Performance Analysis of ABR Congestion Control Algorithm using Self-Similar Traffic

Dong-Il Kim, Sung-Ho Jin, *Member, KIMICS*

Abstract—One of the most important issues in designing a network and realizing a service is dealing with traffic characteristics. Recent experimental research on LAN, WAN, and VBR traffic properties has highlighted that real traffic specificities can not be displayed because the current models based on the Poisson assumption under estimate the long range dependency of network traffic and self-similar peculiarities. Therefore, a new approach using self-similarity characteristics as a real traffic model was recently developed. In This paper we discusses the definition of self-similarity traffic. Moreover, real traffic was collected and we generated self-similar data traffic like real traffic to background load. On the existing ABR congestion control algorithm transmission throughput with the representative ERICA, EPRCA and NIST switch algorithm show the efficient reaction about the burst traffic.

Index Terms—ATM, Self-Similar Traffic, ABR Congestion Control

I. INTRODUCTION

The congestion control algorithm based on transmission rate for ABR(Available Bit Rate) service in ATM network consists of explicit rate feedback algorithm and binary feedback algorithm. There have been several kinds of congestion control algorithms, which were representatively suggested through ATM Forum, such as EFCI(Explicit Forward Congestion Indication) algorithm of explicit rate feedback algorithm, EPRCA(Enhanced Proportional Rate Control Algorithm) algorithm of binary feedback algorithm, ERICA(Explicit Rate Indication for Congestion Avoidance) algorithm using congestion avoid method, NIST(National Institute of Standards and Technology) algorithm developed by NIST, USA, so far. But it appears that manufactured switches based on the results of study of performance analysis about these algorithm have many differences in their performances in the event of adapting to the real network. One of the most important matters in designing networks and realizing service, is to grip on the traffic characteristics.

Conventional assumptions about traffic data include that the multiplication of large numbers of independent traffic streams should be based on the Poisson procedure.

However, since network traffic involves multimedia and an increasing number of users, the conventional Poisson model can result in a burst in the system characteristics when transferring a large-size file. To solve these problems, A recent paper on network traffic stated that LAN, WAN, and VBR video traffic could be more effectively modeled using a self-similar procedure. This method has quite different theoretical characteristics compared to the Poisson procedure, which only considers short-time dependence. The self-similar procedure has the same statistical characteristics, i.e. regardless of variations in aggregation on a wide time-scale. These characteristics are called self-similarity, including long-range dependence, infinite variance syndrome, and slowly decaying variance. This paper presents the basic concept of the self-similarity theory. Then, the existing Poisson traffic and self-similarity traffic models are compared using real network traffic. Finally, we generated self-similar data traffic like real traffic to background load. On the existing ABR congestion control algorithm transmission throughput with the representative EFCI, ERICA, EPRCA and NIST switch algorithm show the efficient reaction about the burst traffic.

However, since network traffic involves multimedia and an increasing number of users, the conventional Poisson model can result in a burst in the system characteristics when transferring a large-size file. To solve these problems, A recent paper on network traffic stated that LAN, WAN, and VBR video traffic could be more effectively modeled using a self-similar procedure. This method has quite different theoretical characteristics compared to the Poisson procedure, which only considers short-time dependence. The self-similar procedure has the same statistical characteristics, i.e. regardless of variations in aggregation on a wide time-scale. These characteristics are called self-similarity, including long-range dependence, infinite variance syndrome, and slowly decaying variance. This paper presents the basic concept of the self-similarity theory. Then, the existing Poisson traffic and self-similarity traffic models are compared using real network traffic. Finally, we generated self-similar data traffic like real traffic to background load. On the existing ABR congestion control algorithm transmission throughput with the representative EFCI, ERICA, EPRCA and NIST switch algorithm show the efficient reaction about the burst traffic.

Manuscript received March 24, 2004.

Dong-Il Kim is with department of information communication engineering, Dongeui University, Busan, 614-714,Korea (phone : 051-890-1689, Fax : 051-890-1684, E-mail :dikim@dongeui.ac.kr)

Sung-Ho Jin is with department of information communication engineering, Dongeui University, Busan, 614-714,Korea (phone : 051-890-1684, Fax : 051-890-1684, E-mail : deodato@joins.com)

II. CHARACTERISTICS OF SELF-SIMILARITY

The self-similarity model for analyzing data communication traffic is an effective concept. Self-similarity means that the probability ratio on a dimension or on a different scale appears the same. As such, the time sequence of data traffic has the same pattern without any correlation to a particular degree of resolution. This is the key point of self-similarity. From a network performance point of view, self-similarity is the durability of a group, such that a group will become smoothed out, even it occurs within a short time, if enough time passes.

A. Definition of discrete time

For a stationary time series χ, the m-aggregated time series $x^{(m)} = \{x_k^{(m)}, k = 0,1,2,...\}$ is defined by adding an adjacent m block without overlapping the existing one. Equation (1) defines this[9].

$$x_k^{(m)} = \frac{1}{m} \sum_{i=km-(m-1)}^{km} x_i \qquad (1)$$

The process can be treated as self-similar if the statistics of the probability process in equation (1) have the same compressed copy. The process can also be recognized as self-similar to the parameter β(0<β<1) if the probability process x is equal to equations. (2) and (3).

$$Var(x^{(m)}) = \frac{Var(x)}{m^{\beta}} \qquad \text{Variance} \qquad (2)$$

$$R_{x^{(m)}}(k) = R_x(k) \qquad \text{Autocorrelation} \qquad (3)$$

Parameter β can be remarked as H=1-(β/2) relative to the hurst parameter H. The static and ergodic processes are the case of β=l, the dispersion of a time average decrease with 1/m ratio. However, in a self-similar process, the decrease is slower ($\frac{1}{m^{\beta}}$). Therefore, H=0.5 means the absence of self-similarity and, as H becomes close to 1, the extent of the continuance or long-range subordination will become larger. If the probability process x is equal to eqs. (4) and (5) for k which is large enough, it can be assumed to be approximately self-similar.

$$Var(x^{(m)}) = \frac{Var(x)}{m^{\beta}} \qquad \text{Variance} \qquad (4)$$

$$R_{x^{(m)}}(k) \rightarrow R_x(k), \text{ as } m \rightarrow \infty \qquad \text{Autocorrelation} \qquad (5)$$

As such, a self-correlation in an assembly process, based on the definition of self-similarity, has the same configuration as the original one. This means that a self-correlation in an assembly process, which includes different variations or bursts, will appear as identical on the time-scale[1,2].

III. SOURCE MODELING, TRAFFIC MEASUREMENT AND ANALYSIS

A. Self-Similar traffic modeling.

Self-similar traffic can be modeled by super-positioning pareto-like ON/OFF sources. These sources are performed alternatively between ON(when the burst package is transferring) and OFF periods(idle period), which are characterized by independent random variables that are distributed equally. If it is assumed that an each source can be divided by the same distribution, the superposition or multiplication of these dependant sources will be self-similar traffic. In the present study, one hundred dependant Pareto-like ON/OFF source models were generated for self-similar traffic modeling. It was also assumed that a unit packet occurred with CBR during an ON period and there was no traffic during an OFF period.

B. Measurement of real-time operating LAN traffic

A large amount of traffic was gathered to study the mathematical and statistical behavior of self-similarity traffic. As such, the current authors used their university network condition to measure a real-time operating network traffic sample. The sample measurement was performed on a 203.230.252 C-class network. An HP Internet Advisor was also used as a measurement tool. The measurement parameter was a total packet number measuring one million within a second. The measurement term was from 12 - 26, June, 2003.

C. Comparison of Long-range dependence

The self-similarity function was calculated to consider the LRD characteristics on each trace. Figure 2 shows the self-similarity function with a lag. In figure 2, the real traffic data and pareto distribution data exhibited a very high correlativity to the lag-k. This means the long-range dependence. In contrast, when using a Poisson distribution, the traffic had a correlation of '1' for k=0. However, as K increased close to '0' the traffic suddenly exhibited characteristics similar to white noise and a short-range dependence. Namely, the bursts in the real traffic had a continual influence on the switch's buffer. Plus, the operation performed using the existing model could not recreate the influence of real traffic because only a short-range dependence could be considered.

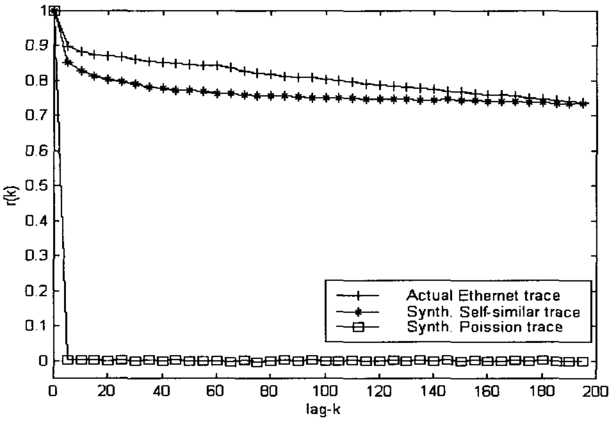


Fig. 1 Autocorrelation function for lag-k.

D. *Hurst parameter measurement*

The Hurst parameter measurements that can evaluate self-similarity include a variance-time plot, R/S plot, and periodergam. In this study the variance-time method was used. The definition of a variance-plot is that the dispersion of an m-aggregated time-sequence $x^{(m)}$ in a self-similarity process can be represented by equation (6).

$$Var(x^{(m)}) \approx \frac{Var(x)}{m^{\beta}} \qquad (6)$$

In equation (6), the self-similarity parameter H is 1-(β/2) and equation (6) can be rewritten as eq. (7) if $\log[Var(x)]$ does not have a relation to m.

$$\log[Var(x^{(m)})] \approx \log[Var(x)] - \beta \log(m) \qquad (7)$$

In equation (7), a line with the value of -β will result if $Var(x^{(m)})$ is plotted to m on a log-log graph. Moreover, the incline of this line represents a Hurst parameter. Figure 2 shows the variance-time plot of the traffic for the real traffic, self-similar traffic, and Poisson modeled traffic and Table 1 represents the respective Hurst parameters. The traffic generated artificially by modeling real and self-similar traffic had a high value close to 1. This means that the self-similarity intensity was very high. In contrast, the Poisson traffic had an H value of 0.5, that is, the same characteristics as white noise. Accordingly, these results demonstrate that the self-similar traffic provided an effective representation of the characteristics of the real traffic, whereas the Poison traffic did not.

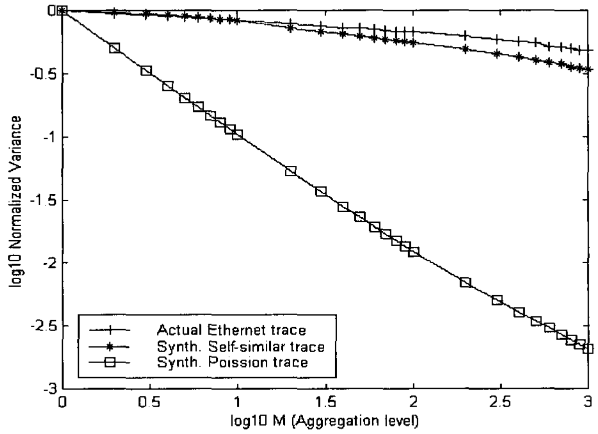


Fig. 2 Variance-time plot of each traffic.

Table 1 Hurst parameter estimation for each traffic

Traffic Trace	β	H
Actual trace	0.1054	0.9473
Synth. Self-Similar trace	0.1688	0.9156
Synth. Poisson trace	0.9478	0.5261

IV. ABR ALGORITHM

A. *EPRCA Algorithm*

For the EPRCA scheme, a switch keeps track of the average value of the current load on each connection, which is termed the mean allowed cell rate(MACR)

$$\mathrm{MACR(I)} = (1 - \alpha) \times \mathrm{MACR(I - 1)} + \alpha \times \mathrm{CCR(I)} \qquad (8)$$

where CCR(I) is the value of the CCR field in the Ith arriving FRM. This is an exponential avearge. Typically, α = 1/16, so that more weight is given to past values of CCR than the current value. Thus, MACR represents an estimate of the avearge load passing through the switch at the current time. If congestion occurs, the switch reduces each VC to no more than DPF × MACR, where DPF is a down pressure factor. Because all VCs are reduced to the same ER, the throttling is performed fairly. Specifically, when the queue length at an output port exceeds a threshold, all RMs for connections that pass through that port are updated as equation (9)

$$\mathrm{ER} \leftarrow \min[\mathrm{ER}, \mathrm{DPF} \times \mathrm{MACR}] \qquad (9)$$

A typical value for DPF is 7/8.

The EPRCA scheme reacts to congestion by lowering the ERs of VCs that are consuming more than their fair share of capacity. The next two schemes are congestion avoidance schemes that attempt to manage the ERs of all connections to avoid the onset of severe congestion. Schemes in this category make adjustments based on a load factor LF defined as equation(10)

$$\mathrm{LF} = \frac{Input\ rate}{Target\ rate} \qquad (10)$$

The imput rate is measured over a fixed averaging interval, and the target rate is set slightly below the link bandwidth(e.g., 85-90%). When LF > 1, congestion is threatened, and many VCs will have their rates reduced. When LF > 1, there is no congestion and rate reduction is not necessary.

B. *ERICA algorithm*

The ERICA algorithm defines the fair share for each connection as equation (11)

$$\mathrm{Fairshare} = \frac{Tarrget\ rate}{Number\ of\ connection} \qquad (11)$$

The current share used by a particular VC is defined as

$$\mathrm{VCshare} = \frac{CCR}{Input\ rate} \qquad (12)$$

This expand using Equation (10)

$$\mathrm{VCshare} = \frac{CCR}{Input\ rate} \times \mathrm{Target\ rate} \qquad (13)$$

The first term on the right-hand side says what fraction of the current load passing through this output port is due to this VC. Multiplying this by the target rate indicates the relative amount of the target rate that would be assigned to this VC if we simply adjusted all VC rates up or down so that the total input rate equals the target rate. Rather than adjust all VC rates up or down, ERICA selectively adjusts VC rates so that the total ER allocated to connections equals the target rate and is allocated fairly. This is achieved by using the as

$$ER = \max[Fairshare, VCshare] \tag{14}$$

Under low loads (LF < 1)each VC is assigned an ER greater than its current CCR, with those VCs whose VCshare is less than their Fairshare receiving a proportionately greater increase. Under high loads (LF > 1), some VCs are assigned an ER greater than their current CCR, and some are assigned a lower ER, done in such a wat as to benefit those VCs with the lesser shares.

C. NIST algorithm

The NIST algorithm uses an early congestion detection measure to tightly control its resources This early congestion detection technique consists of computing the derivative of the queue length, a load factor and a buffer queue threshold. A positive derivative indicates an instantaneous bandwidth demand that may be exceeding the available port bandwidth on the long run. Thus the switch declares the state of early congestion based on a positive derivative. The congestion state would not get cleared until the queue size falls below a certain threshold value (QT). This is important to eliminate the queue as soon as possible once the queue is built up due to congestion. This avoids the additional queue latency for consecutive incoming traffic. This will always insure that in the no congestion operation mode, the switch buffer will be maintained below QT. Therefore, the incurred buffer delay will be always less than the delay of QT.

The NIST algorithm monitors the aggregate queue input rate. It then uses that rate to compute the LF as equation(10)

The switch reads the ACR value contained in each forward RM cell that passes through. It then keeps a running exponential weighted average for all VCs ACR. This value constitute a Mean Allowed Cell Rate (MACR) that is common to the aggregate VCs sharing the same bandwidth. MACR is given by equation(15)

$$MACR = (1\text{-}AVF) \times MACR + AVF \times ACR \tag{15}$$

where AVF is a weight variable, and is chosen for example to be 1/16.

The exponential weighted average computed is similar to dividing the available line bandwidth equally among contending VCs. It does not have the Max-Min fairness criteria embedded in it. When one or more contending VCs have their rate controlled by some other node in the path, their rate might be further constrained to some lower value than the one given by MACR. In this case, the leftover bandwidth is reallocated based on small incremental steps at a time, Mean Additive Increase Rate (MAIR). The final MACR is adjusted according to the equation(16)

$$MACR=MACR+MAIR \tag{16}$$

At this point, the switch needs to come up with the ER that it is going to write in the backward RM cells. In the congestion operation mode,

$$ER = MACR * MRF \tag{17}$$

where MRF is the MACR Reduction Factor (default value=0.95). In the no congestion operation mode[3,4]

$$ER=MACR \tag{18}$$

V. SIMULATION AND ANALYSIS

Figure 3 is network model to see whether transmission rate executes fairly from each source about self-similar traffic. It established transmission speed of 155Mbps and 100Km between the switchs and 1Km between terminal and switch from each model. Table 2 is volume of traffic and value of fairshere to show from each source.

Figure 4 is a case of EPRCA algorithm to apply from ABR1. In this case, average transmission rate regardless of change of H value is seen regular. but if it compares with NIST algorithm to use similar form, change of transmission rate following change of H value has no change but it has a big change of amplitude. It is the reason that EPRCA algorithm reduces ACR about all sources as much as each

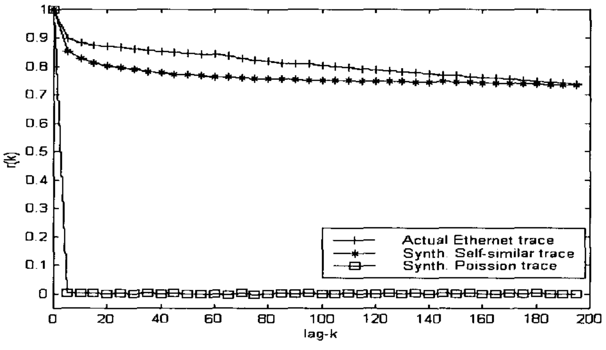
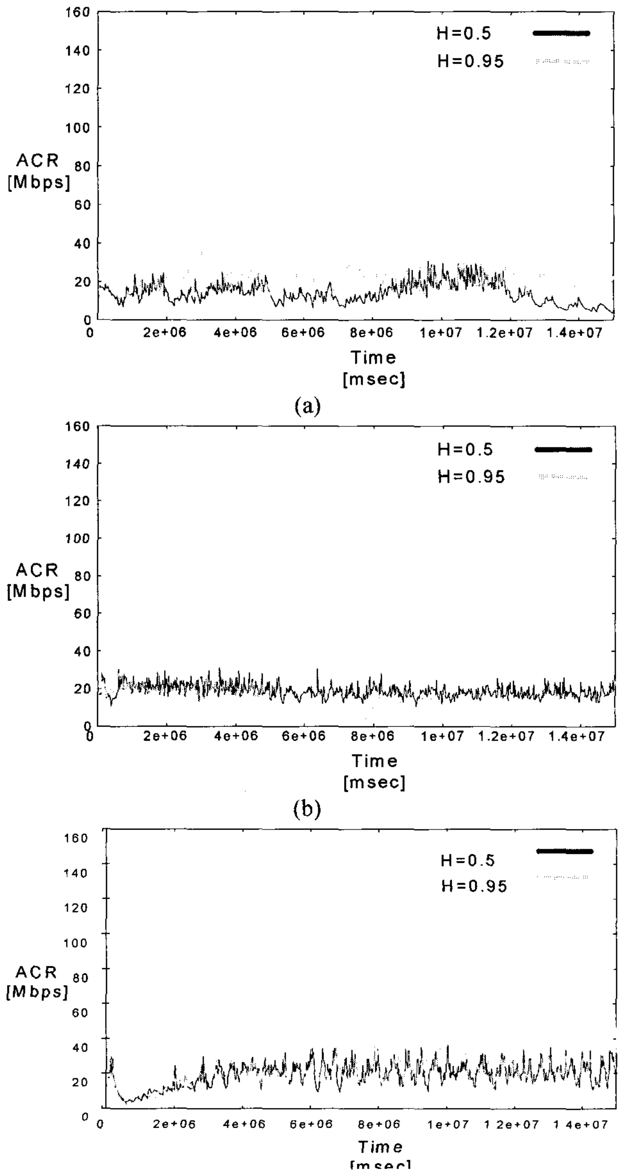


Fig. 3 Simulation network model.

Table 2. Transmission rate of each source

source	Transmission rate	Destination	Fairshare value
ABR 1	20Mbps	ABR 1-1	20Mbps
ABR 2	110Mbps	ABR 2-1	30Mbps
ABR 3	10Mbps	ABR 3-1	10Mbps
ABR 4	80Mbps	ABR 4-1	50Mbps
ABR 5	10Mbps	ABR 5-1	10Mbps
ABR 6	100Mbps	ABR 6-1	50Mbps
ABR 7	100Mbps	ABR 7-1	30Mbps
ABR 8	150Mbps	ABR 8-1	30Mbps
SS 1	40Mbps	SS 1-1	40Mbps

MRF to decrease transfer rate when this Algorithm meets extreme congested situation. If the congested situation is released, ACR goes to the value of PCR and it is also increased as much as multiplication between PCR and RIF by this sudden decrease. But this processing is not doing in one place. All sources are repeating the same progressing. Thus the switch gets congested quickly. It is also not the way of finding early congestion. So the queue n switch reach quickly to the threshold value. therefore the change of amplitude of transfer rate becomes big. because of such cause, fareshare has accomplished better than others form regardless of burst input. but grow longer han NIST algorithm because the queue in switch increases he amount used. On the other side, NIST algorithm is using MACR algorithm that is the same as EPRCA algorithm. But NIST is not sudden increase of transfer rate by using MAIR when transfer rate goes up. Accordingly, t is rare for the queue of switch to reach to the threshold value. Besides, by using the increase of the length of queue, NIST algorithm can

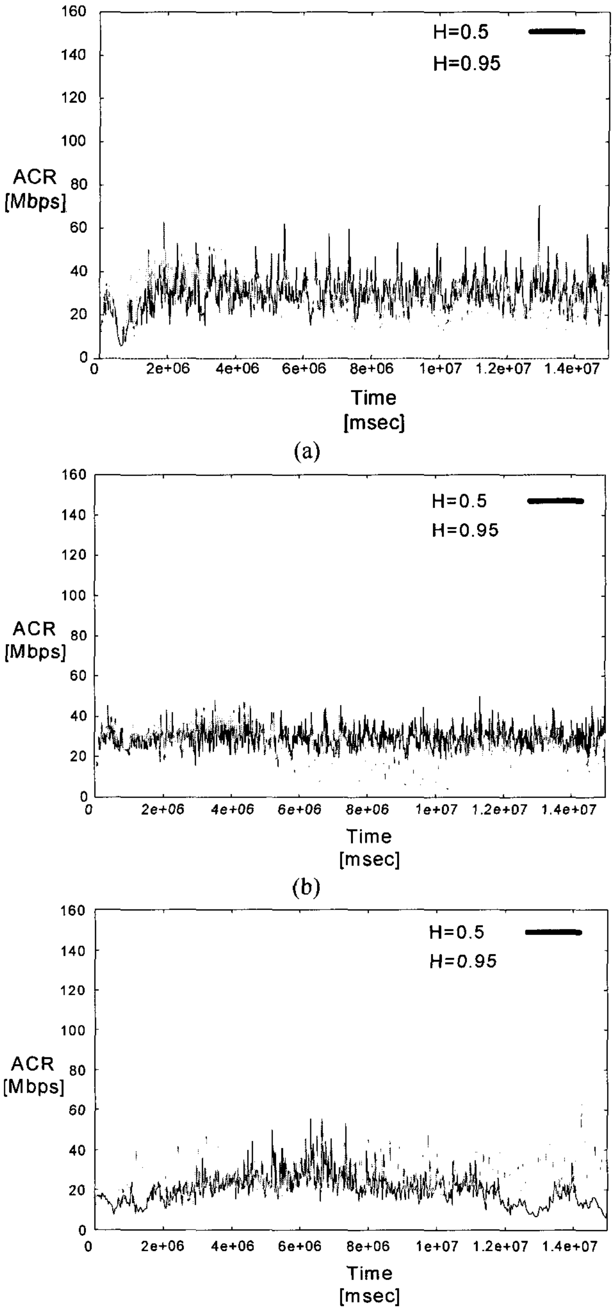


(a)

(b)

(c)

Fig 4. ACR of ABR 1 according to algorithm.

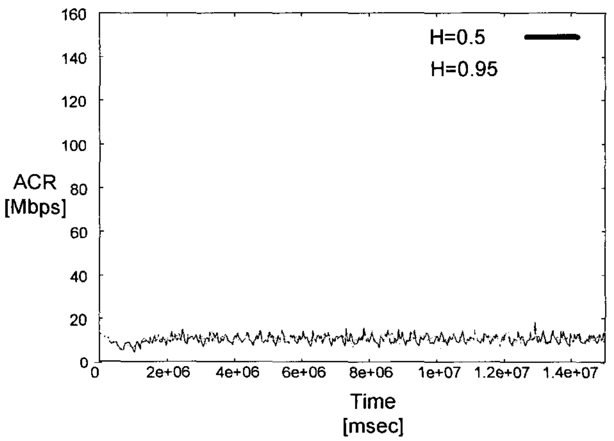


(a)

(b)

(c)

Fig 5. ACR of ABR 1 according to algorithm.



(a)

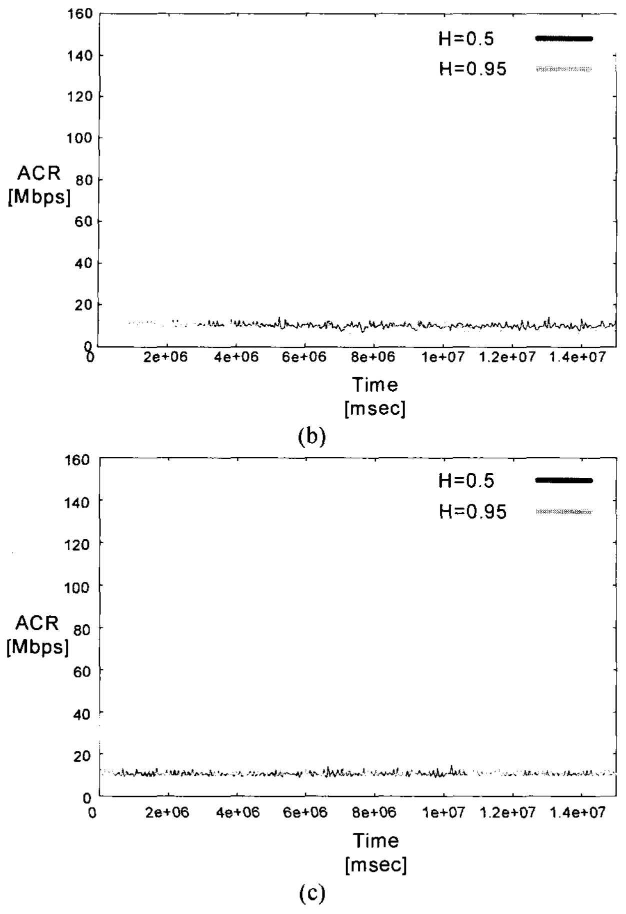
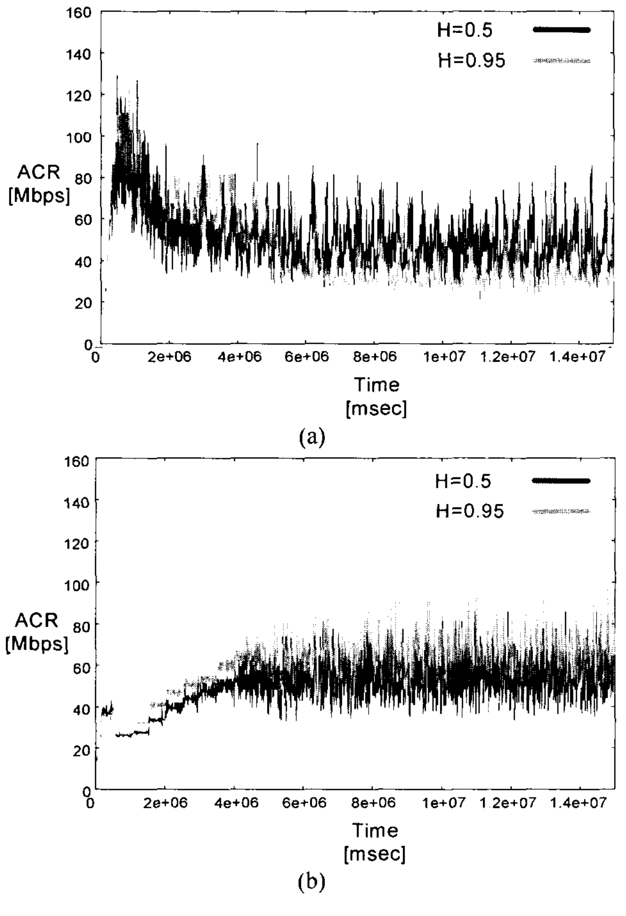


Fig 6. ACR of ABR 1 according to algorithm



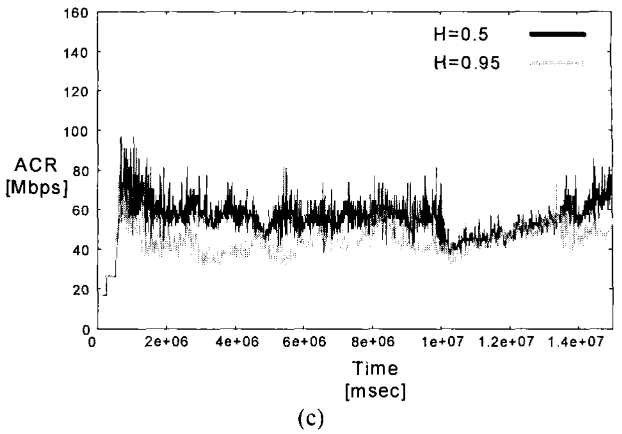


Fig 7. ACR of ABR 1 according to algorithm.

comprehend the congested situation quickly. For that reason, NIST algorithm can cope with the congested situation in its early stage. But the case of long distance source from destination, there is an element to make an error when NIST algorithm decides the value of ER passing through several switches and recognizing the level of congestion. (c) is the case that NIST algorithm is applied. the strength of burst traffic is getting heavier, ABR1 misunderstand congestion in network, and it can not reduce ACR during congestion. Therefor, In H is 0.5 and 0.95 case, H=0.95 can't prevent about burst input. And when H is 0.95,it has value of fairshare. But such case is just a case from wrong judge of network congestion. (b) is a case of ERICA algorithm. In case of fairshare, when we look at reactivity of the ERICA strength of burst, ERICA switch of its algorithm save the value of CCR per VC unit and then use it for calculation of the value of EF. Consequently, we can notice it change fast that the change of ACR in source according as the change of bandwidth. H=0.5 H=0.95 in the (b), there is transfer rate difference. when available bandwidth increases, short distance source from the congestion has the problem that the increasing of ACR becomes faster compare to long distance source and the more burst traffic is coming in, the more problems happen. Figure 5 presents ACR change according to the change of H result on the each algorithms with ABR2. (a) is the case that EPRCA algorithm is applied. According as increasing the strength of traffic burst, it is reacting properly. Therefore the sharing is executing very well. (b), (c) is the case that ERICA, NIST algorithms are applied. Although the input is higher than ABR1, fairshare according to the strength is presenting the same characteristic as ABR 1 Figure 6 shows the change of ACR according to the change of H result on the each algorithms with ABR3. (a) is the case that EPRCA algorithm is applied. It never makes complicated condition because of little input and the lack of distance. therefore it presents the same characteristic regardless of the strength of burst. About the fairshare decreased a little bit in proportion as the strength of input traffic burst becomes heavy. (b) is the case that ERICA algorithm is applied, we can notice the decrease of transfer rate when strength of burst becomes heavy. With the same reason we can notice that the fairshare is not reasonably executing. (c) is the case that NIST algorithm is applied. When the distance between sourse and destination is short. The control of transfer rate and fairshare is executing very well without the strength of input data burst. Figure 7 presents the change of ACR as the H result change with ABR4. (a) is the case that EPRCA algorithm is applied. the reaction according to the strength of burst shows the similar characteristic to ABR2.

H=0.95, the fairshare decreases a little as input traffic burst becomes heavy. (b) is the case that ERICA algorithm is applied. We can notice it oversensitive according to the strength of burst becomes heavy. We can notice the fairshare is not executing very well as the strength of input traffic becomes heavy because of the sensitiveness of ACR change. (c) is the case that NIST is applied. We can notice that it manages ABR properly when the strength of burst is heavy. Although the result of fairshare is a little bit light as the proper transfer rate controls, it is executing well concerning fairshare.

VI. CONCLUSION

In this paper we compared and analyze the Poisson distribution which was used in existing simulation model, self-similar traffic and real operating network traffic. We found that the traffic which Poisson distribution is different from the real operating network traffic so, we conclude that self-similar traffic is very similar to real operating network traffic model. And we applied self similar traffic and analyzed the reaction about switch algorithm EPRCA, ERICA, NIST among the typical ATM switch algorithm.

About fairshare NIST, EPRCA, ERICA, were shown the high quality in order. Through this analysis, when algorithms decide congestion of switch, deciding the level of congestion by using the condition of queue has better performance about burst traffic than the LF which is only used. We can notice it is required that the result of dynamic parameter compare to fixed constant parameter when ACR is changed. and that it is big different from real performance when traditional Poisson distribution is used in this case.

REFERENCES

[1] W. Willinger, Taqqu M., R. Sherman, "Self-Similarity Through High Variability," ACM/Sigcomm, pp. 100-113, 1995.

[2] Leland, W., Taqqu, M., Willinger, W., Wilson, D., "On the Self-similar Nature of Ethernet Traffic (Extended Version)," IEEE/ACM Trans-action on Networking, pp. 1-15, Feb, 1994.

[3] Iftekhar Hussain, Kuldip Bains, "An Explicit Rate ABR Algorithm for New-generation ATM Switches," Int. J. Network Mgmt, Vol. 9, pp. 323-338, 1999.

[4] Shivkumar Kalyanaraman, Raj Jain, Sonia Fahmy, Rohit Goyal, and Bobby Vandalore, "The ERICA Switch Algorithm for ABR Traffic Management in ATM Networks," IEEE/ACM Transactions on Networking, Vol. 8, No. 1, pp. 87-98, Feb, 2000.

Dong-Il Kim

He received the B.S., M.S, and Ph.D. degrees in electronic communication engineering from Kwangwoon University in 1981, 1983, and 1992 respectively. Since he joined LG R&D in march 1983, he participated in the various projects, related to electronic switching system. He was a member of technical staffs at LG R&D fro, 1983 to 1991. Currently, he is a professor of department of information communication engineering, Dongeui University, Busan, Korea. And he is also a editor of KICS(Korean Institute oc Communication Sciences) and strategic planning member of TTA (Telecommunication Technology Association of Korea). His current research interests include performance evaluation in switching system network and wireless network protocol. Dr. Kim is a member of the KIMICS.

Sung-Ho Jin

He received the B.S. degrees in electronic communication engineering from Dongeui University in 1994, M.S degrees in electronic engineering from Dongeui University in 1998 and Ph. D degrees in electronic communication engineering from Korea Maritime University. His current research interests include performance evaluation in ATM ABR algorithm and traffic analysis. Dr. Jin is a member of the KIMICS.