18

Simulated performance of TCP over UBR and ABR networks

Mika Ishizuka, Hideo Kitazume and Arata Koike
NTT Telecommunication Networks Laboratories
3-9-11 Midori-cho Musashino-shi, Tokyo, 180, Japan
Telephone: +81 422 59 4441, Fax: +81 422 59 3290
email: {mika, kitazume, koike}@hashi.tnl.ntt.co.jp

ABSTRACT

The performance of the Transmission Control Protocol (TCP) over the Unspecified Bit Rate (UBR) and the Available Bit Rate (ABR) ATM service classes was investigated by simulation. These service classes are both for data communication but have different characteristics. It is therefore important to reveal the performance of TCP over UBR and ABR.

Since UBR service does not guarantee the cell loss ratio, huge buffers are needed in the ATM switches to prevent cell loss, which degrade the performance seriously in the TCP layer. If many connections are multiplexed, it is difficult to have large enough buffers in the ATM switches. Furthermore, huge buffers are undesirable for delay-sensitive TCP applications. Therefore, cell loss may be inevitable with UBR connections. Simulation showed that when there is cell loss, TCP performs better when the timer granularity is set finer because finer granularity means faster detection of data loss.

When the ABR service class is used, feedback rate flow control is used to minimize cell loss. Simulation showed that switches with Explicit Forward Congestion Indication mode, which have simple architecture, have good performance in small networks but the performance is degraded when many connections are multiplexed or when the round trip time becomes long. We recommend to set low Peak Cell Rate in such cases. Otherwise, switches with Explicit Rate mode are effective even in the networks with many connections and long round-trip times.

Whichever switch architecture is used, simulation showed that huge buffers cause a large variance in the Round-Trip-Time (RTT). As a result, TCP cannot estimate Retransmission-Time-Out correctly and retransmits TCP segments spuriously.

Keywords

ABR, TCP, window control, rate control, timer granularity

1. INTRODUCTION

New high-speed-transport architecture such as Asynchronous Transfer Mode is expected to enable high-speed data communication over Wide Area Networks (WANs).

TCP/IP, the most common protocol for data communication, is mainly used for Local Area Networks (LANs) or for low-speed internetworking between LANs. Thanks to the introduction of high-speed transport architecture, TCP/IP is coming to be used for high-speed WANs based on ATM. However, the use of TCP/IP over high-speed WAN environment may cause various problems, resulting from its poor performance (Kleinrock, 1992)

The performance of ATM is generally measured by the cell-loss ratio or the cell transfer delay characteristics in the ATM layer (Hasegawa, 1995 and Koike, 1996); where the performance of the TCP is measured by the throughput or delay in the TCP layer. Since performance characteristics may be different between these layers, it is important to evaluate whether the overall performance of "TCP over ATM" will meet users' requirements. There is some research on this area. Ramanow (1994) and Lakshman (1996) discuss strategies of cell dropping, Lin (1995) gives experimental comparison of TCP over an actual ATM network and TCP over an actual ethernet or FDDI network.

The effect of the interaction between the TCP and the ATM layers must be carefully considered, because the controls and parameters in each layer must interact, even though they are independently designed. This interaction has not been sufficiently investigated in previous papers. We have thus investigated the simulated performance of TCP over an ATM networks from the viewpoint of such an interaction.

ATM has two service categories: guaranteed and best effort (The ATM Forum, 1996). Best effort is suitable for data communications, which are not delay sensitive. The best effort category has two service classes: Unspecified Bit Rate (UBR) and Available Bit Rate (ABR). Since UBR service does not guarantee the cell-loss ratio, cell loss is inevitable in UBR connections, which may seriously degrade the performance in the TCP layer. In the ABR service class, a Source End System (SES) calculates the Allowed Cell Rate (ACR) based on the congestion information received from the network. The cell rate of the SES is then reduced to less than the ACR based on this feedback in order to reduce the cell-loss ratio. The ABR service class is thus more effective for data communication. When the ABR service class is used, we must clarify two points:

1. the interaction between the window-based flow control in TCP and the rate-based flow control in ABR, and
2. in what networks is TCP over ABR more effective than TCP over UBR.

In this paper we report the effectiveness of the ABR service class for TCP/IP communication focusing on these points.

2. SIMULATION MODEL

The behavior of TCP and ATM protocols and a model of end systems are described in this section.

2.1 The protocol stacks of TCP and ATM

The protocol stacks of TCP and ATM are shown in Figure 1. Application data is sent to the TCP layer as a data stream. In the TCP layer, it is decomposed into segments and a TCP header is added to each segment. In the IP layer, the segment is encapsulated into an IP packet. In ATM Adaptation Layer 5 (AAL5), a trailer is added and then segmented into ATM cells.

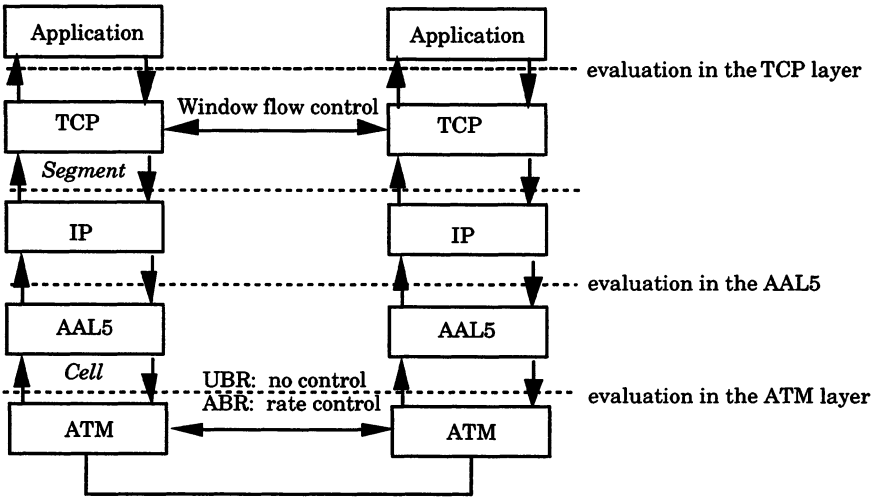


Figure 1 Protocol stacks of TCP and ATM

2.2 Behavior of TCP

TCP is a transport protocol, which guarantees end-to-end transmission without errors. TCP flow control, error detection, and recovery are as follows:

1. Window-based flow control
2. When a Destination End System (DES) receives data correctly, it sends an Acknowledgment segment (Ack) to the Source End System (SES).
3. An SES estimates Round-Trip-Time (RTT) and calculates Retransmission-Time-Out (RTO) based on it. We adopt Karn's algorithm and exponential back-off to calculate RTO.

4. When an SES does not receive an Ack within the RTO, the segment is considered to be lost (called as a time-out). Then the SES retransmits all segments following the lost unacknowledged segment (go-back-N).
5. An SES adjusts TCP window size dynamically during congestion (Slow start, Congestion avoidance).

Details about TCP are in Barden (1989). The TCP model we used follows an implementation in the 4.3Berkeley Software Distribution (BSD)-Tahoe release without fast retransmission.

2.3 Behavior of UBR and ABR

Behavior of UBR and ABR is described below:

UBR

An SES can transmit cells at the Peak Cell Rate (PCR) whenever it has cells. The network does not guarantee any Quality of Service (QoS).

ABR

An SES calculates the Allowed Cell Rate (*ACR*) based on the feedback information from the network. If the cell rate of an SES is reduced to less than the *ACR* according to the feedback, the cell loss ratio will be small.

ABR rate control is briefly summarized below. We use both the Explicit Forward Congestion Indication (EFCI) mode and the Explicit Rate (ER) mode. Our implementation is completely based on the specifications (The ATM Forum, 1996, and Barnhart 1995).

1. An SES generates an RM cell and sends it after *Nrm* data cells are transmitted. The SES sets *PCR* to the *ER* field in the RM cell.
2. EFCI switches set the *EFCI* bit of the data cell when its queue length exceeds the threshold.
3. When the RM cell reaches the DES, it returns the RM cell to the SES. The *CI* bit in the RM cell must be set to 1 when the *EFCI* bit in the last received data cell was 1.
4. ER switches calculate ER for the Queue (*ERQ*), which is the rate that they can support. When ER switches receive an RM cell and when the value of the *ER* field in the RM cell is greater than *ERQ*, *ERQ* is placed in the *ER* field.
5. When an SES receives an RM cell, the SES calculates *ACR* as described below:
   if the *CI* bit in the RM cell=1

   $ACR=ACR-ACR\times RDF$

   $ACR=\max(MCR,\min(ACR,ER))$

RDF: Rate Decrease Factor

if the *CI* bit in the RM cell=0

$ACR=ACR+PCR\times RIF$

$ACR=\min(PCR,\min(ACR,ER))$

RIF: Rate Increase Factor

2.4 Models of end systems

The models of the end systems are implementation dependent, and this greatly affects end-to-end performance. We built up our TCP over ATM simulator by using BONeS, even though TCP and ATM modules are provided in BONeS. As mentioned above, the models of end systems affects the performance much. Nevertheless, the implementation in BONeS is not so clear for us. Thus, we made them ourselves.

Our implementation completely follows the standards for TCP and ATM. Some results from experiments show validity for our implementation; however, details in end systems such as data copy or check sum are not completely modelled, because its mechanism is too complicated.

We describe the models we used in our simulation briefly.

The model of the SES is shown in Figure 2.

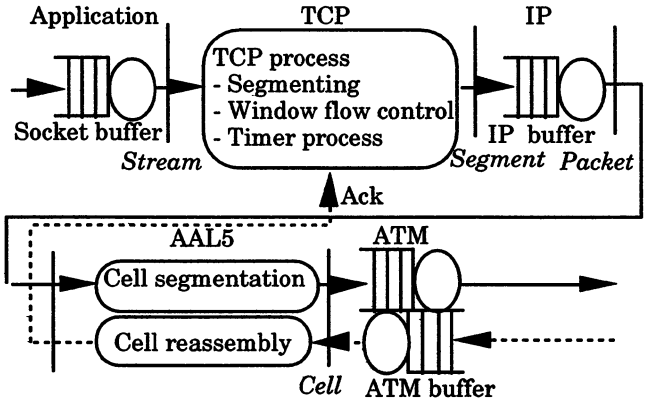


Figure 2 Model of the SES

- An infinite source model is used to characterize Application-level traffic. This model emulates File Transfer Protocol (FTP). This traffic is queued in the socket buffer.

- The service time for generating a TCP segment is exponentially distributed and its mean is 1 ms.
- The timer for a TCP segment starts just before it is queued in the IP buffer.
- When all cells in one packet have been sent, AAL5 gets another packet from the IP queue and segments it into cells.

The model of the DES is shown in Figure 3.

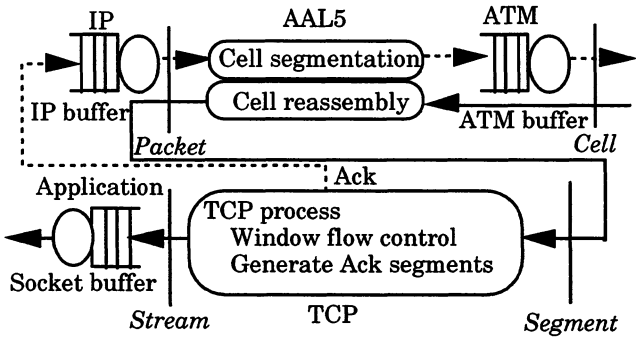


Figure 3 Model of the DES

- The DES sends an Ack segment for each data segment.
- The service time for generating an Ack segment is exponentially distributed and its mean is 1 ms.

2.5 Network Model

The network configuration in this simulation is shown in Figure 4.

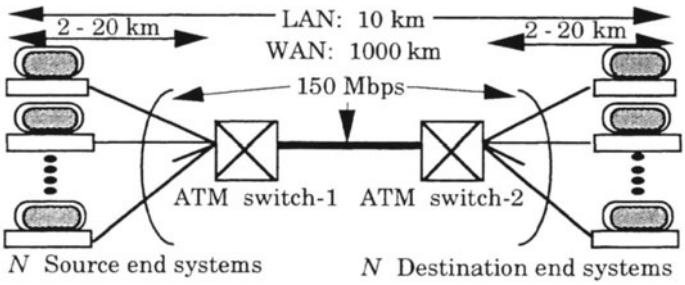


Figure 4 *N* VC, 2-node model

We set the parameters as follows:

- Maximum segment size: 9140 bytes (Atkinson 1989)
- Maximum window size: 64 kbytes

- Timer granularity: 0.1,100 ms*
- Distance between SWs and end systems: 2 - 20 km
- Distance between end-systems: 10 km (LAN), 1000 km (WAN)
- Link rate: 150 Mbps
- Buffer size of SW1†: 512 - 8192 cells
- Number of virtual connections: 5, 10, 25

In this configuration, the ATM SW1 is a bottleneck.
We used throughput as a measure of performance. The following throughput is desirable:
When N Virtual Connections (VCs) are active, the throughput of each VC should be $150/N*48/53$ Mbps. In other words, our target is good throughput and fairness.

3. SIMULATION RESULTS

We define the following notations:
- Link rate: C bps
- Window size: W bit
- Propagation delays in the round trip call path: $FRTT$ s
- Round-trip-time: RTT s
- Buffer size of the SW1: K cells
- Number of VCs: N

3.1 TCP over UBR

In order to achieve our target, it is necessary both to utilize the full link capacity and to get rid of cell loss. Since the number of TCP segments in the network is limited by the TCP window size, we need to set an optimal window size that achieves good performance.

$$C' \times RTT < N' \times W' \quad (1)$$

$$N' \times W' > K' \times 48' \times 8 + FRTT \times C' \quad (2)$$

$$W' = W' \times 9188/9140$$

$$C' = C' \times 48/53$$

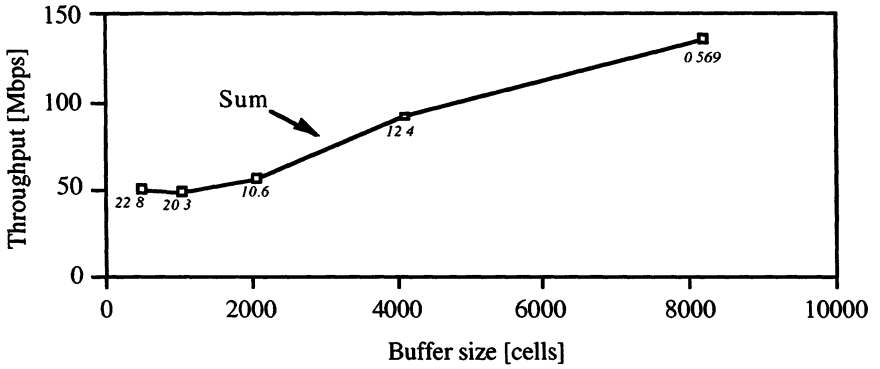
---

* In ordinary implementation of TCP, timer granularity is over 100 ms. Nevertheless, timer granularity significantly affects the performance, so we use finer timer granularity.

† We sometimes call buffer size of SW1 as buffer size, because the SW1 is the only switch of some concern.

When (1) is not satisfied, the link cannot be fully utilized. When (2) is not satisfied, cells will be lost at the bottleneck switch, which will cause retransmission of data and performance degradation. Therefore (1) and (2) are necessary for good performance, but TCP window size cannot be set by taking into account such conditions.

A typical example of performance degradation is shown in Figure 5 ("Sum" means the sum of the throughput in each VCs and italicised digits in the figure show "Fairness", which means the maximum difference in throughput between VCs).

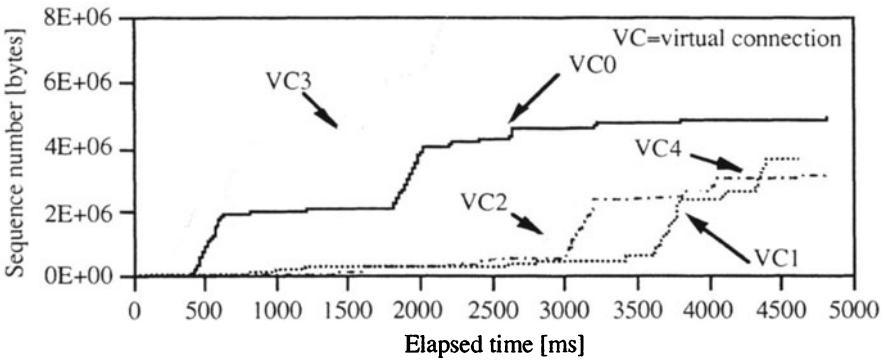


Number of VCs=5, Timer granularity=100 ms, LAN environment

Figure 5 Performance of TCP over UBR

When buffer size is large enough, that is, when (2) is satisfied, no cell loss occurs ((1) is always satisfied in this example). In this configuration, we need about a 7000-cell buffer. However the performance is poor when the buffer size is small. In Figure 6, the snapshot of the sequence number‡ of each connection is plotted to show the performance in detail.

‡ Sequence number means the number of data segments that are correctly received by a destination host.



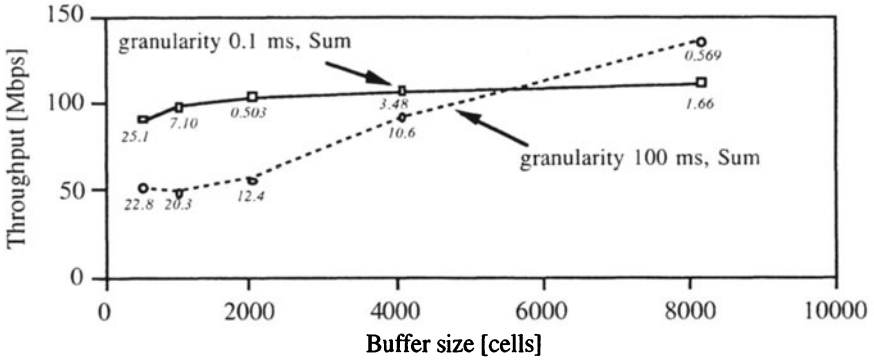
Buffer size=1024 cells, Timer granularity=100 ms, LAN environment

Figure 6 Snapshot of sequence number

Once a segment is lost, a DES does not receive additional segments until the lost segment has been retransmitted and received. Since time-out value is doubled for each retransmission, it takes longer to detect data loss when many retransmissions take place, which means slow recovery from the data loss. This is reflected in the slow increase in the sequence number in Figure 6.

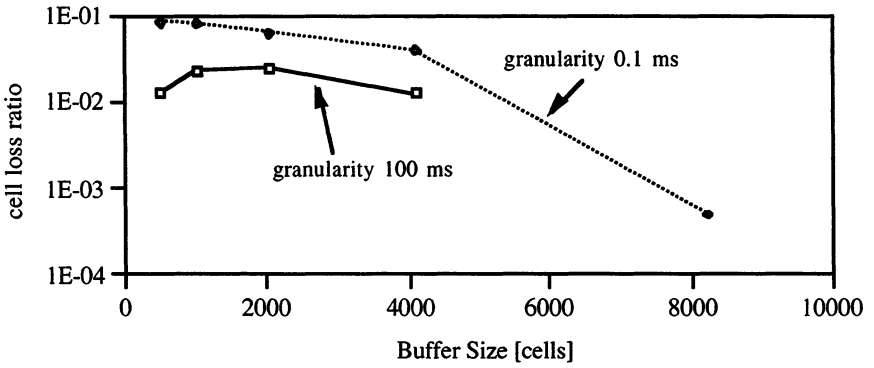
Some or the VCs have long idle periods, which means it takes a longer to start retransmission after cell loss occurs. This is because RTO is too long compared with RTT because a coarse grained timer is used. In this case, timer granularity is 100 ms and the minimum RTO is set to 200 ms, which is too long for a high-speed network. Furthermore, while one VC is idle because of loss, other VCs can continue transmitting data; this causes unfairness between connections.

One solution to this problem is to use finer timer granularity, which means faster detection of data loss.



Number of VCs=5, LAN environment

Figure 7 Effect of finer granularity

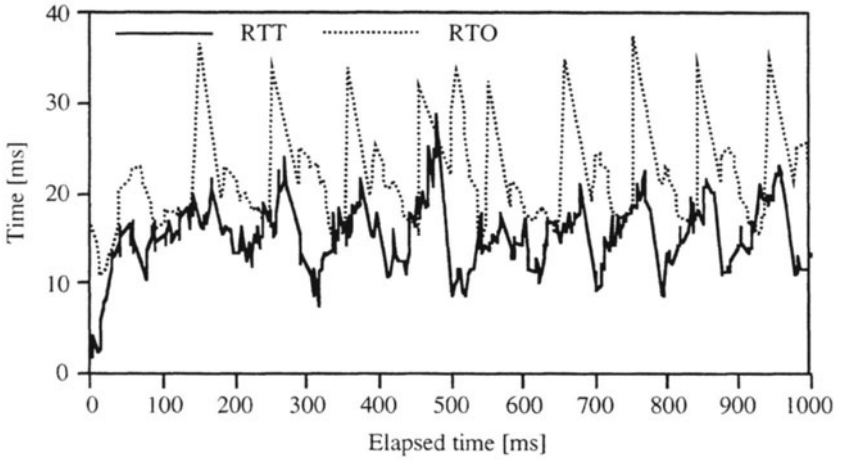


Number of VCs=5, LAN environment

Figure 8 Cell loss ratio for TCP over UBR

With a finer timer granularity, say 0.1 ms, even when the buffer is small and much cell loss occurs, good throughput and fairness are observed (Figure 7). However, the load of SW1 is higher when finer granularity is used, because it takes little time to start retransmission after cell loss occurs. This leads to a high cell loss ratio (Figure 8).

Throughput is not improved with buffers over 2000 cells when granularity is 0.1 ms. This is because the RTT estimation algorithm cannot work well and the timer goes off prematurely when no data loss has occurred. An example of these problems is shown in Figure 9. RTO is ordinarily calculated based on both the Smoothed RTT (SRTT) and the smoothed mean deviation. When retransmissions take place, RTO is doubled by using the exponential back off (see Figure 9). In this example, many retransmissions occur without cell loss. This is because RTO cannot keep up with fluctuations in the RTT, causing spurious retransmissions. Particularly when the queue length grows rapidly, RTO cannot catch up with the increase in RTT.



Number of VCs=5, Buffer size=8192 cells, Timer granularity=0.1 ms, LAN environment

Figure 9 Snapshot of RTT, SRTT and RTO

This result indicates the timer has limitations in achieving high performance. The timer algorithm has two conflicting targets:

1. Detect failure quickly and
2. Minimize false time-outs.

Since the RTT is affected by various network conditions, it is very difficult for the timer algorithm to meet both targets.

3.2 Summary of TCP over UBR

1. The TCP parameters cannot be set to optimize performance when the UBR allows cell-loss. As a result, much cell loss occurs when many VCs are multiplexed. Cell loss significantly degrades performance.
2. To achieve good performance with cell loss, fast failure detection is important. One solution is finer timer granularity. This reduces the effect of cell loss but may be difficult to implement. Furthermore finer timer granularity leads to false time-outs in TCP. The TCP timer goes off prematurely because buffering in SW causes oscillation in RTT. The RTT estimation algorithm sometimes cannot catch up with the change in RTT. As described before, the timer algorithm has limitations. Therefore, another trigger to detect data loss or preventive control is needed for better performance.

3.3 TCP over ABR

The parameters for ABR rate control are shown in Table 1.

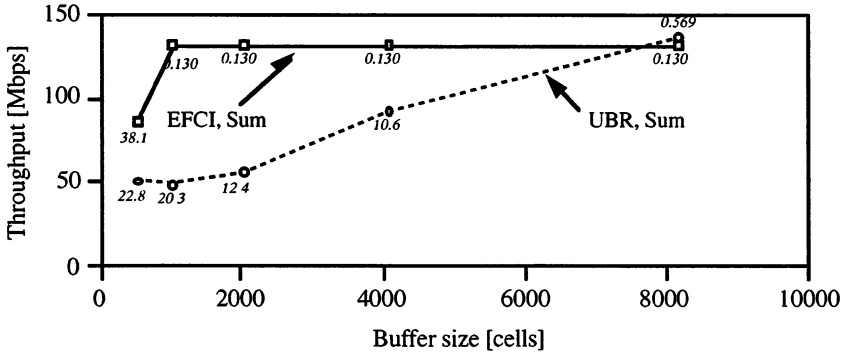
Table 1 Parameters for rate control

	EFCI		*ER*	
	LAN	*WAN*	*LAN*	*WAN*
RIF	1/64	1/512	1	1
RDF	1/16	1/128	1/16	1/16
Nrm [cells]	32	32	32	32
Threshold§ [cells]	256	256	256	256

In EFCI mode, only binary information about congestion is available. This results in poor performance when feedback delay is long or many VCs are multiplexed.
In ER mode, the explicit rate that switches can support is given to the SES. ER mode is thus more effective than EFCI mode.

Simulation with EFCI switches

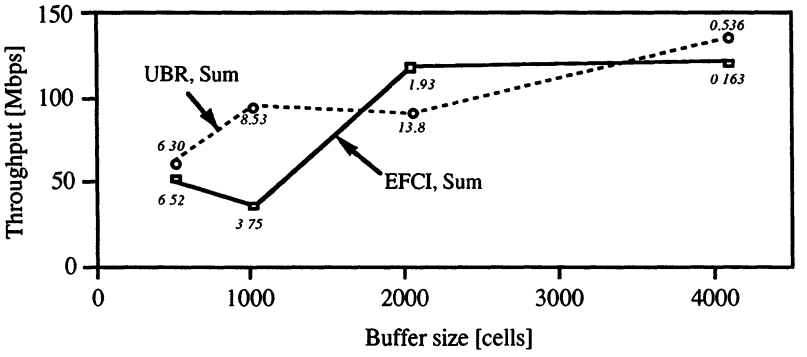
Performance of TCP over EFCI in a LAN environment is shown in Figure 10. In a LAN environment, in which feedback delay is short, cell loss does not occur when the buffer size is over 1024 cells. The difference in the performance between UBR and ABR decreases when the buffer size is large. This is because cell loss ratio decreases in UBR (Figure 10).
In a WAN environment, the performance is worse than in the LAN environment because of long feedback delay (Figure 11).
As a whole, the cell loss ratio in ABR is smaller than in UBR (Figure 12).



Number of VCs=5, Timer granularity=100 ms, LAN environment

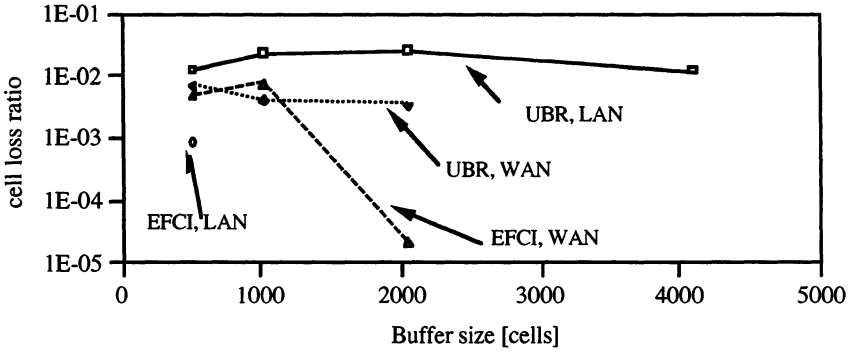
Figure 10 Performance of TCP over EFCI in a LAN environment

§ Threshold is used for judging congestion.



Number of VCs=5, Timer granularity=100 ms, WAN environment

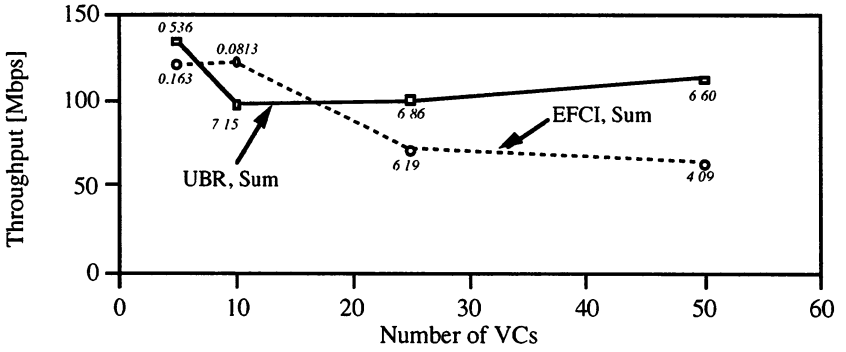
Figure 11 Performance of TCP over EFCI in a WAN environment



Number of VCs=5, Timer granurality=100 ms

Figure 12 Cell loss ratio for TCP over EFCI

When many VCs are multiplexed, cell loss occurs due to the heavy load caused by many TCP connections. The performance of EFCI is significantly worse than when there are 5 or 10 VCs because of the greater cell loss (Figure 13). However, we do not completely understand the mechanism that causes this. We will analyze this and clear this problem further and hope to clarify the cause.



Buffer size=4096 cells, Timer granularity=100 ms, WAN environment

Figure 13 Performance of TCP over EFCI

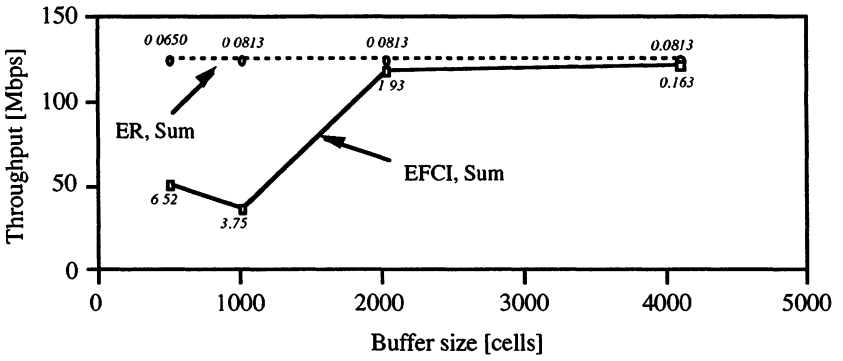
We recommend that PCR be set low in such cases, because cell loss is not observed when it is low. As is shown in Table 2, the Sum in EFCI mode is improved by setting a low PCR. Fairness is also improved.

Table 2 Effect of a low PCR (Number of VCs=50, Buffer size=4096 cells, Timer granularity=100 ms, WAN environment)

	Sum [Mbps]	*Fairness [Mbps]*
UBR: PCR=150 Mbps	113	6.60
UBR: PCR=25 Mbps	110	5.59
EFCI: PCR=150 Mbps	68.7	6.19
EFCI: PCR=25 Mbps	129	4.09

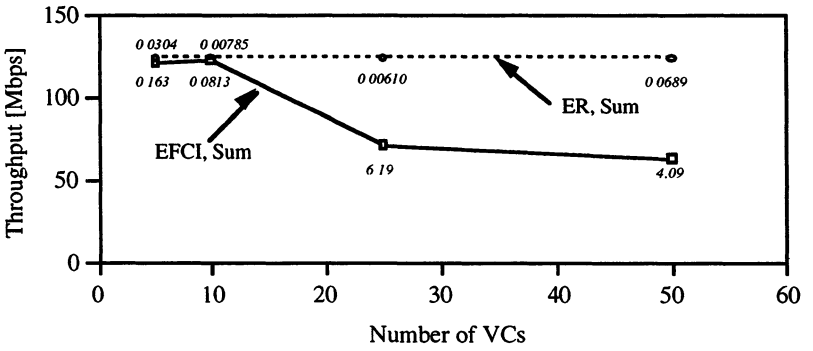
Simulation with ER switches

If users want to set a high PCR, ER switches are needed instead of EFCI switches, because ER switches indicate the rate that they can support. These are effective even when feedback delay is long or many VCs are multiplexed. As is shown in Figure 14, TCP over ER shows better performance than TCP over EFCI, particularly with a small buffer. TCP over ER also shows better performance with many connections (Figure 15).



Number of VCs=5, Timer granularity =100 ms, WAN environment

Figure 14 Performance of TCP over ER in a WAN environment



Buffer size=4096 cells, Timer granularity=100 ms, WAN environment

Figure 15 Performance of TCP over ER

When ABR service is used, finer timer granularity is not effective because cell loss does not occur but spurious retransmissions take place because of RTT oscillation. For example, "Sum" decreases from 124 Mbps to 104 Mbps when finer granularity (0.1 ms) is used (Number of VCs=25, Buffer size=4096 cells, WAN environment).

3.4 Summary of TCP over ABR

1. Switches with EFCI mode are effective when feedback delay is not too long and the number of multiplexed VCs is small.
2. With long feedback delay or many multiplexed VCs, their performance is poor unless we avoid such degradation by setting a low PCR.

3. Switches with ER mode are effective even when feedback delay is long and many VCs are multiplexed. If switches with ER mode are used, we can maintain a high PCR.
4. Finer timer granularity is not effective in TCP over ABR, because the cell-loss ratio is small enough and spurious retransmission degrades performance.

4. SUMMARY AND FURTHER WORK

In this paper, we reported the performance of TCP over UBR and ABR.
In the case of UBR, cell loss significantly degrades performance. This problem can be reduced with finer granularity of the timer in TCP. Spurious retransmissions tend to occur because of the RTT oscillation.
Using switches with the EFCI mode for rate control works well, particularly for small networks. To reduce cell loss in large networks, parameter tuning such as reducing the Peak Cell Rate is needed. Otherwise, we need another control method such as an ABR switch with the ER mode.
In this paper, we used only a simple TCP application model and network configuration to determine basic performance of TCP over ABR. Further studies are needed to investigate:

1. asymmetric network model in propagation delay and access link rates,
2. networks with multiple cascaded links in congestion,
3. networks in which both interactive and bulk data TCP segments exist, and
4. networks in which background traffic such as UDP traffic exit.

We also need to clarify the protocol requirements in each layer to achieve high performance in a high-speed network.

REFERENCES

Kleinrock L. (1992) The Latency/Bandwidth Tradeoff in Gigabit Networks. *IEEE Communication Magazine*, **30,** No. 4, 36-40.

Hasegawa H., Yamanaka N. and Shiomoto K. (1995) A Virtual ABR Transmission achieved by Local Switch on a Large-scale Network. *IFIP 1st Workshop on ATM Traffic Management*, 197-204.

Koike A. Kitazume H. Saito H. and Ishizuka M. (1996) On End System Behavior for Explicit Forward Congestion Indication of ABR Service and Its Performance. *IEICE Trans. Commun*, **E79-B**, No. 4, 605-610.

Romanow A. and Floyd S. (1994) Dynamics of TCP Traffic over ATM Networks. *Proc. ACM SIGCOMM Conference*, 79-88.

Lakshman T. V. Neidhardt A. and Ott T. J. (1996) The Drop from Front Strategy in TCP and in TCP over ATM. *Proc. IEEE INFOCOM'96*, 1242-1250.

Lin M. Hsieh J. Du D. J. C. Thomas J. P. and MacDonald J. A. (1995) Distributed Network Computing over Local ATM Networks. *IEEE Jounal on Selected Areas in Communicaion*, **13**, No 4, 733-748.

Barden R. (1989) Requirements for Internet Hosts-Communication Layers. RFC1122.

Atkinson R. (1989) IP MTU over ATM AAL5. RFC1626.

The ATM Forum (1996) ATM Forum Traffic Management Specification 4.0.

Barnhart A. W. (1995) Example Switch Algorithm for Section 5.4 of TM Spec. ATM_Forum/95-0195.

BIOGRAPHIES

Mika ISHIZUKA graduated from Keio University with B.E. and M.E. degrees in control engineering in 1992 and 1994, respectively. In 1994, she joined NTT. She has been engaged in research of traffic issues on computer networks. Ms. Ishizuka is a member of the IEICE and the Operations Research Society of Japan.

Hideo KITAZUME graduated from Gunma University with B.E and M.E. degrees in computer science in 1987 and 1989, respectively. In 1989, he joined NTT. He has been engaged in research and development of ATM LAN systems. He is currently working on service integration in ATM networks. Mr. Kitazume is a member of the IEICE and the Operations Research Society of Japan.

Arata KOIKE graduated from St. Paul's University with B.S. and M.S. degrees in theoretical physics in 1989 and 1991, respectively. In 1991, he joined NTT. He has been engaged in research of traffic issues on intelligent networks, mobile communications, and ATM networks. He is currently a research engineer. Mr. Koike is a member of the physical Society of Japan and IEICE.