Performance Evaluation of Adaptive Rate Control (ARC) for Burst Traffics over ATM network

Surasee Prahmkaew [†] *and Chanintorn Jittawiriyanukoon* [†]

[†] Department of Telecommunications Science
Faculty of Science & Technology
Assumption University, Bangkok 10240, Thailand

Summary -- *The paper presents an Adaptive Rate Control (ARC) implemented to improve the performance of high-speed network to handle burst traffic by guaranteeing the cell loss ratio (CLR) for all cell streams. First, the cases in which a Tahoe, Reno, New Reno, SACK and Plain schemes are applicable in peak-cell-rate (PCR) are discussed. The ARC improves the performance by regulating the increment (up) and the decrease (down) of window size (flow control). Incoming traffic rate, number of cell drop, preset size of the window and estimated delay time are taken into account for this regulation. Simulations are used to investigate how Tahoe, Reno, New Reno, SACK and Plain can conduct, as congestion existed. Then we compare these results from four schemes to the "Plain" scheme (no flow control applicable) and to the proposed ARC. By altering windows size for the mentioned six schemes, we can obtain the supportive results.*

Key words ARC, Tahoe, Reno, New Reno, SACK, Sliding windows

1. Introduction

Only the high-speed network can service a traffic. The interface to Asynchronous Transfer Mode (ATM) or the architecture of ATM network would provide a multilevel of services. In networks as such, the burst information will be segmented into cells and the tremendous number of cells is traversed from sender to the destination via multiple hops transmission in the network. Not all traffic control methods can be applicable to the high-speed networks such as ATM [1][12].

There are many previous studies involving flow control algorithms [3] and a source descriptor [2], however the behavior of each flow control scheme [12] with regulating window sizes is not found. In this paper, we proposed an ARC flow control that improves the performance of high-speed network such as ATM network by altering an appropriate size of the flow control window. Our proposed ARC against four existing flow control schemes that are Tahoe, Reno, New Reno, and SACK [4],[15] plus one "Plain" scheme are discussed. Finally the performance evaluation, especially in term of throughput, number of cells loss, mean time in queue, mean queue length and utilization of ATM link, between these six schemes will be compared.

2. The Model of Four Schemes

The principle of ATM traffic flow control is that at connection setup, the user specifies both QoS requirements and using the anticipated traffic characteristic of the connection. Network resources for the connection are assigned on the basis of the source traffic descriptor values and the QoS requirements. If there are not enough network resources, the connection is cancelled. If the connection is accepted, actual amount of the traffic is examined to specify a connection set-up. If the amount of traffic is too large then the connection set-up for the whole is not accepted. But a portion that fits the connection set-up will be accepted then a penalty is imposed on the connection, e.g. some cells from the connection may be discarded. To simplify traffic flow control specification based on best QoS requirements and monitoring by the network, the traffic descriptors are required to be observable and easily adjustable through some mechanisms. The existing congestion avoidance algorithms are discussed. Tahoe algorithm includes Slow Start, Congestion Avoidance, and Fast Retransmit. The Reno is the enhancement to Tahoe by softening the Fast Retransmit process with inclusive Fast Recovery. Selective Acknowledgments (SACK) has been presented to recover multiple segment losses by transmitting a duplicate acknowledgement. The information contains the out-of-sequence bytes SACK, RFC 2018, [5] has received. SACK also allows the transmitter to reconstruct the information about the non-received bytes at the destination. Farther detail can be found in [6][7]. Partial ACK takes Reno out of frame, deflates window size. Sender may have to wait for timeout before proceeding. In New Reno, partial ACK indicates lost packets and retransmits immediately. Retransmits 1 lost packet per round trip time until all lost packets from that window are retransmitted. New Reno also eliminates timeout (RFC 2583) [6].

Consider the Tahoe, Reno and New Reno, and SACK when the burst traffic occurred either in short or long time duration. They started window with an advertised size ranging from 1 to maximum size. It is increased by 1 (slow–start technique) for every successful transmission. When the window is topped up to the maximum size, most of input traffic would be discarded or tagged with the reason of capacity exceeding. When this situation occurred, most of window size will start with congestion window (cwnd) size (recovery technique). Cwnd always set to 1 for Tahoe, half of maximum windows size for Reno, New Reno and up to multiple losses for SACK.

3. Adaptive Rate Control Scheme

In the proposed ARC, we alleviate the number of cells by shrinking or expanding the window size automatically based upon source rate, cells drop, and cells delay.

With ARC algorithm [16], it works like a control gate for all arriving cells. When cells arrive at gate and if no cell drop presents, the cell will be transmitted immediately (at no delay). If cell drop is present, the dropped cell will be firstly blocked in a cell queue (Q_c) and waiting for a chance of retransmission as ARC finishes regulating the new windows size in order to conquer the cells drop. At the same time to maintain quality of service (QoS), the maximum cell delay time has been defined as CDVT. It means the cells have been waiting in the cell queue longer than CDVT will be discarded finally. Figure 1 illustrates the ARC flow control model for our analysis regarding the cell arriving process, conforming and non-conforming cells, window size adjustment and two states of problem for ARC.

In case that the arrival traffic (average arrival cell rate or traffic (λ_a) is less than the cell drop rate (λ_p) and cell drop is not yet present. ARC will initially set window size to be one (the minimum size).

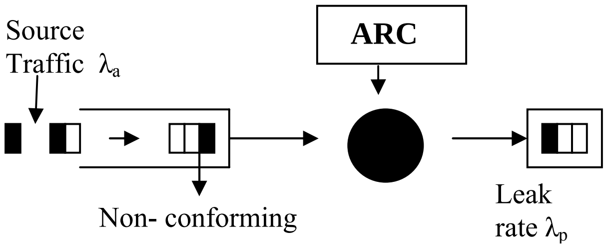


Figure 1. Flow control

On the other hand, if (λ_a) is larger than (λ_p) , ARC will regulate the window size with reference to cell drop rate (λ_p) and arrival traffic rate (λ_a). ARC will regulate the window size between one and three (three is the maximum size based on analytical model shown in figure 3). ARC algorithm is shown below.

```
/*********** ARC Algorithm ************/
PROCEDURE
/****** Window Size Calculation ******/
 Current allocation rate (Ai) ;
 Current win_size (Wi) ;
 Bandwidth (BW) ;
DO WHILE Transmission is Ongoing {
    IF Cell-Drop $\lambda_p \geq$ Ai THEN {
    Calculate new allocation rate (An) ;
    Calculate new win_size (Wn) ;
      Ai <= An ;
      Wi <= Wn ;  }
 ELSE {
      Ai <= ABW  ;
      Wi <= Wn ;  }
  }  END_DO  ;
/*** Calculate new allocation rate (An) ***/
An <= BW * cell_size/( $\lambda_p$)$^{1/2}$ ;
/*** Calculate current win_ size (Wn) ***/
IF  Cell-Drop exists  THEN
{
 Wn <= Wn ++ ;
IF Wn  >  Win_max THEN Wn <= Win_max;
}
ELSE {
Wn <= Wn – –;
 IF Wn  $\leq$  0  THEN Wn = 1  ; }
/*** Calculate available BW (ABW) ***/
ABW <= Max_BW – Used_BW;
/******* END OF ARC Algorithm *******/
```

Figure 2. ARC Algorithm

4. The Model of Plain Scheme

Unlike the proposed ARC, we are neither alleviating the number of cells by shrinking or expanding the window size automatically based upon source rate, cells drop, and cells delay nor applying any four schemes (Tahoe, Reno, New Reno and SACK) in the "Plain" scheme. We want to use this scheme for comparing the performance as well as to study what if all four (control) schemes and ARC are transparent to the system. This will give the idea how much these flow control schemes will help ease the congestion.

5. Simulation

Figure 3 demonstrates a simulation model utilized in the paper.

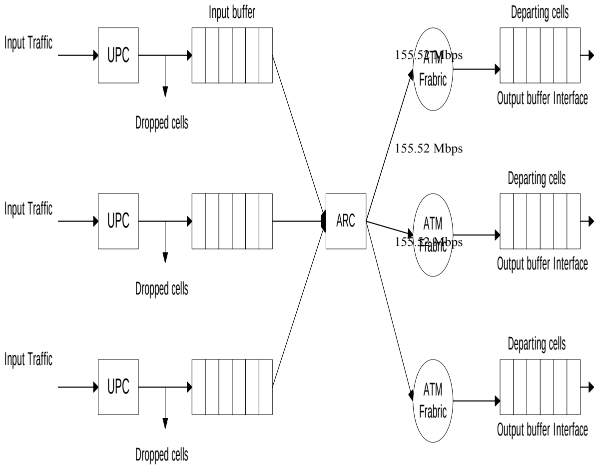


Figure 3. Simulation model

5.1 Input Traffic

The traffic can be basically classified into five categories: data, voice, video, image and graphics [14]. This research confines the discussion to mainly data, voice and video. Data sources are generally bursty in nature whereas voice and video sources can be continuous or bursty, depending on the compression and coding techniques used. Continuous sources are said to generate constant bit rate (CBR) traffic and bursty sources are said to generate variable bit rate (VBR)[9] traffic. Hence, only VBR traffic will be considered as an input for the study.

5.2 Characteristics of a Queuing Network Model

There are three components with certain characteristics that must be examined before the simulation models are developed.

5.2.1 Arrival Characteristics

The pattern of arrivals input traffic mostly is characterized to be *Poisson arrival processes* [11]. Like many random events, Poisson arrivals occur such that for each increment of time (*T*), no matter how large or small, the probability of arrival is independent of any previous history. These events may be individual cells, a burst of cells, cell or packet service completions, or other arbitrary events.

The probability of the inter-arrival time between event t, is defined by the *inter-arrival time probability density function (pdf).* The following formula gives the resulting probability density function (pdf), which the inter-arrival time t is larger than some value x when the average arrival rate is λ events per second:

$$fx(t) = \begin{cases} e^{-\lambda t}, \ for & t \geq 0 \\ 0, \quad for & t < 0 \end{cases}$$

$$p(t \leq x) = Fx(x) = \int_0^x e^{-\lambda x} dx = 1 - \lambda e^{-\lambda x}$$

$$p(t > x) = 1 - Fx(x) = \lambda e^{-\lambda x}$$

Queuing theorists call Poisson arrivals a *memoryless process,* because the probability that the inter-arrival time will be *X* seconds is independent of the memory of how much time has already expired. The formula of memoryless process is shown accordingly:

$$P(x > s + t \mid X > t) = P(X > s) = e^{-\lambda s}, \ for \qquad s_i t > 0$$

This fact greatly simplifies the analysis of random processes since no past history, or memory, affects the processes commonly known as *Markov processes*. The probability that n independent arrivals occurs in T seconds is given by the formula *Poisson distribution*:

$$P(n, T) = (\lambda T)^n (e^{-\lambda T}) / n!$$

where

$P(X)$ = probability of X arrivals,
n = number of arrival per unit of time,
λ = average arrival rate,
$E\{n|T\}$ = λT = expected value of n for a given interval T, and e = 2.7183

The combination of these two thoughts in a commonly used model is called the Markov modulated Poisson process (MMPP) or ON/OFF bursty model. In this paper, the burstiness is varied by altering the T_{ON} and T_{OFF}.

5.2.2 Service Facility Characteristics

In this paper, service times are randomly distributed by the *exponential probability distribution*. This is a mathematically convenient assumption if arrival rates are Poisson distributed. In order to examine the traffic congestion at output of ATM link (155.52 Mbps), the service time in the simulation model is specified by the speed of output link, giving that a service time is 2.726 µs per cell.

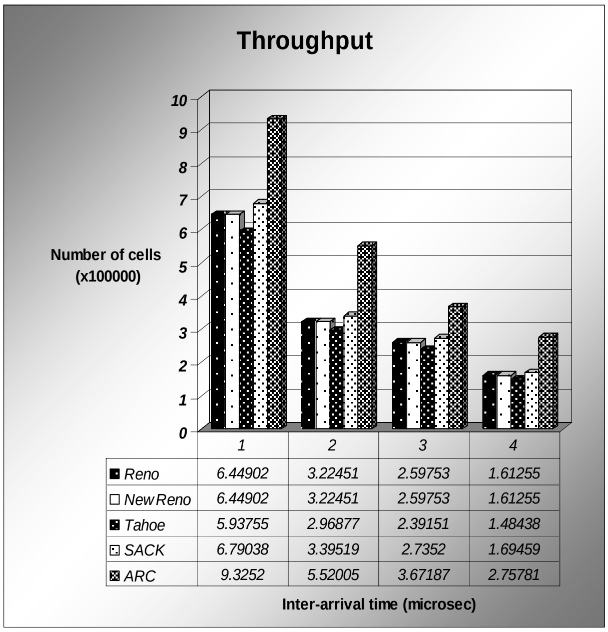
5.2.3 Source Traffic Descriptor

The source traffic descriptor is the subset of traffic parameters requested by the source (user), which characterizes the traffic that will (or should) be submitted during the connection [13]. The relation of each traffic parameters referring to the ATM Forum [10] used in the simulation model is defined below.

- PCR = λ_a = 1/T in units of cells/second, where T is the minimum intercell spacing in seconds (i.e., the time interval from the first bit of one cell to the first bit of the next cell). This research focuses on four cases as follows.
  a. *PCR = λ_a = 423.94 Mbps (999,739 cells/s).*
     Hence, T = 1.0 µs (1/999,739 s).
  b. *PCR = λ_a = 212 Mbps (499,933 cells/s).*
     Hence, T = 2.0 µs (1/499,933 s).
  c. *PCR = λ_a = 141.31 Mbps (333,288 cells/s).*
     Hence, T = 3.0 µs (1/333,288 s).
  d. *PCR = λ_a = 105.9 Mbps (249,966 cells/s).*
     Hence, T = 4.0 µs (1/249,966 s).
- *CDVT* = τ in seconds. This traffic parameter normally cannot be specified by user, but is set instead by the network. Recommendation I.371 defines the minimum CDVT at a public UNI. For LB mechanism, a single bucket depth of *CDVT* cells and a nominal cell inter arrival spacing *T*, note that approximately *CDVT/T cells* can arrive back-to-back.

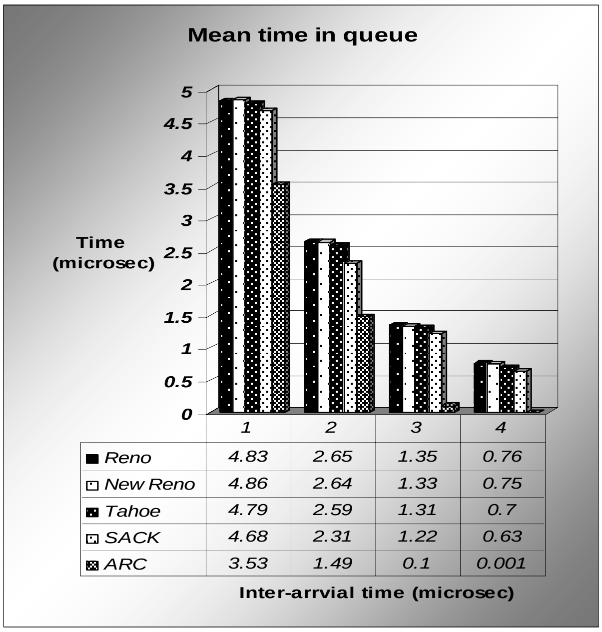
6. Results

The comparison between four schemes namely Tahoe, Reno, New Reno, SACK, the proposed ARC and the “Plain” scheme is illustrated in graphs. The experiment has been set the maximum window size to be 3 for total six schemes. With the burst/silence ratio 100:0, the average inter-arrival cell rate defines as 1, 2, 3, and 4 µs. Figure 4 illustrates the throughput against inter-arrival cell rate. Figure 5 illustrates mean time in queue against inter-arrival cell rate. Figure 6 illustrates mean queue length against inter-arrival cell rate. Figure 7 illustrates utilization of link against inter-arrival cell rate and Figure 8 illustrates cells drop against inter-arrival cell rate.



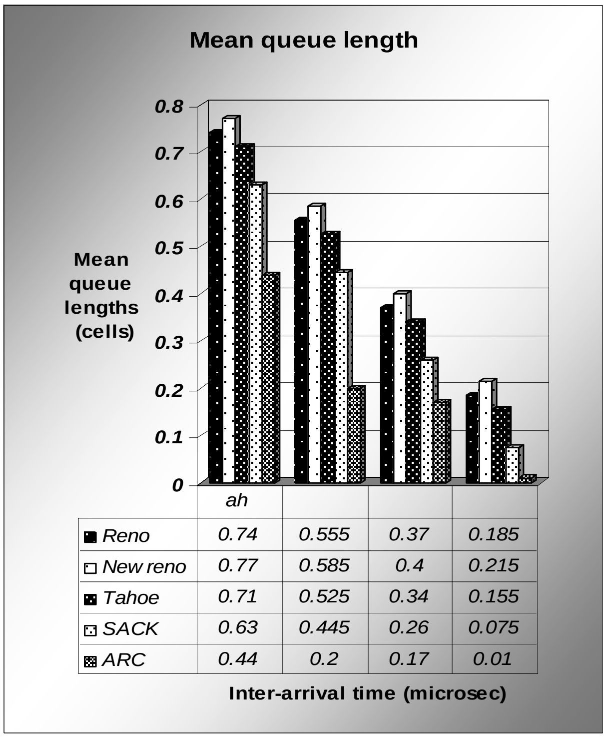
	1	2	3	4
■ Reno	6.44902	3.22451	2.59753	1.61255
□ NewReno	6.44902	3.22451	2.59753	1.61255
▣ Tahoe	5.93755	2.96877	2.39151	1.48438
⊡ SACK	6.79038	3.39519	2.7352	1.69459
▩ ARC	9.3252	5.52005	3.67187	2.75781

Figure 4. Throughput



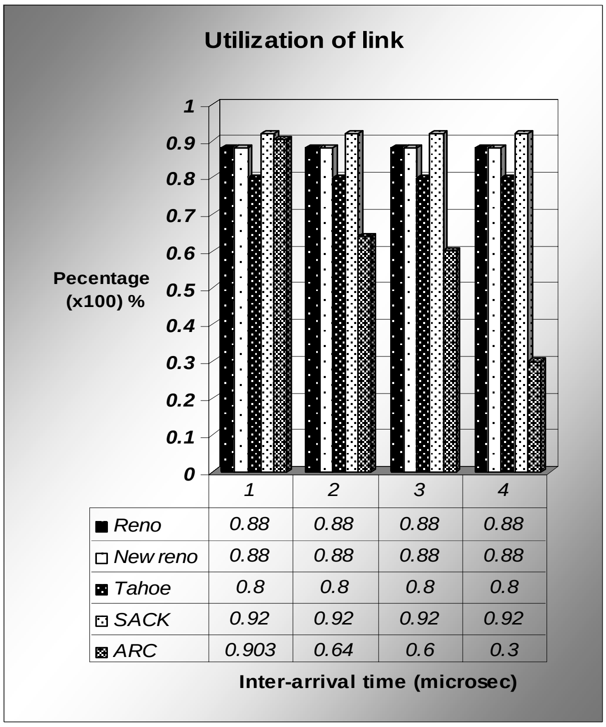
	1	2	3	4
■ Reno	4.83	2.65	1.35	0.76
□ New Reno	4.86	2.64	1.33	0.75
▣ Tahoe	4.79	2.59	1.31	0.7
⊡ SACK	4.68	2.31	1.22	0.63
▩ ARC	3.53	1.49	0.1	0.001

Figure 5. Mean time in queue



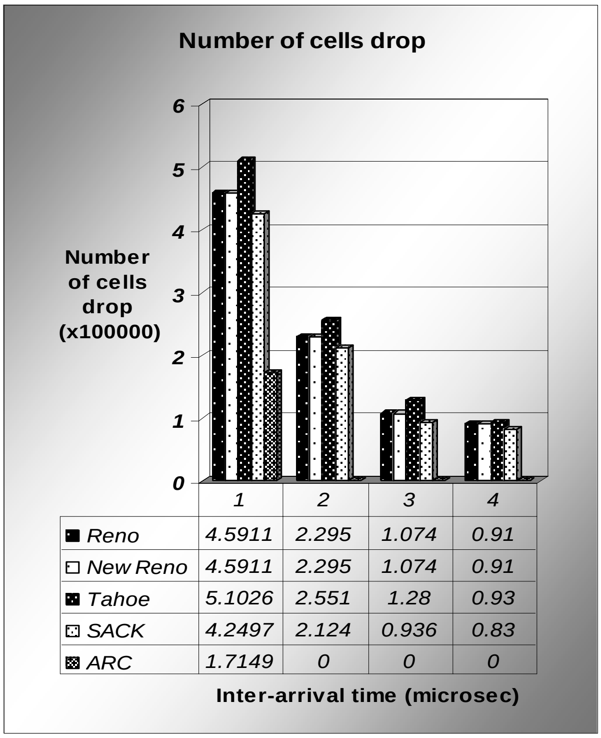
	ah			
■ Reno	0.74	0.555	0.37	0.185
□ New reno	0.77	0.585	0.4	0.215
▣ Tahoe	0.71	0.525	0.34	0.155
⊡ SACK	0.63	0.445	0.26	0.075
▩ ARC	0.44	0.2	0.17	0.01

Figure 6. Mean queue length (cells)



	1	2	3	4
■ Reno	0.88	0.88	0.88	0.88
□ New reno	0.88	0.88	0.88	0.88
▣ Tahoe	0.8	0.8	0.8	0.8
⊡ SACK	0.92	0.92	0.92	0.92
▩ ARC	0.903	0.64	0.6	0.3

Figure 7. Utilization of Link



	1	2	3	4
■ Reno	4.5911	2.295	1.074	0.91
□ New Reno	4.5911	2.295	1.074	0.91
▩ Tahoe	5.1026	2.551	1.28	0.93
⊡ SACK	4.2497	2.124	0.936	0.83
▨ ARC	1.7149	0	0	0

Figure 8. Number of Cells drop

From figure 4, ARC offers the best performance, followed by SACK at any input data rates. Figure 5 describes mean time cells have to reside in the waiting room. Plain scheme gives the longest delay time in queue while ARC contributes the shortest time, followed by four schemes equally. ARC is a disarmingly short network residual time with a higher number of throughputs compared to other four schemes and the Plain scheme. From figure 6, Plain scheme is the poorest as there are huge number of cells waiting in the queue. This could possibly gear to the future bottleneck problem. One of the most successful schemes is ARC by which most of the queue length is trivial, followed by SACK. As a result, the ARC provides a faster traverse time for variation of data rates over the high-speed network. To some extent, the performance ARC provides may be redundant, but from perspective of burst traffics it is clear that many users actually prefer working at a faster speed in transmission. Figure 7 illustrates all schemes keep the high-speed link (155.52 Mbps) busy most of the simulation time duration, especially when the congestion (inter-arrival time = 1 microsec) nearly approaches. However, as the input rate drops the ARC seems to perform more efficient than other five schemes. In figure 8, ARC will not have any problems with the dropped cells (cell loss). Plain scheme will and that could be a potential annoyance out of the way. With ARC, there might be an alternative solution to the burst traffics although it may raise some cell loss (but fewer) at the point of congestion.

7. Summary and Future Works

ARC will offer the highest performance in case of congestion (as the input arrival rate of burst traffic is higher than the ATM link capacity). Simulations demonstrate ARC outperforms compared to “Plain” and other four schemes. It does not have to be either costly or complicated but simply allows dropped cells to retransmit by regulating the window size directed to the arrival rate (λ_a) and number of dropped cells. On the other hand, ARC also gives remarkably better performance compared to five schemes (Tahoe, Reno, New Reno, SACK and Plain) in the case of non-congestion. There are many variations and the number of features available keeps asking about our future works. Predictably we have argued against this, so we would conduct some experiments on the extension of maximum window size. Different link capacities of high-speed networks and ARC extra-ordinary processing time will be further investigated. Also we plan to apply ARC scheme to the extent of low-speed wireless communication [17]. To experiment the case with wider size of the windows we need a lot more modifications than the current simulation. In fact we are on that boundary then our experiences tell it will be hard pressed to provide a decent running platform for our future works.

8. References

[1] M. Allman, V. Paxson, and W. Stevens. “TCP Congestion Control”, RFC 2581, 1999.

[2] Shigo Shioda and Hiroshi Satio, “Connection Admission Control Guaranteeing Negotiated Cell-Loss Ratio of cell Streams Passing through Usage Parameter Control,” IEICE Trans. Commu, vol E80-B, no. 3, page 399-411,1997.

[3] Pitiporn P. et. al, “Performance Comparison of ATM Policy Mechanisms with Telecommunications Traffic", Proceeding of The 6th World Multiconference on Systematics, Cybernetics and Informatics, pp. 373-378, 2002.

[4] Ghanthiya L. et. al, "Performance Evaluation for TCP/IP over Satellite Communication with Sliding Window Algorithms, "Proceedings of the 4th International Conference on Intelligent Technologies (InTech’03),pp.17-24, 2003.

[5] M. Mathis, J. Mahdavi, S. Floyd and A. Romanow, “TCP Selective Acknowledgement Options”, Internetdraft, RFC 2018, 1996.

[6] K. Fall and S. Floyd, “Simulation-based comparisons of Tahoe, Reno and SACK TCP”, Computer Communication Review, vol 26, pp. 5-21, RFC 2583, 1996.

[7] Dannis Yang, “TCP performance and flow control”, Candle computer report, vol 22. no. 2, 2000.

[8] Steven Low , “Equilibrium & Dynamic of TCP/AQM”, Sigcomm,2001, http://www.cds.caltech.edu/~doyle/Networks/TCPaqm.ppt

[9] D.D., Kouvatsos, “Performance Modelling and Evaluation of ATM Networks,” Vol. 1. Chapman & Hall,1995.

[10] ATM Forum, “User-network interface (UNI) Specification version 3.1,” The ATM Forum Technical Committee, Prentice Hall, 1995.

[11] B. Khoshnevis, “Discrete Systems Simulations,” McGraw-Hill, 1994.

.

[12] E.P. Rathgeb, ”Modelling and performance comparison of policing mechanisms for ATM networks,” IEEE Journal on selected areas in communications, Vol. 9, No. 3, pp.325 – 334, 1991.

[13] M. Aida and H. Saito, “Traffic contract parameters and CAC guaranteeing cell-loss ratio in ATM networks,” IEICE Trans. Commu., vol.E78-B,no.1, 1995.

[14] S.S. Al-Wakeel; N.-E. Rikli; and A.A. Al-Wehaibi, “Evaluation of Fairness Strategies for ATM Congestion Control Policing Mechanisms,” IASTED Proceeding of International Conference on Applied Modeling & Simulation (AMP’99), 1999.

[15] Ghanthiya L. et. al, "Evaluation of TCP over Satellite Network with the Proposed Sliding Window Algorithms," Proceedings of the 6th IEEE International Conference on Advanced Communication Technology (ICACT’04),pp.917-922, 2004.

[16] S. Prahmkaew et. al, "A Study of Adaptive Rate Algorithm in ATM Network Flow Control," Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA’04),pp.668-673, 2004.

[17] S. Prahmkaew et. al, "BTSnet: Model and Usage of Regular Traffic Patterns in Bangkok for Mobile Ad Hoc Networks and Inter-Vehicular Communications," Proceedings of the International Mobility Conference (MobiCon ’04),S1-4, pp.1-9, 2004.

He received the B.S. and M.S. degree in computer science from Assumption University, Bangkok, Thailand in 1995 and 1998 respectively. He has been a Ph.D. candidate in telecommunications science in the faculty of science and technology, Assumption University since 2002. His research interests include ATM network performance evaluation, fiber optic communication and mobile Ad Hoc networks.

Mr. Chanintorn Jittawiriyanukoon received the B.Eng degree in electrical engineering from Kasetsart University, Bangkok, Thailand in 1984, the M.Eng. and D.Eng. in communications engineering from Osaka University, Osaka, Japan in 1987 and 1990 respectively. He is presently an assistant professor of faculty of science and technology, Assumption University of Thailand. He is also a leader in a data communication research group at the faculty. His research interests include satellite communications, performance evaluation of parallel processing, fast routing algorithm and high speed networks.