** GI 2014 **

Experimental TCP Performance Evaluation in an ABR Access Environment

JORDI NELISSEN*
nelissej@rc.bel.alcatel.be

Alcatel Alsthom Corporate Research Centre
Network Architecture
Traffic Technology Team
Francis Wellesplein 1, B-2018 Antwerp
Belgium

ARJEN HENDRIKSE**
hendrika@sebb.bel.alcatel.be

University of Twente
Tele-Informatics and Open Systems group
PO Box 217, 7500 AE Enschede
The Netherlands

** Corresponding author*
*** Now with Alcatel Alsthom*

Abstract : *The Available Bit Rate (ABR) Service Category has a great potential in avoiding congestion in data communication networks. ABR relies on a closed loop flow control mechanism, attempting to push buffer requirements towards the boundaries of the ABR aware network regions, be-it to the ATM end-systems in an ATM end-to-end environment or to the edge routers in a router based environment. On the other hand, Internet traffic is most often flow controlled at layer 4 by means of the Transmission Control Protocol (TCP). Consequently it is of great importance to investigate the potential of ABR in carrying TCP traffic efficiently. Most work reported in literature is based on simulation studies. As interesting these simulation studies are, they can never capture all the real-world effects. The availability of an ABR ATM multiplexer and ABR Network Interface Cards (NIC) gave us the unique opportunity to acquire hands-on experience in dimensioning an ABR system for Internet traffic. This paper studies the performance and buffer requirements of TCP traffic multiplexed with priority VBR background traffic in an ABR queue. We applied several modes of ABR responsiveness by changing the*

ABR connection parameters. We optimised the ABR node parameters, varied the amount of TCP sources and changed the VBR traffic profile. The outcome of this study is a set of guidelines for dimensioning an ABR access configuration for the efficient transport of Internet TCP traffic.

1. Introduction

Several papers report on simulations done with TCP over ABR [1], [2], [3], [4], [5], [6], [7], but there are very few or even no real-world experimental studies that investigate the interaction of TCP flow control with the ABR control loop. While simulations are very interesting to model large networks at relatively low cost, they can never capture all the aspects of real world TCP implementations and ATM equipment. Performing measurements with ABR equipment however remains problematic due to the lack of commercially available ABR Network Interface Cards (NICs). During our measurements a simple ABR configuration, developed within the framework of [8], was used to investigate the performance of TCP over ABR and UBR in the presence of a priority VBR background traffic. Different VBR profiles were applied to create variability in the available bandwidth for the low priority UBR or ABR traffic. Various ABR connection parameter settings were utilised in order to quantify the impact of the ABR responsiveness on the performance of one or several TCP flows. And finally, ABR node parameters were tuned in order to achieve optimal TCP performance.

Section 2 provides a condensed description of the Transmission Control Protocol (TCP) and ABR flow control mechanism, followed in section 3 by a description of the network configuration that is used in our TCP over ABR experiments. Section 4 discusses the ATM Access Service Multiplexer which is the core of our configuration and section 5 describes our TCP over ABR parameter tuning approach.

2. Transmission Control Protocol and the Available Bit Rate Service Category

TCP's flow control and packet retransmission mechanism provides a reliable byte stream service for data applications over a best effort network. One can distinguish two types of flow control : flow control imposed by the receiver and flow control imposed by the sender. Flow control imposed by the receiver is achieved by a so called sliding window mechanism. The maximum amount of outstanding data in the network is then limited by the receiver's advertised window, merely to avoid buffer overflow at the receiver side. Flow control imposed by the sender is accomplished by a combination of several mechanisms, namely Slow Start, Congestion Avoidance, Fast Retransmit and Recovery (FRR) [9]. These mechanisms maintain and update a second window, the Congestion Window, which is the amount of data a TCP source is willing to send into the network. Loss of packets in a network is an indication of congestion and triggers a TCP source to reduce its sending rate by reducing its Congestion Window. A detailed description of all the above mechanisms and the way they update the Congestion Window can be found in [10].

The Available Bit Rate service category [11] uses a closed loop flow control mechanism in order to match the ABR source rate to the available ABR capacity in the network. Feedback is given to the source about the state of the network, where upon the source has to adjust its rate accordingly. ABR is using special cells, called Resource Management (RM) cells, to probe the network and to return the state at that moment. Each source periodically sends these RM cells intermixed with normal data cells. The destination turns around all the RM cells and sends them back to the source. Each time an RM cell is passing an ABR node, the node may write information about the state of the network in that cell. When the source gets an RM cell back, it will decrease, increase or maintain its rate according to the information in that cell. The

Explicit Rate (ER) class of ABR mechanisms is believed to produce the best performance. In contrast with binary feedback control mechanisms, ER-ABR tells the source exactly what the network is willing to accept. Besides monitoring the load and detecting pending congestion, each switch now also computes the maximum Allowed Cell Rate (ACR) for each of its active connections. This value may be written in the Explicit Rate field of a backward RM cell and is used by the source to adjust its Current Cell Rate (CCR).

3. Description of experimental setup

Our experimental set-up consists of a LAN configuration with two PC based ABR end-systems interconnected by means of an ABR Access Service Multiplexer (ASM) (Figure 1). The traffic is monitored in the downstream direction to capture TCP traces and ABR RM cell information. A generator produces an ON/OFF VBR pattern which is handled by the high priority VBR buffer and bottlenecks the low priority TCP over ABR connections. In case of congestion in the ASM, Explicit Rate feedback information is written in the ER field of a Backward RM cell (BRM) in order to quench the TCP/ABR source and avoid queue overflow. The next section describes the ABR Explicit Rate implementation used in the ASM in more detail.

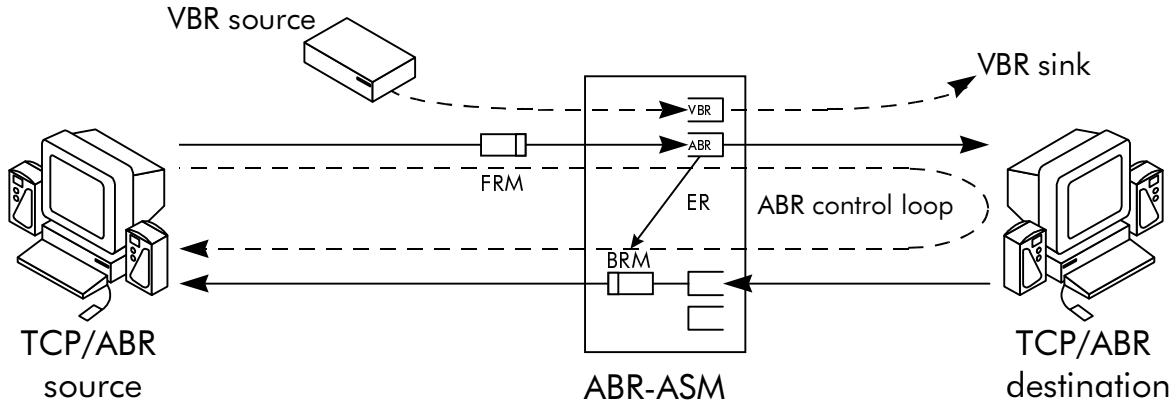

Figure 1: Experimental set-up for TCP over ABR experiments

4. The ABR Access Service Multiplexer

The ABR Access Service Multiplexer consists of a *Network Termination* (NT) card interfacing with the ATM core network and one or more *Line Interface Modules* (LIM) terminating the *User Network Interfaces* (UNI), as depicted in Figure 2 and described in more detail in [12]. The LIMs are connected with the NT via an internal bi-directional ATM cell bus, which operates up to the speed of the NT. Since, in our measurements, we only used a single Line Interface Module, the ATM cell bus behaves as a direct link between the NT and the LIM and hence the bus mechanism does not come into play.

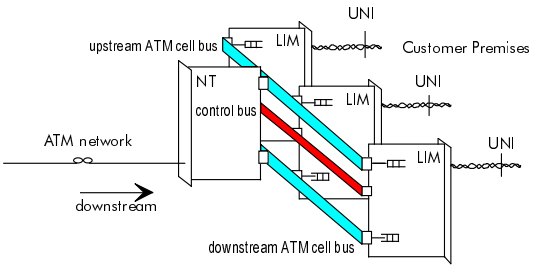

Figure 2: The internal structure of the ABR Access Service Multiplexer with 3 LIMs

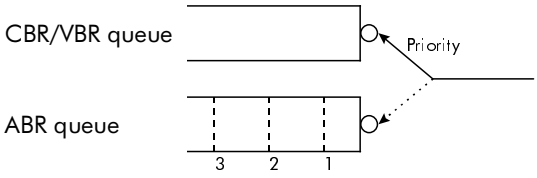

Figure 3: Queue thresholds for congestion control

The ABR feedback mechanism implemented on each LIM is a simple threshold based Explicit Rate indication mechanism. Three ABR queue thresholds, as shown in Figure 3, are used to indicate the extent of congestion in the LIM. Higher priority traffic like CBR and VBR, is handled in a separate buffer and is served with a strict priority over the ABR/UBR traffic. During periods of congestion, as soon as the first queue threshold is exceeded, the ASM provides negative feedback information to the source by writing a smaller rate value in the ER field of the BRM cells. If congestion persists, the buffer content exceeds thresholds 2 and 3,

resulting in a sharper rate reduction of the source's sending rate. The calculation of the ER value depends on the *FairShare* and the *VCShare* of that moment and is obtained as follows :

The *FairShare* is the rate that can be used by each ABR VC and is calculated as:

$$FairShare = GB / Number\ of\ active\ VCs \quad (1)$$

Where GB is the Granted Bandwidth, granted by the ASM bus mechanism to an individual LIM. Since in our setup, we only use 1 LIM, GB corresponds to the bottleneck rate of the ASM. *VCShare* is an estimation of the capacity that is used per VC in order to redistribute the unused capacity left by ABR connections using less than their FairShare. The *VCShare* of a VC is calculated as:

$$VCShare_i = (VC_{length,i} / Queue\ occupancy) * GB \quad (2)$$

Where $VC_{length,i}$ is the number of cells in the queue belonging to VC *i.* The ER value that is sent back to source *i* when the first threshold is passed, is now:

$$ER_i = min\ (ER_{BRM,i},\ max\ (\ VCShare_i,\ FairShare)) \quad (3)$$

Where $ER_{BRM,i}$ is the current ER value of the incoming BRM cell of VC *i* and equals the PCR of source *i* as long as no other network elements have changed this field. For the given configuration we only consider a single ABR virtual connection and hence $VC_{length,i}$ is equal to the queue occupancy. The FairShare and the VCShare thus equal the bottleneck rate of the ASM and equation (3) simplifies to:

$$ER = min\ (ER_{BRMi},\ bottleneck\ rate) \quad (4)$$

5. Parameter tuning approach

TCP and ABR are complex protocols with a wide range of configurable parameters. In this paper we adopted the approach listed below for optimising TCP throughput and buffer

requirements in our ABR system. The performance of TCP over UBR is always taken as reference throughout this paper.

1. ABR connection parameter tuning for zero TCP packet loss

   Since cell loss is known to reduce TCP throughput drastically [13], our first concern is limiting the buffer requirements as much as possible in order to avoid cell loss at all times.

2. ABR node parameter tuning for high TCP over ABR efficiency

   The optimization in STEP 1 might result in a sub-efficient ABR flow control mechanism. The ABR feedback mechanism in our multiplexer is governed by three thresholds, th1, th2 and th3. An adequate choice of these thresholds might compensate for the sub-efficiency of the ABR connection parameter tuning in order to reach a good TCP over ABR efficiency.

3. Evaluation of the robustness of the resulting ABR mode

   The result of optimization STEP 1 and STEP 2 is a set of ABR connection and node parameters, which yield a certain ABR mode of operation. In order to evaluate its robustness, this ABR mode is evaluated as a function of the number of TCP sources and for several VBR background profiles.

5.1 STEP 1 : ABR connection parameter tuning for zero TCP packet loss

In our measurements, six different ABR modes differentiated by their responsiveness are considered and are being compared with plain UBR. The responsiveness of an ABR source is characterised by the way it reacts on positive (rate increase) or negative (rate decrease) feedback. In case the ASM indicates an increase in available bandwidth, the sending rate of the ABR source will increase towards this new ER with a speed determined by the Rate Increase Factor (RIF). A rate decrease on the other hand can be imposed by the ASM by means of the ER value in a BRM cell, but may also be triggered by the ABR source itself when the Crm

counter is exceeded. Crm is often referred to as the missing RM cell count, i.e. the number of Forward RM cells the source is allowed to send in absence of Backward RM cells (open loop!). The speed of the rate decrease is then governed by the value of the Cutoff Decrease Factor (CDF) as shown in equation (5) [11].

$$ACR = \max (MCR, ACR - ACR * CDF) \qquad (5)$$

For each ABR mode we focus on two metrics :

- Throughput at TCP level (so called goodput)
- Buffer requirements (i.e. maximum buffer occupancy observed during the experiment)

TCP throughput is measured by means of Netperf [14]. Netperf is a benchmark tool that can be used to measure the memory-to-memory TCP throughput performance emulating a single, unidirectional, greedy TCP source.

5.1.1 VBR and ABR parameter settings

Our TCP over ABR traffic stream is multiplexed with a priority ON/OFF VBR cell stream whose output rate fluctuates between 0 and 16000 cells/s (Figure 4). The output scheduler of the ASM is fixed at 16000 cells/s and consequently the bandwidth available for ABR is zero during VBR ON periods and jumps to 16000 cells/s during VBR OFF periods.

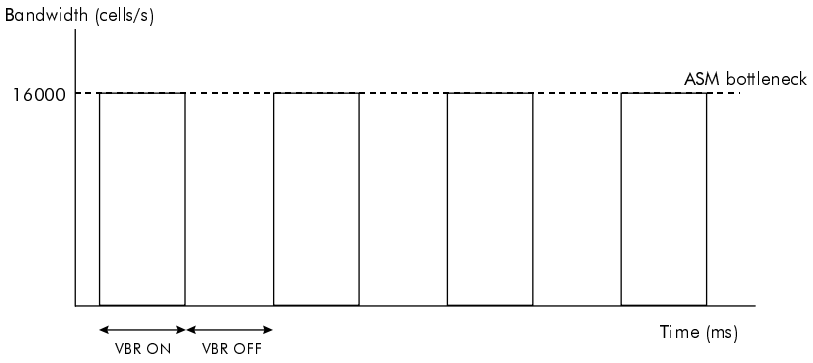

Figure 4: VBR background traffic pattern

Both VBR ON and OFF periods equal 11 ms, resulting in an average VBR background of 8000 cells/s. The mean bandwidth available for ABR then also equals 8000 cells/s. Since the ASM does not measure the VBR background, it has to rely on CAC to determine the average left-over bandwidth for ABR. This average left over bandwidth is calculated as Bottleneck Rate (BR) minus SCR, where SCR is the Sustainable Cell Rate reserved for VBR and BR is the ASM scheduler rate. In case of more severe congestion, a more stringent feedback is given, as shown below :

- queue occupancy < th1 : no congestion -> ER_{new} = BR = 16000 cells/s
- th1 < queue occupancy < th2 : minor congestion -> ER_{new} = BR-SCR = 8000 cells/s
- th2 < queue occupancy < th3 : severe congestion -> ER_{new} = 4096 cells/s
- queue occupancy > th3 : overflow alert -> ER_{new} = 0 cells/s

ABR mode		**RIF**	**Crm**	**CDF**	**RDF**
No VBR sensing	Slow increase	$^1/_{64}$	10	not relevant	not relevant for ER
	Medium increase	$^1/_8$	10	not relevant	not relevant for ER
	Fast increase	$^1/_2$	10	not relevant	not relevant for ER
VBR sensing	Slow increase	$^1/_{64}$	1	$^1/_2$	not relevant for ER
	Medium increase	$^1/_8$	1	$^1/_2$	not relevant for ER
	Fast increase	$^1/_2$	1	$^1/_2$	not relevant for ER

Table 1: Parameter setting for the different ABR modes

Fixed Parameters		
ABR Source	PCR	16000 cells/s
	ICR	100 cells/s
	Nrm	32
	TBE	100 cells
TCP[*]	MTU	1500 bytes
	TCP window	64 Kbyte

Table 2: Relevant fixed parameters for TCP over ABR

Table 1 shows the parameter settings for six ABR modes : Slow, Medium or Fast increase with or without "VBR background sensing". VBR background sensing is emulated by setting Crm equal to one. During a VBR ON period, the bandwidth remaining for ABR is zero and Forward RM cells are blocked in the ABR queue, hence no Backward RM (BRM) cells reach the TCP/ABR source. The lack of BRM cells provides the source with implicit information concerning the bandwidth availability in the ASM. Setting the Crm equal to one and CDF equal to one half, actually provides a very aggressive rate reduction during VBR ON periods. This exotic parameter choice is merely intended to investigate the advantages of an ABR mechanism which "senses" all the details of the VBR background hopefully resulting in smaller buffer requirements. It is not to be seen as a general ABR connection parameter setting since it might effect in unexpected and undesired behaviour when used with other VBR background types than the one used in this experimental scenario.

During the measurements the ABR buffer is varied from 50 up to 2000 cells and the threshold values *th1*, *th2*, *th3* are arbitrarily set to 20%, 40% and 60% of the total buffer respectively. The remaining parameters in all the measurements are kept fixed.

[*] The end-systems were running Windows NT Service Pack 1, which does not support Fast Retransmit and Recovery

5.1.2 Buffer requirements

Figure 5 shows the maximum observed buffer occupancy (fraction of the total buffer) of our single TCP/ABR connection as a function of the ABR responsiveness mode and the ABR buffer size. As a basis of comparison, the buffer occupancy of a single TCP connection carried by a UBR VC is also shown.

As expected, the UBR buffer requirements amount to a full 64 Kbyte TCP window and cell loss already occurs with a buffer of 1000 cells. The buffer requirements of a TCP connection carried by an ABR connection however depend on the ABR mode used but from Figure 5 we learn that these buffer requirements are much smaller compared to UBR. Figure 5 also clearly indicates the difference between the different ABR modes. Looking at the ABR modes without VBR sensing, we notice that the bound on the buffer occupancy is mainly enforced by the second threshold since the feedback of 8000 cells/s at the first threshold is too weak to keep the queue size from growing. For smaller buffers, even the feedback of 4096 cells/s at the crossing of the second threshold is to feeble and a very stringent rate reduction, enforced by the third threshold is needed to keep the queue from diverging. For very small buffer sizes, 50 to 100 cells, the ABR flow control mechanism is being stressed to its maximum and the relative buffer requirements increase up to a point where cell loss cannot be avoided. This effect also occurs for the ABR modes relying on VBR sensing.

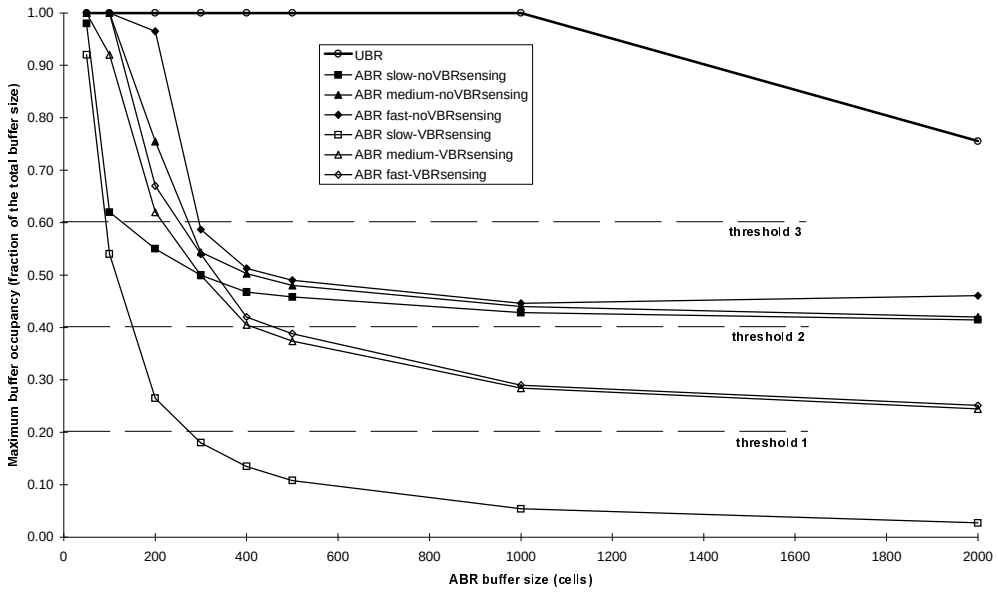

Figure 5: ABR buffer requirements for zero loss as a function of the ABR buffer size and the ABR responsiveness

Figure 6 and Figure 7 compare the behavior of an ABR mode with or without VBR sensing and indicate that without VBR sensing, the highly fluctuating priority VBR stream is perceived by ABR merely as an average background traffic. Important to notice is that in both measurements, no losses were recorded. The ABR mechanism present in the ASM relies completely on its thresholds for giving feedback to the ABR source and due to the non-zero time which is needed for the queue to build up towards one of the thresholds, feedback always lag behind the VBR background and cells have to be buffered during the VBR ON periods. Compared to the VBR sensing mode, this clearly results in larger buffer requirements, as mentioned earlier.

ER feedback based on VBR background sensing can lead to very poor performance if not combined with an aggressive Rate Increase Factor (RIF). Figure 8 shows a trace of the Current Cell Rate for the slow-VBRsensing ABR mode and indicates an average ABR sending rate far

below the 8000 cells/s. The low buffer requirements seen in Figure 5 for this ABR mode are only a reflection of this low efficiency.

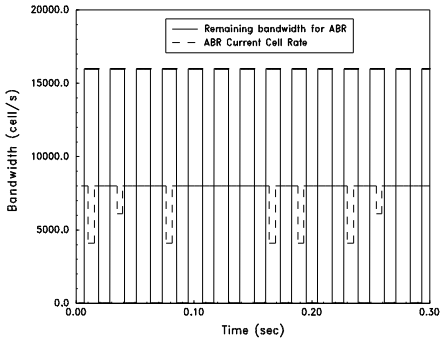

Figure 6: Behavior of the Medium-noVBRsensing ABR mode (500 cell buffer)

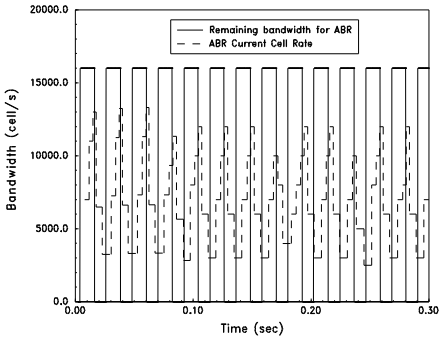

Figure 7: Behavior of the Medium-VBRsensing ABR mode (500 cell buffer)

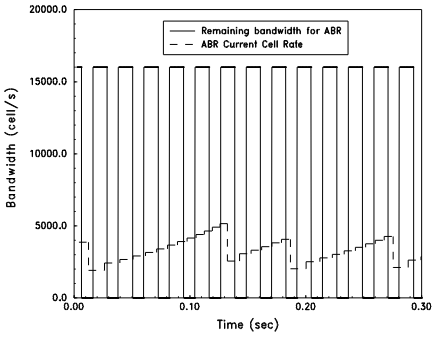

Figure 8: Behavior of the Slow-VBRsensing ABR mode (500 cell buffer)

5.1.3 TCP throughput

In an ABR end-to-end configuration, no rate mismatch between TCP and ABR is to be expected as TCP cannot send at a rate higher than the ABR ACR. However losses can still occur due to a rate mismatch between ABR and the actual available bandwidth. Once TCP encounters losses, it is forced into slow start resulting in large idle times during which it is not able to utilise the available bandwidth provided by ABR. The throughput figure (Figure 9)

clearly points out that for medium to large buffers, where ABR is able to avoid cell loss, TCP over ABR is quasi fully efficient compared to UBR. However once ABR starts to encounter losses, TCP over ABR throughput collapses just as it does with UBR. This throughput collapse might be less severe with a TCP implementation supporting the Fast Retransmit and Recovery mechanism. It is interesting to notice that the ABR modes with VBR sensing hardly perform any better than the ABR modes relying on threshold based feedback, despite their smaller buffer requirements. The reason for the low TCP throughput performance of the slow-VBRsensing ABR mode is caused by the low ABR efficiency, as explained earlier, not due to TCP being unable to use the ABR bandwidth efficiently since no losses occurred during these measurements.

Notice however the good performance of the conservative slow-noVBRsensing ABR mode both from the point of view of throughput and buffer requirements. A slow rate increase ensuring small buffer requirements, hence reducing the risk to experience losses, combined with a threshold based rate decrease ensuring a filled ABR buffer, hence high efficiency seems to guarantee the best TCP over ABR performance in our environment. Fang and Lin [15] show similar good TCP over ABR performance with conservative ABR connection parameters. Consequently we adopt the slow-noVBRsensing ABR mode as the result of this ABR connection parameter tuning.

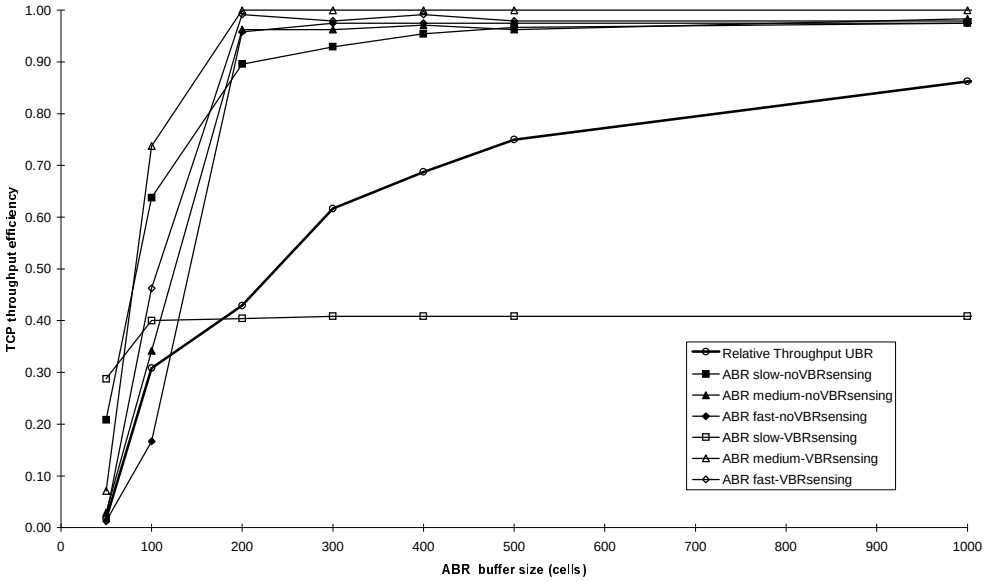

Figure 9: TCP throughput efficiency as a function of the ABR buffer size and ABR responsiveness

5.2 STEP 2 : ABR node parameter tuning to achieve a high TCP over ABR efficiency

The result of STEP 1 is a choice of the ABR connection parameters, resulting in the slow-noVBRsensing ABR mode, which meets the most important requirement for TCP i.e. avoid packet losses, even down to a buffer size of 50 cells (cfr. Figure 5). The next step is to tune the ABR node parameters aiming at a higher ABR efficiency, hoping to compensate for the compromises on ABR efficiency made during STEP 1.

th1=5 th3 \ th2	10	20	30
25	0.17	0.17	NA
35	0.59	0.47	***0.05***
45	0.62	0.61	***0.11***

th1=15 th3 \ th2	10	20	30
25	NA	0.17	NA
35	NA	0.44	***0.15***
45	NA	0.61	***0.18***

th1=25 th3 \ th2	10	20	30
25	NA	NA	NA
35	NA	NA	***0.14***
45	NA	NA	***0.14***

Table 3: TCP over ABR throughput efficiency for slow-noVBRsensing ABR mode and an ABR buffer of 50 cells

In case of our ABR ASM, the ABR node parameters are the three buffer thresholds, th1, th2 and th3. We performed a full factorial analysis on the three dimensional threshold parameter space, obviously limiting ourselves to the region where th1 < th2 < th3. The metric used in this analysis is the throughput efficiency of TCP over the slow-noVBRsensing ABR mode for a buffer size of 50 cells. As indicated in Figure 9, this efficiency before threshold tuning is only 21%. The results of the threshold tuning are shown in Table 3. The figures in bold-italic correspond to threshold settings where cell loss occurs. The measurements reveal an efficiency increase from 21% up to 62 % by setting the thresholds th1, th2 and th3 respectively equal to 5, 10 and 45 cells, or 10%, 20% and 90% of the total buffer size instead of the arbitrary 20%, 40% and 60%. In the three dimensional parameter space, the point (th1, th2, th3) equal to 10%, 20% and 90% of the total buffer size seems to corresponds with a stable operation point since the surrounding points also yield a high TCP throughput efficiency. Consequently, the (th1, th2, th3) = (10%, 20%, 90%) ABR node parameter setting is our preferred choice.

5.3 STEP 3 : Evaluation of the robustness of the resulting ABR mode

The logical step after parameter optimisation is the testing of the robustness of the obtained ABR connection and node parameter setting. For this we apply different VBR ON/OFF profiles and check the performance of TCP over ABR as a function of the number of TCP sources.

5.3.1 Effect of the VBR background

During VBR ON periods, the priority VBR background completely blocks the ABR traffic in the multiplexer causing ABR queue build-up. These queues are cleared during VBR OFF periods. We investigated the TCP over ABR throughput efficiency as a function of the VBR ON duration. A longer VBR ON period results in a longer duration of the congestion period, which in turn puts higher pressure on the buffer requirements. We performed measurements with three VBR ON periods : 11 ms, 25 ms and 53 ms. Each VBR ON period resulted in a different average VBR background, and consequently in a different SCR to be taken into account in the CAC of the ASM. As indicated in Figure 10, as long as losses can be avoided, this has very small impact on the TCP over ABR throughput performance. However when losses are unavoidable, which is the case with a very small buffer of 50 cells, TCP over ABR throughput collapses, as expected.

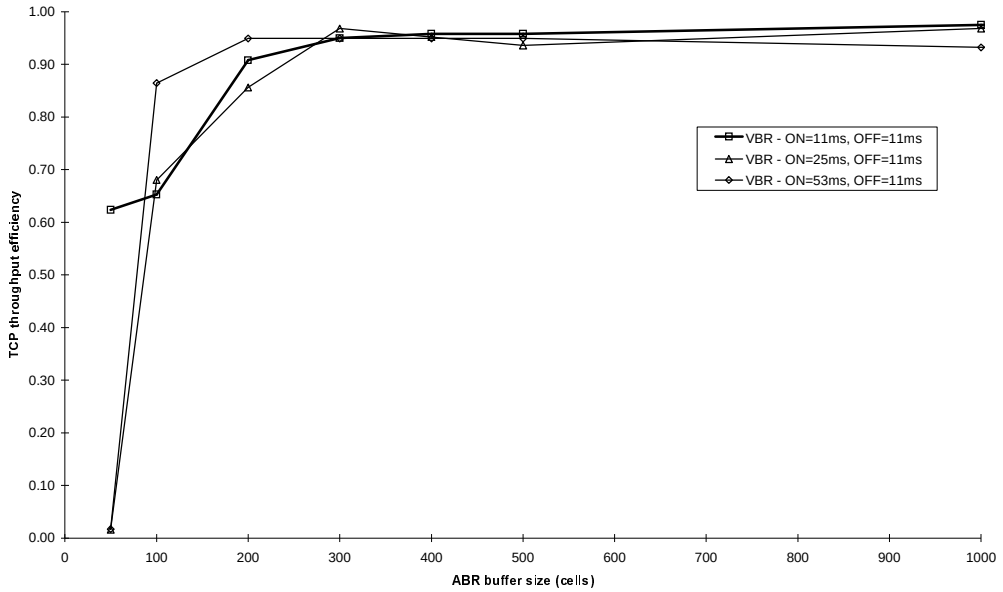

Figure 10: TCP throughput efficiency as a function of the ABR buffer and the VBR ON duration

5.3.2 Number of TCP sources

We considered the situation of a shared ABR VC, i.e. a VC carrying more than 1 TCP connection. Similarly to the UBR VC carrying a single TCP connection, with 3 TCP connections the buffer requirements for zero loss amount to the total sum of three TCP windows, i.e. in our case 192 Kbyte, or about 4000 cells. Consequently cell loss occurs for all buffer sizes considered in our measurements. However the buffer requirements of an ABR connection carrying several TCP connections are independent of the number of TCP sources (Figure 11).

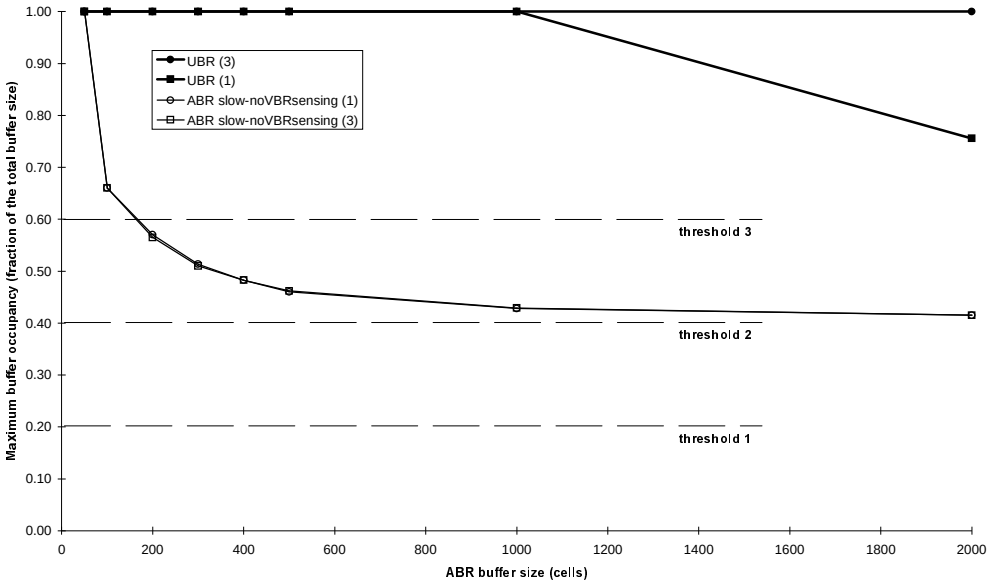

Figure 11: ABR buffer occupancy as a function of the ABR buffer size for 1 and 3 TCP sources

Figure 12 shows the aggregate throughput of 3 TCP connections carried by an ABR or UBR connection. We learn that ABR's capability in avoiding cell loss is independent of the number of TCP sources, even for very small ABR buffers down to 50 cells and that the aggregate throughput is independent of the number of TCP sources. Despite the larger buffer requirements, UBR performs better with increasing number of TCP sources. This is explained by the reduction of the link idle time when more TCP sources are active. During Time-out and Slow Start of one source another takes over and keeps the link from draining.

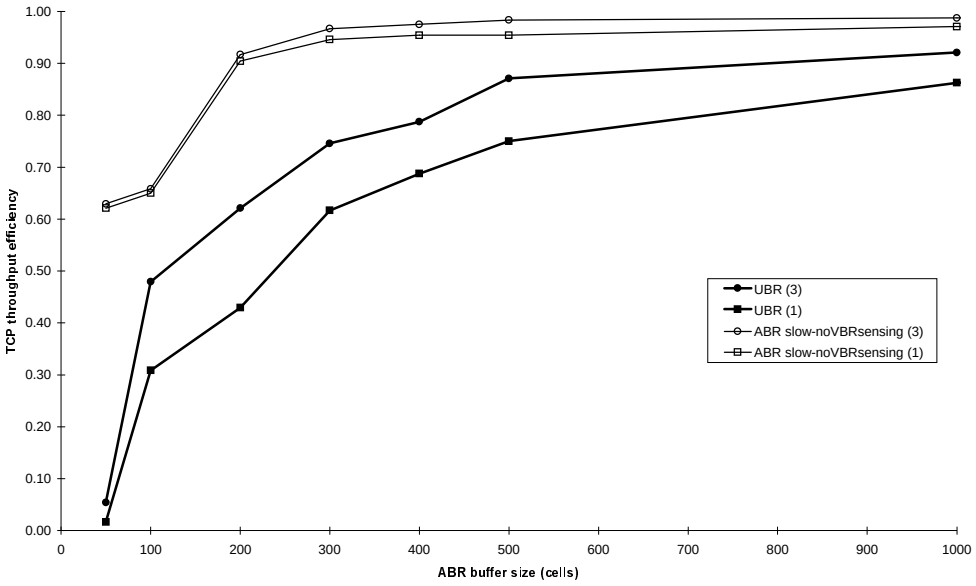

Figure 12: TCP throughput efficiency as a function of the ABR buffer size for 1 and 3 TCP sources

6. Conclusions

In this paper we propose a general approach for the parameter tuning of an Explicit Rate ABR system in order to efficiently support Internet TCP traffic in a low delay access environment. Since cell loss is known to reduce TCP throughput drastically, the first concern is to keep the buffer requirements within the bounds of our total buffer size by means of an appropriate ABR connection parameter choice. We achieve this with a slow Rate Increase combined with a moderate ER feedback, i.e. ER feedback based on the average VBR background instead of nervously reacting on every VBR rate change. Strictly aiming at minimal buffer requirements might however result in a sub-efficient ABR flow control mechanism. To compensate for the compromises on ABR efficiency, we performed an optimisation of the ABR node parameters, resulting in a considerable increase of the TCP over ABR throughput. Clearly this optimisation

of node parameters is completely dependent of the ABR implementation. As a last step, an evaluation of the robustness of the obtained ABR flow control mechanism is required.

Our observations are that in a low delay access environment, even a simple threshold based ABR Explicit Rate notification implementation like the one used in our measurements assures nearly full TCP efficiency with ATM buffers much smaller than those required for UBR. The ABR buffer requirements show to be independent of the number of TCP connections carried by the ABR VC, which is an interesting behavior if LAN traffic needs to be carried. To make the ABR mechanism robust with respect to the VBR background profile, a minimum amount of buffering is required. For very small ABR buffers and long VBR ON periods, the ABR flow control mechanism fails and TCP throughput collapse is observed similar to UBR.

7. Acknowledgments

The results obtained in this project and development work undertaken was only possible with the financial support of the Flemish Institute for the Promotion of Scientific and Technological Research in the Industry (IWT). In particular, the authors would like to thank R. Hoebeke for his technical support and advise during the measurements.

8. References

[1] B. Feng, D. Ghosal, N. Kannappan, “Impact of ABR Control on the Performance of TCP-Tahoe and TCP-Reno”, *Proceedings Globecom ’97*

[2] S. Manthorpe, J. Le Boudec, “A comparison of ABR and UBR to support TCP traffic”, DI Technical Report 97/224, April 1997, available from http://lrcwww.epfl.ch/PS_files/TR97_221.ps

[3] H. Li, K. Siu, H. Tzeng, C. Ikeda, H. Suzuki, “A Simulation Study of TCP Performance in ATM Networks with ABR and UBR Services”, *Proceedings Infocom ’96*

[4] P. Johansson, E. Wedlund, J. M. Karlsson, “Interaction Between TCP Flow Control and ABR Rate Control”, *Proceedings IEEE ATM Workshop ’97*

[5] I. Iliadis, “Performance of TCP Traffic and ATM Feedback Congestion Control Mechanisms”, *Proceedings Globecom ’96*

[6] G. Hasegawa, H. Ohsaki, M. Murata, H. Miyahara, “Performance of TCP over ABR Service Class”, *Proceedings Globecom ’96*

[7] B. Vandalore, S. Kalyanaraman, R. Jain, R. Goyal, S. Fahmy, “Worst Case Buffer Requirements for TCP over ABR”, *submitted to SICON ’98*, available from http://www.cis.ohio-state.edu/~jain/papers.html

[8] “IWT project : Available Bit Rate” sponsored by Flemish Institute for the Promotion of Scientific and Technological Research in the Industry (IWT)

[9] W. R. Stevens, “TCP Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery Algorithms”, *IETF, Network Working Group*, RFC 2001, January 1997

[10] W. R. Stevens, “TCP/IP Illustrated, Volume 1: The protocols”, Addison-Wesley, 1994.

[11] ATM Forum, “Traffic Management Specification Version 4.0“, *ATM Forum Technical Committee*, AF-TM-0056, April 96.

[12] R. Hoebeke, J. Witters, T. Schillemans, “ABR functionality in an ABR Service Multiplexer : Implementation, Experiments and Simulations”, *to be presented at Syben 98, International Symposium on Broadband European Network*, 18-22 May 98, Zurich, Switserland

[13] A. Romanow, S. Floyd, “Dynamics of TCP over ATM Networks”, *IEEE JSAC*, vol. 13, No 4, pp.633-641, May 1995.

[14] More information about Netperf can be found at : http://www.cup.hp.com/netperf/NetperfPage.html

[15] C. Fang, A. Lin, “TCP performance in ATM Networks : ABR Parameter Tuning and ABR/UBR Comparisons”, *Proceedings SICON ’97*