以動態頻寬調整之方法降低差異式服務網路之用戶阻絕率

郭芳璋

國立宜蘭技術學院電子系助理教授

摘 要

為了簡化網路的複雜度，差異式服務網路將網路上的訊務分成少數的幾個類別，並指定適量的頻寬供其使用。每一個類別必須有足夠的剩餘頻寬才能允許新的呼叫請求使用網路，否則就會將其回絕。然而，當某一種類別缺少頻寬而回絕用戶之請求時，別的類別可能有相當多的剩餘頻寬。為了將頻寬有效的利用以降低阻絕率，本文提出一種動態頻寬調整的方法。而且，為了使同一種優先權的訊務能有近似的延遲，該方法並搭配「最糟狀況公平權重」排程法來處理路徑器中的訊務。在本文中，動態頻寬調整時機有二種：「即時調整」與「週期調整」，其中，週期調整又分10秒與100週期來討論。模擬結果顯示即時調整可以很有效地降低阻絕率，但必須花費相當的時間處理頻寬調整的事情。相反地，週期性調整的方法雖然不須系統隨時調整頻寬，但其反應能力也會較差，阻絕率也就較高。

關鍵詞：差異式服務網路、動態頻寬調整、阻絕率、最糟狀況公平權重

Dynamic Bandwidth Adjustment for Class-based Differentiated Services Networks

Fang-Jang Kuo

Assistant Professor, Department of Electronic Engineering, National Ilan Institute of Technology

Abstract

Differentiated Services (Diffserv) networks classify packets into one of a small number of aggregated classes. Each class is usually allocated with fixed bandwidth. Whether a new call request is admitted or rejected depends on whether the corresponding class has enough redundant bandwidth. This article presents a dynamic bandwidth allocation scheme which tries to adjust bandwidth between different classes so as to prevent call blocking. In order to balance the latency of packets of different classes, this scheme is cooperated with the ‘worst case weighted fair queuing’ method. Two time points, ‘real-time’ and ‘periodic,’ for adjusting bandwidth are discussed. The periodic adjusting scheme is evaluated with two periods, 10 and 100 seconds. The simulation results reveal that the real-time adjusting can effectively elevate the blocking ratio at the cost of processing load of adjusting work. On the contrary, the periodic adjusting scheme can relieve the processing time at the cost of higher blocking ratio.

Key Words：Diffserv networks, dynamic bandwidth allocation, blocking ratio, worst case weighted fair queuing

I. Introduction

Work on QoS-enabled IP networks has led to two distinct approaches: the Integrated Services architecture (Intserv) [1] and its accompanying signaling protocol, RSVP [2], and the Differentiated Services architecture (Diffserv) [3]. The integrated services architecture defined a set of extensions to the traditional best effort model of the Internet with the goal of allowing end-to-end QOS to be provided to applications. One of the key components of the architecture is a set of service definitions; the current set of services consists of the controlled load and guaranteed services. The architecture assumes that some explicit setup mechanism is used to convey information to routers so that they can provide requested services to flows that require them. While RSVP is the most widely known example of such a setup mechanism, the Intserv architecture is designed to accommodate other mechanisms.

Intserv services are implemented by "network elements". While it is common for network elements to be individual nodes such as routers or links, more complex entities, such as ATM "clouds" or 802.3 networks may also function as network elements. A Diffserv network (or "cloud") may be viewed as a network element within a larger Intserv network.

The current prevailing model of RSVP usage is based on a combined RSVP/Intserv architecture. In this model, RSVP signals per-flow resource requirements to network elements, using Intserv parameters. The following factors have impeded deployment of RSVP (and the Intserv architecture) in the Internet at large:

1. The use of per-flow state and per-flow processing raises scalability concerns for large networks.
2. Only a small number of hosts currently generate RSVP signaling. While this number is expected to grow dramatically, many applications may never generate RSVP signaling.
3. The necessary policy control mechanisms -- access control, authentication, and accounting -- have only recently become available [4].

In contrast to the per-flow orientation of RSVP, Diffserv networks classify packets into one of a small number of aggregated flows or "classes", based on the Diffserv codepoint (DSCP) in the packet's IP header. This is known as behavior aggregate (BA) classification [3]. At each Diffserv router, packets are subjected to a "per-hop behavior" (PHB), which is invoked by the DSCP. The primary benefit of Diffserv is its scalability. Diffserv eliminates the need for per-flow state and per-flow processing and therefore scales well to large networks.

Intserv, RSVP and Diffserv may be viewed as complementary technologies in the pursuit of end-to-end QoS. Together, these mechanisms can facilitate deployment of applications such as IP-telephony, video- on-demand, and various non-multimedia mission-critical applications. Intserv enables hosts to request per-flow, quantifiable resources, along end-to-end data paths and to obtain feedback regarding admissibility of these requests. Diffserv enables scalability across large networks.

From the perspective of Intserv, Diffserv regions of the network are treated as virtual links connecting Intserv capable routers or hosts (much as an 802.1p network region is treated as a virtual link in [5]). Within the Diffserv regions of the network routers implement specific PHBs (aggregate traffic control). The total amount of traffic that is admitted into the Diffserv region that will receive a certain PHB may be limited by policing at the edge. It is expected that the Diffserv regions of the network will be able to support the Intserv style services requested from the periphery.

The default bandwidth of each class is usually fixed. A new call of a certain class will be rejected if the bandwidth of the corresponding class is insufficient, even though another class has redundant bandwidth. This article presents a dynamic bandwidth allocation scheme to adjust bandwidth between different classes so as to relieve the blocking ratio of the system. Beside, in order to balance the latency of packets of different classes, the ‘worst case weighted fair queuing’ method is employed.

In the next section, the operation of the proposed dynamic bandwidth adjustment scheme is described in detail. The simulation results are presented in section 3. Finally, the article is ended with concluding remarks.

II. Proposed Dynamic Bandwidth Adjusting Scheme

1. Adjusting Quantity

The flow chart of the proposed scheme is shown in figure 1. When a call request arrives at the DiffSev network, the edge node of the network will dispatch it to one of the default classes, each of which is allocated with a default bandwidth. A class-i request is admitted by the edge node if the corresponding class has enough bandwidth. Otherwise, the edge node tries to increase the bandwidth of this class by decreasing that of another class who has redundant one. The principles for decreasing bandwidth of the target class are as follows:

(a). Only about one-half of the redundant bandwidth is transferred.
(b). The remaining bandwidth after being decreased must be more than one-third of the default initial bandwidth.
(c). The granularity of the transferred bandwidth is set as the maximal required bandwidth of a session of any class.

For class-i traffic, let the initial default bandwidth be B_i, the current allocated bandwidth be a_i, the bandwidth occupied by the existing sessions be o_i. Besides, assume the maximal required bandwidth of any session be r. Based on the principles listed above, the first step for the edge node to adjust bandwidth is checking whether another class has redundant bandwidth which is larger than twice of the maximal bandwidth of a connection (i.e., $a_j - o_j \geq 2r$). If negative, it tries whether another class can release bandwidth. If no class can release bandwidth, the request is rejected. If j class is suitable to be adjusted, the bandwidth is re-allocated as

$$\begin{cases} a_i = a_i + b_u \\ a_j = a_j - b_u \end{cases}, \text{ where } b_u = \begin{cases} \left\lfloor (a_j - o_j)/(2r) \right\rfloor r & (a_j + o_j)/2 \geq B/3 \\ \left\lfloor \left(a_j - \dfrac{B}{3} \right) \Big/ r \right\rfloor r & otherwise \end{cases},$$

2. Adjusting Time

The quantity for adjusting bandwidth has been presented in the last section. However, the time for adjusting the bandwidth would also effect the performance of the system. In this article, two ways, 'real-time' and 'periodic,' are discussed.

(a). Real-time adjusting:

This way has just been described in section 2.1 for illustrating the adjusting quantity. The node immediately tries to adjust the bandwidth, when a call request arrives at the network and finds that the bandwidth of the corresponding class is insufficient.

(b). Periodic adjusting

The adjusting work is performed periodically, and independent of the call request.

3. Class-Based Scheduling Scheme

After the request is admitted, the user can send packets to the network. Packets arrived at the node are served according to the class-based scheduling scheme. It is illustrated in figure 2. Initially, each of the high priority classes is allocated with the same bandwidth, (e.g. B). When the network is light-loaded, packets of the high priority classes are served before that of the low priority one. In order to keep the similar latency for the packets of the same class, the sessions of the class are fed to the same FIFO queue. Afterwards, the FIFO queues of different classes of the same priority are served in round-robin. At the last stage, packets are served according to their priority. That is to say, the low priority traffic, such as data, is served when no high priority traffic is pending.

When the bandwidth of any high priority traffic class is in-sufficient, the system will try to adjust bandwidth according to the procedure and time described in the previous section. It is predictable that the number of connections of a certain class increases with the increasing of the corresponding allocated bandwidth. Besides, the corresponding delay increases if the system still employs the round-robin method. The adjustment of the bandwidth of a class means the adjustment of the corresponding weight. As a result, the 'worst case weighted fair queuing' method is employed to avoid the unnecessary latency of the class which has increased bandwidth.

III. Simulation Results

This section describes the performance of the proposed dynamic bandwidth adjustment cooperated with the class-based scheduling scheme. It is evaluated by the BONeS network simulator. For simply illustration, the traffic is only classified into three classes:

- Constant Bit Rate (CBR) Traffic:

The classical source of CBR is voice. Such traffic is usually assigned as high priority because of its delay-constrained characteristic. The arrival pattern is emulated by Poisson arrival process in the simulation work.

- Variable Bit Rate(VBR) Traffic:

The data rate vibrates during the connection time. The classical source of VBR is compressed video. Similarly, it is usually assigned as high priority because of its delay-constrained characteristic. The arrival pattern is also emulated by Poisson arrival process.

- Available Bit Rate (ABR) Traffic:

Such traffic is served when no CBR or VBR packet is waiting. As a result, it is classed as low priority. The classic source of CBR traffic is data. The arrival pattern is also emulated by Poisson arrival process.

The traffic parameters of the three classes are listed in table 1. Table 2 lists the mean arrival rate of call-request with respect to the simulation time. This table also illustrates the variation of the system load. During the simulation period, the ABR traffic keeps the constant mean call arrival rate. The variation of system load is due to the CBR and VBR traffic. During the beginning 2000 seconds, the system is loaded with light CBR and VBR traffic. During 2000~3000 seconds, the CBR traffic grows up and exceeds the default bandwidth. Then, the VBR traffic becomes heavy and exceeds the default bandwidth during the 4000~5000 seconds. Finally, both CBR and VBR traffics are heavy. The simulation result is described according to the drive of time, and is divided into some sub-sections.

- Loaded with Heavy CBR

Initially, the traffic of the all classes is light, no call request will be rejected, and there is no need of bandwidth adjustment.

From the 2000 seconds on, the number of CBR call request increases and exceeds the default bandwidth. The result is shown in figure 3. Under the scheme of static bandwidth allocation, many call requests are rejected because of insufficient bandwidth. However, the real-time adjusting scheme can quickly react and reduce the blocking ratio. The periodic adjusting scheme can not react immediately with the suddenly increased arrival. There is slight blocking ratio when the default bandwidth is temperately insufficient. However, the bandwidth is adjusted after a certain period, and the blocking ratio is alleviated. As we can expect, the scheme with 100 second period need longer time to relieve the congestion.

- Loaded with heavy VBR

The result during 3000~5000 seconds is shown in figure 4. After the 3000-th second, the arrival rates of all the three classes return to the initial rate. However, the system remains the same bandwidth allocation. After the 4000-th second, the VBR customers increase quickly; while that of the CBR customers keep the same. The real-time scheme can adjust the bandwidth and avoids blocking. However, there is still blocking at about 4700-th second because no other class can support redundant bandwidth. The periodic scheme cannot react to the bandwidth requirement immediately. As a result, the system tends to congestion. Because the bandwidth of the VBR class has been moved to the CBR, the congestion condition occurs earlier than the static scheme. However, they can admit more customers after periodic adjustment. As for the effect of the period on the performance, long period can not effectively adjust the bandwidth to match the varying bandwidth requirement. The performance of the periodic scheme with 10 second period is better than that with 100 seconds.

- Loaded with heavy CBR and VBR

Between 5000~6000 seconds, the arrival rate of the VBR and CBR are a little higher than the normal condition. But, the system is not heavily loaded. Figure 5 and 6 show that the static scheme can satisfy the bandwidth requirement of all classes. Both the real-time and periodic scheme can also meet the requirement.

After the 6000-th second, both the VBR and CBR exceed the default bandwidth. As a result, there occurs some block. As expected, the scheme with smaller periodic can react more quickly and result in slighter blocking ratio.

From the view point of blocking ratio, the real-time adjusting is the best scheme, because it can try to adjust bandwidth before blocking a call request. However, the system is much more unstable and busy because it tries to adjust the bandwidth whenever a request is going to be rejected. The periodic scheme can relieve the processing time because it tries to adjust bandwidth only at certain period. However, it can not react to the instant variation of bandwidth requirement and relieve the blocking ration if the period is long. On the contrary, the similar drawbacks of the real-time scheme appear if the period is short.

IV. Concluding Remarks

In this paper, we propose the dynamic bandwidth allocation scheme to relieve blocking ratio. Two time point ‘real-time’ and ‘periodic,’ for adjusting bandwidth are discussed. The periodic scheme is evaluated with two periods, 10 and 100 seconds. The simulation results reveal that the real-time adjusting can effectively elevate the blocking ratio at the cost of processing load of adjusting. On the contrary, the periodic scheme can relieve the processing time. However, there is trade-off between the blocking ratio and processing time when determining the period.

Acknowledgement

The author would like to thank Yih-Jiunn Tsai for the simulation work, and Jung-Shyr Wu for the writing advisement. The author also wish to thank the National Science Council of Republic of China for financial support under grant NSC 91-2213-E-197-002.

Reference:

1. Braden, R., Clark, D. and S. Shenker, (1994), “Integrated Services in the Internet Architecture: an Overview,” RFC 1633.
2. Braden, R., Zhang, L., Berson, S., Herzog, S. and S. Jamin, (1997), “Resource Reservation Protocol (RSVP) Version 1 Functional Specification,” RFC 2205.
3. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and W. Weiss, (1998), “An Architecture for Differentiated Services,” RFC 2475.
4. Boyle, J., Cohen, R., Durham, D., Herzog, S., Rajan, D. and A. Sastry, (2000), “COPS Usage for RSVP", RFC 2749.
5. Seaman, M., Smith, A., Crawley, E. and J. Wroclawski (2000), “Integrated Service Mappings on IEEE 802 Networks,” RFC 2815.
6. E. Rosen, A. Viswanathan, R. Callon, (2001), “Multiprotocol Label Switching Architecture,” RFC3031.
7. Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, E. Felstaine, (2000), “A Framework for Integrated Services Operation over Diffserv Networks,” RFC2998.
8. G Mamais, M. Markaki, G. Politis, I. S. Venieris, (1999), “Efficient Buffer Management and scheduling in a combined IntServ and DiffServ architecture：a Performance Study,” ICATM '99, 2nd International Conference on ATM, pp. 236 –242.
9. J. C. R. Bennett and H.Zhang, (1996), “WF^{2}Q：Worst-case fair weighted fair queuing,” in Proc. IEEE INFCOM’ 96, San Francisco, CA, pp. 120-128.
10. S. Brim, B. Carpenter, F. Le Faucheur, (2000), “Per Hop Behavior Identification Codes, ” RFC2836.

91 年 09 月 16 日投稿
91 年 10 月 20 日接受

Table 1　Traffic parameters.

	Class 1	Class 2	Class 3
Data type	CBR	VBR	ABR
priority	High	High	Low
Packet size(bytes)	500	500	500
Default bandwidth(bps)	3M	3M	-------
Data rate (bps)	8k	80k　30% 64k　40% 48k　30%	-------
Duration time	100sec	100sec	-------
Segment size(bytes)	-------	--------	1.5M
Minimum data rate(bps)	-------	--------	1k
Reserved bandwidth(bps)	1M	1M	-------

Table 2　Mean arrival rates of call-request.

	CBR	VBR	ABR
0~1000 second	1.5	0.2	0.1
1000~2000 second	1.5	0.2	0.1
2000~3000 second	5	0.2	0.1
3000~4000 second	1.5	0.2	0.1
4000~5000 second	1.5	0.4	0.1
5000~6000 second	2.8	0.3	0.1
6000~7000 second	4	0.33	0.1

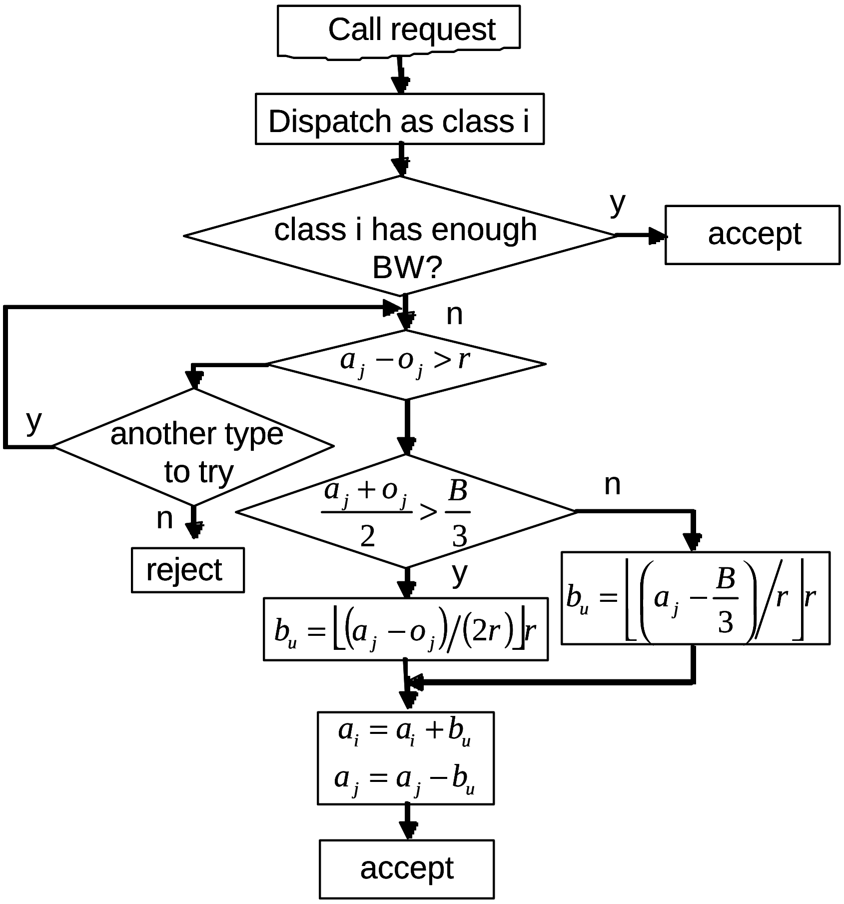

Fig 1　Dynamic bandwidth adjustment.

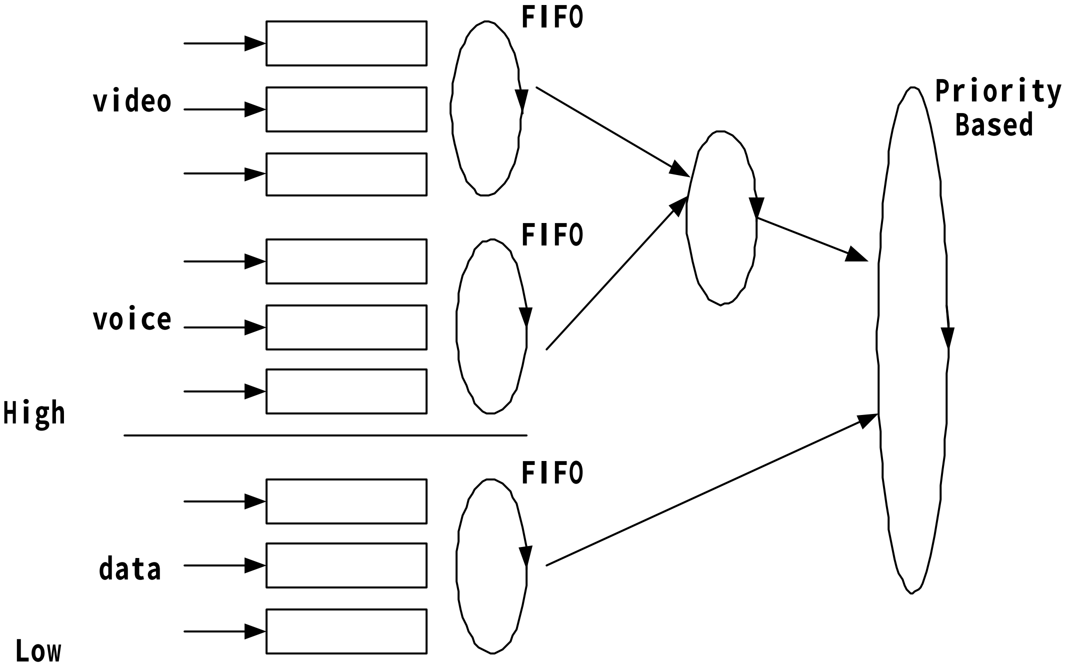

Fig 2　Class-based scheduling scheme.

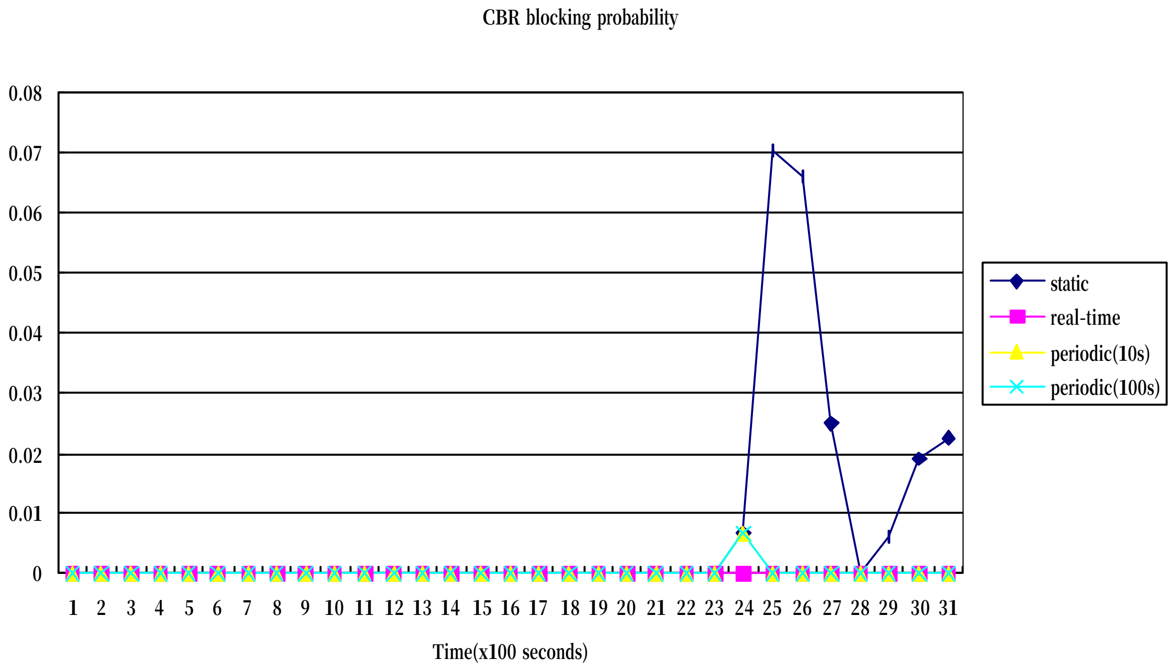

Fig 3 Blocking ratio of CBR before 3000 seconds.

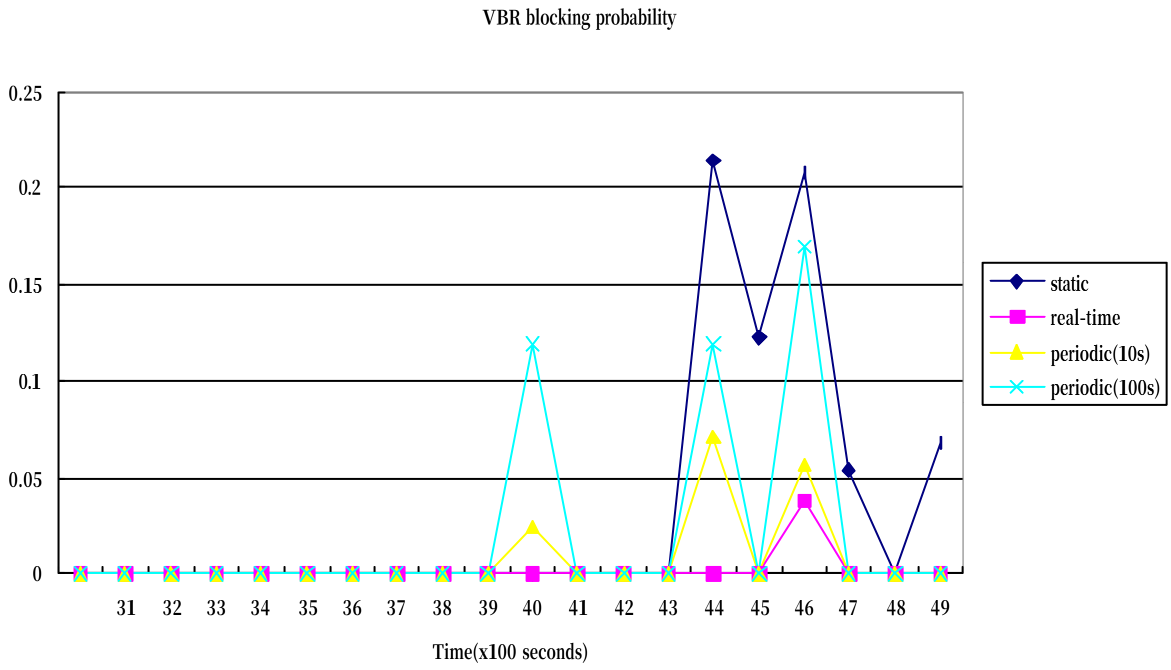

Fig 4 Blocking ratio of VBR after 3000 seconds.

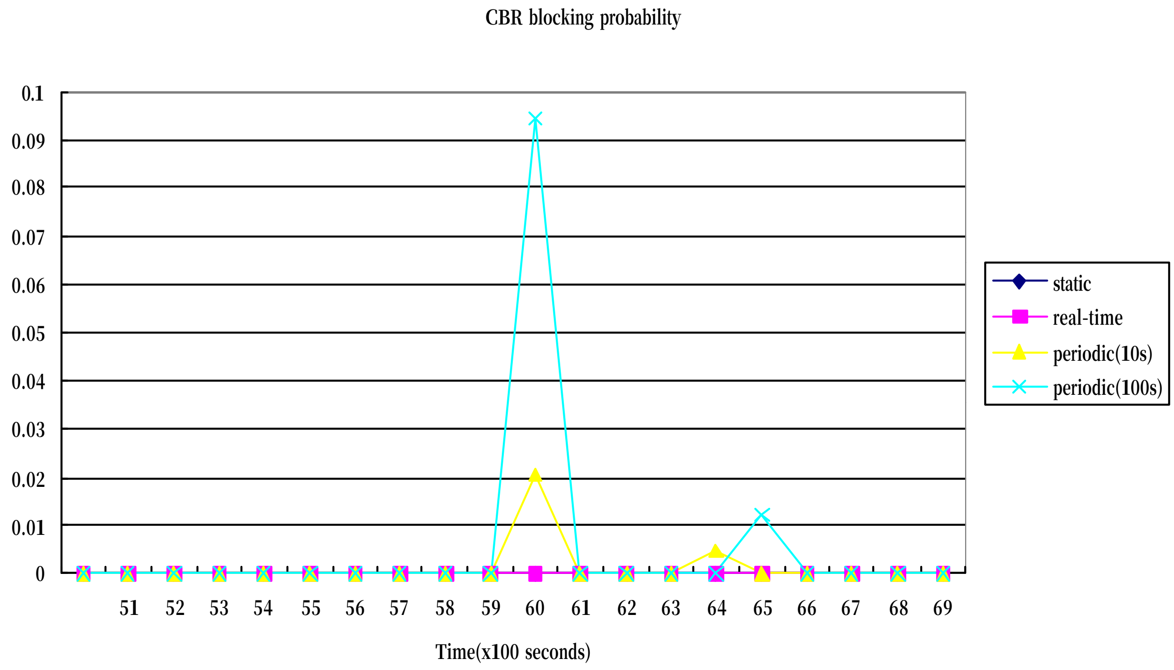

Fig 5 Blocking ratio of CBR after 5000 seconds.

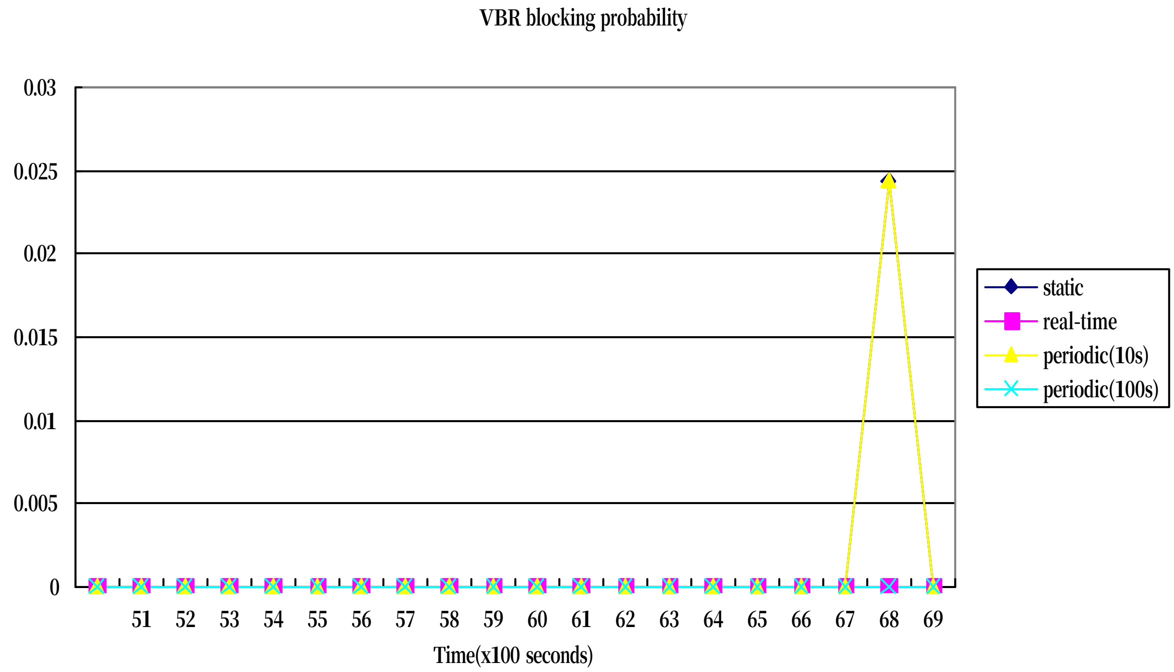

Fig 6 Blocking ratio of VBR after 5000 seconds.