Characterization and Control of Highly Correlated Traffic in High Speed Networks

Ying-Dar Lin Tian-Ren Huang Yuan-Cheng Lai
Department of Computer and Information Science
National Chiao Tung University
Hsinchu, Taiwan
ydlin@cis.nctu.edu.tw

ABSTRACT

By assuming network traffic to be independent from each other, the analysis of network performance can be simplified. However, the real traffic sources may have some correlation which makes their behavior tends to converge or diverge. This phenomenon has tremendous influence on the congestion control. In this paper, we explain the possible reasons of the correlated behavior, namely, top-down and client-server correlation, and analyze their impacts. We also use the rapid matrix-geometric solution to investigate the cost to pay when applying the Leaky Bucket input control scheme on the independent, such as Poisson, traffic sources and the correlated, such as ON-OFF and HAP (Hierarchical Arrival Process), traffic sources.

1. Introduction

In high speed networks, we not only demand large volume data transmission but also need to control the transmission quality, such as packet delay, packet loss, jitters, etc. Because of the uncertainty in the behavior pattern of traffic sources, it costs much overhead to maintain the quality of services and tends to waste more resources. If we have a clear understanding of the traffic source behavior, a better control can be exercised to lower down network operation cost and improve transmission efficiency. Thus, efficient control mechanisms certainly require a solid understanding of the behavior pattern of traffic sources.

In the past, we often assume a traffic source to have a Poisson packet arrival rate. That is, the packet interarrival times are exponentially distributed. This implies that packet arrivals are independent from each other. Researches have shown that packet arrivals are highly correlated [1-3] and bursty [4-6]. Under this situation, it is difficult to control the network service quality since the results based on the Poisson assumption may not be valid.

In this paper, we present two fundamental reasons which cause correlated traffic. They are called *top-down* and *client-server* correlations. Traffic arrival processes are modulated by their upper-level parent processes or their peer processes. For example, when a user arrives, he or she will invoke applications. In the top-down correlation model, users and applications have hierarchical relationships: Only when a user in the upper level is active can the application in the lower level be activated. In the client-server correlation model, the client applications and their corresponding server applications have interactive relationships: They are usually activated together. We analyze the characteristics of both correlation models according to our definitions of burstiness and ratio of burst.

As we know, Leaky Bucket is an effective control mechanism for traffic policing and shaping. It can shape bursty traffic into a more regular stream and prevent a bursty traffic stream from damaging the performance of other traffic. Lots of papers have investigated its functions and effectiveness [7-9], but they often ignore the fact that it must pay cost at the same time. In this paper, we analyze queue length and packet loss Leaky Bucket scheme induces, and these conditions should be well noted when we use Leaky Bucket to control correlated traffic. Correlated traffic may suffer long queueing delay and high packet loss while passing the Leaky Bucket controller, which may result in violation of appointed QoS.

The paper is organized as follows. Section 2 describes and analyzes the top-down and client-server correlation models. In section 3, we present the cost analysis of Leaky Bucket fed by different types of correlated traffic; Section 4 presents the results and differences calculated from section 3 and compares their differences. Finally, conclusion is given in section 5.

2. Characteristics of Correlated Traffic

In this section, we present two models (top-down correlation model and client-server correlation model) to explain the reasons why correlated traffic happens and the differences between correlated traffic and non-correlated traffic. We quantify the characteristics of these models and compare them with the Poisson model which is the one we use most frequently to analyze the network performance.

0-8186-7617-5/96 $05.00 © 1996 IEEE

2.1 Analysis of Top-down Correlation Model

There are many hierarchical relationships in the real world. We observe that there are also hierarchical relationships between processes in the network.

The model is composed of many fundamental units. Each fundamental unit consists of a user and two applications. (See Fig. 1.) Each user and application is either active or inactive. Only when a user in the upper level is active can the application in the lower level be activated.

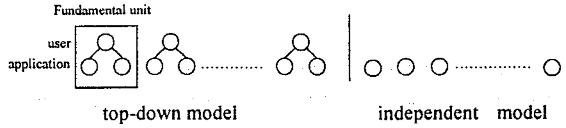

Fig. 1 The top-down correlation model and independent model

Assume that there are n fundamental units in the model and thus there are $2n$ applications in the lower level. The active probability of both user and application is, assumed to be identical for simplicity, p. The average number of active applications is $n \times p \times 2p = 2np^2$.

Compare with another model which is composed of $2n$ independent applications. (See Fig. 1.) The probability of each application being active is p^2. Note that both these two will have the same mean number of active applications, $2np^2$. But they have quite different distributions.

Below is the probability distribution function of the number of active applications in the top-down correlation model:

$$p(x) = \sum_{i=\left\lceil \frac{x}{2} \right\rceil}^{n} \left(C_i^n p^i (1-p)^{n-i} \times C_x^{2i} p^x (1-p)^{2i-x} \right),$$

where x is the number of active applications.

In the formula above, we compute the production of the probability that i users are active, $C_i^n p^i (1-p)^{n-i}$, and the conditional probability that x applications are active when there are i active users, $C_x^{2i} p^x (1-p)^{2i-x}$, for every possible i (from $\left\lceil \frac{x}{2} \right\rceil$ to n).

The probability distribution function of the number of active applications in the independent model is

$q(x) = C_x^{2n} \left(p^2\right)^x \left(1-p^2\right)^{2n-x}$, where x is the number of active applications.

In Fig. 2, we can see that the independent model has a distribution that is more concentrated around the mean. That is, a top-down correlated model will produce more bursty behavior pattern than the independent model. In the following, we compare the top-down correlated models with the same mean but different n, the scale of the system. From Fig. 3, we can find when n becomes larger, the distribution will scatter even wider and there is a higher probability that more applications will be active simultaneously.

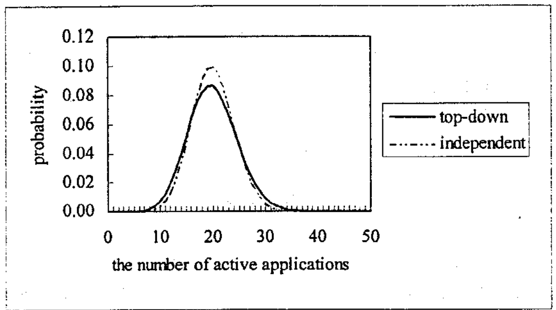

Fig. 2 The top-down model vs. independent model ($2n$=100, p^2=0.2)

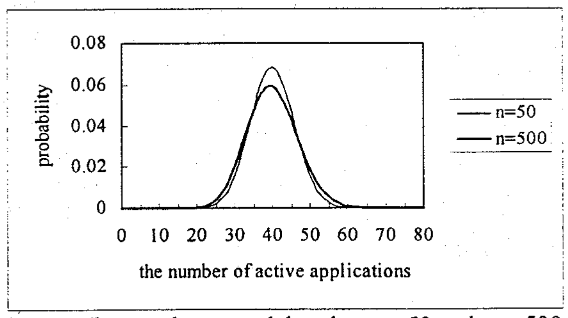

Fig. 3 The top-down model under n=50 and n=500, with same mean.

For this particular model, we define when the number of active applications exceed 1.5 times of its mean as "bursty state". And define "burstiness" as the probability that the system is in the bursty state, and the "ratio of burst" as the probability that an active application becomes active in a bursty state. We define these two as follows:

$$\text{burstiness} = \sum_{x > 1.5 \times \bar{x}} p(x)$$

$$\text{ratio of burst} = \sum_{x > 1.5 \times \bar{x}} x \times p(x) \Big/ \bar{x}$$

Let we use Fig. 4 to explain the term "ratio of burst". In the long run, the area under the curve of x is $\bar{x}$ which is the denominator of the definition formula. The nominator, on the other hand, is the sum of gray areas. Thus, the intuitive meaning of the definition is the percentage of bursty traffic among total traffic, or the probability that an application becomes active when the system is in a bursty state.

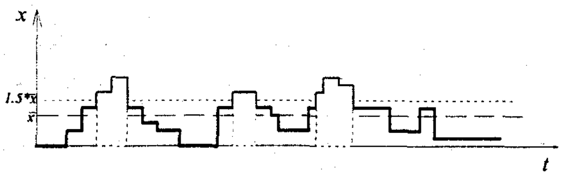

Fig. 4 The number of active applications

From Table 1, we can observe that under the same mean rate, the top-down model will raise variance by 1.17~1.39 times, , burstiness by 2.23~13 times and ratio of burst traffic by 2.24~13.1 times.

		top-down	independent	ratio
2n=100	variance	33.3	24.0	1.39
p^2=0.4	burstiness	2.34×10^{-4}	1.80×10^{-5}	13.0
(mean=40)	ratio of burst	3.64×10^{-4}	2.79×10^{-5}	13.1
2n=200	variance	41.9	32.0	1.31
p^2=0.2	burstiness	1.34×10^{-3}	2.78×10^{-4}	4.82
(mean=40)	ratio of burst	2.10×10^{-3}	4.32×10^{-4}	4.86
2n=400	variance	44.8	36.0	1.24
p^2=0.1	burstiness	2.00×10^{-3}	6.35×10^{-4}	3.15
(mean=40)	ratio of burst	3.14×10^{-3}	9.90×10^{-4}	3.17
2n=1000	variance	44.80	38.40	1.17
p^2=0.04	burstiness	2.11×10^{-3}	9.48×10^{-4}	2.23
(mean=40)	ratio of burst	3.32×10^{-3}	1.48×10^{-3}	2.24

Table 1 The difference of variance, burstiness and ratio of burst between top-down correlation model and independent model under same mean

		top-down	independent	ratio
2n=100	variance	20.9	16.0	1.31
p^2=0.2	burstiness	1.45×10^{-2}	6.06×10^{-3}	2.30
(mean=20)	ratio of burst	2.34×10^{-2}	9.69×10^{-3}	2.42
2n=200	variance	41.9	32.0	1.31
p^2=0.2	burstiness	1.34×10^{-3}	2.78×10^{-4}	4.82
(mean=40)	ratio of burst	2.10×10^{-3}	4.32×10^{-4}	4.86
2n=400	variance	83.8	64.0	1.31
p^2=0.2	burstiness	1.43×10^{-5}	7.37×10^{-7}	19.4
(mean=80)	ratio of burst	2.20×10^{-5}	1.127×10^{-6}	19.5
2n=1000	variance	209	160	1.31
p^2=0.2	burstiness	2.73×10^{-11}	2×10^{-14}	1239
(mean=200)	ratio of burst	4.13×10^{-11}	3×10^{-14}	1242

Table 2 The difference of variance, burstiness and ratio of burst between top-down correlation model and independent model under same mean

From this result, variance, burstiness or ratio of burst of the top-down model are obviously higher than those of the independent model. So, if we replace the top-down model with the independent model to analyze some traffic behavior, it is very probable the result will be very different from that in the real situation. On the other hand, under the condition of a fixed mean, the difference between the top-down model and the independent model in variance, burstiness and the ratio of burst will shrink as n increases. That is, if the mean is fixed, the top-down model will resemble the independent model as n increases.

Table 2 shows the variance, burstiness or the ratio of burst with some parameter p^2. From this table, we can see if the active probability p^2 is fixed, as n increases, the burstiness and the ratio of burst will decrease in both the top-down model and independent model. But they shrink much faster in the independent model than in the top-down model. As a result, the difference between the two models is getting larger and larger.

2.2 Analysis of Client-server Correlation Model

In addition to the hierarchical relationships, there are also interactive relationships, such as the cooperation between departments in a company to reach some goal. Applications in the networks also have this kind of relationship. We name it the client-server correlation.

The model is composed of $2n$ applications. (See Fig. 5.) There are n pairs with each pair being either active or inactive. The active probability is, again assumed to be identical, p. The average number of active applications is $2n * p = 2n\ p$. Compare with $2n$ applications which are independent. They have the same active probability and thus the same mean.

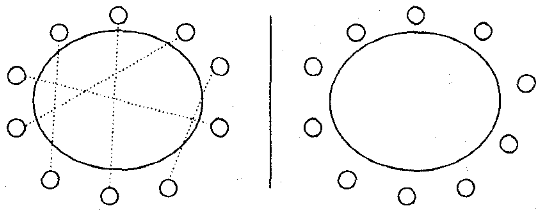

Fig. 5 The client-server correlation model and independent model

The probability distribution function of the number of active applications in the client-server correlation model is

$$p(x) = \begin{cases} C^n_{\frac{x}{2}} p^{\frac{x}{2}} (1-p)^{n-\frac{x}{2}} & ,x \text{ is even} \\ 0 & ,x \text{ is odd} \end{cases}$$

, where x is the number of active applications.

The probability distribution function of the number of active applications in the independent model is $q(x) = C^{2n}_x p^x (1-p)^{2n-x}$, where x is the number of active applications.

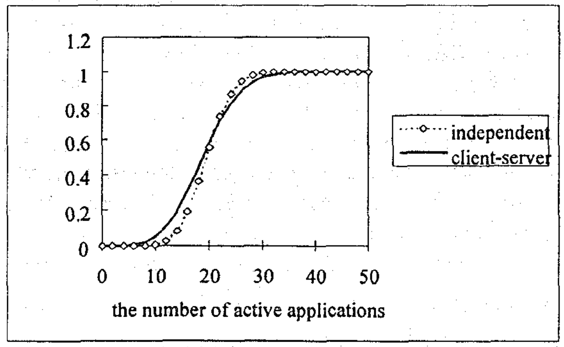

Fig. 6 The probability accumulation function of client-server model and independent model ($2n=100$, $p=0.2$)

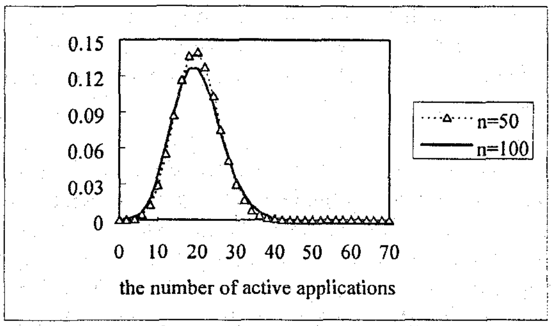

Fig. 7 $p(x)$ of client-server model under $n=50$ and $n=500$, with the same mean.

In Fig. 6, same with top-down correlation model, we can see that independent model has a distribution that is more concentrated around the mean than the client-server model. In Fig. 7 and Table 3, we compare the client-server model with different n under the same mean number of active applications. We can observe, as n increases, the distribution will scatter wider and the divergence between the client-server model and the independent model will decrease.

From Table 3, we observe that under the same mean, the client-server correlation model will raise variance by 2 times, burstiness by 12.5~76.1 times and ratio of bursty traffic by 12.9~78 times!

From Table 4, we observe that under the same p, the client-server correlation model will raise variance by 2 times, burstiness by 5.08~1809000 times and ratio of bursty traffic by 5.34~1823000 times! Obviously, when p is fixed and n is increased, the difference between the client-server model and the independent model will increase tremendously as a result. The behavior of the client-server model is much more bursty than that of the independent model, which is especially true when we scale up the system.

		client-server	independent	ratio
$2n=100$	variance	48	24	2
$p=0.4$	burstiness	1.37×10^{-3}	1.80×10^{-5}	76.1
(mean=40)	ratio of burst	2.17×10^{-3}	2.79×10^{-5}	78.0
$2n=200$	variance	64	32	2
$p=0.2$	burstiness	6.06×10^{-3}	2.78×10^{-4}	21.8
(mean=40)	ratio of burst	9.69×10^{-3}	4.32×10^{-4}	22.4
$2n=400$	variance	72	36	2
$p=0.1$	burstiness	9.51×10^{-3}	6.35×10^{-4}	15.0
(mean=40)	ratio of burst	1.53×10^{-2}	9.90×10^{-4}	15.4
$2n=1000$	variance	76.8	38.4	2
$p=0.04$	burstiness	1.18×10^{-2}	9.48×10^{-4}	12.5
(mean=40)	ratio of burst	1.91×10^{-2}	1.48×10^{-3}	12.9

Table 3 The difference of variance, burstiness and ratio of burst between client-server correlation model and independent model under the same mean.

		client-server	independent	ratio
$2n=100$	variance	32	16	2
$p=0.2$	burstiness	3.08×10^{-2}	6.06×10^{-3}	5.08
(mean=20)	ratio of burst	5.18×10^{-2}	9.69×10^{-3}	5.34
$2n=200$	variance	64	32	2
$p=0.2$	burstiness	6.06×10^{-3}	2.78×10^{-4}	2.18
(mean=40)	ratio of burst	9.69×10^{-3}	4.32×10^{-4}	22.4
$2n=400$	variance	128	64	2
$p=0.2$	burstiness	2.78×10^{-4}	7.37×10^{-7}	337
(mean=80)	ratio of burst	4.32×10^{-4}	1.13×10^{-6}	383
$2n=1000$	variance	320	160	2
$p=0.2$	burstiness	3.98×10^{-8}	2.02×10^{-14}	1810000
(mean=200)	ratio of burst	6.07×10^{-8}	3.33×10^{-14}	1823000

Table 4 The difference of variance, burstiness and ratio of burst between client-server correlation model and independent model under the same parameter p.

3. Leaky Bucket Traffic Control

We have mentioned the two reasons to cause correlation : top-down and client-server correlation. Because Leaky Bucket scheme is used in one end; it can not control the traffic in both ways, it is meaningless to use Leaky Bucket to control the traffic with the client-server correlation, Thus, we only introduce the HAP model [10] which generalizes the top-down model for our analysis. We analyze and compare the cost Leaky Bucket will pay under different input traffic sources, including Poisson model [11], ON-OFF model [12,13] and HAP model . The cost we focus includes system's throughput, the average queueing time of the packets, and the packet loss rate under the condition of limited buffer.

As we know, Leaky Bucket has constant token arrival rate and service rate, and this makes it very difficult to quantify the result of the Leaky Bucket by mathematical analysis. Therefore, in this section, we apply an approximate Poisson-Leaky-Bucket model by assuming both the token arrival rate and service rate are of Poisson distribution rather than a constant rate. This simplifies our analysis, but the accuracy of this approximation should be further evaluated.

3.1 Survey on Traffic Models

Below are three classic models. With these models, we can analyze the performance of traffic control and resource management schemes.

(1) **Poisson Model**: This classic model has been used extensively due to its simplicity in analytical derivations. However, this model has its limitation. Packets generated from Poisson processes have exponentially distributed interarrival times. That is, packet arrivals are independent from each others, which is not true in the real world. The index of dispersion of Poisson processes is always one, while the real traffic has a monotonically increasing index of dispersion [11].

(2) **ON-OFF Model**: An ON-OFF source alternates between two states: ON and OFF. When an ON-OFF source is in the ON state, it generates packets at the peak rate. When it is in the OFF state, it keeps silent.

The ON-OFF model, generating bursty traffic, represents the characteristics of, for example, graphic sources and voice sources with silent removal [12,13].

(3) **HAP** (Hierarchical Arrival Process) **Model**: A HAP has three levels--user, application and message. A set of parameters describes the arrival and departure processes at each level. The model captures the fact that a packet arrival process is modulated by its upper-level arrival processes.

The HAP model parameters are defined below. Suppose that the reciprocal of each parameter is the mean of its distribution:

- λ : user interarrival time distribution,
- μ : user service time distribution,
- λ_i : interarrival time distribution for application type i, (Application arrivals are enabled only during user service time),
- μ_i : service time distribution for application type I,
- λ_{ij} : interarrival time distribution for message type j of application type i, (Again, messages are generated during application life span.)
- μ_{ij} : service time distribution for message type j of application type i, where $i = 1..l$ and $j = 1..m_i$.

Fig. 8 shows the hierarchy of the HAP model. A user may invoke up to l different application types simultaneously. Each application i can generate m_i types of messages.

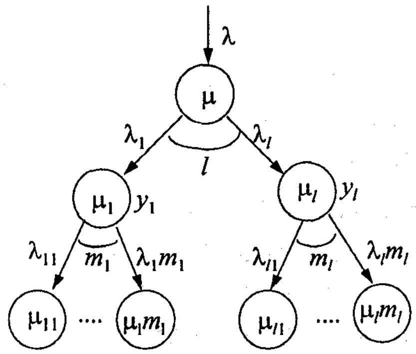

Fig. 8 The HAP model

3.2. Leaky Bucket on Poisson Traffic

For the sake of apprehending the influence of the Leaky Bucket on the Poisson traffic, we obtain the solution in numerical form by the method below. In Fig. 9, the Markov chain represents a Leaky Bucket with a Poisson data input rate, Poisson token input rate, and Poisson service time.

The state (i,j) means the buffer in the Leaky Bucket has i packets and there are j tokens in the token pool. We use a two-dimension array $P_{(I+1)\times(J+1)} = \left[P_{i,j}\right]$ to indicate the probability array, where $p_{i,j}$ is the probability of state (i,j), and

- α is the input rate of the token,
- μ is the rate that the packet finishes being serviced,
- λ is the rate that the traffic source produces new packets,
- I is the size of the buffer, and J is the size of the token pool.

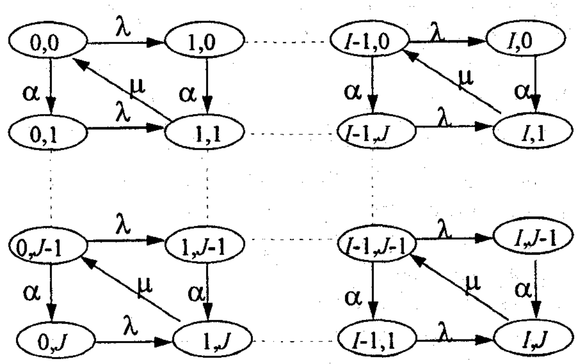

Fig. 9 The state-transition-rate diagram of Leaky Bucket on Poisson traffic

In the following operations, $p'_{i,j}$ represents the probability of state (i,j) in next iterative stage, which equals to the conditional probability that the system move from other states into state (i,j) minus the conditional probability that the system leaves state (i, j). Thus, we can write down the state transition equations as

$$p'_{i,j} = \lambda \times p_{i-1,j} + \alpha \times p_{i,j-1} + \mu \times p_{i+1,j+1} + (1-\alpha-\mu-\lambda) \times p_{i,j},$$

$$p'_{0,j} = \alpha \times p_{0,j-1} + \mu \times p_{1,j+1} + (1-\alpha-\lambda) \times p_{0,j},$$

$$p'_{I,j} = \lambda \times p_{I-1,j} + \alpha \times p_{I,j-1} + (1-\alpha-\mu) \times p_{I,j},$$

$$p'_{i,0} = \lambda \times p_{i-1,0} + \mu \times p_{i+1,1} + (1-\alpha-\lambda) \times p_{i,0},$$

$$p'_{i,J} = \lambda \times p_{i-1,J} + \alpha \times p_{i,J-1} + (1-\mu-\lambda) \times p_{i,J},$$

$$p'_{0,0} = \mu \times p_{1,1} + (1-\alpha-\lambda) \times p_{0,0},$$

$$p'_{0,J} = \alpha \times p_{0,J-1} + (1-\lambda) \times p_{0,J},$$

$$p'_{I,0} = \lambda \times p_{I-1,0} + (1-\alpha) \times p_{I,0},$$

$$p'_{I,J} = \lambda \times p_{I-1,J} + \alpha \times p_{I,J-1} + (1-\mu) \times p_{I,J}.$$

where $0<i<I$; $0<j<J$.

By repeating the operations above until the change of every element in P is less than some default value (10^{-9}), we can obtain the convergence probability of every state in the Markov chain.

By doing so, we can compute the effective throughput, say T, of the system. T equals the product of the probability of the server being busy and the service rate of the server, which is

$$T = \left(\sum_{i \neq 0, j \neq 0} p_{i,j} \right) \times \mu.$$

We can get the packet loss rate, L. L is equal to the difference of the rate of the input packets and the rate of the output packets divided by the rate of the input packets. And that is,

$$L = (\lambda - T)/\lambda.$$

3.3 Leaky Bucket on ON-OFF Traffic

We use a similar method to analyze the Leaky bucket on On-Off traffic. We represents this system by the state-transition-rate diagram in Fig. 10.

In the notation of state (f, i, j), f represents the state of the which the traffic source ($f = 0$ means the off-state; $f = 1$ means the on-state), i represents the number of data packets in the buffer, and j represents the number of tokens in the token pool.

In Fig. 10, we use the following parameters:

β_1 is the rate at which the traffic source transits from on-state to off-state.

β_2 is the rate at which the traffic source transits from off-state to on-state.

μ is the rate at which the packet finishes being serviced.

λ is the rate at which the traffic source produces new packets in on-state.

α is the input rate of the token.

Similarly, we use a three-dimension array to represent the Markov chain probability array and use the operations like those in section 4.2 to acquire the convergence solution in numerical form.

As the last section, we can compute the effective throughput T, of the system. T equals the production of the probability of the server is busy and the service rate of the server,

$$T = \left(\sum_{f, i \neq 0, j \neq 0} p_{f,i,j} \right) \times \mu.$$

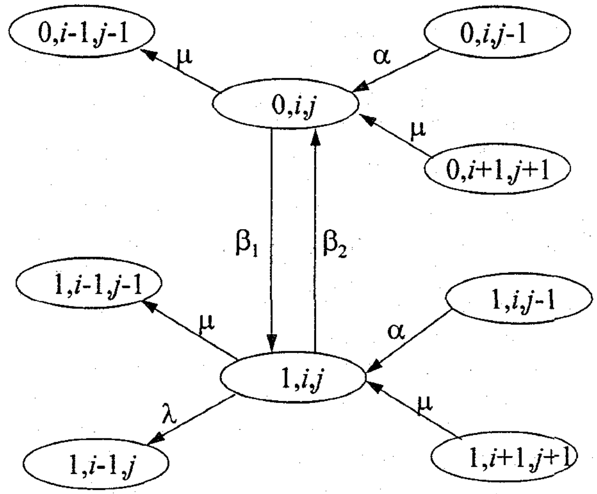

Fig. 10 The state-transition-rate diagram of Leaky Bucket on On-Off traffic

On the other hand, the average input rate of the traffic source, say $\bar{\lambda}$, is equal to the product of the probability of processing the data in on-state, ($\frac{\beta_1}{\beta_1+\beta_2}$), and the data input rate in on-state. Thus,

$$\bar{\lambda} = \left(\frac{\beta_1}{\beta_1+\beta_2} \right) \times \lambda.$$

By doing so, we can get the packet loss rate L. L is equal the difference of the rate of the input packets and the rate of the output packets divided by the rate of the input packets. And that is,

$$L = \frac{(\bar{\lambda} - T)}{\bar{\lambda}}.$$

3.4 Leaky Bucket on HAP Traffic

Analogous to the methods we mentioned in the previous two sections, we use the state-transition-rate diagram (Fig. 12(a)(b)) to represent the Leaky Bucket on HAP traffic. As we assume homogeneous user, application, and message, we have only one type in each level of the HAP hierarchy. Below are the parameters used in Fig. 11:

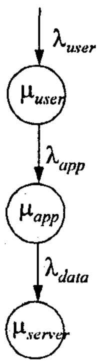

Fig. 11 The HAP model and its parameters

λ_{user} is the rate of the user's arrivals in HAP.

μ_{user} is the rate of the user's departures in HAP.

λ_{app} is the rate of the application's arrivals in HAP.

μ_{app} is the rate of the application's departures in HAP.

λ_{data} is the rate of the application produces the packets.

μ_{server} is the rate that the packet finishes being serviced.

α is the input rate of the token.

In the notation of state(x, y, i, j), x represents the number of the users in HAP. y represents the total number of the applications in HAP, i represents the number of the packets in the buffer, j represents the number of the tokens in the token pool.

Similarly, we use a four-dimension array to represent the Markov chain probability array and then use the repetitive operations to compute the convergence solution in numerical form.

Then, we can compute the effective throughput of the system, T, as

$$T = \left(\sum_{x,y,i\neq 0,j\neq 0} p_{x,y,i,j} \right) \times \mu_{server}$$

The average input rate of the traffic source, $\bar{\lambda}$, is computed by multiplying the rate at which every application produces the packets and the average number of applications, which is

$$\bar{\lambda} = \left(\sum_{x,y,i,j} y \times p_{x,y,i,j} \right) \times \lambda_{data}$$

Packet loss rate, L, is equal to the difference of the rate of the input packets and the rate of the output packets divided by the rate of the input packets. And that is,

$$L = \frac{\left(\bar{\lambda} - T\right)}{\bar{\lambda}}$$

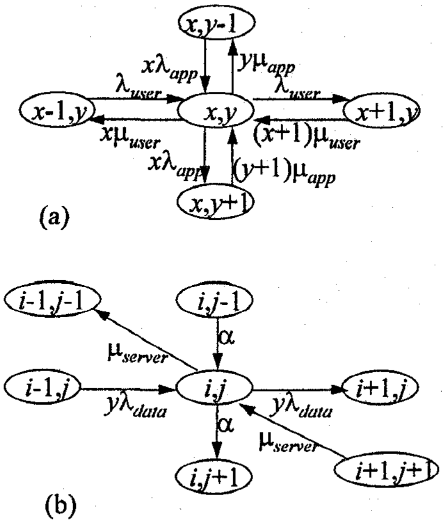

Fig 12 The state-transition-rate diagram of Leaky Bucket on HAP traffic

4. Analytical Results and Discussion

In this section, we use the method introduced in section 3 to quantify the behavior of Leaky Bucket and then compute and compare the cost Leaky Bucket will pay when the different traffic models as the input sources, including Poisson, ON-OFF and HAP models. The cost we are interested in includes queueing delay, and packet loss.

We use the parameters in the simulation: buffer size $I = 50$, token tool size $J = 25$.

(1) Leaky Bucket on Poisson vs. Pure Poisson

Input rate: Token rate: Transmission rate Input rate: Transmission rate	on Leaky Bucket on pure Poisson	Leaky Bucket on Poisson	pure Poisson
λ=0.008 : α=0.009 : μ=0.01	Queueing length	4.000	3.999
λ=0.008 : μ=0.01	Packet loss	0.1206%	0.0003%
λ=0.009 : α=0.0095 : μ=0.01	Queueing length	8.832	8.741
λ=0.009 : μ=0.01	Packet loss	0.2507%	0.05756%
λ=0.0095 : α=0.00975 : μ=0.01	Queueing length	15.82	14.83
λ=0.0095 : μ=0.01	Packet loss	0.4523%	0.4387%

Table 5 The difference between Leaky Bucket on Poisson traffic and pure Poisson traffic

In Table 5, under the condition of different loads, we compare the Leaky Bucket on Poisson traffic and pure Poisson traffic. Then, we can observe, their average queueing lengths and packet loss rates are of few differences at various parameters, even when the load is as high as 95%. The Leaky Bucket could limit the long-term transmission rate of the Poisson traffic to an amount which is not greater than the token input rate but does not raise the packet loss rate and average queueing delay too much, under the condition that the short-term transmission rate remains the same.

(2) Leaky Bucket on ON-OFF vs. Pure ON-OFF

Input rate λ	token rate α	transmission rate μ	β_1 $(=\beta_2)$		Leaky Bucket on ON-OFF	pure ON-OFF
0.008	0.009	0.01	0.001	Queueing length	15.66	13.05
				Packet loss	3.19%	1.70%
0.008	0.009	0.01	0.1	Queueing length	5.002	4.157
				Packet loss	0.0119%	0.000559%
0.0095	0.00975	0.01	0.001	Queueing length	23.24	21.68
				Packet loss	9.11%	7.51%
0.0095	0.00975	0.01	0.1	Queueing length	23.48	15.16
				Packet loss	1.85%	1.848%

Table 6 The difference between Leaky Bucket on ON-OFF traffic and pure ON-OFF traffic

In Table 6, we compare the Leaky Bucket on ON-OFF traffic source and pure ON-OFF traffic source with different parameters. Using Leaky Bucket method, only long-term transmission rate is guaranteed to be not greater than the token input rate and the short-term transmission rate is allowed to reach the server rate. Therefore, compared with the pure ON-OFF traffic, Leaky Bucket on ON-OFF traffic would have more packets waiting in the buffer. This phenomenon makes both the queueing delay and packet loss rate raise.

On the other hand, the higher the switching rate, (β_1,β_2), of the traffic source between On and OFF state, the lower the average queueing delay and packet loss rate. Because when this switching rate is high, the ON-OFF traffic source will have a more smooth traffic pattern. On the contrary, when this switching rate is low, the traffic source may remain in ON state for a long period of time. A stream of bursty traffic will be produced.

(3) Leaky Bucket on HAP vs. Pure HAP

In Table 7, we compare the Leaky Bucket on HAP traffic source and pure HAP traffic source with different parameters. No matter there exist a Leaky Bucket or not, we can observe the obvious queueing delay and packet loss using the HAP traffic model.

Owing to the high correlation of HAP traffic model, its traffic pattern is very bursty. Unless there are huge amount of buffers, the packet loss rate is still very high even when the average load is only medium.

λ_{user}	μ_{user}	λ_{app}	μ_{app}	λ_{data}		Leaky Bucket on HAP	pure HAP
0.03	0.01	0.03	0.02	0.0021	Queueing length	23.56	13.72
					Packet loss	15.62%	9.139%
0.03	0.01	0.05	0.02	0.00127	Queueing length	18.97	5.929
					Packet loss	8.397%	4.631%
0.03	0.09	0.045	0.01	0.00070	Queueing length	4.566	3.313
					Packet loss	2.418%	1.674%

Table 7 The difference of average queueing length and packet loss rate between Leaky Bucket on HAP traffic and pure HAP traffic(average load≅0.7, μ_{server} =0.012; α =0.01).

(4) Leaky Bucket on Poisson vs. Leaky Bucket on On-Off

We now observe an ON-OFF traffic source. When it is in the ON state, it will behave as a Poisson traffic source with a generating rate 2λ. However, when it is in the OFF state, it does not produce any traffic. If this ON-OFF traffic source switches between the ON and OFF state very rapidly, then the whole behavior is just like a Poisson traffic source with a generating rate λ. In contrast, if the switching rate between the ON and OFF state is low, then the whole behavior is very different from that of the Poisson traffic source though their average input rates are the same.

Comparing Table 5 with Table 6, when ON-OFF traffic source has larger switching rate α and β, the queueing delay and packet loss rate are very close to those of Poisson traffic source. When α and β are smaller, the ON-OFF traffic source produces a more bursty traffic pattern. The pattern causes higher queueing delay and packet loss rate.

(5) Leaky Bucket on Poisson vs. Leaky Bucket on HAP

From Table 5 and Table 7, when we use Leaky Bucket to adjust the Poisson traffic source, the long-term transmission rate of the traffic is restricted and the short-term transmission rate is unchanged at the same time. It does not raise the delay and loss rate much. However, in a HAP traffic source, the high correlation of HAP will cause a long period of bursty traffic, it is hard to shape the HAP traffic.

As we can see, Leaky Bucket could achieve, with little cost, its claimed effect on less bursty traffic. But for highly bursty traffic, it produces extremely large queueing delay and packet loss, and this is what users do not expect. Therefore, we have to improve the network control scheme to lower down the cost in controlling bursty traffic.

5. Conclusion and Future work

In this paper, we analyze the differences between the correlated traffic models and noncorrelated traffic models. A fundamental analysis on the top-down and client-server correlation is conducted. We observe, that the correlated traffic has higher burstiness than noncorrelated traffic, and this phenomenon becomes more serious as the network scale increases. When the network size increases, burstiness and ratio of burst are lowered in the correlation model and independent model. But it shrinks much faster in the independent model than in the correlation model. As a result, the difference between the two models is getting larger and larger.

Moreover, we use an approximate but rapid Matrix-geometric solution to analyze the control effects the Leaky Bucket has on various traffic sources. Then, we observe, the Leaky Bucket can control Poisson traffic effectively and it will not bring about too high packet loss and too long queueing delay.

However, under the condition of bursty traffic from HAP and ON-OFF models, Leaky Bucket is not as effective. It produces much higher queueing delay and packet loss in both models than in Poisson model, which may result in violation of appointed QoS.

In the future, we shall continue analyzing the performance of Leaky Bucket fed by different input traffic models, while using the constant token and server rate rather than Poisson distribution. We shall run simulations to verify our analysis results. Also, we hope to obtain more solid knowledge about the real traffic in the future and try to find the correspondence between the real traffic and HAP traffic model. Finally, using the knowledge of real traffic and realistic traffic models, we could develop much better congestion control schemes.

Reference

[1] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the Self-similar Nature of Ethernet Traffic (Extended Version)", *IEEE/ACM Trans. on Networking,* vol. 2, no. 1, pp. 1-15, Feb. 1994.

[2] I. I. Makhamreh, D. McDonald, and N. D. Georganas, "Approximate Analysis of A Packet Switch with Finite Output Buffering and Imbalanced Correlated Traffic," *Proc. ICC,* pp. 1003-1007, 1994.

[3] F. Kamoun and M. Mehmet Ali, "Queueing Analysis of ATM Tandem Queues with Correlated Arrivals", *IEEE INFOCOM'95,* Boston, vol. 2, pp. 709-716, Apr. 1995.

[4] Y. F. Jou, A. A. Nilsson, and F. Lai, "Tractable Analysis of A Finite Capacity Polling System under Bursty and Correlated ATM Arrivals," *Proc. ICC,* pp. 340-344, 1993.

[5] S. Turner, "Managing Bandwidth in ATM Networks with Bursty Traffic," *IEEE Network Mag.,* pp. 50-58, Sept. 1992.

[6] J. Sole-Pareta, and J. Domingo-Pascual, "Burstiness Characterization of ATM Cell Streams," *Computer Networks and ISDN System,* pp. 1351-1363, 1995.

[7] C. Partidge, Gigabit Networking, *Addison Wesley, Reading, Mass.*, pp. 255-263, 183-192. 1994.

[8] P. Pancha and M. El Zarki, "Leaky Bucket Access Control for VBR MPEG Video," *IEEE INFOCOM'95*, Boston, vol. 2, pp. 796-803, Apr. 1995.

[9] X. Wu, I. Lambadaries, H. Lee, and A. R. Kaye, "A Comparative Study of Some Leaky Bucket Network Access Schemes," *Proc. ICC,* pp. 1586-1591, 1994.

[10] Y. D. Lin, T. C. Tsai, J. Huang, and M. Gerla, "HAP: A New Model for Packet Arrivals," *Proc. ACM SIGCOMM'93*, San Francisco, pp. 212-223, 1993; also in *Journal of High Speed Networks,* vol. 2, no. 4, pp. 327-354, 1993.

[11] H. J. Fowler and W. E. Leland, "Local Area Network Traffic Characteristics, with Implications for Broadband Network Congestion Management," *IEEE Journal on Selected Areas in Communications*, vol. 9 no. 7, pp. 1139-49, Sept. 1991.

[12] A. La. Corte, A. Lombardo, and G. Schembra, "Modeling Superposition of ON-OFF Correlated Traffic Sources in Multimedia Applications," *IEEE INFOCOM'95,* Boston, vol. 3, pp. 993-1000, Apr. 1995.

[13] P. J. Kuehn, "From ISDN to IBCN (Integrated Broadband Communication Network)," *Proceedings of the World Computer Congress IFIP'89,* pp. 479-486, San Francisco, 1989.