Front. Optoelectron. China 2010, 3(2): 198–204
DOI 10.1007/s12200-010-0010-3

RESEARCH ARTICLE

Characteristics analysis of parasitic storage

Jincai CHEN (✉), Jiang ZHOU, Gongye ZHOU

College of Computer Science and Technology, Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology,
Wuhan 430074, China

Abstract It has been approved that the message checking services of Internet protocols possess the capacity of parasitic computing for some non-deterministic polynomial (NP)-complete problems. Inspired by this thought, we wonder how to dig storage capacity over the Internet by means of the message processing mechanism of protocols without extra cost. As data packets may travel over the network for a period of time and return to the source site circularly, a large capacity and dynamic parasitic storage could be created by establishing the mechanism to guarantee a certain amount of data packets being sent and received repeatedly among network nodes. However, parasitic storage may affect the network to some extent and vary itself with the fluctuation of the network. To analyze the interaction between them, an analysis model has been constructed. With this model, we have explored the parasitic storage impacts on network and the characteristics of parasitic storage over complex network in the sense of probability.

Keywords parasitic storage, complex network, Internet, performance evaluation

1 Introduction

The Internet is a complex network spread with huge dynamic computing and communication resources [1]. Similar to the ever-changing colorful biological world, various computing and communication nodes on the Internet seem like a large amount of organisms that deposit "nutrition" for the "parasitic survival" of others. As we know, communication protocol is the base of network operation and data packets transfer among the network nodes according to certain protocols. By analyzing the processing mechanism of communication protocols, we can dig out the data return transferring capacity and hence implement the dynamic online data caching and acquiring. Owing to the extensive distribution and increasing spread of Internet resources, the Internet owns great power of storage, and thus a new kind of free-of-charge networking storage can be achieved [2].

With the development of information technology and the popularization of the computer system, data becomes more important in personal information applications. The Moore's law of storage indicates that the newly added storage capacity during every 18 months is equal to the summation of information that occurred before. With the rapid inflation of information, people cannot afford the additional storage facility in the traditional storage mode. As a complex system with various kinds of protocols, the Internet has huge potential resources to be used. For example, Barabasi et al. [3] illustrated that the existing message checking services of Internet protocols can be used in the computation of non-deterministic polynomial (NP)-complete problems, and a new computing mode named "parasitic computing" has been put forward. Inspired by the parasitic computing mechanism, it can be seen that when handling the communication, the time of message transmission is nonzero, and there is time gap between the source and destination nodes. Therefore, the information can be maintained on network by circularly sending and receiving data packets. It is just like a ball-tossing acrobatic performance: the ball is the package, and the toss height is distance. Therefore, when the data packets travel on the network for a period of time and finally return to the source, it indeed establishes a mechanism to guarantee a certain amount of data to be stored on the network, which is called parasitic storage [4].

Like many biological organisms, a computer may be taken as a parasitic entity for others. As with organisms, the computer under such circumstance may benefit or be harmed from the parasitical relationship in some way. Parasitic organisms can thrive best when all the organisms meet certain conditions. Similarly, parasitic storage mainly exploits the potential network resources by making use of a certain property of existing communication protocols. It

---

Received July 16, 2009; accepted September 17, 2009

E-mail: jcchen@hust.edu.cn

can send data packets to remote hosts (i.e., parasitifers) and receive the echoed one circularly. Once a data packet is pushed out for a period of time, it will be sustained on the network accordingly and thus it is possible to put some data information on the network until it is echoed back. Hence, we know that the Internet has a momentary data storage capacity. By establishing a mechanism for cyclic sending and receiving data among a lot of available hosts, a certain amount of data can be maintained on the network and thus data storage is actually achieved. This is applicable for large space, short-term and free of charge data storage, without being sensed by hosts. It does so without threatening the security of the hosts [5] because it only makes use of the idle resources of the network.

Parasitic storage system can be implemented on the basis of the existing network protocols, such as Internet control message protocol (ICMP), simple mail transfer protocol (SMTP) [6]. Considering the case where ICMP echo (PING) messages are used as the basis of storage, the storage capacity and latency bounds are characterized in terms of payload of the messages, error rate, the round-trip time, etc. It uses standard Internet protocols to store information. Essentially, the storage capability is contributed by the time delay taken for the data information being sent and reflected back to the original node. One communication channel of parasitic storage involves a local node, a host, and a path comprising of some sequence of systems and links that connect the local node and hosts. Utilizing the mechanism of requesting and echoing packages, the network becomes storage media in a certain sense. It stores data by distributing information over multiple hosts. However, owing to the Internet is a complex network based on multiple protocols, parasitic storage would be disturbed by various factors. On the other hand, parasitic storage will also affect the network performance to a certain extent. In this paper, a special queuing model oriented to parasitic storage was constructed. Based on the model, the impact of parasitic storage on network was discussed, and the dynamic characteristics of parasitic storage on complex network were analyzed in detail.

2 Queuing model of parasitic storage system

Queuing theory is mainly seen as a branch of applied probability theory [7]. It can be used in different fields, e.g., communication networks, computer systems, machine plants, and so forth. The main purpose of utilizing queuing theory is to analyze and optimize various services systems.

In the parasitic storage system, how to arrange the available network resources and distribute data over them is an important problem for getting better storage performance. We can describe parasitic storage system from the perspective of a queuing system [8]. A parasitic storage link involves a local node, some remote parasitifers, and the attached data queues. The sending and receiving packages can be seen as customers and the parasitifers as servers. All these customers are subject to the first-come, first-served (FIFO) service policy. The queue attributes oriented to parasitic storage are as follows.

2.1 Packets arrival distribution

The packets arrival complies with the following properties:

1) Stationary. In parasitic storage, the packets are sent to remote parasitifers circularly, and the time interval only depends on the model mechanism and the network status. Once the mechanism is established, the number of packets sent to parasitifers is only related to the duration but not the starting point.

2) Independence. When the rate of sending packages gets stable, the arrival numbers of packets that appear in two different periods of time are independent with each other.

3) Uniqueness. As mentioned above, on computer networks, the time of message round trips between local and remote parasitifer is nonzero; there will be no more than two arriving data packets during a very short time.

From above, we can assume that the packages arrival to a parasitifer is a Possion stream with parameter υ in parasitic storage.

2.2 Packets service time distribution

In parasitic storage, when the initiator (local node) is sending the packets to the parasitifer and waiting for a reply, there exists a round-trip delay time (RTT) between them and it can be seen as the service time in the queuing model. To analyze the RTT property, from an Internet protocol (IP) node 202.114.29.217 (located at the Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan, China), we have completed a network test within a certain IP space (222.20.192.0–222.20.255.255), which is a part of the campus network of Wuhan University, Wuhan, China. In this test, we imitated the basic principle of "PING" to detect the active parasitifers and recorded the average RTT, in which 772 active nodes were detected, while 16384 nodes were searched. We calculate the percentages of different RTTs as represented in Table 1.

In order to avoid the happenstances, we consider the distribution of RTTs without regard to the two least RTTs in Fig. 1. From the figure, we can see that the service time approximately complies with an exponential distribution.

Now, we take the $M/M/N$ queuing system (with FIFO discipline) as the reference analysis model for the parasitic storage. Before packets are dealt with by parasitifers, they should be checked using given protocols first. The newly arriving packet is arranged directly to the end of the

Table 1 Percentages of different RTTs

RTT/ms	number	percentage/%
1	7	0.91
2	65	8.40
3	256	33.10
4	192	24.80
5	114	14.70
6	91	12.50
7	31	4.01
8	9	1.16
9	3	0.38
10	2	0.26
11	1	0.13
38	1	0.13

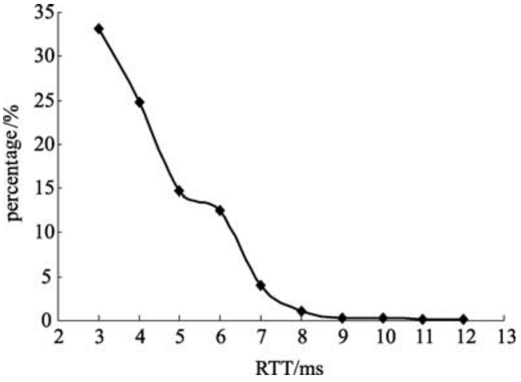

Fig. 1 Distribution of RTTs

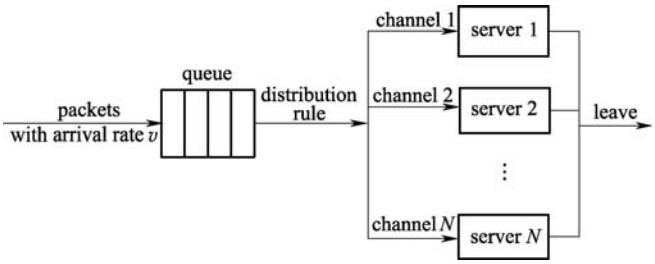

Fig. 2 Multiple servers queuing model

waiting queue in the cache of parasitifer if the parasitifer is busy. We consider that the system composes of a certain scale of parasitifers with waiting queue of infinite capacity. The packets arrive (namely, being received by parasitifers) according to a Poisson distribution with the parameter v, and the time of processing packets is exponentially distributed with the parameter μ.

Supposing the system distribution rule guarantees the packets arriving N parasitifers averagely, which means that each parasitifer will finally converge to the same stable situation. Hence, N branches of packets (each branch is a time-delayed discrete memoryless erasure link or channel) flow away from the total packets stream at arrival rate of $\lambda(=v/N)$.

We will analyze the system performance based on the model shown in Fig. 2.

3 Primary characteristics of parasitic storage

According to what was mentioned in Sect. 2, we can consider the parasitic storage model as N independent $M/M/1$ queues. When a packet is sent to the parasitifer and applying for the response on one channel, it is just like the customer waiting for the service of the parasitifer in the queue. For each $M/M/1$ queuing system, the packets arrive according to a Poisson process with parameter λ, and their arrival interval is $1/\lambda$. With the service rate μ, the package service time accords with an exponential distribution of parameter μ. The service times in different parasitifers are mutually independent and comply with the same distribution. We can first analyze one $M/M/1$ queue for further parsing the performance of the whole queuing system. For convenience, we give some symbols definitions [9] as follows:

γ: rate of the initiator sending packages;
ρ: parasitic utilization of the parasitifer;
q: average number of packages in the queue;
C: storage capacity of parasitic storage;
N: current packages number in the queue;
R: response time of a package in the parasitic storage;
T_q: average spending time of packages in parasitic storage.

Now, we consider the system characteristics at stable state. While the parasitifer is busy, the packages will enter the waiting queue and be maintained on the network for a period of time. The parasitic utilization of the parasitifer is

$$\rho = \frac{1/\mu}{1/\lambda} = \frac{\lambda}{\mu}. \tag{1}$$

A stable state of the system can be reached under the condition $\rho < 1$. Then, the average number of packages in the queue is

$$q = E(N) = \lambda E(R), \tag{2}$$

where the symbol $E(\cdot)$ denotes the mathematical expectation of a random variable and q represents the average capacity of one channel in parasitic storage. Intuitively, there will be a fraction of packages lost in the queue on the varying network, so it is needed to establish data redundancy mechanism to recover them.

A parasitic storage requires continuous retransmission of the packet in the queue. Note that γ and λ are the rate of sending packages and that of receiving packages, respectively. The speed of transmitting packages between the initial node and parasitifer is determined by the minimum of γ and λ. Since the RTT (the service time $1/\mu$) is the time delay taken for packages sending and receiving, the storage requires at least μ retransmissions per unit time. To describe the data loss on the network channel, we denote the probability of data loss as the parameter P_e. Supposing each package contains M bytes of data, then the maximum storage capacity that one queue supports is

$$C_{\max} = \frac{\min(\lambda,\gamma)}{\mu} \times (1-P_e) \times M = \min(\lambda,\gamma) \times \frac{1}{\mu} \times (1-P_e) \times M. \tag{3}$$

This result indicates that the maximum storage capacity increases with the service time and decreases with the packages loss on the data channel.

Therefore, we can get

$$E(N) \leqslant \min(\lambda,\gamma) \times \frac{1}{\mu} \times (1-P_e) \times M. \tag{4}$$

Consequently, the average response time of the package in the queue is

$$E(R) \leqslant \min(\lambda,\gamma) \times \frac{1}{\mu} \times \frac{1}{\lambda} \times (1-P_e) \times M. \tag{5}$$

Since the packages arrival rate is never bigger than the rate of the initiator sending packages, the minimum of the two is λ. Then, it is noted that the response time is limited by the service time and the probability of package loss at a given package scale.

4 Interaction between parasitic storage and network

In the parasitic storage, the spare bandwidth of network channel and the performance of the remote parasitifers would be depressed as the system is continuously sending and receiving data packets. In this section, we will discuss the impact of parasitic storage on the parasitifers and network and analyze the suitable rate at which the initiator sends packages.

4.1 Impact of parasitic storage on network

The network would be affected by the data packets transmission of parasitic storage. In order to make sure that the parasitic storage would not make obvious impact on network, we should consider the maximum capacity of a link that parasitic storage can occupy. On the other hand, the parasitifer performance would also be affected as it continually responds to the data packets of parasitic storage. However, we found that the impact on the network and parasitifers is difficult to be detected by traditional network testing, since the remote parasitifers are unaware of the storage and would response no extra data useful for tests.

In order to consider the impact of parasitic storage on the network, we take the parasitic effect on the parasitifer as reference. As the packages arrival rate is always smaller than the sending packages rate, we mainly consider the sending packages rate. The maximum rate of the initiator sending packages (denoted as $\gamma_{\max}$) is the margin between the packets number of the general period and that of the peak period on the network. It means that the network communication would be seriously depressed if the rate of the initiator sending packages surpasses $\gamma_{\max}$. With the given maximum rate of the initiator sending packages, $\gamma_{\max}$, the largest availability of each parasitifer on network can be obtained as

$$\rho_{\max} = \frac{\gamma_{\max}}{\mu}. \tag{6}$$

In order to control the affection on the parasitifer efficiently, we can set the value of γ under $\gamma_{\max}$.

4.2 Variance of parasitic storage capacity

Since the Internet is a large complex network, the packets transmission is easy to be affected by unexpected factors. Then, the capacity of parasitic storage, whose packets are being sustained on the network, would vary with the fluctuation of the network. Therefore, we consider using the average and variance of packets number in the queuing model to describe the dynamic capacity of parasitic storage. According to these, we could calculate the average number (denoted as E_c) and variance (denoted as D_c) of parasitic storage capacity as follows:

$$E_c = \sum_{i=1}^{\infty} i q_i = \frac{\rho}{1-\rho}, \tag{7}$$

$$D_c = E_c(N^2) - E_c^2(N) = \sum_{i=1}^{\infty} i^2 q_i - \left(\sum_{i=1}^{\infty} i q_i\right)^2 = \frac{\rho}{(1-\rho)^2}. \tag{8}$$

Based on the capacity variance of the parasitic storage, we can further consider the effective capacity of the storage. Using the queuing model, the effective parasitic storage capacity could be considered as the current packets number sustained on the network with the probability of $r\%$. Now, we let Q be the variable of packets number over one link of the network and $m_q(r)$ as the largest packets number with the probability of $r\%$ on the network. According to the probability theory, we have

$$P\{Q = N\} = (1-\rho)\rho^N, \tag{9}$$

$$\frac{r}{100} = \sum_{k=0}^{m_q(r)} (1-\rho)\rho^k = 1-\rho^{1+m_q(r)}. \tag{10}$$

4.3 Parasitic storage lifetime

From the queuing theory, the average spending time of packages in parasitic storage can be calculated as

$$T_q = E(R) = \frac{E(N)}{\lambda} = \frac{1}{\mu(1-\rho)}. \tag{11}$$

It indicates that the lifetime of parasitic storage is related to the latency of the network and the parasitic utilization of the parasitifer. For further analysis, we define $s\%$ as the maximum percentage of resources that the parasitifer distributes to parasitic storage, and the parameter $m_{T_q}(s)$ as the maximum expending time of a package in the parasitic storage under the former limit. Thus, we can get

$$m_{T_q}(s) = T_q \times \ln\frac{100}{100-s} = \frac{\rho}{\lambda(1-\rho)} \times \ln\frac{100}{100-s}$$
$$= \frac{1/\mu}{1-\rho} \times \ln\frac{100}{100-s}. \tag{12}$$

From Eq. (12), we can see that under certain scale of $s\%$, the utilization and service time of the parasitifer determine the maximum lifetime of parasitic packets.

4.4 Test and analysis of parasitic storage

In order to illustrate the practicability of parasitic storage, we have made corresponsive network test with Sniffer and other network test tools [10]. In the test, we collected a large number of available parasitifers, within a certain part of the China Education and Research Network (CERNET) as mentioned in Sect. 2, and considered the average of the RTTs as one of the main factors for judging whether a parasitifer is suitable or not. The test shows that the average value of RTTs is 3.9 ms. For appropriately determining the rate of the initiator sending packages, we take the gap between the transmitting packages rate during overload period of time and that of the general period of time as the appropriate rate of sending packages. The distributions of packages sustained on the network within the two periods are shown in Fig. 3, which are similar to the status of actual web traffic [11]. The upper curve is the distribution of transmitting packets rate in the overload stage of the network (e.g., the parasitifers are occupied by a great deal of video chatting, file transferring, etc.), and the lower curve is that of the normal period of time.

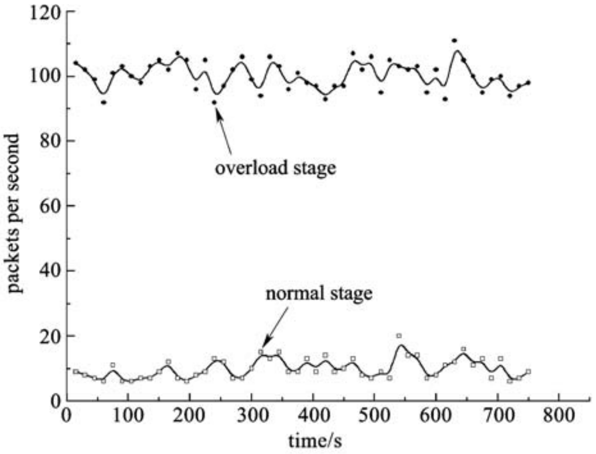

Fig. 3 Distributions of transmitting packets rate in overload and normal stages

From this, we can calculate that the average rate of transmitting packets during general period of time is 9.90 packages per second and 100.24 packages per second during overload stage. The subtraction between them is regarded as the maximum permissible rate, $\gamma_{\max}= 90.34$ packages per second, of the initiator sending packages for parasitic storage. Then, the maximum available utilization of the parasitifer is

$$\rho_{\max} \leqslant \frac{\gamma_{\max}}{\mu} = 90.34 \times 3.9 \times 10^{-3} \times 100\% = 35.23\%.$$

Under this utilization, the impacts of parasitic storage on the parasitifers and the network can be ignored. Furthermore, we could calculate the average of transmitting packages number $E_c \approx 0.544$ and the variance of it is $D_c \approx 0.84$.

From the above, we can get the largest packages number with the probability of 90%,

$$m_q(90) = \frac{\ln 0.1}{\ln 0.3523} - 1 = 1.21.$$

Now, assuming each package size is 50 kB, and the number of available parasitifers is 70000 (about 100 campus networks), we can get that the credible capacity of parasitic storage is more than 4 GB. If the package size is larger, and more available parasitifers are used, the capacity of parasitic storage will reach TB or even more.

As both the network protocols and the number of available parasitifers restrict the data packet size, the capacity of parasitic storage depends on the former two factors essentially. By the way, the utilization of parasitifer is also an important factor for parasitic capacity.

4.5 Scalability and practicality of parasitic storage

As is well known, the next generation of Internet (namely Internet2, which is based on IPv6) possesses the properties of more fast access, even huge space with rapidly increasing resources of computing, communication and storage. On such a network environment, it is no doubt that much more free resources might be dug out for further use. For instance, the spare bandwidth could be transformed as a new kind of storage power. Hence the scalability of parasitic storage on the next generation of Internet is obvious.

In addition, as discussed in Sects. 4.1–4.4, the practicality of parasitic storage can be seen owing to the following process strategies: firstly, as to make no obvious impact on network, we can consider the maximum capacity of a link to be occupied by parasitic storage; secondly, by taking the gap between the transmitting packages rate during overload period of time and that of the general period of time as the appropriate rate of sending packages, the maximum rate of the initiator sending packages can be determined, and under such limitation the network communication would not be seriously depressed; finally, under the maximum available utilization of the parasitifer, the impacts of parasitic storage on the parasitifers can be ignored. Moreover, the parasitic storage can be constructed on the base of existing network protocols without any need to install particular services in remote parasitifers.

5 Certain issues to be considered

Anyway, parasitic storage is a threat to the current Internet even in small scale. It utilizes the resources of network without authorization, which results in choking of network and overloading of hosts if the network is flooding with immense parasitic packages. However, current network nodes do not have the ability to identify and deny parasitic storage due to their foundation of defective protocols.

Resource sharing is a benefit that Internet brings, but sometimes, it may be accompanied by other problems. For instance, a file transfer protocol (FTP) server may break down owing to too much client's downloading, and a possible solution is to restrict the download speed or clients number. However, usually, the FTP server has to turn to clients' moral for sustainable download. The situation is the same as in parasitic storage. Even if we have lots of hosts, but we have much more users, if every user squeezes hosts as much as possible the host absolutely will collapse. Therefore, parasitic storage requires advisably utilization to reduce its affection on network and hosts.

Technically, there are some solutions to prevent abuse of parasitic storage:

1) Limit the number of every user's hosts and the frequency of users resending packets. Parasitic storage influences network by filling the channels connected to hosts with packets, so to decrease the packets sending rate is effective for maintaining utilization rate within an acceptable range.

2) Modify common protocols to prevent them from echoing data back to the sender without authentication. It is impractical to modify current protocols absolutely, but we can do some little tricks to achieve our purpose. Taking ICMP as an example, we can close the port corresponding to ICMP, which many servers used to shield PING request. However, parasitic storage can be implemented in many ways and is not easy to be denied.

The methods mentioned above are effective in a certain sense. Though parasitic storage is not as harmful as virus, it is necessary to make use of the resource on the network carefully.

6 Conclusions and future work

In this paper, the basic concept of parasitic storage is introduced, and a specific queuing model is given for analyzing parasitic storage. Based on this model, the interaction between parasitic storage and the network is also discussed. Furthermore, the dynamic characteristics of parasitic storage on complex network are tested and analyzed. Finally, some other related problems are pointed out.

Based on the analysis, some important parameters of parasitic storage, such as the service rate μ, the packages arrival rate λ, and the host utilization ratio ρ are described. The average E_c and variance D_c of capacity, etc., are also considered to reflect the parasitic storage dynamical characteristics. Furthermore, the following properties of parasitic storage can be achieved: 1) remote hosts are almost unaware of the storage; 2) affection of parasitic storage on the network can be ignored under certain conditions; 3) capacity of parasitic storage is restricted by protocols and status of the network; 4) lifetime of parasitic storage is related to the service time and host utilization ratio on the network; 5) unlike virus, parasitic storage does no harm to the network and hosts.

With varying network environments, parasitic storage may behave with different dynamic characteristics. Our future work will focus on the particular analysis of parasitic storage on different networks, such as wireless network, etc. According to network protocols, we can establish different storage mechanisms for parasitic storage to meet various storage demands. With the burst of the network, parasitic storage will alleviate the growing pressure on storage by storing data on the network without additional cost, and hence have extensive prospects.

Acknowledgements The authors thank Ms. Wang Ning for her helpful discussion. This work was supported by the National Natural Science Foundation of China (Grant No. 60773189), the National Basic Research Program of China (No. 2004CB318201).

References

1. Chen J C, He P, Ge X Z. Dynamics analysis method of cellular automata for complex networking storage system. Journal of Software, 2008, 19(10): 2517–2526
2. Chen J C. The potential of storage parasitized on the Internet.

Communications of the CCF, 2008, 4(11): 59–62 (in Chinese)

3. Barabasi A L, Freeh V W, Jeong H, Brockman J B. Parasitic computing. Nature, 2001, 412(6850): 894–897
4. Zalewski M. Silence on the Wire: A Field Guide to Passive Reconnaissance and Indirect Attacks. San Francisco: No Starch Press, 2005
5. Gill J, Burge L, Li J. Floating parasitic data storage. In: Proceedings of the International Conference on Internet Computing. Bogart: CSREA Press, 2004, 960–965
6. Rosenfeld K, Sencar H T, Memon N. Volleystore: a parasitic storage framework. In: Proceedings of the IEEE Workshop on Information Assurance and Security. Los Alamitos: IEEE, 2007, 67–75
7. Wang Y X, Jiang B L, Wang C Y. Probability Theory, Stochastic Processes and Mathematical Statistics. Beijing: Beijing University of Posts and Telecommunications Press, 2008 (in Chinese)
8. Bolch G, Greiner S, de Meer H, Trivedi K S. Queuing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications. New York: John Wiley and Sons, 1998
9. Lin C. Computer Networks and Computer Systems Performance Evaluation. Beijing: Tsinghua University Press, 2001 (in Chinese)
10. Melander B, Bjorkman M, Gunningberg P. A new end-to-end probing and analysis method for estimating bandwidth bottlenecks. In: Proceedings of the IEEE Global Telecommunications Conference. Piscataway: IEEE, 2000, 1: 415–420
11. Barford P, Crovella M. Generating representative web workloads for network and server performance evaluation. ACM SIGMETRICS Performance Evaluation Review, 1998, 26(1): 151–160