An End-to-End QoS Framework with On-Demand Bandwidth Reconfiguration

Mei Yang[a], Yan Huang[a], Jaime Kim[b], Meejeong Lee[c], Tatsuya Suda[a], Matsubara Daisuke[d]

[a]*Department of Information and Computer Science*
University of California-Irvine
Irvine CA 92612, USA
[b]*Department of Computer Science*
California State University-Northridge
Northridge, CA 91330, USA

[c]*Dept. of Computer Science and Engineerin*
Ewha Womans University,
Korea
[d]*Central Research Lab of Hitachi, Ltd.*
Japan

Abstract--- This paper proposes a new QoS framework, called the On-Demand QoS Path framework (ODP). ODP provides end-to-end QoS guarantees to individual flows with minimal overhead, while keeping the scalability characteristic of DiffServ. ODP exercises per-flow admission control and end-to-end bandwidth reservation at the edge of the network and only differentiates traffic classes in the core of the network. In addition, to adapt to dynamically changing traffic load, ODP monitors the bandwidth utilization of the network and performs dynamic bandwidth reconfiguration in the network core. Through extensive simulations, the performance of ODP is investigated and compared with that of IntServ and DiffServ frameworks. The simulation results clearly show that ODP provides end-to-end QoS guarantees to individual flows, which DiffServ can not provide, with much less overhead than IntServ.

Keywords: End-to-End QoS; DiffServ; IntServ over DiffServ, Admission control; Bandwidth management

I. INTRODUCTION

In next generation high-speed IP networks, it is important to provide quality-of-service (QoS) guarantees to a wide range of applications in a scalable manner.

Significant efforts have been made recently, and a few network frameworks, such as the Integrated Services (IntServ) framework [1] and Differentiated Services (DiffServ) framework [2], have emerged as promising QoS-enabling frameworks. IntServ provides end-to-end QoS guarantees to individual flows by having routers maintain the states of each passing flow and by reserving bandwidth for each flow at routers along the path to the destination. As a consequence, IntServ does not scale well as the network size grows. On the other hand, DiffServ aggregates individual flows into different traffic classes at the edge of the network, and core routers within the network forward each packet to its next hop according to the per-hop behavior associated with the traffic class of the packet. In DiffServ, core routers do not keep states for individual flows, and this greatly improves the scalability. However, due to flow aggregation and the lack of admission control, DiffServ does not provide end-to-end QoS guarantees to individual flows.

In this paper, we propose and study a new QoS framework, called the On-Demand QoS Path framework (referred to as ODP in the rest of the paper). ODP provides end-to-end QoS guarantees to individual flows with much less overhead than IntServ, while keeping the scalability characteristic of DiffServ. In order to provide QoS to individual flows with minimal overhead, ODP exercises per flow admission control and end-to-end bandwidth reservation at the edge of the network. In the core of the network, ODP only differentiates the traffic classes as in DiffServ.

In the core of the network, ODP provisions bandwidth statically for each traffic class (provisioned link) on each physical link as DiffServ does, and applies class-based Weighted Fair Queue to differentiate them. In ODP, at each provisioned link, a certain amount of its bandwidth is allocated to each of the edge routers in the network. The bandwidth assigned to a specific edge router can only be used by flows originating from that edge router. In ODP, by having each edge router record utilization of its assigned bandwidth on each provisioned link and by deploying source routing, an edge router can make admission decisions instantaneously without hop-by-hop signaling, unlike in IntServ, when a flow with QoS requirement arrives at an edge router. For each admitted flow in ODP, an edge router first chooses an appropriate path for the flow and then reserves bandwidth required on each provisioned link along the path by updating the local bandwidth utilization table. In order to adapt to dynamically changing traffic load, the amount of bandwidth assigned to each edge router is reconfigured on an on-demand basis. This bandwidth assignment and reconfiguration are based on aggregated flows, not based on individual flows, and thus, the signaling overhead of the dynamic bandwidth assignment and reconfiguration is significantly reduced compared to IntServ.

In ODP, three approaches are considered to dynamically reconfigure bandwidth assigned to edge routers: (1) the Central Control approach, (2) the Router-Aided approach, and (3) the Edge-to-Edge approach. The three approaches are distinguished by the network entity that maintains the information of the bandwidth utilization at the provisioned link level and reconfigures bandwidth. The Central Control approach assumes a central network entity in the network, and the central network entity maintains bandwidth utilization at the provisioned link level and makes reconfiguration deci-

0-7803-8356-7/04/$20.00 (C) 2004 IEEE

sion accordingly. In the Router-Aided approach, core routers maintain the bandwidth utilization and make bandwidth reconfiguration decisions. In the Edge-to-Edge approach, edge routers maintain bandwidth utilization and reconfigure bandwidth. These three approaches in ODP explore a centralized, a semi-centralized and a distributed architecture, respectively.

The proposed ODP with the central control, router-aided and edge-to-edge approaches are evaluated through simulations in this paper. In addition, performance of ODP is compared with that of IntServ and DiffServ. Service quality metrics, such as the average packet transmission delay, average packet transmission delay variance, and signaling overhead due to dynamic bandwidth reconfiguration are obtained to show that ODP provides QoS at the same level as IntServ with much less signaling overhead.

This paper is organized as following. Section II surveys related work and shows that the proposed framework is new and original. Section III describes the proposed ODP, along with the three proposed bandwidth reconfiguration approaches (Central Control, Router-Aided and Edge-to-Edge approaches). In Section IV, simulation models are described, and in Section V, simulation results are presented to investigate the performance of the proposed ODP framework. Finally, concluding remarks are given in Section VI.

II. RELATED WORK

This section describes existing major QoS frameworks: IntServ, DiffServ, IntServ over DiffServ, DiffServ extensions with admission control, and DiffServ-aware Traffic Engineering (DS_TE).

In the IntServ framework [1], RSVP (Resource Reservation Protocol) is used to reserve bandwidth at routers along the path of a flow. When a flow arrives with a QoS requirement, the ingress edge router initiates the path establishment process by sending a PATH message to the destined egress edge router. The egress edge router responds by sending a RESV message back to the ingress router and tries to reserve the bandwidth required to meet the requested QoS along the path to the source ingress edge router. Core routers on the path configure their traffic control mechanisms such that each admitted flow is guaranteed to receive the bandwidth reserved, and thus the requested QoS. Through this per-flow based hop-by-hop signaling, IntServ provides end-to-end QoS guarantees. However, the overhead of processing per-flow bandwidth reservation and maintaining per-flow state at each router is significant, especially when a network is large. Thus IntServ has serious scalability concerns.

In the DiffServ framework [2], scalability is achieved by providing services on a per-class basis instead of a per-flow basis. In DiffServ, flows are grouped into a small number of classes at the boundary of a network, and the routers within the network merely implement a suite of scheduling/buffering mechanisms based on the classes. This per-class approach simplifies a router's functionalities and reduces the states that a router has to keep. The per-class approach also removes the per-flow QoS signaling overhead. Thus, DiffServ is more scalable than IntServ. However, DiffServ does not provide end-to-end QoS guarantees to individual flows. In addition, DiffServ does not exercise admission control at the network edge. When overload conditions occur in a given service class, all flows in that class suffer potentially harsh service degradation regardless of the QoS requirements of the class.

IntServ and DiffServ attempt to provide QoS on top of the current Internet, but with different approaches. IntServ promotes end-to-end QoS guarantees by establishing an end-to-end connection for each flow and maintaining the state of all connections, whereas DiffServ promotes scalability by aggregating individual flows in to a smaller number of traffic classes and providing differing treatments for different traffic classes. In attempt to meet both end-to-end QoS guarantees and scalability, RFC 2998[3] proposes a combination of the IntServ and DiffServ frameworks, an IntServ over DiffServ framework. In this framework [4] [5] [6], IntServ is applied at the edge of a network where scalability is not a major goal, and DiffServ is applied in the core of the network where a large number of flows coexist and scalability is a primary goal. Edge routers are responsible for admission control of individual flows and service mapping from IntServ service type to DiffServ traffic class. Routers in the core of the network only recognize traffic classes, not individual flows. Core routers also may or may not be RSVP aware.

In the IntServ over DiffServ framework, bandwidth can be provisioned either statically or dynamically using RSVP. With static RSVP provisioning, the core network provides a predefined set of service levels to customers (e.g. ingress edge routers) and statically provisions bandwidth according to the service levels. Each edge router contains a local table that describes the bandwidth statically provisioned for each traffic class that the edge router supports. Based on this table and traffic class mapping, an edge router makes admission decisions for incoming (IntServ) flows. Compared with the pure DiffServ framework, this scheme (IntServ over Diffserv with static provisioning of bandwidth) guarantees QoS for admitted flows by exercising admission control at the network boundary (i.e., at edge routers). However, this scheme does not readily support the addition of new service levels since bandwidth is statically provisioned. In addition, bandwidth utilization is usually low because bandwidth is often over-provisioned to accommodate short-term surge of flows.

With dynamic provisioning, RSVP is extended to reserve bandwidth for an aggregation of flows between edges of a network. When bandwidth provisioned for a certain path is not sufficient to support new flows, RSVP signaling procedure is initiated to dynamically reconfigure bandwidth. Admission control is invoked, and new flows are rejected when sufficient bandwidth cannot be dynamically reconfigured. This scheme (IntServ over Diffserv with dynamic provisioning of bandwidth) adopts admission control, end-to-end bandwidth reservation through RSVP, and recognizes individual flows at the edge of a network and traffic classes in the core of a network. This scheme tries to achieve better quality of service and better bandwidth utilization, while maintaining the same level of scalability as DiffServ. However, RSVP aggregation is one way to limit the signaling overhead at the cost of some loss of optimality in bandwidth utilization. In addition, if bandwidth provisioned on a path is used only by a single flow, this scheme loses its advantages and degrades to IntServ.

In the schemes proposed in [7] [8] [9] [10], an admission control function is provided over DiffServ network by means of the Endpoint Admission Control (EAC). EAC builds upon the idea that admission control can be implemented purely in an end-to-end manner, involving only the source and destination hosts. At connection setup, each source-destination pair starts a probing phase to determine whether a connection can be admitted to the network. The source node sends probing packets that reproduce the traffic characteristics of the connection to be established. Upon reception of the first probing packet, the destination host starts monitoring probing packets' statistics (e.g., loss ratio, inter-arrival times) for a given period of time. At the end of the measurement period, the destination makes the decision as to whether to admit or reject the connection and notifies the decision to the source. Although this scheme (DiffServ with EAC) is scalable since it does not involve inner routers, there are a number of downsides. Measurements taken in a short probing time may not capture stationary network states, and thus, the admission control decision is made based on a snapshot of the network that may not reflect the true status of network congestion. On the other hand, if measurements are taken in a longer probing period, the admission control process will be very slow. In addition, since probing packets are actually transmitted through a network during the measurement period, the probing packets increase the traffic load and contribute to network congestion.

In [11] [12], a new QoS framework, named DiffServ-aware MPLS Traffic Engineering (DS_TE), is proposed. In this framework DiffServ is complemented by MPLS Traffic Engineering mechanisms that operate on an aggregate basis across all DiffServ traffic classes. DS_TE further aggregate traffic classes to class types (CTs) and provision bandwidth for each CT in the core of a network. In order to reduce flooding overhead of link state advertisements, Inter Gateway Routing Protocol (IGP) extension of per class-type link-state advertisements (LSA) is used to exchange information on the available bandwidth for each class type. When an edge router makes an admission control decision, the edge router chooses a path for the incoming flow using Constraint Based Routing. Although the scalability of IGP LSAs is improved by propagating information on a per-class-type basis instead of on a per-class basis, no bandwidth provisioning is enforced for each traffic class within a class type. Flows of different traffic classes within a class type may interfere with each other.

Compared with the existing QoS frameworks discussed in this section, the ODP framework is alleviated from the scalability problem of IntServ, coarse granularity of QoS guarantees (i.e., per traffic class based, not per flow based, QoS guarantees) of DiffServ and DS_TE, and inaccurate admission control decisions of DiffServ with EAC. ODP guarantees the QoS for admitted flows, while keeping the scalability characteristic of DiffServ. ODP is somewhat similar to IntServ over DiffServ schemes (i.e., IntServ over DiffServ with static provision and with dynamic provision). In all schemes, the edge routers keep track of bandwidth utilization in order to carry out the admission control at the network edge. However, admission control in ODP is as simple as admission control in IntServ over DiffServ with static provision, while bandwidth utilization in ODP is as effective as that in IntServ over DiffServ with dynamic provisioning.

III. SCHEME DESCRIPTION

To provide end-to-end QoS guarantees for each flow, the On-Demand QoS Path framework (ODP) performs admission control and end-to-end bandwidth reservation for each flow at the edge of a network. To achieve scalability, ODP also performs class-based service differentiation in the network core. In Section 3.1, the basic architecture of ODP is described; in Sections 3.2 and 3.3, two major functionalities of ODP, namely, admission control and bandwidth reconfiguration are described respectively.

3.1 ODP Architecture

In ODP, it is assumed that two types of routers, edge routers and core routers exist in a network. Edge routers make an admission decision for each incoming individual flow, map individual flows to different traffic classes and transmit packets (belonging to a flow) to the network. Core routers are DiffServ routers. Namely, core routers do not recognize individual flows; Core routers only recognize traffic classes. Core routers also provide class-based service differentiation.

In order to provide class-based service differentiation and (local) admission control at the network edge without hop-by-hop signaling, ODP organizes link bandwidth in a hierarchical manner. Each physical link is statically divided into multiple Provisioned Links (PLs), and one PL is dedicated to one traffic class. Therefore the number of PLs on a physical link equals the number of traffic classes that the physical link can support. Each PL is further divided into multiple trunks; one trunk is dedicated to one edge router. A trunk (of a given provisioned link) supports flows (of the given traffic class) originating from the same source edge router irrespective of their destination. Flows going through the same PL in the network but originating from different source edge routers use different trunks. A source edge router keeps track of available bandwidth of its assigned trunks in the network and performs admission control locally without hop-by-hop signaling through the network. (Admission control procedure is described in detail in Section 3.2.). A Virtual IP Path (VIP) is a path from a source edge router to a destination edge router for a specific traffic class. It is a concatenation of multiple trunks over a source-destination path belonging to the source edge router. Figure 1 illustrates this hierarchical bandwidth organization in ODP.

In ODP, it is assumed that the bandwidth is statically assigned to provisioned links based on a long-term traffic prediction made by a central network management server (NMS) or by the network administrator during network initialization. This assignment remains unchanged for a reasonably long period of time compared to the time scale of our interests in this paper. In ODP, short-term traffic changes are managed by dynamically adjusting bandwidth assignment at the trunk level; a portion of the bandwidth of an underutilized trunk (or trunks) is reassigned to a trunk (or trunks) whose bandwidth utilization is high. (The bandwidth reconfiguration procedure is described in detail in Section 3.3)

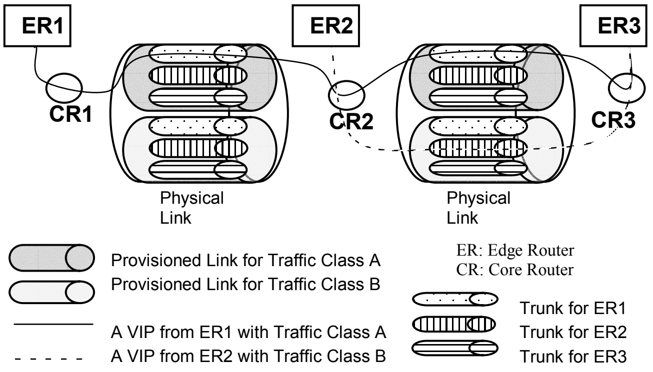

Figure 1 Hierachical Bandwidth Organization

Based on the network entity that maintains the bandwidth utilization information and makes bandwidth reconfiguration decisions, there are three possible approaches in ODP: the Central Control, the Router-Aided, and the Edge-to-Edge approaches. Detailed description of each approach is described in section 3.3.

3.2 Admission Control

As described in section 3.1, edge routers in ODP perform admission control without hop-by-hop signaling through a network. For admission control, an edge router maintains two tables: a virtual IP path (VIP) table and a trunk table.

A VIP table at an edge router records all VIPs originating from the edge router to all other edge routers. A VIP table at an edge router has an entry for each of the VIPs that originate from the edge router. An entry for a VIP path consists of the VIP ID, the ID of the destination edge router, traffic class that the VIP supports, and a list of trunks constituting the VIP path. (See Table 1)

TABLE 1. A VIP TABLE

VIP ID	Destination Edge Router	Traffic Class	List of Trunks

A trunk table at an edge router records bandwidth utilization of all trunks belonging to the given edge router. A trunk table at an edge router has an entry for each trunk that belongs to the edge router. An entry for a trunk consists of the trunk ID, ID of the provisioned link to which the trunk belongs, the amount of the reserved bandwidth on the trunk, and the amount of the bandwidth being used on the trunk. (See Table 2)

A VIP table and a trunk table may be pre-configured or constructed by routing mechanisms [18].

TABLE 2. A TRUNK TABLE

Trunk ID	Provisioned Link ID	Reserved Bandwidth	Bandwidth being Used

When a new flow arrives at a source edge router, the source edge router exercises admission control. The source edge router first examines its VIP table and identifies a VIP to the destination edge router. (It is assumed that traffic classification is already done before the source edge router examines the VIP table. Note that traffic classification is one of the basic functions of a DiffServ edge router.) The source edge router, then, examines its trunk table and obtains bandwidth utilization of each trunk on the VIP to the destination edge router. If the VIP has enough bandwidth to support the incoming flow, the source edge router accepts the incoming flow and makes end-to-end bandwidth reservation for this flow by updating the Bandwidth being used field for each trunk on the chosen VIP in its local trunk table. If the VIP does not have sufficient bandwidth to support the incoming flow, the incoming flow is rejected. Note that the admission control decision is made locally at a source edge router without hop-by-hop signaling through a network.

When a flow finishes, the bandwidth used by the flow becomes available for other flows. The source edge router updates its trunk table to reflect the change.

3.3 Trunk Reconfiguration

In order to adapt to short-term traffic changes, ODP dynamically adjusts the amount of bandwidth assigned to trunks. A source edge router requests additional trunk bandwidth or releases unused trunk bandwidth based on the trunk bandwidth utilization. This bandwidth adjustment is done within a provisioned link, allowing flows in the same traffic class (regardless of their originating source edge routers and destination edge routers) to share the provisioned link bandwidth. The trunk reconfiguration process in ODP is described below.

Step 1: Maintaining a Provisioned Link Table

A provisioned link table maintains bandwidth utilization of provisioned links in the network. Table 3 shows the format of a provisioned link table. An entry for a provisioned link consists of the provisioned link ID, ID of the physical link to which the provisioned link belongs, the amount of reserved bandwidth on the provisioned link, the amount of the bandwidth being used on the provisioned link, and the traffic class that the provisioned link supports.

TABLE 3 A PROVISIONED LINK TABLE

Provisioned Link ID	Physical Link ID	Reserved Bandwidth	Bandwidth Being Used	Traffic Class

Based on the network entity that maintains the provisioned link table, the following three approaches are proposed: the Central Control approach, the Router-Aided approach and the Edge-to-Edge approach. In the Central Control approach, a Network Management Server (NMS) maintains the provisioned link table, and the provisioned link table records the bandwidth utilization of all provisioned links in the network. In the Router-Aided approach, each (and every) core router in the network maintains a provisioned link table and records the bandwidth utilization of the provisioned links on all physical links directly connected to the core router. (Note that a physical link is connected to two core routers. Provisioned links on a physical link may only be maintained by one core router. This makes provisioned link tables maintained at different core routers unique.) In the Edge-to-Edge approach, each edge router maintains a provisioned link table, and a provisioned link table records the bandwidth utilization of all provisioned links in the network. (All edge routers maintain the same table.)

Table 4 summarizes how the provisioned link table(s) and also the trunk tables are maintained in each of the three approaches. Detailed descriptions on how these tables are updated and used to dynamically adjust trunk bandwidth are provided later in this subsection.

TABLE 4 PROVISIONED LINK TABLE AND TRUNK TABLE MAINTAINENCE

	Central Control	**Non Central Control**	
		Router-Aided	***Edge-to-Edge***
Provisioned Link Table	NMS maintains the bandwidth status (reserved bandwidth, bandwidth being used) for all provisioned links in the network	Each core router maintains the bandwidth status for the provisioned links directly connected to the core router.	Each edge router maintains the bandwidth status for all provisioned links in the network
Trunk Table	Each edge router maintains the bandwidth status of all trunks that are directly connected to the edge router		

Step 2: Releasing Unused Trunk Bandwidth

A source edge router periodically examines its trunk table (see Table 2) and obtains the bandwidth utilization of its trunks. If the bandwidth utilization of a trunk is under the predetermined lower threshold (l), the source edge router adjusts the bandwidth of the trunk, updates its trunk table to reflect the change, and then sends a control message to the network entity (or entities) that maintain(s) the provisioned link table.

Step 3: Acquiring Additional Trunk Bandwidth

When a new flow arrives at a source edge router, the source edge router assigns a VIP to the incoming flow from its VIP table (see Table 1) and examines its trunk table (see Table 2) to find if the bandwidth utilization of any trunks on the assigned VIP exceed the predetermined upper threshold (h). If the bandwidth utilization of a trunk exceeds the predetermined upper threshold, the source edge router sends a control message to the network entity (or entities) that maintain(s) the provisioned link table to request additional bandwidth for the trunk. Upon receiving a confirmation message from the network entity (or entities) that maintains the provisioned link table, the source edge router increases the bandwidth of the trunks.

As described in Steps 2 and 3 above, the trunk reconfiguration mechanism in ODP is always initiated by a source edge router. A threshold driven mechanism is used to dynamically adjust the bandwidth assigned to trunks. In ODP, it is assumed that the time interval (m) with which a source edge router periodically examines its trunk table, the upper and lower thresholds (l, h) for the bandwidth utilization, and the amount of bandwidth to increase (r1) and decrease (r2) in adjusting the trunk bandwidth are predetermined by a network administrator when the network is initialized.

As described above, in releasing unused trunk bandwidth (in Step 2) and acquiring additional trunk bandwidth (in Step 3), three types of control messages are exchanged: a Trunk Reconfiguration Request message (TRRequest), a Trunk Reconfiguration Release message (TRRelease) and a Trunk Reconfiguration Confirmation message (TRConfirm). The format of these three messages is the same, and it is shown in Figure 2. The message format consists of the message type to indicate the type of the message (i.e., TRRequest, TRRelease or TRConfirm), ID of the source edge router, ID of the destination edge router, ID of the VIP used by this control message, and length of this control message. The message format also contains a series of (trunk ID, amount of bandwidth requested/released) pair tuples. This is to indicate the amount of bandwidth requested (or the amount of bandwidth released) for the trunk identified by trunk ID in the tuple.

To reconfigure trunk bandwidth, a source edge router first identifies the trunks that require bandwidth reconfiguration and determines how much additional bandwidth may be assigned to them or how much unused bandwidth may be released from them. Then, the source edge router sends a TRRequest (if a trunk requires additional bandwidth) or a TRRelease (if a trunk releases unused bandwidth) to the network entity (or entities) that maintain(s) a provisioned link table. Upon receiving a TRRequest or a TRRelease, the network entity (or entities) maintaining the provisioned link table make(s) bandwidth reconfiguration decisions in the following manner.

Upon an arrival of a TRRequest message, a network entity with a provisioned link table, say X, performs the algorithm described in Figure 3. Note that in ODP, bandwidth is dynamically reconfigured, not on all trunks of a VIP, but only on trunks that do not have sufficient bandwidth.

Upon an arrival of a TRRelease message, network entity X performs the algorithm described in Figure 4.

Message type	**Source edge router ID**	**Destination edge router ID**	**VIP ID**	**Length**	**(Trunk i, Bdw_requested or Bdw_released)**	**(Trunk j, Bdw_requested or Bdw_released)**	…

Figure 2 TRRequest / TRRelease/ TRConfirm Message Format

```
1. Generate a TRConfirm message.
2. Copy each (trunk i, Bdw_requested or Bdw_released) pair tuple in the received TRRequest to the TRConfirm.
3. For trunk i specified in the TRRequest,
   3.1 Examine if provisioned link $PL_i$ that trunk i belongs to is in X's provisioned link table;
       If yes, then do 3.2
       Else set the Bdw_requested field of trunk i in TRConfirm message to 0 (0 indicates that X cannot make a decision for this request.)
   3.2 Examine if provisioned link $PL_i$ has sufficient bandwidth available for reconfiguration;
       If (the reserved bandwidth on $PL_i$ – bandwidth being used on $PL_i$) $\geq$ Bdw_requested for trunk i
       Then increase the bandwidth for trunk i, update the provisioned link table by
           Bandwidth being used on $PL_i$ = bandwidth being used on $PL_i$ + Bdw_requested for trunk i;
           And set the bandwidth requested field for trunk i in TRConfirm message to Bdw_requested (i.e. requested bandwidth has been assigned).
       Else set the bandwidth requested field for trunk i in TRConfirm message to -1 (i.e. the bandwidth reconfiguration request
           for the trunk has failed, and no bandwidth bas been assigned.)
4. Send the TRConfirm message to the edge router which initiated the TRRequest message.
```

Figure 3 TRRequest Message Process Algorithm

```
For trunk i specified in the TRRelease message,
1. Examine if provisioned link $PL_i$ that trunk i belongs to is in X's provisioned link table.
2. If yes, update the provisioned link table by
   bandwidth being used on $PL_i$. = bandwidth being used on $PL_i$ - Bdw_released for trunk i.
3. If no, ignore the TRRelease message.
```

Figure 4 TRReleaset Message Process Algorithm

TRRequest, TRRelease and TRConfirm messages are also used to update the provisioned link table(s) and trunk tables. In the Central Control approach, a TRRequest or a TRRelease is sent from a source edge router to the NMS that maintains the provisioned link table. The NMS makes bandwidth reconfiguration decisions and updates its provisioned link table. A TRConfirm is sent from the NMS to the source edge router that initiated a TRRequest. This allows the source edge router to update its trunk table. (Note that the NMS does not send a TRConfirmation for the TRRelease. The source edge router sending a TRRelease simply updates its trunk table when it sends a TRRelease, and the NMS updates its provisioned link table upon receiving a TRRelease. Similarly, there is no TRConfirm needed for a TRRelease message in the Router-aided and Edge-to-Edge approaches.)

In the Router-Aided approach, each core router maintains a provisioned link table that only records provisioned links directly connected to the core router. Upon an arrival of a new flow, a source edge router sends a TRRequest only to the core routers along the VIP assigned to the incoming flow. Core routers then perform trunk bandwidth reconfiguration, update their provisioned link tables, and send a TRConfirm back to the source edge router that initiated a TRRequest. On the other hand, a source edge router broadcasts a TRRelease to all core routers in the network, as a TRRelease can be for any trunk in the network.

In the Edge-to-Edge approach, each edge router maintains a provisioned link table, and provionsed link tables at different edge routers are identical. A source edge router broadcasts a TRRequest or a TRRelease to all other edge routers. Each edge router makes a bandwidth reconfiguration decision and sends a TRConfirm to the source edge router. It is possible for multiple source edge routers to compete for the bandwidth on the same provisioned links. Therefore, the initiating source edge router waits to receive a TRConfirm from all other edge routers before the source edge router updates its trunk table. After the initiating source edge router receives a TRConfirmation from all other edge routers, the source edge router can determine if its bandwidth reconfiguration request has been granted. If the bandwidth reconfiguration request has been granted, the source edge router updates its trunk table as well as its provisioned link table, and broadcasts a TRComplete to all other edge routers. If the bandwidth reconfiguration request has been rejected, the source edge router broadcasts a TRCancel to all other edge routers. TRComplete and TRCancel messages are two additional types of control messages used in the edge-to-edge approach, and they allow synchronization of provisioned link tables maintained at edge routers. Their message format is the same as that in Figure 2.

IV. SIMULATION MODEL

In order to evaluate ODP, ODP as well as IntServ/RSVP and DiffServ, are simulated using OPNET.

In simulations, vBNS+ [15] is used as the backbone network that connects a number of LANs to simulate a realistic network environment. (See Figure 3 for the vBNS+ topology used in the simulations.) There are 15 core routers, each of which supports 3 edge routers. Each edge router supports a LAN and transmits packets to and from the LAN it supports. Thus, there are 45 edge routers and 45 LANs in our simulation model. As in the vBNS+ network, OC-3 fiber links are used as physical links between core routers, and the distance between two neighboring core routers is also obtained from vBNS+. In the ODP Central Control approach, a network management server (NMS) is connected to core router 6 with an OC-3 fiber link. NMS maintains a provisioned link table. In the router aided and edge-to-edge approaches, core routers and edge routers maintain a provisioned link table, respectively.

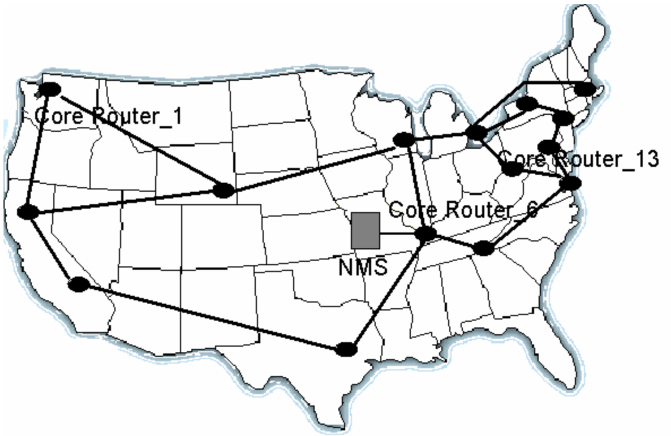

Figure 5 Topology from vBNS+

In the simulation, three traffic classes are assumed, namely, video, audio and UDP traffic classes. Table 5 shows characteristics of these three types of traffic. For each traffic class, Table 5 shows the average bandwidth (Aver Bdw), the distribution and average duration of flows (Flow Dis and Duration), the size of packets (Packet Size), and the distribution and average of the inter-arrival times of packets within a flow (Packet Inter-arrival time). In the table, notation "EXP (X)" refers to an exponential distribution with the average X.

Parameter values for the video traffic class are obtained from MPEG-4 with the assumption that the ratio between peak time and non-peak time is 1:3; audio parameter values are obtained from VoIP [16] [17] ITU-T G.723.1 standards with the assumption that an audio packet consists of a 4-byte compressed IP/RTP header and a 26-byte payload; UDP parameter values are randomly chosen. UDP traffic is used to represent background traffic in the simulation.

TABLE 5 TRAFFIC PARAMETER VALUES

Traffic Class	Aver Bdw	Flow Dis and Duration	Packet Size (bits)	Packet Inter-arrival time (seconds)
Video	3 Mbps	EXP (900s)	Constant (8000)	Peak: EXP (0.0016)
				Non-peak: EXP(0.0033)
				Peak: Non-peak =1:15
Audio	8 Kbps	EXP (300s)	Constant (240)	Constant(0.03)
UDP			Constant (8192)	EXP (0.03)

In the simulation, three different levels of network traffic load (high, medium and low levels of network traffic load) are considered by changing the inter-arrival times of video flows and audio flows, while keeping the ratio of the average arrival rates of audio flows and video flows to 99:1. For the low traffic load, it is assumed that the inter-arrival times of two successive flows follow an exponential distribution with the average inter-arrival time of 50 seconds. This corresponds to a traffic load of 275.4 Erlang. For the medium traffic load, the average inter-arrival time of flows is reduced to 25 seconds, and it corresponds to a traffic load of 518.4 Erlang. For the high traffic load, the average inter-arrival time is further reduced to 10 seconds, and it corresponds to a traffic load of 1377 Erlang. Each simulation is run for 5000 seconds. In the simulation, when a new flow arrives at a source edge router, its destination edge router is randomly chosen.

Using the simulation model described above, the performance of ODP is evaluated and compared with that of IntServ and DiffServ. The following five performance metrics are obtained.

- Blocking rate: Newly arriving flows may be blocked when there is not sufficient bandwidth available for the flow even after attempting to dynamically reconfigure bandwidth. The blocking rate is the ratio of the number of blocked flows to the number of all arriving flows.
- Signaling overhead: This is measured in the total number of control messages transmitted during the entire simulation duration. In IntServ, the signaling overhead includes RSVP signaling messages, such as PATH, RESV and PATH-TEAR messages. In DiffServ, no control messages are transmitted. In ODP, the signaling overhead includes TRRequest, TRConfirm and TRRelease. In the ODP edge-to-edge approach, the signaling overhead also includes TRComplete and TRCancel.
- Average connection setup time: This is the average time interval from when a flow arrives at a source edge router to when the admission decision is received by the flow.
- Average packet transmission delay: This is the average time interval from when a source edge router transmits a packet until when the destination edge router receives the packet. In the simulation, the average packet transmission delay is measured between an edge router connected to CoreRouter_1 and an edge router connected to CoreRouter_13. (See Figure 5.) Note that this is the longest path in the simulated network.
- Average packet transmission delay variance: This is the variance in the packet transmission delays. The transminssion delay variance is measured for packets traversing between an edge router connected to CoreRouter_1 and an edger router connected to CoreRouter_13 in the simulation.

V. EXPERIMENTAL RESULTS

In ODP, there are some parameters that impact the performance of ODP. Values of parameters that provide optimal ODP performance are first determined in Section 5.1, and such parameter values are used to further simulate ODP and compare the ODP performance against that of IntServ/RSVP and DiffServ in Section 5.2.

5.1. ODP Parameters

In this subsection, the following five ODP parameters are considered.

- Time interval (*T*) for a source edge router to examine the trunk bandwidth utilization: A source edge router periodically examines its trunk table with the time interval of *T* (in Step 2 of the trunk reconfiguration, section 3.3) and obtains the bandwidth utilization of its trunks. T is measured in seconds in the simulation.
- Lower bandwidth utilization threshold (*L*) and Bandwidth decrement granularity (*Bandwidth_decrement*): When the bandwidth utilization on a trunk is below the lower threshold *L*, namely, when $\frac{B_{used} + B_{average}}{B_{reserved}} < L$, a percentage (*Bandwidth_decrement*) of the available trunk bandwidth ($B_{reserved} - B_{used} - B_{average}$) is released from the trunk and made available to other trunks (in Step 2 of the trunk reconfiguration, section 3.3). Here, B_{used} is the bandwidth currently being used on a trunk, $B_{reserved}$ is the total amount of the bandwidth reserved on the trunk, and $B_{average}$ is the average bandwidth of a flow.
- Upper bandwidth utilization threshold (*U*) and Bandwidth increment granularity (*Bandwidth_increment*): When the bandwidth utilization of a trunk is above the upper bandwidth utilization threshold *U*, namely when $\frac{B_{used} + B_{average}}{B_{reserved}} > U$, a source edge router requests *Bandwidth_increment* times of the average bandwidth ($B_{average}$) of a flow, namely, the amount of the requested bandwidth is *Bandwidth_increment x* $B_{average}$, for the trunk (in Step 3 of trunk reconfiguration in section 3.3).

The values of the above five parameters are varied in the simulation to determine their optimal values. In each set of simulation results shown below, only one of the five parameters is varied, while others are kept unchanged. Once the optimal value is determined for a parameter, that value is used for the parameter in the subsequent simulation. In determining the optimal parameter values, the blocking rate is used as the criteria, since the major objective of ODP is to provide QoS guarantees to as many flows as possible (i.e., to minimize the blocking rate). When the blocking rate is the same, then the signaling overhead is used as a secondary metric.

In all the figures presented in this subsection, the x-axis shows different values for the parameter. The left y-axis represents the blocking rate, and the right y-axis represents the signaling overhead. All figures also show simulation results for three different traffic loads (i.e., low, medium and high traffic loads).

Figures 6, 7 and 8 show the performance of the ODP Central Control approach (ODP-CC), ODP Router-Aided approach (ODP-RA) and ODP Edge-to-Edge approach (ODP-E-to-E), respectively, for various values of the time interval (*T*) for a source edge router to examine the trunk bandwidth utilization. In these figures, only *T* is varied, while other parameter values are set as follows: *L* = 1, *Bandwidth_decrement* = 100%, *U* = 1 and *Bandwidth_increment* = 1, based on the preliminary flow-level simulations (where only flows are considered, and no packets within flows are considered).

Figures 6, 7 and 8 show that as the value of *T* decreases (or as a source edge router examines the bandwidth utilization of its trunks more frequently), the blocking rate first decreases. This is because the frequent examination of the bandwidth utilization provides a more accurate view of the network, allowing more effective reconfiguration of trunk bandwidth. However, as the value of *T* further decreases, the blocking rate starts increasing. This is because excessive monitoring of the trunk bandwidth utilization creates overhead (i.e., control messages such as TRRequest and TRRelease messages). From Figures 6 through 8, it is seen that the time interval (*T*) of 10 seconds for a source edge router to examine the trunk bandwidth utilization results in the significant improvement in the blocking rate for ODP-CC and ODP-RA, whereas the time interval (*T*) of 25 seconds yields in significant blocking rate improvement for ODP-E-to-E.

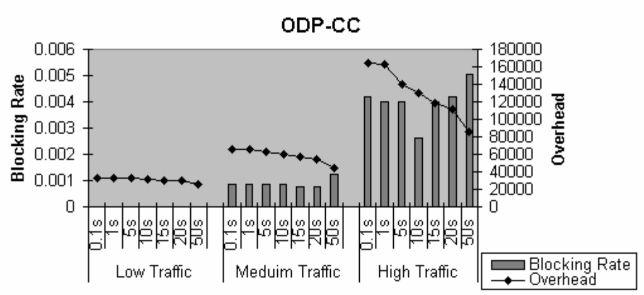

Figure 6 ODP-CC with Varying *T*

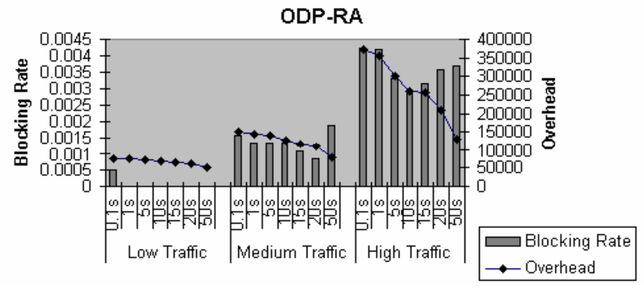

Figure 7 ODP-RA with Varying *T*

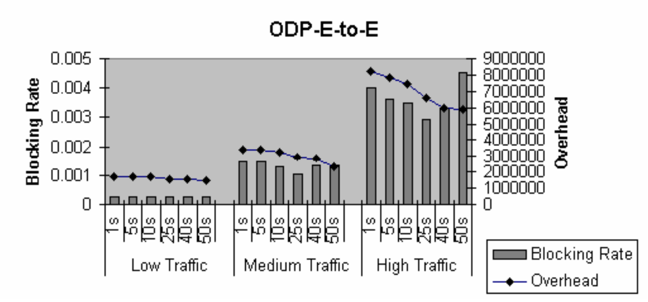

Figure 8 ODP-E-TO-E with Varying *T*

Figures 9, 10 and 11 show the performance of ODP-CC, ODP-RA and ODP-E-to-E, respectively, for various values of lower bandwidth utilization threshold (*L*). In these figures, only *L* is varied, while other parameter values are set as follows: *T* = 10 seconds (for ODP-CC and ODP-RA) or 25 seconds (for ODP-E-to-E), *Bandwidth_decrement* = 100%, *U* = 1 and *Bandwidth_increment* = 1.

Figures 9, 10 and 11 show that as *L* increases, the blocking rate decreases, and the signaling overhead increases. This is because that as the lower bandwidth utilization threshold value *L* increases, unused trunk bandwidth is released (to other trunks) sooner when the trunk bandwidth becomes under-utilized, allowing the more effective sharing of the unused bandwidth among trunks. From Figures 9 through 11, it is seen that the lower bandwidth utilization threshold value of *L* = 1 results in the smallest blocking rate for all three approaches of ODP.

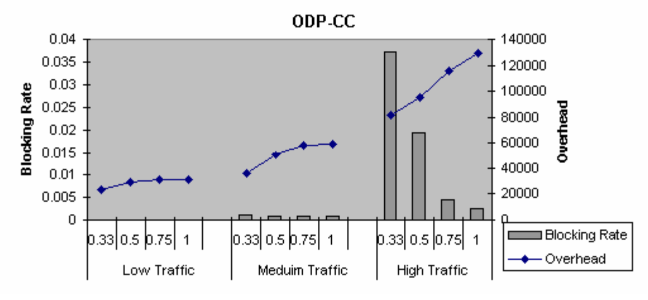

Figure 9 ODP-CC with Varying *L*

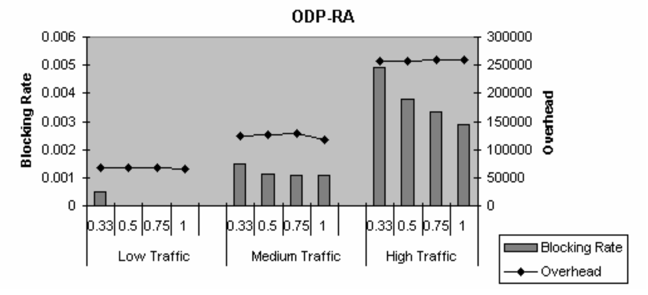

Figure 10 ODP-RA with Varying *L*

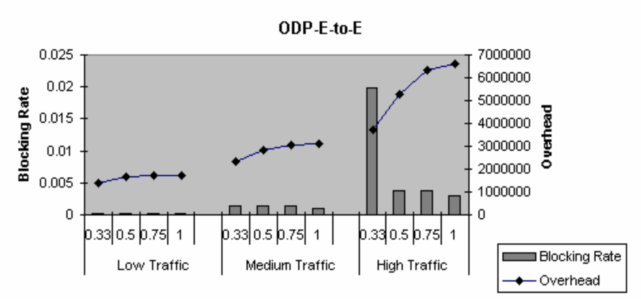

Figure 11 ODP-E-to-E with Varying *L*

Figures 12, 13 and 14 show the performance of ODP-CC, ODP-RA and ODP-E-to-E, respectively, for various values of bandwidth decrement granularity (*Bandwidth_decrement*). In these figures, only *Bandwidth_decrement* is varied, while other parameter values are set as follows: *T* = 10 seconds (for ODP-CC and ODP-RA) or 25 seconds (for ODP-E-to-E), *L*=1, *U* = 1 and *Bandwidth_increment* = 1. Figures 12, 13 and 14 show that as the value of *Bandwidth_decrement* increases, the signaling overhead decreases. This is because that with a larger value of *Bandwidth_decrement*, trunks release larger amount of bandwidth, and thus, bandwidth may be reconfigured less often, resulting in smaller signaling overhead. In addition, a larger amount of bandwidth is added to the bandwidth pool with larger value of *Bandwidth_decrement*. From Figures 12 through 14, it is seen that the bandwidth decrement granularity value of *Bandwidth_decrement* = 100 % results in the smallest blocking rate for all three approaches of ODP.

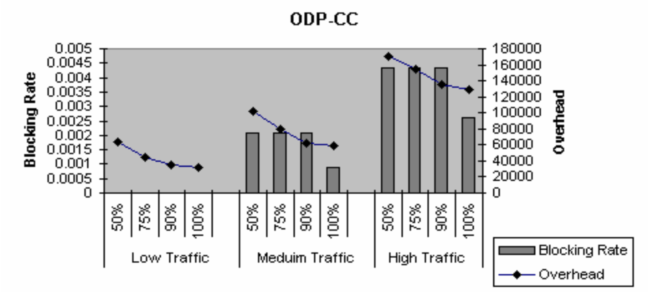

Figure 12 ODP-CC with Varying *Bandwidth_decrement*

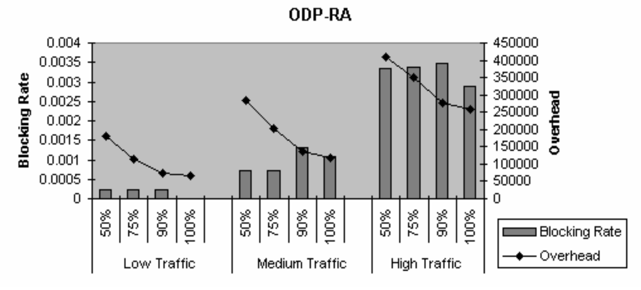

Figure 13 ODP-RA with Varying *Bandwidth_decrement*

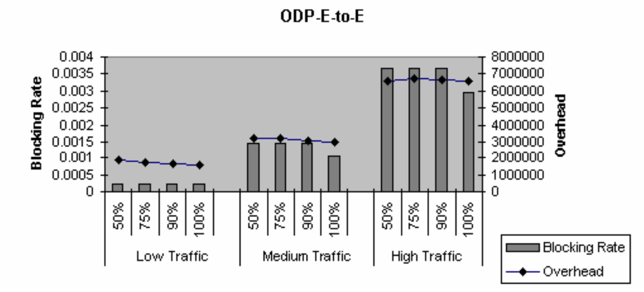

Figure 14 ODP-E-TO-E with Varying *Bandwidth_decrement*

Figures 15, 16 and 17 show the performance of ODP-CC, ODP-RA and ODP-E-to-E, respectively, for various values of the upper bandwidth utilization threshold *U*. In these figures, only *U* is varied, while other parameter values are set as follows: *T*= 10 seconds (for ODP-CC and ODP-RA) or 25 seconds (for ODP-E-to-E), *L*=1, *Bandwidth_decrement* = 100% and *Bandwidth_increment* = 1. Figures 15, 16 and 17 show that as the value of the upper bandwidth utilization threshold *U* increases, both the blocking rate and the signaling overhead decrease. This is because that with a larger value of *U*, a source edge router sends TRRequest messages less frequently, avoiding unnecessary control message exchanges that occur with a small value of *U*. This results in less overhead. In addition, additional bandwidth is only assigned to trunks whose bandwidth utilization is high (i.e., trunks that can benefit from additional bandwidth) with a large value of *U*, thus more effective sharing of the bandwidth among trunks is feasible, resulting in the smaller blocking rate. From Figures 15 through 17, it is seen that the upper bandwidth utilization threshold value of *U* = 1 results in the smallest blocking rate for all three approaches of ODP.

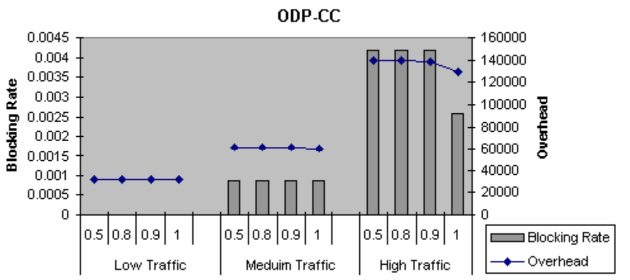

Figure 15 ODP-CC with Varying *U*

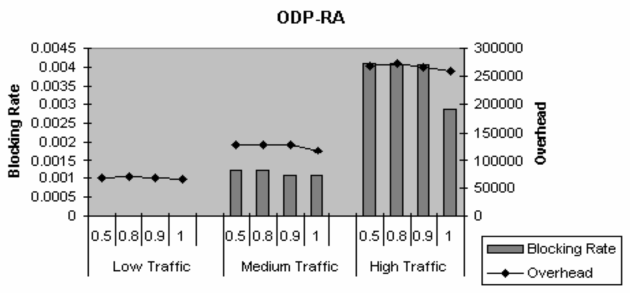

Figure 16 ODP-RA with Varying *U*

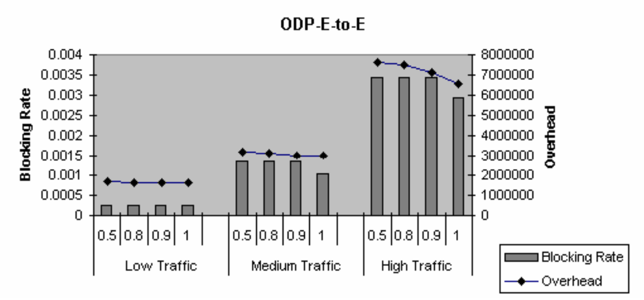

Figure 17 ODP-E-TO-E with Varying *U*

Figures 18, 19 and 20 show the performance of ODP-CC, ODP-RA and ODP-E-to-E for various values of Bandwidth increment granularity (*Bandwidth_increment*). In these figures, only *Bandwidth_increment* is varied, while other parameter values are set as follows: *T* = 10 seconds (for ODP-CC and ODP-RA) or 25 seconds (for ODP-E-to-E), *L*=1, *Bandwidth_decrement* = 100% and *U* = 1.

Figures 18, 19 and 20 show that as the value of *Bandwidth_increment* increases, the blocking rate generally increases. This is because that as a greater amount of bandwidth is assigned to requesting trunks, the smaller amount of bandwidth is left in the bandwidth pool to be shared by other trunks, resulting in a greater number of TRRequest messages being denied. Thus, as the value of *Bandwidth_increment* increases, the greater number of future flows will be blocked, and this results in the larger blocking rate. Figures 18, 19 and 20 also show that there exists trade-off for the signaling overhead. Note that as the value of *Bandwidth_increment* increases, a greater amount of bandwidth is assigned to requesting trunks. More bandwidth being assigned to trunks implies that trunks release the unused bandwidth more often and that a greater number of TRRelease messages are generated. This leads to a larger signalizing overhead. On the other hand, more bandwidth being assigned to trunks also implies that trunks may not require additional bandwidth frequently and that a smaller number of TRRequest messages are generated. This leads to a smaller signaling overhead.

From Figures 18 through 20, it is seen that bandwidth increment granularity value of *Bandwidth_increment* = 1 results in the smallest blocking rate for all three approaches of ODP.

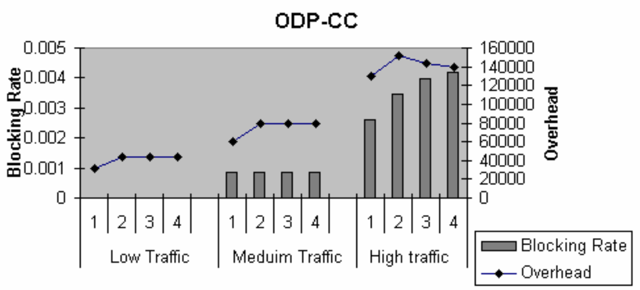

Figure 18 ODP-CC with varying *Bandwidth_increment*

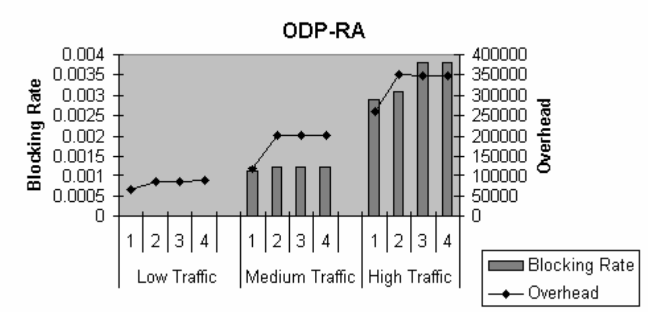

Figure 19 ODP-RA with Varying *Bandwidth_increment*

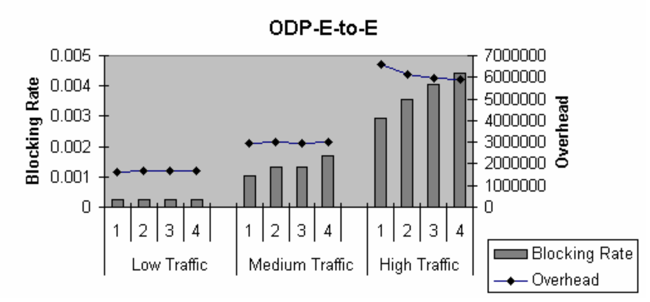

Figure 20 ODP-E-to-E with Varying *Bandwidth_increment*

5.2. Performance Comparison

In this subsection, ODP, IntServe and DiffServ are simulated, and their performance is compared. In comparing the performance of ODP and IntServ, the five performance metrics discussed in Section IV are used. Note that since neither admission control nor signaling messages are employeed in DiffServ, only the performance metrics of the average and variance of the packet transmission delay are used in comparing the performance of ODP with that of DiffServ.

Figure 21 shows the blocking rate and signaling overhead of IntServ and three approaches of ODP. Among the three approaches of ODP, ODP-E-to-E exhibits the largest blocking rate and signaling overhead. This is because of the conflict among multiple source edge routers on the trunk bandwidth. When multiple source edge routers compete for the bandwidth on the same provisioned link, it is possible with ODP E-to-E that requests for additional bandwidth (TRRequest messages) from different source edge routers arrive at different edge routers at different times. This creates a situation where one edge router accepts a request from a source edge router, and at the same time, another edge router denies the request from the same source router (because a request from a different source edge router arrived there first, and the available bandwidth has been assigned to that source router). Because of the reason described above, ODP-E-to-E exhibits the largest blocking rate. In addition, note that ODP E-to-E uses TRConfirm and TRCancel messages to avoid inconsistency in the provisioned link tables maintained at different edge routers, as explained in section 3.3. Thus, it also exhibits the largest signaling overhead among the three approaches of ODP.

Figure 21 shows that between IntServ and the three ODP approaches, IntServ shows the smallest blocking rate. This is because that due to the static bandwidth provisioning at the provisioned link level, it is possible with ODP that a new flow of a given traffic type is rejected even when there is sufficient bandwidth reserved for other traffic classes. This results in a higher blocking rate with ODP than that with IntServ. However, Figure 21 also shows that the signaling overhead of IntServ is significantly higher than that of ODP-CC and ODP-RA. This is because the trunk-level bandwidth management in ODP requires significantly less signaling messages than the per-flow bandwidth management in IntServ.

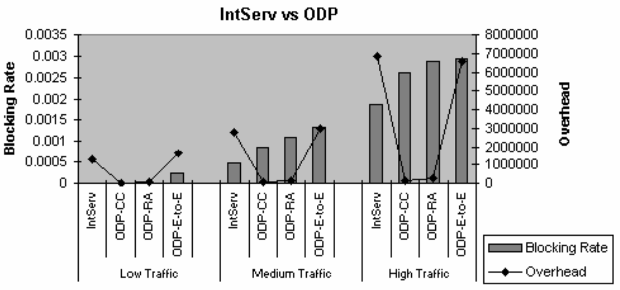

Figure 21 Blocking Rate and Signaling Overhead Comparison

Table 6 shows the average connection setup time of IntServ and three approaches of ODP. Three approaches of ODP exhibit significantly lower connection setup time than IntServ. This is because in ODP, source edge routers perform local admission control without hop-by-hop signaling.

TABLE 6 AVERAGE CONNECTION SETUP TIME COMPARISON

	Average Connection Setup Time		
	Low Traffic	**Medium Traffic**	**High Traffic**
IntServ	0.089	0.095	0.108
ODP-CC	0.0153	0.0153	0.0153
ODP-RA	0.0153	0.0153	0.0153
ODP-E-to-E	0.0153	0.0153	0.0153

Figure 22 shows the average and variance of the packet transmission delay of the audio traffic for IntServ, DiffServ and the ODP three approaches. It is seen that ODP-CC and ODP-RA provide approximately the same level of the average and variance of the packet transmission delay as IntServ. It is also seen that DiffServ provides the largest average and variance of the packet transmission delay among all the frameworks. For admitted flows, IntServ and ODP provide end-to-end QoS guarantees by reserving bandwidth on the path that an admitted flow will take. DiffServ, on the other hand, neither exercises admission control nor performs per-flow end-to-end bandwidth reservation. Thus, flows in DiffServ experience larger packet transmission delay and variance. Figure 22 also shows that the average and varience of the packet transmission delay in ODP-E-to-E is approximately the same as that of DiffServ. This is because in ODP-E-to-E, control messages that do not exist in other two ODP approaches (such as TRComplete and TRCancel messages) consume bandwidth, making less bandwidth available for data packet transmission.

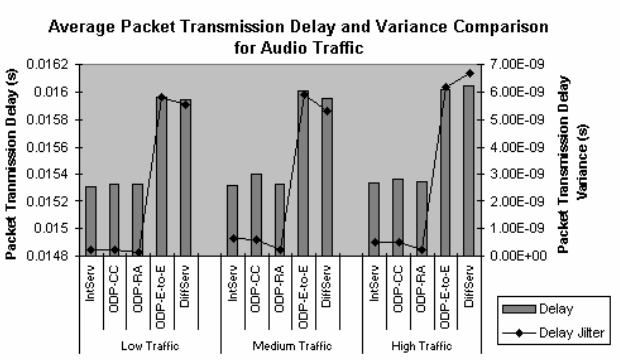

Figure 22 Average Packet Transmission Delay and Delay Variance Comparison

To summarize, simulation results presented in this section clearly show that the two approaches of ODP, namely, ODP-CC and OCP-RA, provide end-to-end QoS guarantees to individual flows with significantly less signaling overhead than IntServ, while keeping the scalability characteristics of DiffServ.

VI. CONCLUSION

In this paper, a new QoS framework, called On-demand QoS Path (ODP), was proposed. The simulation results clearly showed that ODP Central Control and Router-Aided approaches provide end-to-end guarantees to individual flows with significantly less overhead than IntServ. Even though the simulation results did not show apparent advantages of the ODP Edge-to-Edge approach, it is the most robust approach since it avoids a single-point-failure. The authors of this paper believe that the ODP Edge-to-Edge approach still merits consideration.

ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation through grants ANI-0083074 and ANI-9903427, by DARPA through Grant MDA972-99-1-0007, by AFOSR through Grant MURI F49620-00-1-0330, by Basic Research Program of KOSEF through grant R04-2000-000-00078-0 and by grants from the University of California MICRO and CoRe Programs, Hitachi, Hitachi America, Novell, Nippon Telegraph and Telephone Corporation (NTT), NTT Docomo, Fujitsu, and NS-Solutions.

REFERENCES

1. J. Wroclawsky, “The use of RSVP with IETF Integrated Services," IETF RFC2210, September 1997.
2. K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and Ipv6 Headers," RFC2474, December 1998.
3. Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer etc., “A Framework for Integrated Services Operation over Diffserv Networks,” RFC 2998, November 2000.
4. F. Reichmeyer, L. Ong, A. Terzis, L. Zhang, R. Yavatkar, “A Two-tier Resource management model for Differentiated Services Networks,” IETF Internet Draft, Nov 1998.
5. F. Borgonovo, A. Capone, L. Fratta, M. Marchese, C. Petrioli, "End-to-end QoS Provisioning Mechanism for Differentiated Services," Internet Draft, July 1998.
6. F. Borgonovo, A. Capone, L. Fratta, C. Petrioli, "VBR Bandwidth Guaranteed Services over DiffServ Networks," IEEE RTAS Workshop, June 1999.
7. V. Elek, G. Karlsson, R. Rönngren, “Admission Control Based on End-to-End Measurements,” INFOCOM 2000.
8. G. Bianchi, N. Blefari-Melazzi, “A Migration Path to provide End-to-End QoS over Stateless Networks by Means of a Probing-driven Admission Control,” Internet Draft, November 2001.
9. M. Grossglauser , David N. C. Tse, “A Framework for Robust Measurement-based Admission Control,” IEEE/ACM Transactions on Networking (TON), v.7 n.3, p.293-309, June 1999.
10. G. Bianchi, A. Capone, C. Petrioli, "Packet Management Techniques for Measurement Based End-to-end Admission Control in IP Networks," KICS/IEEE Journal of Communications and Networks (special issue on QoS in Internet), July 2000.
11. G.Bianchi, N.Blefari-Melazzi, M.Femminella, “Per-flow QoS Support over A Stateless Differentiated Services IP Domain,” Computer Networks, Elsevier, 40(1) 2002.
12. F. L. Faucheur, “Protocol Extensions for Support of Diff-Serv-aware MPLS Traffic Engineering,” IETF Internet Draft, October 2002.
13. F. L. Faucheur, W. Lai, “Requirements for Support of Diff-Serv-aware MPLS Traffic Engineering,” IETF Internet Draft, February 2003.
14. G. Apostolopoulos, S. Kamat, R. Guerin, SK. Tripathi, "Quality of Service Based Routing: A Performance Perspective," SIGCOMM, 1998.
15. vBNS IPV6 Service Overview White Paper, vBNS.
16. H. Lin, T. Seth, A. Broscius, C. Huitema, “VoIP Signalling Performance Requirements and Expectations,” IETF Internet Draft, Jun 1999.
17. P. T. Brady, "A Model for Generating On-Off Speach Patterns in Two-Way Conversation," The Bell System Technical Journal, September 1969, pp.2445-2471.