Performance Comparison of UT and CT under Realistic Traffic Models*

J. R. Gallardo, D. Makrakis

Department of Electrical Engineering
Faculty of Engineering Science
The University of Western Ontario
London, Ontario,
N6A 5B9 CANADA

M. López, L. Orozco Barbosa

School of Information Technology and Engineering
University of Ottawa
161 Louis Pasteur
Ottawa, Ontario
K1N 6N5 CANADA

S. Rosenberg

Newbridge Networks Corporation
600 March Road
P.O. Box 13600
Kanata, Ontario
K2K 2E6 CANADA

ABSTRACT

In this paper we analyze the performance of a rate-based control mechanism recently proposed by Fulton *et al*, namely the Uniform Tracking algorithm and a credit-based algorithm called the Controlled Transfer algorithm. The performance study has been carried out via simulation and considering realistic traffic models. Our results show that UT is efficient in providing bursty, elastic data applications. As for the CT algorithm, it was observed that it is very sensitive to buffer size selection: too small buffers cause low link utilization at the switches, while too large buffers produce high queuing delays.

I. INTRODUCTION

The main objective of this paper is to present the results of a performance study comparing a rate-based ABR flow control mechanism recently proposed by Fulton *et al* [1], namely, the Uniform Tracking, or UT algorithm and the credit-based flow control algorithm called Controlled Transfer (CT), which in turn is based on the Quantum Flow Control (QFC) mechanism [2]. The main contribution of our study is the operation evaluation of the above mentioned control mechanisms, using realistic traffic models. The traffic models used in this study were developed based on the measurements and analysis reported in [3][4]. For the remaining, this paper is organized as follows. Section II briefly describes the flow control schemes under study. Section III introduces the two network topologies utilized in the evaluation study. Section IV describes the traffic models used. Section V defines the network performance measures considered. The results of our study are given in section VI. Finally, section VII concludes the paper.

II. FLOW CONTROL MECHANISMS

A. Uniform Tracking (UT) Algorithm

In [1], Fulton *et al* proposed a new rate-based control scheme, namely Uniform Tracking (UT). The operation of UT can be briefly described as follows. During each control interval, denoted by t_c, UT switches compute an effective Fair Share, FS_{eff}. The Fair Share is written into the ER field of a backward RM cell if it is smaller than the presently stored value. The FS_{eff} is computed by:

$$FS_{eff} = \frac{\rho_T C - \bar{r}_B(n)}{N_{eff}} \quad (1)$$

where: ρ_T, C and $\bar{r}_B(n)$ are the target link utilization, the link capacity and the lowpass-filtered background traffic rate at time interval n, respectively. ρ_T is a configurable parameter with values between 0 and 1. The effective number of sources N_{eff} is tracked by:

$$N_{eff} = \max\left\{\frac{r_A(n)}{\tilde{FS}_{eff}},\ 1\right\} \quad (2)$$

where: $r_A(n)$ is the total ABR rate at time interval n and $\tilde{FS}_{eff}$ is a moving average of previously assigned FS_{eff}.

The control interval t_c is an important parameter for UT, in the sense that it must be chosen sufficiently large to obtain accurate rate estimates (a single cell should not be interpreted as a high increase in the rate of transmission) and to allow practical implementation; however, selection of too long an interval may lead to sluggish results. The authors in [1] have proposed to select the control interval from:

$$t_c \approx \max\left\{\frac{1}{0.005\rho_T C},\ 10\,ms\right\} \quad (3)$$

where the link capacity C has units of cells/ms.

B. Controlled Transfer (CT) Algorithm

In [2], a protocol is described that is characterized by the exchange of buffer state information. The basic idea is that a transmitter will send a cell whenever it has at least one cell ready and it is informed via feedback from the receiver that there is space to receive and store that cell. The receiver, in turn, will forward cells onto the next logical link at every available opportunity thus freeing buffer space. As cells are forwarded, the receiver informs the transmitter of the buffer space that has been made available. The feedback message is referred to as a buffer state update (BSU). A receiver uses the following state variables:

- Rx_Counter[i], which indicates how many cells have been received.

* This work was partially funded by a Newbridge Research Contract. M. López was supported in part by a scholarship of the Fundación UNAM (Mexico).

- Fwd_Counter[i], which counts the received cells that have already been forwarded to the next element in the path. The current value of this variable is included in BSU messages.
- Policy_Limit[i], which specifies the maximum number of received cells that can be stored in the buffer.
- N2_Counter[i], which contains the number of cells that have been forwarded since the time when the last update message was sent back to the transmitter.

A transmitter uses the following state variables:

- Tx_Counter[i], which indicates how many cells have been transmitted.
- BSU_Fwd_Count[i], which stores the Fwd_Counter value received in the last update message from the receiver.
- Policy_Limit[i], which indicates the size in cells of the buffer allocated at the receiver for this connection.

In the previous state variables, the index i indicates that the stored value corresponds to the i-th connection (VP or VC). A transmitter may continue sending cells as long as the difference between Tx_Counter[i] and BSU_Fwd_Count[i] remains less or equal to Policy_Limit[i].

Even though our work only focuses on flow control per VP, our results can be equally applied to the Extended Control option (VP and VC) as proposed in [2].

III. NETWORK CONFIGURATIONS

A. Basic 2-Node Configuration

This configuration, depicted in figure 1, is intended to be a test for the control schemes in the presence of differing delays and differing number of sources. This simple configuration has been used in many studies [1] to evaluate the way control algorithms distribute the link capacity among the active sources. This network configuration has proved to be a useful framework to compare the performance of various schemes. However, in our case we go beyond by using more realistic traffic models. This configuration was studied under the scenarios depicted in table I. In the figure D_1, D_2 and D_3 denote the propagation delays. They have been set as follows: $2D_1 = 2\ D_3 = D_2$.

Table I. Parameters - 2-Node configuration

Case Number	Number of Sources	Link Rates	RTT (ms)
1	10	DS3	1
2	100	OC3	1
3	10	DS3	25
4	100	OC3	25
5	10	DS3	50
6	100	OC3	50

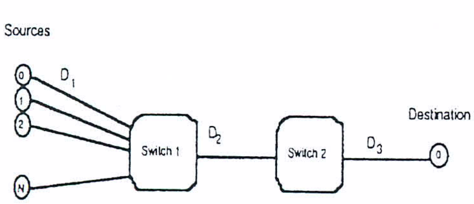

Figure 1. Basic 2-Node Configuration

B. Max-Min Fairness Configuration

This configuration, see figure 2, is intended to show the performance of the algorithms in a more realistic environment including short and long paths. This arrangement is often called the Max-Min Fairness Configuration and it is used to evaluate the way traffic control mechanisms allocate the capacities to sources exposed to different propagation delays and how long haul connections are affected by cross traffic.

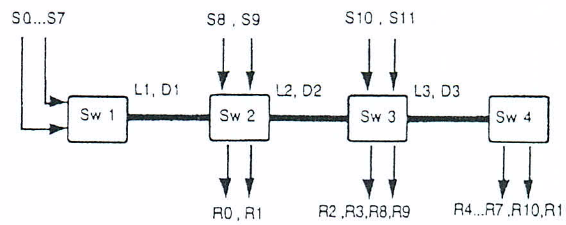

Figure 2. Max-Min Fairness Configuration

In figure 2, S_0 through S_{11} represent the sources and R_0 through R_{11} represent the corresponding receivers. The link rates L_1, L_2, and L_3, between switches, were assigned the values shown in table II for the different simulations. This was done with the purpose of varying the position of the network bottleneck. All delays between the sources and the closest switch and between the switches and the destinations were assumed to be 100 µs. The propagation delays, D1, D2 and D3, were all set at 5 ms.

Table II. Parameters - Max-Min Configuration

Case Number	Link Speeds		
	L1	L2	L3
1	DS3	DS3	DS3
2	DS3	DS3	25 Mbps
3	25 Mbps	DS3	DS3

IV. TRAFFIC MODELS

In the sequel, the reader will encounter quite frequently the term *source*. By source we refer to an individual user running an application that is generating and delivering traffic to the network.

We based the selection of our models on the findings reported in [3][4], because their results are extracted from actual measurements performed at several sites. Despite the fact that these results were collected in dissimilar environments and under different circumstances, the models obtained (probabilistic distributions and parameter values) are very similar. This suggests the great generality of these results. The above mentioned references involve two main classes of applications: *a)* interactive (telnet, rlogin), and *b)* bulk-data transfer (ftp, smtp [mail transfer], nntp [news transfer]). However, our simulations will only involve bulk-data transfer applications, since interactive applications produce less intensive traffic, and therefore are less significant in congestion situations. Bulk-data traffic produces the dominant amount of traffic, in terms of bytes.

Bulk-data traffic sources are modeled as follows:

- Arrivals of new sessions (a new user starting an application, essentially) are modeled using a Poisson random variable with mean arrival rate $\lambda = 0.014$. This model is taken from [4].
- The number of data bursts transmitted per session is selected randomly using empirical data from TCPlib [3].
- Burst inter arrival time within a session, from the end of a burst to the beginning of the next one, is modeled as a log-normal random variable with log-mean = $ln(4)$ = 1.386, and log-sigma = 1.8. This distribution and values are based on the results reported in [4].
- The size of a data burst is modeled as a Pareto random variable, with $\alpha = 2\times10^6$ and $\beta = 1.2$, as proposed in [4].

The above-mentioned models are used to mimic the traffic generated at the application level. In our simulation model, we have included a protocol stack consisting of the application level, TCP, an ATM adapter implementing the SAR functions and the transmission line or physical layer at the bottom.

A simulator model of TCP has been implemented. We have paid particular attention to model the TCP congestion control mechanisms. The congestion control mechanisms implemented in our simulation are based on the scheme proposed by Van Jacobson.

The maximum TCP window size has been optimized to take into account the link rate and the propagation delays of the links. Accordingly, the window size has been set to the bandwidth-delay product, that is:

$$window\_size = RTT \times link\_rate \tag{4}$$

where the window size is expressed in bytes, RTT denotes the round-trip time in seconds and the link rate is expressed in bits per second.

V. PERFORMANCE METRICS

The measures to be taken into account for evaluating and comparing the algorithms' performance are:

- Buffer occupancy at the switches.
- Cell transfer delays (CTD).
- Link utilization at the switches.

For the performance of the whole system, the following measures are considered only for the 2-node configuration:

- Fairness.
- Goodput.

The cell transfer delay is defined as the time elapsed from the moment the cell is at the front of the queue of the source adapter, ready to be transmitted, until it is received by the destination adapter.

In the case of a rate-based control mechanism, the buffer occupancy together with the link utilization are essential metrics to evaluate. If at any point in time, the buffer occupancy is very high, this will mean that the control mechanism is accepting more cells into the network than it should. This will obviously translate into a high link utilization. However, if the link utilization is low even in the case that the sources are active and the occupancy of the buffer related to that link is low, this will mean that the control mechanism is overreacting.

A. Fairness Index

One of the objectives of a congestion control is to provide fairness to all users of a network. Fairness ensures that no circuits are arbitrarily discriminated against and no set of circuits is arbitrarily favored. A fairness index is defined in ATM Forum Traffic Management Specification Version 4.0 to evaluate the share of the available bandwidth among the users:

$$Fairness\_Index = \frac{\left[\sum_i x_i\right]^2}{n \cdot \sum_i x_i^2} \tag{5}$$

where n is the number of connections (or sources) sharing the network resources, and x_i is the ratio of the actual throughput of a connection to the optimal throughput. The optimal throughput is the fair share of the available bandwidth for the considered connection.

B. Goodput

Goodput is defined as the ratio of the achieved throughput to the maximum achievable throughput. Throughput is defined as the rate of good data received by the TCP receiver. Retransmissions triggered by the TCP stack or duplicate packets received at the receiver are not counted as good data.

The maximum achievable throughput is limited by the bottleneck in the network or at the source. Usually, goodput is expressed as a percentage of the bottleneck link and reflects the efficiency in using the link. The goodput is then given by:

$$Goodput = \frac{\sum_i (Good\_data)_i}{N \cdot T \cdot Line\_Rate \cdot \dfrac{Pkt\_Size}{53 \cdot \lceil Pkt\_Size/48 \rceil}} \tag{6}$$

where N is the number of bottleneck links, $Good\_Data$ is the total amount in bits of data corresponding to successfully transmitted packets, T is the measurement period (simulation time in this case), Pkt_Size is the size of the TCP packet in bytes, and $Line\_Rate$ is the maximum transmission rate of the bottleneck link between the two switches. $\lceil x \rceil$ is the smallest integer greater than or equal to x.

VI. SIMULATION RESULTS

For all configurations the TCP timer granularity was set to 0.5 seconds, the PDU size was fixed to 1536 bytes and the maximum buffer size at the ATM adapter s was set to 1,130 cells. For UT, the Fixed Round-Trip Time (FRTT) was fixed according to the total propagation delay for each case. The set of parameters used for the UT control mechanism is given in table IV. The value of N2 for the CT mechanism has been set to 32 cells and the time between BSC messages was selected as 1 second for all our CT simulations.

Table III. Queue Lengths for Switch 1

Case Number	Mean (cells)	Maximum (cells)
1	1.613	156
2	3.169	24
3	1.618	259
4	3.528	31
5	3.022	750
6	3.485	82

A. Results - 2-Node Configuration

We start by evaluating the mean and maximum queue lengths for UT via simulation. These results showed that UT requires very limited buffer sizes to operate. The target channel utilization of 90% was achieved with a mean queue length of less than 5 cells and a maximum queue length of 750 cells (see table III).

In a first set of simulations for CT, the maximum queue lengths given in table III were used to dimension the buffer sizes. The use of these small buffer sizes resulted in a channel utilization as low as 1% for cases with a large propagation delay. In order to be able to carry a fair comparative study, CT was tested in this configuration under two other different sets of buffer sizes.

The first case uses buffer sizes selected using the bandwidth-delay product (BDP) criterion, while the second one uses buffer sizes optimized to avoid packet retransmissions due to excessive queuing delays. It is important to mention at this point that sizing the buffers using the BDP criterion alone can cause CT to incur excessive queuing delays at the switches and even retransmissions due to TCP timeouts.

Table IV. UT Parameters

Symbol	Definition	Value
t_c	Control Interval	10 ms
MCR	Minimum Cell Rate	2×10^4 bps
ICR	Initial Cell Rate	$1/t_c$ cells/s
PCR	Peak Cell Rate	Link Rate*
ρT	Target Link Utilization	0.90**
RDF	Rate Decrease Factor	1/32768 *
RIF	Rate Increase Factor	1*
Nrm	Number of data cells between forward RM cells	32*
Mrm	It controls bandwidth allocation for RM and data cells	2*
TBE	Transient Buffer Exposure	16e6*
CDF	Cutoff Decrease Factor	1/16*
Trm	Upper bound on the time between forward RM cells	0.1s*
ADTF	ACR Decrease Time Factor	0.5s*

* Default for TM 4.0. ** Configurable.

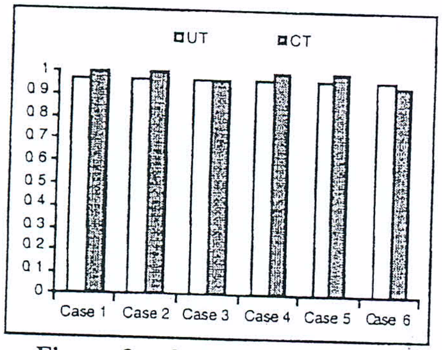

Figure 3a. Goodput comparison.

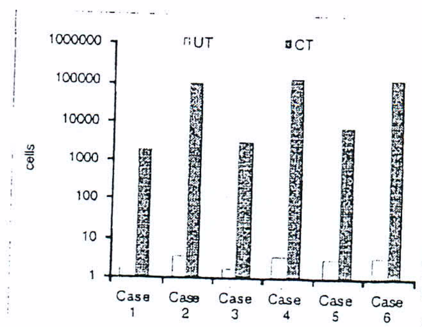

Figure 4. Mean queue length comparison (logarithmic scale)

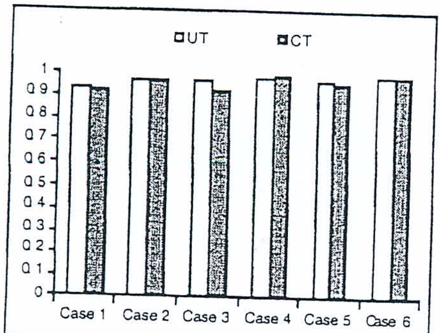

Figure 3b. Fairness comparison.

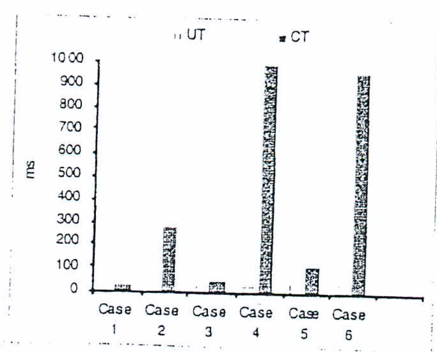

Figure 5. Average CTD comparison

• BDP buffer sizes

The buffer sizes at the switches for this set of simulations were selected using the *bandwidth-delay* product criterion. The *bandwidth* corresponds to the transmission rate of the link joining the source to the closest switch. The *delay* used is from the transmission of the first cell to the reception of the first BSU message (sent by the receiver after forwarding N2 messages to the next element in the path).

Figures 3a and 3b compare the performance of both approaches in terms of goodput and fairness, and show that the difference is not very significant. It was observed that CT uses much more memory (up to over 30,000 times more, in cases 2, 4 and 6) than UT; figure 4 depicts graphically this big difference.

Regarding average cell transfer delay (CTD) through the network, Figure 5 depicts the difference. Notice that while the average CTD for UT is very close to the propagation delay alone, that is not the case for CT, in which queuing delays are very significant.

Because of these high CTD values, a total of 82,817 packet retransmissions due to TCP timeouts were observed in case 6; as an example, a single source had to retransmit 1576 packets during the simulation time (150 seconds).

• Buffer sizes to avoid TCP timeouts

For this set of simulations, the buffer sizes were reduced for cases 2, 4 and 6 so as to avoid retransmissions due to excessive queuing delays at the switches. The criterion used was to constrain the time from the transmission of a packet to the reception of its ACK to be less than 0.5 seconds.

Figure 6 depicts a graphical comparison of the mean cell transfer delays (CTD). It can be observed that the values obtained in all cases are much larger for the CDT scheme than for UT. The results for the other metrics were almost the same than for the bandwidth-delay case.

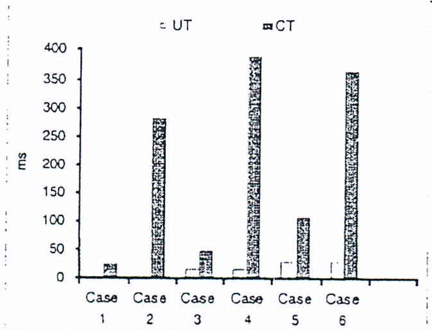

Figure 6. Average CTD comparison

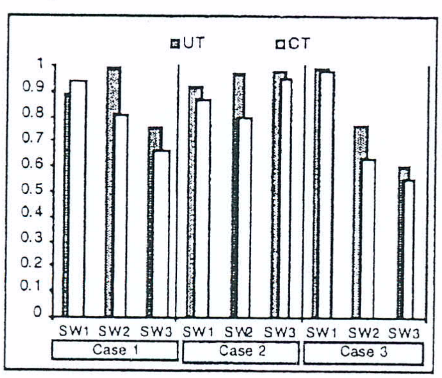

Figure 7. Link utilization comparison

B. Results - Max-Min Configuration

For this configuration we only consider the case when the buffer sizes are set according to the bandwidth-delay product. Figure 7 shows graphically the link utilization for the two schemes. It can be seen that UT's performance is better in all cases and for all switches, except in case one, switch 1. Figure 8 depicts the mean CTD for both approaches. It is apparent the great difference that exists between them.

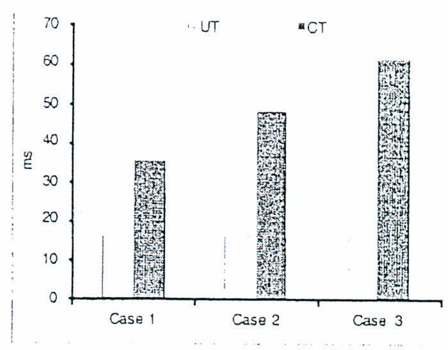

Figure 8. Average CTD comparison

For the Max-Min Fairness configuration, UT still outperforms CT as can be observed from the link utilization and cell transfer delay values, figures 7 and 8, respectively.

VII. CONCLUSIONS

In this paper we have analyzed the performance of a rate-based control mechanism recently proposed by Fulton *et al*, namely the Uniform Tracking algorithm [1] and a credit-based algorithm called the Controlled Transfer algorithm. The performance study has been carried out via simulation and considering realistic traffic models.

Our results show that UT is efficient in providing bursty, elastic data applications with high link utilization, minimum required buffering at the switches, and minimum transfer delay through the network. It was also confirmed that UT scales with increasing delays and increasing number of connections.

As for the CT algorithm, it was observed that it is very sensitive to buffer size selection: too small buffers cause low link utilization at the switches, while too large buffers produce high queuing delays. In both cases, many packet retransmissions due to timeouts were observed.

REFERENCES

[1] C. Fulton, S. Li, C. S. Lim, *"UT: ABR Feedback Control with Tracking"*, ATM Forum 97-0239 - Traffic Management, 1997.

[2] QFC Alliance. *"Quantum Flow Control Specification"*, July 1995. (http://www.qfc.org).

[3] P. B. Danzig, S. Jamin, R. Caceres, D. Mitzel and D. Estrin, *"An Empirical Workload Model for Driving Wide Area TCP/IP Network Simulations"*, Internetworking: Research and Experience, Vol. 3, No. 1, March 1992, pp. 1-26.

[4] V. Praxson and S. Floyd, *"Wide Area Traffic: The Failure of Poisson Modeling"*, IEEE/ACM Transactions on Networking, Vol. 3, No. 3, June 1995, pp. 226-245.