采用 RBF 来支撑互联网络上的电路模拟服务

金　淬　张　斌　赵　阳　王庆波　陈　滢　(IBM 中国研究院 北京 100193)

Using RBF to Enable Circuit Emulation Service over Internet

Xing Jin, Bin Zhang, Yang Zhao, Qingbo Wang, Ying Chen (IBM China Research Laboratory, Beijing 100193)

Abstract: Circuit Emulation Service (CES) aims to enable packet switched networks to provide guaranteed services with comparable qualities of circuit switched networks. Our paper addresses the key issue of QoS of CES flows over Internet. Enlightened by the time division idea popularly used in circuit switched networks, we propose a time division based control mechanism to provide guaranteed QoS for the constant-rate CES flows. The control mechanism is able to estimate the arrival times of the coming packets in CES flows, and reserve the time slots for them. Accordingly, it enables the packets to consume the reserved time slots of their own, so the CES flows are guaranteed to be processed. Refreshing Bloom Filter (RBF), an efficient data representation structure, is proposed to support the time division control mechanism. It consists of multiple bloom filters, and can efficiently record the arrival time slots of millions of packets. The proposed control system model could be a practical tool to support Circuit Emulation Services over Internet.

Key words: refreshing bloom filter; circuit emulation service

1 Introduction

Circuit-switched networks were initially designed and deployed to provide voice communication services. With the boom of Internet and advanced IP technology, telecommunication is transformed to NGN which is packet based. It will bring a lot of benefits to bridge and converge circuit switched network services and packet switched networks. Unfortunately, current packet switched networks just provide best-effort service that one connection has to compete with other connections for network resources. It makes the packet networks obtuse to provide reliable quality of service to support real-time communication applications. Circuit Emulation Service (CES) aims to upgrade existing best-effort packet switched networks to provide the services with the good properties of circuit switched networks, which have guaranteed transport quality and end-to-end time delay.

Recently, much effort[1-4] has been made to enable packet-switched network to emulate circuitswitched network services, e.g. the widely used VoIP applications. Fig.1 illustrates the constitution of the interwork scenarios of CES. The CES IWF (Interworking Functions)equipments translate the circuit-switched frames into IP packets, which are suitable to transport over IP networks. Those IP packets are mostly transmitted at constant rates, and every two consecutive packets have equal time intervals. In this paper, we name those flows as CES flows. Many devices and software, such as media servers and VoIP phones, also produce constant rate CES flows.

Aweya's work[2,3] discussed several important issues on how to design CES end equipments to make clock synchronization and bounded time delay over Ethernet. They limited their works on Ethernet without considering the network conditions between the CES end equipments over Internet, and they assume that Ethernet is capable to offer sufficient bandwidth for CES flows. Noro researched the problems of CES over IP networks, and tried to construct a circuit emulation layer on top of the internet protocol stack[4]. Their works are also confined to handle variable packet delay at CES end equipments and rarely considered the problem of variable network conditions over Internet, where the best effort traffic control policies make it more challenge to guarantee the service quality of CES flows.

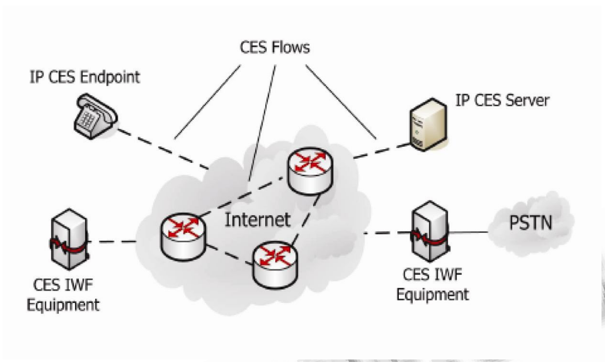

Fig.1 The general framework of CES over Internet

Our work addresses how to guarantee the quality of CES flows over Internet. The most popular related works are resource reservation approaches[5,6]. They have similar ideas to construct quality of service guaranteed flows traversing the network. Those algorithms try to allocate suitable bandwidth for served flows. They adjust the network occupancy of flows and offer unblocked traffic for the served flows. Early approaches just offer sufficient but fixed bandwidth for each flow, regardless of actual bandwidth usage. Later, the adaptive ideas[7,8] is introduced by measuring the traffic conditions(e.g. the rates of flows) to dynamically adjust the bandwidth offered for each flow. These approaches need to monitor the rates of every flow and modify each flow's available bandwidth, which would suffer from unaffordable storage requirements caused by numerous network flows.

We learned that those CES flows have the constrate feature and the arrival times of consecutive packets, named arrival pattern here, exhibits arithmetic pattern. This character makes it possible to follow the time division idea in circuit switched network to implement a similar control mechanism in IP networks. Our control mechanism predicts the time slots of the future packets, and reserves the time slots for them. Different from conventional approaches, it reserves the time slots rather than network bandwidth to guarantee the quality of service. To support this mechanism and make the storage efficient, we invent a novel data structure named Refreshing Bloom Filter (RBF), which can concisely represent the arrival packets belonging to the specific time slots, to constitute the key elements of our system.

Our contributions are three folders:

• We observe the const-rate feature of CES flows and introduce the concept arrival pattern.

• We propose a time division based control mechanism that can perform accurate flow control at packet level.

• We invent a randomized date structure, RBF, to efficiently represent a temporally refreshing multi-set.

This paper is structured as follows. Section 2 describes our idea on the admission control system in detail. Section 3 introduces the data structure of RBF. The system modal is presented in Section 4. Section 5 evaluates the performance of RBF. The conclusion is given in Section 6.

2 From the Idea of Time Division

In circuit switched networks, time division multiplexing is widely used to mix and transport multiple communication channels in a single link. The link is divided into separate time slots, each of them for a channel. Although all the channels share the bandwidth, the quality of each channel can still be guaranteed. This good feature is achieved by accurately allocating time slots for each channel and each channel monopolizes the link in its time slot.

It is found that the CES flows have the const-rate arrival pattern. This helpful feature make it possible to control the CES flows with a time division based control mechanism, which divides the time line into several consecutive time slots as well as the time division multiplexing technology in circuit switched network. This control idea can also be generalized and applied on flows with other arrival patterns.

In following paragraphs, we first explain the concept of arrival pattern, and then describe the idea how to apply time division mechanism to CES flow control.

2.1 Exploiting arrival pattern of flow

The serial of arrival times of packets in a flow can be recorded as:

$$P = \{t_0, t_1, ..., t_k, ..., t_n\} \quad (1)$$

where t_k is the arrival time of k-th packet. Arrival pattern represents the character of such a time array. There are two popular types of arrival patterns: const-rate pattern and exponential pattern.

Const-rate arrival pattern is an arithmetic sequence of arrival times:

$$P = \{t_0, t_0 + T, t_0 + 2T, ..., t_0 + nT\} \quad (2)$$

Const-rate patterns are usually found with CES flows.CES IWF equipments generally translate the frames in circuit switched networks into constant-rate packets, e.g., some RTP voice sources send out one packet in every 20ms period. Looking inside the CES flows, we find that the arrival times are with equal intervals and only one packet arrives in each time slot.

Another predictable type of arrival pattern is exponential pattern, with the time intervals following the exponential law:

$$P = \{t_0, t_0 + T, t_0 + 2^1 T, t_0 + 2^2 T, ..., t_0 + 2^n T\} \quad (3)$$

The flows with exponential patterns are commonly used for retransmission technologies[9].

Time division multiplexing technology essentially divides the time line into a serial of consecutive slots. With the similar idea, it is also possible to divide the time line into proper slots according to the arrival pattern of flow, and future packets will arrive in predicted time slots. It gives the chance to perform the time division based control.

2.2 Time division idea for exponential-rate flow

Jamjoom proposed an admission control algorithm to handle the busty request and keep out the useless retransmissions[9], which have the exponential patterns. This approach could be considered with the time division idea.

His main idea is to study the exponential arrival pattern and divide the time line into proper time slots. Abacus Filter[10] is used to count the reserved requests of every time slot. When a request arrives, if it can not be accepted immediately, the system will predict the time slot which its retransmissions will arrive in. If the server has no capacity left to serve the retransmitted request in the time slot, the request will be rejected directly to avoid any more retransmission retries.

2.3 Time division idea for const-rate flow

With the knowledge of arrival pattern of a constrate flow, the system can estimate the average arrival time interval and predict the arrival time slots of coming packets. The time division-based control mechanism divides the time line into consecutive time slots, and ensures the packets whose arrival in the predicted time slots is to be processed. Following the fashion of resource reservation, the predicted time slots can be considered as reserved time slots for coming packets. The mechanism could be the more natural emulation of circuit-switched services.

Firstly, the system estimates the average arrival interval length from history recorded arrival patterns. The average interval may also be informed by the service quality request or by other ways. Secondly, when a packet arrives, the arrival time slots of next coming packet is predicted and reserved. Finally, if the coming packet comes in the reserved time slot, the packet will be processed with high priority. Otherwise, it will be assigned with lower priority.

3 Time Division Idea for Const-Rate Flow

To enable efficient management of packets and their time slots, we invent a novel structure Refreshing Bloom Filter (RBF) to represent the serial of refreshing time slots. RBF consists of multiple Hash Function Groups (HFG) and multiple Standard Bloom Filters(SBF). SBF, which can concisely represents a set to support approximate membership query, was first used for word processing in 1970[11], and later widely adopted in large-scale internet applications. A throughout survey of Network Applica- tions of Bloom Filters was presented in Ref.[12]. In this section, we will introduce the evolution from SBF to RBF.

3.1 Standard bloom filter (SBF)

SBF leverages a hash function group (HFG), including k independent hash functions, to hash all elements into a bit array. An element can be judged as a member of the set, only if the k hash values of this element are all flagged. In short, SBF novelly identifies the membership of an element by a k-dimension flag vector. The elaborate description can be found in Ref. [12] as below:

A set S = $\{x_1, x_2, \cdots, x_n\}$ of n elements can be represented by a SBF, which is described by an array of m bits, initially all set to 0. The i-th bit is denoted as SBF (i). A hash function group HFG consists of k independent hash functions $\{h_1, \cdots, h_k\}$. When inserting an element x into a set, the corresponding bits hashed from x are set to 1:

$$SBF(h(x)) = 1, \qquad h \in HFG \tag{4}$$

A bit may be set to 1 several times, but only the first change has an effect. A given item y can be checked by (5).

$$\begin{cases} y \in S & if: SBF(h(y)) = 1, \quad \forall h \in HFG \\ y \notin S & if: SBF(h(y)) = 0, \quad \exists h \in HFG \end{cases} \tag{5}$$

It is unavoidable that a SBF may yield a false positive, where it suggests that an element y is in S even though it is not.

3.2 Multi-set bloom filters (MBF)

To support the membership query of multiple sets, several approaches are proposed in this subsection. First, multiple SBFs were used to represent multiple sets. Intuitively, each SBF is for an independent set. Second, with the same idea of tradeoff between storage efficiency and computation cost, multiple HFGs approach is introduced to represent multiple sets. Third, multi-SBF and multi－HFG are combined in a cross way to form a matrix like structure called M^2BF.

In following subsections, the total number of sets is l, and the i-th set is Si. The number of SBFs is u, and the i-th SBF is denoted as SBFi. The value of m-th bit of SBFi is SBFi(m). The Number of HFGs is v, and HFGi denotes the i-th HFG. The hash function instance is denoted as h.

1) *Multiple SBFs Approach (MSBF)*: Multiple Standard Bloom Filters (MSBF) approach is the most straightforward way to handle the multi-set membership representation. One HFG and v SBFs are used, where v = l. i-th set is represented by the HFG and SBFi. To insert an element x in to Si, the hashed bits in SBFi are set to 1. A new element y is in Sj only if all the hashed bits of SBFj equals to 1. MSBF makes use of l SBFs to deal with the l sets query. Under the same accuracy of query in single set, the storage cost is l times of that of SBF.

2) *Multiple HFGs Approach (MHBF)*: In this approach, unlike MSBF, multiple HFGs are associated with one SBF to represent multi-set. The data structure is called MHBF. Kumar proposed a similar data structure called SCBF for Per-Flow traffic measurement [13]. During inserting an element x into Si, the hashed bits with HFGi are all set to 1. A new element y is in Sj only if all the bits hashed from HFGj equals 1. If MHBF has same storage efficiency as SBF, the false positive obviously increases, since the membership flags of all sets are located in a single bit array.

3) *Multiple SBFs and Multiple HFGs Approach (MMBF)*: Compared with MSBF, MHBF achieves lower false positive rate but brings higher computation load. In order to balance the two effects, we propose a so called MMBF data structure, which is combined by multiple SBFs and HFGs (Fig.2). It needs a little more storage but reduces false positive rate and computational complexity effectively.

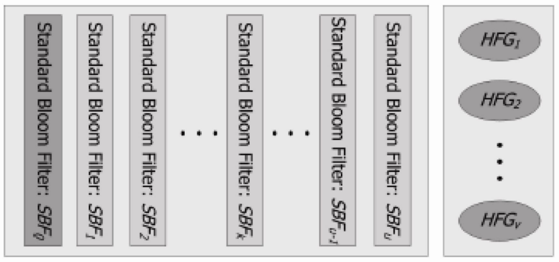
Standard Bloom Filter: SBF0
Standard Bloom Filter: SBF1
Standard Bloom Filter: SBF2
Standard Bloom Filter: SBFk
Standard Bloom Filter: SBFu-1
Standard Bloom Filter: SBFu
HFG1
HFG2
HFGv

Fig.2　A u×v MMBF consists of v HFGs and u SBFs, from SBF_1 to SBF_u. RBF uses u + 1 SBFs to represent u × v refreshing sets.

Given l = u×v sets to be handled, an array of u SBFs and an array of v HFGs are setup. Each set can be represented by a SBF and a HFG. The i-th set can be expressed by SBFp and HFGq, where, like the matrix pattern, the couple (p,q) satisfies (6):

$$i = q + (p-1) \times v \tag{6}$$

Given an element x to be inserted to Si, the bits are all set to 1:

$$SBF_p(h(x)) = 1, \quad h \in HFG_q \tag{7}$$

Given an element y, is true, only if both (8) and (9) are satisfied:

$$B_{p*}(h(y)) = 1, \quad \forall h \in HFG^{q^*} \tag{8}$$

$$j = q^* + (p^* - 1) \times v \tag{9}$$

The evolution from SBF to MBF can be considered from another perspective. MSBF occupies l bit arrays for l sets. MHBF uses l HFGs to describe multiple sets. They can be imagined as 1-Dimension ways: a set is coordina- ted by one indicator (a SBF or a HFG). While MMBF extends the idea and characterizes the multiple sets by a 2-Dimension way: using an array of SBFs and an array of HFGs. Each set must be represented by a HFG-SBF couple(a HFG and a SBF).Table 1 compares SBF with MBF.

Table 1　Comparison between SBF and MBF

	Single HFG	*Multiple HFGs*
Single SBF	*SBF*	*MHBF*
Multiple SBF	*MSBF*	*MMBF*

3.3 Refreshing bloom filter (RBF)

In RBF, each time slot can be treated as a set with an expire time, denoted as $S_i(t)$. When t attenuates to 0, the corresponding time slot S_i will be eliminated, and a new set with maximum expire time will be generated. The RBF keeps constant amount of time slots all the time. This character of the time slots is called “Refreshing”. An element is attached with an expire time, denoted as (x,t_k). Two operations are necessary: insertion and query. An RBF is constructed to support the two operations for the refreshing sets. RBF is the extension of MMBF with an additional SBF (SBF_0), which is used to store the new time slots temporarily. Fig.2 illustrates the basic data structure.

In the RBF structure, a SBF represents v sets, and most of bits in the SBF are occupied by the other v - 1 sets instead of one set. When a slot is being eliminated during refreshing, we just invalidate the corresponding SBF-HFG couple, but not erase the bits of corresponding SBF. If we simply clear the SBF, the rest living slots will be damaged. Only if v slots in the same SBF expire, this SBF can be cleaned up. At the same time, a fresh slot is generated, which will be located in the additional SBF.

When an element (x,t_k) is being inserted, first we find out the slot of the same expire time as tk, then flag the bits hashed from x.

When an element x is being queried, the query result is $(x,t(x,1),t(x,2),...,t(x,k))$, where $t(x,1),...,t(x,k)$ are the time slots in which the hashed values from x are all true.

4　Design of CES Flow Control System

RBF offers the natural ability to divide the time line into multiple time slots and record the elements belonging to each slot. CES flows, the packets of which arrive in predictable time slots, can better fit for the time division control mechanism. The policy of our traffic control mechanism is: a part of network bandwidth is reserved for CES flows following accurate time division control, and the rest bandwidth is left for other flows under best-effort control (Fig.3).

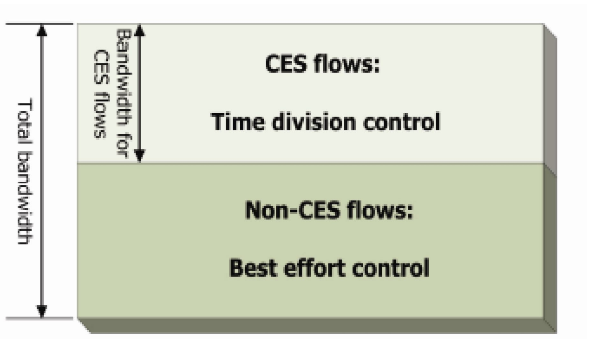

Fig.3　Our mechanism concentrates on the control of CES flows

As mentioned in Section 1, some traditional control approaches can not be widely deployed because of their storage and computation cost. Our approach tries to overcome these problems by two efforts. On one hand, our controls concentrate on CES flows, which are just a fraction of the whole network traffic flows. The computation scale is affordable. On the other hand, our control system utilizes the storage efficient structure RBF. It only costs a little more storage even though millions of more flows arrive. Those two features effectively make our system applicable. Fig.4 depicts the system design.

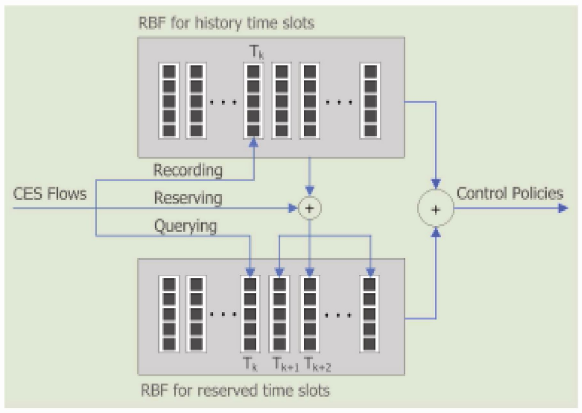

Fig. 4　The admission control system model

The control mechanism includes three processes: recording history context, reserving time slots, and generating control policies. The following subsections will describe them separately.

4.1 Recording history context

An RBF (denoted as RBF_h) is utilized to manage the history context in a time window, which is divided into u × v time slots. The time window is shifting and the time slots are refreshing: the oldest time slot should be removed when the time window shifts. In our design, the lengths of these time slots are selected according to the application configuration.

When a packet arrives, RBF_h records the arrival time of the packet. The term "RBF records the arrival time of a packet" means to insert the pair(x,t_k)into the RBF, where x is the flow indicator of the packet, and usually selected as the combination:{src_address,src_port,det_address,det_port}.t_k is the time slot which the arrival time of this packet belongs to.

4.2 Reserving time slots

When a packet comes, the system does not only record the history context, but also reserves the time slots for the coming packet. Reserving time slots consists of two operations. First, it predicts the possible arrival time slots of the next coming packets according to the arrival time of current packet and the estimated arrival interval from history packets. Second, the predicted time slots are recorded into the RBF (denoted as RBFr), which is employed to manage the reserved time slots.

The first operation is to predict the arrival time of next packet. It is necessary to know the average arrival time interval (denoted as t_v) as well as the arrival time of current packet (denoted as t_i). t_v can be either given by the application which requests CES flow guarantee or calculated from the history arrival pattern which has been recorded in RBF_h. Then the time period to be reserved is:

$$\hat{T}_{i+1} = [t_i + t_v - \delta, t_i + t_v + \delta] \qquad (10)$$

where, δ is the tolerance of predicted arrival time, and it can depend on the variance of the packet arrival times. This period may consist of several consecutive time slots. All of them should be recorded in RBFr.

4.3 Generating control policies

At the same time, when the time slots for next coming packets are reserved, the system queries the rese- rved time slots of current coming packets in RBF_r. Given the indicator x of current packet,the query result is a set of reserved slots: $\{T(x,1),T(x,2),...,T(x,k)\}$. If $\exists k, t_i \in T(x,k)$, we say that the arrival time of current coming packet belongs to the reserved time slots.

We can obtain three state variables $S = \{s_f, s_p, s_r\}$ of current processing packet. sf denotes if the packet is in the served CES flows. sp indicates if the actual arrival time of the current packet belongs to the reserved time slots. sr represents if the reservation for future packets successes. If the served flows cost most resource and no more packets can be reserved, the reservation will probably fail. Those three variables would provide most information to decide the policies for current processing packet.

We have synthesized some possible control policies (Tab.2) and they may be not adapted enough. In fact, the policies may be different and customized under varieties of circumstances. In our future work, we would do further researches on designing more practical policies following the time division idea.

Table 2　Some possible control policies

Conditions	Control Policies
s_f is false	Process it as non-CES flows
s_f is true, s_p is true	Process it as CES flows, let it high priority.
s_f is true, s_p is false	Process it as CES flows, let it low priority.
s_f is true, s_r is false	Set s_f to false, and process it as non-CES flows, let it high priority

5　Performance Evaluation

In this section, we will analyze and compare the computational complexity, storage efficiency, and false positives of different RBF implementations.

5.1 Computational complexity

To query or record an element x in a set Si, it is just needed to compute hash operations of the HFG associated with Si, then get or set the values of corresponding bits in the bit array. Both querying and recording are implemented based on hushing operations. So the calculation of hash functions contributes the most computation of the system.

It is assumed that the numbers of hash functions in HFGs are the same, and all the hash function instances have the same computation cost. So the computation complexity of each HFG can be treated as equal, denoted as C. For MHBF with l HFGs. A query operation needs the complexity of l × C. For MMBF structure with u Bloom Filters and v groups of hash functions, the complexity of query operation is v × C. We can draw the conclusion that the computational complexity only depends on the number of HFGs (Fig.5). The number of SBF only affects the memory cost and false positive rate.

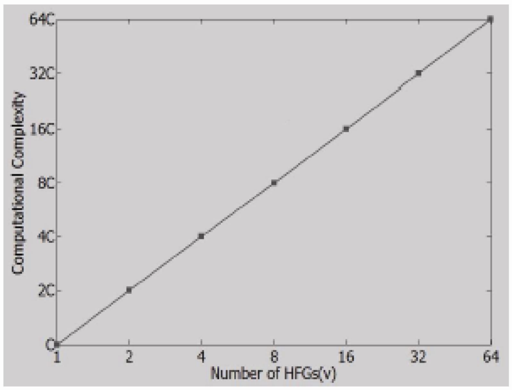

Fig.5　The computation complexity is proportional to the number of HFGs

5.2 Storage Efficiency

As mentioned in section 3.3, the RBF includes u + 1 SBFs for l = uxv time slots. By MMBF approach, u + 1 SBFs and v HFGs can represent (u + 1) × v = l + v sets. It means that the storage of v sets are waste in RBF. The storage waste rate w is

$$w = \begin{cases} \dfrac{1}{u+1}, & (v \geq 2) \\ 0, & (v = 1) \end{cases} \tag{11}$$

If a SBF only represents one set, the backup SBF will be useless and removed from RBF. So the waste rate is 0 when v = 1. Fig.6 shows the waste rate of each RBF with different SBF number.

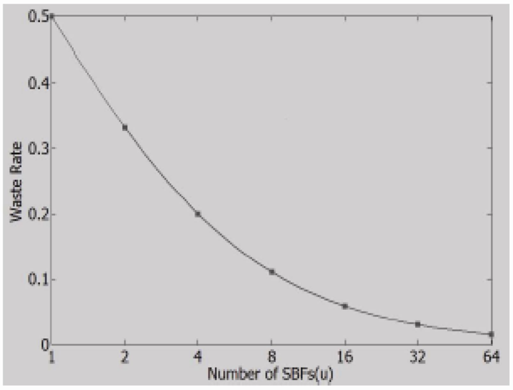

Fig.6　Waste rate of RBF (v≥1). More SBFs (less sets a SBF represents), less waste storage

5.3 False positive

Bloom Filter approaches unavoidably bring some false positives. If we assume that all elements are perfectly mapped to uniform random numbers, and don't consider the effects caused by waste SBFs, the theoretical false positive of a set is:

$$\phi_S = (1-(1-\frac{1}{m})^{v\times k\times n})^k \tag{12}$$

where n is the number of elements each set has, m is the length of bit array of SBF. In the following simulation, m changes following u. Each HFG consists of k hash functions and each SBF is associated with v HFGs. The joint false positive of multi-set query is

$$\phi_{RBF} = (1-\phi_S)^{u\times v} \tag{13}$$

In actual network, the number of coming packets changes dynamically. Here, we would like to compare the false positives by simulation. The dynamic behavior of current network traffic can be modeled by a self-similar traffic model called Multifractal Wavelet Model(MWM)[14]. It generates random variable with the desired distribution, which can describe the arrival time of new elements. The MWM-based approach is shown to have better flexibility and accuracy in modeling a wide range of real network traffic conditions, especially the burst traffic.

In our simulation, we present the conditions as follows: the number of time slots is 64, and each HFG consists of 4 hash functions. The total storage for constructing RBF is of constant size of 1024×48×256 =12M bits. That memory is fully occupied all the time: u×m=12M, so if the number of SBFs increases, the length of each SBF bit array will decrease. MWM model we used has two important parameters: mean of coarse-scale scaling coefficients and the standard deviation of coarse-scale scaling coefficients (denoted as stdc). In the simulation, the first parameter is set to 0.7 and the second parameter, the stability of the random variable, is increaseing from 0.1 to 0.65.About 217 packet arrival times will be generated under these conditions. Thus the RBF false positives are compared under dynamic circumstances. In order to exclude the affects caused by uncertainty, we repeated the simulations with each stdc value 20 times, and took the mean value as the final result.

Fig.7 shows the results of simulations. Each curve represents the accuracies of different u values. The figure indicates that using less SBFs (larger size of each SBF) is more robust under dynamic circumstances, especially with strong bursts.

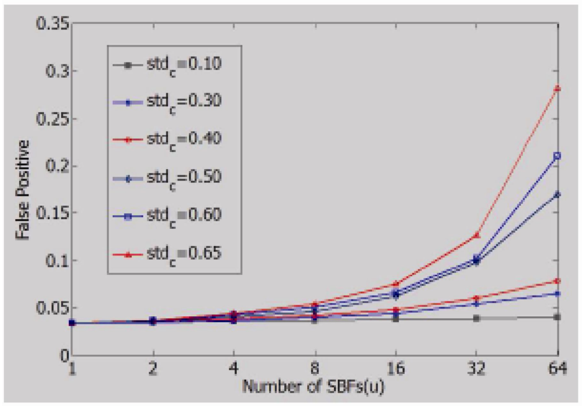

Fig.7 Comparison of false positives of RBF under different network situations.

6 Conclusions and Perspectives of RBF

In this paper, we studied the arrival pattern of CES flows, and proposed the time division based control mechanism inspired by time division multiplexing in circuit switched network. This new control mechanism instinctively enables Circuit Emulation Service over Internet. Moreover, to make the control system more efficient, we invented the Refreshing Bloom Filter (RBF) structure which can support efficient recording and querying of multi-set membership even with a large amount of elements. Our analysis and simulation results also indicate that RBF is capable to achieve good performance under dynamic circumstances.

RBF plays as the backbone of our control system. As the membership recording and querying operations are the the basic requirements for some real time decision making systems, we are now exploiting the new applied fields for RBF. These fields include traffic measurement and accounting[15], security systems[16], cache technology[17], and etc.

References

1 Aweya J. Trunking of TDM and narrowband services over IP networks. International Journal of Network Management, 2003,13:33－60.

2 Aweya J. Circuit emulation services over Ethernet-Part1: Clock syschronization using timestamps. International Journal of Network Management, 2004, 14:29－44.

3 Aweya J. Circuit emulation services over Ethernet-Part 2: Prototype and experimental results. International Journal of Network Management, 2004,14:45－58.

4 Noro R, Hamdi M, Hubaux JP, Circuit Emulation over IP Networks. Technical Report, Siss Federal Institute of Techonology Lausanne, February, 1999.

5 Zhang L, VirtualClock: A New Traffic Control Algorithm for lPacket-Switched Networks. ACM Transactions on Computer Systems, 1991,9(2):101－124.

6 Zhang L, Deering S, Estrin D, Shenker S, Zappala D, RSVP: A new resource reservation protocol. IEEE Network.

7 Grossglauser M, Tse D. A Framework for Robust Measurement-Based Admission Control. ACM Proc.

of SIGCOMM, Cannes, France, 1997.

8 Jamin S, Danzig PB, Shenker S, Zhang L. Measurement-based Admission Control Algorithm for Integrated Services Packet Networks. ACM Proc. of SIG COMM, Cambridge, MA, 1995.

9 Jamjoom H, Shin KG. Persistent Dropping: An Efficient Control of Traffic Aggregates. Proc. of SIGCOMM, Karlsruhe, Germany, August 2003.

10 Jamjoom H, Pillai P, Shin KG. Re-synchronization and Controllability of Bursty Service Requests. IEEE/ ACM Transactions on Networking, August 2004.

11 Bloom B. Space/time tradeoffs in hash coding with allowable errors. CACM, 1970,13(7):422－426.

12 Broder A, Mitzenmacher M. Network Applications of Bloom Filters: A Survey. Proc. of 40th Allerton Conference on Communication, Control, and Computing, Urbana, IL, October 2002.

13 Abhishek Kumar, Jun (Jim) Xu, Li (Erran) Li, Jia Wang, Oliver Spatschek. Space-Code Bloom Filter for Efficient Per-Flow Traffic Measurement. Proc. of IEEE Infocom, Hong Kong. March 2004.

14 Riedi RH, Crouse MS, Ribeiro VJ, Baraniuk RG. A Multifractal Wavelet Model with Applications to Network Traffic. IEEE Trans. on Information Theory, 1999,45(4):992－1018.

15 Estan C. Varghese G, New directions in traffic measurement and accounting.Proc.of the 2001 ACM SIG COMM Internet Measurement Workshop, 2001,11:75－80.

16 Aguilera MK, Ji M, Lillibridge M, MacCormick J, Oertli E, Andersen D, Mann MBT, Thekkath CA. Blocklevel security for network-attached disks. Proc. of the Conference on File and Storage Technologies, USENIX Association, San Franci- sco, CA, 2003:159－174.

17 Fan L, Cao P, Almeida J, Broder AZ. Summary cache:a scalable widearea Web cache sharing protocol. IEEE/ACM Transactions on Networking, 8(3):281－293.