A DYNAMIC POLICY FOR RESOURCE MANAGEMENT IN NEXT GENERATION NETWORKS

Mahmoud Pirhadi[1], *Mojtaba Yaghobi Waskasi*[2], *Seyed Mostafa Safavi Hemami*[3]

[1]Islamic Azad University, Karaj Branch, Karaj, Iran, m.pirhadi@kiau.ac.ir
[2]University of Tehran, Tehran, Iran, mojy1979@yahoo.com
[3]Amirkabir University of Technology, Tehran, Iran, msafavi@aut.ac.ir

ABSTRACT

Quality of Service (QoS) control and resource management in Next Generation Networks (NGN) is provided by a particular architecture called RACF (Resource and Admission Control Function) which has been introduced by ITU. In this paper, we propose a dynamic and scalable policy for RACF to manage the bandwidth reservation for arriving calls. This threshold based method will decrease processing load compared to the current RACF per call resource allocation method while keeping the network utilization in a reasonable range. The basic concept lies in aggregating calls into bandwidth-provisioned trunks and admitting calls on trunks only within the limit of their bandwidth. The performance of the algorithm has been evaluated through numerical computations.

1. INTRODUCTION

End-to-end QoS control and resource management is gaining increasing interest in Next Generation Networks (NGN). The main goal of NGN is to migrate from all existing circuit-based networks to packet switching networks. The main concern of this migration is various QoS requirements of different existing technologies [1, 2]. One of the missing pieces is a scalable and efficient integration with traditional IP QoS technologies, e.g., differentiated services (DiffServ) and integrated services (IntServ). Both would increase and guarantee the quality of calls over the IP-based Networks. Finding a solution to run services such as voice traffic over the NGN with a quality comparable to PSTN would be a big step towards the convergence of voice and data networks. However, a mediator is needed between session controllers (e.g., SIP proxy servers and call servers) and QoS mechanisms in transport layer [3].

These have made standardization organizations such as ITU and ETSI [4] to propose models and architectures provisioning resource management in NGN networks. An architecture which has been introduced by ITU-T is called RACF (Resource and Admission Control Function) [5, 6]. Figure 1 illustrates a simplified model showing the role of RACF in NGN architecture.

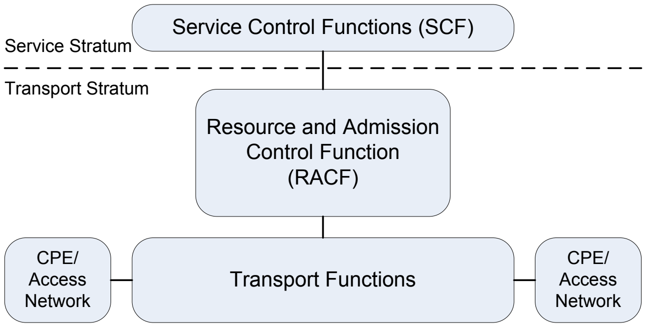

Figure 1. RACF architecture in NGN

RACF acts as the mediator between Service Control Functions (SCF) and transport functions for QoS-related transport in an NGN architecture. One of the main functionalities of RACF is to make decisions according to defined policies based on resource availability in transport layer [7].

The SCF represents the functional entities of NGN service layer which can request QoS resource and admission control for media flows of a given. The RACF interacts with the SCF and transport functions for the applications that require resource control in the transport layer. SIP-based calls are examples of such applications [8].

It is clear that the number of service requests per unit time that a RACF can handle and process is limited, because it needs to configure transport functions (i.e., routers) in an appropriate way [6]. These functions might be very time consuming, and so the number of dynamic reconfigurations of the QoS-enabled network elements should be reduced. This can be done by defining and reserving a set of bandwidth-provisioned trunks between all the possible edge nodes in the administrative domain controlled by the RACF and aggregate calls into these trunks.

An important issue is that while we reduce the processing load we have to keep the overall network utilization in a reasonable range. In this paper, we are going to propose an algorithm for this purpose.

The rest of the paper is organized as follows. The next section briefly overviews the main idea of dynamic

978-1-4244-5711-3/10/$26.00 ©2010 IEEE

bandwidth control. The proposed algorithm is modeled and analyzed in section 3. Section 4 presents the numerical results and finally, Section 5 concludes the paper.

2. DYNAMIC BANDWIDTH CONTROL OF TRUNKS: OVERVIEW AND BACKGROUND

As mentioned before, the main idea is to aggregate calls into bandwidth-reserved trunks and to admit calls on these trunks only within the limit of their bandwidth. When a call and resource reservation request is received by the RACF, it extracts the required bandwidth for the call and checks whether the involved trunk has still enough available bandwidth. If yes, the RACF permits the call establishment and updates its internal state of resource allocation within the trunks. If not, the lack of resources is signaled back to the caller, and the call is dropped. Note that differently from the simple per call reservation approach, the RACF does not need to interact with the transport functions to reconfigure them and decide whether to admit the call.

The available bandwidth of a trunk varies with time due to the concurrency of its established calls. As regards the trunk sizing, the approach can work with static and fixed choices, for example, based on conservative estimations of call arrivals. Because of this, bandwidths of very few trunks are effectively employed. This is very inefficient as the transmission capacity of network cannot be shared among calls. This decreases transmission efficiency.

Our suggestion is dynamic resizing of the trunks which is performed by RACF. When the connections on one trunk increase, remaining link capacity is assigned to the busy trunk. Thus, transmission efficiency is improved because each trunk in the link is well utilized.

In a simple per call approach, a trunk bandwidth could be increased or decreased each time a call arrives or tears down, then trunk's bandwidth would be always equal to the bandwidth in use and we would have 100% trunk utilization. But this approach will increases the processing load on RACF. At the opposite extreme, trunks may be statically provisioned, but this causes inefficient use of bandwidth.

The advantage in reduced processing load is expected to be maintained by changing the trunk bandwidth less frequently than call setup and clearance. This can be done allowing some spare capacity on the trunks, so that the RACF requests a trunk resizing only when this spare capacity exceeds or falls below a given threshold. Intuitively, the larger this over-sizing the lower the average trunk utilization, and also the lower the rate of trunk resizes.

In the recent literature, some papers dealt with similar problems. The pioneering work in this area is the one performed in [9] on an ATM network. This paper's approach is a threshold-based method described in [9], however, the idea has been generalized and adapted to IP-based networks and the analysis method is quite different.

3. ANALYSIS OF THE ALGORITHM

The basic algorithm for dynamic control of trunks bandwidth which has to be performed by RACF is as follows:

- Request a trunk bandwidth increase by a specified step if current bandwidth is insufficient to admit the new call.
- If the increase is allowed, then increase the bandwidth and allow the SCF to set up the call; otherwise, keep the current bandwidth and reject the call request.
- Decrease the trunk bandwidth by a specified step if possible, according to the trunk utilization condition.

The algorithm is valuable only if the processing load decreases while keeping the transmission utilization in the acceptable range.

3.1 Model

A single link in the network is analyzed as a multidimensional loss model. All of the Calls need the same bandwidth and arrive in a Poisson process of rate λ_i and they have exponentially distributed holding times with mean $1/\mu_i$. The channel used by a call is released at the end of the holding time. If an arriving call can not be established due to the lack of resources (free channels) in the link then it is blocked and lost without further affecting the system. The parameters used in this analysis are defined as follows.

k the number of trunks on the link

C the total link capacity normalized to the call bandwidth

T_i the i^{th} trunk ($1 \le i \le k$)

a_i offered traffic to T_i in erlangs

$a = (a_1, a_2, ..., a_k)$ the traffic vector

The effect of algorithm depends on bandwidth change step size. It is the bulk of bandwidth which should be added to or subtracted from the trunk size upon RACF request. When all the channels of the trunk are occupied, it should be increased to the next threshold by a new call arrival. $S=(S_1, S_2, ..., S_k)$ is the step size vector in which S_i indicates the step size of trunk T_i.

Now the $W_i(n_i)$ function is defined which indicates the normalized bandwidth of the i^{th} trunk handling n_i concurrent calls. $W_i(n_i)$ is an integer multiple of step size, i.e., $W_i(n_i)=l_iS_i$ and is defined as follows.

$$l_iS_i < n_i \le (l_i+1)S_i \;\Rightarrow\; W_i(n_i) = (l_i+1)S_i \qquad (1)$$

The basic idea is illustrated in Figure 2.

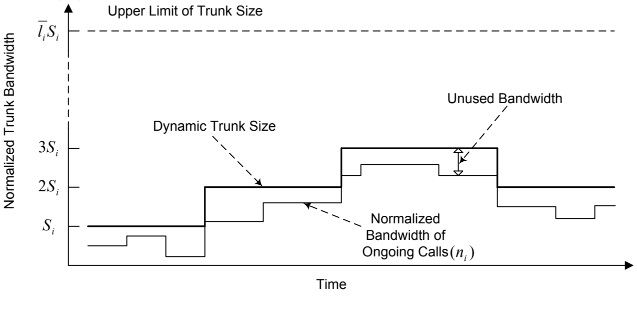
$W_i(n_i)$
Upper Limit of Trunk Size
$\bar{l}_iS_i$
Normalized Trunk Bandwidth
Unused Bandwidth
Dynamic Trunk Size
$3S_i$
$2S_i$
S_i
Normalized Bandwidth of Ongoing Calls(n_i)
Time

Figure 2. Trunk bandwidth function

The transmission efficiency is defined as the ratio of the carried traffic and the minimum link capacity needed to achieve a specified call blocking probability. It is formulated as follows.

$$\eta = \frac{\sum_{i=1}^{k} a_i (1 - P_{b_i})}{C} \tag{2}$$

where P_{b_i} is the blocking probability for calls arrived at i^{th} trunk. The main challenge is deriving P_{b_i} for all trunks. Obviously, the *resource sharing policy* has a strong effect on the blocking experienced by calls. Let define the system state vector *n* as

$$n = (n_1, n_2, ..., n_k)$$

where n_i indicates the number of concurrent calls in T_i. The set of allowable states (Ω) depends on resource sharing policy. The general condition for *complete sharing* policy is

$$n \in \Omega \Rightarrow n_i \ge 0 \ ; \ i=1,\ ...,\ k \quad \text{and} \quad 0 \le \sum_{i=1}^{k} n_i \le C \tag{3}$$

while our model imposes another constriction on Ω as follows

$$0 \le \sum_{i=1}^{k} W_i(n_i) \le C \tag{4}$$

or

$$0 \le \sum_{i=1}^{k} l_i S_i \le C \qquad l_i = 0,1,...,\overline{l_i}$$

In other words, it is a *complete sharing* (none of trunks have exclusive use of a portion of the link) with *dynamic trunk sizing* policy. Thus, Ω is defined as follows.

$$\Omega = \left\{ n : 0 \le \sum_{i=1}^{k} n_i \le C \ , \ 0 \le \sum_{i=1}^{k} W_i(n_i) \le \mathrm{C} \right\} \tag{5}$$

If *P(•)* denotes the state distribution, then

$$P_{b_i} = \sum_{n \in \Omega_i'} P(n) \tag{6}$$

where Ω'_i is the subset of states in Ω in which another request can not be accepted in trunk *i*. Then the steady-state probability mass function has a simple *product-form* as follows [10, 11]

$$P(n) = \prod_{i=1}^{k} \frac{a_i^{\,n_i}}{n_i!} . G^{-1}(\Omega) \tag{7}$$

with the normalization constant

$$G(\Omega) = \sum_{n \in \Omega} \prod_{i=1}^{k} \frac{a_i^{\,n_i}}{n_i!} \tag{8}$$

In other words

$$P_{b_i} = \frac{G(\Omega_i')}{G(\Omega)} = 1 - \frac{G(\Omega - \Omega_i')}{G(\Omega)} \tag{9}$$

The normalization constant $G(\Omega)$ can be directly expressed as a *k*-fold nested sum

$$G(\Omega) = \sum_{n_1=0}^{\overline{n_1}} \frac{a_1^{\,n_1}}{n_1!} \sum_{n_2=0}^{\overline{n_2}} \frac{a_2^{\,n_2}}{n_2!} ... \sum_{n_k=0}^{\overline{n_k}} \frac{a_k^{\,n_k}}{n_k!} \tag{10}$$

For our sharing policy, the upper limit of summation $\overline{n_i}$ depends on $\overline{l_i}$ which is derived as follows

$$\overline{l_i} = \left\lfloor \frac{C - \sum_{m=1}^{i-1} l_m S_m}{S_i} \right\rfloor \tag{11}$$

and

$$\overline{n_i} = \overline{l_i} S_i \tag{12}$$

3.2 Numerical Example

As a trivial example, suppose that *k=2* (i.e., two trunks in the link), *(S₁, S₂)=(2, 3)* and *C=13*. Then

$$\overline{l_1} = \left\lfloor \frac{13-0}{2} \right\rfloor = 6 \rightarrow \overline{n_1} = \overline{l_1}.S_1 = 6 \times 2 = 12$$

and

$$n_1 = (0,1,2,3,4,5,6,7,8,9,10,11,12)$$

$$\overline{n_2} = (12,9,9,9,9,6,6,3,3,3,3,0,0)$$

The state space Ω is illustrated in Figure 3.

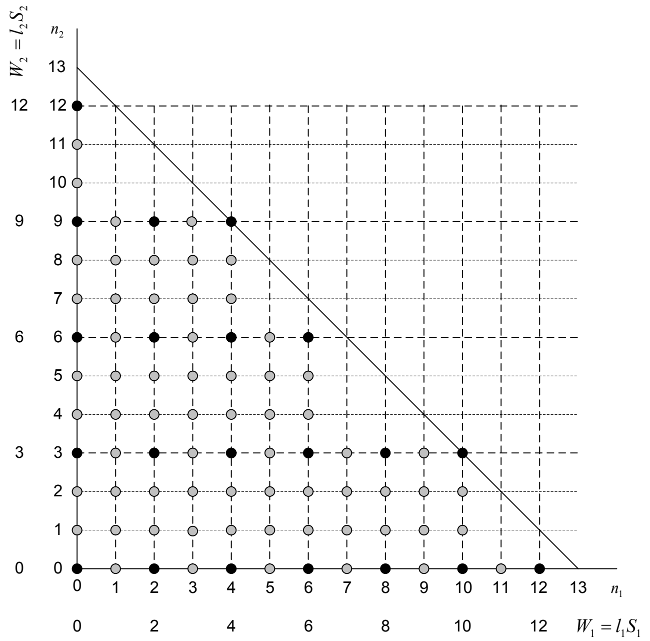

Figure 3. Allowable states in dynamic trunk resizing

3.3 Processing Load

In order to evaluate the load on RACF, consider again the state space Ω. In per call resizing approach, whether the trunk size is changed or not is decided at every call setup request. This places a similar load on RACF as in the case without bandwidth-provisioned trunks. However, in the dynamic bandwidth control approach, the trunk resizing is not necessary for all states. For those states in which

$$l_i S_i < n_i < (l_i + 1) S_i$$

a new call arrived at T_i is admitted and established without any reconfiguration processing i.e., the RACF is not required to perform the trunk resizing procedure. The trunk resizing occurs only in those states in which the trunk is full ($l_i S_i$ channels are occupied) and a new call is arrived at T_i.

Thus, the processing load can be interpreted and derived in the following way. Suppose that an average of λ_i calls arrive at T_i during time unit, and let P_i be the state probability that the T_i is full i.e., the unused bandwidth of T_i is zero. Then, the mean number of calls which causes the bandwidth increase request is $\lambda_i P_i$ which can represent the mass of load on the RACF. Since P_i is probability that n_i is equal to $l_i S_i$, the processing load for T_i, denoted by L_i is expressed as follows.

$$L_i = \lambda_i \cdot \sum_{n_1=0}^{\overline{n_1}} \frac{a_1^{n_1}}{n_1!} \cdots \sum_{l_i=0}^{\overline{l_i}} \frac{a_i^{(l_i S_i)}}{(l_i S_i)!} \cdots \sum_{n_k=0}^{\overline{n_k}} \frac{a_k^{n_k}}{n_k!} \cdot G^{-1}(\Omega) \qquad (14)$$

In other words we are calculating the probability distributions for those states in Ω which n_i gets the discrete values equal to $l_i S_i$. In the above numerical example these states for L_1 will be

$$n_1 = (0,2,4,6,8,10,12)$$
$$\overline{n_2} = (12,9,9,6,3,3,0)$$

The processing load for the link is obtained from the L_i's for each trunk.

4. NUMERICAL RESULTS

In this section the simulation results of the model is presented using a code written in MATLAB. Figure 4 shows the link transmission efficiency versus offered traffic for different values of trunk step sizes. The number of trunks in the link is $k=3$, the step size S is the same for all trunks, the offered traffic is $a_i=a$ (in erlangs), and the maximum available link bandwidth is $C=100$. For comparison, the figure shows the transmission efficiency for simple complete sharing model (i.e., $S=1$).

It can be seen that the link efficiency is improved for smaller values of S. In addition, there is no significant decrease in the link efficiency, especially for small S, compared to the simple complete sharing model.

Figure 5 shows the processing load versus offered traffic for the same conditions. As it is shown, there is a considerable decrease in processing load of RACF using trunks compared to the simple complete sharing policy without trunks. The greater the step size, the more decrease in processing load.

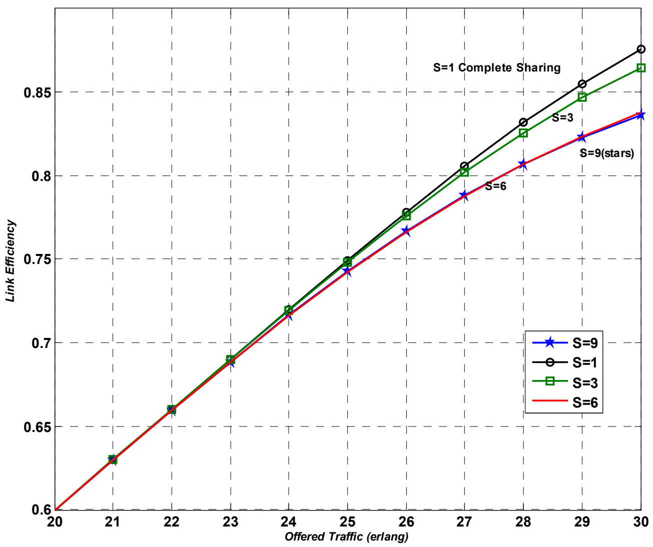

Figure 4. Link bandwidth efficiency for different trunk step sizes

Since there is negligible decrease in bandwidth efficiency while we have considerable improve in processing load, we can select an appropriate step size to achieve the best network performance. Selection of the step size S is an important decision in implementing bandwidth control. The above results show that there is a trade-off between the transmission efficiency and the processing load for the step size.

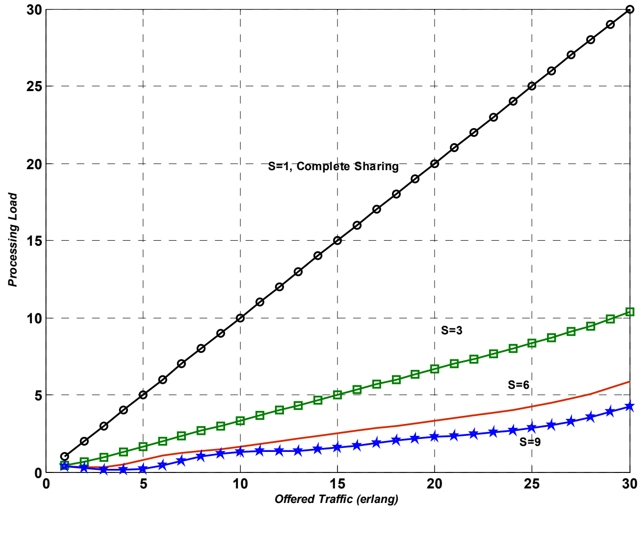

Figure 5. Proccessing load for different step sizes

Figure 6 illustrates the link efficiency and the processing load versus S where $a=25$ erlangs. An optimal value for the step size can be obtained by employing these relationships when the efficiency-processing ratio is provided.

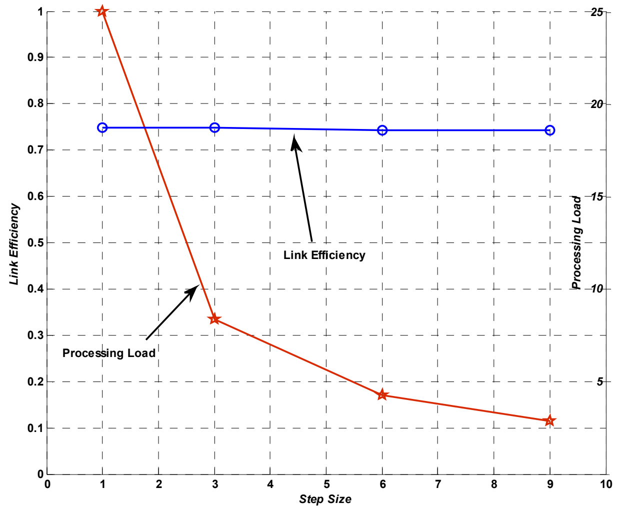

Figure 6. Link efficiency and proccessing load versus step size

The blocking probability versus offered traffic for different step sizes is shown in Figure 7. For example, assuming that maximum allowed blocking probability is $P_b=0.01$ for all trunks, the maximum offered traffic can be determined using this figure.

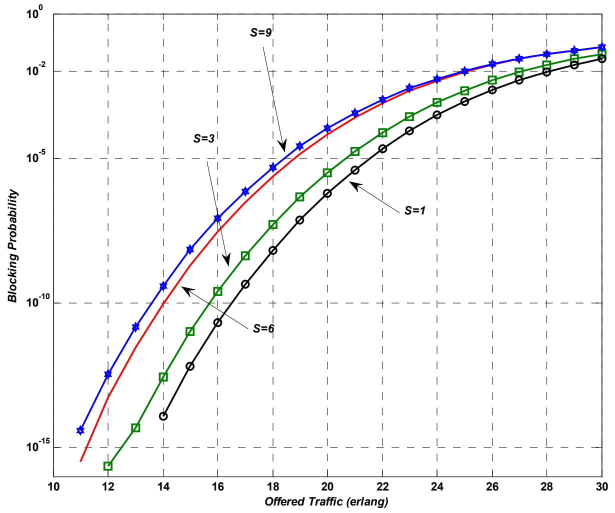

Figure 7. Blocking probability for different step sizes

5. CONCLUSION

This paper proposed a threshold based dynamic bandwidth control policy for RACF in NGN architecture to reduce the reservation processing load while keeping the network transmission efficiency in an acceptable range. The simulation results show that the proposed algorithm could achieve this goal and the optimal point can be found. The proposed model also can be used to determine the required link capacity, given the maximum allowed blocking probability and the desired bandwidth efficiency. Future work is intended to generalize the model for multi-class multi-rate traffics and to propose an efficient recursive algorithm to decrease the computation time of the model.

6. ACKNOWLEDGMENT

We would like to thank Professor Glen Takahara from the department of mathematics at Queen’s university for his valuable comments that helped to improve the paper.

7. REFERENCES

[1] ITU-T Rec. Y.2001, "General overview of NGN", 2004

[2] ITU-T Rec. Y.2201, "Requirements and capabilities for ITU-T NGN", 2009

[3] Rosenberg J. *et al.*, "SIP: Session Initiation Protocol", RFC 3261, 2002

[4] ETSI ES 282 003 v1.1.1, "Resource and Admission Control Sub-system (RACS); Functional Architecture", 2006

[5] ITU Rec. Y. 2111, "Resource and admission control functions in Next Generation Networks". (Release 2), 2007

[6] ITU-T Rec. Y.2175, "Centralized RACF architecture for MPLS core networks", 2008

[7] M. Safavi Hemami., M. Pirhadi and A. Iravani Tabrizipoor, "Analysis and Optimization of Resource Control Schemes in Next Generation Networks", K.INGN08, Geneva, 2008

[8] M. Pirhadi, M. Safavi Hemami and A. Khademzadeh, "Resource and Admission Control Architecture and QoS Signaling Scenarios in Next Generation Networks", World Applied Siences Journal, 2010

[9] Ohta S, Sato KI., "Dynamic bandwidth control of the virtual path in an asynchronous transfer mode network", IEEE Transactions on Communications, 1992

[10] G. L. Choudhury, K. K. Leung, and W. Whitt, "An algorithm to compute blocking probabilities in multi-rate multi-class multi-resource loss models", Advances in Applied Probability, Vol.27, No.4, pp.1104-1143, 1995

[11] ITU-R Document 8F/434-E, "Refined calculation method of multi-Dimensional Erlang-B Formula", 2005