Fair joint multiple resource allocation method in all-IP networks

Kenichi HATAKEYAMA† , Shigehiro TSUMURA †† and Shin-ichi KURIBAYASHI†

† Faculty of Science and Technology, Seikei University, Tokyo, JAPAN
†† 2nd Computers Software Division, NEC Corporation, Tokyo, JAPAN

Abstract: *In all-IP networks including NGN, both computing and network resources are required to be allocated simultaneously to each service request. The authors proposed optimal joint resource allocation methods for multiple resource types. Although those methods can achieve an efficient use of resources, it may result in an 'unfair' use of resources in which resources may be monopolized by a specific service.*

This paper proposes basic principles for achieving 'fairness' among multiple services, assuming that both computing and network resources are required to be allocated simultaneously to each service request, and proposes a measure for evaluating fair allocation. Next, this paper proposes a ***fair*** *joint multiple resource allocation method which tries to equalize the total amount of key resources allocated for each service in each time block. Key resource is the resource type for which the total amount of requested resource is the largest proportion of the maximum resource of that type. It is demonstrated by simulation evaluations that the proposed method enables fair allocation among multiple services.*

Key words: fairness, resource allocation, all-IP network

1. Introduction

All-IP networks including NGN[1],[2], grid computing[3] and ubiquitous networks, need to provide both computing power of servers distributed in various locations and access bandwidth between the servers and their clients for each service. The resource allocation should be one of the important issues in all-IP networks, to offer economical services to users.

We had regarded resource allocation in all-IP networks as a model in which a system selects one resource set from among multiple resource sets of servers and bandwidths whenever a request occurs, and 'simultaneously' allocates both the requested computing power (computing resource) and bandwidth (network resource) [4]. Since, the requested size of computing power and that of bandwidth are not uniform for all requests (the size varies from request to request), the resource allocation algorithms designed for a single resource type could not result in an optimal allocation of multiple resource types. To solve this problem, the authors had proposed optimal joint resource allocation methods for multiple resource types, and demonstrated the effectiveness by simulation evaluations [4]-[6]. However, those methods, which focus on an efficient use of resources, may result in an 'unfair' use of resources in which resources may be monopolized by a specific service or a specific user.

This paper first discusses basic principles for achieving fairness among multiple services, assuming that both computing and network resources are required to be allocated simultaneously to each service request, and a measure for evaluating fair allocation. When there is no resource available at the time when a request occurs, some resource allocation algorithms, such as those in References [3] and [7], the delayed resource allocation with queuing until a resource becomes available. However, this paper assumes that resources are in principle allocated when a request occurs. Next, this paper proposes a fair joint resource allocation method that is based on the proposed principles, and demonstrates its effectiveness by simulation evaluations. Finally, the paper presents the conclusions.

2. Principles for achieving fairness in joint multiple resource allocation

2.1 Evaluation model [4]

We adopt the evaluation model as illustrated in Figure 1. There are Y different centers at different locations, and each center has a resource set of computing resource and network resource. When a request occurs, an optimum resource set is selected from among Y resource sets, and both computing and network resources are simultaneously allocated in the selected resource set. The resources are allocated for the period of resource hold time, and then released.

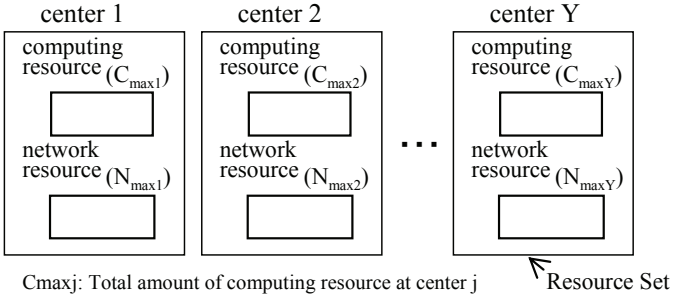


Figure 1. Evaluation model

2.2 Principles for achieving fairness among multiple services

The following four principles are proposed to achieve fairness in joint multiple resource allocation:

<Principle 1> One possible condition for achieving fairness is to put requests in a queue and allocate resources on a delayed basis as in References [3] and [7]. Since this paper assumes that a resource is allocated at the time when a request occurs, it aims to achieve fairness without queuing. However, it allows delayed allocation of resources to serve some requests if it is necessary to achieve fairness.

<Principle 2> Fairness should be pursued while taking multiple types of resource into consideration.

There are many papers that discuss algorithms for achieving fairness for cases where only one type of resource is allocated [7]-[10]. Such algorithms cannot be applied to cases where multiple resource types are allocated simultaneously, because an allocation that is fair for one specific resource type may be unfair when considering other resource types. This is based on the fact that the size of the requested computing resource and that of the requested network resource are not uniform for all requests and both resource types should be allocated simultaneously.

<Principle 3> Equalizing the total amount of both computing and network resources for all services would not achieve fairness.

A possible solution to satisfy principle 2 may provide an equal amount of both computing and network resources to each service. However, this solution has a problem as illustrated in Figure 2. In this example, there are two services, each of which is allocated half the amount of the computing and network resources combined. If resources are allocated like pattern 2, it will take up almost all network resources, although the total amount of allocated resources to the service does not exceed the maximum allowable size. In this case, it is impossible to meet requests from other services that may have patterns 1 or 2 (which require a large total network reqources) in Figure 2.

<Principle 4> It is considered that enough resource has been allocated to the service and it is not unfair when there are no request rejections occurred for all services, even if there is an imbalance in the amount of allocated resources to each services.

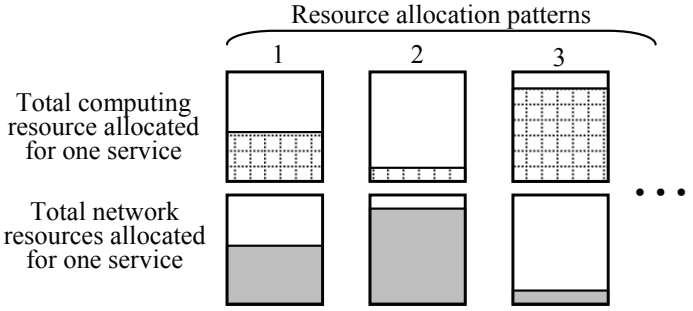


Figure 2. Possible resource allocation patters

3. Joint multiple resource allocation for achieving fairness

3.1 Basic concepts

1) It is proposed to focus on one resource type, "**key resource**" for which the requested amount is the largest proportion of the maximum resource of that type, comparing the total amount of required resource with the maximum resource size for each resource type, as in the following example. Suppose that the maximum amount of computing resource is 1000 MIPS and the maximum amount of network resource is 100Mb/s. A request for 200MIPS and 30Mb/s requires 20% of computing resource and 30% of network resource respectively. As the proportion of required network resource is larger than that of required computing resource, the network resource is identified as the key resource in this case. This idea is very similar to the idea of Method II in Reference [4].

2) It is supposed that there are G types of service (service 1 ～ service G). This paper defines that the **fairness** will be achieved by allocating an equal amount of key resource to each service in every time block. Note that the key resource can change over time and can be different for each service.

3.2 Measure of 'fair allocation'

3.2.1 Conditions

1) The total amount of resources allocated to each resource type in a time block of length L is calculated, and then the resource type for which the total requested amount is largest portion of the maximum resource of that type is selected as the key resource. Note that L is assumed to be longer than resource hold time H.

2) When service g_0 is the service to which the largest total amount of key resources assigned in i-th time block among G services, the difference between the total amount of key resources assigned to service g_0 in i-th time block and those assigned to service g in i-th time block is set to **Mi (g)**. Mi (g) is assumed to be an imbalance on allocated resources in i-th time block for service g (g=1～G).

3) If no request of service g is rejected in i-th time block, it is considered that the resource allocation is not unfair for service g, and Mi (g) is set to 0, even if there are some differences in the amount of resource allocated.

3.2.2 Measure of fairness

It is proposed to check the value F by equation (1) and to judge that the smaller the value of F is, the fairer the resource allocation is:

$$\mathbf{F}= \sum_{g=1}^{G} \left[\{ \sum_{i=1}^{k} \mathrm{Mi}\,(g) \}/ k \right] \qquad (1)$$

where k is the total number of time blocks.

3.3 Fair joint multiple resource allocation method

3.3.1 Overview

This section proposes a new joint multiple resource allocation method, achieving fair resource allocation to multiple services. As it is assumed in this paper that resources will be allocated only when the service request occurs (no queuing is assumed), it is difficult to take any action in advance, avoiding any imbalance on the total amount of resources allocated to different services.

After this section, it is supposed that there are two types of service (G=2). An imbalance on allocated resources in i-th time block between service 1 and service 2 is set to **Mi**, and Mi (1) is set to Mi if Mi (1) is larger than Mi (2), otherwise Mi (2) is set to Mi.

It is proposed that the imbalance Mi in i-th time block will be filling up in (i+1)-th time block, by applying the delayed resource allocation only to the service that suffered imbalance in i-th time block. That is, if there are not enough resources available when a request of the service occurs, the resource allocation will be delayed until the required size of resources are available, instead of rejecting the request. This is the same idea with Method III in Reference[5]. The delayed resource allocation will be achieved until the total amount of key resources allocated for delayed requests exceeds Mi in (i+1)-th time block.

It is impossible to decide which resource type is key resource before the end of (i+1)-th time block. Therefore, both Mi-c (imbalance of computing resource) and Mi-n (imbalance of network resource) are calculated from the beginning of each time block and the request will be rejected when Mi-c or Mi-n exceeds Mi.

3.3.2 Algorithm to fill up the imbalance in (i+1)-th time block

1) It is supposed that there are two types of service (service 1 and service 2).

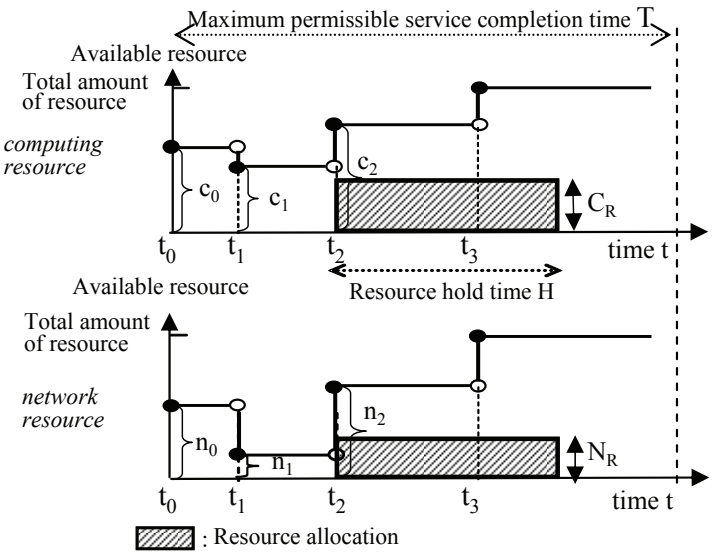
2) If requests for a service to which the amount of key resource allocated is small are rejected for a lack of resource in

i-th time block, the resource amount Mi will be filled up in (i+1)-th time block, regardless of whether requests for a service to which the amount of key resource allocated is large are rejected or not.

3) If no requests for a service to which the amount of key resource allocated is small are rejected for a lack of resource in i-th time block, it is considered that enough resource has been allocated to the service and no action to fill up the imbalance is made in (i+1)-th time block (Mi is set to 0), even if requests for a service to which the amount of key resource allocated is large are rejected.

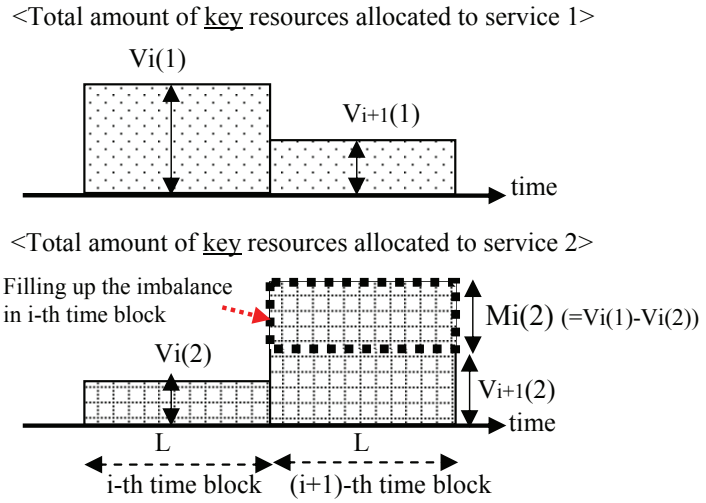
4) When the delayed resource allocation is applied to fill up the imbalance, it is necessary to determine when to start the service. For this purpose, it is proposed to manage the available resource management diagram as illustrated in Figure 3. This diagram is managed per service.

An example of resource allocation is illustrated in Figure 4. The detailed algorithm to fill up the imbalance is illustrated in Figure 5. We call the method which follows the above algorithm in unfair case but follows Method II [4] in the fair case, as '**Method V**' in this paper.



t_0: the time when a request occurs
t_i (i=0～x) : the time when an new resource allocation starts for another request or resources allocated to other requests are released
t_x: the last ti until t_0+T
c_i, n_i: the size of available resource at t_i
C_R:Size of required computing resource
N_R:Size of required network resource

Figure 3. Available resource management diagram (managed per service)



Vi(g):total amount of key resources allocated to service g in i-th time block

Figure 4. Example of resource allocation based on the proposed algorithm to full up the imbalance in the previous time block (the case where G is two)

4. Simulation evaluations

4.1 Conditions

1) The evaluation is performed by a computer simulation using the C language.

2) The simulation model is based on Figure 1 with Y=2. That is, center 1 and center 2 are at different locations. There are both computing resources and access bandwidth in each center.

3) It is supposed that there are two services (service 1 and service 2). The size of required computing resource and that of required network resource for two services follows a Gaussian distribution, and average values are given by C and N respectively. (variance is 1.0) The generation interval of requests of both services follows an exponential distribution.

4) The period of resource hold time H is assumed to be constant. In addition, maximum permissible service completion time T is applied to the requests for filling up the imbalance. T is assumed to be constant in this paper.

5) The request generation pattern is given by {C=i1,N=j1;C=i2,N=j2;…;C=in,N=jn} which means that n requests will be occurred repeatedly.

4.2 Simulation results and analyses

Figure 6 shows simulation results. The request generating pattern of service 1 and service 2 is indicated as {C=x, N=x}, and {C=z*x, N=z*x}, respectively. z is the ratio of the size of service 2 request as opposed to the size of service 1 request. It is clear from Figure 6 that compared with the conventional method (Method II), which does not take fairness into consideration, Method V has a small value of F and it turns out that fairness is securable. The difference of F becoming larger as z becomes large.

5. Conclusions

This paper has discussed principles and a measure for evaluating fair allocation among multiple services, assuming both computing and network resources are required to be allocated simultaneously to each service request and there are two types of service. This paper has proposed a fair joint multiple resource allocation method, Method V, which tries to equalize the total amount of key resources allocated to each service in each time block. Key resource is the resource type for which the requested amount is the largest proportion of the maximum resource of that type. Key resource can change over time and can be different for each service.

It has been demonstrated by simulation evaluations that

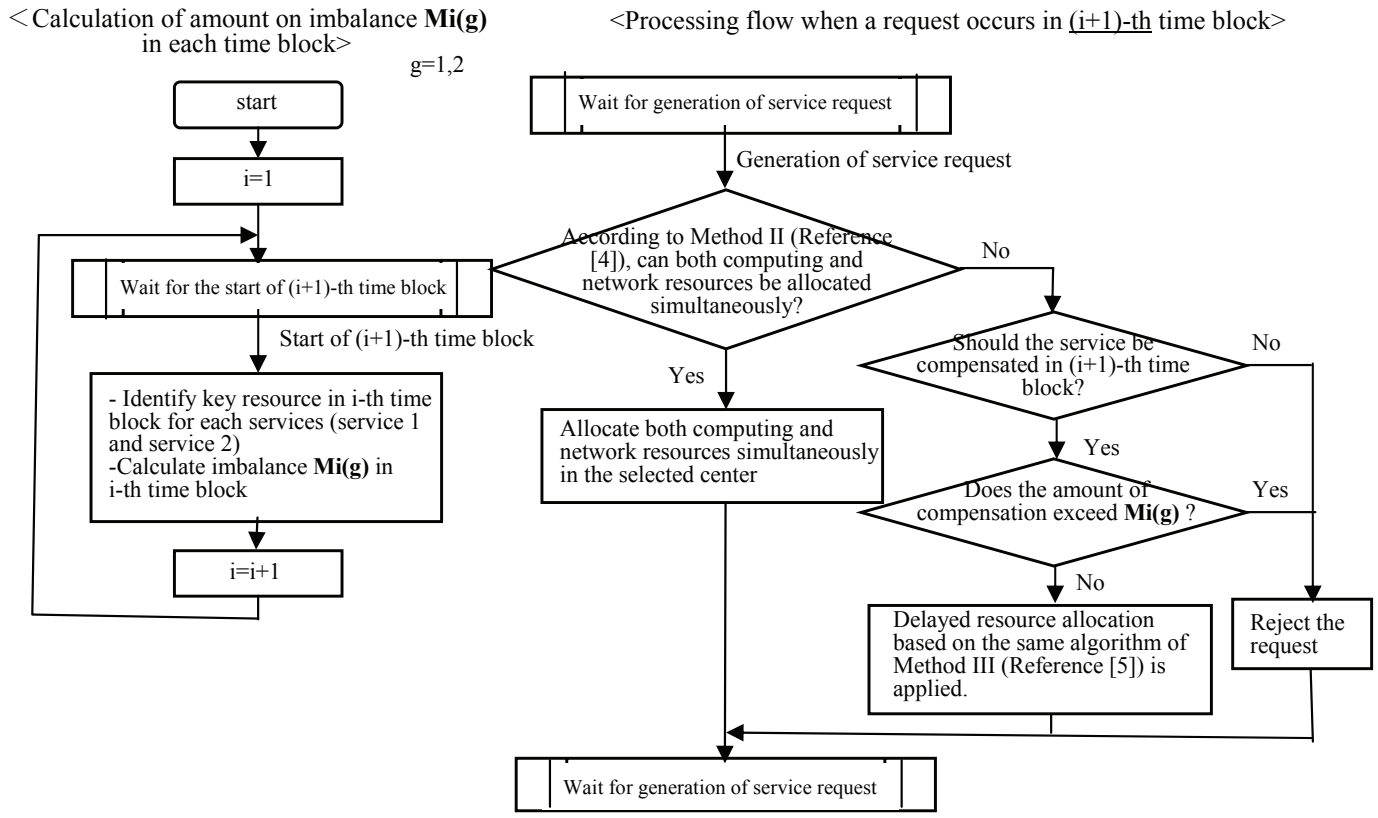
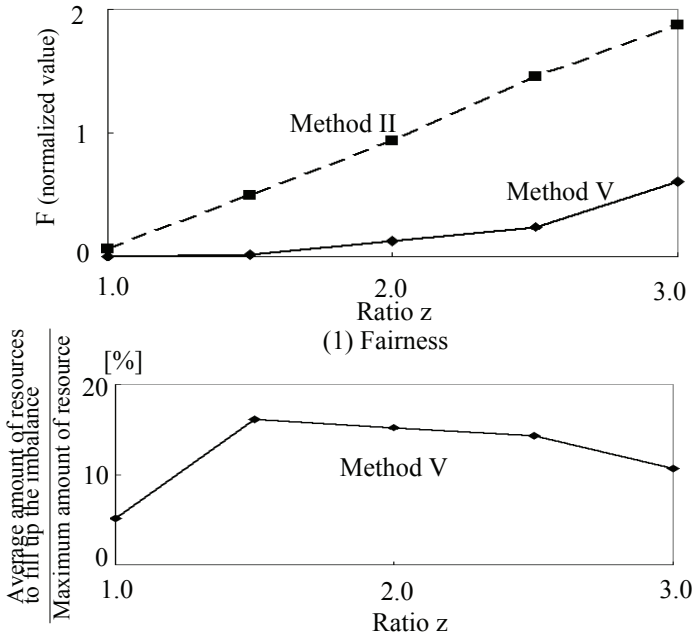


Figure 5. The algorithm to fill up the imbalance in the previous time block (the case where G is two)



(2) Amount of resources which are used to fill up the imbalance

<Conditions>
$C_{max1}=C_{max2}=20$, $N_{max1}=N_{max2}=20$, H=6, T=7.2, L=10*H
{C=2,N=2} :service 1, {C=2*z,N=2*z} : service 2

Figure 6. Simulation results

Method V enables the fair allocation among multiple services, compared with the conventional method which does not consider the fair allocation.

References

[1] M.Poikselka, G.Mayer, H.Khartabil and A.Niemi,"THE IMS –IP Multimedia concepts and services in the mobile domain", John Wiley & Sons, Ltd., 2004.
[2] N.Morita and H.Imanaka,"Introduction to the functional architecture of NGN", IEICE Trans. Commun., Vol.E90-B, No.5, May 2007.
[3] A.Takefusa, H.Casanova, S.Matsuoka, and F.Berman, "A study of deadline scheduling for client-server systems on the computational grid", Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing.
[4] S.Tsumura and S.Kuribayashi, "Simultaneous allocation of multiple resources for computer communications networks", APCC2006 (2006-8).
[5] S.Tsumura and S.Kuribayashi, "Delayed resource allocation method for a joint multiple resource management system", APCC2007 (2007.10)
[6] K.Hatakeyama and S.Kuribayashi, "Proposed congestion control method for all-IP networks including NGN", ICACT2008 (2008.2)
[7] M.Shreedhar and G.Varghese, "Efficient fair queuing using deficit round robin", IEEE/ACM Transactions on Networking, vol.4,No.3, June 1996.
[8] Z. Cao and E. W. Zegura, "Utility max-min: An application-oriented bandwidth allocation scheme," in Proc. IEEE INFOCOM, New York, Mar. 1999, pp.793–801.
[9] Peter Marbach, "Priority Service and Max-Min Fairness", IEEE/ACM Trans. on Networking, Vol. 11, No. 5, Oct. 2003.
[10] P.Yue, Z.Liu and Z.Zhang, "Fair bandwidth allocation for responsive and unresponsive flows using approximate fairness dropping scheme", IEICE Trans. Commun., Vol.E89-B, No.4, April 2006.