Queue Management (AQM) design based on a Proportional control system, which is shown to provide good conditions for rate adaptive real-time applications. Simulation results show that rate adaptive media sources over RTP/UDP achieve aggregate core link utilization in the region 97.5–99.9%, with controlled delay and stochastic packet drop in the region 0.1–1.3% per node. In addition, individually streaming media traffic is balanced fairly and stable in the congested network.

1. Introduction

1.1 Related work on optimizing TCP and UDP throughput

Active Queue Management (AQM) aims to optimize the network throughput of TCP connections by either random packet drop or packet marking, based on a congestion cost function. AQM is processed in routers and gateways. RED [1] is the best known AQM where incoming packets are randomly dropped with a probability based on a cost function of average queue length. ECN [2], now an IETF approved standard, is a technique where packets are marked by a RED enabled routers, instead of dropped. The destination node signals the ECN marking back to the source which acts accordingly. ECN gains higher TCP throughput than RED with dropping mainly because the number of retransmissions are lowered. Due to the direct coupling of traffic load and queuing delay in RED, both packet drop or marking probability *and* packet delay will increase on increasing traffic

load. Also, the response time of RED on traffic load transients is quite slow and sluggish [3]. Therefore, in the recent years there has been research into how to design AQM algorithms that provide a stable equilibrium steady state operating point, which in practice means controlled delay and high link utilization. It is shown that AQM fulfilling these criteria can be designed based on classical control system approaches [3]–[5]. These proposals suggest modifications to RED enabled routers, or are completely new approaches.

The Internet traffic has been dominated by elastic applications using TCP, but real-time streaming media over RTP/UDP has had large growth over the last years and is expected to continue growing [6]. AQM should therefore be designed also with these flows in mind. What is even more important, with the streaming media occupying more and more Internet best effort resources, it will cause TCP traffic to bandwidth starvation [9], if not rate adaption is deployed.

Audiovisual traffic has been known for *not* being rate adaptive: the content is encoded (either live or off-line) into a specific quality giving one average bit rate (being either CBR—constant bit rate, or VBR—variable bit rate). It is argued that media applications should be rate adaptive for network resource utilization being fair [7][9]. The MPEG standards have had bit rate scalability as available functionality since MPEG-1, but this feature is implemented in limited scale by the vendors, due to efficiency issues. Recent research focuses now on Wavelet based transform video which provides much better scalability qualities than the commonly used DCT based compression techniques. Interlaced techniques can also be used both for scalability and error resilience [10][13]. Other research focuses on how to jointly optimize media compression with congestion control and packet loss [8][9][11][12][13].

The main difference between elastic applications such as Web browsing, e-mailing and ftp downloads, and real-time streaming media, is that the former needs a transparent channel, while the latter needs a channel with controlled latency. Another important difference is bandwidth stability requirements. Based on this knowledge one can argue that elastic and real-time applications could share congested best effort capacity fairly *if one finds a balanced way of doing it*. In fact, ABE [14] is a queuing scheduling proposal that trades loss for latency in giving fairness to links carrying both TCP and UDP traffic. Another approach is to use measurement based admission control to gain high performance by limiting the number of simultaneously allowed flows [15]. Ultimately, if leaving the best effort Internet, we have the IntServ and DiffServ QoS mechanisms for network resource reservation and prioritization, respectively. The latter two QoS techniques have been available as IETF standards for a decade with limited market deployment. One reason for this is that most users and applications tend to choose free best-effort services without having to care about costly QoS mechanisms, and providers hesitate in deploying complex SLA solutions.

There has been much recent focus on forcing UDP media sources to behave TCP-friendly in a best-effort packet switched network. Sender-driven adaption proposals include:

- TFRCP [18], which achieves very good TCP-friendliness, but suffer from low link utilization. It uses acknowledgement of every UDP packet, and equation-based rate control [17].
- RAP [23] uses UDP packet acknowledgement and AIMD for rate control.
- LDA+ [20] uses RTP with RTCP feedback, and AIMD (additive increase, multiplicative decrease) for rate control. LDA+ achieves fairness comparable with TFRCP and RAP.
- FEC based Multiple Description Coding [8][9], which transforms layered encoding into multiple streams of packets with equal importance, while using an AIMD rate adjustment policy.

Receiver-driven adaption includes RLM [25] and RLC [26], which use layered encoding and multiple multicast groups so clients can join the number of groups that fits its network and terminal capacity [30]. Finally we have transcoding-based solution such as codec filters [22], and perceived quality based adjustment policy such as QOAS (Quality Oriented Adaptation Scheme) [21].

1.2 Identified research topic — rate adaptive media using AQM

To the best of our knowledge, the AQM proposals have so far focused on TCP throughput. Rate adaptive streaming media research has not been considered taking advantage of AQM. The benefit potential of coupling AQM and media rate adaption is controlled packet delay and loss, high link utilization, and fast and fair rate adaption to traffic congestion events. In this paper we present a method where rate adaptive media sources streamed over best effort Internet scale their bit rates based on metrics from our proposed AQM enabled network routers. We assume that the rate adaption uses spatial, temporal, and/or quality scalability in order to reach the desired bit rates. Chapter 2 gives the design requirements and implementation, while Chapter 3 investigates the system performance by simulations.

2. System Design

2.1 Systems requirements

Our proposed rate adaptive streaming media service over best effort Internet includes the following features and requirements:

1. AQM enabled network nodes, where statistical packet dropping take place at link congestion.

2. “UDP fairness”: All streaming media flows are treated equally, with no priority of bandwidth greedy sources over low bandwidth sources. E.g., one 200kbit/s, one 300kbit/s and one 500kbit/s source sharing a capacity of 600kbit/s link should be granted 200kbit/s each. If sharing a 800kbit/s link, the first source gets its 200kbit/s, while the latter two are granted 300kbit/s each. Flows are treated equally and independent of number of hops.
3. TCP friendliness. The streaming media UDP flows should share available bandwidth with TCP sessions in a fair way. E.g., a congested link carrying 60% TCP and 40% UDP flows should in the long run carry 60% TCP and 40% UDP resources, respectively.
4. UDP bandwidth stability. Streaming media cannot normally produce stable perceived quality if the media sources are forced to directly copy TCP congestion behavior, i.e. additive increase, multiplicative decrease, with the same parameter values as common TCP implementations (e.g. TCP Reno). This is because the bandwidth oscillation under congestion is far too high. This oscillation should be brought down to much lower levels, while at the same time making sure that TCP friendliness is achieved in the long run.
5. Packet queuing delay should be controlled to low levels for UDP sources. Packet loss can be traded for latency. For TCP packets, priority should be given to low packet loss, e.g. by using ECN marking.
6. The congested link(s) should operate with high utilization. If dominated by UDP flows, the available capacity should be shared fairly and the aggregate bit rate having small oscillation and equal closely to that of the link capacity.

It is clear that it is a tremendous task to balance all these requirements. Nevertheless, this paper aims at showing a solution focusing on feature 1, 2, 4, 5, and 6. When it comes to TCP friendliness, it can be argued as in [9] that in the long run this can be achieved by finding suitable parameter pairs for the increase/decrease rate. It is left to follow-up work to prove this assertion. It is also left to further study how UDP and TCP can be given different treatment so that UDP have priority to low latency while TCP on low packet loss ratio. However, in this paper we assume that methods like ABE queue scheduling can be used for this discrimination [14], or that the AQM is using ECN marking of the TCP packets, and dropping of UDP packets.

2.2 System blocks overview

The proposed system design is based on two main functional blocks:

1. The AQM enabled routers based on a Proportional controller.
2. The rate adaptive media source using *explicit congestion feedback* (ECF) parameters from the AQM nodes.

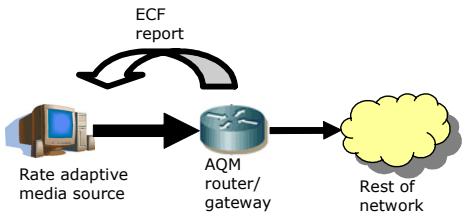

Figure 1 *The communication between AQM enabled router and the rate adaptive media source.*

These two main blocks shall communicate as given by Figure 1. Each source may receive ECF reports from one or more AQM nodes. The AQM deployment can be done gradually. If a non-AQM node is congested, the rate adaption algorithm should be modified by RTCP reports in addition to the ECF reports.

2.2.1 Classical control system AQM

The development of this control system based AQM builds on earlier work where exponential weighting of buffer queue sizes and queue size changes was used as cost function [11][12][13]. It was observed that the previous system did not achieve a stable equilibrium under stationary conditions. This motivated for developing a control system design alternative.

The Proportional controller scaling the input traffic at every node is given by

$$u_l(k+1) = \frac{c_l}{r_l(k)} + \frac{K_l}{r_l(k)}(N_l^* - N_l(k))\max\{0, r_l - c_l\} \quad (1)$$

where l is queue number, k is time index, $u_l(k)$ is the control signal deciding the probability of dropping packet, c_l is output link capacity, $r_l(k)$ is estimated instantaneous input rate (before any stochastic packet dropping) at the end of period k, K_l is the proportional constant, N_l^* is the queue size equilibrium value, and $N_l(k)$ is the sampled queue size each period k. The max-term is added to enable the equation give correct values of excess bandwidth when $r_l < c_l$. In Figure 2 the signal $1 - u^*_l(k)$ is the probability of packet drop, given by

$$u^*_l(k) = \max\{0, \min\{1, u_l(k)\}\} \quad (2)$$

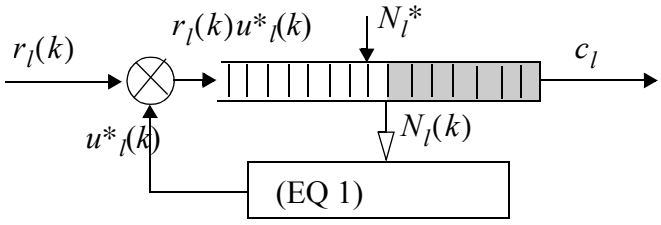

Figure 2 *The AQM based on the Proportional controller of (EQ 1). The incoming flow with rate $r_l(k)$ is exposed to random packet drop with probability $1 - u^*_l(k)$. The packet drop probability is recalculated once every 1ms. Packets not dropped are put into queue (gray cells are occupied cells). If all cells are occupied, tail drop will result.*

i.e. the value is truncated to lie in the region zero to one. This makes the feedback control system non-linear.

The control law (1)–(2) is the discretized result of a simple continuous-time Lyapunov analysis using the dynamic model

$$\frac{dN_l}{dt} = r_l u^*_l - c_l, \tag{3}$$

and assuming $r_l > c_l$. When $r_l < c_l$ the queue is not controllable to the state $N_l = N^*_l$. Consider the Lyapunov function candidate

$$V = \frac{1}{2}(N_l - N^*_l)^2. \tag{4}$$

Its time derivative along solutions of systems (3) is

$$\dot{V} = (N_l - N^*_l)(r_l u^*_l - c_l). \tag{5}$$

Suppose first that u_l is not saturated. Inserting (1) for u^*_l in (5), we get

$$\dot{V} = -K_l(r_l - c_l)(N_l - N^*_l)^2. \tag{6}$$

Next, suppose $u_l > 1$. Then (1) implies $N_l - N^*_l < 0$, so from (5) we get

$$\dot{V} = (N_l - N^*_l)(r_l - c_l) < 0. \tag{7}$$

Finally, suppose $u_l < 0$. Then (1) implies $N_l - N^*_l > 0$, so from (5) we get

$$\dot{V} = -c_l(N_l - N^*_l) < 0. \tag{8}$$

Therefore, we get in general that $\dot{V} < 0$, for all $N_l \neq N^*_l$. It now follows from standard results [27] that $N_l \to N^*_l$, and from (6) we see that the local convergence rate is proportional to the feedback gain K_l.

For input rate $r_l(k)$ estimation, an exponentially weighted filter using only the last 10 samples of number of packets per 1ms period (i.e. filter window is 10ms) is used to represent the instantaneous input rate at every calculation of (1).

Important features of the AQM block are:

- The AQM is using stochastic packet dropping. The probability of *not* dropping packets, u^*_i, is directly a measure of the node traffic load. E.g., if it is 0.9, then 10% of the packets in current period will be dropped on average. The second term in (1) makes sure that the queue size converges towards the equilibrium setting.
- Due to the decoupling of stochastic dropping and queue length, the latency is controlled to an equilibrium setting, regardless of traffic load at congestion.
- When input rate to a specific output link is below capacity, the AQM equilibrium set point is neglected, so that the *u*-value is kept as a direct measure of traffic load.

2.2.2 Explicit congestion feedback (ECF) and rate adaption

Not to be confused with Explicit Congestion Notification (ECN)[2], explicit congestion feedback (ECF) is a novel approach for accurate traffic congestion measures brought back to UDP media sources for rate adaption. The ECF reports are sent periodically (termed "ECF period") directly from the nodes on the path between sender and receiver. (An alter-

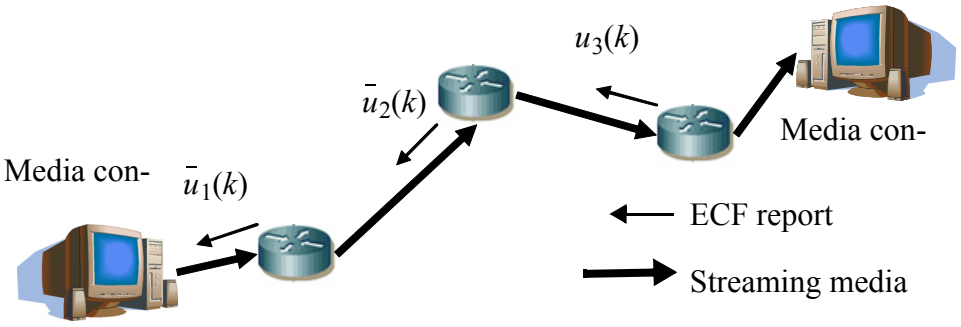

Figure 3 *Once every 40ms an explicit report is send from each router to every source that has send UDP packets to it the last 40ms period. The report, sent in small UDP packets, contains an exponentially averaged u-value, where the recent values have larger influence than the older ones.*

native approach could be to tag IPv6 extension headers with congestion parameters, and use end-to-end signaling such as RTCP.) The nodes need to keep track of which sources have sent UDP packets in the last ECF period. It is an averaged AQM *u*-value that is contained in the small ECF report and sent to the UDP packet sources. In this report, the ECF period is chosen to be 40ms, which is the time lag between frames of video running at 25 frames per second (fps), and typically 1-4 audio frames, depending of the audio compression scheme used. The media source can actually *select* to calculate its new rate each period, or on a lower frequency basis. However, for fast rate adaption, it is recommended to follow the ECF periods, as done here.

In our earlier work where the focus was to optimize throughput of UDP streams in a dedicated multimedia DiffServ class, the rate could be adapted directly to the *u*-value [11][12][13]. Since we in this paper targets best effort networks, UDP and TCP fairness are added requirements. Because of this the rate adaption algorithm is based on the same principles as TCP congestion control: additive increase, multiplicative decrease (AIMD). AIMD has proven to provide fair share of link resources in the long run [9]. The reason is simple: the sources having reached high bandwidth are reducing their bit rates more than the sources having less bandwidth. However, since we also target bandwidth stability, the TCP parameter values are modified to match media traffic. Shortly the rate adaption details will be given.

Before that, we like to stress that the ECF solution presented is in contrast to the ECN enabled UDP stream proposal [3], but also to the classical end-to-end RTP feedback using

RTCP. It is also in contrast to the equation-based fairness algorithms [18] since it does not estimate any RTT: thus intra UDP RTT-unfriendliness is avoided [28]. The use of dropping instead of marking is motivated from the fact that since streaming media using RTP/UDP will not retransmit lost packets, the dropping itself will not create more traffic as in the TCP case. We argue that the solution is scalable since the AQM works on the aggregate traffic belonging to every output link buffer at a router and/or gateway. However, the router/gateway have to keep a record of the UDP sources for each ECF period to be able to send valid reports. The clear benefit of using explicit congestion feedback information is that there is no need for the sources to estimate the congestion as in normal RTCP usage; the congestion is explicitly given by the ECF reports.

Stochastic packet drop adds a variance to the number of packets admitted to the queue every period. To provide ECF report values with less variance to the sources, the 20 last u-value samples in the ECF period are exponentially filtered in the AQM router to provide an estimate of the average u-value representing the second half of the 40ms period, termed $\bar{u}$.

The rate adaption algorithm and parameter settings used in this paper are:

- If received more than one ECF report in a period, select the one with smallest $\bar{u}$-value.
- If there exist an ECF report coming from a closer located node (based on TTL tag), which has an $\bar{u}$-value not more than 25% larger, select that one instead. The reason is to speed up rate adaption fairness when some sources have multiple congested nodes in network, while others have not.
- *Rate adaption algorithm* used in this paper:
  $\bar{u}$-value < 0.99: new_adapted_rate = previous_adapted_rate * $\bar{u}$-value * extra_bp
  $0.99 \leq \bar{u}$-value < 1.02 : no change
  $1.02 \leq \bar{u}$-value < 10.0 : new_adapted_rate = previous_adapted_rate + 16Mbit/s
  $10.0 \leq \bar{u}$-value : new_adapted_rate = original_maximum_rate
- The extra_bp value was in this paper set to 0.95, i.e. 5% extra down-scaling of the bit rate: this was done to speed the process of finding balanced bit rates of the UDP sources.

Note that UDP sources of less than 16Mbit/s bit rates are back to full bit rates after only one up-scaling (the up-scaling is of course limited by the original bit rate). This parameter set has shown through simulations to provide good UDP fairness. It is expected that the parameters should be further optimized in order to also provide sufficient TCP friendliness, which will be verified in follow-up work.

3. Simulations

The simulations carried out were all performed using Simula/Demos event driven simulator [16]. Figure 4 shows the main structure of the network under study. The following

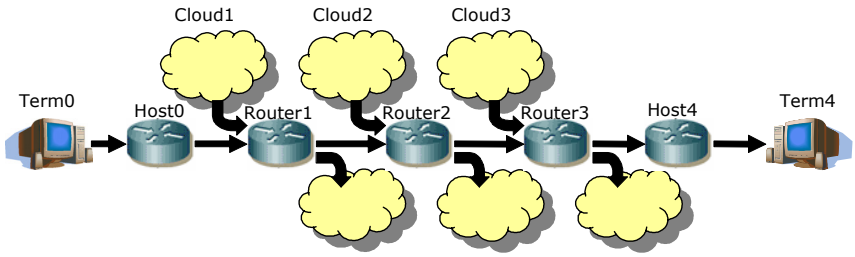

Figure 4 *The sample network used for network simulations. The clouds symbolize aggregated cross traffic sources and sinks. The streaming media flow under investigation is run through Host0 throughout Host4 node. The actual rate adaption is done inside Term0 and in the source clouds.*

two sub-chapters show single-node performance, while the third shows the performance of the sample network.

3.1 Single node performance

The performance of the Proportional controlled AQM is tested with two different type of input: bursty square-shaped input rate, and saw-toothed input rate. The results are shown in Figure 5a) and b), respectively. The square-shaped bursty input rate is changing between zero and 1.5Gbit/s, while the saw-toothed input rate increases linearly from zero to 2Gbit/s, and then drops instantly to zero. The first scenario is tested both with IP packet inter-arrival times being negative exponentially distributed (n.e.d.) and deterministic, while the latter only with deterministic input. The IP packet sizes are fixed to 1500 bytes. The node under investigation has output link capacity of 1Gbit/s. 200 packets are the queue size maximum, and the equilibrium set-point is set to 100 packets. The top-most plots in Figure 5 shows the estimated input rates. The middle and bottom plot shows the u^*-value and queue size, respectively. The a) test results (left part of figure) show that the queue size is hold to an average of 100 packets, which was the desired equilibrium. Although the deterministic input rate achieves more stable results, also the n.e.d. traffic input is controlled satisfactorily. In Figure 5b) we notice that due to some delay in the control loop and the constantly increasing input rate, the queue size is controlled, but is not kept as close to the desired equilibrium as in the square-shaped input rate.

3.2 Single node network performance with ECF

In this scenario the Term0 is sending 1.5Gbit/s towards Host0 router that has an outbound capacity of 1.0Gbit/s. When referring to Figure 4, only these two nodes are used in this test scenario. The ECF reports make Term0 downscale the sending rate so that the target bit rate of 1.0Gbit/s is reached and packet drops are minimized. Figure 6 shows that the target bit rate is reached after 1–3 ECF periods, depending on the synchronization of the start of the burst relative to the fixed ECF periods. The dashed “ECF scaling” line con-

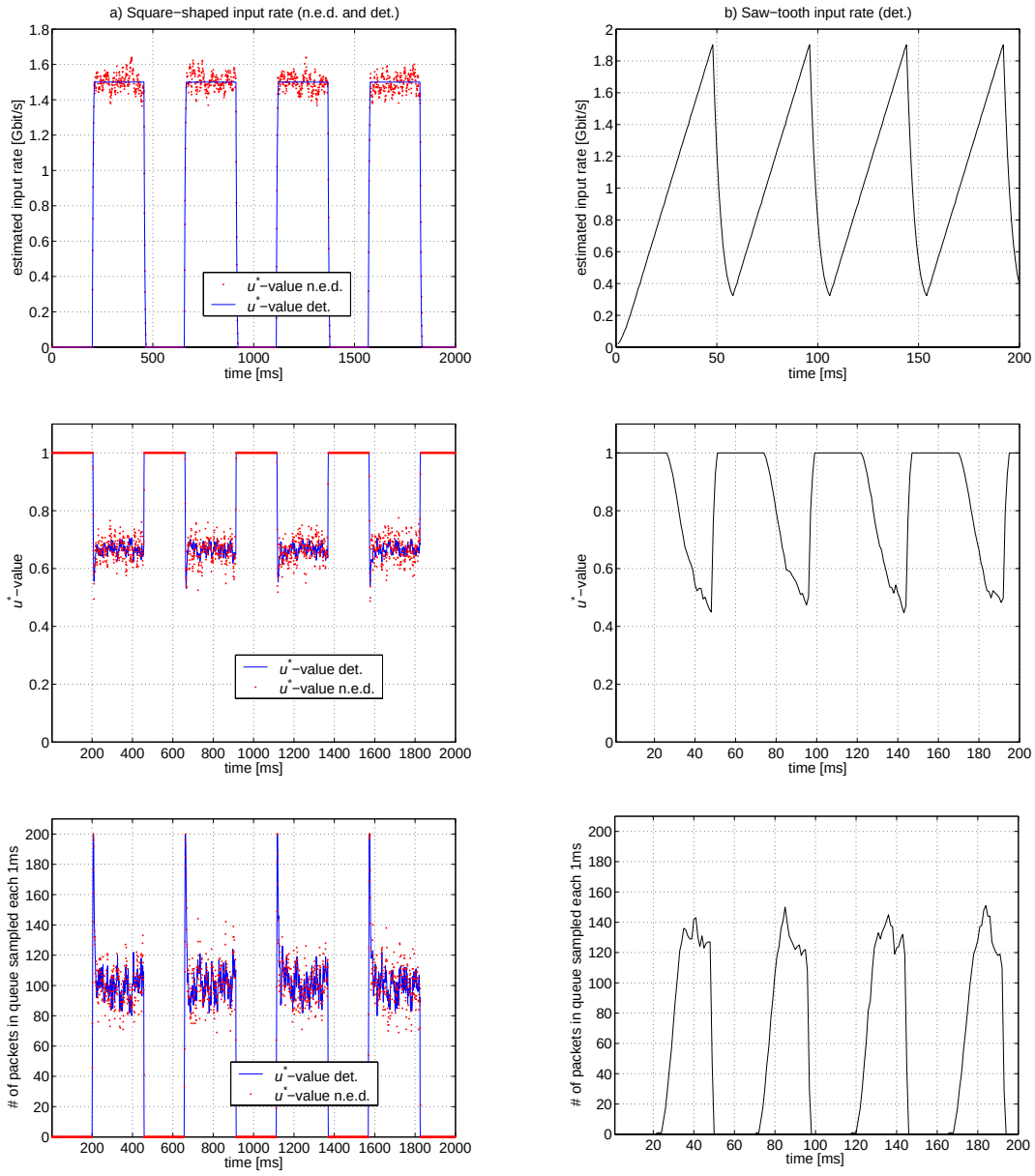

Figure 5 a) *A single 1Gbit/s node experiencing bursty traffic (zero and 1.5Gbit/s input), modelled both as having n.e.d. and constant IP packet inter-arrival time distributions.* b) *Saw-tooth shaped input rate single node behavior (constant input only). Node capacity is 1Gbit/s. Notice that the u-value correctly begins to drop when input rate exceeds the input capacity. Notice also the input rate estimator not quite being able to follow the rapid change in input rate due to the rate estimator filtering.*

verges against 1/1.5=0.667 because the 1.5Gbit/s source shall downscale to 1.0Gbit/s. In Figure 6 the plots of these values are not quite correct, because the scale-value goes back to one in the first ECF-period without any traffic (due to $\bar{u}$-value ≥ 10.0 is fulfilled). Due to the event-driven implementation, it looks like this happens first at burst start, which is incorrect, but the rate adaption is still correct.

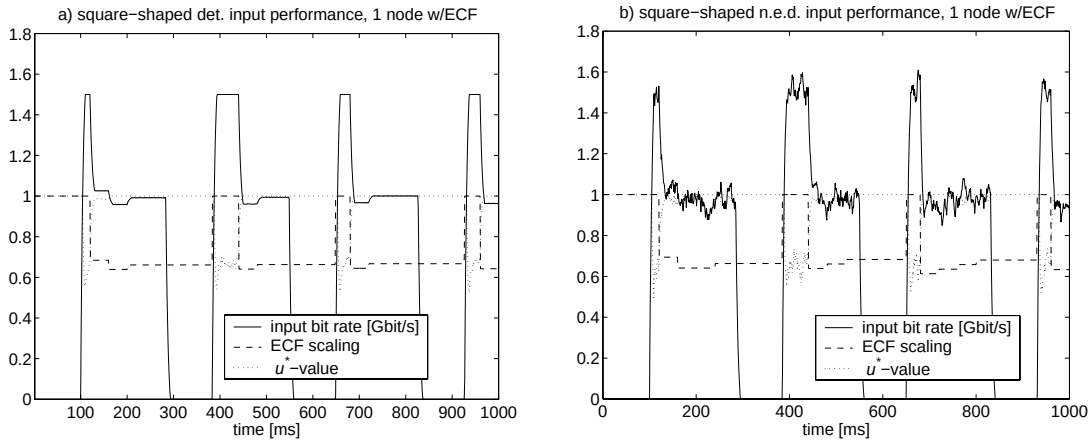

Figure 6 *Single node performance with ECF and square-shaped input.* a) *has deterministic input,* b) *has n.e.d. input. The speed of accurate rate adaption is somewhat dependent of when the burst starts relative to the fixed ECF 40ms periods, but all tests show accurate rate adaption after 2–3 ECF periods.*

3.3 Network performance with ECF

In these tests the full network depicted in Figure 4 is used. In the *first set* of tests Cloud1 to Cloud3 sources produced 1.5Gbit/s input rate of n.e.d. traffic, and all router output links have 1.0Gbit/s capacity. The Term0 source generated 1.5Gbit/s and 750Mbit/s n.e.d. input traffic, with test results shown in Figure 7, a) and b), respectively. In both scenarios the fair share of bandwidth at the congested output links of Router1, Router2, and Router3,

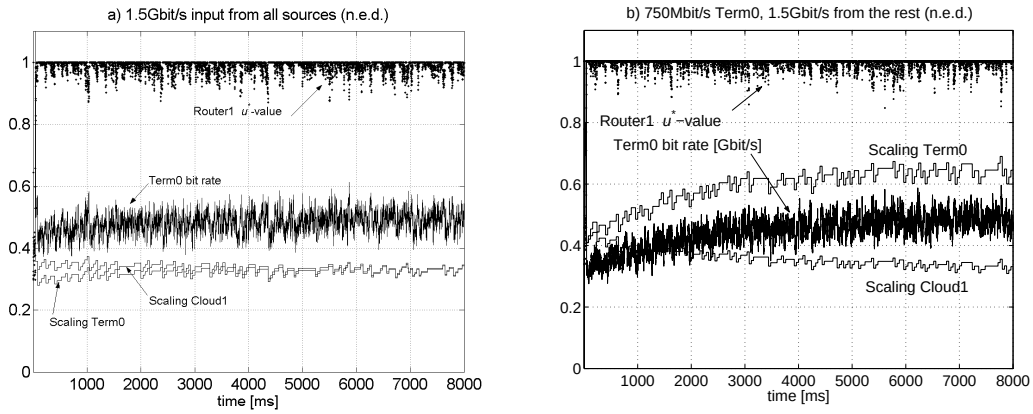

Figure 7 *UDP fairness.* a) *1.5Gbit/s input at Term0.* b) *750Mbit/s input at Term0. Both* a) *and* b) *meets 1.5Gbit/s flow from Cloud1, Cloud2, and Cloud3. After about 4 seconds (somewhat more in* b)*) the Term0 flow is correctly granted half of available bandwidth, i.e. 500Mbit/s, but already after 1–2 seconds it has about 90% of this (450Mbit/s). The Term0 ECF scaling converges to 0.33 in* a)*, and 0.67 in* b)*. The packet loss due to u*-values less than one seems significant, but is in fact no more than ~2.3% in total for the whole path from Term0 to Term4 (see Table 1).*

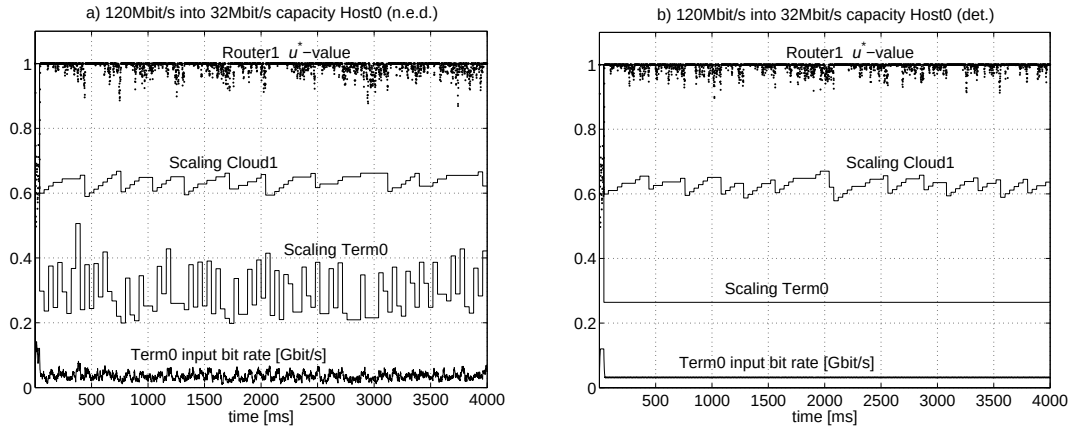

Figure 8 a) *The n.e.d. input traffic makes steady rate adaption very difficult to handle due to very short AQM-controlled buffer in Host0, however it converges towards 32Mbit/s on average, which is correct. In fact, Host0 AQM tries to remove the n.e.d. variance of flow from Term0.* b) *The input rate is steady and therefore the scaling is constant.*

are 500Mbit/s for Term0 source and all of the Cloud sources. We see that this is achieved since the ECF scaling converges to 0.5/1.5=0.33 for the Clouds, and 0.33 for the Term0 in a) and 0.5/0.75=0.67 in b).

In the second test the Term0 bit rate were reduced to 120Mbit/s, and Host0 and Host4 output link capacity reduced to 32Mbit/s. The maximum queue size of these two nodes are reduced to 8 packets, which yields a maximum queue latency of $8 \times 12000 / (32 \cdot 10^6) = 3\text{ms}$. The equilibrium set point is set to 6. Such a short queue operated at 1ms and n.e.d. input traffic is very hard to control. For this test both n.e.d. and deterministic input from Term0 is used, while cross traffic is all n.e.d. Figure 8 shows that the 120Mbit/s input rate is scaled down to 32Mbit/s due to the Host0 capacity, but not more. I.e., the Cloud traffic sources are scaled down to 968Mbit/s to make room for the smaller stream from Term0, following correctly the UDP fairness rules from Chapter 2.1.

When it comes to link utilization vs. packet loss results, they are summarized in Table 1. The scenarios tagged 1 have the same parameters as the scenario in Figure 7 a), but now with three different traffic combinations. The scenarios tagged 2 are the same as in Figure 8, but with four traffic type combinations. We notice that in Scenario 1, the Host0 utilization is about 50%, which is correct due to the equal sharing of the 1Gbit/s core links. In Scenario 2 the Host0 link should ideally have 100%, but due to the short AQM buffer, 8–10% utilization loss is observed at n.e.d. input, but only 1.5% loss at deterministic input. Path packet drop numbers are in fact misleading a bit because much of packet drop occurrences that influence the statistics comes from the first 40ms period were traffic is well over-saturated.

Table 1 *Link utilization [%] and path packet drop [%]*

Scenarios	1a	1b	1c	2a	2b	2c	2d
Type of Traffic (Term0, Cloud*N*)	(n.e.d., n.e.d.)	(n.e.d., det.)	(det., det.)	(n.e.d., n.e.d.)	(det., n.e.d.)	(n.e.d., det.)	(det., det.)
Router1	97.73	98.14	99.99	97.52	97.08	99.61	98.25
Router2	98.17	98.96	98.38	97.27	97.20	99.27	99.67
Router3	98.50	99.38	99.15	97.72	97.53	99.97	99.16
Host0	47.93	49.86	50.79	90.47	98.43	92.39	98.43
Host4	45.58	47.96	49.33	87.39	94.69	91.08	97.25
path packet drop	2.3	1.9	1.5	3.1	3.8	0.4	0.3

Most of the nodes experience working conditions slightly below equilibrium queue sizes. One of the exceptions is Router1 in Scenario 1c, which works around equilibrium throughout the 4s test period. One example of packet delay distribution is shown in Figure 9 which shows that almost no packets are delivered with delay above 6ms.

4. Conclusion

The paper has shown that AQM can be designed based on classical control system principles. The main motivation has been that AQM enabled routers and gateways can control packet delay by statistical packet drop, thus avoiding bursty packet drops at traffic congestion. Further, an averaged value of the generated AQM control signal was fed back to UDP sources for streaming media rate adaption. The continuing growth of streaming media traffic over UDP in best effort networks motivates for rate adaption, else elastic TCP traffic will suffer from bandwidth starvation.

The AQM and rate adaption algorithm performance have been studied through simulations. The results for a 5 node test network having 3 core routers on 1Gbit/s links and 2 edge routers and input traffic having both n.e.d. and deterministic distributed packet inter-

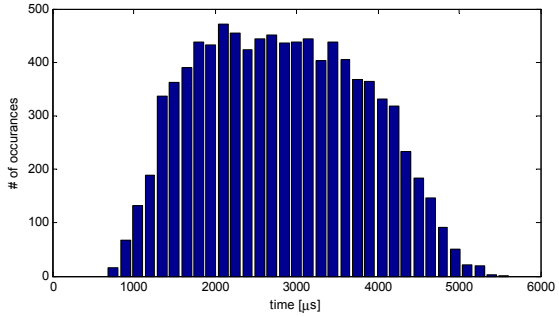

Figure 9 *Histogram showing the packet delay distribution from Term0 to Term4 for Scenario* 2a*. The average delay through RouterX was around 0.25ms, 1.2ms for Host0, and 0.2ms for Host4.*

arrival time, show that the utilization is kept close to 100% while UDP packet loss due to stochastic dropping is kept in order 0.3–3.8% (for the whole path, which is 0.1–1.3% per node) depending on the scenario. Under all circumstances the delay is kept at or below the queue size equilibrium point set by the AQM. The simulation results also show that it exhibit UDP fairness.

Follow-up work will study the algorithms TCP friendliness behavior, possibly by using ns-2 based simulation model, and further optimization of algorithm parameter values.

References

[1] S. Floyd, V. Jacobson, “Random Early Detection gateways for congestion avoidance”, IEEE ACM Transactions on Networking, Vol. 1, No. 4, August 1997.

[2] K. Ramakrishnan, S. Floyd, D. Black, “The Addition of Explicit Congestion Notification (ECN) to IP”, IETF RFC3168, September 2001.

[3] C. V. Hollot, et al., “On Designing Improved Controllers for AQM Routers Supporting TCP Flows”, IEEE Infocom 2001.

[4] K. B. Kim, S. H. Low, “Analysis and Design of AQM based on State-Space Models for Stabilizing TCP”, American Control Conference, 2003. Proceedings of the 2003,Volume: 1, 4-6 June 2003.

[5] S. Athuraliya et al., “REM: Active Queue Management”, IEEE Network, Volume: 15, Issue: 3, May-June 2001.

[6] http://www.broadcastpapers.com/data/VoleraExcelerator01.htm

[7] C. Diot, C. Huitema, and T. Turletti, “Multimedia Application Should Be Adaptive,” HPCS, Aug. 1995.

[8] Kang-Won Lee, Rohit Puri, Tae-Eun Kim, Kannan Ramchandran and Vaduvur Bharghavan, “An Integrated Source Coding and Congestion Control Framework for Video Streaming in the Internet,” Proceedings of INFOCOM 2000, Tel-Aviv, Israel, March 2000.

[9] Rohit Puri, Kang-Won Lee, Kannan Ramchandran and Vaduvur Bharghavan, “Application of FEC based Multiple Description Coding to Internet Video Streaming and Multicast,” Proceedings of the Packet Video 2000 Workshop, Forte Village Resort, Sardinia, Italy, May 2000.

[10] T. Halbach, T. Ramstad, “Multidimensional Adaptive Non-Linear Filters for Concealment of Interlaced Video”, Proceedings Norwegian Signal Processing Symposium (NORSIG), Bergen (Norway), October 2003.

[11] L. A. Rønningen, A. Lie, “Transient Behaviour of an Adaptive Traffic Control Scheme”, EUNICE 2002.

[12] L. A. Rønningen, A. Lie, “Performance Control Of High-Capacity IP Networks For Collaborative Virtual Environments”, IBC 2002 Conference Proceedings, 12–15 September 2002.

[13] A. Lie, L. A. Rønningen, “Distributed Multimedia Plays with QoS guaranties over IP”, IEEE Wedelmusic’03 14–17 Sept., Leeds UK, 2003.

[14] P. Hurley, J.-Y. Le Boudec, P. Thiran, M. Kara, “ABE: providing a low-delay service within best effort”, IEEE Network, Volume: 15, Issue: 3, May-June 2001.

[15] J. W. Roberts, “Internet Traffic, QoS and Pricing”, http://perso.rd.francetelecom.fr/roberts/Publications.html

[16] ftp://ftp.ifi.uio.no/pub/cim/win32/index.html

[17] J. Padhye et al., “Modeling TCP Throughput: A simple model and its empirical validation”, in ACM SIGCOMM ‘98, Vancouver, Oct. 1998.

[18] J. Padhye et al., “A Model Based TCP-Friendly Rate Control Protocol”, in Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV), NJ, June 1999.

[19] M. Miyabayashi et al., “MPEG-TFRCP: Video Transfer with TCP-friendly Rate Control Protocol”, Proceedings of IEEE International Conference on Communications (ICC2001), vol. 1, pp. 137-141, June 2001.

[20] D. Sisalem, A. Wolisz, “LDA+ TCP-Friendly Adaptation: A Measurement and Comparison Study”, Proc. NOSSDAV 2000.

[21] G.-M. Muntean, et al., “Performance Comparison of Local Area Video Streaming Systems”, IEEE Communication Letters, 2003.

[22] D. Wu, Y. T. Hou, W. Zhu, Y.-Q. Zhang, J. M. Peha, “Streaming Video over the Internet: Approaches and Directions”, IEEE Trans. On Circuits and Systems for Video Technology, vol. 11, no. 3, 2001, pp. 282–300.

[23] R. Rejaie, M. Handley, D. Estrin, “RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet”, Proc. of INFOCOM, March 1999.

[24] R. Rejaie, M. Handley, D. Estrin, “Layered Quality Adaptation for Internet Video Streaming”, IEEE Journal on Selected Areas of Communications (JSAC), Special Issue on Internet QoS, 2000.

[25] Steve McCanne, Van Jacobson and Martin Vetterli, “Receiver-Driven Layered Multicast”, Proc. of SIGCOMM, USA, Aug. 1996, pp. 117-130.

[26] L. Vicisano, J. Crowcroft, L. Rizzo, “TCP-like Congestion Control for Layered Multicast Data Transfer”, Proc. INFOCOM, vol. 3, March 1998.

[27] H. K. Khalil, *Nonlinear Systems*, 3rd Edition, Prentice-Hall, 2002.

[28] L. Xu, K. Harfoush, and I. Rhee, “Binary Increase Congestion Control for Fast, Long Distance Networks”, to appear in INFOCOM 2004.

[29] J. Babiarz, K. Chan, “Congestion Notification Process for Real-Time Traffic”, IETF Internet-Draft, Feb. 5, ‘04.

[30] M. Johanson, “Scalable video conferencing using subband transform coding and layered multicast transmission”, Proceedings of ICSPAT’99, October 1999.

Paper C

Optimization of Active Queue Management based on Proportional Control System

Arne Lie, Ole Morten Aamo, Leif Arne Rønningen

Published in
IASTED Communications, Internet, and Information Technology (CIIT),
ISBN 0-88986-445-4

Virgin Islands, Nov. 2004

Paper C

Optimization of Active Queue Management based on proportional Control System

Arne Lie*, Ole Morten Aamo‡, Leif Arne Rønningen*

*Dept. of Telematics — NTNU,
‡Dept. of Engineering Cybernetics — NTNU

Abstract

This paper shows the design and performance of an AQM (Active Queue Management) enabled router. The design is based on a classical proportional control system. The application of such a router is to enhance streaming media over IP performance by avoiding bursty packet drop situations, control the router queue delay, and balancing packet drop, delay, and link utilization. The selected AQM equilibrium point limits the average package delay to the selected value, which is of great importance for real-time applications. It also limits the delay jitter.

The AQM performance was optimized by adjusting the proportional gain, searching for lowest possible packet drop probability when the node was 100% loaded by aggregated IP traffic (modelled as n.e.d. traffic). The results show that at 100% traffic load the packet loss probability can be kept as low as 0.57% and output link utilization at 99.4%. An alternative gain setting with better stability performance at high overload shows only slightly other results at the same load (0.69% drop and 99.3% link utilization).

KEY WORDS

Streaming media, queuing theory, control theory.

1. Introduction

While AQM originally was developed to enhance TCP throughput, its randomization of packet drops is also a positive property when it comes to streaming media performance.

Most Internet routers implement FIFO queues with tail-dropping, i.e. packets arriving on full buffer are discarded. At traffic congestion this will result in many consecutive packets in a flow being discarded. Even if the packet drop probability *on average* is low, the media decoder will not be able to conceal the missing data when too many packets in sequence are missing at decoding time.

The best known AQM implementation RED (Random Early Detection) [1] has however a serious drawback in that its cost function (average queue length) is directly coupled with packet drop probability [2]. This means that the delay will increase and/or have large variations when traffic load is close to or above link capacity. AQM based on classical control systems is motivated from the fact that it creates a decoupled system where a *queue size equilibrium point* can be obtained at traffic congestion [2][3][4]. This means that when the input aggregate traffic load is larger than the output link capacity, random packet drop starts to occur, with the working condition of *keeping the buffer queue filled with a fixed number of packets* on average. Although such AQM defines a good basis for enhancing TCP throughput using ECN [5] marking instead of dropping, our work was initially motivated from the search for streaming media technologies in need for controlled delay over DiffServ enabled IP networks [6][7][8]. It turns out that such AQMs can provide a good basis also for best effort networks fairly sharing both elastic and real-time applications, and where the operating point of each AQM-enabled router can be used to serve efficient media rate adaptation [9]. In addition, its congestion control qualities can support link utilization close to 100% on average, which means return of invested capacity for the network operators.

The work is motivated within the research area of TCP-friendly rate adaptation for best effort Internet [10][11][12], and within robust media content codec development [13][14]. While [9] gave results for the media rate adaptation, this paper focuses on the AQM router itself, and finds optimum gain setting by simulation experiments. The paper focuses especially on the performance when the AQM router is close to being saturated, i.e. the input traffic load ρ is close to one (i.e. 100% traffic load). This sort of knowledge is vital for being able to find suitable rate adaptation algorithms that make the full benefit of this type of AQM. The results are compared to classical M/D/1 analysis.

2. The P-controller AQM design

Streaming media with real-time requirements over UDP is more tolerant to packet drop than to latency: thus it is better to drop the packets than to mark them. Thus the design in this paper employs packet dropping of UDP packets, which is in contrast to the IETF proposal of modifying ECN to also support UDP [15].

The development of this control system based AQM builds on earlier work where exponential weighting of buffer queue sizes and queue size changes was used as cost function

[6][7][8]. It was observed that the previous system did not achieve a stable equilibrium under stationary conditions. This motivated for developing a control system design alternative, first published in [9].

The Proportional controller scaling the input traffic at every node is given by

$$u_l(k+1) = \frac{c_l}{\hat{r}_l(k)} + \frac{K_l}{\hat{r}_l(k)}(N_l^* - N_l(k))\max\{0, \hat{r}_l - c_l\} \tag{1}$$

where l is queue number, k is time index, $u_l(k)$ is the control signal deciding the probability of dropping packet, c_l is output link capacity, $r_l(k)$ is estimated instantaneous input rate (before any stochastic packet dropping) at the end of period k, K_l is the proportional gain, N_l^* is the queue size equilibrium value, and $N_l(k)$ is the sampled queue size each period k. The max-term is added since there is no point in struggling for equilibrium when

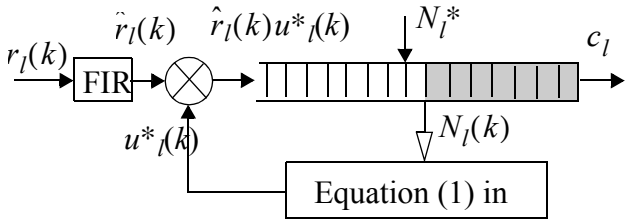

Figure 1 *The AQM based on the Proportional controller of eq. (1) and (2). The incoming flow with rate $r_l(k)$ is exposed to random packet drop with probability $1 - u^*_l(k)$. The packet drop probability is recalculated once every 1ms. Packets not dropped are put into queue (gray cells are occupied cells). If all cells are occupied, tail drop will result. The $\otimes$ object is used in this figure to illustrate multiplication, i.e. scaling, of a continuous flow, while it is implemented in the algorithm as randomized packet drop.*

$r_l < c_l$. The signal $1 - u^*_l(k)$ is the probability of packet drop, given by

$$u^*_l(k) = \max\{0, \min\{1, u_l(k)\}\} \tag{2}$$

i.e. the value is truncated to lie in the region zero to one. This makes the feedback control system non-linear.

The control law (1)–(2) is the discretized result of a simple continuous-time Lyapunov analysis using the dynamic model

$$\frac{dN_l}{dt} = r_l u^*_l - c_l, \tag{3}$$

and assuming $r_l > c_l$. When $r_l < c_l$ the queue is not controllable to the state $N_l = N^*_l$. Consider the Lyapunov function candidate

$$V = \frac{1}{2}(N_l - N^*_l)^2. \tag{4}$$

Its time derivative along solutions of system (3) is

$$\dot{V} = (N_l - N^*_l)(r_l u^*_l - c_l). \tag{5}$$

Suppose first that u_l is not saturated. Inserting (1) for u^*_l in (5), we get

$$\dot{V} = -K_l(r_l - c_l)(N_l - N^*_l)^2 . \quad (6)$$

Next, suppose $u_l > 1$. Then (1) implies $N_l - N^*_l < 0$, so from (5) we get

$$\dot{V} = (N_l - N^*_l)(r_l - c_l) < 0 . \quad (7)$$

Finally, suppose $u_l < 0$. Then (1) implies $N_l - N^*_l > 0$, so from (5) we get

$$\dot{V} = -c_l(N_l - N^*_l) < 0 . \quad (8)$$

Therefore, in general we get that $\dot{V} < 0$, for all $N_l \neq N^*_l$. It now follows from standard results [16] that $N_l \rightarrow N^*_l$, and from (6) we see that the local convergence rate is proportional to the feedback gain K_l. The limitations of this analysis are that it assumes continuous time, and the rate regulation is also continuous. In reality, a finite number shall be selected as the update rate (i.e. the time between k and $k+1$). The discretization of time, together with the proportional gain, will influence the stability concerns. A more thorough study of this relationship is left for future

It was decided to use 1ms period for a 1Gbit/s link capacity c_l. With 1500 bytes per IP packet this gives on average 83 new packet arrivals per periods when traffic load ρ is 1.0. 83 packets can be regarded as sufficient for proper operation of the loop. Still, stochastic packet drop with probability equal to $1 - u^*_l(k)$ is employed, which generates a variance to the number of packets put into the system queue. Optimal setting of the proportional gain K_l is therefore obtained by model simulation.

For input rate $r_l(k)$ estimation, a FIR filter is used over the last 10 measurements of number of packets per 1ms period. The FIR coefficients are exponential weighted, and creates a moving average of the input rate termed $r_l(k)$, which is used in (1).

Important features of the AQM block are:

- The AQM is using randomized packet dropping. The probability of *not* dropping packets, u^*_i, is directly a measure of the node traffic load. E.g., if it is 0.9, then 10% of the packets in current period will be dropped on average.
- The second term in (1) makes sure that the queue size converges towards the equilibrium setting. When input rate is below output link capacity, the AQM equilibrium set point is neglected, so that the *u*-value is kept as a direct measure of traffic load.
- Due to the decoupling of stochastic dropping and queue length, the latency is controlled to a maximum equilibrium setting when traffic load is at or above output link capacity.

What is not that obvious is how well the AQM operates close to its congestion point. This is of great concern since the goal is to reduce the bit rate of the aggregate stream (by rate adaptation both for TCP and UDP) from causing congestion to a level very close to the

link capacity. Also, there will obviously be a trade-off between how aggressively the AQM shall control its queue size, and how bursty the behavior of $u^*_l(k)$ will be. If it becomes too bursty, streaming media traffic will suffer from bursty perceived quality, which would destroy one of the design motives. One therefore needs to find a good balance between fast adaptation and steady behavior, which is to be investigated by model simulation.

3. Simulation results

A 1Gbit/s AQM node based on Figure 1 was implemented in the event driven simulator Demos/Simula [17]. The input traffic characteristics were chosen to have negative exponentially distributed (n.e.d.) packet inter-arrival times, to mimic aggregated MPEG VBR sources. N.e.d. can be considered to be a valid distribution of an aggregate of several tens of ordinary audiovisual (AV) flows, and actually fewer if complex MPEG-4 scenes with multiple AV objects are in use [7]. All packets were of 1500 bytes in size.

The AQM node was injected with this n.e.d. traffic with a series of different traffic loads ρ, ranging from 0.95 to 1.25. The three important resulting parameters was average queue delay, drop probability and output link utilization. In addition, the dynamics of $u^*_l(k)$ had to be inspected for each selected gain setting, to monitor performance stability.

3.1 Queue delay

With n.e.d. input traffic and fixed packet sizes, the AQM node under investigation reduces to a M/D/1 system if the buffer size is infinite and the $u^*_l(k)$ value is forced to 1.0 regardless of traffic load. The average delay in queue Q of a M/D/1 system can be derived from the well-known Pollaczek-Khintchines' formula [18] to give

$$\overline{Q} = \frac{\lambda m^2}{2(1-\lambda m)} = \frac{\rho m}{2(1-\rho)} \tag{9}$$

where λ is input packet rate (packets/s), m is the constant service time per packet

$$m = \frac{1500 \times 8}{10^9} = 12\mu s, \tag{10}$$

and ρ is the traffic load λm. A M/D/1 queue system is lossless and the traffic load is therefore bounded by $\rho < 1$ to ensure the average queue length being finite. Eq. (9) is plotted in Figure 2 together with simulated M/D/1 performance in our model simulation.

More important, the figure shows the simulation results of the AQM enabled node from three n.e.d. input traffic series, with proportional gain K_l equal to 5, 15, and 90 (times 10^{-7}). A period of 8 seconds was simulated. In addition, one series with constant input rate and K_l equal to 5×10^{-7} is shown for comparison. Notice that for $\rho < 1$ the delay converges towards the M/D/1 performance. For $\rho \geq 1$ the AQM works actively to achieve

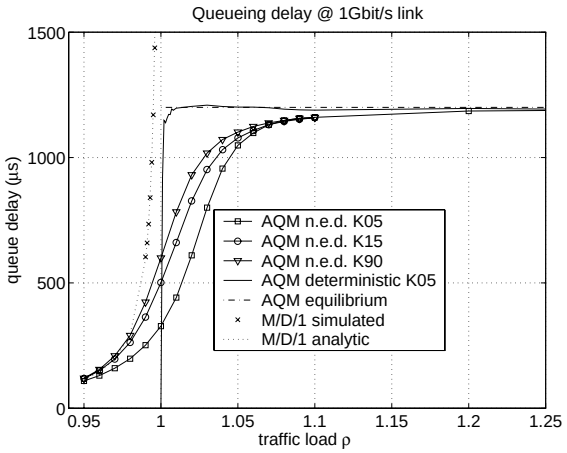

Figure 2 *Queue delay as function of traffic load. Comparison of M/D/1 system and AQM. AQM limits the queue length to a maximum of 1.2ms on average in this example.*

the desired equilibrium setting, which is chosen here as 1.2ms (i.e. 100 packets, with 200 packets maximum buffer size). We see that for increasing traffic load, the average delay converges towards the desired equilibrium.

3.2 Loss and utilization

Due to the variability of n.e.d. traffic rate, the AQM has a positive but small $u^*_l(k)$ also for $\rho < 1$, while its optimal behaviour is to have $u^*_l(k) = 0$ for such traffic loads. We notice that the test with deterministic input rate achieves exactly such performance. It is therefore part of the optimization process to choose a gain factor giving small $u^*_l(k)$ values for ρ close to, but less than, 1. In Figure 3 it is seen that the loss curves is more optimal at higher gain factor values.

Also noticed is that gain factors achieving low loss values also exhibit high utilization values. The explanation is that the less packets that are dropped, the more packets are fed into the buffer, and thus higher link utilization. However, more packets to service means larger queue sizes. This is verified by looking into Figure 2 again, where we notice that simulation series with higher gain factors achieve larger delays.

3.3 Optimal gain

The simulations, as expected, have shown that the delay, loss, and utilization is dependent on the AQM gain factor. Of particular interest is the performance at traffic load $\rho = 1.0$. The reason for this is that the sources shall react to congestion and try to adjust their sending rates so that almost 100% of capacity is used, if possible (other bottlenecks and/or bandwidth capacities can of course prevent this). A new simulation series was therefore run where the traffic load was kept constant, but the gain factor was varied from $K_l = 3 \times 10^{-7}$ to $K_l = 300 \times 10^{-7}$. The results are shown in Figure 4, where it can be

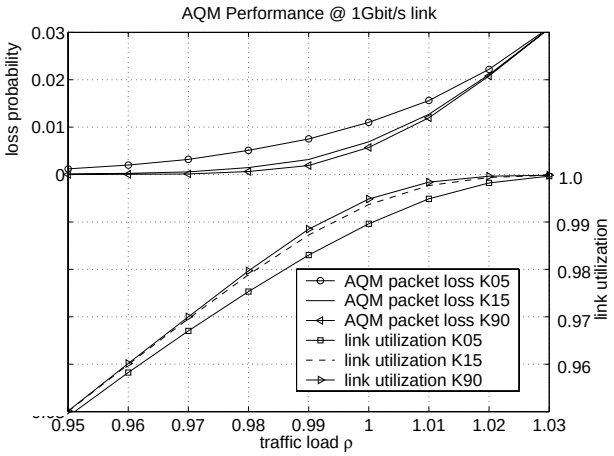

Figure 3 *Packet drop probability and link utilization shown as function of traffic load for three n.e.d. traffic series. The optimal behavior is to have packet drop as close to zero for traffic loads at or below 1.0. The y-axis must be compensated with +1 for valid utility numbers. Optimal utility is equal to 1 (100%) for traffic loads at or above 1.0.*

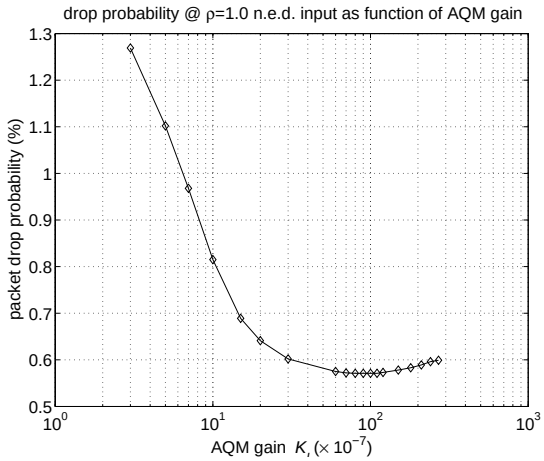

Figure 4 *Optimization of AQM gain factor at traffic load equal link capacity, measured by average drop probability.*

seen that the lowest drop probability (0.57%) is achieved at a gain factor at or close to $K_l = 90 \times 10^{-7}$. However, at $K_l = 15 \times 10^{-7}$ the loss is only 0.12% higher. It remains to be seen if $K_l = 90 \times 10^{-7}$ is too aggressive for the system to provide a steady drop probability. Also interesting is the delay variance at traffic loads close to one. These two topics will now be studied in more detail.

3.4 Burstiness

One major motivation for constructing such an AQM is to use its $u^*_l(k)$ values for streaming media rate adaptation. Although rate adaptation is not the focus of this paper, a short introduction is given here to explain *why* the burstiness of $u^*_l(k)$ is important. Typically, when $u^*_l(k)$ is below 1.0, media sources shall reduce its rate proportional to that factor.

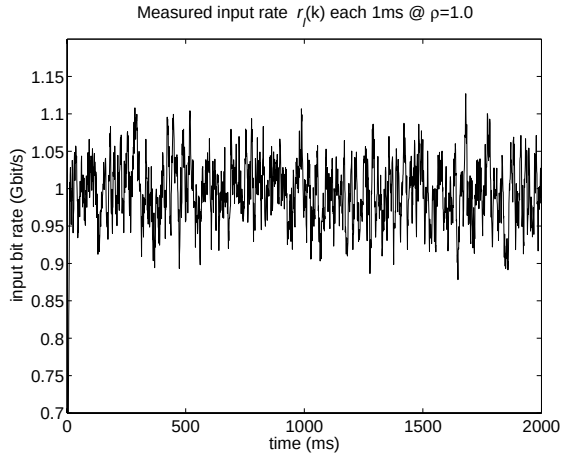

Figure 5 . *The n.e.d. input traffic with average rate equal to link capacity was estimated to these values (sampled each 1ms).*

When $u^*_l(k)$ is one, it should increase its rate at a fixed increment. High link utilization means traffic load at or close to one, i.e. the link closest to saturation should have an operating point with $u^*_l(k)$ close to 1.0. The $u^*_l(k)$ values are signaled back to the media sources either by inband signaling (e.g. using the IP header) or explicit packets. For stable perceived quality under such conditions the variability of $u^*_l(k)$ should however not be too high, because this will force the media sources to a continuous rate adaptation. The following test assumes that the dynamic sources are capable of maintaining a traffic load of 100% all time.

Three gain factor values were selected for further inspection: $K_l = 5 \times 10^{-7}$, $K_l = 15 \times 10^{-7}$, and $K_l = 90 \times 10^{-7}$. Figure 5 shows the variability of the input traffic estimates $r_l(k)$. The random generator seeds are identical in all three simulations, so the node was dealing with exactly the same input sequence in all simulations.

The AQM scaling factor $u^*_l(k)$ and AQM maintained queue sizes are shown in the next two figures. In Figure 6 we notice that packet dropping is present for almost all time for the lowest gain factor, while it happens much more seldom but much more aggressively for the highest gain value. Although the latter gives lowest drop probability on average, these $u^*_l(k)$ values might be difficult to use for a rate adaptation algorithm, since they indicate sort-of highly variable traffic conditions (longer period of time with no packet drop, and short periods of "aggressive" dropping, see bottom plot in Figure 6). In [9] $K_l = 5 \times 10^{-7}$ was used in a larger network to investigate rate adaptation capabilities, and showed satisfactory performance of approximately 1.0% drop per node, which now can be verified by Figure 4 to be close to optimum for this proportional gain setting. Figure 7 shows the three cases again and the maintained queue sizes each 1ms. This figure verifies that higher gain factor achieves higher link utilization because empty buffer occur more seldom. More packets in queue also means higher packet delay on average, as verified by Figure 2.

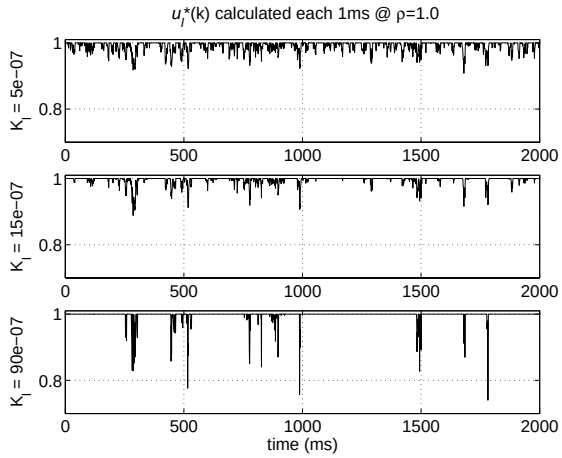

Figure 6 *The AQM scaling factor shows different aggressivity for the three cases. Low gain factor gives a small but always present scaling, while large gain factor gives more seldom but more aggressive scaling.*

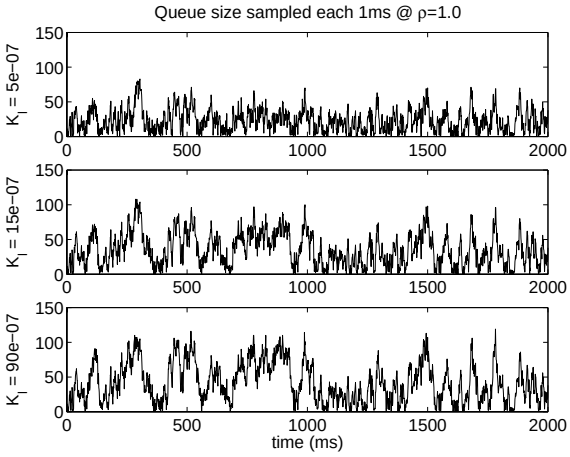

Figure 7 *Sampled queue size each 1ms. Higher gain factor gives better ability to maintain packets in queue. This gives higher link utilization and a bit longer packet delay.*

3.5 Delay variance

Figure 7 indicates that the queuing delay will vary quite substantially for traffic loads close to link capacity. This is a result of the variability of the input traffic (see Figure 5), and the fact that at $\rho = 1.0$ the P-controller has just started getting enough input packets to be able to obtain the equilibrium setting of 100 packets. As we see from Figure 2, when increasing the input rate to e.g. 10% overload (1.1 on the x-axis), the P-controller is capable of holding the mean delay close to equilibrium. It is interesting to note that if the media applications can tolerate a bit higher percentages of packet drop probability than what is obtained at $\rho = 1.0$, then the delay variability might be kept lower, and thus utilization even higher (because buffer is more seldom empty). Figure 8–11 show histograms giving indications of the delay distribution for the three chosen gain factors, and at traffic load

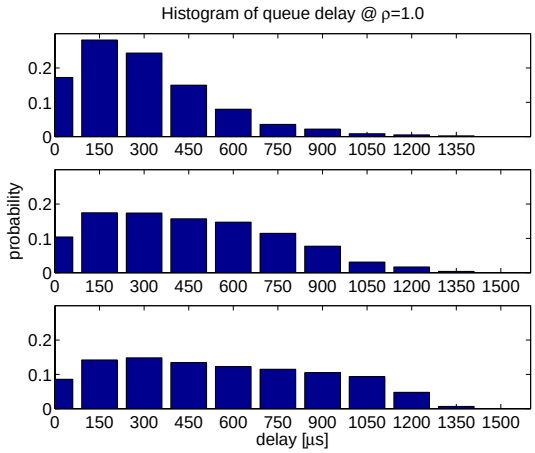

Figure 8 . *Queue delay histogram at traffic load 1.0. K=5 at top, 15 in the middle, and 90 at bottom.*

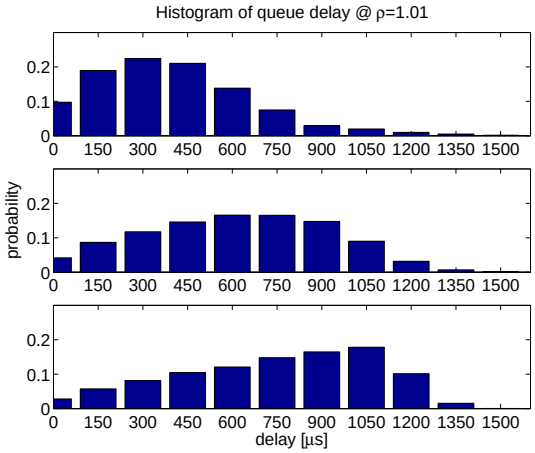

Figure 9 *Queue delay histogram at traffic load 1.01. K=5 at top, 15 in the middle, and 90 at bottom.*

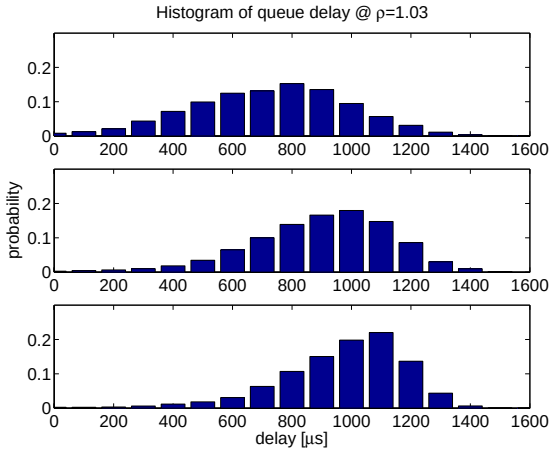

Figure 10 . *Queue delay histogram at traffic load 1.03. K=5 at top, 15 in the middle, and 90 at bottom.*

ρ = 1.0 , 1.01, 1.03, and 1.50. I.e., the two middle cases have 1% and 3% more input traffic than the link is capable of supporting. Inspecting Figure 3 indicates that the losses due

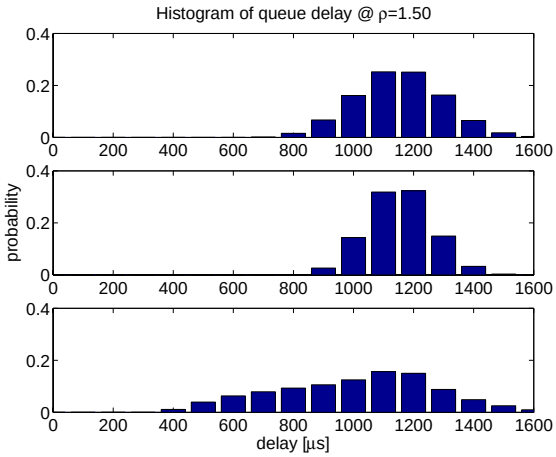

Figure 11 . *Queue delay histogram at traffic load 1.50. K=5 at top, 15 in the middle, and 90 at bottom.*

to this congestion will be only slightly higher than these overload values, most for K=5, and least for K=90.

Figure 10 shows in fact that it is the system with highest gain factor (K=90) that achieves lowest delay variability at 3% traffic overload. At 0% overload (Figure 8) the high-gain system has almost no significant peak in its histogram. For the low-gain system, the situation is opposite: at 0% overload there is a significant peak in the histogram (however far below the wanted equilibrium setting), while the peak gets weaker at 3% overload.

Figure 11 shows the performance at 50% overload, which is just an example of how the AQM reacts at sudden traffic peak changes, before the sources have reacted with their congestion control and rate adaptation algorithms. Surprisingly, at this load the high-gain system seems to break; the feedback loop gain is so high that oscillations might be introduced. Here it is the middle-gain (K=15) system that shows superior performance.

The plots also reveal that a small percentage of the packets are delayed more than the wanted 1.2ms, up to 1.4ms for 3% overload and 1.6ms for 50% overload. This is considered as satisfactory, considering that the delay control is only “guaranteed” stochastically.

4. Conclusions

The paper has shown the design of a P-control system based AQM. Its performance was investigated under varying traffic conditions by model simulation. The proportional gain of the controller was varied to find an optimum setting. $K_1 = 90 \times 10^{-7}$ was shown to provide a minimum of packet drop probability (0.57%) at traffic input load of 1.0, which is the most important traffic load considering the motivation for low loss and high link utilization. However, this gain is so high that oscillating behavior might occur at extreme high input traffic. $K_1 = 15 \times 10^{-7}$ is probably a satisfactory trade-off between performance at input load 1.0 and 1.5 (0.69% drop).

Queuing delay was controlled by the AQM equilibrium setting, which in this paper was set to 1.2ms. Histogram plots show that the stochastic guarantees are well controlled by the AQM, even at 50% traffic overload and gain setting $K_l = 15 \times 10^{-7}$. In practice this means that at 50% overload, 1/3 of the packets will be dropped, but the AQM ensures that the packets that survive will experience almost identical low delay through the node, decided by the setting of the N_l^* equilibrium parameter.

Using such AQMs for streaming media, rate adaptation at the sources can be based on feedback signaling of the $u^*_l(k)$ variable from the AQM nodes. Future work will focus more on stability concerns, as well as optimal rate adaptation and TCP-friendly behavior in a mixed-traffic best effort Internet.

References

[1] S. Floyd, V. Jacobson, Random Early Detection gateways for congestion avoidance, *IEEE ACM Transactions on Networking, 1*(4), August 1997.

[2] S. Athuraliya et al., REM: Active Queue Management, *IEEE Network, 15*, Issue: 3, May-June 2001.

[3] C. V. Hollot, et al., On Designing Improved Controllers for AQM Routers Supporting TCP Flows, *Proc. of IEEE Infocom,* 2001.

[4] K. B. Kim, S. H. Low, Analysis and Design of AQM based on State-Space Models for Stabilizing TCP, *Proc. of the American Control Conference 2003,* Volume: 1, 4-6 June 2003.

[5] K. Ramakrishnan, S. Floyd, D. Black, The Addition of Explicit Congestion Notification (ECN) to IP, *IETF RFC3168,* September 2001.

[6] L. A. Rønningen, A. Lie, Transient Behaviour of an Adaptive Traffic Control Scheme, *Proc. of EUNICE Workshop,* 2002.

[7] L. A. Rønningen, A. Lie, Performance Control Of High-Capacity IP Networks For Collaborative Virtual Environments, *IBC 2002 Conference Proceedings,* Amsterdam, 12–15 September 2002.

[8] A. Lie, L. A. Rønningen, Distributed Multimedia Plays with QoS guaranties over IP, *Proc. of IEEE Wedelmusic'03,* 14–17 Sept., Leeds UK, 2003.

[9] A. Lie, O. M. Aamo, L. A. Rønningen, On the use of classical control system based AQM for rate adaptive streaming media, *Proc. of 17th Nordic Teletraffic Seminar,* Norway, August 2004.

[10] C. Diot, C. Huitema, and T. Turletti, Multimedia Application Should Be Adaptive, *IEEE HPCS Workshop,* Aug. 1995.

[11] J. Padhye et al., A Model Based TCP-Friendly Rate Control Protocol, *Proc. of International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV),* NJ, June 1999.

[12] M. Miyabayashi et al., MPEG-TFRCP: Video Transfer with TCP-friendly Rate Control Protocol, *Proc. of IEEE International Conference on Communications (ICC2001), 1*, June 2001, 137-141.

[13] Rohit Puri, Kang-Won Lee, Kannan Ramchandran and Vaduvur Bharghavan, Application of FEC based Multiple Description Coding to Internet Video Streaming and Multicast, *Proc. of the Packet Video 2000 Workshop,* Forte Village Resort, Sardinia, Italy, May 2000.

[14] T. Halbach, T. Ramstad, Multidimensional Adaptive Non-Linear Filters for Concealment of Interlaced Video, *Proc. of Norwegian Signal Processing Symposium (NORSIG),* Bergen (Norway), October 2003.

[15] J. Babiarz, K. Chan, Congestion Notification Process for Real-Time Traffic, *IETF Internet-Draft,* Feb. 5, 2004.

[16] H. K. Khalil, *Nonlinear Systems*, *3rd Edition,* (Prentice-Hall, 2002).

[17] Demos home page at ftp://ftp.ifi.uio.no/pub/cim/win32/index.html

[18] Donald Gross, Carl M. Harris, *Queueing Theory, Third edition* (Wiley Inter-Science, 1998), pp. 212.

Paper D

A performance comparison study of DCCP and a method with non-binary congestion metrics for streaming media rate control

Arne Lie, Ole Morten Aamo, Leif Arne Rønningen

Published in
Proceedings of the 19th International Teletraffic Congress (ITC'19),
ISBN 7-5635-1141-5, Beijing University Post and Telecommunications Press

Beijing China, 29 August to 2nd September, 2005

Paper D

A performance comparison study of DCCP and a method with non-binary congestion metrics for streaming media rate control

Arne Lie*, Ole Morten Aamo‡, and Leif Arne Rønningen*

*Dept. of Telematics — Norwegian University of Science and Technology (NTNU)
{arne.lie, leifarne}@item.ntnu.no
‡Dept. of Engineering Cybernetics — NTNU
aamo@itk.ntnu.no

Abstract

This paper compares the performance of two algorithms for congestion control of streaming media. The two methods are Datagram Congestion Control Protocol (DCCP) over RED, and a solution based on an Active Queue Management (AQM) combined with explicit feedback of congestion level experienced at routers. DCCP relies on binary congestion metrics, either as packet dropping or ECN marking at AQM routers. In contrast, our proposed solution uses 32 bit congestion level metrics. Transmitted by ICMP Source Quench packets, this enables much faster and accurate response than the binary DCCP. The simulation tool ns-2 is used to compare the two methods transient and stationary behavior, focusing on adaptation speed and accuracy, delay and delay jitter, and fairness. The results reveal that DCCP is inferior in almost all tests, and that the non-binary method proposed in this paper forms a sound network base to provide stable quality and controlled delay for rate adaptive streaming media.

KEY WORDS

Streaming media, rate adaptation, congestion control, Active Queue Management, queuing theory, control theory.

1. Introduction

The study of this paper is motivated towards making packet switched networks more suitable for real-time streaming media using RTP/UDP packets, and is a continuation of previous studies of ours [1]–[4]. The main network challenge in carrying streaming video is to obtain *low router backlog, low packet drop ratio*, and *high link utilization*. Although the ATM research conducted throughout the 1980's and 90's resulted in many advanced traffic management tools ensuring QoS guarantees for video traffic [5,6], they turned out to be very complex and comprehensive. While admission control could be applied to ensure a minimum quality (i.e. throughput) per flow, *congestion control* is an unavoidable tool in order to sustain the before mentioned challenges for the admitted flows, i.e. to balance the aggregate input traffic to the network capacity.

Audiovisual sources employing rate adaptation (RA) is still deployed in limited scale. However, as the amount of non-TCP traffic increases rapidly with the success of VoIP and videoconferencing, the need for RA persists. In order to prevent congestion collapse of the Internet [7], IETF is currently pushing DCCP (Datagram Congestion Control Protocol) [8] to become the new protocol standard for streaming media replacing UDP. DCCP controls the available bandwidth of the application, while the RA itself is still left open to the application providers. For best DCCP performance in congested environment RED [9] or other type of AQM-enabled routers with ECN [10] enabled should be used. TCP-friendliness as well as DCCP fairness is obtained by TCP-like congestion control or TCP friendly rate control equation (TFRC) at the source nodes [11]. However, the binary congestion level feedback (either packet drops or ECN marking) limits the accuracy of the information, and the TCP-friendliness constraint limits the speed of adaptation [12]. In contrast, our proposed method uses a router AQM that calculates an explicit 32 bit metric based on input information rate, queue size, and output capacity. The TCP-friendliness is decoupled from the congestion control by the AQM itself, and hence the RA speed can be fast. This paper compares the performance of our AQM to DCCP, focusing on latency, packet drop, link utilization, adaptation speed and accuracy. The comparison is performed using the *ns*-2 [13] simulation environment, extended with our AQM and an external DCCP module [14].

The paper is organized as follows: Chapter 2 gives the introduction to our AQM design and the congestion control algorithm itself. Chapter 3 gives the results from a set of simulation scenarios, comparing the performance of our AQM design to DCCP. Chapter 4 gives a discussion before the conclusions follows in chapter 5. As an introduction to DCCP see e.g. [8].

2. The AQM design

2.1 Two-queue scheduler and inner and outer loop

The proposed AQM design consists of two main functionalities: (i) the "inner loop", which is essentially a Proportional gain controller, running for matching input rate and equilibrium queue size in steady state, and (ii) the "outer loop" which is the congestion level algorithm and the signaling back to the streaming media sources. The inner loop has

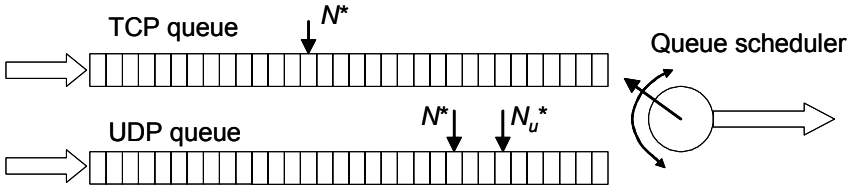

Figure 1 *The two-queue solution of the "inner loop". The queue scheduler provides built-in TCP-friendliness by monitoring the number of active flows.*

been proven to provide good working conditions for both TCP and UDP flows [15] at traffic congestion, as the TCP flow can utilize ECN marking, while UDP packets are dropped. TCP-friendliness is assured by the AQM itself, as it monitors the number of significant active flows, and provides a two-queue scheduler system. The UDP queue inner loop working conditions constitute the basis for the UDP outer loop calculations, while the TCP queue (consisting of all non-UDP packets) runs only a separate inner loop.

2.2 P-AQM — the "inner loop"

The AQM in Figure 2 is a Proportional controller (thus "P-AQM"), run separately at both the TCP and the UDP queue, and scaling the input traffic at node l (i.e. queue l) given by

$$u_l(k+1) = \begin{cases} \dfrac{c_l}{\hat{r}_l(k)} + \dfrac{K_l}{\hat{r}_l(k)}(N_l^* - N_l(k)), & \hat{r}_l(k) \geq c_l \\ \dfrac{c_l}{\hat{r}_l(k)}, & 0 < \hat{r}_l(k) < c_l \end{cases} \tag{1}$$

where l is queue number (just for giving all queues and nodes in the network a unique queue number), k is time index with granularity dT, $u_l(k)$ is the control signal giving the probability of packet survival, c_l is output link capacity for queue l, $\hat{r}_l(k)$ is estimated input rate (before any stochastic packet dropping) at period k, K_l is the proportional gain, N_l^* is the target queue size equilibrium value, and $N_l(k)$ is the sampled queue size each period k. This equation is slightly modified compared to an earlier publication [4] to give more stable and predictable performance. Note that the first term (left both lines) is a rate

matching term (regardless of value of r), while the second term (right term top line) is a buffer matching term, thus making this AQM designed to match both the rate and the queue size. The buffer matching term is only used when the input rate is larger than the output capacity.

In Figure 2, the signal $u^*_l(k)$ is the probability of packet *survival*, given by

$$u^*_l(k) = \max\{0, \min\{1, u_l(k)\}\} \tag{2}$$

i.e. the value is truncated to lie in the region zero to one. This makes the feedback control

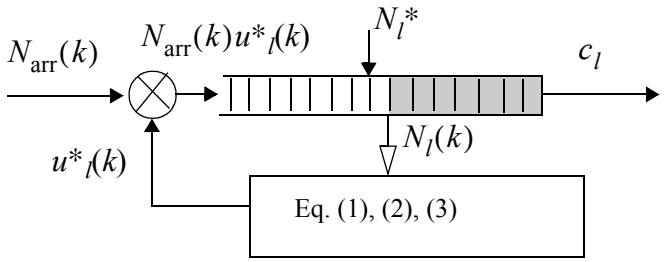

Figure 2 *Illustration of how the inner loop P-AQM works. Each loop period dT it counts the arrival of bytes to the queue,* $N_{arr}(k)$, *and calculates the probability of dropping new arriving packets,* $1 - u^*_l(k)$.

system non-linear. Note that when $r < c$, the u-value is above one, but is truncated to one through (2). The estimation of the input rate $\hat{r}(k)$ is performed through a simple one-tap re-cursive filter

$$\hat{r}(k) = \gamma N_{arr}(k) + (1 - \gamma)\hat{r}(k - 1) \tag{3}$$

where γ is a constant selected between zero and one (the smaller value the lower cut-off frequency of the low-pass filtering), and N_{arr} is number of arriving packets or bytes (depending on the P-AQM is running in byte or packet mode). A stability analysis for the inner loop is performed in an earlier publication [4], which showed that the loop index k granularity in seconds, dT, should be set to equivalently 50–100 packets service time, and K in the region 0.2–0.5.

2.3 ECF (Explicit Congestion Feedback) — the "outer loop"

Rate adaptation (RA) must be carried out at the media sources to avoid further congestion. Max-min fairness [16] is chosen as the fairness criteria between the different streaming media sources. This implies that the RTT (round-trip-time) of the different sessions do not impact the long-term fairness, only the speed in which fairness is obtained. This is in contrast to DCCP that adjusts its speed also by RTT measures, in order to stay TCP friendly.

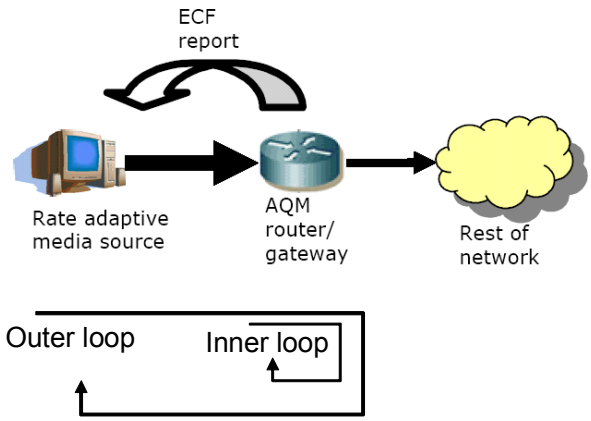

Figure 3 *Depiction of the inner and outer loop. The P-AQM runs two separate inner loops for the TCP and UDP, while only the UDP flow influence the outer loop.*

The RA engine at the source needs information about available bandwidth. Since our AQM is unaware of the layers above the transport layer, and there is no IP or UDP/RTP packet header field to transport this information in-band, ICMP Source Quench (SQ) [17] packets have been chosen as explicit information carrier of the congestion level, and named Explicit Congestion Feedback (ECF). Since the ICMP SQ signaling is directed directly towards the sources, the response will also be faster in addition to being much more precise. The ICMP SQ header has a 32 bit wide unused field that we allocated for our purpose. ICMP Echo packets are also submitted periodically from the P-AQM towards the contributing UDP sources to monitor both the average ($\overline{RTT}$) and maximum ($RTT_{\max}$) round-trip delay. The reason for this is that the new aggregated UDP byte rate is not completely visible at the AQM node before $RTT_{\max}$ seconds after the last ECF report was submitted. The time between each ECF report must therefore at least be $RTT_{\max}$. However, in order for the new aggregated UDP rate to be estimated with good accuracy, we define the time separation between each ECF report to be

$$ECF_p = RTT_{\max} + \tau_{est} \text{ [s]} \tag{4}$$

where τ_{est} is the period where the new aggregate byte rate is stable (this includes both the sources that adapt to the information, and any ill-behaving UDP sources not running any RA). ECF_p is rounded upwards so that divided on dT it gives integer κ. The new aggregate input UDP byte rate for period n is calculated as

$$\hat{r}_u(n) = \#UDP_B / \tau_{est} \tag{5}$$

where $\#UDP_B$ is the number of incoming UDP bytes in the period τ_{est}.

The goal of the RA is to obtain stability in the queue length with low delay and delay jitter; at least to avoid empty buffer (to ensure high link utilization) and avoid too full buffer (to

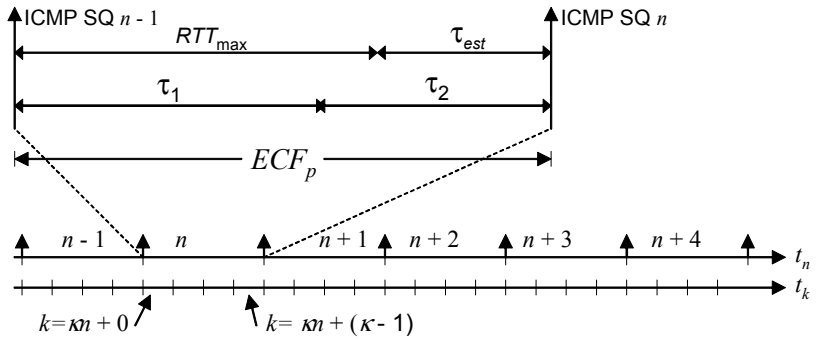

Figure 4 *The inner loop run at intervals as given by time line t_k. The outer loop run at intervals as given by time line t_n, in this example by granularity $\kappa = 4$. While the granularity of k is fixed and proportional to link capacity, ECF_p is adaptive, since $\tau_1 = \overline{RTT}$.*

avoid packet drops). The "inner loop" assists in avoiding tail-dropping, but bringing the queue length too close to the AQM equilibrium point N^* causes randomized packet drop ([4] shows 1% or more packet drop ratio in steady state). The solution is therefore to use two separate equilibrium settings for the UDP queue: N^* for the inner loop, and a smaller queue equilibrium point $N_u^* = \eta N^*$ for the outer loop, $\eta \in (0,1)$. The inner loop runs at a finer granularity than the outer loop, see Figure 4. In this figure, $\tau_1 = \overline{RTT}$ and $\tau_2 = ECF_p - \tau_1$. Thus, the change in queue size for UDP packets over one ECF_p period will be given as

$$\Delta N_u(n+1) = (\hat{r_u}(n) - \hat{c}_u)\tau_1 + (R_u(n+1) - \hat{c}_u)\tau_2 \tag{6}$$

where $R_u(n+1)$ is the new aggregate rate wanted by the P-AQM node to obtain N_u^* packets in the UDP-queue, and the output capacity $\hat{c}_u$ for UDP is estimated dynamically as

$$\hat{c}_u(n) = c \cdot \frac{\#UDP_f}{\#UDP_f + \#TCP_f} \tag{7}$$

i.e. the capacity c scaled by the number of UDP flows relative to all flows. The working condition was to keep the queue size of UDP packets close to N_u^*. Thus we have that

$$\Delta N_u(n+1) = N_u^* - N_u(\kappa n + (\kappa - 1)) \tag{8}$$

where $N_u(\ )$ time index $k = \kappa n + (\kappa - 1)$, i.e. the UDP-queue size sample read the last inner loop period in ECF_p period n (see Figure 4). Solving (6) and (8) explicitly for $R_u(n)$ we get

$$R_u(n+1) = \frac{\Delta N_u(n+1) - (\hat{r_u}(n) - \hat{c}_u)\tau_1}{\tau_2} + \hat{c}_u \tag{9}$$

which is the new wanted aggregated UDP byte rate. Since we seek a $\Delta N_u(n+1)$ that will remove the deviation from having exactly N_u^* UDP packets in the queue, the terms in the numerator will converge to zero and $R_u(n)$ towards $\hat{c}_u$.

To obtain fairness among the different UDP flows, we use additive increase (AI) and multiplicative decrease (MD). MD is calculated at the P-AQM as

$$MD(n+1) = R(n+1)/\hat{r}(n), \qquad R(n+1) < \hat{r}(n). \tag{10}$$

To avoid fairness convergence stalling when aggregate input byte rate matches closely both capacity and buffer equilibrium, $MD(n+1) = 0.96$ is signaled if five sequential *MD*s of values above 0.96 has been calculated the previous ECF_p periods.

The additive increase is chosen to be 50% of the value $\Delta R(n+1) = R(n+1) - \hat{r}(k)$ to provide smoothness in the increase of aggregate byte rate. In addition, the value is divided equally among the sources. Thus, the additive increase per source is given as

$$AI(n+1) = \frac{R(n+1) - \hat{r}(n)}{\#UDP_f(n)} 0.5, \qquad R(n+1) > \hat{r}(n). \tag{11}$$

So, depending on the relative size of $R(n+1)$, either $MD(n+1)$ or $AI(n+1)$ is stored in the ICMP SQ packets "unused" 32 bit field. Denoting this field $\mathrm{ECF}(n)$, and limiting (11) to $(1.0, \infty)$, each media source controlled by this signaling changes its average byte rate to

$$\begin{aligned} r(n+1) &= \mathrm{ECF}(n) \cdot r(n), && \mathrm{ECF}(n) < 1.0 \\ r(n+1) &= \min\{r(n) + \mathrm{ECF}(n), r_{\max}\}, && \mathrm{ECF}(n) > 1.0 \end{aligned} \tag{12}$$

where $r_{\max}$ is the maximum (original) rate of the source.

The media sources might receive ECF-values from multiple AQM nodes, decided by the number of such AQM nodes in the network path towards the receiver. The solution is for each RA media source to distinguish which node it receives these ICMP SQ packets from, and compute as many resulting byte rates as the number of different ICMP SQ senders. The media source then simply selects the lowest resulting byte rate, which will ensure max-min fairness.

The AIMD rate adaptation presented does not follow the AIMD(*a*,*b*) guidelines of [11], so it is the fairness between the UDP sources that this algorithm targets only. The TCP-friendliness is taken care of by the queue scheduling at the router itself. The challenge is to monitor and count the number of active TCP and UDP flows. Also, it differs from traditional TCP-friendliness [18], in that it does not differ its bandwidth share as function of RTT.

In this paper the media RA engine is assumed to make precise change of the transmitted byte rate in correspondence with the ECF(n) received in the ICMP SQ packets. Typically, state-of-the-art rate controllers of live encoders and transcoders are not able to meet the exact rate requirements of every group of frames, but we can assume it is met sufficiently well when averaging over longer intervals. Another assumption is that the change of target byte rate is made effective immediately after the reception of a new ECF(n). Typically, audio and video is compressed in blocks of frames and Group of Pictures (GOP = typically 12 frames for video, or 480ms for 25 frames per second). The compression parameters for CBR mode is optimized so that each GOP produce the same number of bytes, while VBR can vary its target rate on a frame by frame basis [21]. This means that the change of target rate is not activated until next frame (VBR) or start of next GOP (CBR). This will influence the adaptation speed, and might give somewhat higher drop probabilities than given by the results of this paper.

Also well worth discussing is our usage of rate adaptive CBR and Poisson sources for ECF/P-AQM test, and FTP source for DCCP/RED tests. The latter is simply a source that will follow the DCCP packet scheduling guidance perfectly, without having any bandwidth requirements, and is proposed by [14]. CBR will follow a steady packet rate following the AIMD given by ECF. The Poisson source intensity λ will follow the ECF, but the packet scheduling variance will vary according to normal Poisson behavior. It has been shown that open loop VBR encoding (no rate control) exhibits self-similar and long-range dependent (LRD) traffic [19,20]. Con-strained VBR has however removed almost all LRD [21]. The usage of Poisson source to model VBR in this paper can therefore be defended by assuming rate adaptive VBR based on the implementation outlined in [21].

3. The comparison of ECF to DCCP

Figure 5 shows the scenario simulated for the comparison of ECF and DCCP. Communi-

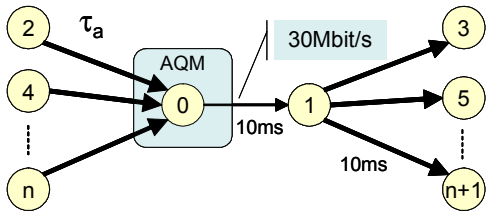

Figure 5 *The dumbbell network scenario simulated. AQM is P-AQM for ECF test, while gentle adaptive RED with ECN enabled for the DCCP tests.*

cating pairs is always (even_number,even_number+1). The propagation delays between node 0 and 1, and node 1 and receiving nodes, are fixed at 10ms. The delay between the sources and the AQM node, termed τ_a, is varied to arrange for different RTT scenarios. Note that DCCP will see end-to-end RTT, while ECF will see the RTT between the source

and the AQM node. Maximum buffer size is 400 packets. The equilibrium queue size for P-AQM outer loop is 70 packets, while RED average target delay is set to 100 packets. The RED parameters are set according to [22], and `gentle_=adaptive_=true`. For P-AQM, $K = 0.2$ and $\eta = 0.7$. The data packet size including transport layer headers is 1500 bytes. Both P-AQM and RED is set to byte count mode to adjust correctly for the shorter ICMP and DCCP ACK packets.

3.1 Transient behavior

In the first test, five sources are started in sequence at *t*=0, 10, 20, 30, and 40s, and in the second test all five sources are started simultaneously but stopped in sequence at *t*=60, 70, 80, 90, and 100s. $\tau_a = 0.01s$ and CBR and Poisson source original average bit rates are 20Mbps. For DCCP, the access bandwidth is limited to 20Mbps, so that the maximum bit rate of DCCP will equal that of ECF. Thus, the fair bandwidth share when 1, 2, 3, 4, and 5 sources are active, is 20, 15, 10, 7.5, and 6Mbps, respectively. While ECF shows stable,

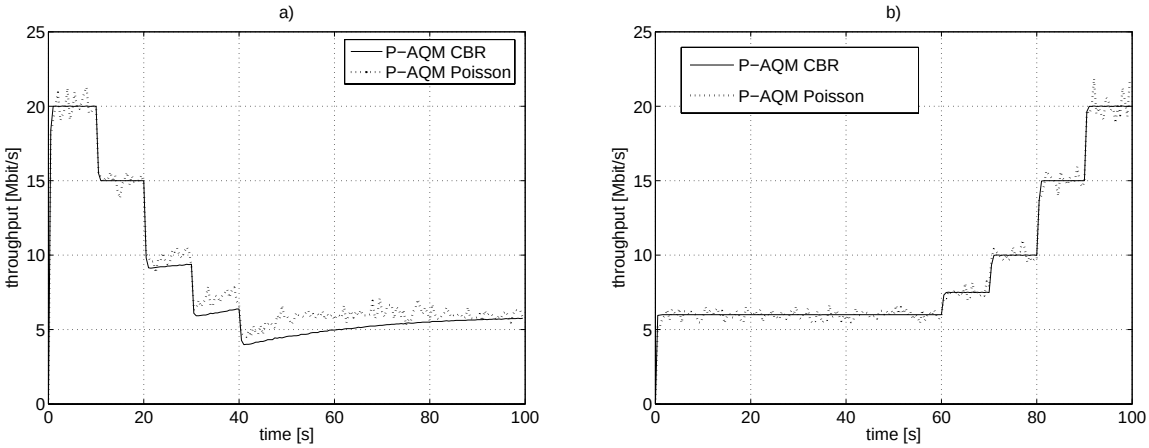

Figure 6 *The precise feedback provided by periodic ICMP SQ packets make ECF very fast and accurate. The curves shows the throughput as counted bytes received per 0.5s at receiving node 3.*

accurate and fast rate adaptation (Figure 6), DCCP (Figure 7) is burdened with sluggish bandwidth (TCP-like) or slow response time (TFRC). The unstable bandwidth of DCCP TCP-like will result in varying perceived media quality even in the periods between new flow arrivals. For ECF we notice that the convergence time to fair bandwidth is a bit slow with CBR sources in the first test because the aggregated rate is adjusted too accurately. The Poisson test reveals that a more natural unstable source than this clean CBR will force a much faster convergence time. This is known from adaptive control theory as "persistent excitation", in that the system needs to be excited by sufficiently rich input signals for efficient estimation of unknown parameters. In follow-up work also more accurate VBR models [19]–[21] will be included in similar tests.

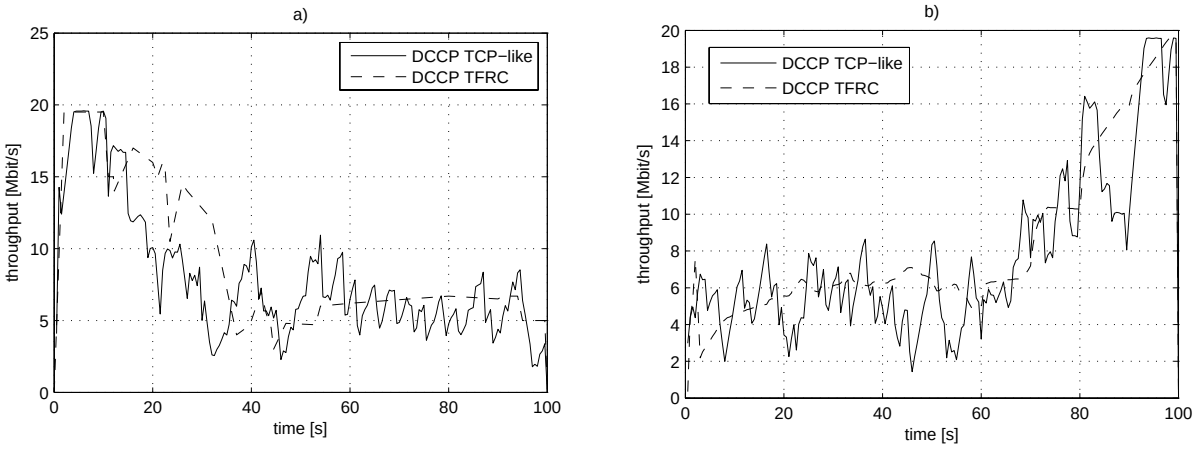

Figure 7 *The throughput received at node 3 when using DCCP TCP-like and TFRC: a) shows the test with increasing number of sources, and b) shows decreasing number of sources. Notice the much more sluggish bandwidth share in comparison with the ECF results in Figure 6.*

3.2 Steady-state behavior

In this section the focus is on packet drop, delay and delay jitter, and fairness of bandwidth share. All simulations were run over 100s, and the data results from the first 10s were removed in order to suppress the transient effects. The sources were started in sequence, but only separated by 10ms. The RTT and the number of sources were varied.

The packet drop statistics showed as expected that DCCP TCP-like over ECN-enabled RED had zero drop, at least for 10 and 30 active sources. However, when 100 sources was activated, even DCCP experienced packet drops, at 5.7, 5.3, and 3.2% for RTT=60, 100, and 240ms, respectively. DCCP TFRC was run in two tests only, and had slightly better performance: zero drop at 30 sources, and 1.2% drop at 100 sources and RTT=100ms. For P-AQM tests using ECF RA, the drop statistics showed approximately 0.01, 0.1, and 0.7% for the same end-to-end (e2e) RTTs, respectively. Most simulations showed between 99.8 and 100% link utilization, since the queue buffer almost never drained. DCCP went down to 92% utilization at some tests; this was due to the use of too short buffer for RED to avoid strong queue oscillations.

Figure 8 shows the results for average queue delay, queue delay jitter, and fairness. The top left plot shows that DCCP over RED is not able to maintain target delay (100 packets of 1500 bytes over 30Mbps is 40ms) when the number of sources and RTT is varied. Not shown in the figure is a test with 100 DCCP TFRC sources: the results were almost identical to DCCP TCP-like. Also not shown in the figure is 100 DCCP TFRC sources run over P-AQM instead of RED: the results show that delay for the same three RTTs is between 40 and 50ms, i.e. 15ms lower. ECF on the other hand is capable of holding its

target delay (70 packets of 1500 bytes over 30Mbps gives 28ms) much better, only for 100 sources and RTT=240ms (e2e) the average delay *drops* to about 18.7ms.

Top right plot shows the delay jitter, which also shows that DCCP delay variance increases as the number of sources and RTT increases. ECF is able to control the jitter around 2–5ms maximum, except for 100 sources at 240ms which gives 9.6ms. DCCP TCP-like with 100 sources has severe jitter due to significant queue oscillations. Not shown in the figure are the similar tests with DCCP TFRC over RED and over P-AQM: the RED test results were comparable to ECF, while the P-AQM tests were in fact even lower (2.8, 3.9 and 10.8ms).

The bottom left and right plots of Figure 8 shows the fairness of bandwidth share, taken from the previous tests of DCCP TCP-like and ECF CBR, where the number of sources was 30, and the RTT was 60 and 240ms, respectively. In addition, the 60ms test was expanded with results running ECF with a Poisson source, and the 240ms test expanded by DCCP TFRC. The ECF CBR tests shows stable fairness close to 1.0, except for the

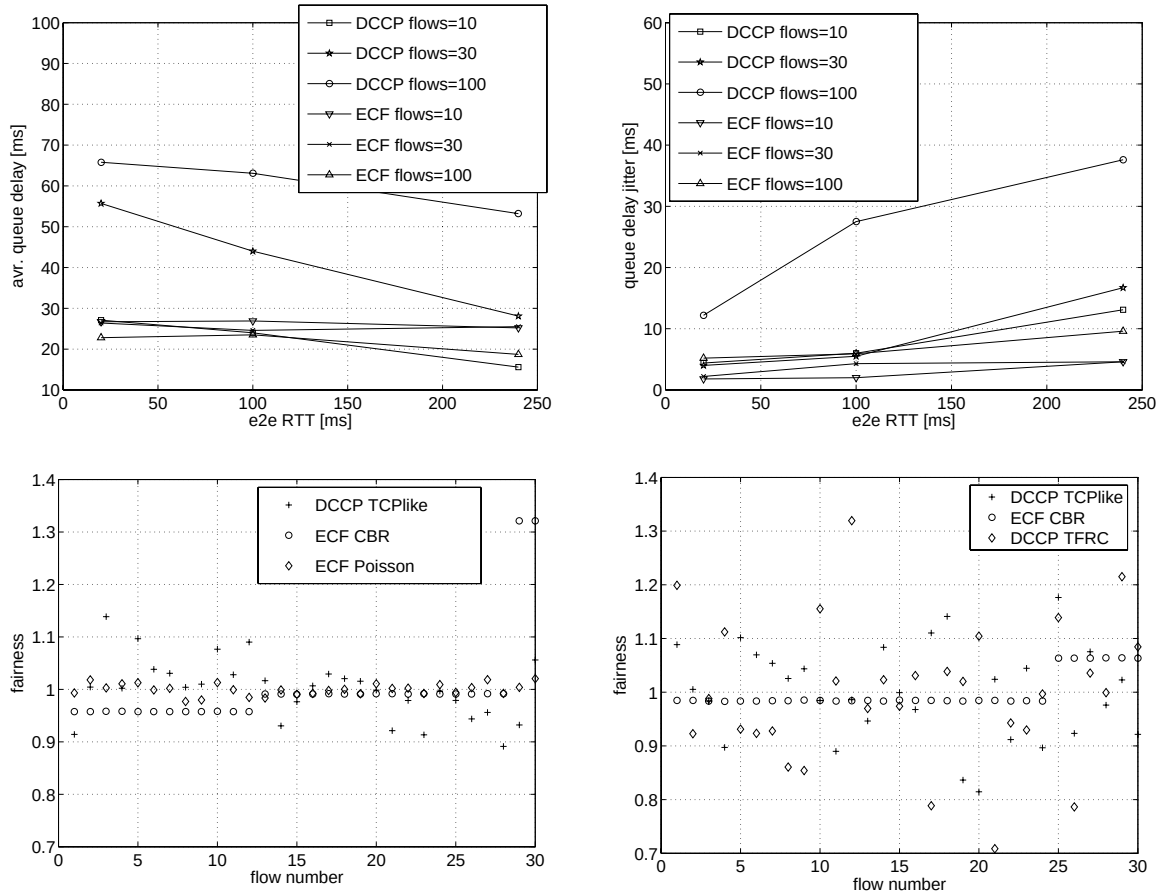

Figure 8 *DCCP TCP-like and ECF control of CBR sources comparison: Top left: average queue delay. Top right: queue delay jitter. Bottom left: fairness at RTT=60ms (e2e). Bottom right: fairness at RTT=240ms (e2e). 30 sources sharing 30Mbit/s link means that fairness=1.0 is 1.0Mbit/s.*

sources started last, which are given higher bandwidth than its fair share. The reason for this is that for these sources, the convergence towards fair share is done from above 1Mbps and towards 1Mbps, while for the majority of the sources, the convergence is from below 1Mbps and upwards (as seen in the example of Figure 6a). Notice that for ECF controlling Poisson (i.e. constrained VBR) sources, this phenomenon is gone, and all values lie in the region 0.98 to 1.02. The DCCP results shows that the fairness lies in the region 0.89 to 1.14 and 0.78 to 1.3 for the RTT=60 and 240ms, respectively.

4. Discussion

The results of DCCP TCP-like show that although it is fast to adapt, the bandwidth share never stabilizes, as one could guess since it follows TCP's AIMD. The fairness among DCCP flows is not satisfactory (Figure 8), which is in compliance with similar research [23]. Run over RED+ECN, the queue delay grows as the number of sources grows, indicating that RED is marking with too low probability. Also the queue delay jitter grows substantially, indicating increasing level of queue oscillations. DCCP TFRC over RED shows smoother bandwidth share, but it is slower in convergence time. TFRC gets the similar average delay as TCP-like, while delay jitter is lower. One surprise was the nice results when running DCCP TFRC over P-AQM+ECN in place of RED+ECN: the target delay deviation was less, as was the delay jitter.

In all tests however, ECF over P-AQM shows superior performance in both the transient and steady-state tests. The periodic signaling of 32 bit ECF(n) congestion metric using ICMP SQ packets is the key component. This signaling will of course cost some capacity. The original ICMP SQ packet consists of the usual 20 byte IP header and 36 byte for the ICMP SQ. For our purpose only 8 of these 36 bytes are usable, so a dedicated ICMP ECF type had been more bandwidth efficient. In the transient tests in Section 3.1 the ECF_p was calculated to 160ms, while in the tests in Section 3.2 with longest RTT the ECF_p was 520ms. In the case with 5 sources in Section 3.1, 5 times 56 byte per 160ms gives 14kbit/s (0.05%), while 100 sources in the latter test gives 86kbit/s (0.3%). These numbers increase with increasing number of sources and decreasing RTT. The periodicity of ECF_p should therefore be adjusted upwards to avoid signaling traffic load above some defined limit. For RTCP it is normally defined to 5% of the RTP traffic, for comparison. DCCP also generates overhead due to the acknowledgement packets. In addition ECF over P-AQM signals ICMP Echo (ping) packets in order to monitor the dynamics in RTT: the periodicity of this signaling should also be limited based on number of sources.

5. Conclusions

This paper has described a novel Active Queue Management approach based on a proportional gain controller (P-AQM), designed for controlling streaming media carried by UDP

packets, and a novel congestion feedback mechanism ECF, using ICMP SQ packets to signal the 32 bit congestion metrics. *ns*-2 simulations is performed in order to compare ECF over P-AQM to DCCP over RED.

DCCP TCP-like and TFRC over RED showed that the average target delay and jitter was increasing at increasing number of contributing sources, and somewhat decreasing at increasing RTT. One test showed DCCP TFRC over P-AQM outperformed TFRC over RED when it comes to average delay and delay jitter. All stationary tests for ECF showed superior performance compared to DCCP. Also, the transient tests revealed that ECF is a much more accurate and faster adaptation scheme compared to both TCP-like and TFRC variants of DCCP, indicating it will provide more stable perceived media quality with less end-to-end delay than DCCP.

References

[1] L. A. Rønningen, A. Lie, “Performance Control Of High-Capacity IP Networks For Collaborative Virtual Environments”, IBC 2002 Conference Proceedings, 12–15 September 2002.

[2] A. Lie, L. A. Rønningen, “Distributed Multimedia Plays with QoS guarantees over IP”, IEEE Wedelmusic’03 14–17 Sept., Leeds UK, 2003.

[3] A. Lie, O. M. Aamo, L. A. Rønningen, “On the use of classical control system based AQM for rate adaptive streaming media”, 17th Nordic Teletraffic Seminar, Fornebu Norway, August 2004.

[4] A. Lie, O. M. Aamo, L. A. Rønningen, “Optimization of Active Queue Management based on Proportional Control System”, IASTED CIIT, St. Thomas, US Virgin Islands, November 2004.

[5] T. V. Lakshman, P. P. Mishra, K. K. Ramakrishnan, “Transporting Compressed Video Over ATM Networks with Explicit Rate Feedback Control”, Proceedings of the IEEE Infocom'97, USA 1997.

[6] T. Lakshman, A. Ortega, A. Reibman, “VBR Video: Trade-offs and potentials”, Proc. of the IEEE, Vol. 86 No. 5, May 1998.

[7] S. Floyd, K. Fall, “Promoting the use of end-to-end congestion control in the Internet”, IEEE/ACM Trans. on Networking, Vol. 7 No. 4, pp. 458–472, 1999.

[8] E. Kohler, M. Handley, and S. Floyd, http://www.icir.org/kohler/dccp/draft-ietf-dccp-spec-09.txt

[9] S. Floyd, V. Jacobson, “Random Early Detection gateways for congestion avoidance”, IEEE ACM Transactions on Networking, Vol. 1, No. 4, August 1997.

[10] K. Ramakrishnan, S. Floyd, D. Black, “The Addition of Explicit Congestion Notification (ECN) to IP”, IETF RFC3168, September 2001.

[11] S. Floyd, M. Handley, and J. Padhye, “A Comparison of Equation Based and AIMD Congestion Control,” ACIRI Technical Report http://www.aciri.org/tfrc/aimd.pdf, May 2000.

[12] S. Floyd, M. Handley, J. Padhye, “A Comparison of Equation Based and AIMD Congestion Control”, Technical report ACIRI, Berkeley, 2000.

[13] The Network Simulator — ns-2, http://www.isi.edu/nsnam/ns/

[14] N. E. Mattsson, “A DCCP module for ns-2”, Luleå Tekniska Universitet, ISSN 1402-1617. www.dccp.org

[15] A. Lie, O. M. Aamo, L. A. Rønningen, “Balanced VBR rate adaptation for controlled delay and quality of streaming media”, submitted to IEEE Infocom ‘06.

[16] H.-F. Hsiao, J.-N. Hwang, “A max-min fairness congestion control for streaming layered video”, In Proc. IEEE ICASSP '04, Volume: 5, 17–21 May 2004

[17] J. Postel, “Internet Control Message Protocol”, IETF RFC 792, September 1981.

[18] F. Kelly, “Fairness and stability of end-to-end congestion control”, European Journal of Control 2003.

[19] M. Garrett, W. Willinger, “Analysis, Modeling and Generation of Self-Similar VBR Video Traffic”, ACM Sigcomm, London 1994.

[20] M. Krunz, A. Makowski, “A source Model for VBR Video Traffic Based on M/G/∞ Input Processes”, Proc. of IEEE Infocom 1998.

[21] M. Hamdi, J. W. Roberts, P. Rolin, “Rate control for VBR video coders in broadband networks”, IEEE Journal on Selected Areas in Communications, Vol. 15 no. 6, 1997.

[22] RED parameters, http://www.icir.org/floyd/red.html#parameters

[23] S. Takeuchi et al., “Performance Evaluations of DCCP for Bursty Traffic in Real-Time Applications”, Proc. of 2005 Symposium on Applications and the Internet, IEEE, (SAINT’05).

Paper E

Evalvid-RA: Trace Driven Simulation of Rate Adaptive MPEG-4 VBR Video

Arne Lie, Jirka Klaue

Published in
Springer Multimedia Systems Journal

ISSN 1432-1882 (online), 2007
ISSN 0942-4962 (print), 2008
DOI 10.1007/s00530-007-0110-0

Paper E

Evalvid-RA: Trace Driven Simulation of Rate Adaptive MPEG-4 VBR Video

Arne Lie and Jirka Klaue

Abstract

Due to the increasing deployment of conversational real-time applications like VoIP and videoconferencing, the Internet is today facing new challenges. Low end-to-end delay is a vital QoS requirement for these applications, and the best effort Internet architecture does not support this natively. The delay and packet loss statistics are directly coupled to the aggregated traffic characteristics when link utilization is close to saturation. In order to investigate the behavior and quality of such applications under heavy network load, it is therefore necessary to create genuine traffic patterns. Trace files of real compressed video and audio are text files containing the number of bytes per video and audio frame. These can serve as material to construct mathematical traffic models. They can also serve as traffic generators in network simulators since they determine the packet sizes and their time schedule. However, to inspect perceived quality, the compressed binary content is needed to ensure decoding of received media. The EvalVid streaming video tool-set enables this using a sophisticated reassembly engine.

Nevertheless, there has been a lack of research solutions for rate adaptive media content. The Internet community fears a congestion collapse if the usage of non-adaptive media content continues to grow. This paper presents a solution named Evalvid-RA for the simulation of true rate adaptive video. The solution generates real rate adaptive MPEG-4 streaming traffic, using the quantizer scale for adjusting the sending rate. A feedback based VBR rate controller is used at simulation time, supporting TFRC and a proprietary congestion control system named P-AQM. Example ns-2 simulations of TFRC and P-AQM demonstrate Evalvid-RA's capabilities in performing close-to-true rate adaptive codec operation with low complexity to enable the simulation of large networks with many adaptive media sources on a single computer.

KEY WORDS

Congestion control, rate control, streaming media, VBR video, network simulation.

1. Introduction

The Internet is today facing a change of the traffic type dominating the aggregates at network core and edges. Interactive VoIP and videoconferencing are currently having an exponential growth of usage, but also one-way streaming media (e.g. VoD and WebTV) is experiencing large growth rates [Pal04, UNI03]. Since the majority of this media content is controlled by technology that does not monitor traffic load nor scale the bit rate during the ongoing sessions, serious quality degradation due to traffic overload (i.e. packet drops and excessive delays) and throughput unfairness might result. Typically, such services probe the network throughput only during session startup, if at all, and initiates one of a few possible quality versions based on current network state and end user terminal characteristics. The MPEG and commercial video communities have developed several advanced solutions to answer the media scalability challenge: (i) scalable video with base layer and enhancement layers [ISO94], (ii) FGS (Fine Granular Scalability) [ISO99], and (iii) several multi-rate coding schemes (e.g. Envivio, Microsoft Intelligent Streaming, Real SureStream). (i) has the benefit of efficient file storage, but the total flow sent has lower compression efficiency than flows from codecs with only a single layer. FGS can be adjusted to finer bandwidth granularity than ordinary scalable coding, at the cost of higher complexity and still lower coding efficiency [Li01]. While MPEG-4 FGS has failed in the market, the new H.264 SVC might give FGS related technologies a new chance [WWH06].[1] Multi-rate coding stores typically three tracks with different optimized bit rates in a single file, and the selected track can be switched on-the-fly during streaming time. While suffering from the highest storage capacity needs, this solution is still receiving most commercial interest, due to its simplicity and good transmission bandwidth utilization.

In contrast to offline coding approaches, which actually build their scalability capabilities on coarse network state assumptions, *online real-time codecs* for e.g. videoconferencing can adjust codec parameters on the fly to adapt to the current network state on much finer time granularity. This paper presents analysis and tools supporting research within real-time encoding, but as our conclusions will argue, its architecture is applicable also within offline encoding.

If popular media continues to be non-adaptive, video services may consume much more than their fair share of capacity such as when competing with TCP flows at network bottlenecks, and this breaks the best effort Internet[2] principles [FF99]. This unfairness adds to the already mentioned problems with queuing delay and packet loss. As an answer to these network challenges, the IETF has during the last years worked on a new real-time

1. Still, FGS is very complex and cannot be part of real-time encoding.
2. Even this paper focus on best effort class of traffic, rate adaptation can also be used within DiffServ classes. Note that also expedited forwarding DiffServ QoS breaks if too many non-adaptive applications are requesting it.

media transport protocol named Datagram Congestion Control Protocol (DCCP) [KHF06], to support the deployment of rate adaptive codecs. UDP has no congestion control mechanism like TCP. The main idea of DCCP is to continue to use UDP's non-reliable packet flow (no retransmissions in case of packet drops), but make it connection-oriented like TCP. The latter will enable better firewall penetration capabilities and the possibility to exchange different parameter values at session initiation, such as the choice of congestion control algorithm. TFRC (TCP Friendly Rate Control) is the most fitted DCCP congestion control profile for video traffic [FKP06], using equation based control in order to obtain smooth rate at an average similar to that of TCP.

Many other solutions have also been proposed over the last decade to solve these problems, among them VBR over ATM ABR services [LMR97], RAP [RHE99], MPEG-TFRCP [MWMM01], LDA+ [SW00], and P-AQM+ECF [LAR05]. All of these have slightly different objectives, but agree on the target goal of assisting the network to provide fair and stable services. The proposals can be divided into two main groups: (i) those who are pure end-to-end oriented and only monitor the network state by packet loss statistics feedback, and (ii) those who in addition also take advantage of more advanced network state information, such as the binary ECN marks [RFB01], or explicit information on traffic load from each node on the path from sender to receiver. Since the Internet community puts a strong focus on scalability, pure end-to-end oriented systems are preferred. However, there are concerns whether this is sufficient to ensure low delay and packet loss in traffic overload situations. Vital parameters are rate adjustment speed and accuracy. The interplay with the other media delivery chain functionalities are also of major importance, such as traffic shaping, jitter buffer dimensioning and control, and decoder robustness to packet losses. The proposed solutions should therefore be compared taking all these parameters into account.

In order to perform research on vital streaming media parameters, both at network/transport layer *and* application layer, the setup of true multimedia test networks might seem necessary. This can however be very expensive and of little flexibility. Thus, network simulations, using tools like the *ns*-2, might seem tempting. The problem with the latter is that one is stuck with either using synthetic video/audio models or static audiovisual trace files for source traffic generation. Since our goal is to implement media rate control based on traffic feedback, the source models need to be rate adaptive. Such modification of synthetic models is straight forward, but then the goal of investigating perceived quality is excluded. For this reason real audiovisual trace files must be used in order to inspect perceived quality. One possibility for support of the latter is to use the EvalVid tools-set [KRW03] invented by J. Klaue. EvalVid is an open-source project, and supports trace file generation of MPEG-4 as well as H.263 and H.264 video. Using it together with the *ns*-2 interfacing code suggested by C.-H. Ke [Ke04], perceived quality and objective measure

like PSNR calculation can be obtained after network simulation. But still, this does not provide a solution for *rate adaptive* video investigation.

All this has motivated the design and implementation of Evalvid-RA, a tool-set for rate adaptive VBR video investigation in *ns*-2, based on modifications to the EvalVid version 1.2 tool-set and the *ns*-2 interfacing code. The solution framework is generic so that it can be implemented within any network simulator, and on any codec, provided that a set of guide-lines is followed. The paper is organized as follows: Section 3 gives first an overview over the standardized methods for video evaluation. In Section 4 the necessary framework building blocks are introduced and explained. The performance of this framework is investigated in this paper using a video rate controller presented in Section 5. By running *ns*-2 simulation example scenarios presented in Section 6, the Evalvid-RA capabilities are demonstrated, focusing on traffic characteristics and rate controller performance in various protocol and network environments. The contributions of this paper compared to referenced work are

- The EvalVid v1.2 tool-set is enhanced to support *rate adaptive* video (Evalvid-RA).
- The SVBR [HRR97] rate controller is modified to become an adaptive rate controller (RA-SVBR).
- The absence of Long Range Dependency (LRD) in aggregate VBR rate adaptive video traffic without the use of traffic shaping buffer is demonstrated using Evalvid-RA's realistic video traffic generators.
- The quality of rate adaptive MPEG-4 streaming with conversational delay constraints is calculated using the Evalvid-RA tool-set (e.g. PSNR). Different protocols (UDP and TFRC) and network types (FIFO, RED, P-AQM) are used and compared in mixed TCP traffic scenarios.

The goal of this paper is to present the Evalvid-RA architecture, to validate its performance, and lastly to exemplify utilization in adaptive streaming media research, showing how increased network intelligence can improve streaming performance.

2. Related Work

Evalvid-RA connects multiple independent research areas: (i) media rate control, (ii) media traffic characteristics, (iii) network congestion control, and (iv) efficient and error resilient coding. Rate control includes sender and receiver buffer dimensioning, to avoid both overflow and underflow, as thoroughly analyzed in [RH92]. VBR video traffic characteristics have been reported e.g. by [GW94, BSTW95] with following Markov and ARIMA modeling by e.g. [KT97, ALS02, LKK04]. The latter modeling however does not take adaptive rate control into account. Network congestion control schemes for media content were listed in the Introduction. Since Evalvid-RA includes real media decoding, coding efficiency and packet loss resiliency will also be taken into account, as PSNR and

possibly other QoS measures are calculated, or decoded video is actually consumed by human observers.

In recent years, papers have been published on the topic of the simulation of rate adaptive media, and also real experimental studies have been set-up to test e.g. early DCCP prototypes efficiency. In [ZML04] MPEG-4 trace files are used to calibrate a TES (Transform Expand Sample) mathematical model, and rate adaptation is incorporated by adjusting the frame size output by a scalar (from rate-distortion curve). The simulation model however has no on-line rate controller, and since the traffic is synthetic, perceived quality cannot be investigated. In [MS02] the authors set up a simulation scenario where both temporal and quantizer scale adaption is possible. But again the traffic is synthetic. H.263 video trace files are used in [XLZ05], and the sending rate is controlled by DCCP TCP-like. However, the video is not rate adaptive, so the video submission is controlled by overruling the real-time constraint. In [XH06] models are derived for pre-recorded media streaming over TFRC and compared to simulations. The models focus on the impact of the TFRC rate changes to the probability of rebuffering events, i.e. events where the receive buffer is emptied. Recently, more realistic simulation implementations have been published, such as [G06] where rate adaptation using frame discard and FGS has been studied and implemented in ns-2 by also inserting the binary content directly into the simulator packets, thus supporting media decoding and PSNR calculation. The benefit of inserting the binary data into the ns-2 packets is that there is no need of keeping track of additional simulation time trace files. However, the penalty is higher computational load at simulation time, limiting the practical size of the network and number of simultaneous video sources. [BENB06] is an example of a recent experimental study of real VoIP traffic using DCCP, using real applications and networks.

To the best of our knowledge, Evalvid-RA is the first tool to create realistic “online” rate adaptive streaming media traffic. It includes

- a simulation time rate controller to modulate the quantizer scale used by a real codec
- realistic frame packetizing
- the ability (through ns-2) to choose network complexity, protocol and queue management support
- a framework that is scalable to a large number of simultaneous video sources
- and finally at the receiver side being able to restore the media files supporting PSNR and other QoS metrics calculation.

Due to the trace file approach of Evalvid-RA, absolute delay and delay jitter impairments to the media decoding process can be investigated in a post-process, thus decoupling network and receiver media player constraints. Although we recognize the importance of mathematical models for traffic and queue statistics analysis, we believe that the complexity of the heterogeneous networks makes realistic simulation a better tool, especially when

being able to compute end-user QoS metrics such as PSNR, or even perform human subjective tests.

3. Video Quality Evaluation

The quality of a video transmission depends on the impression a human observer receives of the delivered video. Though traditional network metrics such as bandwidth, packet loss, jitter and delay, certainly influence the video quality, the perceived *subjective* quality impression of a human observer is nevertheless the most important factor. The subjective video quality test results are expressed by means of e.g. the mean opinion score (MOS) as defined by the ITU. The MOS is a scale from 5 (excellent) to 1 (bad). In contrast, *objective* video quality metrics are calculated by computers. Basically, these can be divided into pixel-based metrics, like SNR or *PSNR*, and *psycho-visual* metrics. The latter approach, which is based on models of the human visual system (HVS), has been shown to outperform standard quality metrics like PSNR in most cases [WP02, Win05][1]. However, sometimes the absolute value of the video quality and its correlation to subjective tests is not the most important factor but rather the relative quality regarding a certain optimum. An example would be the comparison of different transport protocols with an assumed error-free transmission. In these cases simple metrics like PSNR are still adequate. Another downside of psycho-visual metrics is their complexity and, thus, huge computational overhead compared with PSNR.

If the influence of network characteristics and parameter optimization is to be assessed in terms of real subjective video quality, a dedicated metric should at least be included in the target function of the optimization. We recommend the application of the video performance estimation method standardized by ANSI [T1.03], since it outperforms PSNR and similar methods as shown in e.g. [WP02]. Though a lot of research about video quality assessment has been done – and is still in progress – the field is by no means finished. Nevertheless a variety of reasons has been identified why objective metrics like PSNR are not adequate for performance evaluation. In [ZKSS03] the influence of the frequency and amplitude of quality fluctuations in layered video transmission has been investigated. It has been shown (amongst others) that it is better to minimize the frequency of fluctuations even if the average PSNR decreases.

Another problem which must be faced is the quality assessment of long video sequences. Usually one quality indicator per video sequence is calculated, which describes the impression of an average (non-expert) human observer. This is well fitted for the relatively short video sequences for which these metrics are verified. However, one quality indicator is not enough for longer video sequences since short but sharp disturbances could be masked by the averaging over longer time spans. Since periodically occurring

1. I.e. their results come closer to subjective tests.

disturbances could influence the overall impression of a video transmission, a quality assessment method should also reflect this. One possible solution is the calculation of the video quality – with any method – in a sliding window of, e.g., 10 seconds. The quality indicator of each window is compared to the quality indicator of the corresponding video part before transmission. The frequency of degradations could be used as overall quality measure for the transmission instead of the averaged quality indicator. Another possibility is the specification of a threshold for a tolerated number of quality indicator deviations. The Evalvid-RA tool-set provides a method which can calculate these figures for long videos. For this purpose the `miv.exe` tool from EvalVid v1.2 is used. This quality indicator is introduced and explained in detail in [GKKW04].

There is no generally accepted method to access the quality of a video transmission system. Though some aspects of the problem have been discussed in this section, an in-depth study would be beyond the scope of this paper. The citations in this section provide a good start for further reading.

The Evalvid-RA framework supports the use of any metric since the calculation of actual quality values is separated from the simulation process. Only PSNR-calculation is included directly in the tool-set, but the use of subjective metrics has successfully been tested in [WBSS04], [ITS] and [Sar]. The included MOS calculation tool uses a simple mapping of PSNR values to MOS (defined in [Ohm95]) which nevertheless achieves quite good correlation with [T1.03] in most cases.

4. The Evalvid-RA architecture guidelines

An efficient tool-set for network simulation must be scalable so that even large networks with many sources and many network nodes can be simulated on a single computer. Two major challenges result from this ambition: (i) perceptual quality inspection at receiving nodes, and (ii) the implementation of an adaptive rate controller having access to both media content and network state feedback. The first challenge could easily be solved by using real binary packet data as packet payload in the network simulator. However, such an attempt will degrade the simulator performance significantly. A more efficient approach is to use unique packet identifiers to support video frame assembly as a post-process. The existing EvalVid tools [KRW03, Ke04] uses this approach, by introducing a trace file generation process, a network simulator process, and a post-process. The second challenge is however in conflict with the division between pre-process and network simulation process, because it is only the pre-process that has access to the media and codec itself. Thus, one need to find a method supporting the exchange of necessary information between those two processes. Obviously, the solution is dependent of the kind of rate controller in use.

4.1 The selection of a rate controller

Traditionally, video rate controllers are divided into three categories: (i) constant bit rate (CBR), (ii) variable bit rate (VBR), and (iii) quality based (open loop VBR). In CBR, the rate controller constraint is to produce a constant number of bits per time unit such as the Group of Pictures (GOP) (if it has a constant number of frames per GOP). To achieve this goal for a hybrid codec using DCT transform (e.g. MPEG), the *quantizer scale* (which holds the quantization value matrix for the DCT transformed 8x8 pixel blocks) is considered changed for each macro-block (16x16 pixel block) [JTC99].[1] The bit rate budget is optimized looking at several sequential video frames, causing an algorithmic delay in the rate controller. Due to this delay, interactive applications are better off with a VBR rate controller, which trades lower delay for higher bandwidth variability. Other benefits with VBR are more constant quality and higher multiplexing gain potential. VBR typically considers quantizer scale changes at each new video frame, or even only at each GOP. The third option, open loop VBR, is actually coding without any rate controller, i.e. the quantizer scale is fixed during the whole sequence[2], thus producing the highest bit rate variability. The bit rate produced is highest in high motion scenes, and when there are many details and hard contrasts.

To limit the size of the trace files needed as input to the network simulation, they are captured at frame granularity, i.e. the size of each frame in bytes is stored in a log file. This rules out CBR, since in that case we would have needed access to sizes on macro block granularity[3]. The rate controller choice will therefore be based on VBR. Before deciding on the granularity of the rate controller, the interplay between the pre-process and network simulator must first be considered.

4.2 The pre-process

The goal is to have an online rate controller in the network simulator, but without having to do the media encoding itself, since that will demand too much CPU resources during simulation time. The media encoding must be performed in a pre-process. In MPEG-4 [ISO99], the valid quantizer scale values are in the range 1 to 31, with 1 producing the highest quality and bit rate. The key idea is then *to encode the media with open loop VBR for all possible quantizer scales*[4]*, store the frame sizes per quantizer scale in separate files, so that the online rate controller in the network simulator can select a new quantizer*

1. Wavelet coding as in MJPEG2000 should also be possible within Evalvid-RA framework.
2. Still, the different frame types may use different quantizer scales, e.g. I-frames use scale 8 while P-frames use scale 12, and B-frames scale 16, but fixed during the sequence.
3. This is however not a big sacrifice, since the most challenging research are within interactive media, where the algorithmic delay of CBR should be avoided. However, it also rules out H.264 slice mode: a future Evalvid-RA upgrade to H.264 should therefore include slice granularity trace files as an option.
4. Here we choose to use the same quantizer scale for all types of frames.

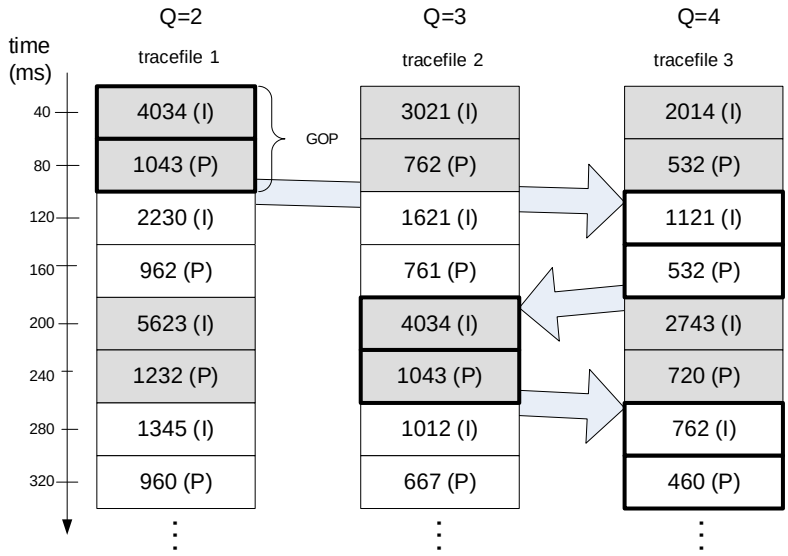

Figure 1 *The Evalvid-RA main concept by letting the simulation time rate controller choose correct frame sizes (emphasized boxes) from distinct trace files valid for each quantizer scale. The figure shows a simplified example of a 25fps video using three quantizer scale values and GOP size of two (one I- and one P-frame).*

scale value and get the correct frame sizes from the corresponding trace file. The simplest and most correct option is to allow the rate controller to consider a new quantizer scale only at the start of a new GOP. By keeping the GOP size fixed, the rate controller will always find an I-frame as the first frame after trace file switching. The concept is depicted in Fig. 1 for GOP size of two frames, and only three different quantizer scale values 2–4. The synchronized GOP boundaries will ensures a refresh of the motion prediction algorithm, and all succeeding P- and B-frames in that GOP will be based on that I-frame.

Changing to another quantizer scale during a GOP is however also possible without causing too noticeable artifacts, but a real encoder with rate controller would then produce the next P- or B-frame based on a slightly different compressed I-frame (i.e. the same frame but not the same quantizer scale) than the one used in the simulation. To explain this with an example, let's consider a codec that produces 12 frames per GOP and only I- and P-frames (the I-frame is number 1, while frames 2–12 are P-frames). The rate controller has chosen quantizer scale 5 for an ongoing GOP. At frame number 7 in that GOP, the rate controller suggests changing to scale 10, since the bit rate budget is somewhat overrun. A real live encoder would then produce frame 7 (P-frame) based on frame 6 having a quantizer scale of 5. However, using a separated "offline" encoder and live rate controller (seen from the network simulator), frame 7 is already produced in the pre-process, based on frame 6 having a quantizer scale of 10 also. Although the artifacts produced would not be too much noticeable (verified by own experiments not documented further in this paper), this observation concludes that the only correct option is to have equally sized GOPs and a VBR rate controller that works on GOP granularity.

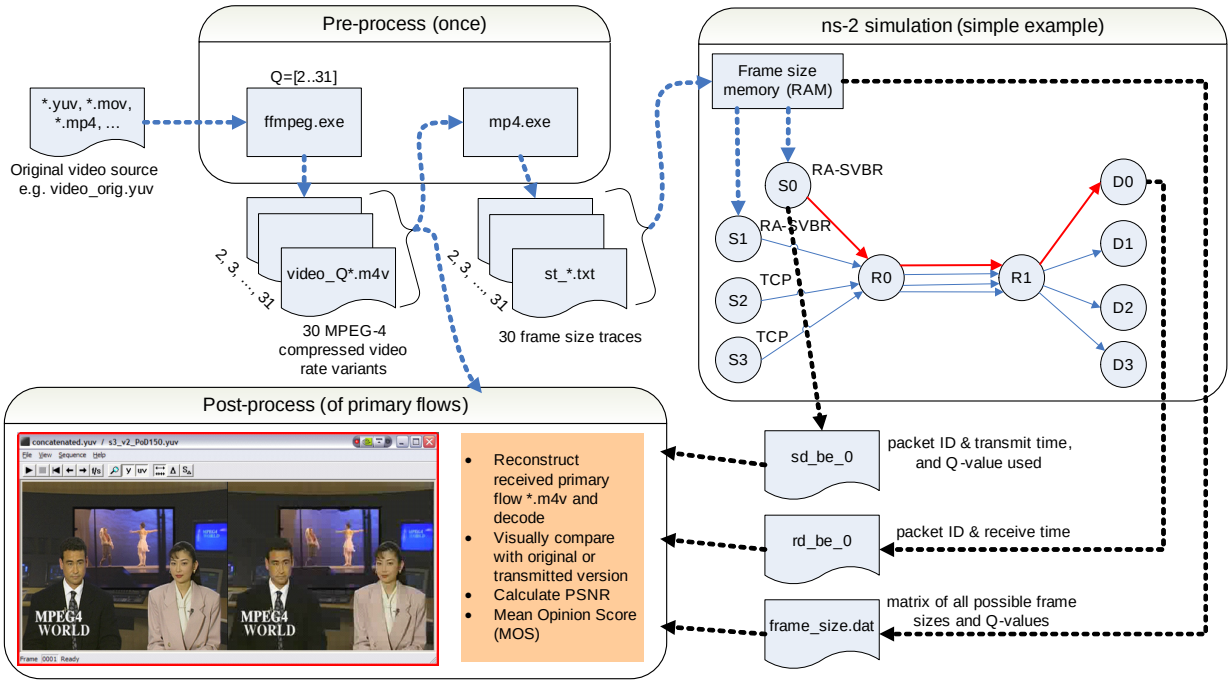

Figure 2 *An overview of the Evalvid-RA framework: pre-process, network simulation, and post-process. The 30 trace files* `st_*.txt`*serve as input to the network simulator. This example shows two video sources competing for network capacity with two FTP over TCP applications. The source S0 to destination D0 is selected as primary flow.*

To summarize this subsection, the pre-process must run an encoder for each media file that shall be used in the network simulation 31 times at open-loop VBR mode (quantizer scale 1–31), and with fixed GOP size, e.g. 12 frames (see Fig. 2 upper left corner, where the pre-process tools are shown schematically with input and output files). In addition, each of these files must be traced to produce 31 frame size trace files. This process is performed only once for each media file, and the trace files can be used over and over again by new network simulations. The required tools are one encoder and one trace file generator. Since Evalvid-RA 1.0 builds on EvalVid v1.2, the codec choice was limited to MPEG-4 encoders. (However, the current EvalVid 2.0 supports also H.263 and H.264 bit streams. In principle every codec which can be encapsulated in an MP4-container as defined in ISO/IEC 14496-12 and -14 could be used.) In this paper we have used ffmpeg's MPEG-4 encoder [LGP] configured to produce equally sized GOPs with fixed quantizer scale. The EvalVid v1.2 `mp4.exe` program has been used to produce the trace files.

4.3 The network simulation

The next step is the network simulation as shown in Fig. 2 (upper right corner). In a real network, the flows in progress will naturally consist of flows carrying independent and

different content. In trace file simulation though, it is common to use the same trace files simultaneously as media input for many or all sending nodes. If the starting *position* inside the trace files is decided randomly and independent for each source, and if the trace files are big enough, this will approximate independent and different sources. In addition, the flows starting *time* in the simulation can also be randomized. This solution will also be used here, but in addition, each source is running independent rate controllers, and these can be pre-set to different target bit rate averages. As they in addition will react on independent network load feedback while running independent rate controllers, it can be concluded that the approximation of independent source modeling is even more valid in our case. Also, in Section 6.4 and 6.5 we will show that different VBR rate controlled media genres give almost the same traffic characteristics. This is confirmed in earlier work, e.g. [HRR97].

To improve the simulator performance, the trace files are read into memory only once to avoid frequent accesses to external files. In our case we have 31 trace files, with equal number of frame size traces. All files are read into memory during simulation initialization, and organized like a matrix (also stored in `frame_size.dat`), similar to the simple depiction in Fig. 1. Along one of the axis is the frame number count (time), while on the other axis is the quantizer scale 1–31. Through simple indexing, the source nodes can start at a randomized frame number count, while the independent rate controllers (explained in Section 5.2) calculate the GOPs quantizer scale that is used as index along the other axis.

Typically, one of the sources is selected as the *primary* flow (Fig. 2, S0–D0 path), i.e. the flow that will be included in the post-process (see Fig. 2 lower part), where the received media will be decoded and perceptual quality like PSNR and MOS values are calculated. This flow must be started at frame number 1, so that the decoded media can be directly compared to the original media, frame by frame. The rest of the flows are used as traffic generators of real rate adaptive media. If desired, more than one flow can be selected as a primary flow, and more than one original media file can be used as source material. In the latter case, using N different original media sources, N matrices of frame sizes must be read into memory in the simulation initialization phase.

To assist the post-process, the quantizer scale used for the primary flow must be logged during simulation time. This information is stored in the senders trace file (`st_be_0` in Fig. 2), together with packet sizes and sending times. Given a simulation time MTU parameter, each frame is typically fragmented into several packets. The packets belonging to the same frame are either submitted back-to-back, or smoothed over one frame interval, eventually smoothed by the TFRC sending buffer (see Section 5.3), decided by simulation time parameters. Received packets are logged at receiving nodes (e.g. `rd_be_0` in Fig. 2), storing packet number, time, size, and if missing (detected by received packet numbering not being sequential), tagged as lost. The `frame_size.dat` will together

with the other simulation time output files support the received media file binary reassembly and decoding to be performed in the post-process.

4.4 The post-process

The main post-processing functionality is depicted in the lower left corner of Fig. 2. Using the trace files generated during network simulation (dashed lines from the right), together with the media files produced during the pre-process (dashed line from the top), several statistics and measures can be calculated from the simulated traffic. As in the original EvalVid [KRW03] with the *ns*-2 interface [Ke04] the following can now be produced:

- loss rate statistics
- delay statistics
- assembly of received compressed media taking packet loss and/or delay into account
- decoding of (possibly) erroneous compressed media
- playing decoded media
- calculate PSNR and/or MOS (decoded media compared to original media)

The first two in the list can be calculated for all flows, while the rest is only available for the primary flow(s). The added functionality, and corresponding challenge, is the assembly of received compressed media. Due to the rate controller, the actual media transmitted is a mix between some or all of the 31 quality variants. Thus, the logging of actual quantizer scale used is a key component, functioning as a pointer to the correct input file. Thus the Evaluate Trace program `et.exe` of EvalVid v1.2 was modified to `et_ra.exe`. It opens all 31 MPEG-4 compressed files for reading, then scans all of them following the size of each compressed GOP and logged quantizer scale, to find the correct start position inside the used MPEG-4 file of every GOP. In this way the correct binary information is copied into the resulting MPEG-4 file, which is the *rate adaptive primary media file* submitted into the network, given the network state feedback at simulation time. Packet losses during simulation will result in corresponding frame loss. The resulting MPEG-4 file will then typically have a varying quantizer scale, but inside each GOP, the quantizer scale is fixed.

A list of the complete Evalvid-RA tools package is given in the Appendix.

5. Adaptive rate controller

Having established the framework guidelines, the online rate controller running at simulation time can now be selected. This rate controller will have very limited input information from the encoder. If assuming connection to a live (online) encoder where low delay is of critical concern, there is no a priori information of the visual complexity

Table 1 *List of terms used in this paper and their respective definitions*

Term	Definition	Units
r	Leaky bucket rate, i.e. the average video rate	bits/GOP
b	Leaky bucket size	bits
$X(k)$	Leaky bucket fullness at time k	bits
$R(k)$	Leaky bucket input during GOP k	bits
$\hat{R}(k)$	Estimate of Leaky bucket input during GOP k	bits
$Q(k)$	static quantizer scale used during GOP k	1–31
$r'(k)$	adaptive Leaky bucket rate used during GOP k	bits/GOP
$b'(k)$	adaptive Leaky bucket size used during GOP k	bits
G	GOP size	frames
r_{new}	current network update of rate	bits/GOP
r_{old}	previous network update of rate	bits/GOP
$\bar{r}$	averaged leaky rate used for TFRC	bits/GOP
r_i^t	partial TFRC rate feedback (number i of N)	bits/GOP
B_i	TFRC sender buffer backlog at feedback i	packets
d_f	decay factor used for forcing sender buffer to drain	
b''	adaptive Leaky bucket size used for TFRC	bits

of the next frame or GOP. The actual number of bytes spent per frame can however easily be monitored, using the information from the input trace files (depicted as `st_*.txt` in Fig. 2). The rate controller constraints are thus target average bit rate, the bit rate variability allowed, plus a possible peak rate limit, all which can be calculated by the rate controller itself.

5.1 Shaped VBR (SVBR) — A compelling candidate

The two first constraints can efficiently be controlled by a leaky bucket. Leaky buckets in different variants are also commonly used by most offline and online rate controllers. When searching the literature, the Shaped VBR (SVBR) by M. Hamdi et al. [HRR97] is a compelling candidate, since it is of low complexity, and also designed to work on GOP granularity. Their paper stress however that the quantizer scale q producing the average

target bit rate r should optimally be known a priori. We found that this requirement could be relaxed without having significant impact on performance, see Section 6.2.

The SVBR is using a leaky bucket $LB(r,b)$ where r is the target average bit rate and b is the bucket size (see Table 1 for paper variables overview). The larger the bucket size, the more rate variability is allowed, producing a more stable quality [LOR98]. The media packets do *not* experience additional delay because the $LB(r,b)$ is used as a virtual buffer, meaning that the packets go straight into the network (or network sending buffer as in TFRC), but is counted in parallel by the $LB(r,b)$. The latter makes it very suitable for interactive communication. The leaky bucket fullness $X(k)$ is calculated at the start of every GOP k as [HRR97]

$$X(k) = \min(\{b, (\max(\{0, X(k-1) - r\}) + R(k-1))\}), \quad (1)$$

where $R(k-1)$ is the actual bits spent during GOP $k-1$. When $X(k)$ is close to zero, the rate control algorithm behaves as in open loop, i.e. with the quantizer scale $Q(k)=q$. When it is close to b, however, it behaves more like CBR, i.e. $R(k)$ is attempted to be close to r. The quantizer scale $Q(k)$ is then calculated as

$$Q(k) = Q(k-1)R(k-1)/\hat{R}(k) \quad (2)$$

assuming that the scene complexity changes slowly from GOP to GOP (i.e. it follows a predefined rate-distortion curve), and $\hat{R}(k)$ is an estimate of the bits to be spent during GOP k. When the scene complexity increases substantially, (2) will calculate too small $Q(k)$, giving too high $R(k)$. This will be "compensated" for in the GOP $k+1$. For pre-stored media and live media allowing a delay equal to one GOP, the next GOP scene complexity will be known a priori, and such bit rate over-shoots can be avoided. For more details on how to calculate $\hat{R}(k)$ we refer to [HRR97].

5.2 Rate Adaptive SVBR (RA-SVBR): the needed modification

Although SVBR was designed for static values of r, b, and q, we have found that r and b actually can be variables influenced by network state feedback. Using r and b as upper limit values used when the network is in non-congested state, $r'(k) < r$ and $b'(k) < b$ can be calculated whenever the congestion control algorithm suggests a new allowed average bit rate r_{new}. Since these events are *not* synchronized to the GOP periods, (1) must be modified to take this into account. Scaling the bucket size $b' = br'/r$ (the time index k is omitted in the time varying r' and b' from now on), and letting $i \in [0,G-1]$ being the time index for the network feedback event counted as the position in the active GOP of size G frames, the equation becomes

$$X(k) = \min(\{b', (\max(\{0, X(k-1) - r'\}) + R(k-1))\}), \quad (3)$$

where $r' = (r_{old}i)/G + r_{new}((G-i)/G)$ and $b' = b_{old}(i/G) + b_{new}((G-i)/G)$. If there is no network feedback during a GOP, $r' = r_{old} = r_{new}$ and $b' = b_{old} = b_{new}$. We have named this SVBR modification *RA-SVBR*. Fig. 3. depicts an overview over RA-SVBR local operation and its interface towards the live network feedback (right) and media encoder trace files (left).

The major limitation of a GOP based rate controller is that the new rate might be delayed up to the time duration of one GOP before effectuated, depending on the time the network feedback event occurs relative to the local GOP period. The result might be packet drops in the network due to traffic overload. However, (3) makes sure that the bit budget is corrected in the next GOP period. A more complex rate controller could take advantage of the possibility of changing the quantizer scale parameter one or multiple times during a GOP, as discussed in Section 4.2.

The major advantage of the rate adaptive version of SVBR is that *any* $r' < r$ can be supported, provided that the quantizer scale needed is within its upper limit. One is not restricted to supporting only the 31 discrete quality variant bit rates — the rate controller ensures that any $r' < r$ can be supported, when averaged over some few number of GOP periods. It is important to note this, since this makes a significant difference to multirate coding where typically only 3–4 different rates are supported.

Very small r' forces the rate controller to select large quantizer scale values. The general video quality when using the very highest quantizer scale values is not very good — visible blocking artifacts show up. In addition, since the quantizer scale value is upwardly bounded to 31, an arbitrary small r' can not be supported. Thus, $r_{min} < r' < r$, where r_{min} is dependent on the current scene complexity.

In practice, this means that also other rate scaling techniques could be considered, such as lowering the frame rate and/or reducing the spatial resolution. Such changes can be supported by signaling repeated headers to the receiver, giving new values for these parameters to the decoder. The simulator implementation could also take lower bounds on quality into consideration and alternatively terminate a session if the allowed throughput is too small. Such information could of course also be used as input to admission control systems in order to prevent starting new flows when available bandwidth is too small. All these features are on the priority list for future Evalvid-RA updates.

5.3 Supported network feedback systems

In general, any congestion control algorithm can be supported. For best possible stability and link utilization, an average rate limit should be calculated and used as r'. In this paper, two different congestion control mechanisms are tested and compared using the Evalvid-

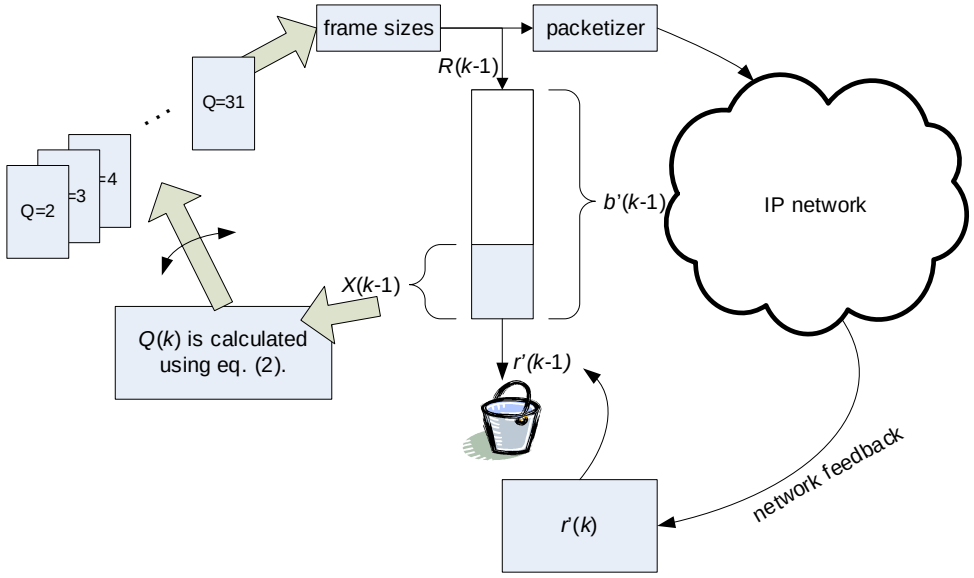

Figure 3 *RA-SVBR with the updates from the network and its selection of frame size information from the available trace files (eventually real frames from online coder in a real implementation).*

RA tool-set: TFRC [FKP06] and P-AQM+ECF [LAR05], where the latter is a proprietary solution where more accurate network state information is exploited.

The two methods differ significantly. Whereas TFRC relies on either packet drop statistics or ECN tagging (e.g. from RED routers) observed at receiver and signaled back to sender using acknowledgment packets, P-AQM+ECF uses explicit packets with congestion state information based on both input rate and queue statistics directly from each P-AQM enabled router on the path. Furthermore, TFRC requires each packet to be of similar size (TFRC is packet rate oriented, not bit rate — TFRC-SP is another TFRC profile where packets per second is constant and size per packet is a variable, to better suite VoIP applications [FK07]), while P-AQM+ECF do not impose any such limitations. Since the output from the VBR encoder can not be guaranteed to produce frame sizes that can be fragmented into integer number of packets, byte stuffing has to be used where the actual packet size is less than the fixed TFRC packet size. Clearly, this is bandwidth waste. Even further, TFRC uses strict traffic shaping, in that the TFRC rate is the maximum rate: packets in the transmit queue are submitted at the TFRC packet rate, as long as there are packets in the queue. In P-AQM+ECF, the packets are submitted directly into the network without any traffic shaping. The benefit with the latter approach is no additional transmit buffer delay, while the disadvantage is much more bursty traffic. However, as will be shown by simulations in Section 6.4, there is no significant LRD (Long Range Dependency), so the router buffer occupancy should be controllable.

P-AQM+ECF calculate r' directly, and is interfaced to adaptive SVBR by just passing this value. However, since TFRC is using a transmit buffer, there is a need for a small modi-

fication to (3) to ensure that the buffer queue is kept reasonably small. The coupling between the TFRC packet rate and adaptive SVBR is therefore decided to be given as

$$X(k) = \min(\{b'', (\max(\{0, X(k-1) - \bar{r}\}) + R(k-1))\}), \quad (4)$$

where $\bar{r} = 1/N(k-1)\sum_{i=1}^{N} r_i^t e^{-B_i/d_f}$, r^t being the TFRC rate calculated as bytes per GOP and B_i is the instantaneous TFRC transmit queue backlog at the TFRC rate feedback events and $b'' = b\bar{r}/r$. The averaging operation in (4) is necessary due to the fact that TFRC feeds back N updates per GOP. The term e^{-B_i/d_f} with the decay factor $d_f = 100$ ensures that the queue backlog is drained over time. A smaller decay factor than 100 would have drained the queue faster, but we observed that the TFRC feedback system became unstable (queue oscillations bigger and bigger). We also simulated with $d_f = 1000$ to show increased stability at the cost of some increased shaping buffer delay.

6. Example Evalvid-RA simulation and results

To demonstrate the capabilities of Evalvid-RA some simulation examples are described and the results are discussed in this section. The ns-2 simulation model runs the RA-SVBR source and a dumbbell network topology providing feedback as depicted in Fig. 3. The actual video sources used are described in the next section.

6.1 Test sequences and the Evalvid-RA pre-process

The video clips for the initial simulations were selected from the official MPEG test clips. This way, our results can be verified by independent researchers. A 1836 frame video sequence was created using a collage of the clips (in given order) News, Football, Akiyo, Stefan, and Paris, at CIF resolution and 30fps (giving a duration of 61.2s). These clips can be downloaded from e.g. http://www.tkn.tu-berlin.de/research/evalvid/cif.html. All sources were using this sequence, however started at different time and frame number (and looped to enable continuous media), thus avoiding traffic synchronization as discussed earlier. The simulation study also covers more elaborate simulations, testing five different IP router architectures. For that study, seven minute long clips from The Matrix (genre “Action movie”, CIF, 29.97 fps) and from “An Inconvenient Truth” (genre “Documentary”, CIF, 25 fps) are used to create even more realistic network traffic. The latter media can also be considered as advanced videoconferencing content, in that there are shots with text, presenter in front of slides with computer graphics, and some shots with natural image content.

Following the Evalvid-RA pre-process these sequences were first compressed with ffmpeg 30 times (static quantizer scale values ranging 2–31 are supported by ffmpeg). The GOP size was fixed to 12 frames with B-frames turned off to avoid algorithmic codec delay[1]. Then the `mp4.exe` trace tool was used on each of these MPEG-4 files to produce

the ASCII trace files giving the compressed frame size and type. These 30 trace files were used as basis for the Evalvid-RA *ns*-2 traffic generator `vbr_rateadaptive.cc` in order to produce realistic video traffic, where each frame is fragmented into MTU sized packets before submission. Note that an optimal packetizer would fragment frames to packets at macro block boundaries – similar to the *slices* defined in H.264 – to enhance error resilience. This approach is not possible in the current version of Evalvid-RA since the trace files from the `mp4.exe` tool are generated with frame size granularity, not macro-block or slice granularity.

6.2 Adaptive SVBR performance vs. ffmpeg's VBR controller

As a first validation of the implementation, a comparison of the RA-SVBR and ffmpeg's rate controller was performed. Using the MPEG test sequence, both RA-SVBR's *r*-parameter and ffmpeg's own 1-pass VBR rate controller (using the -b switch) were set to 600kbit/s. *b* in RA-SVBR was set equivalent to 1.5 GOP size in bytes. We noted that ffmpeg's rate controller used some GOPs before stabilizing the rate output, at start it was a bit too high. RA-SVBR was simulated in *ns*-2 using the MPEG sequence produced as described in Section 6.1. There were no bandwidth bottlenecks and network feedback reading was turned off. This ensured that the RA-SVBR rate controller was working at $r' = r = 600$ bit/s fixed during the whole session. The resulting *ns*-2 trace files were used as input to the `et_ra.exe` tool for MPEG-4 file assembly. This file and the file generated by ffmpeg were now decoded with ffmpeg to raw YUV files. These two YUVs were compared to the original MPEG test sequence to produce the PSNR results, which are shown in Fig. 4a). There is only a minor difference in performance. Not surprisingly, the ffmpeg's own rate controller produces the best result, since it can vary the quantizer step from frame to frame, and even macro-block to macro-block, and not only from GOP to GOP as in RA-SVBR. Inspecting the figure more closely, one can see that the I-frames have significantly better PSNR (about 1.5–2.0 dB), while the P-frames have almost the same PSNR. This is achieved by lowering the quantizer scale of the I-frames, thus producing a better I-frame which is also a better key-frame for the motion estimation of the following P-frames. Nevertheless, this comparison proves that the quantizer scale adjustments made by RA-SVBR and its implementation follow the proposed performance as given in [HRR97].

Fig. 4 a) – c) also show the different complexity of the clips compromising the MPEG test sequence: News (frame 1–300) is medium, football (300–400) high-motion, Akiyo (400–700) is very low complexity, thus the PSNR values get very high, Stefan (tennis player, 700–800) is very complex, thus giving very small PSNR values, and at last Paris (800–1836) which is high to medium.

1. B-frames are however fully supported by Evalvid-RA.

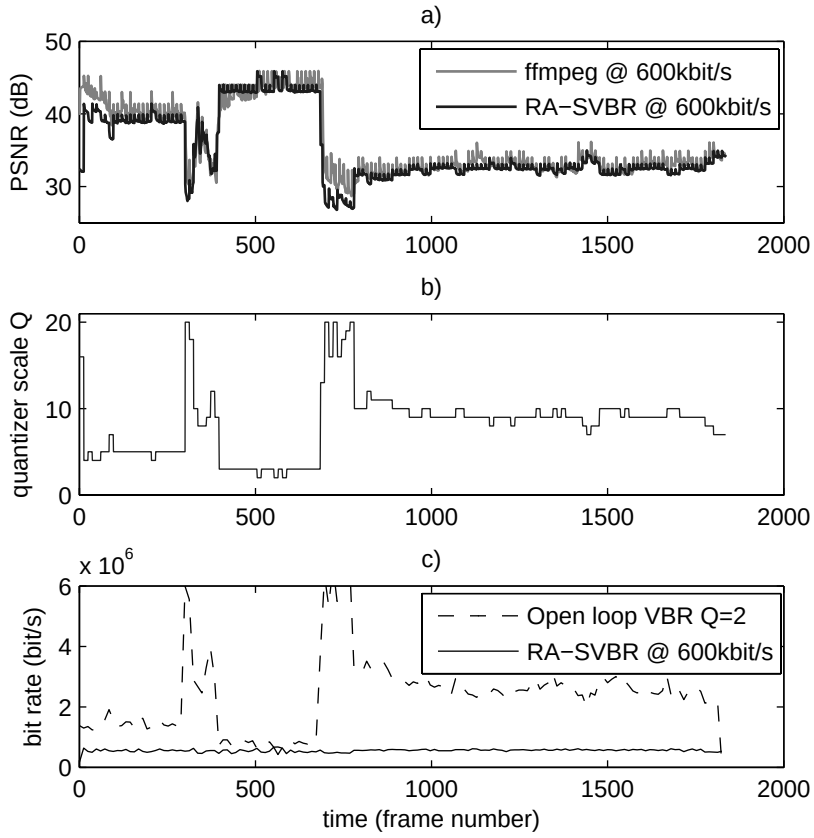

Figure 4 Comparison of PSNR values of RA-SVBR and ffmpeg's rate controller in test sequence. b) The quantizer scale values Q used by RA-SVBR in test sequence in a). c) The bit rate of Q=2 VBR and RA-VBR at 600kbit/s.

6.3 TFRC and P-AQM initial performance comparison

In this section, a simple scenario with VBR traffic only is tested using the MPEG sequence, in order to address behavior specific for homogeneous video traffic and network characteristics. The TFRC streaming media flows are routed through a network with either ordinary FIFO or RED routers with ECN enabled, while adaptive UDP is streamed over P-AQM with ECF signaling. A simple dumbbell network topology was used. The bottleneck link capacity was 40 Mbit/s with a propagation delay of 10 ms. The access network capacities were 32 Mbit/s [1] with 5 ms delay (each side of the bottleneck link), thus producing a total one-way propagation delay of 20 ms. 64 media sources were started at random time, uniformly distributed over the first 16 s period of simulation time, but all ended simultaneously at 64 s. The only exception was the primary flow that started at 10ms and ended at 61.21 s. Each source had a target RA-SVBR average bit rate set to

1. to make sure that the access network does not cause any form of queuing

r=1.0 Mbit/s. The fair share bandwidth after all sources have started was however $40Mbit/s/64$=625 kbit/s. The challenge of the network congestion control and the rate adaptive SVBR was then to make the sources produce 625 kbit/s on average (after 16 s, packet headers included), ensuring bandwidth fairness and smallest possible delay between sender and receiver. The end-to-end delay budget includes sender buffer (TFRC only), packet transmission delay, propagation delay, and network router queuing delay. MTU was set to 1036 bytes for the TFRC case, and 1028 bytes for the P-AQM case. These numbers resulted from 1000 byte payload, 20 byte IP header, 8 byte UDP header (P-AQM) and 16 byte DCCP/TFRC header [KHF06]. In a real implementation the RTP protocol could have been used additionally — this would have added typically 12 bytes. The RED router (used by the TFRC simulations) was configured to gentle adaptive RED with the target delay set to half of the maximum queue buffer size. The buffer size was set equal to the bandwidth-delay product (BDP) assuming an RTT of 200 ms[1], which gives 0.200 x 40e6 / 8 = 1 MB, i.e. approximately 1000 packets (assuming 1000 byte packets). The RED target queue equilibrium was thus about 500 packets. Smaller queue equilibrium was also tried but resulted in severe queue length instability. P-AQM, which is designed to control aggregate traffic with small persistent queue sizes, was configured to a target queue size of only 50 kB. Both RED and P-AQM were run in byte count mode. Transmitter (encoder) frame discard as additional rate control was not allowed.

Table 2 *Ns-2 simulation results*

Sim. #	Cong. Control	d_f	Utiliz. (%)	P. drop (%)
s1	P-AQM+ECF	—	88.2	0.0
s2	RED/ECN+TFRC	1000	89.9	0.0
s3	RED/ECN+TFRC	100	90.0	0.0
s4	RED/ECN+TFRC	40	89.6	0.0
s5	FIFO+TFRC	100	92.0	1.1

Table 2 lists the simulations and their parameters and results, showing ~ link utilization and zero loss for all the simulation cases, except s5 which uses packet drops to signal congestion.

1. The RED router must be set to cope with typical average RTT of the flows traversing it, and not the special case with low RTTs as in this example; this also makes it more robust to handle many flows, see e.g. [CC03].

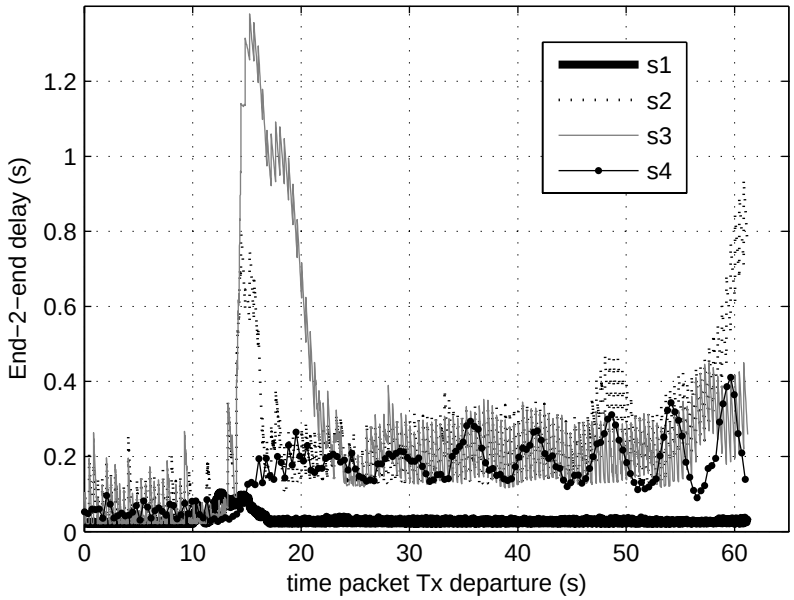

Figure 5 *The packet delay end-2-end for the primary flow, including traffic shaping buffer, transmit delay, propagation delay and router queue delay.*

Fig. 5 shows the end-to-end delay for simulation cases s1–s4. S1 (P-AQM) has very low delay, at equilibrium it is below 30ms. The TFRC simulations s2–s3 show that there is a significant period in which the delay is very high. An inspection shows that this delay is both due to excessive queue delay and significant shaping buffer backlog. S4 shows that a too low decay factor leads to unstable behavior. We believe, the reason is that the stable packet submission of TFRC is discontinued by the completely drained shaping buffer. The TFRC "Fast Restart" functionality, which should assist in stability for self-limiting sources, was however enabled.

The Evalvid-RA post-processing tools for the primary flows were now used to generate PSNR and MOS values for s1–s3, given three different delay constraint scenarios: (i) no delay constraint, (ii) receiver play-out buffer size constraint (PoB), and (iii) absolute play-out buffer time constraint (PoD). In (i) all received packets were used in the frame assembly process (by `et_ra.exe`), while in (ii) packets were dropped if the packet inter-arrival jitter was higher than a specified receiver play-out buffer size could tolerate (due to memory limitation). In (iii), an absolute play-out time was specified relative to the frame transmission time, due to the real-time constraint. We tested the simulated scenario with 150 ms and 500 ms equivalent play-out buffer size constraint and 150 ms absolute play-out delay constraint. The latter is reflecting the recommended one-way delay for conversational media. Fig. 6 displays the results for s1 and s3. Since these scenarios had zero loss (due to the ECN and ECF signaling), the PSNR values reflect two other QoS parameters: bandwidth and delay. Bandwidth fairness can be examined by calculating per

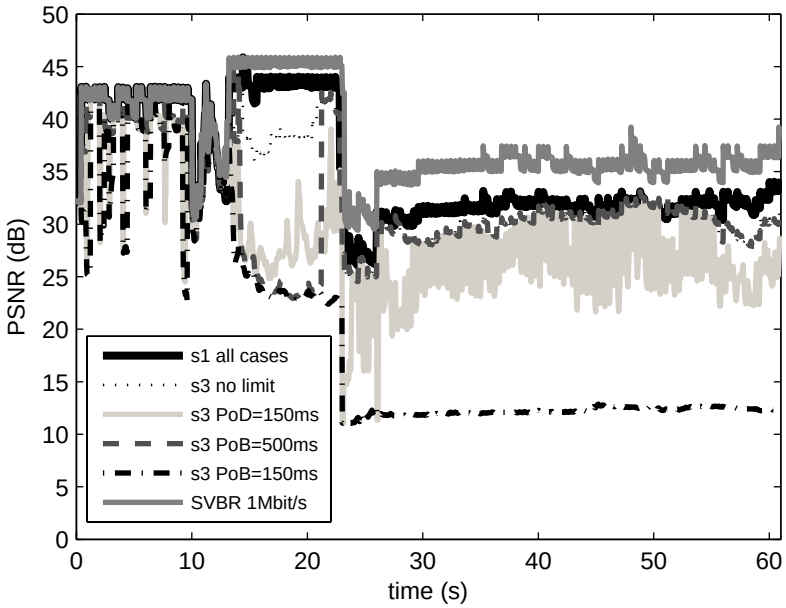

Figure 6 *The resulting PSNR values (frame by frame) of the primary flows in s1 and s3 simulation, given the different delay constraints.*

flow bandwidth, using e.g. Jain's Fairness Index [JCH84]: it was better than 0.99 for all simulations (1.0 is perfect fairness), showing that the bandwidth was fairly distributed over the flows. P-AQM performs best at all tests, due to its superior end-to-end delay performance, both in receive frame jitter and absolute delay. However, the delay caused by TFRC's traffic shaping buffer and RED router affected the perceived quality of TFRC in terms of objective PSNR values. Constraint (i) gives almost the same PSNR as P-AQM, while (iii) shows that the absolute delay of 150ms results in PSNR degradation in the order 1–10dB. This degradation is due to the fact that the decoder has to render the last successfully decoded frame when the current frame number is not yet arrived at the receiver. In case (ii), a PoB of 500 ms is sufficient to handle most of the inter-arrival packet jitter to avoid too much PSNR degradation, while with a PoB of 150 ms, a lot of frames will be dropped due to buffer limitation so that decoding collapses. Statistical delay and PSNR values for the tests s1–s3 are shown in Table 3, with corresponding average MOS values shown in Fig. 7.

Table 3 *Evalvid-RA post-processing results*

Sim. #	avr. delay (ms)	max delay (ms)	avr. PSNR (dB)				frames slipped / total frames
			PoB = ∞	PoB = 500 ms	PoB = 150 ms	PoD = 150 ms	
s1	30.3	104.0	35.0	35.0	35.0	35.0	0/1836

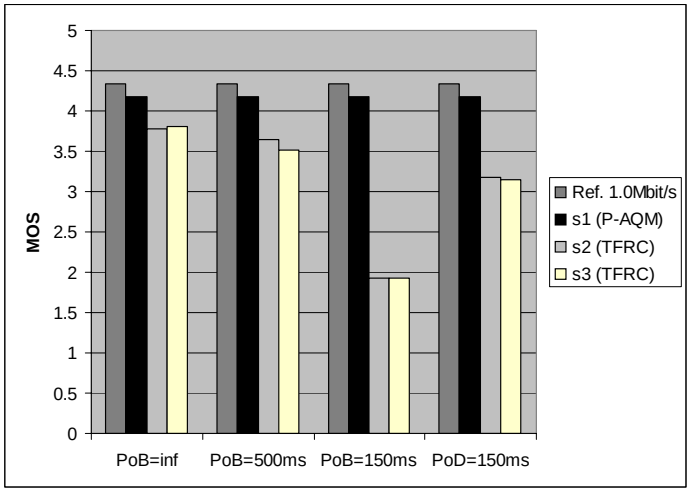

Figure 7 *Average MOS values calculated from the PSNR values following guidelines in [KRW03, Ohm95]. A reference MOS value is calculated for a 1.0 Mbit/s flow of the same sequence, which would have resulted if there were fewer than 40 flows in the bottleneck.*

Table 3 *Evalvid-RA post-processing results*

Sim. #	avr. delay (ms)	max delay (ms)	avr. PSNR (dB)				frames slipped / total frames
			PoB = ∞	PoB = 500 ms	PoB = 150 ms	PoD = 150 ms	
s2	212.1	939.4	33.7	32.8	19.1	29.2	1259/1836
s3	231.2	1379	33.1	31.5	19.3	28.8	1250/1836

This subsection has demonstrated that the perceptual quality of interactive video flows is not only a function of bandwidth and packet drop ratio, but also on end-to-end delay. The network feedback systems are shown to cooperate closely with the adaptive rate controller so that the aggregate traffic gives link utilization close to capacity while packet drops are limited. Due to the inherent TFRC traffic shaping, it is probably natural that this non-bursty traffic can be strictly controlled. The non-traffic shaped traffic output of the P-AQM+ECF system is however not so evident since it submits the VBR traffic directly into the network. The reason why this works well is examined in the next subsection.

6.4 Adaptive VBR rate control avoiding LRD

In [GW94] and also later work by others it was proven and demonstrated that VBR video traffic exhibits long range dependence (LRD). LRD traffic characteristic means that the resulting rate (measured in bytes per frame or per GOP) occupied by the VBR coder varies significantly and that its ACF (autocorrelation function) has significant values for large

lags n, i.e. the ACF $\rho(n) \propto n^{-\beta}$ as $n \to \infty$ and $0 < \beta < 1$ (compared to the exponential fast decay $\rho(n) \propto \alpha^n$, $n \to \infty$ and $0 < \alpha < 1$, valid for Poisson sources). With other words, the VBR coder traffic output has a self-similar behavior. Obviously, such traffic makes it very difficult to have high link utilization without risking periods with persistent packet losses due to queue buffer overflow. However, the work cited did statistical analysis of *VBR open-loop* coders only, i.e. no rate controller was present at all. Applying VBR rate control means that an average bit rate is established, possibly also with variance constraints. This is exactly what is gained by adaptive SVBR in the form of (3) and (4). [HRR97] also shows that the rate controller almost completely eliminates any LRD, i.e. it becomes more like SRD (short range dependent). *This is why the deployment of VBR rate controllers makes high link utilization obtainable, since the aggregate of SRD sources will exhibit Poisson characteristics.* When scaling both the r and b of the leaky bucket in SVBR, variability is also reduced per source to adjust to the potentially increased variance of the aggregate. Thus, congestion control combined with *adaptive* rate controllers makes way for even more flows and stabilizes the network throughput at high utilization. The accuracy of the feedback system and buffer dimensioning then determines if this can be accomplished with small buffer delays.

An Evalvid-RA ns-2 simulation was carried out to substantiate the claims made above. It was similar to the P-AQM simulation described in the previous subsection, except that it was run over 300 seconds to get more data for the statistics. All flows were looped back to the beginning of the trace files when finished, except for the primary flow that stopped at 61.2 s as before. In Fig. 8 the primary flow rate is shown together with flow 4 and 5.

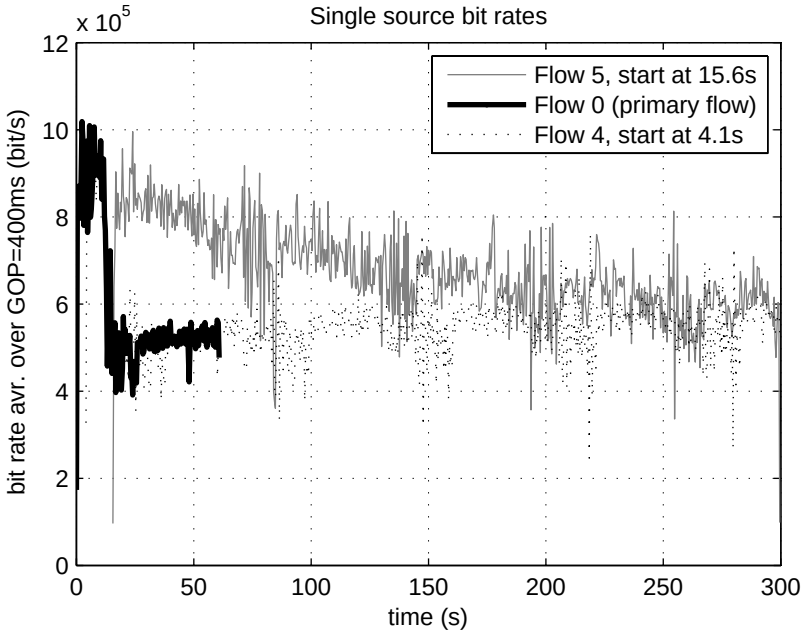

Figure 8 *Three of the 64 flows, showing the VBR behavior, and the adaptive rate control slowly adjusting the rate to the 600kbit/s fair application rate.*

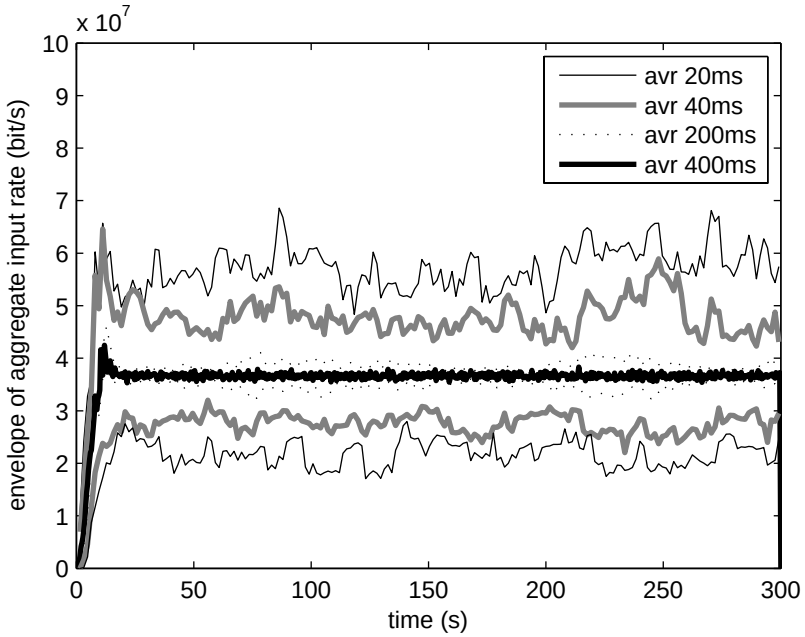

Figure 9 *Averaging at larger and larger time scales reveals a stationary time series.*

Note that since flow 5 starts at 15.6 s, it is one of the last flows to start in the 0–16 s starting period, thus its convergence against the fair bandwidth share is slower than "normal" (e.g. compared to flow 4). This plot shows that it takes some time before the flows become stationary. However, the *aggregate* of the flows entering the bottleneck router has a stationary behavior much sooner, as shown in Fig. 9. The reason for this is that the congestion control of P-AQM works on the aggregate, while the AIMD behavior of the sources themselves control the fairness issue. Here the aggregate bit rate has been calculated using four different averaging time units: 20, 40, 200, and 400 ms (= GOP). As shown by the curve for GOP sized averaging rates, stationary behavior is obtained already at approximately 20 s. Its variability at smaller time scales is much higher. However, the figure shows that the averaging operation reduces the variance considerably, which is typical for Poisson and Poisson-like traffic aggregates. Calculating a histogram of the packet inter arrival times (Fig. 10) reveals that the traffic is indeed Poisson-like, since a negative exponential distribution shape is produced.

The only exception is the spike at 0.27 ms, caused by the frequent event of multi-packet frames arriving back-to-back (1028 B packets in 32 Mbit/s access links have 0.27 ms spacing). Calculating the autocorrelation function of the bit rates as shown in Fig. 9, gives the results as shown in Fig. 11.

With 400 ms average time windows, the sequence is clearly uncorrelated. At smaller time scales a correlation peak at the lag corresponding to 61.2 s is evident. This is not surprising as the flows repeat themselves after this amount of time. This is a result of a "synthetic" aggregate behavior and motivates a modification of Evalvid-RA to jump to an

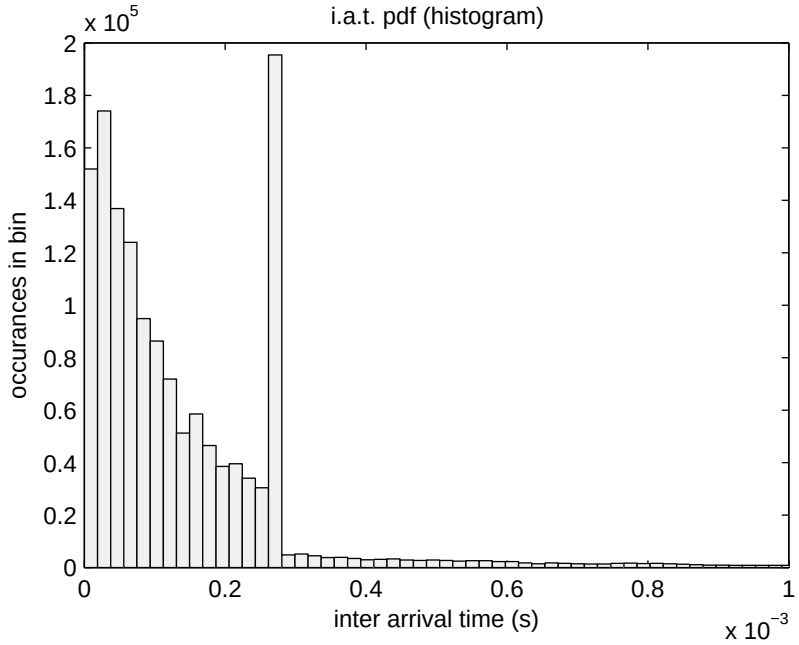

Figure 10 *The histogram of the inter arrival time of packet received at bottleneck router*

arbitrary GOP after ending the trace file instead of jumping to the very beginning. Nevertheless, the envelope shape reveals that the ACF converges fast to zero at increasing lag, as is the nature of Poisson-like traffic sources.

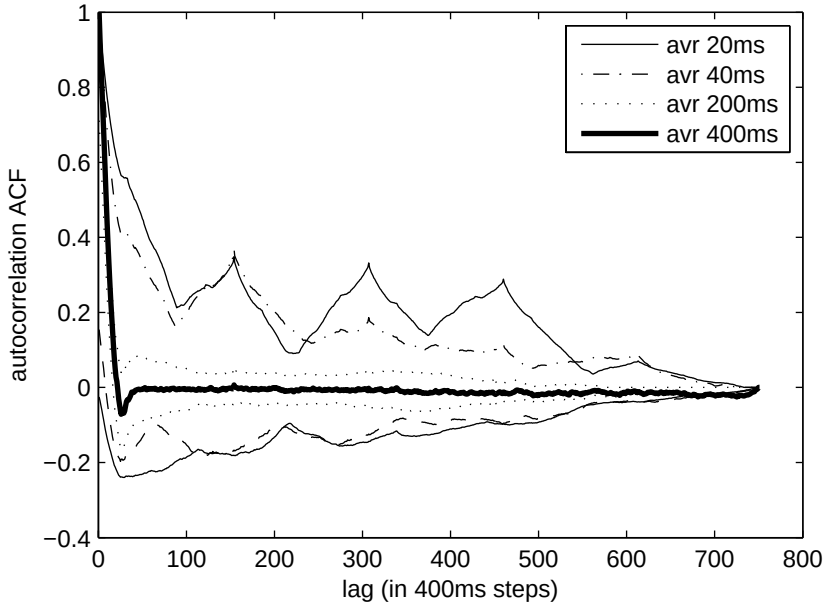

Figure 11 *The envelope of the autocorrelation function of aggregate input traffic to bottleneck router, calculated at four different time units. Lag units are scaled to fit corresponding time unit.*

It is the near Poisson traffic nature that makes it possible to control an aggregate of VBR rate controlled video streams close to full link utilization, with zero packet loss and very small queue delay. As future bottleneck router capacity increases, higher link utilization is obtainable without adding delay, possibly even decreasing the delay at the same link utilization.

6.5 Mixed VBR and TCP traffic

In this section we aim to demonstrate the Evalvid-RA capabilities in video transmission protocol analysis using more realistic Internet traffic and running a high number of different work loads in order to compare the different protocols and network architectures. The focus is on relative performance, thus we present the results as PSNR values as function of the number of VBR flows.

A common bottleneck link of 40 Mbit/s is shared by 32 long-lived New Reno TCP flows (e.g. continuous FTP download) and 120 sources generating HTTP Web traffic using a recommended model generating Poisson distributed flow arrival times and Pareto distributed flow sizes (with shape factor of 1.35) [KHR02]. The access network capacity is 3.0 Mbit/s, while the rest of the parameters are similar to Section 6.3. In this environment the VBR flows are transmitted. We vary the number of VBR flows from 2 up to 128, using the clips from “The Matrix” and “An Inconvenient Truth”. In addition to P-AQM and TFRC over RED/ECN routers (TFRC 1), we also test TFRC over RED without ECN marking (TFRC 2), TFRC over ordinary FIFO routers (TFRC 3), and non-adaptive

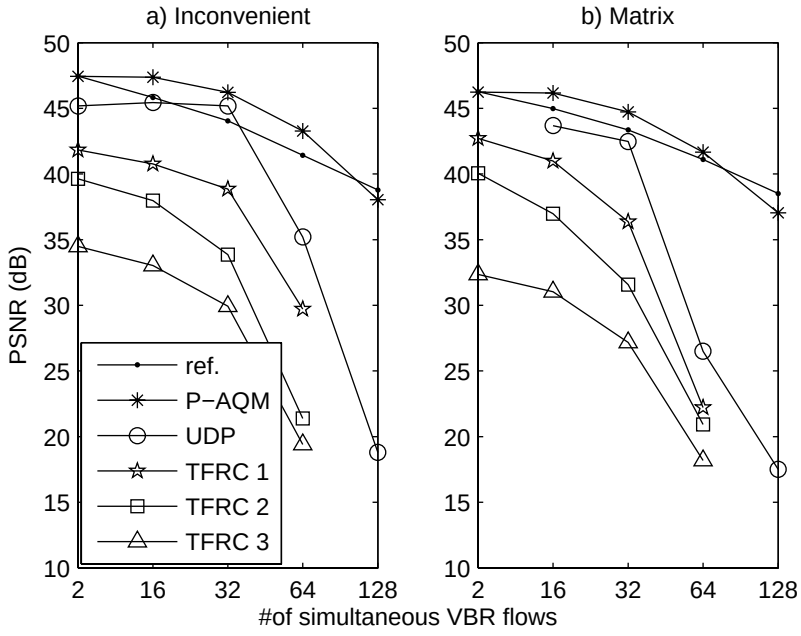

Figure 12 *PSNR values as function of number of VBR flows in mixed network traffic. Play-out delay constraint is 150ms (videoconferencing delay constraint).*

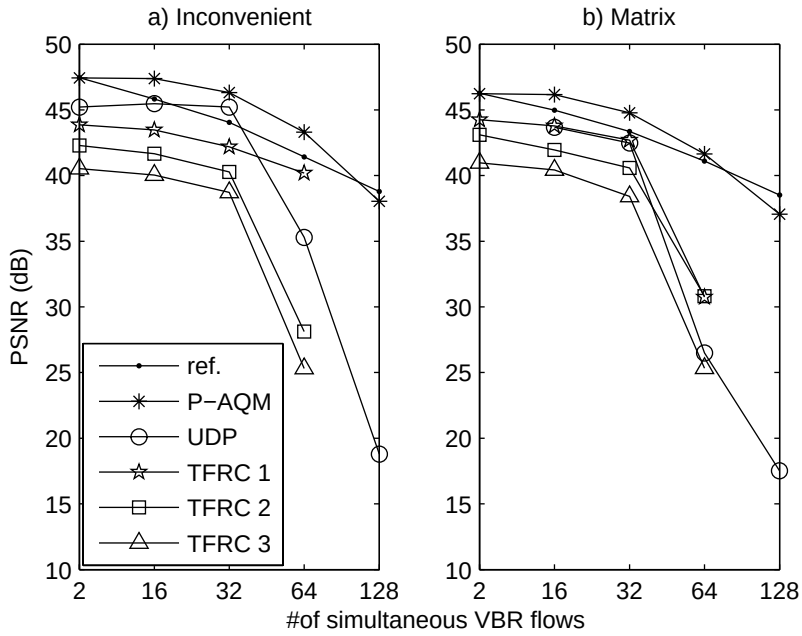

Figure 13 *PSNR values as function of number of VBR flows in mixed network traffic. Play-out delay constraint is 2s (VoD and WebTV delay constraint).*

1.0 Mbit/s UDP flows over FIFO routers (UDP). To obtain reference PSNR values ("ref." curves in Fig 12 and 13), we also simulated a single UDP flow with target bit rates 1.0 Mbit/s, 740 kbit/s, 570 kbit/s, 392 kbit/s, and 240 kbit/s, corresponding to the fair bandwidth share in the different cases.

In Fig. 12 the results for the videoconferencing 150 ms delay constraint case are depicted, while in Fig. 13 the corresponding results for the VoD/WebTV 2 s delay constraint case are shown. The P-AQM performance is equal and above the reference quality, where the reason for the latter is in fact *statistical multiplexing gain* (SMG): since both movie clips have large variations in their bit rate for all quantizer scales Q, and for Q=2 the bit rate range is approx. 0.25–4.5 Mbit/s, there is room for other flows to exploit other flows inability to fully utilize their fair bandwidth share. The flows are upper bit rate limited at 1.0 Mbit/s, explaining the absence of SMG at 2 VBR flows. With 128 VBR flows, the fair bandwidth share is below the minimum bandwidth at Q=2, which also renders SMG impossible. P-AQM is robust also in mixed traffic due to a two-queue scheduler that separates the UDP and TCP traffic and marks the TCP packets with ECN as in RED routers [LAR05]. Non-adaptive UDP streaming also achieves very high PSNR values (but not has high as P-AQM due to packet losses) before it collapses above 32 flows due to very high packet losses. It must be noted that the high performance of non-adaptive UDP is on the cost of starved TCP flows! It is also evident that TFRC performs best when run over ECN enabled RED routers. Performance drops a little when ECN is not supported, while ordinary FIFO queues gives TFRC the lowest performance. The better quality of TFRC at the

2 s delay constraint is due to the fact that most frames do arrive with a latency between 150 ms and 2 s. Again, like in Section 6.3, this delay is a combination of traffic shaping buffer delay and RED and FIFO queuing delay. TFRC also pays a PSNR penalty at any number of flows, in that it has constant packet size, and thus must often use bandwidth-wasting byte stuffing.

All simulated cases had bottleneck link utilization above 99.0%. Packet drops increased with the number of VBR flows: for P-AQM it was in the range 0.001–1%, for TFRC 0.01% with ECN, 0.1–2% without ECN and with FIFO, and ill-behaving UDP 0.6–89%. Jain's Fairness Index of the VBR flows was better than 0.99 for all TFRC and P-AQM simulations. When comparing all long-lived flows, the index was 0.96 or better. Also worth noting is that the results for the two media clips were very similar, demonstrating that VBR rate control reduces LRD and thereby genre differences.

7. Closing Remarks and Conclusion

In this paper we have presented Evalvid-RA, a framework and tool-set to enable simulation of rate adaptive VBR video. Evalvid-RA's main capability is the generation of true rate adaptive MPEG-4 VBR traffic, i.e. the codec output is dependent of the aggregate traffic passing through the network bottlenecks. In addition, the received media traces are used to restore true media files that can be visually inspected and PSNR and MOS scores can be calculated when comparing with the original material. The tool-set includes an online (at simulation time) rate controller that, based on network congestion signals, chooses video quality and bit rates from corresponding pre-processed trace files.

Evalvid-RA's capabilities were demonstrated by the simulation of a VBR rate controller, modulated by TFRC and P-AQM+ECF congestion signals. Up to 128 simultaneous independent VBR sources were run, together with 32 long-lived TCP flows and background Web traffic generated by 120 independent sources. The 420 second network simulation took about 10 minutes to complete on a three year old laptop running ns-2 under Cygwin. Thus, even higher numbers of sources should be feasible.

Statistical analysis of P-AQM+ECF controlled VBR traffic revealed that the traffic aggregate did not exhibit self-similarity. That's why high link utilization and controlled queuing delay and packet loss is obtainable without strict traffic shaping as e.g. TFRC is using. The P-AQM system had both the highest PSNR score and could also support more flows at reasonably high PSNR values. The cost of these achievements is however a new router algorithm (at least located at the bottleneck link) and some additional signaling traffic. The corresponding simulations of TFRC revealed that the performance was increasing with higher network router intelligence. They also showed that delay constraint results were both dependent on the traffic shaping buffer backlog and router queue backlog. Our solution of draining the traffic shaping buffer could probably be improved, e.g. by using

frame discard if the buffer contains more than e.g. 2–3 frames, depending on the application. But then, also other means should be developed that prevent unstable TFRC behavior when using such aggressive buffer draining.

Evalvid-RA can be used as a test tool for new ideas and early implementations. The usage of TFRC for media applications is expected to grow substantially in the coming years and improved performance in real-time applications with strict delay constraint, such as videoconferencing, would make it even more valuable. Obviously, the Internet community prefers simple scalable solutions over new ideas involving e.g. new router architecture as in P-AQM. However, this does not prevent the use of novel architectures in dedicated media networks, such as digital-TV.

More advanced Evalvid-RA usage includes fairness and delay performance tests in scenarios with multiple bottlenecks, heterogeneous RTTs, and scenarios where some sources are self-limited while others are bottleneck limited [Phe04]. Advanced routers with selective packet drop can be tested with new error resilient media features, since PSNR and MOS scores can be calculated in the Evalvid-RA post process. Work on rate adaptive media over wireless networks will be more and more relevant. In fact, such work has already been started at NTNU using Evalvid-RA and ns-2 802.11 models.

Future tool enhancements could include support for audio codecs and more video codecs (such as H.264/AVC, which is already supported by Evalvid 2.0), as well as transmitter frame discard and relaxed quantizer scale constraints. The quantizer scale modulation demonstrated in this paper can in fact be expanded to also include temporal and spatial scalability, perhaps even modality changes, provided that the scaling follows a predefined rate-distortion curve. Ordinary multirate coding can be supported, with trace files resulting from optimized CBR or VBR rate controlled media, with arbitrary quantizer scale values on frame, slice, or even macro block granularity. In fact, this awakens the idea of using the multiple precoded media with fixed quantizer scale (as used in Evalvid-RA to simulate real-time codecs) also as content on real streaming servers, thus enabling streaming media services of pre-stored VoD content with rate adaptation at much finer granularity than ordinary multirate coding. Some sample tests reveal that the additional storage cost is six times that of storing only the highest quality stream, which can be justified with the dropping prices of storage media. In such a way Evalvid-RA could also become not only an analysis concept, but also a concept of implementation, and a bridge in rate adaptive media deployment.

By publishing the Evalvid-RA source code online, we hope that the Internet real-time media research community successfully uses this tool-set to investigate, develop, and optimize adaptive media codecs and network architecture jointly, so that current and future adaptive packet video systems are better suited to handle the varying wired and wireless network capabilities and conditions. The latest version of Evalvid-RA can be

downloaded from `http://www.item.ntnu.no/~arnelie/Evalvid-RA.htm`.

APPENDIX

Listing of the Evalvid-RA tools

Table 4 is included to ease the understanding of what tools are included in the Evalvid-RA download package, their origin, their purpose, and how to use them. Since all tools are command-line based, they are accompanied by sample script files (Linux shell scripts and ns-2 TCL scripts).

Table 4 *The Evalvid-RA tools overview: pre-process, simulation, and post-process.*

Tool	Original Evalvid-RA?	Evalvid-RA script	Purpose
ffmpeg	No	manyQ.sh	To encode video file with the full range of quantizer scale values 2–31.
mp4.exe	No (Evalvid 1.2)	manyQ.sh	Create frame size trace files of all encoded files from previous step.
ns-2: `vibrate adapt.cc`	Yes	concat_ TFRC*.tcl	Simulation: Module running RA-SVR and interfacing the frame size trace files and network feedback.
ns-2: `ra_evalvid_udp.{cc,h}`	Yes (i.e. modification of [Ke04])	concat_ TFRC*.tcl	Simulation: modified udp.ccin where sender trace files are written, including tx time, packet type and Q-value used.
ns-2: `ra_evalvid_udp_sink2.{cc,h}`	Yes (i.e. modification of [Ke04])	concat_ TFRC*.tcl	Simulation: modified udpsink.ccin where receiver trace files are written, including rx time and packet type.

Table 4 *The Evalvid-RA tools overview: pre-process, simulation, and post-process.*

Tool	Original Evalvid-RA?	Evalvid-RA script	Purpose
ns-2: awk scripts	Yes	See `commands.txt`	Sample scripts for simple post-processing of ordinary ns-2 packet trace files.
et_ra.exe	Yes (mod. `et.exe` Evalvid 1.2)	`runPoD.sh` and `runPoB.sh`	Post-process: Re-assembly of the rate adaptive MPEG-4 file sent during simulation time.
fixyuv_ra.exe	Yes (mod. `fixyuv.exe` Eval. 1.2)	runPoD.sh	Post-process: Inserts missing frames due to drop or late arrival so that sent and received video consists of equal number of frames.
psnr.exe	No (Evalvid 1.2)	`runPoD.sh` and `runPoB.sh`	Post-process: Calculate the PSNR.
mos.exe	No (Evalvid 1.2)	`runPoD.sh` and `runPoB.sh`	Post-process: Map MOS values from PSNR.
miv.exe	No (Evalvid 1.2)	`runPoD.sh` and `runPoB.sh`	Post-process: calculate quality indicator for longer sequences.

References

[ALS02] N. Ansari, H. Liu, and Y. Q. Shi. On Modeling MPEG Video Traffics. *IEEE Trans. on Broadcasting*, 48, December 2002.

[BENB06] Horia V. Balan, Lars Eggert, Saverio Niccolini, and Marcus Brunner. An Experimental Evaluation of Voice Qualit over the Datagram Congestion Control protocol. Technical report, NEC Europe, Germany, 2006.

[BSTW95] J. Beran, R. Sherman, M.S. Taqqu, and W. Willinger. Long-range dependence in variable-bit-rate video traffic. *IEEE Transactions on Communications*, 43(234):1566–1579, Feb/Mar/Apr 1995.

[CC03] Jae Chung and M. Claypool. Analysis of active queue management . In *Second IEEE International Symposium on Network Computing and Applications*, pages 359–366, April 2003.

[FF99] S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the Internet. *IEEE/ACM Transactions on Networking*, 7(4):458–472, 1999.

[FK07] S. Floyd and E. Kohler. TCP Friendly rate Control (TFRC): The Small-Packet (SP) Variant. Technical report, IETF RFC4828, April 2007.

[FKP06] S. Floyd, E. Kohler, and J. Padhye. Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC). Technical report, IETF RFC4342, March 2006.

[G06] Eren Gürses. *Optimal Streaming of Rate Adaptable Video*. PhD thesis, The Graduate School Of Natural And Applied Sciences Of Middle East Technical University, 2006.

[GKKW04] James Gross, Jirka Klaue, Holger Karl, and Adam Wolisz. Cross-layer optimization of OFDM transmission systems for MPEG-4 video streaming. *Computer Communications*, 27:1044–1055, 2004.

[GW94] M. Garrett and W. Willinger. Analysis, Modeling and Generation of Self-Similar VBR Video Traffic. In *Proc. of ACM Sigcomm*, London, 1994.

[HRR97] M. Hamdi, J. W. Roberts, and P. Rolin. Rate control for VBR video coders in broad-band networks. *IEEE Journal on Selected Areas in Communications*, 15(6), August 1997.

[ISO94] ISO/IEC 13818-2, Information technology – Generic coding of moving pictures and associated audio information – Part 2: Visual, 1994.

[ISO99] ISO/IEC 14496-2, Information technology – Coding of audio-visual objects – Part 2: Visual, 1999.

[ITS] VQM Software. http://www.its.bldrdoc.gov/n3/video/vqmsoftware.htm.

[JCH84] R. Jain, D. Chiu, and W. Hawe. A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Systems. Technical report, DEC Research Report TR-301, Sept 1984.

[JTC99] ISO/IEC JTC1/SC29/WG11. Information technology – Coding of audio-visual objects – Part 2: Visual, 1999. ISO/IEC 14496-2.

[Ke04] Chih-Heng Ke. How to evaluate MPEG video transmission using the NS2 simulator. http://hpds.ee.ncku.edu.tw/ smallko/ns2/Evalvid_in_NS2.htm, 2004.

[KHF06] E. Kohler, M. Handley, and S. Floyd. Datagram Congestion Control Protocol (DCCP). Technical report, IETF RFC4340, March 2006.

[KHR02] D. Katabi, M. Handley, and C. Rohrs. Congestion Control for High Bandwidth-Delay product Networks. In *Proc. of ACM Sigcomm*, 2002.

[KRW03] Jirka Klaue, Berthold Rathke, and Adam Wolisz. EvalVid - A Framework for Video Transmission and Quality Evaluation. In *Proc. of the 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, Urbana, Illinois, USA, Sept. 2003.

[KT97] M. Krunz and S. K. Tripathi. On the Characterization of VBR MPEG Streams. In *Proceedings of ACM Sigmetrics'97*, Seatle, Washington, May 1997. ACM.

[LAR05] A. Lie, O. M. Aamo, and L. A. Rønningen. A Performance Comparison Study of DCCP and a Method with non-binary Congestion Metrics for Streaming Media Rate Control. In *Proc. of 19th International Teletraffic Congress (ITC'19)*, Beijing, China, Aug–Sept 2005.

[LGP] LGPL. FFMPEG Multimedia System. http://ffmpeg.mplayerhq.hu/.

[Li01] W. Li. Overview of fine granularity scalability in MPEG-4 video standard. *IEEE Trans. Cct. Syst. for Video Tech.*, 11:3:301–317, March 2001.

[LKK04] C. H. Liew, C. Kodikara, and A. M. Kondoz. Modelling of MPEG-4 Encoded VBR Video Traffic. *IEE Electronic Letters*, 40(5), March 2004.

[LMR97] T. V. Lakshman, P. P. Mishra, and K. K. Ramakrishnan. Transporting Compressed Video Over ATM Networks with Explicit Rate Feedback Control. In *Proceedings of the INFOCOM'97*, page 38, Washington, DC, USA, 1997. IEEE Computer Society.

[LOR98] T. Lakshman, A. Ortega, and A. Reibman. VBR Video: Trade-offs and potentials. *Proceedings of the IEEE*, 86(5):952–973, May 1998.

[MS02] Waqar Mohsin and Masood Siddiqi. Scalable Video Transmission and Congestion Control using RTP. Technical report, Department of Electrical Engineering, Stanford University, May 2002.

[MWMM01] M. Miyabayashi, Naoki Wakamiya, Masayuki Murata, and Hideo Miyahara. MPEG-TFRCP: Video Transfer with TCP-friendly Rate Control Protocol. In *Proc. of IEEE International Conference on Communications (ICC2001)*, volume 1, pages 137–141, June 2001.

[Ohm95] Jens-Rainer Ohm. *Digitale Bildcodierung - Repräsentation, Kompression und Übertragung von Bildsignalen*. Springer, 1995.

[Pal04] Paul A. Palumbo. Broadband streaming video: Viewer metrics and market growth analysis 2000 - 2004. Technical report, Accustream Research, 2004.

[Phe04] Tom Phelan. TFRC with Self-Limiting Sources. Technical report, Sonus Networks, Oct 2004.

[RFB01] K. Ramakrishnan, S. Floyd, and D. Black. The Addition of Explicit Congestion Notification (ECN) to IP. Technical report, IETF RFC3168, September 2001.

[RH92] A. R. Reibman and B. G. Haskell. Constraints on Variable Bit-Rate Video for ATM Networks. *IEEE Trans. on Circuits and Systems for Video Technology*, 2(4):361–372, December 1992.

[RHE99] R. Rejaie, M. Handley, and D. Estrin. RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet. In *Proc. of IEEE Infocom*, March 1999.

[Sar] Sarnoff. JNDmetrix. http://www.sarnoff.com/products_services/video_vision/jndmetrix/.

[SW00] D. Sisalem and A. Wolisz. LDA+ TCP-Friendly Adaptation: A Measurement and Comparison Study. In *Proc. of NOSSDAV*, 2000.

[T1.03] T1.801.03. Digital transport of one-way video signals – parameters for objective performance assessment. Technical report, ANSI, 2003.

[UNI03] UNINETT. Digital Brytningstid Uninett 10å r. Technical report, UNINETT, October 2003.

[WBSS04] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, 2004.

[Win05] Stefan Winkler. *Digital Video Quality – Vision Models and Metrics*. John Wiley & Sons, 2005.

[WP02] Stephen Wolf and Margaret Pinson. Video quality measurement techniques. Technical Report 02-392, U.S. Department of Commerce, NTIA, June 2002.

[WWH06] Stephen Wenger, Ye-Kui Wang, and Miska M. Hannuksela. RTP payload format for H.264/SVC scalable video coding. *Journal of Zhejiang University*, 7(5):657–667, May 2006.

[XH06] Lisong Xu and Josh Helzer. Media Streaming via TFRC: An Analytical Study of the Impact of TFRC on User-Perceived Media Quality. In *Proc. of Infocom*, March 2006.

[XLZ05] Changbin Xu, Ju Liu, and Caihua Zhao. Performance analysis of transmitting H.263 over DCCP. In *IEEE Int. Workshop VLSI Design and Video Technology*, May 2005.

[ZKSS03] Michael Zink, Oliver Künzel, Jens Schmitt, and Ralf Steinmetz. Subjective impression of variations in layer encoded videos. In *Proceedings of the 11th IEEE/IFIP International Workshop on Quality of Service (IWQoS'03), Monterey, CA, USA*, pages 134–154, June 2003.

[ZML04] Jian Zhu, Ashraf Matrawy, and Ioannis Lambadaris. Models and tools for simulation of video transmission on wireless networks. In *Proc. of IEEE Electrical and Computer Engineering*, 2004.

Paper F

P-AQM: low-delay max-min fairness streaming of scalable real-time CBR and VBR media

Arne Lie

Published in
Proceedings of IASTED EuroIMSA '08

Innsbruck, Austria, March 17–19, 2008

Paper F

P-AQM: Low-Delay Max-Min Fairness Streaming of Scalable Real-Time CBR And VBR Media

Arne Lie

Abstract

The increase in Internet streaming media deployment and consumption has created a network stability challenge. The reason is that in overload situations, such sources continue submitting packets at an unmodified rate. The elastic applications using the TCP protocol will back-off and receive a throughput below their fair rate. The DCCP/TFRC congestion control mechanism is one possible remedy. However, due to its steady packet rate requirement, the end-to-end delay constraints of conversational real-time media like VoIP and videoconferencing can be ruined by the TFRC transmit buffer and router queue backlog.

In this paper the P-AQM alternative is presented. P-AQM routers provide more accurate traffic load state than RED routers, to ensure high link utilization yet low queue backlog. The paper gives the theoretical stability criteria of P-AQM with explicit congestion feedback (ECF) and explicit rate feedback (ERF). Elaborative ns-2 simulations compare scenarios with a mix of MPEG-4 VBR video and TCP traffic, demonstrating P-AQM robustness and performance. The ECF and ERF max-min fairness is explained and compared to TFRC using both CBR and VBR traffic in a GFC-2 network scenario. The paper also discusses deployment strategies both for the general best effort Internet as well as specialized networks.

KEY WORDS

Congestion Control, Video rate adaptation, Active Queue Management, Control Theory, Max-min fairness.

1. Introduction

The Internet is facing a stability challenge due to the rapid deployment of multimedia services, such as VoD, IPTV, VoIP, and videoconferencing. The main reason for the worry among the Internet research community including the IETF is the lack of congestion control of the media services, since the majority of services use the UDP protocol [Phe07]. Many LAN and WAN operators set up firewalls that prevent unknown UDP services to enter, to avoid starvation of the elastic TCP traffic. The IETF DCCP protocol is a new protocol that is aimed to assist this situation [KHF06]. DCCP includes presently a choice between two different congestion control mechanisms, TCP-like and TFRC (TCP Friendly Rate Control), including fine tuning parameters that are negotiated during the connection-oriented session start-up phase. The DCCP flows are unreliable of nature as UDP flows are, to avoid re-transmissions of lost packets since late arrival packets have limited value for real-time applications.

However, many early-adopters of DCCP report of worse than UDP performance, throughput far beyond fair rate, and broken real-time latency budgets [BENB06, Phe04]. TFRC [FKP06], the most suited DCCP congestion control mechanism for video traffic, is an equation based congestion control method that is designed to give equivalent throughput as if it were TCP, but at a much smoother rate. TFRC uses fixed packet size, and adapts its packet rate based on TFRC feedback packets. Since the media source has variable rate output, a transmit buffer must be located between the media source packetizer and the actual packet sender. This buffer introduces latency, and may cause TFRC to violate real-time latency budgets [LK07]. The coarse network state feedback given by packet losses or ECN marks may result in unfair throughput when the bottleneck is shared by a mix of short and long flows. The unfairness gets even worse if some of the flows are self-limited, such as VoIP [Phe04].

The active queue management system P-AQM (Proportional Active Queue Management) with explicit congestion feedback (ECF) optimizing live streaming performance was previously published in [LAR05], providing an UDP congestion control architecture that minimizes UDP flow latency with a global max-min fairness constraint. P-AQM+ECF provides max-min fairness by exploiting more detailed network state information than in ordinary FIFO or RED/ECN equipped networks. Thus, it belongs to the class of solution where network intelligence has been increased in order to overcome the aforementioned weaknesses of pure e2e (end-to-end) systems like TFRC. Low e2e latency is obtained by low router queue backlog and by allowing rate controlled VBR media injected directly into the network using legacy UDP or RTP protocol, without any sender buffer. It has been shown [HRR97, LK07] that rate controlled VBR media does not exhibit self-similarity, and thus controlled queue backlog at high link utilization ρ can be obtained.

P-AQM is an AQM router that calculates the traffic load, using ECF to signal this information back to the sources. The main contributions of this paper are the presentation of a modified version named P-AQM+ERF (Explicit *Rate* Feedback), a control theoretical analysis of the complete feedback loop of both ECF and ERF systems, and the demonstration of its global max-min fairness capabilities.

The remainder of this paper is organized as follows: In Chapter 2 the related research is outlined. In Chapter 3 the P-AQM with ECF and the new ERF signaling is explained, and a control theoretical analysis is given to show its robustness to varying RTTs, capacity, and number of flows. Elaborative *ns*-2 simulations verify the analysis in scenarios with a mix of different traffic (TCP, CBR, MPEG-4 VBR). The following Chapter 4 motivates and demonstrates P-AQM's max-min fairness capabilities, accompanied by GFC-2 (Generic Fairness Configuration version 2) [Sim94]. Chapter 5 discusses deployment issues before the conclusions are given in the final Chapter.

2. Related Work

Many solutions have been proposed over the last decade to solve the challenge of supporting robust low-latency and fair video streaming in packet switched networks. Among these are VBR over ATM ABR services [LMR97], RAP [RHE99], MPEG-TFRCP [MWMM01], and LDA+ [SW00]. However, there are actually more similarities between P-AQM (with ECF or ERF) and many of the congestion control mechanisms invented for elastic traffic, including ATM ABR services [KJF+00], MaxNet [WAZ03], and especially to XCP [KHR02] and RCP [DKZSM05]. XCP and RCP are designed to overcome the poor link utilization of TCP flows over network paths with large bandwidth-delay product (BDP). The network load is calculated as a weighted sum of the input rate difference to output capacity, and the current queue backlog. Both must be regarded as rather radical modifications to TCP, so co-existence with TCP flows is a challenge. Other more incremental approaches such as Fast TCP [JWL04] and High Speed TCP [Flo03] are generally outperformed, on the cost of more difficult deployment strategies in the current Internet.

Among the new radical approaches, the key issue is how spare capacity can be utilized and max-min fairness can be obtained with as simple as possible architecture in order to provide a scalable approach. The different ATM ABR suggestions did either provide exact max-min fairness with too high complexity, or only approximate max-min fairness with relaxed complexity. MaxNet uses integrated link prices to find the bottlenecks in a simple manner, but fairness is only achieved if all sources use the very same *utility function*. MaxNet, XCP, RCP, and P-AQM all need a multi-bit network state feedback. The one-bit ECN marking, supported by many routers today, can improve TCP throughput to some extent, but is far from sufficient in obtaining max-min fairness.

To the best of the author's knowledge, P-AQM is the first approach that adapts research supporting elastic flows over large BDP, to scalable streaming media. The result is a scalable architecture that minimizes source and network delay, utilizes available bandwidth, and scales the media bandwidth to global max-min fairness. As an example, long-distance VoIP services over P-AQM supported best effort Internet will have equal bandwidth to competing short-distance VoIP.

3. P-AQM: appropriate congestion control for adaptive streaming media

3.1 Media characteristics

Video sources are typically encoded using hybrid DCT transform codecs (e.g. MPEG and WMV) producing I-, P- and B-frames, structured in GoPs (Group of Pictures). Interactive communication uses short GoP periods (typically some hundred milliseconds in duration [JTC99]) for increased error resilience, and B-frames are not used to avoid algorithmic delay. The codecs have basically two different rate control operations: CBR and VBR. While CBR produces constant number of bits per GOP with variable quality, VBR mode produces variable rate at more constant quality. The latter is the natural choice for interactive videoconferencing and VoIP transmission over IP networks, since CBR mode have to include algorithmic "look-ahead" delay in order to fine tune the quantization parameters within each GoP to produce constant bit rate.

Most rate controllers use some variant of the *leaky bucket* algorithm. This means that even if the quantization parameter scale is changed only once per GoP, the leaky bucket *rate* can follow the network feedback information quite accurately. In practice, the result is that the media source can adapt to the suggested rate when seen over a large enough time window (in the order of 0.5–1.0s). The AQMs inside the network *must take this into account*. Even more important is the fact that rate controlled media does not exhibit any Long Range Dependencies [HRR97, LK07]. In sum, the conclusion is that media like video and audio can be VBR encoded and submitted into IP networks without any transmit buffer, and still be rate controlled so that the aggregate of real-time and elastic traffic can have high capacity utilization and controlled low delay.

3.2 P-AQM: more accurate IP network state feedback

To obtain these goals, an efficient communication channel between network bottleneck routers and media rate controllers must be established, which signals accurate network traffic load information. The solution is thus to add some intelligence into the IP network, but in the same time ensuring that the solution is scalable, and works for all kind of traffic (both elastic and real-time).

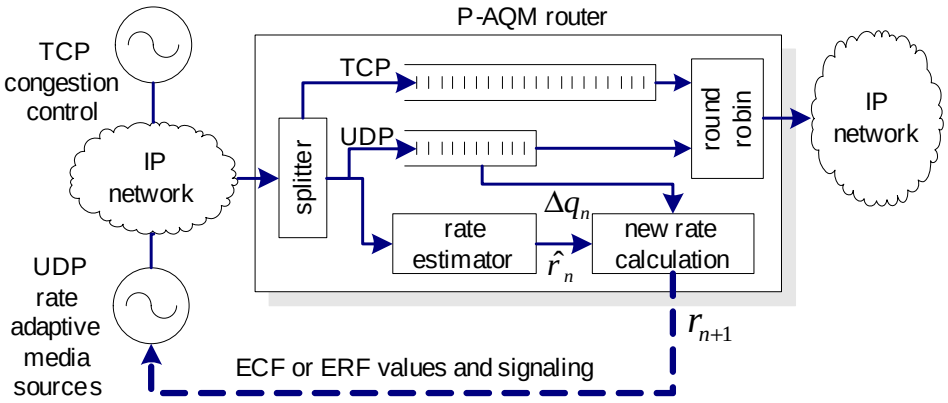

Figure 1 *P-AQM decouples e2e congestion controlled traffic (TCP/TFRC) from non-elastic (UDP) with a two-queue scheduler. The TCP buffer sizing is as ordinary FIFO or RED queue, and has both tail drop and ECN marking. The shorter UDP queue uses P-AQM rate control (input rate estimation $\hat{r}_n$ and queue backlog Δq_n) calculating the feedback r_{n+1}.*

The solution presented in this paper is P-AQM routers that monitor both aggregate input average media (UDP) rate r (bytes/s) and instantaneous queue backlog size q (bytes). Fig. 1 shows that the P-AQM router separates the TCP (and TFRC) traffic from the UDP traffic with a two-queue scheduler, and merge them again with a byte oriented round robin (RR) scheduler. The RR weighs the two queues based on the number of flows, thus obtaining decoupled fairness: TCP friendliness is therefore obtained even if the VBR rate control algorithm does not include this constraint. Since real-time media traffic sources are non-elastic, and also have a significant variance in output rate, the average UDP rate must be calculated over a time window comparable to a typical GoP size (τ_2 in Fig. 2 is a fixed value and equal to 0.5s in this paper). P-AQM is designed to do this. In addition, in its periodic feedback design, it will postpone the start of the next average rate calculation until the transition period (τ_1) from old to new rate is exceeded. τ_1 is actually an estimate of the average RTT of all flows passing through the P-AQM outbound link, calculated by the router itself. P-AQM+ECF acquires this information through the use of ICMP Echo packets between the P-AQM routers and the sources. The new P-AQM+ERF simplify this by using an RCP-like protocol in replace of UDP: RCP carries special header fields for both the rate and the RTT, where the latter is calculated by the sources instead.

In the end of period n (see Fig. 2), P-AQM calculates the wanted input rate for the next period, r_{n+1}, so that either the queue is drained to equilibrium value (Δq=0) if there has been an overload period, or the input rate is increased, if possible, to have a higher link capacity utilization. P-AQM+ECF calculates the *aggregate* rate, and signals either a relative multiplicative decrease MD∈ [0.1,1), or an absolute additive increase AI. In order to calculate the AI, the difference $r_{n+1} - \hat{r}_n$ is divided on the current number of media flows. In order to achieve global max-min fairness (which is covered in detail in

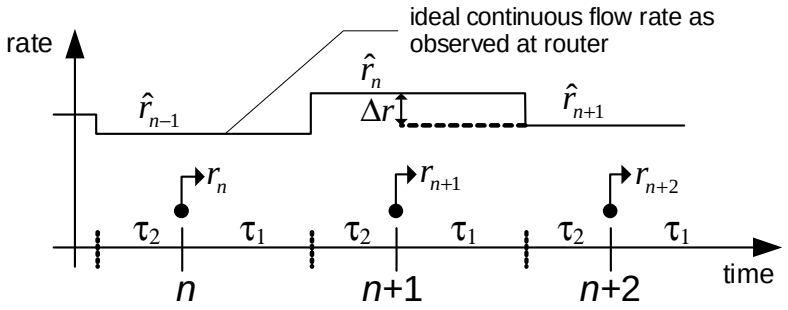

Figure 2 *The relations between estimated rates $\hat{r}_n$, feedback rates events r_n, timing periods n and intervals τ_i, as seen from the router.*

Chapter 4) the MD and AI is signaled explicitly from every router on the path, using ICMP source quench packets. In the revised P-AQM+ERF version the equation is modified to calculate the wanted absolute *flow* rate directly. This simplifies the signaling to obtain max-min fairness, in that this value can be tagged directly into each packet (in its application header part, or in the RCP header), and only updated by a downstream router calculating a lower value. The resulting value received by the destination node is signaled back using RCP ACK packets or RTCP.

3.3 P-AQM rate equations

The P-AQM+ECF rate equation is given as

$$r_{n+1} = \hat{r}_n + \alpha d(c_u - \hat{r}_n)/\tau_2 + \beta \Delta q_n/\tau_2, \tag{1}$$

where d=τ_1+τ_2 (see Fig. 2), c_u is the output link capacity, Δq_n is difference between wanted queue equilibrium N^* and queue backlog size Q_n, and α and β are constants to provide stability. The derivation of (1) from the variant presented in [LAR05] is given in Appendix A, while noting that $\Delta r = r_{n+1} - \hat{r}_n$. By dividing the update term of (1) on an estimate of the number of flows $\hat{N}$, the initial P-AQM+ERF rate equation is given by

$$R_{n+1} = R_n + \frac{\alpha d(c_u - \hat{r}_n)/\tau_2 + \beta \Delta q_n/\tau_2}{\hat{N}_n}, \tag{2}$$

where upper-case R is flow rate, as contrasted to r which is aggregate rate. However, it was discovered (section 3.4) that the stability region became dependent on the ratio between τ_1 and τ_2. To become independent, a better approach is to divide the rate excess and queue backlog terms on d instead of τ_2. The final P-AQM+ERF equation is thus

$$R_{n+1} = R_n + \frac{\alpha(c_u - \hat{r}_n) + \beta \Delta q_n/d}{\hat{N}_n}. \tag{3}$$

$\hat{N}_n$ is directly calculated as the output capacity divided on the calculated rate per flow, as in RCP. Thus,

$$\hat{N}_n = c_u / R_n . \tag{4}$$

P-AQM+ECF stability analysis will be presented first, followed by P-AQM+ERF.

3.4 P-AQM+ECF stability analysis

The derivative of rate change part of (1), i.e. Δr, can be approximated as $\Delta r / d$. Thus

$$(\dot{\Delta r}) \approx \frac{\Delta r}{d} = \alpha \frac{(c_u - \hat{r}_n)}{\tau_2} + \beta \frac{\Delta q_n}{\tau_2 d} . \tag{5}$$

The stability criteria of P-AQM with ECF feedback loop will now be analyzed by applying traditional control theoretical methods, assuming a fluid flow model with continuous P-AQM router input rate $r(t)$. Defining $x(t) = r(t) - c_u$, i.e. $x(t)$ is the excess input rate compared to output capacity, one can write

$$\dot{x}(t) = -\xi x(t-d) - \theta \Delta q(t-d) \tag{6}$$

where $\xi = \alpha / \tau_2$ and $\theta = -\beta / (\tau_2 d)$, and where the $(t-d)$ arguments are due to that x and q are observed at time d (discrete index n) before the derivative is observed at time t (discrete index n+1). This is actually the very same equation as found in the analysis of XCP [KHR02] and RCP [DKZSM05], except for the definitions of the scaling factors ξ and θ. This difference is due to the stepwise adjustment of rate for the media sources in P-AQM once per congestion signaling period (which is larger that RTT), while in XCP and RCP there is typically multiple updates per RTT.

This homogeneous differential equation can now be Laplace translated. Following well known Laplace translation rules, where $X(s)=L\{x(t)\}$, and observing that $(\dot{\Delta q})(t) = -x(t)$, we now have

$$sX(s) + \xi X(s) e^{-ds} + \theta \frac{X(s)}{s} e^{-ds} = 0 \tag{7}$$

where the e^{-ds} terms take care of the feedback delay d. If running this system with a dedicated reference signal $u(t)$ on the right hand side, we get the Laplace transform of the inhomogeneous differential equation to be

$$sX(s) + \xi X(s) e^{-ds} + \theta \frac{X(s)}{s} e^{-ds} = U(s) . \tag{8}$$

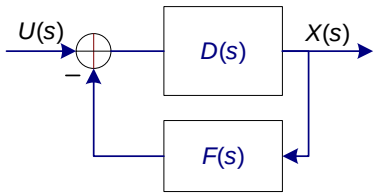

Figure 3 *The Laplace transformed block schematic of the feedback system, where* $D(s) = e^{-0.5ds}/s$ *and* $F(s) = e^{-0.5ds}(\xi s + \theta)/s$. *The transfer function is* $X(s)/U(s) = D(s)/(1 + D(s)F(s))$.

The transfer function from $u(t)$ to $x(t)$ can now be expressed as

$$\frac{X(s)}{U(s)} = \frac{1/s}{1 + e^{-ds}(\xi s + \theta)/s^2}. \tag{9}$$

The open loop transfer function $A(s)=D(s)F(s)$ is thus

$$A(s) = \frac{\xi s + \theta}{s^2} e^{-ds}. \tag{10}$$

Classical Bode/Nyquist stability criteria say that a closed loop is stable if the open loop transfer function has a gain < 1 when the phase of the same function crosses $-\pi$, i.e.

$$|A(j\omega)| < 1 \text{ at } \angle A(j\omega) = -\pi \tag{11}$$

(true for all loop functions crossing magnitude 1 only once, as in this case). The open loop magnitude and phase is given as

$$|A(j\omega)| = \sqrt{\frac{\xi j\omega + \theta}{(j\omega)^2} \cdot \frac{-\xi j\omega + \theta}{(-j\omega)^2}} = \frac{\sqrt{\xi^2 \cdot \omega^2 + \theta^2}}{\omega^2} \tag{12}$$

observing that $|e^{-d \cdot j\omega}| = 1$ for all ω, and

$$\angle A(j\omega) = -\pi + \operatorname{atan}\frac{\omega\xi}{\theta} - \omega d, \tag{13}$$

respectively. Defining ω_{180} as the frequency where the phase is $-\pi$, and ω_z as the frequency where the gain is 1, the stability criteria is $\omega_z<\omega_{180}$. Putting (12) equal to unity and inserting $\xi = \alpha/\tau_2$ and $\theta = -\beta/(\tau_2 d)$ leads to

$$\omega_z = \frac{1}{\sqrt{2}}\sqrt{\left(\frac{\alpha}{\tau_2}\right)^2 + \sqrt{\left(\frac{\alpha}{\tau_2}\right)^4 + 4\left(\frac{\beta}{\tau_2 d}\right)^2}} < \omega_{180}. \tag{14}$$

Putting (13) equal to $-\pi$ gives

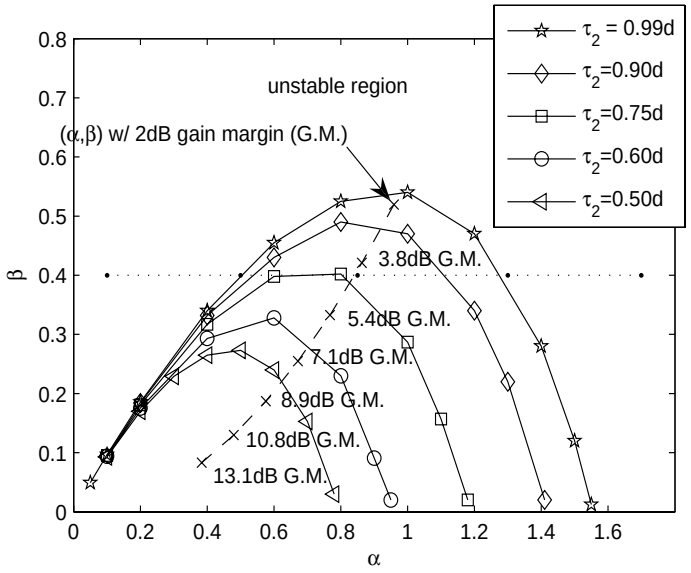

Figure 4 *The stability region is below the convex lines. The stability area decreases at decreasing τ_2/d ratio. The dashed line shows (α,β) pairs with constant $\omega_b=0.5\omega_z$ and $\tau_2/d=0.99$: note the decreasing gain margin at increasing α and β.*

$$d\omega_{180}\frac{\alpha}{\beta} = \tan(d\omega_{180}). \tag{15}$$

The latter has to be solved numerically, and a solution will only exist as long as $\alpha/\beta>1$.

In order to ensure that solutions exist for (14) and (15), the values of α and β have to be bounded. In order to find this stability bound, a coordinate system with axis α and β as x- and y-axis, respectively, can be drawn. (α_i,β_i) is selected and inserted into (15). If now also (14) is fulfilled, the (α_i,β_i) coordinate is within the stability region. Finding the (α_i,β_i) pairs that make $\omega_{180}=\omega_z$ enables drawing a line separating the stable and unstable region. Actually, this region is independent on the absolute values of d and τ_2, only on their ratio, as shown in Fig. 4.

In order to find known stability conditions within the stability region, closed expressions for α and β can be found when first deciding on the relationship between the nominator *break frequency* in (10), $\omega_b=\theta/\xi$ (i.e. where the nominator term has produced a phase shift of $\pi/4$), and the crossover frequency ω_z. Bodes simplified stability criteria says that for stability robustness, the break frequency should be located at a lower frequency than the crossover frequency, so that the phase can "climb" from $-\pi$ towards $-\pi/2$ and thus create a larger *phase margin* (i.e. the open loop phase at ω_z). E.g., if choosing $\omega_b=\theta/\xi=0.5\omega_z$ and inserting this into (12) and put the latter equal unity, one get

$$\beta = \frac{\sqrt{5}}{4}\alpha^2 \cdot \frac{d}{\tau_2}. \tag{16}$$

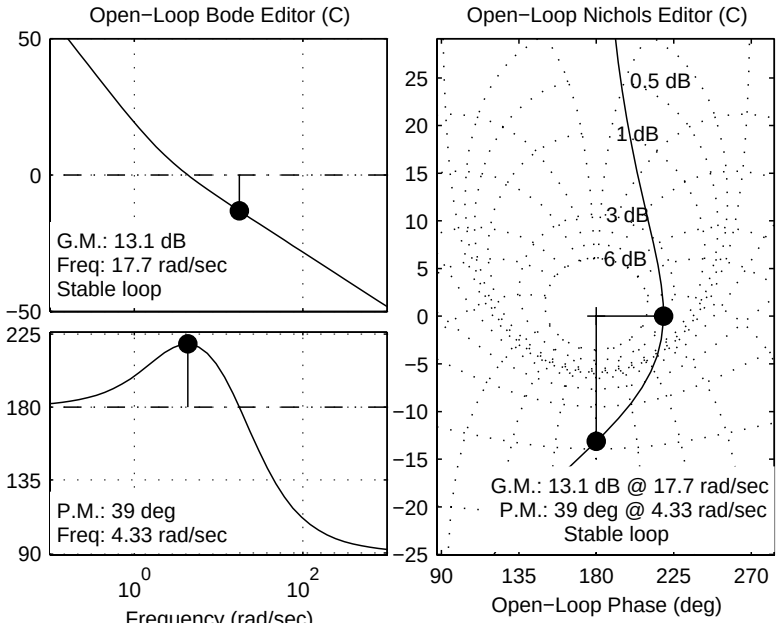

Figure 5 *Bode plot (open loop) and Nichols plot of the loop margins for $\omega_b=0.5\omega_z$, of the example where $\alpha$is 0.4 times the limit given by (17).*

To ensure stability, (13) says that $\omega_z d < \text{atan}(2) \sim (4/2.9)\pi/4$, i.e.

$$\alpha < \frac{\pi}{2\sqrt{5}} \cdot \frac{\tau_2}{d} \cdot \frac{4}{2.9}. \tag{17}$$

In Fig. 4 it is also shown (the dashed line) how the gain margin increases as the α and β is made smaller and smaller (the α is 0.99, 0.9, 0.8, 0.7, 0.6, 0.5, and 0.4 times the limit set by (17), while β is calculated using (16)). The increased stability is however achieved at the cost of slower settling time of the control loop. In Fig. 5 the Bode and Nichols plot of the largest gain margin is shown. In order to maintain a constant gain and phase margin at dynamic delay conditions, the P-AQM router should update (17) at every periodic update, and thus recalculate both α and β.

3.5 P-AQM+ERF stability analysis and verification

Due to the fact that the rate change part of (1) and (2) is identical except for the scaling factor $\hat{N}_n$, the stability analysis is exactly the same. In particular, it has the same stability dependency property: α and β must be dynamically updated in the P-AQM nodes to capture any temporary changes in τ_1 and τ_2, in order to keep a constant phase margin. In order to avoid this unwanted dependency, the P-AQM+ERF rate equation was changed to (3), in where it is observed that the rate change asked for will be valid for $\tau_2+\tau_1$ (s), and not only τ_2. This causes a modification of the stability equations, in that all τ_2 terms is replaced by d. Thus, the stability area will remain constant and independent of the network

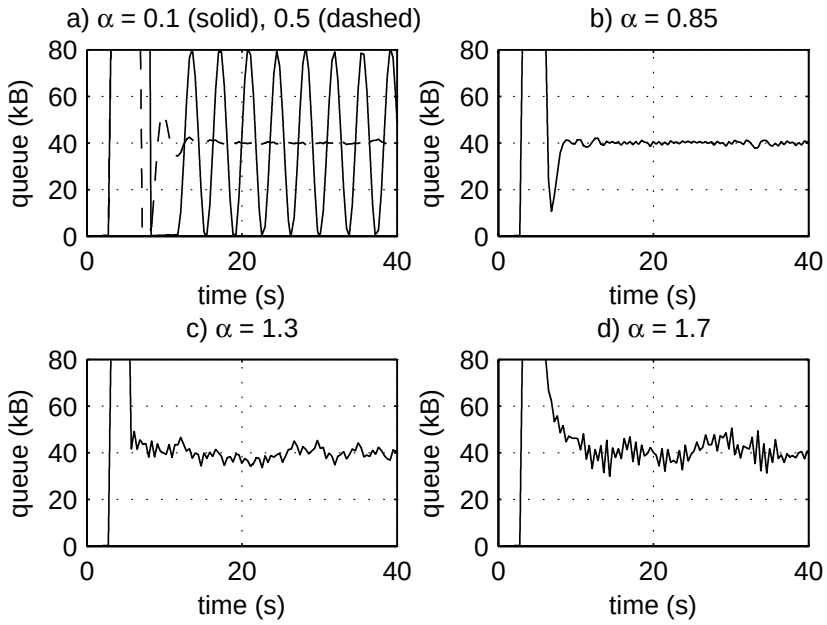

Figure 6 *Test scenario with RTT=100ms and P-AQM N* of 40kB. β=0.4 and α varied 0.1–1.7, to show unstable and stable queue performance.*

delay *d*. The actual stability region curve will be almost identical to the τ_2=0.99*d* line in Fig. 4.

In order to verify the correctness of the stability analysis, the algorithm is implemented in *ns*-2 for simulation purposes. A simple dumbbell network scenario with 10 competing 2Mbit/s CBR flows all flowing through a 16Mbit/s bottleneck link. The CBR source is submitting 400 byte packets, and is capable of rate adaptation through packet rate modulation. The end to end RTT in Fig. 6 is 100ms, while it is 400ms in Fig. 7. β=0.4 for all simulations (chosen as an example), while α is varied. As can be seen, stable queue performance is obtained in satisfactory correspondence to the mathematical analysis depicted in Fig. 4: α=0.1 makes a very unstable and oscillating system, 0.5 makes it marginally stable, 0.85 is well inside stable area, 1.3 marginally stable, and 1.7 unstable. The claim of RTT independence also holds, taking into account that the update frequency of P-AQM is lower at high RTT, thus creating a slower settling time. While α clearly influence the settling time, the estimation of the number of flows via (4) may cause packet drops if it underestimates the true value, which may happen when many new flows arrive simultaneously. This is also a known phenomenon in RCP.

However, the main motivation behind P-AQM was the support of variable rate coded media (VBR). VBR encoded video has significant traffic variability, both from GoP to GoP, but perhaps most noticeable within each GoP, due to the frame size differences of I-frame, P- frames, and B-frames. This variability actually shows a positive effect in the aforementioned problem with flow number estimation, i.e. (4) needs fewer cycles before reaching correct value. The MTU is dictating the maximum size of IP packets of each

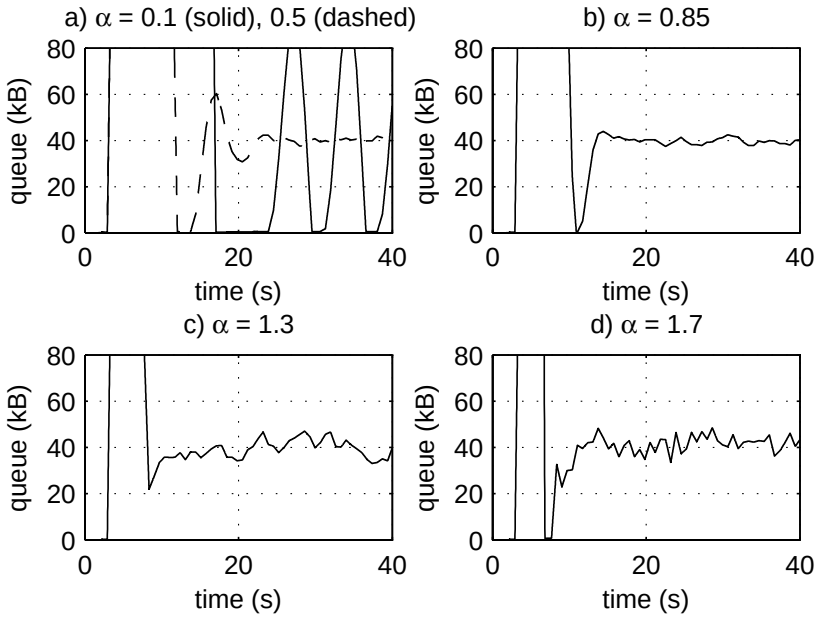

Figure 7 *Test scenario with RTT=400ms and P-AQM N* of 40kB. β=0.4 and α varied 0.1–1.7, to show unstable and stable queue performance.*

flow, guiding the packetizing of video frames to packets. Increasing the MTU to 700 bytes, and using a video test sequence consisting of a collage of official MPEG test sequences (News, Football, Stefan, and Akiyo) in CIF resolution at 30fps, results are created for variable number of non-synchronized adaptive MPEG-4 VBR flows targeting (if no network bandwidth contention) 1Mbit/s on average. The Evalvid-RA tool-set for *ns*-2 is used for this experiment. Through adaptive setting of the quantizer scale in the range Q = 2–31, the average bandwidth can be compressed down to a minimum of approximately 200kbit/s. All flows are passing through a bottleneck link, which capacity is varied from 16 to 256Mbit/s, and the RTT is varied from 10 to 800ms. The VBR flows are all started within the first seconds of the simulation, and mean and standard deviation queue delay is measured during 10–60 second of the simulation period (transient part removed, and 95% confidence intervals calculated based on 12 independent runs). Simulations reveal that τ_2 must be greater than the average GoP size in order to maintain a stable queue size, which is reasonable. The target link utilization capacity was reduced from 100% to 95% (except for the highest link capacity), which is achieved by changing c_u to $0.95c_u$ in the equations. The reason for this is that the VBR aggregate has significant rate variability, especially for low number of flows, meaning that router queue delay jitter should be limited on the cost of some 5 percent link utilization. This also means that the queue will be drained completely if the input rate is smooth CBR. Lastly, in all simulations, there is TCP background traffic consisting of long-lived flows (the RTT and number of TCP flows are equal to that of the VBR flows) and short-lived flows (Web traffic modeled by Poisson distributed flow arrivals and Pareto distributed flow sizes, following guidelines given in [KHR02]).

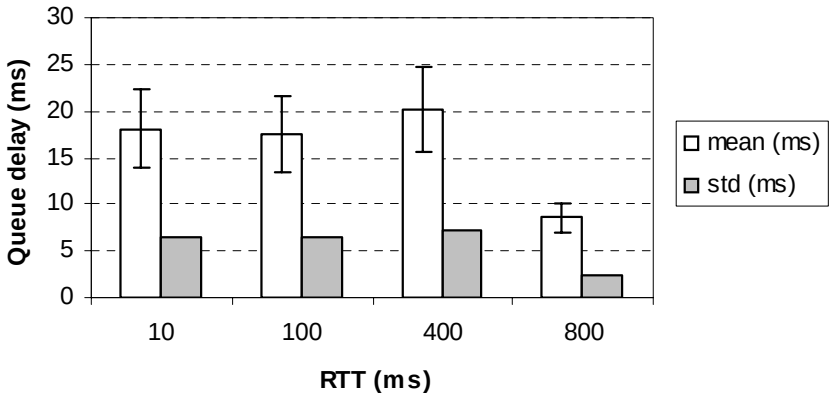

Figure 8 *Mean and average bottleneck UDP queue delay as function of RTT. The bottleneck bandwidth is 16Mbit/s, the number of VBR flows is 20, which equals the number of long TCP flows. 95% CI calculated based on 12 replicated independent runs.*

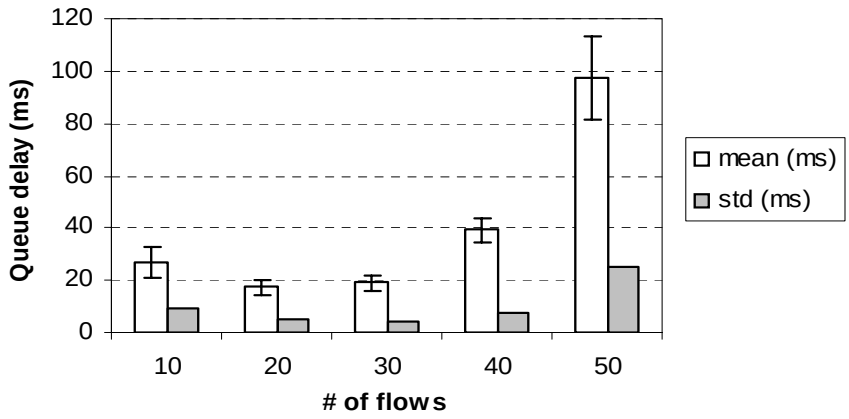

Figure 9 *Mean and average bottleneck queue delay as function of number of VBR flows (20 long TCP flows for all cases). The bottleneck bandwidth is 16Mbit/s, and RTT=50ms. 95% CI calculated based on 12 replicated independent runs.*

The results are displayed in Fig. 8 to Fig. 10. All results are obtained for α=0.75 and β=0.35, τ_2=0.5s, and target UDP and TCP queue equilibrium size is 35kB and 350kB respectively. The first figure shows that the delay is almost independent of RTT (the 800ms RTT is so high that the TCP traffic struggles to keep high utilization, i.e. the TCP queue drains completely occasionally, which in turn also decreases the UDP queue due to RR). Fig. 9 shows that when the number of flows increase, and thus the bandwidth per flow decrease, the queue delay actually first decreases somewhat (30 flows). The reason for this is that the bit variability of VBR flows decreases at lower bit rates. However, for 40 and 50 flows the delay increases again, which is a result from the fact that the fair bandwidth share is approaching the lower bound constraint of 200kbit/s. Fig. 10 shows that the method scales with larger link capacities, which actually implies that the queue delay becomes less and less as link capacity increases. At 256Mbit/s a test with 100% link uti-

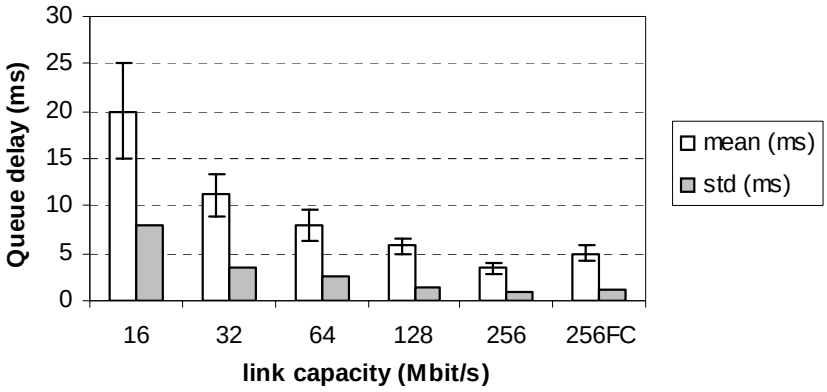

Figure 10 *Mean and average bottleneck queue delay as function of capacity. The RTT = 50ms. Target queue size is 40kB, and number of VBR (and long TCP) flows is 20, 40, and 120, respectively. 95% CI calculated based on 12 replicated independent runs.*

lization was run (256FC). The result reveals that full capacity is obtainable with low delay when the capacity and the number of flows are high enough. Since the traffic aggregate converges against a Poisson process at higher capacities, this result is in fact a direct outcome of well-known *M*/*D*/1 traffic theory, where the queue waiting time can be expressed as

$$E[W_{M/D/1}] = \frac{m(2-\rho)}{2(1-\rho)} \tag{18}$$

where m is the service time of one packet. Thus, at a given link utilization ρ, the queue delay is directly inverse proportional to link capacity. For all tests, there was no packet drops during the stable period of the simulations, and the total link utilization (including both TCP and UDP traffic) was in the region 90–98%. To conclude, these results supports the mathematical analysis and the envisioned VBR performance, in that the method provides stable VBR operation, which scales well with RTT, N, and c_u.

4. Throughput fairness

Streaming media consists generally of long-lived flows, so it is quite fundamental that the deployed congestion control algorithm maintains a stable and fair bandwidth allocation. The fairness should be independent of RTT and number of bandwidth bottlenecks, to support equal quality of short- and long-distant VoIP and videoconferencing. The concept *global max-min fairness* [BG92] has exactly these requirements. This is in contrast to the TCP congestion control, which gives long-lived flows a throughput proportional to $1/\mathrm{RTT}^{\rho}$ where $1 < \rho < 2$, thus the name *proportional fairness* [KMT98]. A third class of fairness control is called *Utility max-min* [She95], where e.g. MaxNet belongs. This section discusses the needed architecture and feedback signal accuracy to support global

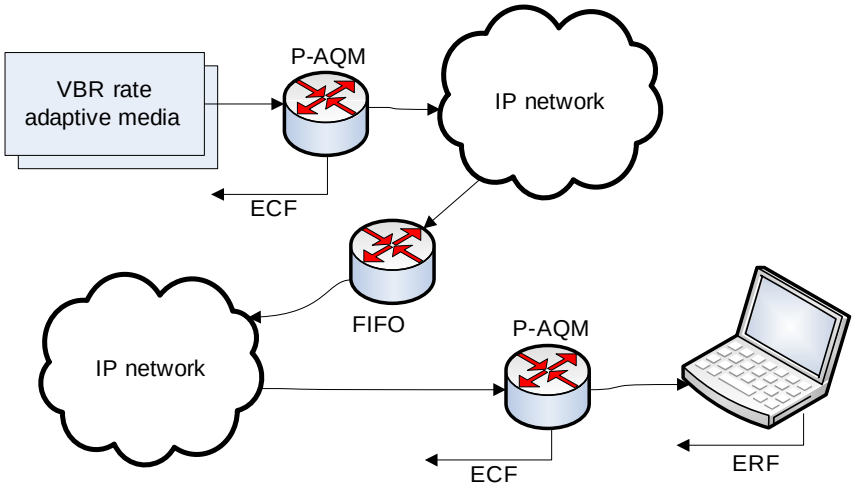

Figure 11 *P-AQM+ECF nodes signal directly from the routers using ICMP packets. P-AQM+ERF signals only from the end user terminal. The total path does not need to consist of P-AQM routers only.*

max-min fairness. It will be argued that P-AQM fulfils these requirements, supported by simulation experiments.

4.1 Is one more bit enough?

One important question is: how accurate must the network load feedback signal be? In VCP [XSSK05] the authors claim that a two-bit ECN marking will create much more efficient network state feedback than the established one-bit ECN. Two bits make room for signaling "low-load", "high-load", and "overload", while the fourth state could signal "ECN/VCP not supported". A new state is computed each time slot t_p, which by the authors is suggested to be close to 200ms. This replaces the one-bit ECN marks that signal the traffic load as a continuous flow of 0 and 1 marks (the more 1's the higher the traffic load).

A thorough analysis of VCP reveals however a fundamental weakness in its design that may create a serious starvation of long flows. The flaw is the combination of periodic updates and AIMD based on worst link state on a flows path towards its destination. In persistent traffic high-load/overload situation, a link's utilization on average is typical 90–100%, which is highly dependent of the AQM in use. The instantaneous input traffic load may vary between 90–110% or more. Now, imagine a network flow that, lets say, passes through two bottlenecked links. Lets say that the AQM algorithm controlling these links together with the rate control algorithm on average calls periodically (t_p) for three additive increase and one multiplicative decrease. Further, let's assume that there are only long-lived steady flows. Thus, each period have p_{MD}=0.25 probability of creating a MD, and p_{AI}=1−p_{MD}=0.75 probability of creating AI. A flow transversing both these links will, if the statistics are considered independent, observe a probability of *at least one MD per* t_p

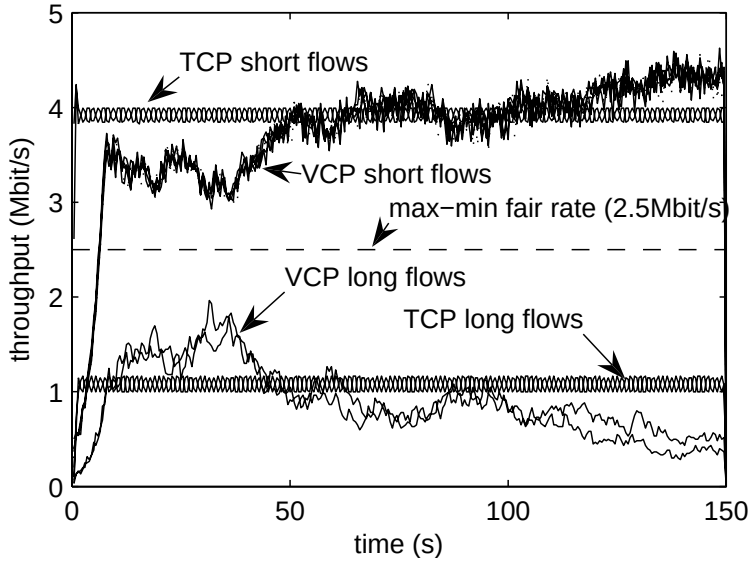

Figure 12 *The VCP long flows are starved by the short flows. TCP is more robust since the probability of being 50% reduced is proportional to its current bandwidth.*

of $p_{\mathrm{MD}}(1-p_{\mathrm{MD}})+(1-p_{\mathrm{MD}})p_{\mathrm{MD}}+p_{\mathrm{MD}}^2 = 0.4375$. Put more generally, having a path of L congested links in equilibrium with independent but assumed equal probability p_{MD} of decrease rate message per congestion signal period, the probability of at least one MD message per period is

$$P(X \geq 1) = \sum_{i=1}^{L} \binom{L}{i} p_{\mathrm{MD}}^{i} (1-p_{\mathrm{MD}})^{L-i}$$
$$P(X \geq 1) = 1-(1-p_{\mathrm{MD}})^{L} > p_{\mathrm{MD}}. \tag{19}$$

This means that long flows with multiple bottlenecks will see larger probability of MD than short flows with fewer bottlenecks. To see how this affects VCP, the following was simulated in *ns*-2: A parking lot topology has 3 bottlenecked links. Each bottlenecked link carries two long flows and two short (cross-traffic) flows. The link capacities are 10 Mbit/s, thus, the max-min fair share per flow is 2.5 Mbit/s. The long flow RTT is 112 ms, while the short flows RTT is 28 ms. *Ns*-2 simulation shows that, when the VCP routers t_p periods are not synchronized (as would be the case in real life), the load factor they calculate can be regarded as independent of each other. As Fig. 12 shows, the long VCP flows throughput are *starved* (continuously lowering the throughput). In comparison, the long TCP flows are not starved. None of these are close to obtaining global max-min fairness (2.5 Mbit/s). To conclude: neither one- nor two-bit ECN marking fulfills the fairness requirement.

4.2 P-AQM: max-min fairness for media flows demonstrated

Obviously, more detailed network state information is needed. When inspecting VCP, MaxNet, XCP, RCP, and P-AQM, it is only RCP and P-AQM that accomplish global max-min fairness [BG92], without requiring synchronized or global fine tuned network parameters. If the relative AIMD signaling is replaced by *absolute* measures of rates performed by e.g. RCP routers/switches, the network would be able to differ between the flows actual bottlenecks and eventually other saturated links. Actually, RCP manages to do this in a scalable way. The only problem is that RCP is dependent on that the sources scale their bandwidth smoothly to exact what the RCP calls for. In contrast, P-AQM is designed to control CBR and VBR video sources, where the latter has significant bandwidth variance.

In this subsection the global max-min fairness properties of P-AQM+ECF, P-AQM+ERF and TFRC is demonstrated and compared by *ns*-2 simulations of a GFC-2 network architecture (Fig. 13), chosen because GFC-2 was explicitly designed to reveal max-min performance of traffic control architectures [Sim94]. The sources used in the simulations are rate adaptive CBR (synthetic) only, and rate adaptive VBR (MPEG-4) plus TCP flows. Capacity unit C was set to 20 Mbit/s, and delay unit D to 5 ms (giving 65 ms propagation delay for the B-flow entering network at far left end). The global max-min fair throughput distribution for this setting is given in the Fig. 13.

In Fig. 14 the P-AQM+ECF is simulated with rate adaptive CBR sources only. The rate throughput is averaged over 0.6 s time window. The number of sources is the same as in Fig. 13, and they are all started with the same random seed uniformly within the first 5 s of the 30 s simulation time. Due to the smooth source rate, the convergence time of P-AQM+ECF to max-min fairness is slow, as expected. The P-AQM+ERF behavior shown in Fig. 15 reveal that this method is much faster, and after approximately 12 s all sources has reached perfect max-min fairness (as indicated by the horizontal dashed lines). E2e

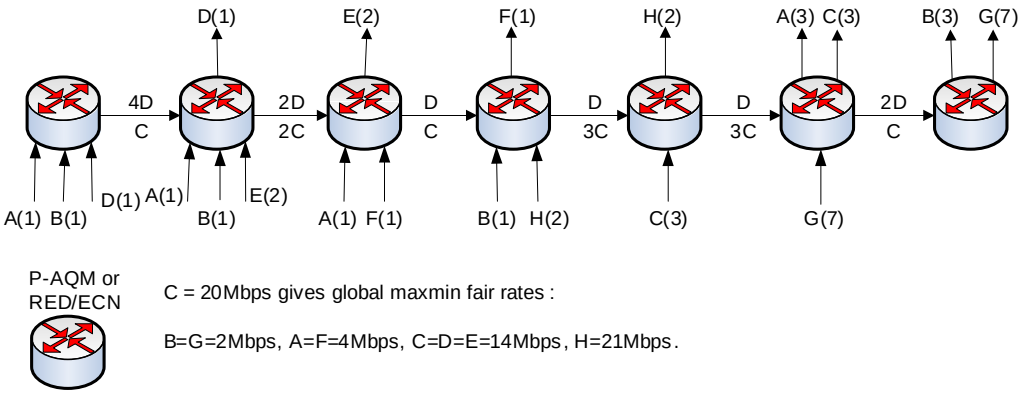

Figure 13 *The GFC-2 network consists of multiple link bottlenecks. The number of flows is given in parentheses. All flows with similar character name should be granted equal throughput. Only the A- and B-flows are long flows, the rest is short (cross-traffic) flows.*

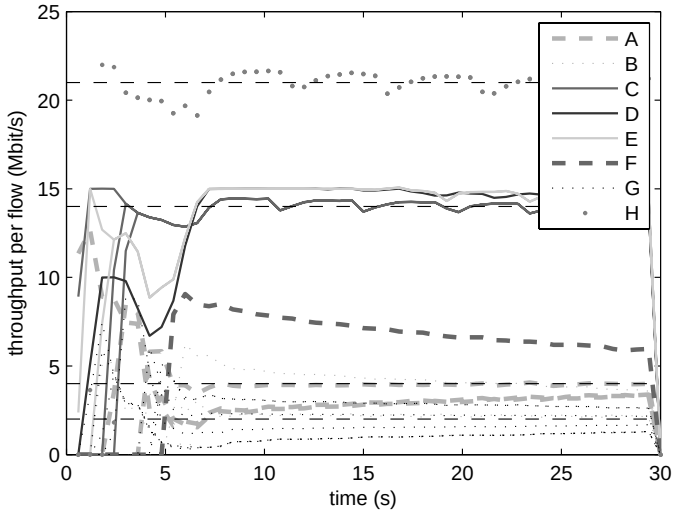

Figure 14 *P-AQM+ECF routers and rate adaptive CBR traffic. The legend is valid also for all subsequent plots.*

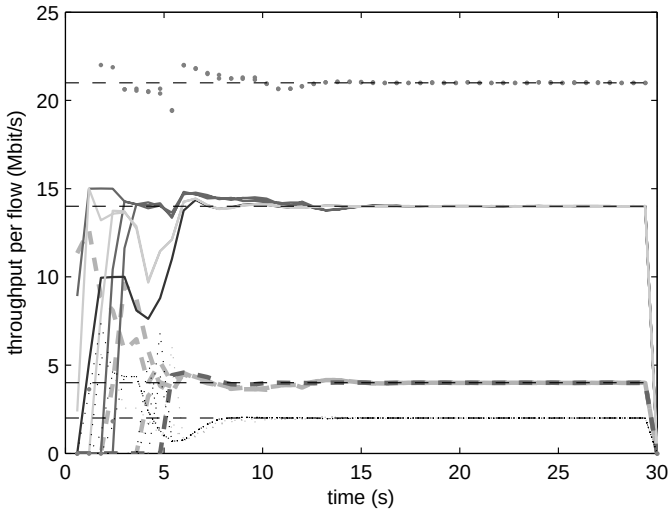

Figure 15 *P-AQM+ERF routers and rate adaptive CBR traffic.*

delay was approx. 75 ms (95% link utilization) and approx. 120ms (100% link utilization), the difference being queue wait time added with 50 kB queue equilibrium. The TFRC simulation (using RED/ECN routers) of Fig. 16 shows that stable max-min fairness is not reached. In fact, if not the H-flow was self-limited to 25 Mbit/s, and C- D- and E-flows limited to 17 Mbit/s, the results would have been even worse. Some of the long flows (A- and B-flows) are granted only one third of their max-min fair share of capacity. E2e delay was approx. 240 ms at 30 s simulation time, but still increasing.

In Fig. 17–19 the GFC-2 networks were simulated with MPEG-4 VBR sources, using the Evalvid-RA tool-set. The throughput was averaged over 1.6 s (approx. 3 GoPs). All sources were of CIF quality at 30 fps, and a self-limiting upper rate of 1 Mbit/s was applied. The aggregate video traffic load would not be high enough to cause congestion,

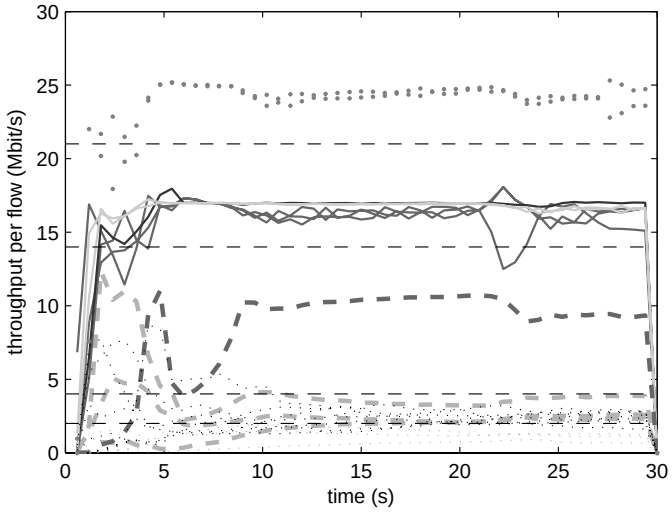

Figure 16 *TFRC over RED/ECN routers and rate adaptive CBR traffic.*

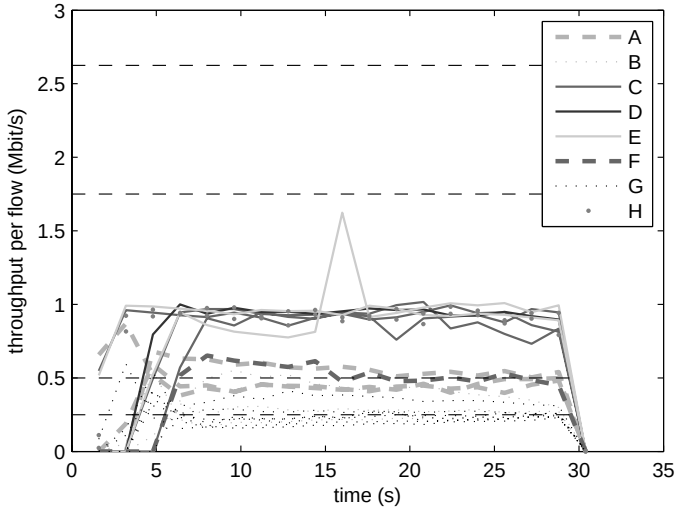

Figure 17 *P-AQM+ECF routers and rate adaptive VBR traffic.*

using the normal number of GFC-2 sources as outlined in Fig. 13 (22 sources). Thus, the number of sources was increased by a factor 8 (at every source location) to 176 sources: this enabled saturation of the third and last (right) link. Thus, the sources are either saturated to 250 kbit/s (B- and G-sources) or 500 kbit/s (A- and F-sources), or their self-limited rate of 1 Mbit/s. To test the networks at even more complex scenario, four greedy cross-traffic TCP sources were started at 15 s passing through the four non-congested links. This ensures that all links are saturated after 15 s, but since the number of TCP flows is much smaller than VBR flows, the max-min throughput sharing is almost unchanged. In all three simulation cases it was confirmed that these four TCP flows got approximately their fair share of the bandwidth (which was 6, 12, 26, and 18 Mbit/s, respectively). Fig. 17 and 18 show that P-AQM is very close to perfect fairness. The ECF version converges faster with VBR sources compared to CBR sources, but still the ERF

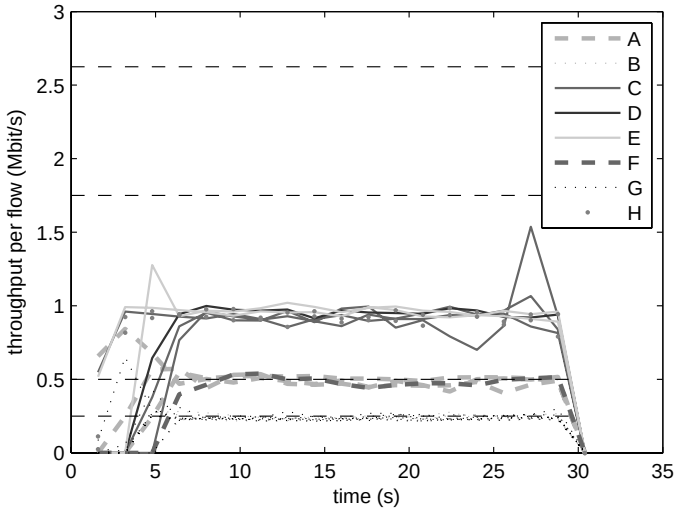

Figure 18 *P-AQM+ERF routers and rate adaptive VBR traffic.*

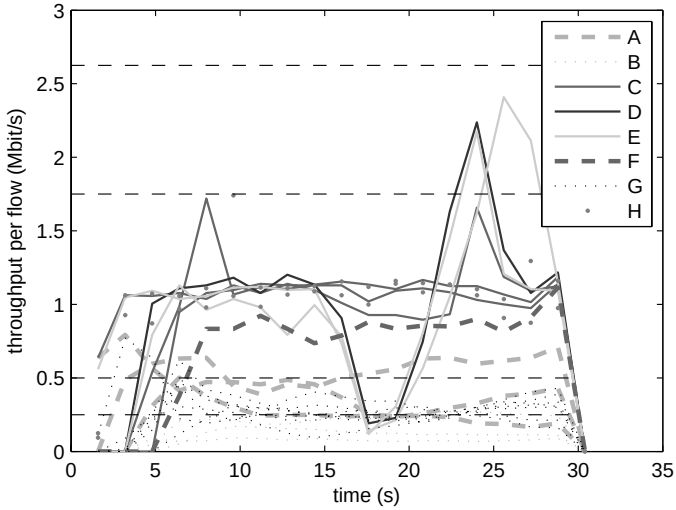

Figure 19 *TFRC over RED/ECN routers and rate adaptive VBR traffic.*

version is the fastest and most stable. The TFRC results of Fig. 19 shows that its performance is more subtle: in the last part of the simulated period some G-flows have much more bandwidth than some A-flows. The TFRC bandwidth is also heavily affected by the TCP flows entering at 15 s, while the P-AQM systems are not. The bandwidth variability of the self-limited flows (all three cases) is due to their VBR nature, which have higher variance at high bandwidth compared to low bandwidth.

Fig. 20 shows the e2e network delay experienced by one of the eight MPEG-4 VBR B-flows entering the GFC-2 network at the left end (passing all six bottlenecks). It showed approximately 120 ms on average for P-AQM (ECF and ERF), while TFRC has much larger and increasing delay. It is also evident that the TCP flows started at 15 s brings a delay spike into the TFRC system, while it is almost invisible in the P-AQM systems.

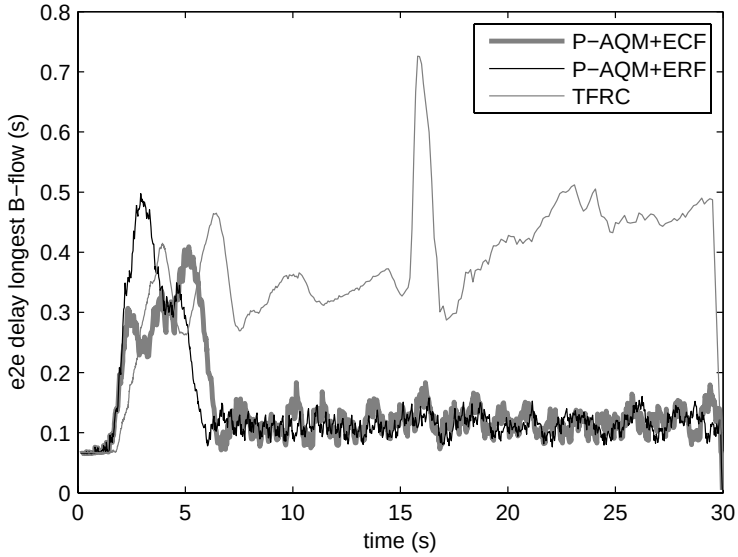

Figure 20 *The e2e delay of the three VBR GFC-2 cases (longest B-flow, 100% link utilization target).*

5. Deployment issues and discussion

It has been shown that P-AQM+ECN and ERF exhibit stable performance, and global max-min throughput distribution between the competing sources. It is superior to TFRC both in e2e delay and fairness considerations. The cost is added intelligence put into the network. The feedback signaling is either direct from routers (ECF and ERF may use ICMP SQ), or only from end node (for ERF only, RTCP is an option if RTP is used on top of UDP, eventually through the use of RCP or RCP-like protocol, or to piggy-back this information in reverse direction packets of interactive communication). The streaming server (or real-time encoder) must support a rate control that reacts on the feedback.

The Internet carries of a mixture of layer 3 and 4 packets. The P-AQM must use its UDP queue for all delay-critical flows. The TCP queue is used for the rest (TCP, DCCP/TFRC, etc.). The P-AQM TCP part uses ECN marking as RED routers do, but with a different AQM algorithm [LAR05]. Gradual deployment is quite feasible, since other type of routers (e.g. FIFO or RED/ECN) can co-exist. If the network bottleneck is located at the P-AQM router(s), the performance will be identical to an all-P-AQM network. If some FIFO or RED/ECN routers also exhibit congestion, the performance will diverge. In the ERF regime, it should be possible to combine packet loss or marking statistics and the P-AQM flow rate suggested at the destination node, calculating an equivalent $R(k+1)$, which is used as the feedback information. Ill-behaving UDP-sources (non-congestion controlled high rate UDP sources) is a problem for all networks: in P-AQM, such flows could be identified and given packet drops starving those flows to their real fair rate constraint, e.g.

following the *strike concept* of FRED [LM97]. This is actually part of the P-AQM implementation found at `www.item.ntnu.no/~arnelie/Evalvid-RA.htm`.

Dedicated networks, e.g. Web-TV over cable or satellite, can be tailor made with P-AQM nodes only. The final TV program multiplex can actually be located in the satellite transponder, i.e. the precious transponder capacity can be almost 100% utilized: the TV sources will adapt their output according to the P-AQM feedback loop algorithm, and ensure low delay at marginal packet loss.

6. Conclusion

In this paper the P-AQM feedback algorithm, proposed as an efficient control algorithm for conversational rate adaptive CBR and VBR video over UDP, has been analyzed by classical control theory methods, and stability region bounds have been settled. These bounds was then verified by *ns*-2 simulations, where both rate adaptive CBR and real rate adaptive MPEG-4 traffic was used as input, on top of TCP background traffic. Further, it was argued why the P-AQM architecture, with its ECF or ERF feedback signaling, would give global max-min fairness throughput. This was also verified by *ns*-2 simulations of a GFC-2 network. When compared to the end-to-end regime controlled TFRC, the performance benefits are many: controlled and low network delay, no transmit buffer (which also adds delay), and no starvation of long flows violating the max-min throughput distribution. While it is true that added network intelligence is required, a gradual deployment of P-AQM routers in the Internet is quite feasible. Dedicated video networks, such as cable-TV and satellite, will benefit from the possibility of low delay and low packet loss probability at close to full link utilization.

Appendix A. The derivation of the new P-AQM+ECF rate equation

In [LAR05] the P-AQM nodes calculates for each period n a congestion metric based on aggregate input rates r and current queue backlog $\Delta q_n = N^* - Q_n$, where N^* is wanted queue backlog size in equilibrium, and Q_n is instantaneous queue size at end of period n. It was argued that P-AQM should calculate its new aggregate rate (to be valid for the next period n+1) as

$$r_{n+1} = \frac{\Delta q_n - (\hat{r}_n - c_u)\tau_1}{\tau_2} + c_u \tag{20}$$

where τ_1 is the average *RTT* of the flows in the aggregate (between this router and the source) and τ_2 is the reminder time of the congestion signal period. $\hat{r}_n$ is the *estimation* of average input rate in τ_2 part of period n, and c_u is the current capacity of the UDP queue. Note that the equation was missing scaling constants at both the rate difference and queue backlog factor. Denoting $\tau_1+\tau_2=d$, (20) can now be rearranged to

$$\begin{aligned} r_{n+1}\tau_2 &= \Delta q_n - \hat{r}_n\tau_1 + c_u\tau_1 + c_u\tau_2 \\ r_{n+1}\tau_2 + \hat{r}_n\tau_1 &= \hat{r}_n d + \Delta r \cdot \tau_2 = dc_u + \Delta q_n \end{aligned} \tag{21}$$

where $\Delta r = r_{n+1} - \hat{r}_n$ is the change in rate asked for, but only measurable in time τ_2 (see Fig. 2). Solving for Δr one gets

$$\Delta r = \frac{d(c_u - \hat{r}_n) + \Delta q_n}{\tau_2}. \tag{22}$$

Applying scaling parameters of both terms in (22) so that the loop stability criteria can be controlled, yields

$$\Delta r = \alpha\frac{d(c_u - \hat{r}_n)}{\tau_2} + \beta\frac{\Delta q_n}{\tau_2}. \tag{23}$$

References

[BENB06] Horia V. Balan, Lars Eggert, Saverio Niccolini, and Marcus Brunner. An Experimental Evaluation of Voice Qualit over the Datagram Congestion Control protocol. Technical report, NEC Europe, Germany, 2006.

[BG92] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, 1992.

[DKZSM05] Nandita Dukkipati, Masayoshi Kobayashi, Rui Zhang-Shen, and Nick McKeown. Processor Sharing Flows in the Internet. In *Proc. of Thirteenth International Workshop on Quality of Service (IWQoS)*, Passau, Germany, June 2005.

[FKP06] S. Floyd, E. Kohler, and J. Padhye. Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC). Technical report, IETF RFC4342, March 2006.

[Flo03] S. Floyd. HighSpeed TCP for Large Congestion Windows. Technical report, IETF RFC 3649 Experimental, Dec 2003.

[HRR97] M. Hamdi, J. W. Roberts, and P. Rolin. Rate control for VBR video coders in broad-band networks. *IEEE Journal on Selected Areas in Communications*, 15(6), August 1997.

[II01] ISO-IEC/JTC1/SC29/WG11. ISO/IEC 14496: Information technology - Coding of audiovisual objects. Technical report, MPEG, 2001.

[JWL04] C. Jin, D. X. Wei, and S. H. Low. FAST TCP: Motivation, Architecture, Algorithms, Performance. In *Proc. of IEEE Infocom*, 2004.

[KHF06] E. Kohler, M. Handley, and S. Floyd. Datagram Congestion Control Protocol (DCCP). Technical report, IETF RFC4340, March 2006.

[KHR02] D. Katabi, M. Handley, and C. Rohrs. Congestion Control for High Bandwidth-Delay product Networks. In *Proc. of ACM Sigcomm*, 2002.

[KJF+00] Shivkumar Kalyanaraman, Raj Jain, Sonia Fahmy, Rohit Goyal, and Bobby Vandalore. The ERICA Switch Algorithm for ABR Traffic Management in ATM Networks. *IEEE/ACM Transactions on Networking*, 8(1):87–98, Feb 2000.

[KMT98] F. P. Kelly, A. Maulloo, and D. Tan. Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability. *Journal of the Operational Research Society*, 49:237–252, 1998.

[LAR05] A. Lie, O. M. Aamo, and L. A. Rønningen. A Performance Comparison Study of DCCP and a Method with non-binary Congestion Metrics for Streaming Media Rate Control. In *Proc. of 19th International Teletraffic Congress (ITC'19)*, Beijing, China, Aug–Sept 2005.

[LK07] Arne Lie and Jirka Klaue. Evalvid-RA: Trace Driven Simulation of Rate Adaptive MPEG-4 VBR Video. *ACM Multimedia Systems Journal*, 2007. Pending Publication.

[LM97] D. Lin and R. Morris. Dynamics of Random Early Detection. In *Proc. of Sigcomm*, 1997.

[LMR97] T. V. Lakshman, P. P. Mishra, and K. K. Ramakrishnan. Transporting Compressed Video Over ATM Networks with Explicit Rate Feedback Control. In *INFOCOM '97: Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution*, page 38, Washington, DC, USA, 1997. IEEE Computer Society.

[MWMM01] M. Miyabayashi, Naoki Wakamiya, Masayuki Murata, and Hideo Miyahara. MPEG-TFRCP: Video Transfer with TCP-friendly Rate Control Protocol. In *Proc. of IEEE International Conference on Communications (ICC2001)*, volume 1, pages 137–141, June 2001.

[Phe04] Tom Phelan. TFRC with Self-Limiting Sources. Technical report, Sonus Networks, Oct 2004.

[Phe07] T. Phelan. Strategies for Streaming Media Applications Using TCP-Friendly Rate Control. Internet-draft, IETF/Sonus Networks, July 2007.

[RHE99] R. Rejaie, M. Handley, and D. Estrin. RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet. In *Proc. of IEEE Infocom*, March 1999.

[She95] Scott Shenker. Fundamental Design Issues for the Future Internet. *IEEE Journal on selected areas in communications*, 13(7):1176–1188, Sept 1995.

[Sim94] Robert J. Simcoe. Test Configurations for Fairness and other Tests. Technical Report AF-TM 94-0557, ATM Forum, 1994.

[SW00] D. Sisalem and A. Wolisz. LDA+ TCP-Friendly Adaptation: A Measurement and Comparison Study. In *Proc. of NOSSDAV*, 2000.

[WAZ03] Bartek Wydrowski, Lachlan L. H. Andrew, and Moshe Zukerman. MaxNet: A Congestion Control Architecture for Scalable Networks. *IEEE Communications Letters*, 7(10):511–513, Oct 2003.

[XSSK05] Yong Xia, Lakshminarayanan Subramanian, Ion Stoica, and Shivkumar Kalyanaraman. One More Bit Is Enough. In *Proc. of Sigcomm*, Philadelphia, August 2005. ACM.

Part III — Appendices

The season when to come, and when to go,
To sing, or cease to sing, we never know.

Alexander Pope — English poet (1668–1744)

Appendix A

Distributed Multimedia Plays with QoS guarantees over IP

Arne Lie and Leif Arne Rønningen

Published in
Proceedings of IEEE Wedelmusic'03, ISBN 0-7695-1935-0

14–17 Sept., Leeds UK, 2003

Distributed Multimedia Plays with QoS guaranties over IP

Arne Lie and Leif Arne Rønningen

NTNU Department of Telematics, N-7491 Trondheim, Norway.
Arne.Lie@item.ntnu.no, Leifarne@item.ntnu.no

Abstract

Musical collaboration over telecommunication networks has marvellous possibilities when it comes to musical education, practise and performance. The deployment and use of high-capacity digital networks makes it possible to obtain end-to-end audio and video latency of 5–20 milliseconds, which is similar to the audio time delay typically experienced between musicians on a stage. The main challenge of reaching this latency budget over digital packet switched network such as the Internet, is how to control the queuing delay of the IP packet experienced at each router on the network paths between each participating musician. Other important aspects are video and audio codec latency, and error resilient tools needed to cope with situations where IP packets arrive too late or have been lost in the network. This paper gives an overview of resent research (work-in-progress) in this field, including network simulations, conducted at the NTNU, and suggests how these techniques can be utilized under live concert performances.

A.1 Introduction

Enabling musical collaboration over IP networks will open new possibilities for both professionals as well as novice musicians. Without the need for using high-cost professional equipment and leased lines, education, practise, "jam sessions" as well as live professional performances can be possible over today's IP networks, provided sufficient access bandwidth and high-capacity core networks are used. The introduction of high bandwidth IP networks such as WDM [1] creates the super-highway backbone for supporting the deployment of new resource demanding multimedia services, including both high-fidelity audio and video. The term *Distributed Multimedia Plays* (DMP) has been launched [2] to indicate Collaborative Virtual Environments for synchronized musical performance.

However, traffic congestion will occasionally happen, and parts of the network path will have more limited capacity, such as the access networks. Care must be taken to maintain

the latency budget. Network latency per link consists of three parts: *propagation delay* (~2.0e8 m/s in optical fibres), *transmission delay* per IP packet (inversely proportional to bandwidth), and *queuing delay* (router buffering). Internet generally provides only best-effort communication, while real-time traffic typically will require special treatment in order to guarantee some limits are met on the variable queuing delay. Quality of Service (QoS) in IP networks introduces differentiation in how IP packets are treated. IETF technologies such as IntServ [6] (reservation based per flow) and DiffServ [8] (prioritization of a small set of classes) have been standardized and implemented with limited success so far as QoS mechanisms to support real-time requirements over IP networks. Only IntServ can *guarantee* latency requirements (queuing delay). However, IntServ does not scale well in large networks, because each router has to keep state information on every flow requiring special treatment. A new method based on the scalable DiffServ has been proposed [2][3], where the multimedia traffic class indeed is given latency guaranties. The method also relies on the usage of low-latency and error resilient audio and video codecs if DMP is the targeted application.

The paper will give an overview of the proposed QoS scheme and codec requirements, show simulation results, and finally give suggestions for applications utilizing DMP enabled networks.

A.2 QoS network architecture

A.2.1 Latency budget

The latency budget of the channel between musicians consists of:

1. audio delay musician–microphone (if any)
2. audio encoding delay / video encoding delay (including buffering), IP packetizing delay
3. IP packet transmission delay
4. end-to-end propagation delay
5. router queuing delay
6. audio decoding / video decoding
7. audio delay loudspeaker–musician

This study investigates encoding/decoding delay and IP queuing delay, with a focus on the latter, which also is the only dynamic delay. The distance alone determines propagation delay. E.g. Trondheim-Oslo experience 2ms delay while Trondheim-Madrid will have 15ms. Transmission delay each 1500 byte IP packet over 1Gbit/s link is 12 µs and

0.4ms over 32Mbit/s link, and is therefore of limited importance. Thus, given the distance and an overall channel latency budget of 20ms, codec and queuing delay together (ignoring the other parts of the budget) must not exceed 20ms minus propagation delay. I.e., DMP QoS requirements will be easier to achieve the smaller distance between the participants. This motivate for good traffic control in order to extend the DMP radius as close to the maximum radius (given by the distance with 20ms delay) as possible.

A.2.2 Traffic Control Algorithm RL-QoS

DMP needs QoS mechanisms even when deployed over high-capacity networks. The reason is that since the delay must be kept low, only short buffers are available. At the same time, we want to utilize available capacity almost 100%, because Gigabit networks enable this possibility without having severe delay penalties. In this way, each source can send with maximum available fidelity. This gives a traffic load operating close to saturation. In order to balance the sources output rates against network throughput, one needs a very fast QoS mechanism that is able to scale dynamically the traffic flows up and down, both at traffic congestion nodes, but also at the origin sources. This is what the invented "RL-QoS" mechanism does. In [2][3] RL-QoS defines three traffic classes. The routers identify these classes by reading the TOS (Type of Service) field of IPv4 or Traffic Class field of IPv6. TC1 is the highest priority class, and will among other things be used for QoS algorithm *signalling*. TC2 is the target multimedia streaming class itself, using RTP/UDP/IP protocol. Guaranties on latency are implemented using one-hop feedback "scale-messages". TC3 is the ordinary TCP/IP best-effort traffic class, for data transfer where delay is not critical (elastic applications like web-browsing and ftp).

A special calculated metric is used for implementing *congestion avoidance* for TC2 class packets. In contrast to other congestion avoidance techniques, this method ensures that the routers in addition to scaling the traffic locally, also signals the traffic load globally via its directly connected routers and hosts. The effect is that an overloaded network is brought under control very fast. In contrast to RTP with RTCP signalling, which signals packet loss only end-to-end, RL-QoS will scale the traffic at all nodes all the time. Simulations have shown that it is possible to control network traffic "avalanche" within 5–10 ms [3]. Also, a connection is created between traffic control mechanisms in the network and at the sources, giving the sources the best possibilities to adapt to the available network resources at any time. Scalable codecs can adapt their bit rate to fit the instantaneous network throughput bit rate.

As in ordinary FIFO queuing, packets must be dropped when arriving at full queues. However, RL-QoS also deletes some packets when free buffer memory *is* available. This is based on *statistical probability dropping*, where the probability is a calculated metric based on past, current, and predicted traffic load [2]. The first version of the RL-QoS does

not schedule each flow separately, but all incoming TC2 packets are treated equally. If the router also reads the RTP header, intelligent packet drops can be performed, based on timing and flow information. Using Kalman filtering can also enhance the traffic load prediction. All these enhancements will be investigated in follow-up work.

Figure 1 shows the model simulated in [2][3]. It assumes 30 Mbit/s dedicated access net-

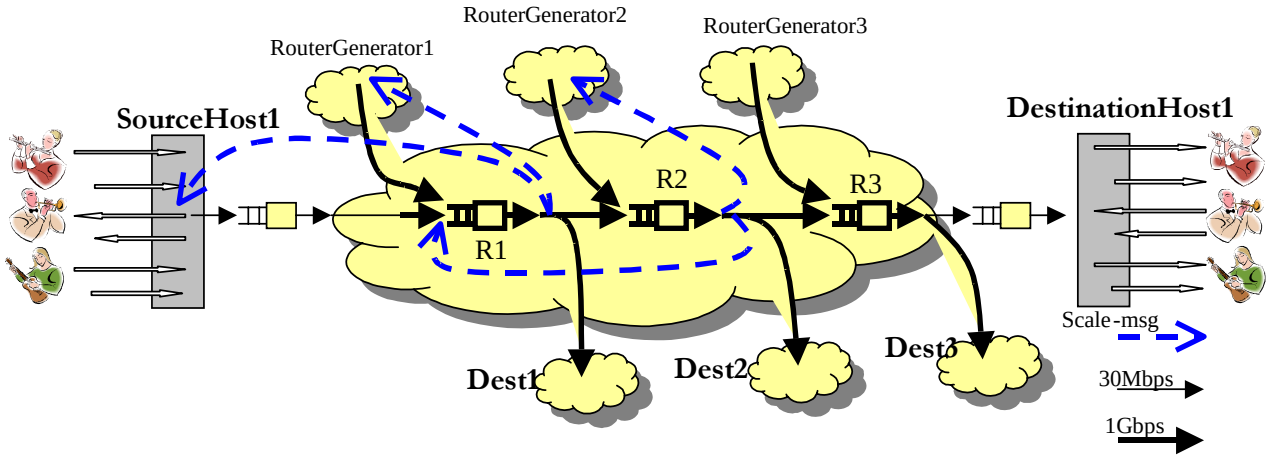

Figure A.1 *The sample high-capacity network simulated in [2] and [3] to give quantitative results applicable for DMP. Core network is 1Gbps, while access network is 30Mbps. Network path is assumed consisting of five routers, with three in the core network. Cross-traffic ensures that the traffic injected into system is above capacity. The RL-QoS scaling signaling (TC1) is shown as dashed lines, while video/audio traffic (TC2) is shown as solid lines. The target overall latency was 10ms.*

work and 1Gbit/s shared core network. The results showed that traffic avalanches were taken care of by statistical packet dropping, reacting within 5–10ms so that (1) the total queuing and transmission delay per packet never exceeded 10 milliseconds, and (2) that the percentage of packet drop in transient periods (5–10ms) never exceeded ~20%, that (3) packet drop on average is around 0.2% inside core network, and finally (4) that all core network links had utilization around 99.9%.

Figure 2 shows the queuing delay statistics through the whole system of the non-dropped packets. No packets arrived later than 9.5 ms. Thus, the target requirement of 10ms for this sample system was met.

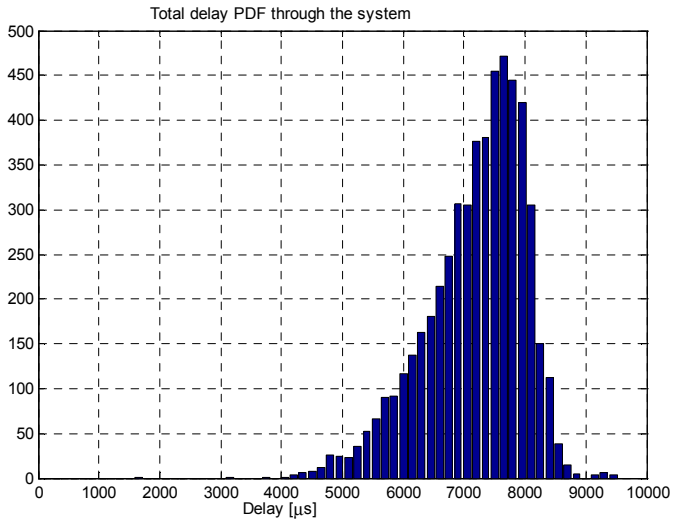

Figure A.2 *Packet delay Probability Density Function of total system. Average delay is about 7.5ms, and maximum delay about 9.5ms.*

A.3 Audio and video codec requirements

A.3.1 Codec latency

The latency budget gave the total codec delay in the order of 1–10 ms, depending of distance and RL-QoS latency requirement. This number is far below the delay of current real-time codecs used for videoconferences. The latter has a total latency budget of ~150ms, while DMP has at least one decade stronger delay requirement. The result is that a video codec run at 25 fps (40ms per frame) must use I-frame compression only, or increase the frame rate significantly to also include temporal compression. There exists low delay *audio* coding for MPEG-4 (LD-AAC) but its 20ms delay is still double the requirement, although investigations [11] show that under certain conditions the delay requirement can be extended 10–20ms. The conclusion is that to achieve these hard latency requirements, very little compression buffering can be used, compared to normal buffering done in current state-of-art video and audio coding principles. This results in lower compression efficiency and higher bit rates. However, this challenge is met by the high-capacity network.

Still, to help reduce the basic data rate generated from a scene, and create as independent statistics in the streams as possible, the RL-QoS simulations assumes *object*-based coding, and that video and sound objects are tracked, captured, and compressed in real time. Each object is treated as an entity with its private RTP/UDP/IP traffic stream. There will always be some dependency between streams, but this approach will reduce this to a min-

imum. Adaptive compression, based on network capacity signalled by the TC1 class, will be performed on each object. Scene description can e.g. be according to MPEG-4 (BIFS), and the synchronization and multiplexing as defined in MPEG-4 are entirely handled by the RTP/UDP/IP combined header [9]. Setup and release of services is assumed handled by the RTSP protocol.

A.3.2 Error resilience and scalability

Besides the codec delay requirements, there are two other main challenges. In traffic con-

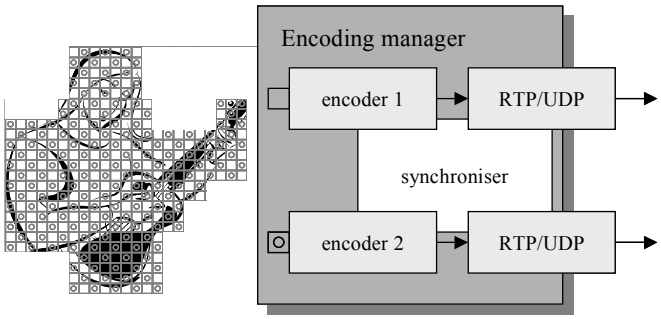

Figure A.3 *2D-Interlaced video encoding, where each field, consisting of a 2x2 sub-sampled frame in both horizontal and vertical direction, is streamed over its own private RTP/UDP session.*

gestion periods, somewhat high packet loss can be experienced. Thus, to avoid perceived quality of service below a certain threshold, strong error resilience qualities are a must. The second challenge is that the encoder must be able to adjust its *average* bit rate (real-time near-CBR mode is assumed) so that it is kept below the current channel throughput.

The benefits of using 2D-interlaced video encoding to enhance error resilience is just discovered [5]. The same approach can also be used for audio in the temporal domain. The basic idea is that the RL-QoS enabled routers never will drop more than one packet belonging to the same frame, by inspecting the RTP header. Thus, low complexity error concealment can be performed by interpolation. Thus, in this way, using two-fields per frame, up to 50% packet loss can be tolerated, without loosing the capability of doing good performance decoding.

A.4 Live concert performance over IP

In contrast to worldwide concert performances such as the opening of the Nagano Olympic Games in 1998, which were one-way interactive, the target DMP application for RL-QoS enabled networks is *two-way interactive performances*. Figure 4 shows a live perfor-

mance of three separated concert stages (ballet, brass-band, symphonic orchestra). The videos can be projected on big on-stage screens in the background, also including the conductor. The conductor can adapt to remote events as if everything is happening locally. A dedicated DMP control centre might be needed to make combined visual and audio content suitable for TV broadcast and/or web streaming. As stated in chapter A.2.1, the distance between the different stages must not exceed the latency budget due to propagation delay. 3000km (equals ~15ms) might be the maximum distance if 20ms total delay is the budget, giving only 5ms for both queuing and codec delay. In the future this latency budget might be fulfilled using Optical Burst Switching [1]. Current packet switching technology imposes shorter distances for DMP.

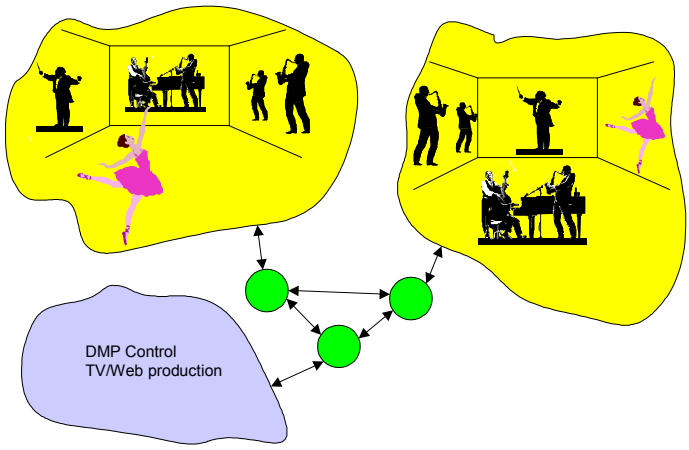

Figure A.4 *Example of two-way interactive DMP. Here, three stages with musicians simultaneously performing a concert (only two depicted).*

An RL-QoS enabled network can of course also be used for ordinary videoconferencing, with relaxed QoS requirements.

A.5 Conclusions

The proposed traffic control scheme RL-QoS has been introduced. RL-QoS builds on DiffServ in the sense that it differentiates the IP traffic into a small number of classes. However, it supports latency guaranties for RTP/UDP traffic by using short buffers, statistical packet dropping, and traffic load signalling to neighbor routers and hosts. Network simulations show that RL-QoS enabled routers and terminals can guarantee strong packet queuing delay requirements. Thus, the method is a promising solution for implementing Distributed Multimedia Plays, i.e. musical collaboration over IP networks, which requires around 10ms network latency. The use of low-delay audio and video codecs is also a must if DMP application is targeted. Strong error resilience can be obtained by using 2D-interlaced video coding in combination with the RL-QoS architecture.

Further work within this field will include investigation of enhanced traffic prediction using Kalman filtering, simulation of interlaced video coding over RL-QoS network, and implementation of RL-QoS algorithms on high-performance hardware.

References

[1] K. Dolzer, et al., "Evaluation of Reservation Mechanisms for Optical Burst Switching", AEÜ Int. Journal of Electronics and Comm., Vol. 55.1, January 2001.

[2] L. A. Rønningen, A. Lie, "Performance Control Of High-Capacity IP Networks For Collaborative Virtual Environments", IBC 2002 Conference Programme.

[3] L. A. Rønningen, A. Lie, "Transient Behaviour of an Adaptive Traffic Control Scheme", EUNICE workshop, Trondheim 2002.

[4] L. A. Rønningen, "The Combined Digital Satellite Broadcast and Internet System", Telenor Satellite Services. February 1999.

[5] T. Halbach, T. A. Ramstad, "Multidimensional Adaptive Non-Linear Filters for Concealment of Interlaced Video", ICIP Conference, Barcelona 2003.

[6] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: an Overview", IETF RFC1633. June 1994.

[7] J. Wroclawski, "The Use of RSVP with IETF Integrated Services", IETF RFC2210. September 1997.

[8] T. Li, Y. Rekhter. "A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)", IETF RFC2430, October 1998

[9] Y. Kikuchi, T. Nomura, S. Fukunaga, Y. Matsui, H. Kimata, "RTP Payload Format for MPEG-4 Audio/Visual Streams", Status: PROPOSED STANDARD. IETF RFC3016. November 2000

[10] L. A. Rønningen, "Analysis of a Traffic Shaping Scheme", the 10th International Teletraffic Congress, Montreal 1983 (part of the authors Ph.D. thesis from 1981)

[11] O. Strand, "Distributed Multimedia Plays", Master Thesis, NTNU 2002.

Appendix B

2D interlaced video

Unpublished concepts, ideas and results of improved video error resilience

2D interlaced video for improved video error resilience

Arne Lie and Leif Arne Rønningen

Abstract

Packet and frame loss in video decoding causes performance degradation, which is most often tried concealed by utilizing information from previous successfully decoded frames. I.e., time redundancy is explored as an error resilience tool. One possibility explored in very limited depth is to use spatial redundancy. Spatial redundancy must however first be created by dividing the video frames into sub-frames, consisting of almost identical copies of each other, with reduced resolution.

B.1 Description

Beside from a small discussion in [LR03], the ideas on "interlaced coding" have not been published. One publication that comes close is [HR03] where the authors seek to utilize *true* interlaced[1] video signals for concealment strategies. In "interlaced coding", progressive video as e.g. 720p, or eventually true interlaced as 1080i, each frame is divided into sub-frames with corresponding smaller resolution. E.g. for a 4:2:0 YUV signal of CIF resolution, the luminance frames of 352 x 288 pixels is divided into two 352 x 144 pixel

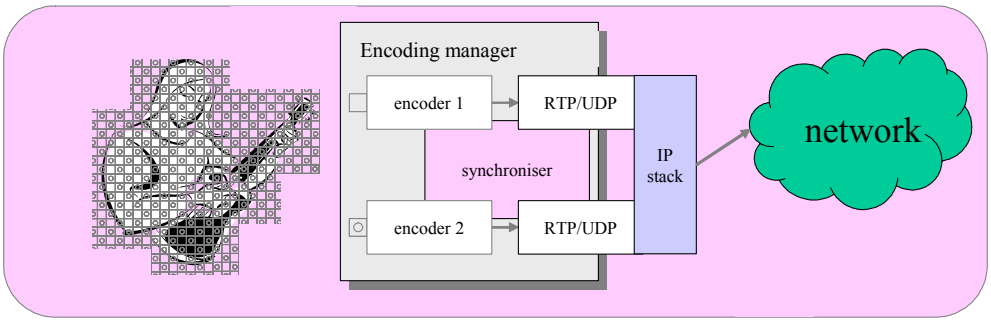

Figure B.1 *An overview of interlaced coding principles combined with possible object segmentation and arbitrary shaped MPEG-4 coding.*

1. i.e. as in e.g. interlaced PAL and NTSC signals, where the capturing time of two sequential interlaced frames are separated with 1/50 and 1/60 fraction of a second, respectively.

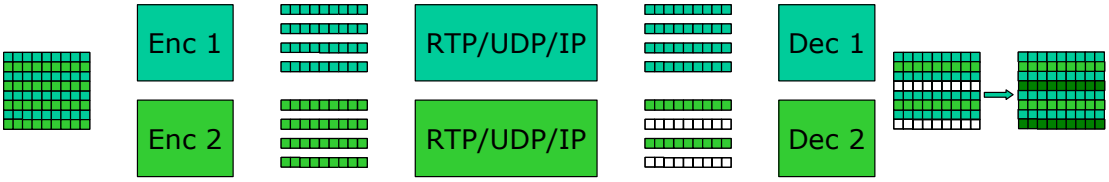

Figure B.2 *In 1D Interlacing, each frame is divided into 2 sub-frames with half resolution (both luminance and chroma). Missing pixels are corrected before rendering time as an average value of the 6 closest available pixels.*

frames, with "interlaced lines" as commonly used in both analog and digital TV signals. The difference is that these two frames, or "fields" as they are called in e.g. interlaced PAL, is having the very same capturing time. The two chroma signals should be divided in a similar manner as the luminance signal, corresponding to their resolution (in 4:2:0 YUV they have quarter the resolution each compared to the luminance frame). Two separate encoding and decoding chains must be provided. Inevitable, the achieved robustness comes at the cost of some added complexity and some reduced coding efficiency. The process is sketched in Figure B.2 (the communication channel is depicted as "RTP/UDP/IP" box), and labeled "1D interlaced coding".

Even more robust concealment strategies open if the sub-frame divisioning is performed using other pattern than strict lines. This can be utilized even for two sub-frame divisioning, in that each sub-frame is given each other pixel in both horizontal and vertical direction. Generally, this can be labelled "2D interlacing". To increase the error resilience even more, again at some added complexity and some loss in coding efficiency, one can divide the frames into four sub-frames. This is sketched in Figure B.3. The "big surprise" is how strong the error resilience becomes. Added robustness can be achieved if one of the four IP flows is transmitted as high-priority class in e.g. a DiffServ enabled network, to limit its possible packet drop ratio.

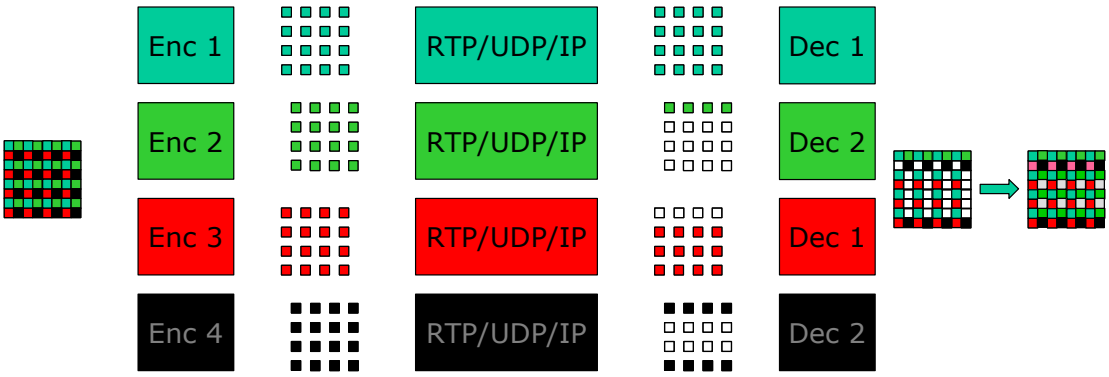

Figure B.3 *In 2D 4-frame Interlacing, each frame is divided into 4 sub-frames with quarter resolution (both luminance and chroma). Missing pixels are corrected before rendering time as an average value of the 8 closest available pixels.*

B.2 Implementation and result examination

The concept of 1D and 2D interlaced coding has been implemented in a software program, working directly on uncompressed 4:2:0 YUV signals. The performance in terms of PSNR measures has not been calculated, but some example sequences have been tested with the algorithm for visual inspection. The algorithm also supports shaped objects, and suitable content where obtained by "shooting" a person in front of a blue wall. The next four figures (CIF resolution) show typical example frames in four different cases, where "protected" means that one flow is submitted with DiffServ EF, and is assumed having zero packet loss:

- 1D interlaced coding, one protected flow, the other flow experiencing 25% information loss (Figure B.4).
- 1D interlaced coding, no protected flows, total flow experiencing 20% information loss (Figure B.5).
- 2D 4-frame interlaced coding, one protected flow, the other flows experiencing 66% information loss, giving a total of 50% information loss (Figure B.6).
- 2D 4-frame interlaced coding, no protected flow, total flow experiencing 20% information loss (Figure B.7).

The blue pixels in the person's texture (left frames) give the indication of where the information loss is located and the amount of loss in that frame. Using a real codec, the packetizing of CIF resolution video would require very small packets in order to have such many regions of uncorrelated errors. However, for resolutions like 720p, 1080i, or especially 1080p, each frame will be fragmented into many well-sized IP packets, even when using H.264. The method is as such not quite as powerful for low-resolution video. When run over networks using AQMs, such as RED and P-AQM, the packet dropping at traffic contention epochs will be randomized. Thus, such "error patterns" as exemplified will not be uncommon. The randomization is important since this increases the possibility of finding error-free pixels in the pixel neighborhood, to enable successful concealment. When inspecting the results, it is obvious that 2D interlacing is better than 1D interlacing. 20% information loss can easily be tolerated for the 2D example using flat best effort network.

Future work in this area should include the implementation of 1D and 2D interlacing with MPEG-4 and H.264 codec. This would enable measurement of the exact cost of complexity and performance loss, so that the method's performance could be compared to other error resilience tools. The method should also be possible to combine with temporal concealment strategies.

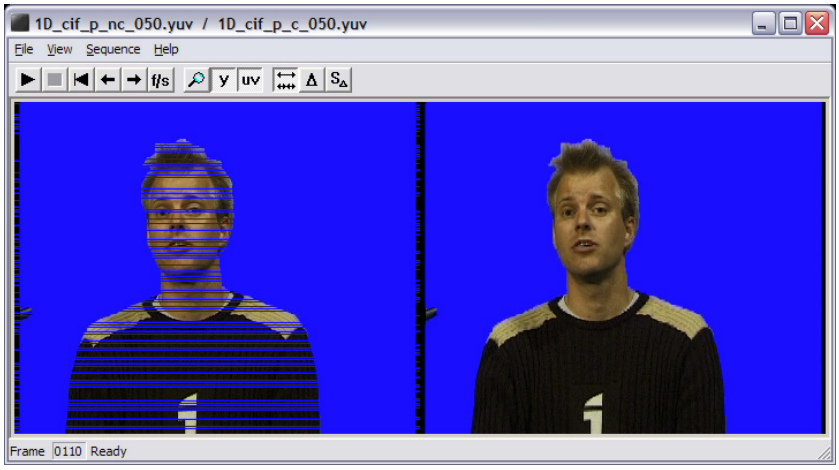

Figure B.4 *1D interlacing, 1 protected flow, 25% loss total. Left: before concealment. Right: after concealment.*

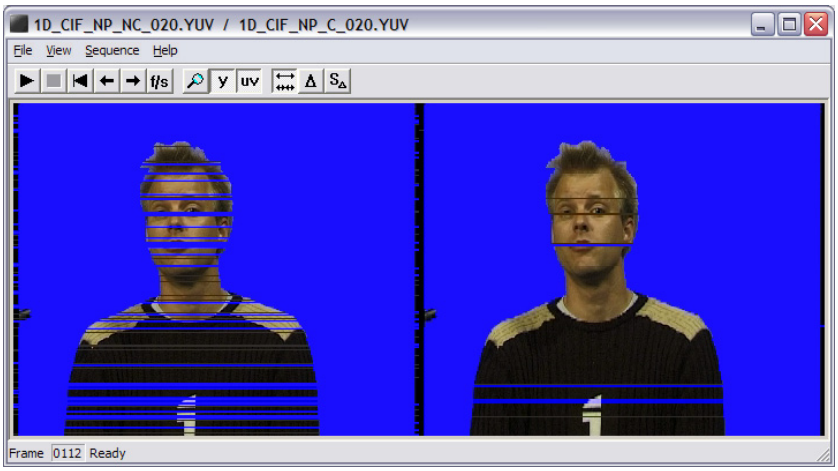

Figure B.5 *1D interlacing, single class, 20% loss total. Left: before concealment. Right: after concealment.*

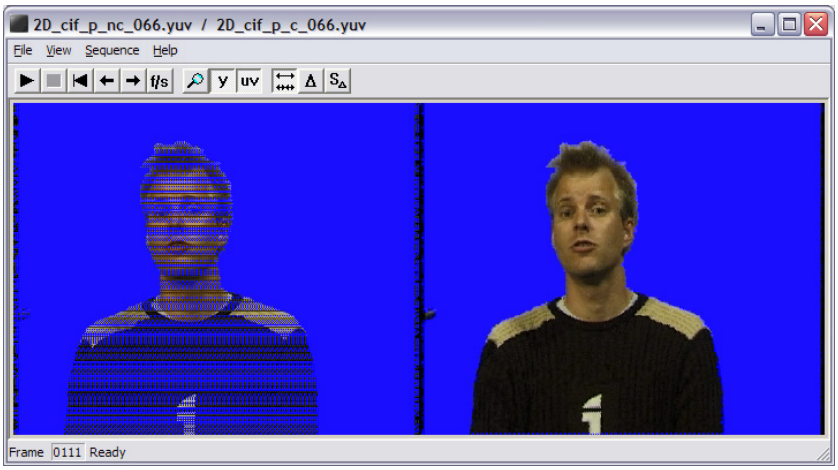

Figure B.6 *2D Interlaced 4-frame coding, one flow protected, 50% loss total. Left: before concealment. Right: after concealment.*

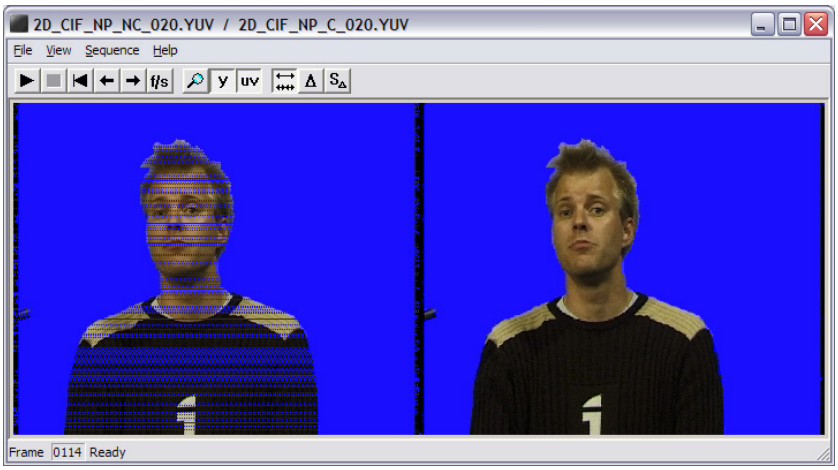

Figure B.7 *2D Interlaced 4-frame coding, single class, 20% total loss. Left: before concealment. Right: after concealment.*

Appendix C

P-AQM and VBR rate control implementation: pseudo code

Ns-2 implementation of P-AQM and VBR rate adaptation expressed in pseudo code

C.1 P-AQM + ECF & ERF pseudo code

The P-AQM router algorithm is implemented in C/C++ the file p_aqm.cc found at `www.item.ntnu.no/~arnelie/Evalvid-RA.htm` for use in ns-2 simulations. The variable names used here in the pseudo source code deviates somewhat from the C/C++ implementation, but conforms with the variable names used in Paper D and Paper F. However, they may include additional information to enhance the readability. Initialization of variables is skipped. The difference between ECF and ERF mode is easily seen since both versions are inserted but separated with a C-syntax look-alike `#ifdef` switch.

The ECF version monitors the flows RTTs by sending periodic ping packets. The feedback `ECF_feedback` is sent as a 32 bit floating point number in ICMP SQ packets. If this number is less than 1.0, it is a multiplicative decrease (MD) ratio. If it is larger than 1.0, the number indicates an additive increase in bit/s. The ECF version avoids using new packet types at the cost of added signaling.

The ERF version can be implemented with the same signaling mechanisms as ECF, but can also skip this completely and instead use packet header fields for insertion of the 32 bit `ERF_feedback`. This could be e.g. RCP packet headers, or any proprietary other solution. If this header allows, it can also have a field for storing the RTT calculated by the end systems, to assist the P-AQM's router need for these values.

The `get_active_udp_flows` function is not explained. As currently implemented in `p_aqm.cc`, it is a full state engine that counts all active flows per ECF or ERF time window period. If a recently active flow is not present with any packets in the next window, it is not counted. The less complex "zombie" flow counting estimation method of SRED was tested but found to have inadequate performance.

The "inner loop" period is static and in the order of 10 ms, while the ECF and ERF periods are dynamic and in the order of 500 ms. The `u_tcp` and u_udp ratios (between 0 and 1) are used in the enqueue function of `p_aqm.cc` to mark or drop incoming packets. Typically, in steady state, none of the UDP packets are dropped because all flows conform to the rate adaption mechanism of the "outer loop". However, the TCP packets are dropped or ECN marked to control normal TCP congestion control mechanism. If found desirable, the `u_tcp` calculation can be replaced by the RED algorithm, modified to take into account a variable output capacity `c_tcp`.

The core AQM pseudo code is here:

```
for (each "inner loop" k) do {
      r_tcp[k] = N_arrivals_tcp * gamma + r_tcp[k-1] * (1 - gamma);
```

```
      r_udp[k] = N_arrivals_udp * gamma + r_udp[k-1] * (1 - gamma);

      tcp_flows = get_active_tcp_flows(k);
      udp_flows = get_active_udp_flows(k);
      flow_ratio = tcp_flows / (tcp_flows + udp_flows);

      c_tcp = c * flow_ratio;
      c_udp = c - c_tcp;

      /* only c_tcp < r_tcp[k] included here: prob. for packet survival*/
      u_tcp = (c_tcp / r_tcp[k]) + (N*_tcp - N_tcp[k])*K/r_tcp[k];

      /* only c_udp < r_udp[k] included here: prob. for packet survival*/
      u_udp = (c_udp / r_udp[k]) + (N*_udp - N_udp[k])*K/r_udp[k];

      if (#of k inner loops since last ECF > ECF_period) {
            r_stable = monitored UDP rate in tau_est period;

            eUDP = (10% of N*_udp) * (1-flow_ratio) - N_udp[k];
            tau_1 = monitored average RTT;
            tau_2 = ECF period - tau_1;
            d = tau_1 + tau_2;

#ifdef P-AQM+ECF
            /* The ECF feedback calculated based on Paper F eq. (1) with
            modification in scaling as in eq. (3) */
            r_ecf = r_stable + alpha * (c_udp-r_stable) + beta * eUDP/ d;

            if (r_ecf <= r_stable)
                  ECF_feedback = r_ecf / r_stable; /* MD */
            else
                  ECF_feedback = (r_ecf - r_stable)/ udp_flows; /* AI */

#else ifdef P-AQM+ERF
            /* The ERF feedback calculated based on Paper F eq. (3) */
            R_erf[n+1] = R_erf[n] + (alpha * (c_udp - r_stable)
                        + beta * eUDP / d) / N_est[n];
            N_est[n+1] = c_udp / R_erf[n+1];
            ERF_feedback = R_erf[n+1]; /* Absolute rate per flow */
            n = n + 1;
#endif
      }
}
```

In addition, there are per packet functionality associated with packet arrivals and departure, i.e. the enqueue and dequeue functions, respectively. The enqueue pseudo code is here:

```
for (each arriving packet for this outbound link) do {
      if (packetType belongs to TCP queue) {
            lucky = random_uniform(0,1);
            if (lucky < u_tcp)
                  enqueue_tcp(packet);
```

```
            else {
                  if (ECN active)
                        ecn_mark(packet);
                        enqueue_tcp(packet);
                  else
                        drop(packet);
      }
      else /* packet belongs to UDP queue */ {
            lucky = random_uniform(0,1);
            if (lucky < u_udp)
                  enqueue_udp(packet);
            else {
                  drop(packet);
      }
}
```

The main part of the dequeue function is the fair queue scheduler, combining the two queues for output. In the C/C++ implementation, it is actually built up as a discrete tabulated form, mapping a `flow_ratio` to a pattern of zeros and ones collected in an array, which distribution corresponds to `flow_ratio`. Thus, its granularity is packets, not bytes. Future work should include a FQ RR scheduler that take into account the bytes spent as a weighting factor. The dequeue pseudo code is here, showing the simple use of the FQ RR array system:

```
while (there are still packets in queue) do {
      type = fair_queue_scheduler[j]; /* 0 means UDP, 1 means TCP */
      if (type == 0)
            dequeue_udp(packet);
      else
            dequeue_tcp(packet);
      j++;
      if (j > max_length)
            j = 0;
}
```

C.2 The RA-SVBR pseudo code

The RA-SVBR algorithm, which is the rate adaptive version of the SVBR by Hamdi et al., is implemented in C/C++ in vbr_rateadapt.cc found at `www.item.ntnu.no/~arne-lie/Evalvid-RA.htm` for use in ns-2 simulations. Here are its main parts expressed in pseudo code. The variable names chosen are kept as close to the original implementation and Paper E as possible, but deviations are inserted to assist better readability, where found appropriate.

```
/* In initialization phase */
for (i = all Quantizer scale variants of the video) do {
      for (j = all frames in file) do {
            memory[i][j++]=load_from_file(videoFile_Qscale[i][j++]);
      }
      i++;
}

/* In running code, once per new video frame: */
q = 8; /* chosen as scaling quantizer in TCL code */
for (each new frame period time k) do {
      Q = current quantizer scale;
      k = current time position in video;
      size = memory[Q][k]; /* size in bytes of next frame to go */
      x = #of full MTU packets;
      y = bytes left for last packet;
      /* all packets for this frame are sent back-to-back: */
      for (i=1:x) do {
            send_message_of_size(MTU+header);
      }
      send_message_of_size(y+header);
      R[k] = R[k-1] + size;
      if (this is last frame of current GOP) {
            r = new_rate(ECF_feedback || ERF_feedback);
            b = new_Leaky_Bucket_size(ECF_feedback || ERF_feedback);
            X[k+1] = Equation 1 in Paper E;
            e1 = 1 - (1 - X[k+1]/b)^5; /* eq. from Hamdi's thesis */
            e2 = (X[k+1]/b)^5; /* eq. from Hamdi's thesis */
            r_open = r * Q / q; /* Based on Eq. 2 Paper E */
            if (r_open > r)
                  Rtemp = (1-e1)*r_open+e1*r; // Eq. 8 in Hamdi paper
            else
                  Rtemp = e2*r_open+(1-e2)*r; // Eq. 9 in Hamdi paper
            /* The quantizer scale next GOP is calculated based on
                  Equation 2 in paper E: */
            Q = closest_integer(q * r_open / Rtemp);
      }
}
```

Finally it should be mentioned that the implemented code can be compiled for *bursty* or *smooth* operation, as well as with *a priori* or *non a priori* next GOP complexity knowledge. The *bursty* variant submits all packets belonging to a frame directly into the network

with no extra delay (no transmit buffer smoothing). The *smooth* mode submits the packets with equal inter-departure time gap, smoothed over one frame interval. The a priori variant uses the knowledge of the size of the next GOP period to make an improved selection of the next GOP quantizer scale value, in order to meet the leaky bucket constraints. Thus the a priori method has somewhat lower bit rate variance compared to the non a priori variant, where the most significant difference is observed during scene change events. Typical video conference sessions have none or very limited number of scene changes. All use of Evalvid-RA SVBR implementation in Paper E and F employs the bursty variant of the a priori method (lowest possible delay for typical video conference type of application).