# A Bandwidth Assignment Method for Downloading Large Files with Time Constraints

Ken Katsumoto, Kazuhiko Kinoshita,
and Koso Murakami
Department of Information Networking,
Graduate School of Information Science and Technology,
Osaka University
1-5 Yamadaoka, Suita-shi, Osaka, Japan
Email:{katsumoto.ken,kazuhiko,murakami}@ist.osaka-u.ac.jp

Nariyoshi Yamai
Center for Information Technology and Management,
Okayama University
3-1-1 Tsushima-naka,
Kita-ku, Okayama-shi, Okayama, Japan
Email: yamai@cc.okayama-u.ac.jp

***Abstract*—In recent years, the numbers of requests to download large files via large high-speed computer networks have been increasing rapidly. Typically, these requests are handled in a "best effort" manner, resulting in unpredictable completion times. In this paper, we consider a model where a download request either must be completed by a user-specified deadline or must be rejected if the deadline cannot be satisfied. We propose a dynamic bandwidth assignment method for reducing the call-blocking probability in a bandwidth-guaranteed network. Finally, we present simulations that show its excellent performance.**

***Keywords*—*file downloading; time constraints; bandwidth assignment.***

## I. INTRODUCTION

In recent years, various types of data have become available in large quantities via large high-speed computer networks [1]. Users hope to be able to access these data files routinely and rapidly by fast downloading.

There are many studies on file downloading, but most focus on shortening the average download-completion time [2][3][4]. In such studies, it is difficult to predict and/or guarantee download completion times, because they depend strongly on the network conditions [5][6].

To overcome this problem, one study has introduced a model where a download request must either be completed by a user-specified deadline or be rejected if the deadline cannot be satisfied [7][8]. Note that, in this model, it is not necessary to shorten the downloading time below its deadline, and it is preferable to accept requests wherever possible, thereby reducing the number of rejected requests. To handle many requests that will meet their deadline and to reduce the call-blocking probability, it is important to consider the bandwidth assignment for each request and to allow a margin in the network for handling future requests.

In this downloading model, a dynamic bandwidth assignment method called $ChangeRates$ has been proposed [8]. This achieves a reduction in call-blocking probability by considering the minimum bandwidth that will meet the deadline.

To be able to accept additional requests, it is preferable that there be as many ongoing requests with loose deadlines as possible in the network. In this paper, we propose a dynamic bandwidth assignment method that reduces the call-blocking probability by giving a higher priority to those requests that potentially allow wider margins.

The remainder of the paper is structured as follows. Section II presents the method for downloading within a deadline, using an existing bandwidth assignment method. In Section III, we propose a dynamic bandwidth assignment method and evaluate its performance in Section IV. Section V concludes the study.

## II. DOWNLOADING FILES WITH TIME CONSTRAINTS

### A. Problem Formulation

A download request with time constraint $R_i (i = 1, 2, ...)$ is defined by the tuple [9]:

$$R_i = (s_i, d_i, A_i, F_i, D_i). \tag{1}$$

As suggested by their names, $s_i$ = source node, $d_i$ = destination node, $A_i$ = arrival time of the request, $F_i$ = file size, and $D_i$ = the request's deadline. In this formulation, request $i$ must be completed by $A_i + D_i$. Note that, as time elapses, $F_i$ and $D_i$ will decrease. We therefore describe them as $F_i(t)$ and $D_i(t)$, respectively, where $t$ denotes the current time.

For each request $R_i$, $MinRate_i(t)$ is defined as the minimum average transfer rate that will meet the request's deadline. $MinRate_i(t)$ can be determined from the file size $F_i(t)$ and deadline $D_i(t)$:

$$MinRate_i(t) = \frac{F_i(t)}{D_i(t)}. \tag{2}$$

In addition, $MaxRate_i(t)$ is defined as the maximum bandwidth that can be assigned to $R_i$, i.e., the available bandwidth for the path [10]. This is given by the minimum available bandwidth among all links within the path. The available bandwidth for each link will vary according to the bandwidth assignment method. For example, if the assigned bandwidth is fixed, the available bandwidth is just the residual capacity of the link. However, if the assigned bandwidth is adaptive, the available bandwidth will be the link capacity minus $MinRate$ for the existing requests.

Fig. 1 shows an example of the available bandwidth for a path. The link capacities for links A–B, B–C, and C–D are all 100 Mbps. The existing request is assigned 30 Mbps for link A–B and 50 Mbps for link C–D. The assigned bandwidth is fixed. In this case, the available bandwidths for A–B, B–C, and C–D are 70, 100, and 50 Mbps, respectively. As a result, the $MaxRate$ for the path A–D is 50 Mbps, which is the minimum available bandwidth for the links A–B, B–C, and C–D.

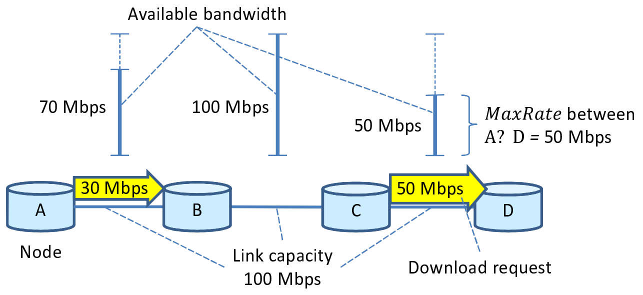

Figure 1. $MaxRate$ for the path

## B. Download Model

In this paper, we assume the following network characteristics. The bandwidth assigned to each download connection is guaranteed. There is a database for managing essential information, such as network topology, link capacity, and ongoing requests, for path finding and bandwidth assignment [11]. The access networks are sufficiently fast that they cannot become potential bottlenecks.

For such a network, we consider the following download model. First, a user requests a download within an allowable deadline. For this new download request, a search is made for a feasible route that will satisfy the request. If such a route is not found, the request is rejected. Otherwise, the request's route is decided, and the assignment of an adequate bandwidth is considered. The assignment of appropriate bandwidth for meeting the deadline is an important problem.

## C. Existing Methods

We introduce two typical bandwidth assignment methods [9].

- $Max$: always assigns $MaxRate$ on demand.
- $Min$: always assigns $MinRate$ on demand.

$Max$'s advantage is that the whole bandwidth is used and almost no bandwidth remains idle, but it tends to lead to resource competition at high network loads and to rejection of future requests.

Conversely, downloading is inefficient and takes more time when using $Min$. However, much bandwidth will remain idle for future requests, and the method can handle many requests in parallel.

For these fixed bandwidth assignment methods, if no path with at least $MinRate$ bandwidth is available for the request, the request is rejected.

Therefore, we consider an existing method called $ChangeRates$ that changes the assigned bandwidth dynamically [12]. For this method, the bandwidth assigned to request $R_i$ is proportional to $MinRate_i(t)$. Note that this will change during downloading. The specific behavior of this method is as follows.

When a new request occurs, $ChangeRates$ first searches for a path with at least $MinRate$. If found, $MaxRate$ is assigned to the request. Otherwise, a process that reassigns the bandwidths for ongoing requests is invoked, as follows.

For each link $C_j$, $\theta_j$ is computed by:

$$\theta_j = \frac{C_j}{\sum MinRate_i}, \tag{3}$$

where $\sum MinRate_i$ is the sum of the $MinRate$ values for ongoing requests using link $C_j$. For cases where $\theta_j \geq 1$, a new request can use the link by changing the assigned bandwidths for the ongoing requests. A path that only uses links with $\theta_j \geq 1$ is therefore sought. If such a path for assignment to the new request cannot be found, the request is rejected. Otherwise, for each link $C_k$ on the path, $Rate_{C_k}$, the assigned bandwidth for $R_i$, is calculated by:

$$Rate_{C_k} = \theta_k \times MinRate_i. \tag{4}$$

$R_i$'s assigned bandwidth $Rate_i$ is the bottleneck bandwidth for the path and is determined by the minimum $Rate_{C_k}$:

$$Rate_i = \min(Rate_{C_k}). \tag{5}$$

$ChangeRates$ can reduce the blocking probability to below that for the fixed bandwidth assignment methods.

# III. A BANDWIDTH ASSIGNMENT METHOD

## A. Proposed Method

For $ChangeRates$, the assigned bandwidth is simply proportional to $MinRate$. However, to reduce the blocking probability, it would be more effective to handle requests preferentially, thereby producing a greater time margin for the network. A time margin is defined as a download time that could be shortened by assigning a bandwidth greater than $MinRate$. We therefore consider a bandwidth assignment method with the following policies.

- Define an evaluation value for each request. This value indicates a time margin to be obtained by considering the use of bandwidth resources and the use of time.
- Assign bandwidths in descending order of evaluation value.

We define $E_i$, which is an evaluation value for each request $R_i$, using the residual file size $F_i$, the number of hops of the assigned path $H_i$, the maximum assigned bandwidth $MaxRate_i$, and $MinRate_i$ as follows:

$$E_i = F_i \times H_i \times (1 - \frac{MinRate_i}{MaxRate_i}). \tag{6}$$

A large value for $F_i$ shows that there is room to produce a time margin for $R_i$. A large $H_i$ indicates that $R_i$ tends to use network resources heavily. Finally, a large

$1 - MinRate/MaxRate$ shows that a greater time margin may be obtained when $R_i$ receives $MaxRate$ compared with $MinRate$. By assigning $MaxRate_i$ to the $R_i$ that has the largest $E_i$, a greater time margin is obtained and the flexibility in bandwidth assignment is improved. As a result, this method is able to handle more requests and reduces the blocking probability.

We now explain the specific procedures in the proposed method. Suppose that a new request $R_{n+1}$ arrives while requests $R_i(i = 1, ..., n)$ are ongoing. First, the proposed method searches for a feasible path for $R_{n+1}$. In this process, Dijkstra's algorithm is applied using the inverse of the available bandwidth of a link as the link cost.

Next, the proposed method calculates the evaluation value for all requests, and assigns $MaxRate$ to the request that has the largest evaluation value. The evaluation value is then recalculated for the requests that are yet to be assigned a bandwidth, with the assigned bandwidth also being determined as $MaxRate$. These processes are repeated until the assigned bandwidths for all requests are determined. Furthermore, on the completion of an ongoing request, the same bandwidth assignment procedure is invoked.

Figs. 2–5 show an example of the execution of the proposed algorithm.

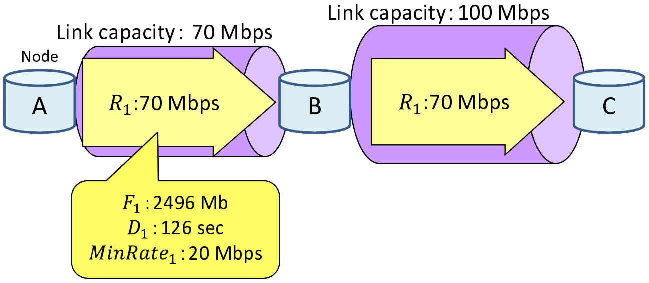

Figure 2. Execution example (1/4)

First, Fig. 2 shows the arrival of a new request $R_1$ with an $F_1$ of 2496 Mb and a $D_1$ of 126 sec. $MinRate_1$ is therefore 20 Mbps. However, in the absence of other requests, $R_1$ receives 100 Mbps, which is the capacity of the link A–B.

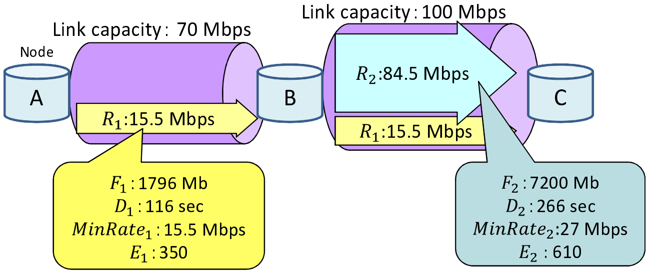

Figure 3. Execution example (2/4)

Next, at 10 sec after $A_1$, Fig. 3 shows the arrival of $R_2$, which has an $F_2$ of 7200 Mb and a $D_2$ of 266 sec. Because $R_1$ has consumed 70 Mbps for 10 sec, $F_1$ and $MinRate_1$ are recalculated as follows:

$$F_1 = 2496 - 70 \times 10 = 1796\text{Mb}. \tag{7}$$

$$MinRate_1 = \frac{1796}{116} \fallingdotseq 15.5\text{Mbps}. \tag{8}$$

Here, $E_1$ and $E_2$ are calculated as follows:

$$E_1 = 1796 \times 2 \times (1 - \frac{15.5}{70}) \fallingdotseq 350. \tag{9}$$

$$E_2 = 7200 \times 1 \times (1 - \frac{27}{84.5}) \fallingdotseq 610. \tag{10}$$

Therefore, 84.5 Mbps of $MaxRate_2$ has to be assigned to $R_2$, with the remaining bandwidth of 15.5 Mbps being assigned to $R_1$.

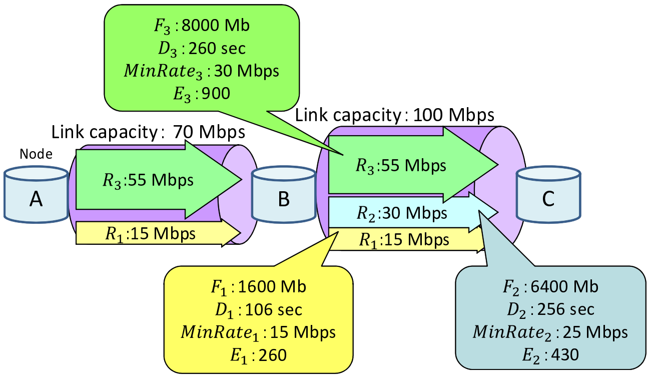

Figure 4. Execution example (3/4)

Next, at 10 sec after $A_2$, Fig. 4 shows the arrival of $R_3$, which has an $F_3$ of 8000 Mb and a $D_3$ of 260 sec. $R_1$ and $R_2$ have consumed 15.5 Mbps and 84.5 Mbps for 10 sec, respectively. $F_1$ and $F_2$ are recalculated as follows:

$$F_1 = 1796 - 15.5 \times 10 \fallingdotseq 1600\text{Mb}. \tag{11}$$

$$F_2 = 7200 - 84.5 \times 10 \fallingdotseq 6400\text{Mb}. \tag{12}$$

In the same way, the $E_i$ value for each request is calculated as follows:

$$E_1 = 1600 \times 2 \times (1 - \frac{15}{45}) \fallingdotseq 260. \tag{13}$$

$$E_2 = 6400 \times 1 \times (1 - \frac{25}{55}) \fallingdotseq 430. \tag{14}$$

$$E_3 = 8000 \times 2 \times (1 - \frac{30}{55}) \fallingdotseq 900. \tag{15}$$

At this stage, an assigned bandwidth for $R_3$ that has the highest evaluation value is considered. For $R_3$, 60 Mbps (100 Mbps of link capacity minus $MinRate_1$ and $MinRate_2$) can be assigned to the link B–C. However, only 55 Mbps (70 Mbps of link capacity minus $MinRate_1$) can be assigned to the link A–B. Therefore, 55 Mbps is assigned to $R_3$. Next, $E_1$ and $E_2$ are recalculated as follows:

$$E_1 = 1600 \times 2 \times (1 - \frac{15}{15}) = 0. \tag{16}$$

$$E_2 = 6400 \times 1 \times (1 - \frac{25}{30}) \fallingdotseq 130. \tag{17}$$

Therefore, 30 Mbps is assigned to $R_2$, which has the larger evaluation value, and the remaining bandwidth of 15 Mbps is assigned to $R_1$.

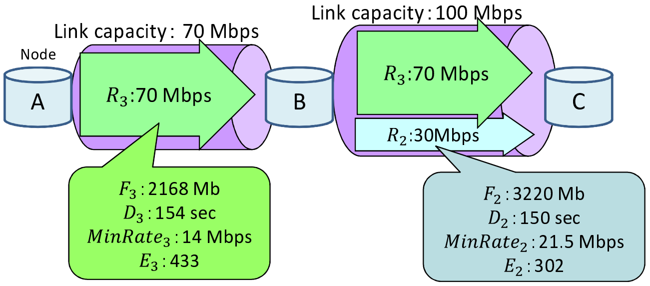

Figure 5. Execution example (4/4)

With no additional requests arriving, $R_1$ completes at 106 sec after $A_3$, as shown in Fig. 5. Here, $F_2$ is $6400 - 30 \times 106 = 3220\text{Mb}$ and $F_3$ is $8000 - 55 \times 106 = 2168\text{Mb}$. $E_2$ and $E_3$ are calculated as follows:

$$E_2 = 3220 \times 1 \times (1 - \frac{21.5}{86}) \fallingdotseq 302. \tag{18}$$

$$E_3 = 2168 \times 2 \times (1 - \frac{14}{70}) \fallingdotseq 433. \tag{19}$$

Therefore, 70 Mbps is assigned to $R_3$, with $R_2$ receiving the remaining bandwidth of 30 Mbps.

## IV. PERFORMANCE EVALUATION

### A. Simulation Model

We evaluated the performance of the proposed method by experimental simulation. In the simulation, the network had Waxman's random topology [13], with 100 nodes and about 300 links. Each link in the network had a uniform capacity of 1 Gbps. The download requests were generated via a Poisson arrival process, with an average arrival rate of $\lambda$. The source and destination nodes for each request were selected randomly. The blocking probability was used as the performance measure and the existing $ChangeRates$ method was used as a method for comparison.

### B. Simulation Results

The proposed method was evaluated for the scenarios described below.

*1) Scenario 1:* This scenario enabled the basic performance of the proposed method to be evaluated. In this scenario, all requests involved a file size of 5 GB and a deadline of 200 sec. Fig. 6 shows the results, where the proposed method outperforms the existing method for any average arrival rate.

*2) Scenario 2:* This scenario was used to evaluate the performance in a situation where three requests with equal $MinRate$ arrive, having file sizes of 2.5 GB, 5 GB, and 7.5 GB, and deadlines of 100 sec, 200 sec, and 300 sec, respectively. The total blocking probability for this scenario is shown in Fig. 7. This graph is similar to that for Scenario 1. Furthermore, as shown in Fig. 8, when the $MinRate$ at each request's arrival was the same, we can note that the number of rejected requests is almost the same regardless of the request's file size and deadline.

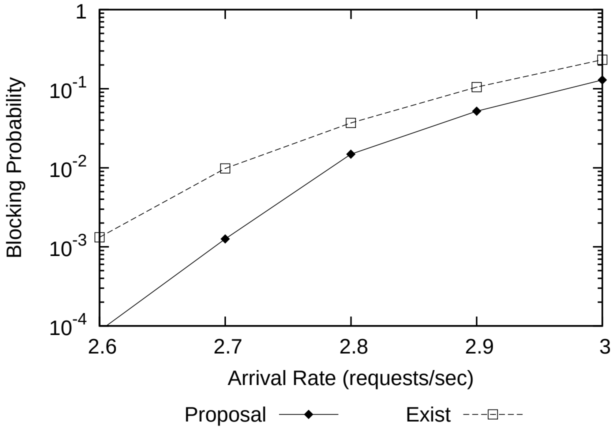

Figure 6. Scenario 1

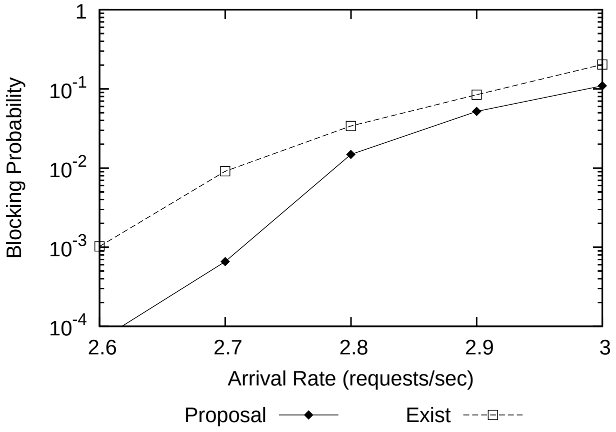

Figure 7. Scenario 2 : total

*3) Scenario 3:* The final scenario aimed to evaluate the effect of differences in the requests' deadlines. We assume the arrival of three requests that have the same file size of 5 GB but different deadlines of 100 sec, 200 sec, and 300 sec. As shown in Fig. 9, we can note the reduction in the blocking probability for the proposed method in this scenario. Fig. 10 shows that the proposed method has a low blocking probability for requests of 100 sec at an high arrival rate, but the existing method is low for requests of 200 sec and 300 sec. This indicates that the existing method could reduce the blocking probability by handle many requests which would load to the network more lightly, and it has no room for the network than the proposed method. Therefore, it is considered that the proposed method to be more effective for requests of the short deadline.

## V. CONCLUSION

This paper has focused on downloading large files with time constraints. We have proposed a dynamic bandwidth assignment method for reducing the call-blocking probability and have evaluated its performance by experimental simulations. The simulation results show that our proposed method is effective.

In future work, we will enhance the proposed method to enable it to work with distributed management. In addition,

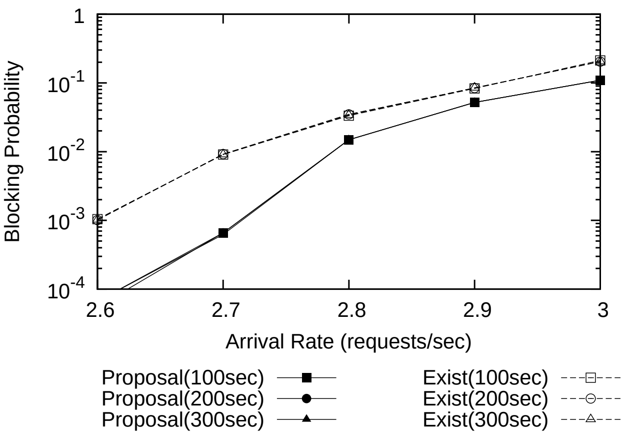

Figure 8. Scenario 2 : each request

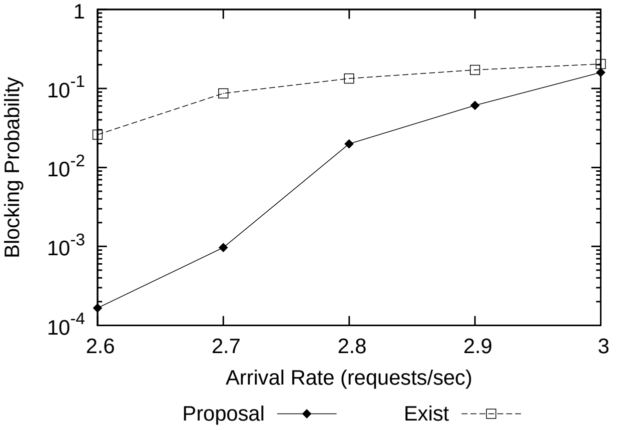

Figure 9. Scenario 3 : total

we will investigate routing methods that are better suited to the proposed bandwidth assignment method.

## ACKNOWLEDGMENT

This research was partly supported by the National Institute of Information and Communications Technology (NICT).

## REFERENCES

[1] A. S. Tanenbaum, “Computer Networks Fourth Edition,” Prentice-Hall, New Jersey, 2003.

[2] H. Okada, T. N. Trung, K. Kinoshita, N. Yamai, and K. Murakami, “A Cooperative Routing Method for Multiple Overlay Networks,” Proceedings of the 6th Annual IEEE Consumer Communications and Networking Conference (CCNC 2009), pp. 1–2, Jan. 2009.

[3] L. Toka, M. Dell’Amico, and P. Michiardi, “Data Transfer Scheduling for P2P Storage,” IEEE P2P, pp. 132–141, Sept. 2011.

[4] Y. M. Chiu and D. Y. Eun, “Minimizing File Download Time in Stochastic Peer-to-Peer Networks,” IEEE/ACM Trans. Networking, vol. 16, no. 2, pp. 253–266, Apr. 2008.

[5] S. Kamei, “Status and Traffic Issues of Peer-to-Peer Technology,” Computer Software, vol.22, no.3, pp. 8–18, 2005.

[6] S. Gorinsky and N. Rao, “Dedicated Channels as an Optimal Network Support for Effective Transfer of Massive Data,” Proceedings of 25th IEEE International Conference on Computer Communications, pp. 1–5 Apr. 2006.

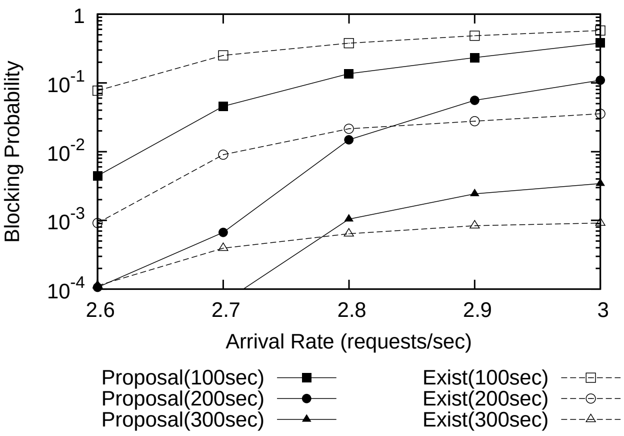

Figure 10. Scenario 3 : each request

[7] B. Chen and P. Primet, “Scheduling deadline-constrained bulk data transfers to minimize network congestion,” Proceedings of 7th IEEE International Symposium Cluster Computing and the Grid (CCGRID), pp. 410–417, May 2007.

[8] D. Andrei, M. Batayneh, S. Sarkar, C. Martel, and B. Mukherjee, “Deadline-Driven Bandwidth Allocation with Flexible Transmission Rates in WDM Networks,” Proceedings of the IEEE International Conference on Communications 2008 (ICC 2008), pp. 5354–5358, May 2008.

[9] D. Andrei, “Efficient Provisioning of Data-Intensive Applications over Optical Networks,” Ph.D. thesis, UC.Davis, 2009.

[10] R. S. Prasad, M. Murray, C. Dovrolis, and K. Claffy, “Bandwidth Estimation: Metrics, Measurement Techniques, and Tools,” IEEE Network, Vol. 17, No. 6, pp. 27–35, Nov.–Dec. 2003.

[11] M. Yamazaki, Y. Hirota, K. Kinoshita, H. Tode, and K. Murakami, “A Service Provision Method for Service Platform Unified with Network Control,” IEICE technical report, ICM2011-43, pp. 67–72, Jan. 2012.

[12] D. Andrei, M. Tornatore, M. Batayneh, C. U. Martel, and B. Mukherjee, “Provisioning of Deadline-Driven Requests With Flexible Transmission Rates in WDM Mesh Networks,” IEEE/ACM Transactions on Networking, Vol. 18, No. 2, pp. 353–366, Apr. 2010.

[13] B. M. Waxman, “Routing of multipoint connections,” IEEE Journal of Selected Areas in Communications, Vol. 6, No. 9, pp. 1617–1622, Dec. 1988.