Smart Box Architecture

Bülent Yener *

Abstract

Fundamentally the IP-based networking is designed for delivering data traffic with best-effort service, thus it is not capable of providing end-to-end QoS. Several architectures have been proposed for providing QoS in the Internet: The *integrated services* (Intserv) model is based on reservations and can provide QoS, however; it is not scalable. The *differentiated services* (Diffserv) approach is scalable but falls short of ensuring deterministic guarantees —in particular for the services that belong to the same class. Finally, the *multi protocol label switching* (MPLS) architecture provides mechanisms for QoS-based routing but does not have the necessary resource management and scheduling support to ensure it.

This work proposes a hybrid solution which combines the best of these technologies. First, at the network boundary Diffserv like Service Level Agreements (SLA) are provided to users by intelligent edge routers called the **SBoX servers**. An SBoX server uses Class Based Queuing (CBQ) with a hierarchy of flow aggregation. At the top a *commodity-flow* is defined for the aggregate flow between a pair of egress points. The packets of the same commodity-flow are marked by an MPLS label, which is globally unique within an Autonomous System (AS). Each commodity flow is partitioned to a set of *macro-flows* which are offered to users as SLAs. An SBoX server manages macro-flows and commodity flows only, and leaves the management of each macro-flow (at the micro-flow level based on some policies) to the enterprise/users which signed the SLA. Second, the commodity-flows are managed and supported inside the network by an add-on Label Switching Router (LSR) called the **SBoX router** which performs MPLS of commodity-flows with CBQ. The main reason for an add on solution is the lack of end-to-end deployment of LSRs, and the vertically integrated architecture of the legacy routers. This paper explains the SBoX architecture and reports experimental results obtained on a prototype network.

*Bell Laboratories, Lucent Technologies, 700 Mountain Ave., Murray Hill, NJ 07974 E-mail: yener@research.bell-labs.com, Tel: (908) 582 7087

1 Introduction and Motivations

As the Internet gets commercialized, the need for providing QoS becomes imminent. Current infrastructure of the Internet cannot support QoS since it is designed for best-effort service model. Its routers operate with FIFO scheduling without any guarantees. Increasing the network capacity by adding more bandwidth and routers is not always feasible or efficient. The network must have mechanisms to distinguish QoS requirements of different applications that share the same infrastructure and process them accordingly.

There are two fundamentally different perspectives to the QoS problem in the Internet: (i) *Integrated Services* (Intserv), and (ii) *differential services* (Diffserv). The Intserv approach [SPG97, Wro97, SW97] requires high-end routers to maintain per-micro-flow [1] state information and to perform complex link scheduling algorithms. As the number of flows increase and/or change frequently, the overhead of this approach does not scale.

The Diffserv approach aims to [Cla97, CW97, NJZ97, SZ98, ea98] to reduce the per-flow complexity by providing an aggregated treatment of user traffic that belongs to the same service class. Packets in the Diffserv model are marked, at the network entry points, to indicate whether or not the source follows its SLA. The packets that violate their SLA (i.e., OUT packets) are dropped with a higher probability than the packets obey their SLA (i.e., IN packets). Several performance studies [BW99, IN98] of Diffserv approach show that (1) it cannot offer a quantifiable service to TCP traffic, (2) there is strong dependency between IN and OUT packets (due to shared queue) and consequently IN packets may be dropped, and (3) mixing IN and OUT packets in a single TCP connection reduces the connection's performance. Non-deterministic and non-quantifiable QoS cannot be acceptable for certain applications, such as IP telephony, real-time interactive transactions, IP high definition video (e.g., HDTV), and Virtual Private Networking (VPN) for which pricing and QoS are strongly coupled.

Other approaches such as Core-Stateless Fair Queu-

[1] A micro-flow is a application level flow defined by its source, destination address, port numbers and protocol identifier.

ing (CSFQ) [SSZ98] attempt to compromise by replacing the per flow state overhead at the routers by Dynamic Packet States (DPS) [ea99b]. The DPS approach requires checking and updating a state information associated with each packet. Thus it introduces extra processing complexity at each node. Furthermore it requires modification of existing routers.

Recently, Multi Protocol Label Switching (MPLS) [RVC98] has become an attractive technology. The main contributions of MPLS are twofold. First, it decouples routing from forwarding. Thus, *explicit routing*, based on metrics different from the ones used by the traditional routing algorithms can be deployed. Second, it aggregated the flows to Forwarding Equivalent Classes using virtual circuit switching. The former enables traffic engineering and QoS based constrained routing while the latter provides a scalable solution. Although it contributes significantly, the MPLS is not enough for providing QoS since it does not address resource allocation, scheduling and admission control problems.

1.1 Principles of the Solution

Proposals to the QoS problem in the IP networks must address the fundamental trade off between the scalability and the QoS guarantees. In order to provide end-to-end QoS guarantees, some notion of resource reservation is necessary. However, such reservation must be for aggregated flows to minimize per-flow state information at the routers for a scalable solution. The granularity of aggregation (i.e., the set of the flows that are treated by the network uniformly) impacts on the level of service each micro-flow receives and may create fairness problems. Furthermore, real-time, adaptive resource management and admission decisions increase the complexity of the interior nodes thus cause performance bottlenecks.

From these observations a separation of the QoS functions at the edge and at the core nodes can be identified as follows. Processing and state information overhead that cause scalability problem at interior nodes should be pushed to the egress points. Thus, edge nodes must be in charge of (i) admission control, (ii) QoS based network access which includes micro-flow management, traffic engineering, and generating aggregate flows, and (iii) service based billing and charging.

In contrast, the responsibility of interior nodes must be limited to a minimum set of operations sufficient to support the egress QoS mechanisms inside the network. Their complexity can be reduced by adapting the following principles: (i) separating routing from forwarding, (ii) operating on a pseudo circuit switching (e.g., MPLS, virtual circuit switching) mode to eliminate per-packet IP lookup and filtering overhead (iii) using aggregated flow management with CBQ or WFQ scheduling, and (iv) using light-weight signaling protocols for aggregate flow reservations.

These goals may be achieved by a hybrid architecture that combines the best of Intserv, Diffserv, and MPLS proposals with QoS provisioning and admission control.

1.2 Overview of the Proposed Solution

This work proposes such a hybrid solution. First, at the network boundary intelligent edge routers called the **SBoX servers** are deployed. An SBoX server combines the Diffserv and MPLS architectures at a network access point. SBoX servers provide Diffserv like SLA. Each agreement is a macro-flow with a specific bandwidth, and established between two egress points. A macro-flow is shared by all the application-level microflows between these end points. All the macro-flows in the SLA are aggregated to a *commodity-flow*. The packets of the same commodity-flow are marked by an MPLS label which is globally unique within an AS. The SLA's are enforced and packed into a commodity-flow by CBQ scheduling.

Second, inside the network, a subset of the best-effort IP routers are enhanced with a programmable, add-on Label Switching Router (LSR) called the **SBoX router** (SBoX). An SBoX router performs MPLS with CBQ over IP, in a totally transparent way to current IP-based network infrastructure and protocols. It controls some of the links of such IP nodes, so that it switches labeled traffic (commodity-flows) directly, and passes other traffic (best effort) to the legacy router. The label switching is done by using the global labels and commodity-flows are protected from each other and from best-effort traffic using CBQ. The main motivation behind add-on SBoX routers is to address the following problems: (1) the lack of end-to-end deployment of LSRs (i.e., many routers are not MPLS capable), and (2) the lack of programmability of the legacy routers. An add-on approach provides service guarantees in a network that contains legacy routers without replacing or changing them.

Third, new algorithms are proposed for provisioning, resource allocation, and admission control. The core of provisioning and resource allocation problem is the placement of SBoX routers and the selection of the links that enforce QoS guarantees. The SBoX routers and the links that they manage induce an virtual premium network which is used for label switching of the QoS traffic.

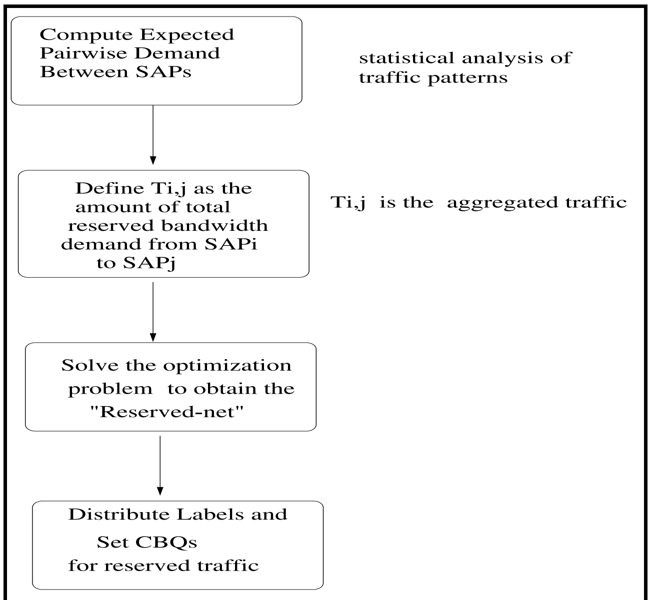

Figure 1: Steps for Provisioning and Configuration of QoS Traffic

Given the pairwise statistical demands for QoS traffic over a fixed interval, we formulate the problem designing an optimal virtual network as a multi-commodity problem with integer variables. The objective function is to *pack* the commodity-flows into a minimal feasible subgraph while avoiding bandwidth fragmentation on the links. An SBoX router is associated (activated) with each node that appears in the solution of this optimization problem. The solution to the optimization problem specifies the amount of bandwidth allocation necessary for each commodity. Bandwidth allocation and protection is done by setting CBQ mechanisms in the involved SBoX routers. The parameters of the CBQ scheduling and the labels of macro-flows are signaled with a light weight protocol which aims to minimize the communication overhead.

This paper is organized as follows. In Section 2 we address provisioning and configuration issues. Section 3 presents the components of the SBoX architecture. In Section 4 we explain the prototype implementation and experimental results. Finally we conclude in Section 5.

2 Network Provisioning

We consider the IP-based Internet as an interconnection of autonomous systems in which a provider has control of all the resources (i.e., it is a single administrative domain). An AS can be accessed through some specific entry points called *Service Access Points* (SAPs). A provider aggregates the user traffic at the SAPs. The network can be connected to the other networks via border routers (BRs) that are treated as SAPs in our model. SAPs are involved with admission control and perform policy based bandwidth management at network boundary. The discussion in this work is focused on a single AS.

We distinguish between two types of traffic carried by the network: (1) reserved or committed QoS traffic, and (2) non-reserved or best-effort traffic. Network resources are provisioned and configured to accommodate to the QoS traffic. Thus, provisioning and configuration require estimating pairwise bandwidth demands for QoS traffic between all SAPs and BRs in an AS. There are several techniques for source modeling that can be adapted for this purpose [ML97]. Provisioning and configuration of QoS traffic in the SBoX architecture are achieved by creating a virtual premium network called the **reserved-net**. The reserved-net is a virtual dedicated network that connects all the SAPs and BRs. It is used to (i) protect the reserved traffic from the best effort traffic, and (ii) forward QoS traffic using MPLS. The best-effort traffic may be permitted to use the reserved-net in addition to the second subnetwork; however, the reserved traffic has preemptive priority. Figure 1 depicts the main steps for provisioning and configuration of QoS traffic.

We note that the problem of provisioning a reserved-net is a version of network optimization problem for which a rich literature exists (e.g., [Dov91a, Dov91b, CFZ94, ACL94, Ash95, For96, Lee95, MMR96]). However, the objective function of our problem is different as we explain in the next section.

2.1 Optimum Virtual Network

Network is represented by a graph $G = (V, E)$ where node set V is partitioned into two subsets. Set R contains the routers, set S contains the SAPs. End-points of a commodity-flow belong to the set S. The set E contains the network links, each of which has capacity (bandwidth) $c_{(i,j)}$ (bits/sec) $\quad \forall (i,j) \in E$. For simplicity we assume a symmetric model so that each direction of a link has the same capacity. The reserved-net is a subgraph $G_r \subset G$, represented as $G_r = (R', S, E')$. Let T be a traffic matrix in which an entry $t_{i,j}$ indicates the aggregated bandwidth request for QoS traffic from i to $j \quad \forall i,j \in S$. In other words $t_{i,j}$ is the sum of the bandwidth of macro-flows from SAP i to SAP j. We assume that T is based on the statistical information capturing a correlated source behavior (e.g., the traffic volume between 8-11 AM). The optimization problem can be formulated as an instance of mixed-integer multi-commodity flow problem, where a commodity k is defined for a pair of nodes $k, l \in S$ such that $t_{k,l} > 0$ and $k \neq l$. In other words, a commodity k is provided from each SAP k to another SAP l, so that k has non-zero bandwidth request to l.

Our cost measure is to *pack* the *routes* of reserved traffic together so that (i) number of links used by the solution is minimum, and (ii) **bandwidth fragmentation** is minimized for each link included into the reserved-net. There are two motivations behind using bandwidth fragmentation as a measure: (1) support policies for sharing of link bandwidth between best-effort and QoS traffic (i.e., as a part of QoS provisioning), and (2) minimize the amount of "unused" or "left-over" bandwidth of a reserved link which is smaller than any expected commodity-flow bandwidth request, if no best-effort traffic is allowed on the reserved-net.

In the formulation, we have a threshold value $\delta_{i,j}$ for each link. If unused or left-over bandwidth is less than $\delta_{i,j}$ then the variable $y_{i,j}$, which is 1 only if link (i,j) is used, contributes to the penalty term in the objective function with a coefficient α as shown in Figure 3. Note that α can be chosen to control the impact of bandwidth fragmentation on the cost function.

The variables and parameters of the optimization problem are as follows:

- Flow variable $f_{i,j}^k$ which takes a real value and indicates the amount of commodity k over the link (i,j).
- A 0-1 integer variable $x_{i,j}$ which indicates the link (i,j) is used by any commodity. It is 1 if used, 0 otherwise. In other words, $x_{i,j}$ denotes whether link (i,j) is included in the reserved-net.
- A 0-1 integer variable $y_{i,j}$ which indicates whether or not the *unused* bandwidth of the link (i,j) is below a given *fragmentation threshold* $\delta_{i,j}$.

Constraint (1) ensures the flow balance for each commodity. If the node is an intermediate node (i.e., $i \in R$ is a router), the in-flow should be equal to the out-flow. Otherwise, the node i is a macro-flow end-node and the flow into this node from commodity k is $t_{k,i}$ for each commodity k. The inequalities (2) and (3) ensure that total load on a link (i,j) does not exceed its capacity. Furthermore, the load is assigned such that the bandwidth fragmentation is minimized by the penalty term $y_{i,j}$. Inequality (3) ensures that the penalty term exists only for the links in the reserved-net. Inequalities (4) and (5) are self explanatory.

The solution to this optimization problem is a set of $x_{i,j}$ and $y_{i,j}$ values, that minimize the cost function while connecting each pair of egress points. The links with $x_{i,j} = 1$ are called **reserved-links** and the nodes incident to reserved-links are called **reserved-nodes**.

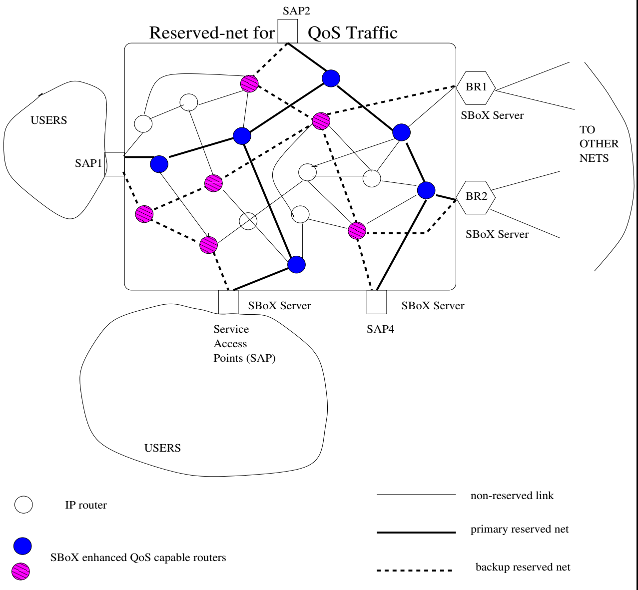

Figure 2: Reserved-Net: Virtual network for MPLS traffic.

Minimize: C$= \sum_{i,j} x_{i,j} + \alpha(y_{i,j}) \quad \forall(i,j) \in E$ and $\forall k$

Constraints:
s.t.

(1) $\sum_{j \neq i} f_{(j,i)}^k - \sum_{j \neq i} f_{(i,j)}^k = \begin{cases} t_{k,i} & \forall k, i \;\; s.t. \;\; k \neq i \\ 0 & \forall i \in R \end{cases}$

(2) $\sum_k f_{i,j}^k \leq (c_{i,j} - \delta_{i,j}) x_{i,j} + \delta_{i,j} y_{i,j} \quad \forall(i,j) \in E$

(3) $y_{i,j} \leq x_{i,j}$

(4) $x_{i,j} \in \{0,1\} \quad \forall(i,j) \in E$

(5) $f_{i,j}^k \geq 0$

Figure 3: Mixed-integer formulation of the problem

The union of the reserved-links induces a connected graph G_r which is our reserved-net as shown in Figure 2.

A path from SAP_k to SAP_l on the reserved-net is called **reserved-path**. It is the label switched path (LSP) that carries the commodity-flow for commodity k. It may be desirable to over allocate the capacity for providing fault-tolerance and dynamic admission control. A simple way to achieve this is to add slack variables to the $t_{i,j}$ and solve the above optimization problem. Note that the exact solution of the above optimization problem is computationally expensive; thus we may either use heuristics to obtain suboptimal solutions, or solve the optimization problem off-line and store it for different traffic matrices T in advance. Since our focus is on the QoS aspects of the problem, we will

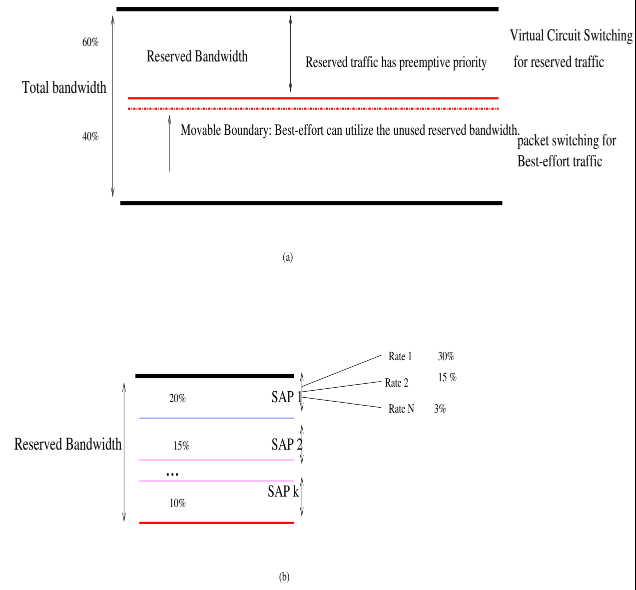

Figure 4: Bandwidth Sharing

not further elaborate on the optimization issues.

2.2 Resource Allocation

The solution to the optimization problem determines reserved-links and the amount of flow that can be assigned to these links. In particular the $f_{i,j}^k$ value indicates the bandwidth request of commodity-flow on link (i, j) that resides in the reserved-path from SAP_k to SAP_l. In order to enforce service guarantees, $f_{i,j}^k$ must be reserved and protected on link (i, j). Resource allocation in the reserved-net is done at the commodity-flow level and enforced by CBQ.

Different policies may be adapted to adjust the interaction between the reserved and the best-effort traffic. We propose a **movable boundary** scheme, in which the best-effort can use the available reserved bandwidth, where reserved traffic has preemptive priority as shown in Figure 4. The position of the boundary can be dictated by the network economics and utilization. For example, an ISP may decide to reserve only 60% of link capacity to the aggregated reserved-traffic.

The policy for setting such a boundary can be enforced in the optimization problem by the threshold variable $\delta_{i,j}$ and the coefficient α.

Distribution of MPLS labels and setting the CBQs can be done using a simple signaling protocol as presented in the next section.

2.3 Light Weight Signaling Protocol

Currently two protocols are being discussed to set a LSP (i.e., to distribute the labels and make the reservations at LSR): RSVP extension, and CR-LDP [ea99a]. The RSVP has the softstate overhead and lacks a reliable protocol (it is based on IP) for distribution. Thus, its response delay is higher. In contrast, the CR-LDP uses TCP for reliable delivery but is subject to all the problems associated with the performance of TCP. This section presents a generic signaling protocol, based on 3-way handshake, with the following steps:

1. SAP_x makes a request of R bps to a SAP_y for the corresponding commodity-flow by
   - 1.1 sending $REQ(Label, R, seq\#)$ message.
   - 1.2 starts a timer for reply.
2. each SBoX router on the reserved-path
   - 2.1 marks the forwarding table for rate R and *Label* and
   - 2.2 appends its ID to the REQ message.
   - 2.3 marking is not a commitment but a tentative state. It is associated with a timer and if no reply (REP) message is received, then there will be timeout and the tentative commitment will be reset.
3. upon receiving the REQ message, the SBoX server at SAP_y
   - 3.1 sends $REP(Label, R, seq\#)$ message back by reversing the reserved-path recorded into the REQ message.
   - 3.2 sets a timer for confirmation (CONF) message
4. each SBoX router on the path
   - 4.1 identifies the REQ/REP using the Label,
   - 4.2 *commits* for the rate R that REP message carries,
   - 4.3 unmarks the REQested rate
   - 4.4 starts a timer for confirmation
5. upon receiving the REP message, the SAP_x confirms the reserved-path by
   - 5.1 sending a $CONF(Label, R)$ message to SAP_y or
   - 5.2 starts transmitting data.
6. if the timer in SAP_x times out for the REQ message then it resends a REQ message with an incremented seq #.
7. if an SBoX router receives a redundant REQ message and
   - 7.1 it has marked the forwarding table then assumes that previous REQ got loss at the segment between itself and SAP_y
   - 7.2 it has also committed then assumes that REP got loss at the segment between itself and SAP_x
   - 7.3 takes no action and waits the edge routers to response.
8. upon committing to a REQ message if an SBoX router receives no CONF message or data then it time outs and releases the commitment.
9. each router maintains the commodity-flow until a TEARDOWN message is received.

Notice that QoS requirement is expressed simply as the amount of the bandwidth. The traffic specifications such as burst size burst length are omitted since they are controlled at the egress points by SBoX servers. Furthermore, constructing a virtual premium network (i.e., the reserved-net) ensures that for any reserved-link (i, j), $\sum_k f_{i,j}^k \leq c_{i,j}$.

3 SBoX Architecture

The SBoX architecture comprises two components: (1) SBoX servers, and (2) SBoX routers. The SBoX servers involve with admission control and management at the network edge. SBoX routers are the interior network nodes that perform only label switching of macro

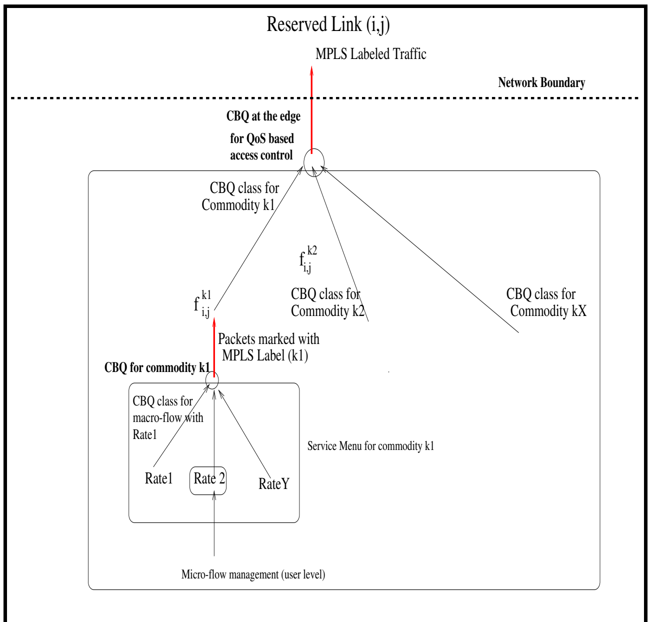

Figure 5: CBQ hierarchy and labeling maintained by an SBoX Server

flows for QoS based routing. We will describe the SBoX servers and routers in detail in this section.

3.1 SBoX Servers

The SBoX architecture deals with the complexity of QoS support by delegating it to the SBoX servers. An SBoX server is an **intelligent edge-router** that resides at network egress point (i.e., SAP) and provides QoS based network access. A SBoX server is a programmable device with an open architecture, and it is in charge of managing the resources and requests from SAPs. It separates QoS traffic from best-effort and labels the QoS traffic for MPLS routing inside the network. An SBoX server uses hierarchical traffic management mechanisms which can be configured dynamically to adapt to the changing policies of an ISP. We note that since the adjacent AS are treated as SAPs, their resource management is also handled by SBoX servers.

3.1.1 Flow Management and Labeling

SBoX server is a configurable device which has hierarchical traffic management capability. Figure 5 depicts the hierarchical management of a reserved-link (i,j) by an SBoX server. At the top of the hierarchy, the bandwidth of (i,j) is shared by multiple commodity-flows whose bandwidth allocation is determined by the optimization problem presented in section 2.1. Link (i,j) needs to be managed to ensure: (1) protection of commodity-flows from best effort, and (2) protection of each commodity-flow from each other. The second level of the hierarchy concerns with the management of each commodity-flow. Each commodity-flow is divided into a set of macro-flows which are offered as SLAs to users of a SAP for this commodity. Thus, each SLA provides a predefined service rate which are offered by a **service menu**. In the last level, macro-flows are managed in application level flow (i.e., micro-flow) granularity. We will explain the service menus and management of macro-flows in the next section.

SBoX servers performs IP-MPLS mapping at the commodity-flow level. Each commodity flow is associated with an MPLS label which can be global within the AS [2]. Packets of each commodity-flow are marked by the MPLS label. As shown in Figure 5, the SBoX server maintains a CBQ scheduling mechanism for each commodity-flow that uses edge (i,j). Each commodity is considered as a class and classes are strictly isolated from each other (i.e., fire-walled). The resource sharing between best-effort and QoS traffic is based on the movable boundary scheme as explained above. Packets of each commodity are filtered and scheduled based on their MPLS labels.

An SBoX server also maintains billing and accounting information at different levels of granularity. It can provide call description record (CDR) type of information at the commodity-flow or at the macro-flow level. We will explain billing and charging issues at the next section further.

3.1.2 Admission Control with SLAs

The committed traffic is accepted to the network through admission control, which ensures that bandwidth allocation satisfies the service descriptions of user requests. An important part of admission control is monitoring and managing the *available bandwidth* for each commodity.

The SBoX architecture provides various data rates called **service rates**, that are a multiple of a base rate (e.g., 1Mbps, 5Mbps, 10Mpbs, 20Mbps) in a service menu. Each data rate represents a different SLA. Users sign a contract with their provider by picking an SLA from a service menu.

As shown in Figure 5, each SLA corresponds to a macro-flow. It is left to the users/enterprises to manage the access to the pipe. There are several commercial products available to differentiate business critical

[2] Note that 20 bits from the 4-byte MPLS label can support 2^{20} commodities which corresponds to approximately 10^3 SAPs within an ASP. If the number of SAPs exceeds such number then labels must be swapped and cannot be global.

traffic based on enterprise policies (e.g., Xedia's Access-point, Packeteer's Packetshaper, Checkpoint's Flood Gate, Allot's AC 200-330).

The challenge is how to decide which service rates to offer and how many requests to grant for each (macro-flow) rate. We believe that the answer should have low overhead and also address the network economics. For example, each service rate in the service menu can be assigned a number of **tokens** (permits) during provisioning. The weighted sum of the tokens is equal to the total amount of reserved traffic that the provider is willing to accept (where the weights represent the data rates). Each SAP and BRs are given a set of tokens based on (i) the expected number of connections, and (ii) the bandwidth demands between them. Thus, each SAP/BR can make a local decision how much traffic to accept to the reserved-net using the token/permit pool it has. For example, lets refer to Figure 5. Suppose that the SAP is given x tokens of 512Kbps and y tokens of 1Mbps for commodity k_1. Then this SAP can accept at most $x + y$ requests for this commodity such that $x512 + y1000 \leq f_{i,j}^{k_1}$ kbps.

There are two advantages of providing SLAs with token based admission control. First, it reduces the traffic management overhead at the commodity-flow level. Since users must pick an SLA from a service menu in advance, the policing, shaping and scheduling in the network can be set ahead of time, even before the SLAs are offered. In contrast, if we replace the predetermined rates with a dynamic management scheme then requests for macro-flows must be accommodated on demand basis. This requires setting the traffic filters and shapers on demand as well. Thus, it may increase the processing time for requests and response delay. Second, it provides efficient billing and accounting mechanism to an ISP. For example, each token can have a price based on the rate that it is associated with. Thus charging can be reduced to the token holding time multiple with the price of this token. The price of a token can be changed based on the demand for bandwidth. For example a discount price can be offered during off-peak hours while price can be increased during peak hours.

3.2 SBoX Routers

An SBoX router is a programmable, stand-alone, label switching router with a limited number of network interfaces. An SBoX router cannot perform IP-based routing and its routing capability is limited only to label switching with CBQ scheduling.

Since the legacy routers are vertically integrated

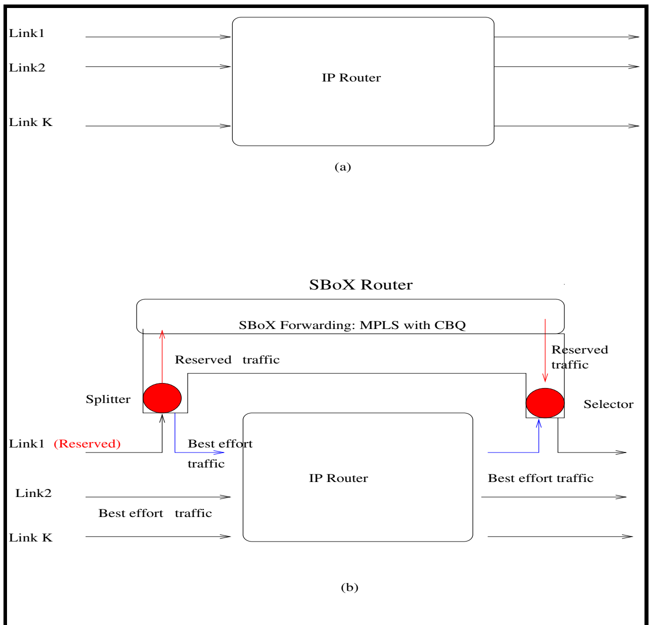

Figure 6: SBoX Operation: QoS traffic has precedence over best-effort traffic. Best-effort traffic uses either links that are not a part of the reserved-net or the idle portion of reserved link. Combination of SBoX and best-effort router provides functionality found in a *dual-switch* [Kum74, Zaf74, OY94, NBS95].

closed boxes, an SBoX router may be deployed as an add-on device [3]. An SBoX router controls the reserved links, identified by the reserved-net, incident to a legacy router, as shown in Figure 6.

The number of network interfaces and the line speeds that an SBoX supports are dictated by the network economics and considered to be outside the scope of this paper. However such constraints can be accommodated into the formulation of the optimization problem to limit the number of reserved links incident to a legacy router that can be in the solution. The main functions of the SBoX are summarized in Figure 7.

Some of these functions are performed by two network interface modules called the **splitter**, and the **selector**. The splitter intercepts all incoming packets, examines their header information, and decides where to forward the packet (outgoing link for MPLS cut-through, or an input link of the best-effort router. The splitter performs the above **Fget**, **Fchk**, **FcuthruS**, **FcuthruR**, and **Fdrop** operations. The splitter uses a buffer to store the initial portion of incoming packets until the label is processed. If the packet does not carry

[3] Certainly, it would be the best if an SBoX can be transplanted into a legacy router however current legacy routers do not provide such modular and open architectures.

Fget: intercept the traffic into the *reserved node*;
- **Fchk:** check the header to determine its type;
- **FcuthruS:** IF the packet belongs to a macro flow then cut through to the SBoX;
- **FcuthruR:** ELSE forward the packet to the same input link incident to the router.
- **Fdrop:**(optional) if the packet is not a IP control packet.
- **Fnext:** check the label and the outgoing link;
- **Fvc:** perform MPLS compliant virtual circuit switching with per commodity-flow CBQ scheduling.
- **Fblock:** prevent the router to access the output link if the SBoX has a packet to be transmitted to it as well.

Figure 7: SBoX Router Operations

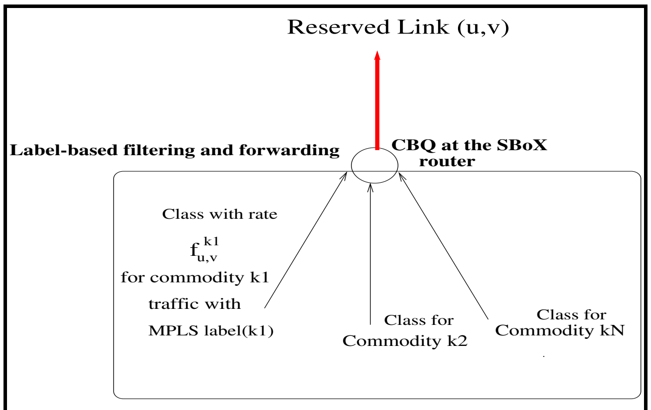

Figure 8: CBQ Class hierarchy in an SBoX router

a label, then it is considered as a best-effort, and it is forwarded immediately to the best-effort router without waiting and storing the rest of the packet. The size of the buffer is a function of the link rate and the splitter processing speed.

Contention occurs if best-effort router is allowed to forward packets to a reserved link. Thus we need to control the access to the output link. The selector is an arbitrator circuit, which ensures that as long as there is data in the SBoX's output buffer, the reserved link cannot be used by the legacy router. The splitter is functionally a a 2x1 multiplexer. The inputs of the multiplexer are the SBoX and the best-effort router. The multiplexer selects the SBoX as long as it has data. The splitter and selector reside inside the SBoX and their functions can be combined in an interface card with two ports and a limited processing capability.

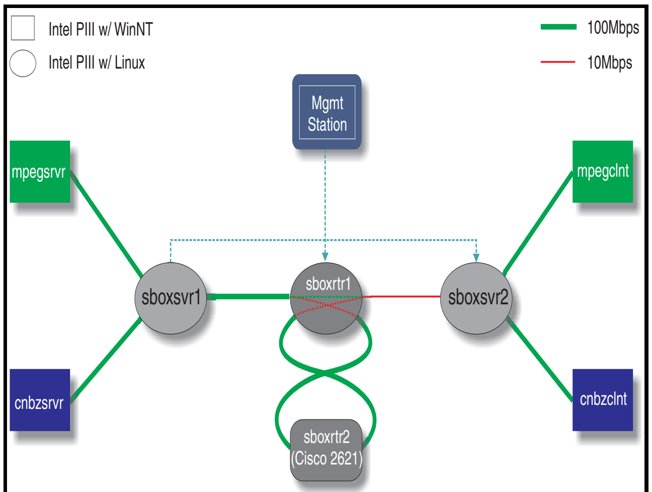

Figure 9: SBoX Prototype Testbed.

3.2.1 Flow Management and Forwarding

While the best-effort router performs IP-based routing, packet forwarding in an SBoX is based on label switching using MPLS. Upon entering the network, QoS traffic is handled exclusively by the SBoX routers. Thus it bypasses all best effort traffic. Each SBoX router has a **forwarding table**, associated with each link. An entry in the forwarding table indicates the outgoing link and the reserved bandwidth for a commodity-flow identified by its MPLS label.

The protection and enforcement of service guarantees is done with CBQ as shown in Figure 8.

4 SBoX Testbed Prototype

The testbed for SBoX project consists of seven Intel Pentium IIIs and a Cisco 2621 router connected with the topology depicted in Figure 9. Four of the seven PIIIs are running Windows NT, a pair of which for the Lucent MPEG video streaming (*mpegsrvr* and *mpegclnt*) and the other pair of which for the Lucent Cineblitz video streaming (*cnbzsrvr* and *cnbzclnt*). Three of the remaining PIIIs are running Linux with MPLS and CBQ support compiled into the kernel, two of which are the SBox servers (*sboxsvr1* and *sboxsvr2*) and one of which is an SBox router (sboxrtr1) managing commodity-flows inside the network. The Cisco 2621 router (sboxrtr2) plays the role of a conventional IP router for best-effort traffic.

One MPEG video stream of approximately 4Mbps flows from *mpegsrvr* to *mpegclnt*. Multiple cineblitz video streams, each about 1.5Mbps, flow from *cnbzsrvr* to *cbnzclnt*. These video streams are the premium

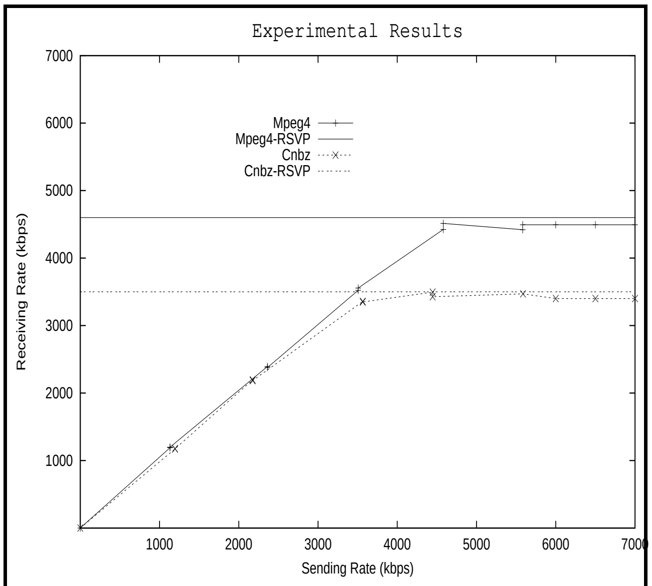

Figure 10: Behavior of reserved traffic

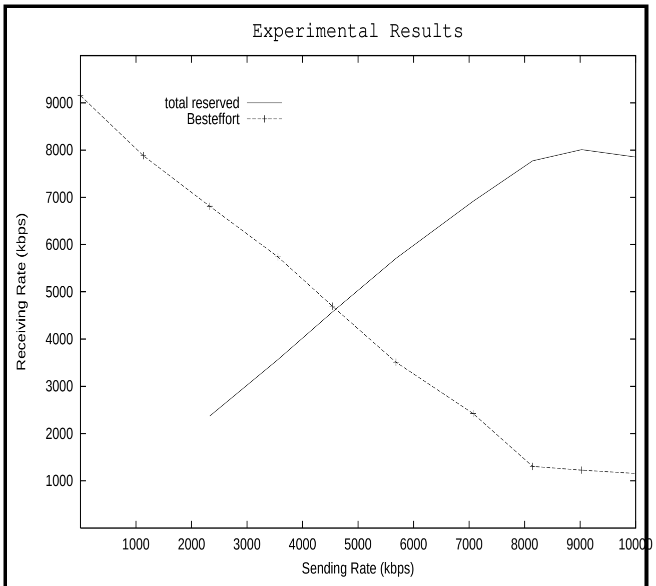

Figure 11: Behavior of best effort and reserved traffic

QoS traffic and can be selectively protected with guaranteed bandwidth. Premium traffic are label-switched across the network between *sboxsvr1* and *sboxsvr2*. A best-effort UDP traffic, with adjustable bandwidth consumption, flows from *sboxsvr1* to *sboxsvr2* as the background noise to create congestion. The noise traffic is not label-switched; it is routed through regular IP routing.

The noise traffic entering *sboxrtr1* is diverted to the Cisco router and then looped back to *sboxrtr1* before it is routed to its final destination. This is to simulate (1) the Split and Select operation, and (2) the ability of SBoX to inter-operate with existing IP routers without affecting the existing IP traffic. Note that all traffic goes through the bottleneck link, the 10Mbps link between *sboxrtr1* and *sboxsvr2*.

4.1 Experimental Results

Using the testbed implementation we conducted experiments to examine the performance of the SBoX architecture. The SBoX router *sboxrtr1* manages two commodity-flows and it is connected to the *sboxsvr2* with a full-duplex bottleneck link with capacity 10 Mbps as mentioned above.

We observe in Figure 10 that MPLS compliant QoS traffic for both commodities are protected from each other. Furthermore, their total sending rate is bounded by 8Mbps which is the total reserved bandwidth on this bottleneck link. Figure 11 shows that the best-effort traffic takes all the unused reserved bandwidth thus maintaining a high link utilization. As the total reserved traffic amount increases, the best-effort rate reduces to the unreserved portion of the link bandwidth. This behavior verifies that the *movable boundary* scheme works efficiently for providing high utilization. Furthermore it also shows that QoS traffic is protected from the best-effort.

5 Discussion

In this work we presented a hybrid architecture to provide deterministic QoS on the IP-based networks. The architecture combines the best features of Diffserv, Intserv, and MPLS proposals. At the core of the proposed architecture, precise provisioning and admission control algorithms are proposed. These algorithms determine a virtual premium network that is used to handle QoS traffic. The complexity of micro-flow management is done at the network boundary by intelligent edge routers called the SBoX servers. The service level agreements offered at the network boundary are supported by add-on label switching routers called the SBoX routers inside the network. A prototype testbed is implemented and experiments indicate that the proposed architecture delivers its features.

Acknowledgments

The author wishes to thank Eran Gabber for valuable discussions and to Gong Su for the implementation of the test bed.

References

[ACL94] G. R. Ash, K. K. Chen, and J-F. Labourdette. Analysis and design of fully shared networks. *Proc. of 14'th International Teletraffic Congress*, pages 1311–1320, 1994.

[Ash95] G. R. Ash. Dynamic network evolution, with examples from at&t's evolving dynamic network. *IEEE Communications Magazine*, 33(7):26–39, 1995.

[BW99] A. Basu and Z. Wang. Fair bandwidth allocation for differentiated services. *in Protocols for High Speed Networks VI, J. Touch and J. Sterbenz ed. Kluwer Academic Publishers*, 1999.

[CFZ94] I. Chlamtac, A. Farago, and T. Zhang. Optimizing the system of virtual paths. *IEEE Trans. on Networking*, 2(6):581–587, Dec. 1994.

[Cla97] D. Clark. Internet cost allocation and pricing. *Internet Economics, L. W. McKnight and J. P. Bailey (eds)*, pages 215–252, 1997.

[CW97] D. Clark and J. Wroclawski. An approach to service allocation in the internet. *Internet Draft, draft-clark-diff-svc-alloc-00.txt*, July 1997.

[Dov91a] R. D. Doverspike. Algorithms for multiplex bundling in a telecommunications network. *Operations Research*, 39:925–944, 1991.

[Dov91b] R. D. Doverspike. A multi-layered model for survivavility in intra-lata transport networks. *Proc. IEEE GLOBCOM'91*, pages 2025–2031, 1991.

[ea98] S. Blake et al. A framework for differentiated services. *draft-ietf-diffserv-framework-0.1.txt*, October 1998.

[ea99a] B. Jamoussi et al. Constraint-based lsp using ldp. *Draft-ietf-mpls-cr-ldp-03.txt*, September 1999.

[ea99b] I. Stoica et. al. Per hop behaviours based on dynamic packet states. *internet draft. http://www.cs.cmu.edu/ istoica/DPS/draft.txt*, February 1999.

[For96] ATM Forum. P-nni specification, version 1.0. March 1996.

[IN98] J. Ibanez and K. Nichols. *internet draft : draft-ibanez-diffserv-assured-eval-00.txt*, August 1998.

[Kum74] K. Kummerle. Multiplexer performance for integrated line and packet switched traffic. *Int. Conf. Comput. Commun. Record*, pages 507–515, August 1974.

[Lee95] W. C. Lee. Topology aggregation for hierarchical routing in atm networks. *Comput. Commun. Rev. (USA)*, 25(2):82–92, 1995.

[ML97] H. Michiel and K. Laevens. Teletraffic engineering in a broad-band era. *Proc. of IEEE*, 85(12):2007–2033, December 1997.

[MMR96] D. Mitra, J. A. Morrison, and K. G. Ramakrishnan. Atm network design and optimization: A multirate loss network framework. *Proc. IEEE INFOCOM'96*, pages 994–1003, 1996.

[NBS95] A. Nguyen, N. Bambos, and M.H. Sherif. Adaptive (t1, t2)-multiplexing transmission schemes for voice/data integrated networks. *IEEE Syp. Comp. and Communications*, pages 430–435, 1995.

[NJZ97] K. Nichols, V. Jacobson, and L. Zhang. A two bit differentiated services architecture for the internet. *Internet Draft, draft-nicolas-diff-svc-arch-00.txt*, November 1997.

[OY94] Y. Ofek and M. Yung. The integrated MetaNet architecture: A switch-based multimedia LAN for parallel computing and real-time traffic. *IEEE INFOCOM'94*, 1994.

[RVC98] E. C. Rosen, A. Viswanathan, and R. Callon. Multi protocol label switching. *draft-ietf-mpls-arch-02.txt*, July 1998.

[SPG97] S. Shenker, C. Partridge, and R. Guerin. Specification of guaranteed quality of service. *rfc2211*, September 1997.

[SSZ98] I. Stoica, S. Shenker, and H. Zhang. Core-stateless fair queueing: A scalable architecture to approximate fair bandwidth allocations in high-speed networks. *Proc. of ACM SIGCOMM'98*, August 1998.

[SW97] S. Shenker and J. Wroclawski. General characterization parameters for integrated services network elements. *rfc2215*, September 1997.

[SZ98] I. Stoica and H. Zhang. Lira: An approach for service differentiation in the internet. *Proc. of NOSSDAV'98, Cambridge, England*, 1998.

[Wro97] J. Wroclawski. Specificiation of the controlled-load network element service. *rfc2211*, September 1997.

[Zaf74] P. Zafiropolo. Flexible multiplexing for networks supporting line switched and packet switched data traffic. *Int. Conf. Comput. Commun. Record*, pages 517–523, August 1974.