

# Performance Evaluation of Resource Reservation Policies for Rate-Controlled Earliest-Deadline-First Scheduling in Multi-Service Packet Networks

Khaled M. F. Elsayed  
khaled@ieee.org

Amr Saad Ayad  
asaad@salec.com

Mahmoud T. El-Hadidi  
hadidi@frcu.eun.eg

*Department of Electronics and Communications Engineering  
Faculty of Engineering, Cairo University, Giza, Egypt 12613*

**Keywords:** Resource Reservation, Rate-Controlled Earliest-Deadline First Scheduling, Deterministic Delay Bounds, Call Admission Control, Multi-Service Networks.

**Abstract:** The paper addresses the issue of reserving resources at packet switches along the path of flows requiring a deterministic bound on end-to-end delay. The switches are assumed to schedule outgoing packets using the Rate-Controlled Earliest-Deadline-First (RC-EDF) scheduling discipline. EDF is known to be an optimal scheduling discipline for deterministic delay services in the single scheduler case. We propose a number of static and dynamic reservation policies for mapping the end-to-end delay requirement of a flow into local delay deadlines to be reserved at each scheduler. These policies are based on non-even resource reservation where the resources reserved depend on the capacities and loading at each node in the network. We define and prove the optimality of a certain non-even policy for the case of a single path network with homogenous static traffic. We present extensive simulation results for different scenarios which show that dynamic non-even resource reservation provides superior performance when compared to simple policies such as even dividing of end-to-end delay among the schedulers.

## 1. INTRODUCTION

One of the main promises of multi-service packet networks is to provide applications with Quality-of-Service (QoS) guarantees, such as maximum packet delay and packet loss ratio. Handling the variety in QoS requirements of different applications requires the network to use a mechanism for serving packets from different applications according to their granted QoS level. Many scheduling disciplines have been proposed in the literature to implement such mechanism (see [3, 6, 7, 11, 13]). Each scheduling discipline requires algorithms for performing call admission control (CAC) and resource reservation. This paper proposes such algorithms for the case of flows requiring a hard (deterministic) bound on end-to-end delay where RC-EDF service discipline (as proposed in [6]) is used. The main focus of this paper is the resource reservation problem and is not an attempt to provide a new scheduling discipline. The work presented here is based on the use of the general EDF schedulability condition given in [9], and the EDF schedulability condition for token-bucket-shaped traffic given in [5].

We address the problem of how to map the end-to-end delay requirement of a flow into a local resource requirement to be reserved at each scheduler along the flow's path. A commonly used resource reservation strategy is to equally divide the end-to-end delay requirement among the schedulers serving the flow so that each scheduler reserves the same amount of resources regardless of its capacity (i.e. the speed of its outgoing links or its loading state). We propose a number of non-even reservation policies that take the scheduler link capacity and its loading state into consideration and present some results of the simulations used to evaluate the performance of such policies.

There have been several proposals in the literature to handle the problem of resource reservation and call admission control for packet networks providing deterministic service. In an early work [16], it is shown that if reservation is based on the peak rate of each connection, the network will be under-utilized by guaranteed service traffic when the traffic is bursty. It is shown that local deterministic delay bounds can be guaranteed over a link for bursty traffic even when the sum of the peak rates of all the connections is greater than the link speed. This allows a multi-fold increase in the number of admitted connections when the traffic is bursty. The results can be efficiently extended from a single switch to a network of arbitrary topology by using rate-controlled service disciplines at the switches.

In [11], development of a nodal metric defined as the "relative gain ratio" that predicts the relative performance of QoS allocation policies in a network setting is presented. Computation of the relative gain ratio and direct evaluation of allocation policy performance for two simple network models are done. It is

found, however, that with the packet loss probability as the QoS metric, there is little difference in the performance of allocation policies in the regime of applications with low loss requirements. However, for applications which tolerate large packet loss or alternate QoS metrics, QoS allocation policies differ significantly in their performance. In comparison with our approach, we focus here on the case of applications requesting deterministic end-to-end delay bounds where the network implements RC-EDF scheduling.

Ref. [8] discusses the problem of resource partitioning which is useful for a number of applications, including the creation of virtual private subnetworks and of mechanisms for advance reservation of real-time network services. The paper gives the results of using admission control tests for resource partitioned servers for four representative scheduling disciplines, First-In-First-Out (FIFO), Packet-by-packet Generalized Processor Sharing (PGPS), Rate-Controlled Static Priority (RCSP) and EDF. The simulations confirm the intuition that resource fragmentation losses due to resource partitioning are small and that resource partitioning reduces the admission control computation overhead. An interesting result from the simulation experiments is that, under circumstances that arise naturally in multi-party communication scenarios, resource partitioning results in higher overall connection acceptance rate.

In a highly related work, Ref. [4] presents a general framework for admission control and resource reservation for multicast sessions. Within this framework, efficient and practical algorithms that aim to efficiently utilize network resources are developed. The problem of admission control is decomposed into several subproblems that include: the division of end-to-end QoS requirements into local QoS requirements, the mapping of local QoS requirements into resource requirements, and the reclaiming of the resources allocated in excess. Our work presents a larger set of allocation policies and extensive simulation results for various topologies and traffic conditions.

Recently, Ref. [10] considers optimal partitioning of QoS requirements for unicast paths and multicast trees. The problem is mapped to a constrained optimization problem and heuristics are used to find an approximate solution for the optimization problem. While the approach is very general, the problem with it is that the assumption of the existence of a cost function. The cost function is assumed to be known and is static in nature. This is not a realistic assumption since the cost function should depend on the current network loading state and the QoS objectives of the flows passing through a particular switch.

In Ref. [1], a study of resource allocation for PGPS based scheduling in packetized voice networks is presented. Also, Refs. [2, 12] study the problem of resource allocation and CAC in a network with PGPS scheduling. Various non-uniform resource allocation policies were handled and their qualitative performance compared. This paper handles the RC-EDF discipline as compared to the PGPS handled in [2, 12].

The rest of this paper is organized as follows: Section 2 presents the system model and details of the EDF scheduling discipline and CAC operation. Section 3 proposes a number of resource reservation policies that are applicable when using RC-EDF schedulers. Section 4 presents simulation results, and section 5 concludes the paper.

## 2. SYSTEM MODEL AND DETAILS OF THE RC-EDF SCHEDULING DISCIPLINE

We consider a network with arbitrary topology represented by a graph  $G=(S, E)$  where  $S$  is the set of network nodes containing the schedulers and  $E$  is the set of edges connecting these nodes. An edge  $e \in E$  has an associated capacity  $C^e$  expressed in bits/sec. The network handles flows with arbitrary QoS requirements; however, we are only concerned with the class of delay-guaranteed flows requesting a deterministic end-to-end delay bound. Such flows are usually assigned a fixed portion of bandwidth. To optimize the resource usage, this reserved bandwidth may be shared with best-effort traffic when no packets from the class of delay-guaranteed flows exist. However, packets from delay-guaranteed flows can preempt other packets. Under these assumptions, we can isolate the delay-guaranteed flows from other flows and analyse them separately.

Each flow  $f$  is shaped by a token bucket traffic shaper and its traffic is specified by either the  $(\sigma_f, \rho_f)$  descriptor or the  $(\sigma_f, \rho_f, c_f)$  descriptor, where  $\sigma_f$  is the maximum burst size (in the appropriate units of bits or cells in ATM case),  $\rho_f$  is the sustained rate (in bits/sec or cells/sec), and  $c_f$  is the peak rate (in bits/sec or cells/sec). For the  $(\sigma, \rho, c)$  model, we further define the parameter  $a = \sigma / c$ , which is the time period the traffic can burst with rate  $c$ . For the  $(\sigma, \rho)$  model, we have  $c \rightarrow \infty, a \rightarrow 0$ , but  $ac \rightarrow \sigma$ .

We assume that the scheduler at each node has full information about the traffic descriptor of all passing flows. After the flow  $f$  passes through the first node, the traffic is no longer specified by  $(\sigma_f, \rho_f)$  or  $(\sigma_f, \rho_f, c_f)$ . We assume that the schedulers at each node are rate-controlling schedulers, i.e. they reshape the traffic of a certain flow  $f$  to its original form. Since the schedulers use the EDF scheduling discipline and traffic reshaping is performed at each node, we have a *Rate-Controlled EDF* scheduling discipline.

The operation of an EDF scheduler is described as follows: a deadline is assigned to each newly arriving packet from flow  $f$ . The deadline is computed as the sum of the arrival time of the packet and the local delay bound reserved for flow  $f$  at this scheduler. The scheduler serves packets in the ascending order of their deadlines.

Maintaining the delay guarantee made to flow  $f$  is equivalent to having all the packets belonging to it transmitted completely before their assigned deadlines. Consequently, all packets from flow  $f$  do not get delayed beyond the delay bound reserved for flow  $f$  at this scheduler. We denote the case in which a packet misses its deadline, i.e. not transmitted completely before its deadline, as a case of *violation*.

The conditions under which a single EDF scheduler operates without violations are:

1- The stability condition: This is a condition that must hold true for all work-conserving disciplines in general. If the condition does not hold true, a scheduler with finite buffers will always overflow and drop packets. For a scheduler  $k$  with capacity  $C^k$  (which is the data rate in bits/sec of the link following the scheduler that is reserved for delay-guaranteed flows), the stability condition is:

$$\sum_{j=1}^{N_k} \rho_j \leq C^k \quad (1)$$

where  $\rho_j$  is the average rate (in bits per second) of the traffic source of flow  $j$  and  $N_k$  is the number of flows that are being served by scheduler  $k$ .

2- The schedulability condition: The schedulability condition guarantees that a scheduler will not make violations and will, therefore, honour the QoS commitment made during the flow set-up phase. The form of the schedulability condition differs, in general, for each scheduling discipline. If the schedulability condition is sufficient but not necessary, then there is a possibility of under-utilizing the scheduler since a violation of the sufficient conditions does not necessarily mean that a given set of flows is not schedulable. On the other hand, a schedulability condition that is both necessary and sufficient guarantees that there is no under-utilization of the scheduler's resources.

## 2.1 Schedulability conditions for token-bucket traffic

Here, we use the results presented in [5] which applies Theorem 1 in [9] to deduce the schedulability conditions for token-bucket traffic models. The analysis in [5] assumes the use of a preemptive EDF scheduler which is equivalent to the use of negligible packet transmission time (which is typical in the case of networks with small packet size and high speed links, e.g. ATM networks). For negligible packet transmission time, [5] defines the function  $F(t)$  as:

$$F(t) = Ct - \sum_{j=1}^N A_j^*(t - d_j) \quad (2)$$

where  $C$  is the data rate of the link serving the EDF scheduler in bits per second,  $N$  is number of flows passing through the scheduler,  $d_j$  is the delay bound reserved for flow  $j$  in seconds, and  $A_j^*(t)$ , in bits, is the traffic-constraint function on the traffic arrivals from flow  $j$  up to time  $(t)$ . In [5], it is shown that the *necessary and sufficient* schedulability condition of an EDF scheduler is equivalent to verifying that:

$$F(t) \geq 0 \quad \forall t \geq 0 \quad (3)$$

We will be dealing with the case in which  $A_j^*(t)$  represents token-bucket traffic models. For the  $(\sigma, \rho, c)$  model the function  $A^*(t)$  is given by:

$$A^*(t) = \begin{cases} ct, & 0 \leq t \leq a \\ \sigma + \rho(t-a), & t > a \\ 0, & t < 0 \end{cases} \quad (4)$$

whereas for the  $(\sigma, \rho)$  model it is given by:

$$A^*(t) = \begin{cases} \sigma + \rho t, & t \geq 0 \\ 0, & t < 0 \end{cases} \quad (5)$$

## 2.2 CAC for bounded delay service in multi-service networks

For bounded delay service, the application of the CAC algorithm on a flow's path requires the computation of the minimum delay that each scheduler along the path of the flow can guarantee to this new flow. This allows the CAC algorithm to determine the minimum achievable end-to-end delay bound for this flow, and thus to determine if the network can guarantee the requested delay bound or not. This section starts by describing the assumed CAC operation when accepting a new flow. We then discuss the computation of the minimum delay bound that a RC-EDF scheduler can guarantee to flows with token-bucket traffic model.

The CAC operation proposed in [4, 5] assumes that the flow is established using a setup protocol such as RSVP or ATM Q.2931 signalling. The operation of this protocol proceeds as follows: the calling party wishing to establish a flow  $f$ , sends a SETUP message to the called party, including the flow's traffic characteristics  $(\sigma_f, \rho_f, c_f)$ , and the required end-to-end delay bound  $(D_f)$ .

This message travels over  $K_f$  schedulers belonging to the path  $P_f$  selected for the connection by the routing algorithm in use. At each scheduler  $i$  on  $P_f$ , the minimum delay that a scheduler  $i$  can guarantee to connection  $f$ ,  $d_f^{i*}$ , is computed and added to  $d_f^*$ , which is the cumulative delay sum included in the setup message. If at some scheduler, the cumulative delay exceeds the required delay bound, then the connection cannot be accepted and a RELEASE message is returned to the calling party. Otherwise, the setup message reaches the

last scheduler which checks if  $D_f \geq D_f^*$ , where  $D_f^* = \sum_{i=1}^{K_f} d_f^{i*}$  is the minimum achievable end-to-end delay

for connection  $j$ . If the condition is true, the connection is accepted, and a CONNECT message is then returned on the same path to the calling party, reserving a delay bound  $d_f^k \geq d_f^{k*}$  to connection  $f$  at each

scheduler  $k$  such that  $\sum_{i=1}^{K_f} d_f^i \leq D_f$ . The values of  $d_f^k$  are chosen according to some delay reservation policy

as will be discussed in section 3 (*the method in which the values  $d_f^k$  are computed by a specific reservation policy is the main subject of this paper*).

We assume that only the final scheduler checks the validity of condition  $(D_f \geq D_f^*)$  and makes the irreversible decision of accepting or rejecting the flow.

## 3. THE PROPOSED RESOURCE RESERVATION POLICIES

In this section, we propose policies for dividing the end-to-end delay requirement among the schedulers along the flow's path. Generally, we can classify resource reservation policies into the following categories:

1. *Static policies*: These are policies in which the delay requirement assigned to a scheduler is independent of the loading state of this scheduler or other schedulers on the flow's path.
2. *Dynamic policies*: These are state-dependent policies in which the delay requirement assigned to a scheduler depends, in general, on its own state and possibly on the state of the other schedulers on the flow path.

We first derive an optimal policy in the case of a single-path initially unloaded network when all flows have the same traffic characteristics and delay requirement. The gain that this optimal policy can attain relative to the even allocation of local deadlines is characterized. Afterwards, we define the static and

dynamic policies that we are proposing in this paper. In all cases, it is assumed that the path for a flow has been selected by the routing process and it consists of  $K$  schedulers from the source node to the destination node.

### 3.1 The optimal policy for a single-path network with static homogeneous flows

We consider the case of a single path network where all flows pass through the same path between two nodes as illustrated in *Figure 1*. The path consists of  $K$  schedulers and the capacity of scheduler  $i$  is  $C^i$  bits/sec. We further assume that the network is initially unloaded and that all flows have the same traffic characteristics and delay requirement and have an indefinite holding time. Under these assumptions, we will define the optimal resource reservation policy and obtain the gain achievable by this policy as compared to the even reservation policy.

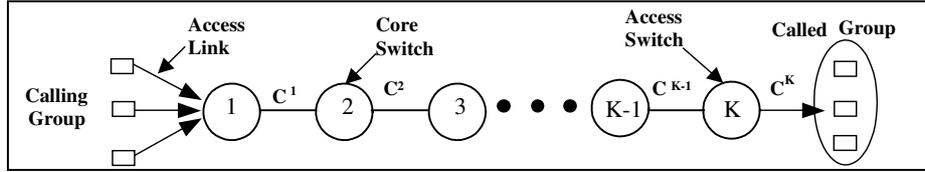


Figure 1. Single-path network model

Let the number of schedulers along the single path network be  $K$  and let us assume that all flows have an end-to-end delay requirement of  $D_f$ . Under the assumption of RC-EDF scheduling, the delay contribution of a scheduler to the end-to-end delay is independent of other schedulers' contribution, and we have:

$$D_f = \sum_{i=1}^K d_f^i \quad (6)$$

Let us denote the maximum number of acceptable flows by a scheduler  $i$  as  $N^i$ . Hence the maximum number of acceptable flows ( $N_{\max}$ ) on the path is given by:

$$N_{\max} = \min_{1 \leq i \leq K} N^i \quad (7)$$

**Definition:** An optimal policy is one that maximizes the number of simultaneous flows acceptable on the flow's path.

*Theorem 1:* For a single path initially unloaded network where all flows have the same traffic characteristics and delay requirement and indefinite holding times, the optimal policy is the one which satisfies the condition that a scheduler  $i$  assigns a delay  $d_f^i$  to flow  $f$  subject to  $D_f = \sum_{i=1}^K d_f^i$  and that achieves:

$$N^i = \text{const} = N_{\max} \quad \forall 1 \leq i \leq K \quad (8)$$

*Proof:* The proof of this theorem is given in Appendix A.

In the sequel, we obtain  $N_{\max}$  for the optimal static policy satisfying (8).

We start by deriving the formula for  $N^i$  for the  $(\sigma, \rho, c)$  model. Since all flows will be assigned the same delay bound at a given scheduler, applying the schedulability condition in (3), we have:

$$C^i t \geq n^i A_f^*(t - d_f^i) \quad \forall t \geq 0 \quad (9)$$

where  $A_f^*(t)$  is the traffic constraint function defined by equation (4),  $C^i$  is the link capacity at scheduler  $i$ , and  $n^i$  is the number of ongoing flows at scheduler  $i$ . We solve (8) for  $N^i = \max(n^i)$  by considering *Figure 2*, which shows the two sides of (9) for the  $(\sigma, \rho, c)$  traffic model.

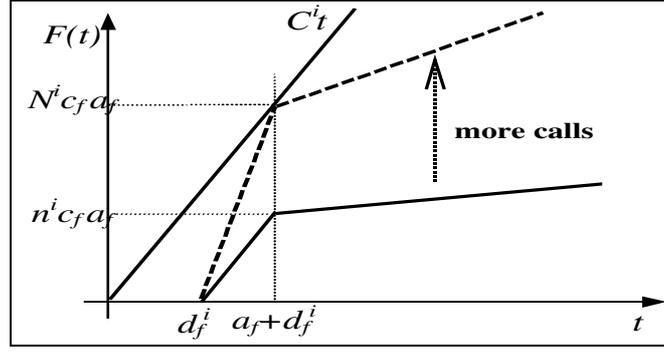


Figure 2. Deriving the value of  $N_i$  for a scheduler  $i$  along the path from source to destination

From the graph, we find that:

$$N^i = \frac{C^i (a_f + d_f^i)}{c_f a_f} \quad (10)$$

To satisfy (8), we must have:

$$C^i (a_f + d_f^i) = C^j (a_f + d_f^j) \quad \forall 1 \leq i, j \leq K \quad (11)$$

Equation (11) defines a set of  $K$  equalities, however there is one which is redundant and can be deduced from the other  $K-1$  equalities. To find the unique solution for the delays  $d_f^i$ ,  $1 \leq i \leq K$ , we use equation (6) as the  $K^{\text{th}}$  equation, from which we get:

$$d_f^j = \frac{C^i (a_f + d_f^i)}{C^j} - a_f \Rightarrow \sum_{\substack{j=1 \\ j \neq i}}^K d_f^j = \sum_{\substack{j=1 \\ j \neq i}}^K \left( \frac{C^i (a_f + d_f^i)}{C^j} - a_f \right), \text{ and}$$

$$D_f - d_f^i = C^i d_f^i \sum_{\substack{j=1 \\ j \neq i}}^K \frac{1}{C^j} + a_f C^i \sum_{\substack{j=1 \\ j \neq i}}^K \frac{1}{C^j} - a_f (K-1), \text{ and}$$

$$d_f^i = \frac{1}{\sum_{j=1}^K \frac{1}{C^j}} \left[ \frac{D_f}{C^i} + a_f \left( \frac{K}{C^i} - \sum_{j=1}^K \frac{1}{C^j} \right) \right] \quad \forall 1 \leq i \leq K \quad (12)$$

Substituting in (10), we get

$$N_{\max} = \frac{D_f + K a_f}{c_f a_f \sum_{i=1}^K \frac{1}{C^i}} \quad (13)$$

For the  $(\sigma, \rho)$  model, we note that it is basically a  $(\sigma, \rho, c)$  model with  $c \rightarrow \infty$ ,  $a \rightarrow 0$ , but  $ac \rightarrow \sigma$ . Taking these limits in (12) and (13) we get:

$$d_f^i = \frac{1}{\sum_{j=1}^K \frac{1}{C^j}} D_f \quad \forall 1 \leq i \leq K \quad (14)$$

$$N_{\max} = \frac{D_f}{\sigma_f \sum_{i=1}^K \frac{1}{C^i}} \quad (15)$$

Note that (14) depicts an inverse capacity proportional policy, as the delay bound assigned to a scheduler is inversely proportional with its capacity, and thus a scheduler with smaller capacity is required to reserve larger delay bound.

In sections 3.2 and 3.3, we define policies that can be applied to any network topology and set of flows.

## 3.2 Static Reservation Policies

In static reservation policies, the delay bound assigned to a scheduler is independent of its loading state. Therefore, the same delay will be assigned to a given scheduler for all flows of the same traffic characteristics and delay requirements offered to a single path network.

### 3.2.1 Even Policy (EVEN)

We use the even policy as a reference policy against which other policies may be compared. In the EVEN policy, all schedulers are required to reserve the same amount of delay, hence:

$$d_f^i = \frac{D_f}{K} \quad (16)$$

### 3.2.2 Optimistic static policy (OPTSTAT)

The OPTSTAT policy is based on the optimal policy for the special case of the single path initially unloaded network with homogenous traffic described in section 3.1. In OPTSTAT the delay at scheduler  $i$  along the flows path is calculated by equations (12) and (14) for the  $(\sigma, \rho, c)$  and the  $(\sigma, \rho)$  models respectively for arbitrary network topologies and traffic profiles.

### 3.2.3 The achievable gain of OPTSTAT over EVEN in a single path network with homogeneous flows

Let us revisit the single path initially-unloaded network with homogeneous flows with indefinite holding time. Let us derive the expression for number of simultaneous flows for the EVEN policy for this model. From (10), we have for the  $(\sigma, \rho, c)$  model:

$$N^i = \frac{C^i \left( a_f + \frac{D_f}{K} \right)}{c_f a_f} \quad (17)$$

and using (7),

$$N_{EVEN} = \frac{C^{\min} \left( a_f + \frac{D_f}{K} \right)}{c_f a_f} \quad (18)$$

In this case, OPTSTAT is optimal as proofed in section 3.1. Comparing with the expression of  $N_{\max}$  in (16), we get the relative gain value of the optimal static policy with respect to even policy:

$$G = \frac{N_{\max}}{N_{\text{EVEN}}} = \frac{K}{\sum_{i=1}^K \frac{C^{\min}}{C^i}} \quad (19)$$

For the  $(\sigma, \rho)$  model, from (10), by taking  $c \rightarrow \infty$ , we have:

$$N^i = \frac{C^i D_f}{K \sigma_f} \quad (20)$$

and using (7), (20)

$$N_{\text{EVEN}} = \frac{C^{\min} D_f}{K \sigma_f} \quad (21)$$

The gain is:

$$G = \frac{N_{\max}}{N_{\text{EVEN}}} = \frac{K}{\sum_{i=1}^K \frac{C^{\min}}{C^i}} \quad (22)$$

It can be readily easily seen that  $G \geq 1$ . Moreover, the larger the network (larger  $K$ ) and more unbalanced the link capacities are, the more gain achievable by OPTSTAT. In the case all capacities are equal,  $G = 1$  and OPTSTAT does not provide any gain. As will be shown in section 4, this holds true for networks with general topologies and heterogeneous traffic. In general, the more the network is unbalanced, the more the degradation in the performance of the EVEN policy.

### 3.3 Dynamic Reservation Policies

In dynamic reservation policies, the delay bound assigned to a scheduler depends, in general, on its loading state as well as the loading states of other schedulers on the flow path. This complicates the analysis of such policies because even when all flows have the same traffic characteristics and delay requirement, the assigned delay bound at a scheduler for a certain flow is not, in general, equal to those of the previous or subsequent flows. Therefore, for the dynamic policies, it is not possible to compute values such as  $N_{\max}$  as in the static policies. We propose three intuitive policies and test their performance using simulation later on. These policies are based on the assumption that each scheduler initially reserves the tightest possible delay value for the incoming flow. It then suggests subsequent relaxation of this reservation by redistributing the excess end-to-end delay on the schedulers according to some criterion. Let  $D_f^*$  be the minimum achievable end-to-end delay bound obtained by having each scheduler reserve the tightest possible delay value. The excess end-to-end delay is defined as:

$$\overline{D}_f = D_f - D_f^* \quad (23)$$

#### 3.3.1 Even distribution of excess delay (DYNEVEN)

This policy is the one suggested in [3]. The delay bound formula is given by:

$$d_f^i = d_f^{i*} + \frac{\overline{D}_f}{K} \quad \forall 1 \leq i \leq K \quad (24)$$

### 3.3.2 Capacity proportional distribution of excess delay (DYNCP)

In this policy, the excess delay is distributed in inverse proportion to the scheduler link capacity. The delay bound formula is given by:

$$d_f^i = d_f^{i*} + \frac{\overline{D}_f}{\sum_{j=1}^K 1/C^j} \quad \forall 1 \leq i \leq K \quad (25)$$

### 3.3.3 Remaining-delay proportional distribution of excess delay (DYNRDP)

In this policy, the excess delay is proportional to the minimum delay bound that the scheduler can guarantee to the incoming flow. The delay bound formula is given by:

$$d_f^i = d_f^{i*} + \frac{\overline{D}_f}{D_f^*} d_f^{i*} \quad \forall 1 \leq i \leq K \quad (26)$$

$$\Rightarrow d_f^i = d_f^{i*} \frac{D_f}{D_f^*} \quad \forall 1 \leq i \leq K \quad (27)$$

## 4. PERFORMANCE EVALUATION OF THE PROPOSED RESERVATION POLICIES

In this section, we consider several network models, and for each model we determine the blocking probability, and in some cases the average allocated delay at each scheduler using discrete-event simulation<sup>1</sup>.

In all models, flows are generated according to a Poisson process with an average arrival rate of  $\lambda$  connections/sec. Duration of a generic flow is assumed to be exponentially distributed with a mean  $1/\mu$  seconds. The value  $\rho = \lambda / \mu$  characterizes the offered traffic load (in Erlangs) at the network under consideration. We fix  $\mu$  to be one unit. For each simulation, we generate a total of between 100,000 and 1 million connections (depending on traffic load and network topology). Each experiment is repeated 10 times with different seeds to obtain confidence intervals (we do not show the confidence intervals in the results as they were generally very narrow except for blocking probability values less than  $1e-4$ ). The blocking probability is computed as the number of blocked connections divided by the number of generated connections.

The parameters of the simulation parameters; most notably link capacities, end-to-end delays, and source descriptors; are carefully chosen to meet the following objectives where possible:

- The delay requirements are as close as possible to the delay requirements of interactive voice calls over multi-service packet networks (e.g. ATM).
- Traffic parameters are similar to the traffic characteristics of standard packetized voice and/or video encoders.
- The schedulability condition of a scheduler, rather than the stability condition, is the one that puts the limit on the maximum number of flows. This is done because the performance of several resource reservation policies can only be differentiated when the stability condition does not limit the number of flows.

To gain insight about the performance of the policies, we start by studying two relatively simple network models (single path, and split network) in sections 4.1 and 4.2. In these two models we further assume a

<sup>1</sup> We use the simpack package available for download from: <http://www.cise.ufl.edu/~fishwick/simpack/howtoget.html>

single type of flows (with same descriptor and delay requirement) in each simulation run. The parameters of flow traffic descriptors and/or delay bounds are changed to determine the effects of those parameters on the performance of the policies. Later in section 4.3, we study a more general network model in which we offer flows with different and more realistic traffic characteristics and delay requirements. The objective is to check the effect of having intersecting network paths on the improvement (if any) introduced by non-even and dynamic reservation policies.

Our simulation code supports both the  $(\sigma, \rho)$  and the  $(\sigma, \rho, c)$  models. However, the paper only presents the results of the  $(\sigma, \rho)$  traffic model.

## 4.1 Model 1: Single Path Topology

*Figure 1* shows the topology of network model. For the results provided here we take  $K = 7$ .

### 4.1.1 Simulation Results

For all experiments, we set the source average rate to 16 Kbps, and set  $C^1 = C^2 = 1$  Mbps,  $C^3 = C^4 = 4$  Mbps,  $C^5 = C^6 = 16$  Mbps,  $C^7 = 64$  Mbps.

First, we study the effect of the source burst size. We set the required delay bound to 100 msec and the offered traffic load to 35 Erlang. We vary the burst size from 1 to 8 ATM cells. The results obtained from simulation when varying the burst size are shown in *Figure 3-a* and Table 1.

We also study the effect of the required end-to-end delay bound. We set the burst size to 3 cells and the offered traffic load to 40 Erlang. We vary the end-to-end delay bound from 50 msec to 200 msec. The results obtained from simulation when varying the end-to-end delay are shown in *Figure 3-b* and Table 2.

From these results, we note the following:

- Using equations (15) and (18) which give the maximum number of flows that can be simultaneously supported on a single path network for OPTSTAT and EVEN policies respectively, we have verified that the results were consistent with the traffic tables of Erlang B- formula for M/M/N/N systems.
- For the initially unloaded path, the DYNCP policy produces a delay distribution for the first flow that is the same as the one assigned by OPTSTAT policy. The use of the same traffic parameters for all flows, results in DYNCP giving the same results as OPTSTAT for all subsequent flows.
- For static policies, note that the delay values at each scheduler are the same for all values of burst size. On the other hand, the delay values of dynamic policies, except for DYNCP, vary with the change in burst size. DYNEVEN policy assigns tighter delay requirements to higher capacity schedulers as the burst size increases while relaxing it for lower capacity schedulers.
- The DYNRDP and OPTSTAT/DYNCP policies outperform other policies with DYNRDP providing even better performance for low values of burst size.
- *Figure 3-b* shows that the performance of the three dynamic policies becomes more similar as the required delay bound gets tighter. This can be explained by the fact the main difference between the three policies lies in the distribution method of the *excess delay* rather than the end-to-end delay. The excess delay is the difference between the requested delay bound and the minimum achievable bound. Therefore, reducing the requested delay bound reduces the excess delay and consequently reduces the performance difference among the dynamic policies.

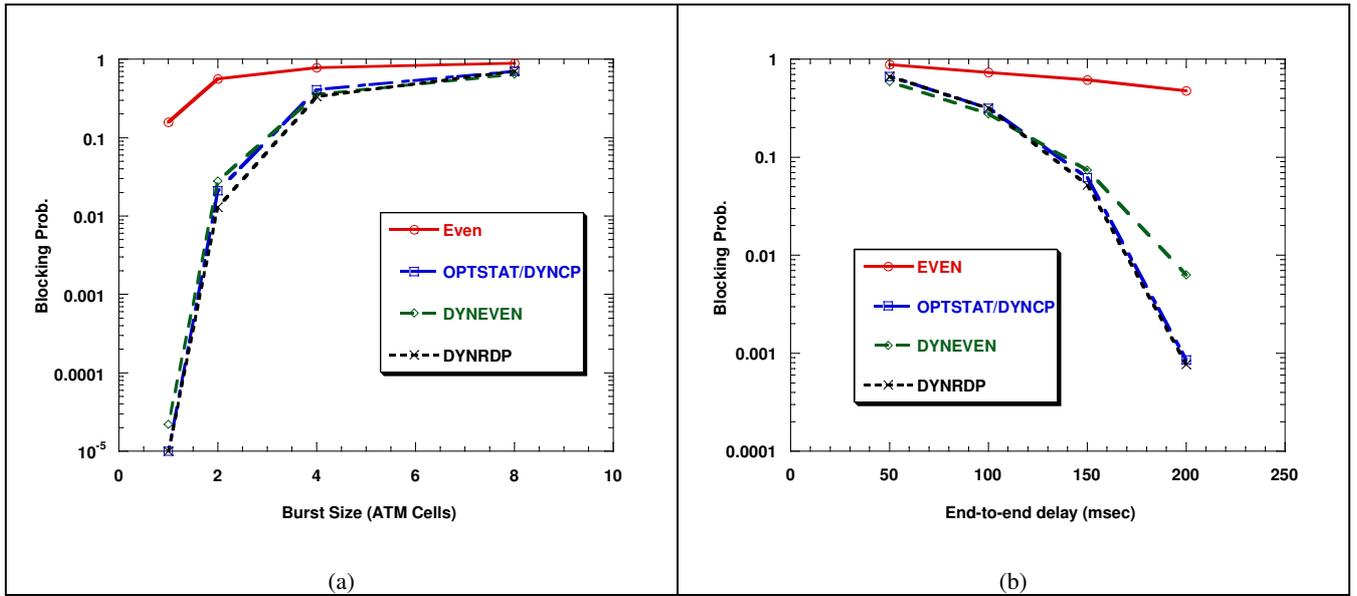


Figure 3. Results for model 1 (a) Effect of burst size variation on the blocking probability (b) Effect of End-to-end delay variation for model 1 on the blocking probability

Table 1. Effect of burst size variation for model 1 on the average allocated delays

Burst Size (ATM cells)	EVEN							OPTSTAT/DYNCP						
	Average allocated delay (msec)							Average allocated delay (msec)						
	d1	d2	d3	d4	d5	d6	d7	d1	d2	d3	d4	d5	d6	d7
1	14.29							37.87	37.87	9.47	9.47	2.37	2.37	0.59
2	14.29							37.87	37.87	9.47	9.47	2.37	2.37	0.59
4	14.29							37.87	37.87	9.47	9.47	2.37	2.37	0.59
8	14.29							37.87	37.87	9.47	9.47	2.37	2.37	0.59
Burst Size (ATM cells)	DYNEVEN							DYNRDP						
	Average allocated delay (msec)							Average allocated delay (msec)						
	d1	d2	d3	d4	d5	d6	d7	d1	d2	d3	d4	d5	d6	d7
1	18.2	18.2	12.8	12.8	12.7	12.7	12.7	37.8	37.8	9.5	9.5	2.4	2.4	0.6
2	28.6	28.6	8.7	8.7	8.44	8.44	8.40	36.7	36.7	10.4	10.4	2.6	2.6	0.6
4	30.1	30.1	8.3	8.3	7.74	7.74	7.65	34.2	34.2	12.3	12.3	3.1	3.1	0.8
8	30.3	30.3	8.85	8.85	7.26	7.26	7.1	37.9	37.9	9.5	9.5	2.4	2.4	0.6

Table 2. Effect of End-to-end delay variation for model 1 on the average allocated delays

End-to-end delay (msec)	EVEN							OPTSTAT/DYNCP						
	Average allocated delay (msec)							Average allocated delay (msec)						
	d1	d2	d3	d4	d5	d6	d7	d1	d2	d3	d4	d5	d6	d7
200	28.57							75.7	75.7	18.9	18.9	4.7	4.7	1.2
150	21.43							56.9	56.9	14.1	14.1	3.52	3.52	0.9
100	14.29							37.9	37.9	9.5	9.5	2.4	2.4	0.6
50	7.14							18.9	18.9	4.7	4.7	1.2	1.2	0.3
End-to-end delay (msec)	DYNEVEN							DYNRDP						
	Average allocated delay (msec)							Average allocated delay (msec)						
	d1	d2	d3	d4	d5	d6	d7	d1	d2	d3	d4	d5	d6	d7
200	56.2	56.2	17.7	17.7	17.4	17.4	17.3	75.7	75.7	18.9	18.9	4.7	4.7	1.2
150	46.3	46.3	11.9	11.9	11.2	11.2	11.2	54.7	54.7	15.9	15.9	3.9	3.9	1.0
100	30.7	30.7	8.2	8.2	7.39	7.39	7.33	37.9	37.9	9.5	9.5	2.4	2.4	0.6
50	15.3	15.3	4.3	4.3	3.64	3.64	3.58	18.9	18.9	4.7	4.7	1.2	1.2	0.3

## 4.2 Model 2: Split Topology

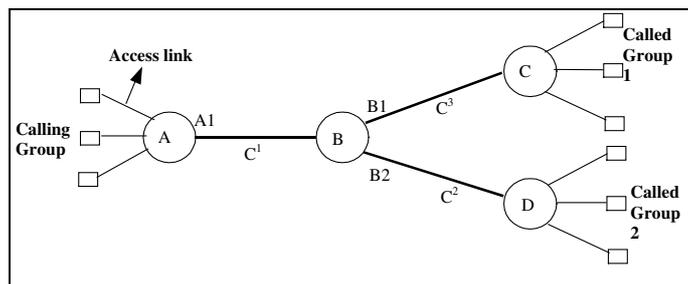


Figure 4. Topology of network model 2

Figure 4 depicts the topology for Model 2, which is a split network. we will simulate the following configurations of link capacities for this network model:

- **Configuration (A):** In this configuration, all links have the same capacity. Therefore, the results for the EVEN and OPTSTAT policies are always the same.
- **Configuration (B):** In this configuration, link capacities are chosen in proportion to their traffic load. This should be the configuration for a properly planned network in which higher capacities are placed at links on which high traffic load is expected.
- **Configuration (C):** In this configuration, link capacities happen to be in inverse proportion to their traffic load.

### 4.2.1 Simulation results for link configuration A

Here we take,  $C^1 = C^2 = C^3 = C = 1.5$  Mbps, and the source average rate = 16 Kbps, and the required delay bound = 100 msec.

We first study the effect of the burst size. We set the traffic load at 60 Erlangs and vary the burst size from 1 to 8 ATM cells. The results obtained from simulation when varying the burst size are shown in Figure 5-a.

We also study the effect of traffic load. We set the burst size equal to 3 ATM cells and vary the traffic loads from 50 to 80 Erlangs. The results obtained are shown in Figure 5-b. We note the following:

- OPTSTAT and EVEN policies give the same results. This is explained by the fact that the OPTSTAT policy for the case of  $(\sigma, \rho)$  traffic model gives the same results as EVEN policy when using equal capacity links.
- Dynamic policies provide much better performance than that of EVEN policies. Dynamic policies automatically account for the fact the link  $A \leftrightarrow B$  has a traffic load that is twice that of branching links. However, we note the improvement diminishes with the increase in the traffic load values.

### 4.2.2 Simulation results for link configuration B

Here we take,  $C^1 = 2C$ ,  $C^2 = C^3 = C = 0.75$  Mbps and the source average rate = 16 Kbps, the required delay bound = 50 msec, and the offered traffic load = 50 Erlangs. The results obtained from simulation when varying the burst size from 1 to 8 ATM cells are shown in Figure 6. We note the following:

- The OPTSTAT policy produces poor performance for such link capacity configuration. OPTSTAT puts higher resource load (i.e. tighter delay requirement) on the link already receiving higher traffic load (link  $A \leftrightarrow B$  in that case).
- Dynamic policies achieves better performance than EVEN policy. The performance improvement does not degrade considerably with the increase in burst size. No significant difference exists, however, among different dynamic policies due to the tightness of the used delay bound.

### 4.2.3 Simulation results for link configuration C

Here we take,  $C^1 = C = 1.5$  Mbps,  $C^2 = C^3 = 2C$  Mbps and the source average rate = 16 Kbps, the required delay bound = 70 msec, and the offered traffic load = 60 Erlangs. The results obtained from simulation when varying the burst size from 1 to 8 ATM cells are shown in Figure 7.

We note that there exists a significant improvement with dynamic policies for this link capacity configuration. Specifically, dynamic policies give less values of blocking ratio over that of the static policies EVEN and OPTSTAT.

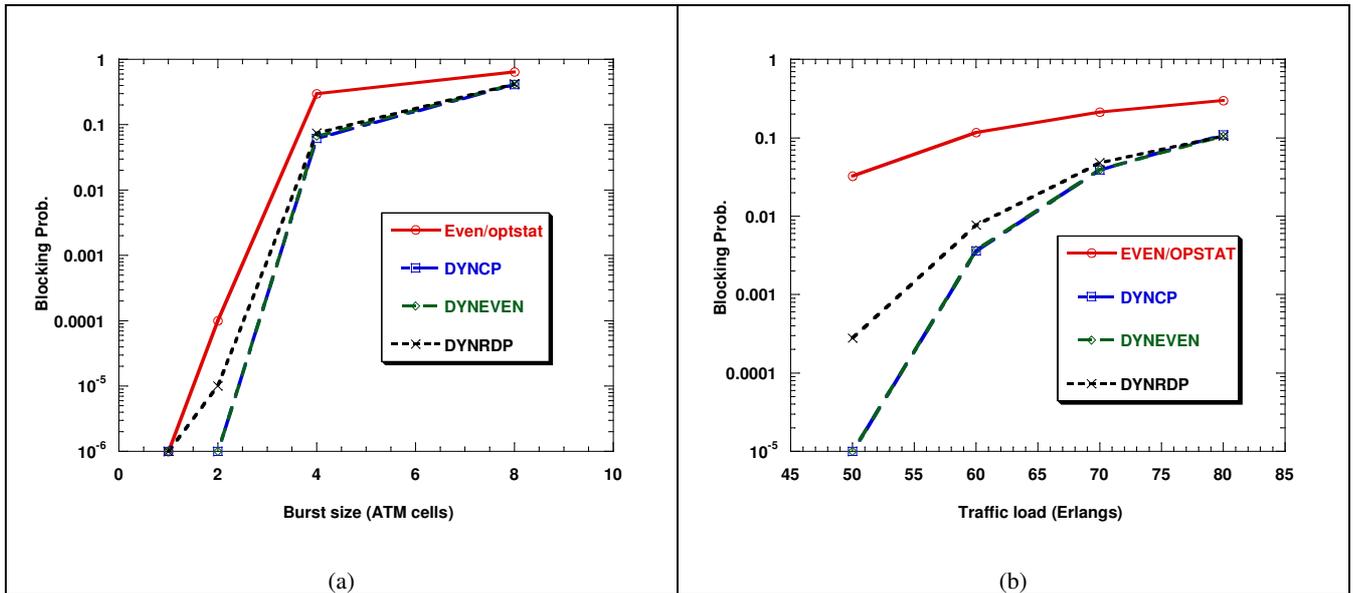


Figure 5. Results for model 2, configuration A. (a) Effect of burst size variation on the blocking probability, (b) Effect of traffic load variation on the blocking probability

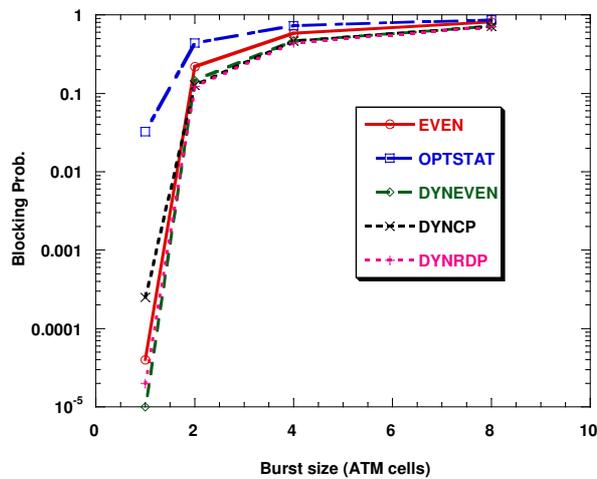


Figure 6. Effect of burst size variation for model 2, configuration B on the blocking probability

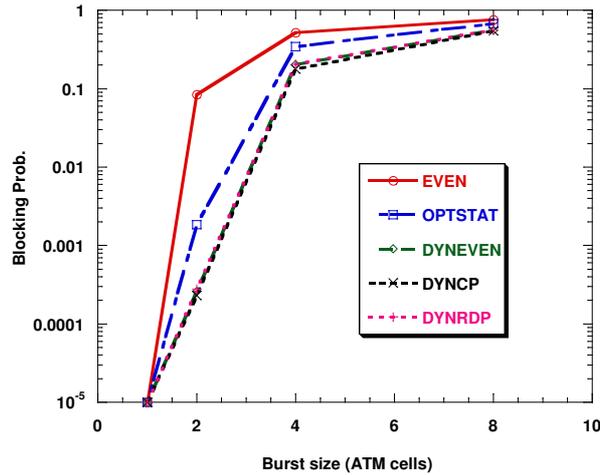


Figure 7. Effect of burst size variation for model 2, configuration C on the blocking probability

### 4.3 Model 3: General Network Topology with Shortest-Path Route Selection

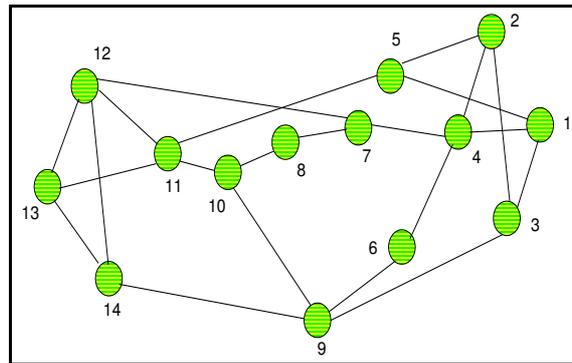


Figure 8. Topology of network model 3

We test the performance of the policies in a general network topology such as the NSFNET model shown in Figure 8. We evaluate the performance for two link configurations. The first configuration is when all links have the same capacity of 34 Mbps (equal-capacity links). In the second configuration, link capacities are randomly chosen such that overall capacity in the network is identical to the equal-capacity links case. The newly arriving flows are assigned with equal probability to one of available source destination pairs. A path is chosen for the flow using the widely popular shortest-path routing where the link cost is set to one unit.

Realistic values of traffic characteristics are used. Parameters for a new flow are selected as follows [5]:

- Average rate  $\rho = 10^m$  Kbps, where  $m$  is uniformly distributed on  $[0, 3]$ . This gives a mean average rate of 144.62 Kbps
- Burst size  $\sigma = y * \rho$  kbits, where  $y$  is uniformly distributed on  $[0.5, 1.3]$ . This gives a mean burst size of 130.16 Kbits
- Delay bound =  $50 * 10^s$  msec, where  $s$  is uniformly distributed on  $[0, 1.52]$

The range of generated traffic patterns include a typical MPEG video source with average rate = 518.4 kbps, and burst size = 576 kbits. It also includes a typical packetized voice source with average rate = 10 kbps, and burst size = 8 kbits.

We vary the offered traffic load from 2 upto 2048 Erlangs and obtain the blocking probability for the different resource reservation policies. The results for the equal-capacity and random link capacity allocation are shown in Figure 9 and Figure 10 respectively. Since some of the graphs are too close, we show in part (b) of the figures, the percentage reduction in the blocking probability achieved by using the proposed policies. For both the equal-capacity and random links case, we note that the dynamic policies outperform

the EVEN policy especially for low to medium values of traffic load. The best performing policy is consistently the DYNCP policy for the two link configurations. This is a good finding since link capacities are usually constant and in contrast to the DYNRDP policy, the signalling overhead needed for DYNCP would be much lower. We note that for very high loads, the DYNRDP policy starts to behave worse than the EVEN policy.

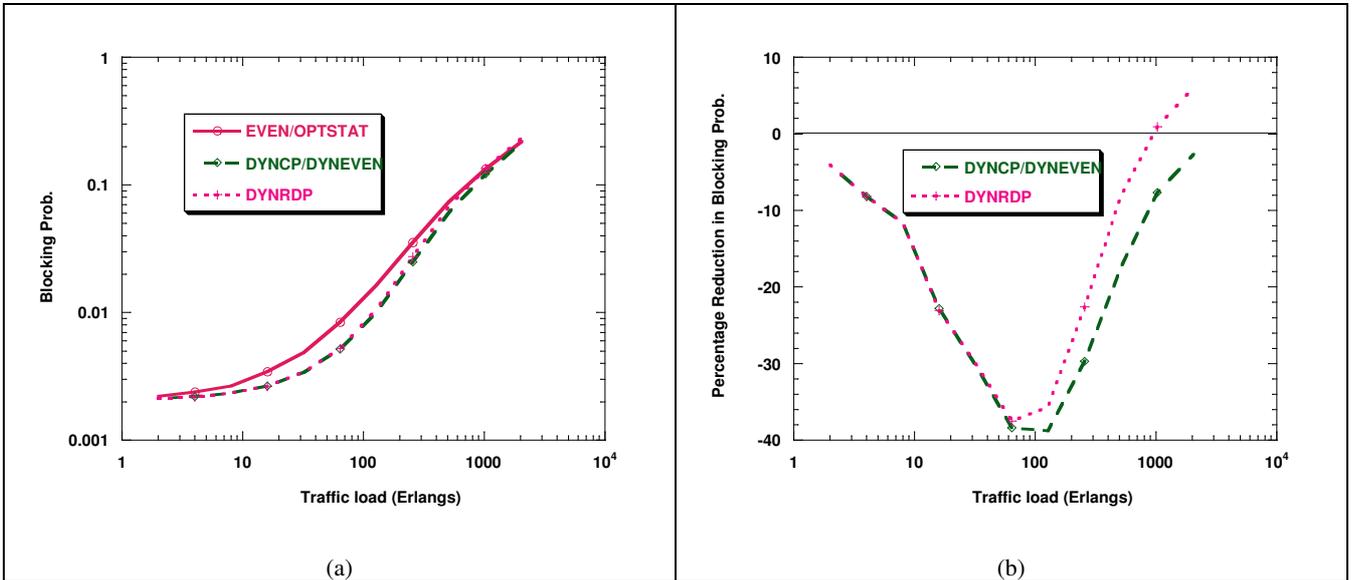


Figure 9. Results for for model 3 with equal link capacities. (a) Blocking probabilities of the various policies (b) reduction compared to EVEN policy

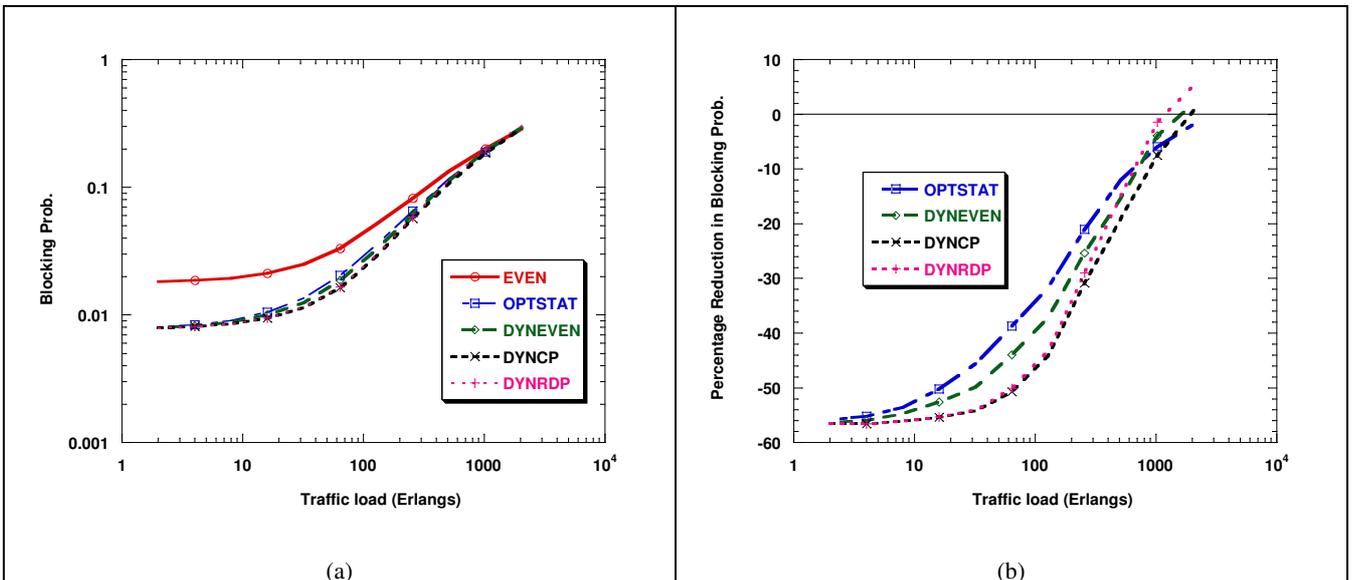


Figure 10. Results for model 3 with random link capacities. (a) Blocking probabilities of the various policies (b) reduction compared to EVEN policy

## 5. CONCLUSIONS

This paper discussed the use of non-even resource reservation policies when performing resource reservation in order to provide bounded delay service in a multi-service packet network. We have derived the required local delay allocations to obtain an optimal static policy for a single path network with homogeneous flows and indefinite holding times. In addition, we have derived an expression for the gain in the number of accepted flows along a single path of schedulers, due to the use of a non-even static resource reservation policies. This gain value increases with the number of schedulers on the flow path and with the imbalance in their link capacities. Also, we have proposed a set of dynamic resource reservation policies for a network with general topology and dynamic heterogeneous traffic.

Simulation was made for many network topologies and flows with various traffic profiles and delay requirements for incoming flows. Simulation results showed a significant advantage when employing non-even reservation policies that take network capacities/loading into consideration when reserving resources. The use of dynamic policies can provide even higher gain in situations where the flow path is initially loaded, which is common in a network with many intersecting flow paths. For a general topology with shortest-path routing, we found that the DYNCP provides the best performance.

It should be noted however that using non-even policies would, in general, involve the exchange of periodic link state packets containing more information about the loading state of network links. This may cause more processing burden on the control processors of network switches specially when employing more complex routing algorithms.

This paper has only considered the case of deterministic end-to-end delay bound as a QoS metric, our work can be extended by considering statistical end-to-end delay bound or other QoS metrics such as bandwidth or loss bounds. Another important area of research is integration with QoS routing for selecting paths that maximizes acceptance ratio of the traffic.

## APPENDIX A: PROOF OF THEOREM 1

To establish the required result, we first formulate two invariants. Firstly, the number of admitted flows is a non-decreasing function of the delay bound. On the other hand, the lowest achievable delay bound for a new flow is a non-decreasing function of the number of on-going flows. Under the condition described in theorem 1, we claim that the optimal policy (OPT) is the one which satisfies the condition that a scheduler  $i$

assigns a delay  $d_f^i$  to flow  $f$  subject to  $D_f = \sum_{i=1}^K d_f^i$  and that achieves  $N^i = \text{const} = N_{\max} \quad \forall 1 \leq i \leq K$

where  $f \in \{1, \dots, N_{\max}\}$ . To prove this claim, assume that another policy  $p$  results in the vector

$(N_1^p, N_2^p, \dots, N_K^p)$  and let  $N^p = \min\{N_i^p\}_{i=1}^K$ . If  $p$  is more optimal than OPT, then we must have

$N^p \geq N_{\max}$  and therefore  $N_i^p \geq N_{\max} \quad \forall 1 \leq i \leq K$ . This can only happen if the delay bounds at all schedulers

is relaxed or at least is equal to the delay bound assigned by OPT. If the delay bounds are equal we get the

same assignment of OPT, if the bounds are relaxed then the condition  $D_f = \sum_{i=1}^K d_f^i$  is violated for some of

the flows in  $f \in \{1, \dots, N^p\}$  and the constraint is violated. Therefore, OPT is the optimal policy under the assumptions of theorem 1.

## REFERENCES

- [1] M. Baldi and F. Risso, Efficiency of Packet Voice with Deterministic Delay, IEEE Comm. Magazine, pp. 170-177, May 2000.
- [2] K. Elsayed, A. Ayad, and M. El-Hadidi, Performance Evaluation of Resource Reservation and Call Admission Policies for Deterministic Services in PGPS-based Packet Networks, Elsevier Computer Communications, Vol. 25, pp. 1513-26, Nov. 2002.
- [3] D. Ferrari and D. C. Verma, A Scheme for Real-Time Channel Establishment in Wide-Area Networks, IEEE Journal on Selected Areas in Communications, Vol. 8, No. 3, pp. 368-379, 1990.

- [4] V. Firoiu and D. Towsley, Call Admission and Resource Reservation for Multicast Sessions, Proceedings of IEEE INFOCOM'96, pp. 94 – 101, March 1996.
- [5] V. Firoiu, J. Kurose, and D. Towsley, Efficient Admission Control of Piecewise Linear Traffic Envelopes at EDF schedulers, IEEE Trans. on Networking, Vol. 6, No. 5, pp. 558-570, Oct. 1998.
- [6] L. Georgiadis, R. Guerin, V. Peris, and K. Sivarajan, Efficient Network QoS Provisioning Based on Per Node Traffic Shaping, IEEE Trans. on Networking, Vol. 4, pp. 482 -501, Aug. 1996.
- [7] J. Golestani, A Framing Strategy for Congestion Management, IEEE Journal on Selected Areas in Communications, Vol. 9, No. 7, Sep. 1991.
- [8] A. Gupta and D. Ferrari, Resource Partitioning for Real-Time Communication, IEEE Trans. on Networking, Vol. 3, No. 5, pp. 501-508, Oct. 1995.
- [9] J. Liebeherr, D. E. Werge, and D. Ferrari, Exact Admission Control for Networks with a Bounded Delay Service, IEEE Transactions on Networking, Vol. 4 No. 6, pp. 885-901, 1996.
- [10] D. H. Lorenz and A. Orda, Optimal Partitioning of QoS Requirements on Unicast Paths and Multicast Trees, IEEE Trans. on Networking, Vol.. 10 No. 1, pp. 102-114, 2002.
- [11] R. Nagarajan, J. Kurose, and D. Towsley, Local Allocation of End-to-End Quality of Service in High-Speed Networks. Proceedings of IFIP Workshop on the Performance Analysis of ATM Systems, Martinique, Jan. 1993.
- [12] A. K. Parekh and R. G. Gallager, A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple Node Case, IEEE Trans. on Networking, vol. 2, pp. 137-150, 1994.
- [13] A. Saad, K. Elsayed, and M. El-Hadidi, Local Resource Allocation for End-to-End Delay Guarantees in Multimedia Networks, Proceedings of *IEEE Mascot99*, pp. 300-37, 1999.
- [14] H. Sariowan, R. Cruz, and G. Polyzos, SCED: A Generalized Scheduling Policy for Guaranteeing Quality-of-Service, IEEE Trans. On Networking, Vol. 7, pp. 669-684, Oct. 1999.
- [15] E. S. Valeroso and M. Alam, Performance Analysis of Resource Reservation Strategies in Broadband Networks, Proceedings of IEEE IPCCC'98, pp. 307 – 313, Feb. 1998.
- [16] H. Zhang and D. Ferrari, Improving Utilization for Deterministic Service in Multimedia Communication, Proceedings of the International Conference on Multimedia Computing and Systems, pp. 295–304, 1994.