Edge Provisioning and Fairness in VPN-Diffserv Networks

Ibrahim Khalil, *Torsten Braun*
Institute of Computer Science and Applied Mathematics (IAM)
University of Berne
Neubrückstrasse 10, CH-3012 Bern, Switzerland
ibrahim,braun@iam.unibe.ch

Abstract— **Customers of Virtual Private Networks (VPN) over Differentiated Services (Diffserv) infrastructure are most likely to demand not only security but also guaranteed Quality of Service (QoS) as there is a desire to have leased line like services. However, it is expected that they will be unable or unwilling to predict load between VPN endpoints. In this paper, we propose that customers specify their requirements as a range of quantitative service in the Service Level Agreements (SLAs). To support such services ISPs would need to have automated provisioning system that can logically partition the capacity at the edges to various classes (or groups) of VPNs and manage them efficiently to allow resource sharing among the groups in a dynamic and fair manner. While with edge provisioning certain amount of resources based on SLAs (traffic contract at edge) are allocated to VPN connections, we also need to provision the interior nodes of a transit network to meet the assurances offered at the boundaries of the network. We, therefore, propose a two-layered model to provision such VPN-Diffserv Networks where the top layer is responsible for edge provisioning and drives the lower layer in charge of interior resource provisioning with the help of a Bandwidth Broker (BB). Various algorithms with examples and analysis are presented to provision and allocate resources dynamically at the edges for VPN connections. We have developed a prototype BB performing the required provisioning and connection admission.**

I. Introduction

There is a growing demand that since private networks built on using dedicated lines offer guaranteed bandwidth and latency, similar guarantees be provided in IP based Virtual Private Networks (VPNs) [9], [14]. While the internet has not been designed to deliver performance guarantees, with the advent of differentiated services [3], [2], IP backbones can now provide various levels of quality of service.Recently proposed Expedited Forwarding (EF) [10] Per Hop Behaviour (PHB) is the recommended method of build such an Virtual Leased Line Line (VLL) type point-to-point connection for VPN.

To provide such service we have (and others, for example [16],[17]) recently implemented [12], [13] a Bandwidth Broker [15] that allows an user to specify a single quantitative value (i.e 1 Mbps or 2 Mbps etc.) and based on this specification the edge routers establish VPN connections dynamically. However, it is expected that users will be unable or unwilling to predict load between VPN endpoints [7]. From the provider's point of view also, guaranteeing exact quantitative service might be a difficult job at the beginning of VPN-Diffserv deployment [2]. We, therefore, propose that users specify their requirements as a range of quantitative service. For example, a user who wants to establish a VPN between stub Networks A and B (Figure 1), and is not sure whether he needs 0.5 Mbps or 0.6 Mbps or 1 Mbps, and only knows the lower and upper bounds of his requirements approximately, can specify a range 0.5- 1 Mbps as his requirement from the ISP when he outsources his service to the latter. An ISP can offer multiple such options via a website (Figure 6) to help customers to select any suitable option to activate services dynamically on the fly.

This has several advantages: Users do not need to specify the exact capacity but it gives the flexibility to specify only a range. The price that customers have to pay is higher than one pays for the lower bound capacity but lower than what is normally needed to be paid for upper bound capacity. During low load it is possible that users might enjoy the upper bound rate (say 1 Mbps in the example) without paying anything extra. This kind of pricing might be attractive to users and ISPs can take advantage of that to attract more customers without breaking the commitment.

This, however, poses significant challenge to the ISPs as they would need to deploy automated provisioning system that are able to logically partition the capacity at the edges to various classes (or groups where each group is identified from it's offer, for example 0.5- 1 Mbps could represent one group, 1-2 Mbps could represent another) of VPNs and manage them efficiently to allow resource sharing among the groups in a dynamic and fair manner.Also, they must provision the interior nodes in the network to meet the assurance offered at the boundaries of the network We have, therefore, proposed a two-layered model in section II to provision such VPN-Diffserv Networks where the top layer is responsible for edge provisioning and drives the lower layer in charge of interior resource provisioning with the help of a Bandwidth Broker (BB).

We have restricted this paper to edge provisioning only considering the fact that most of complexities lie at the boundaries of the network and is the main driving force for overall provisioning. In section III various algorithms with examples and analysis have been presented to provision and allocate resources dynamically at the edges. Fairness issues while allocating unused resources have been addressed in section III-D. A prototype BB performing the required provisioning and connection admission has been described in section IV. Section V concludes the paper with a summary of our contributions and a discussion of future research directions.

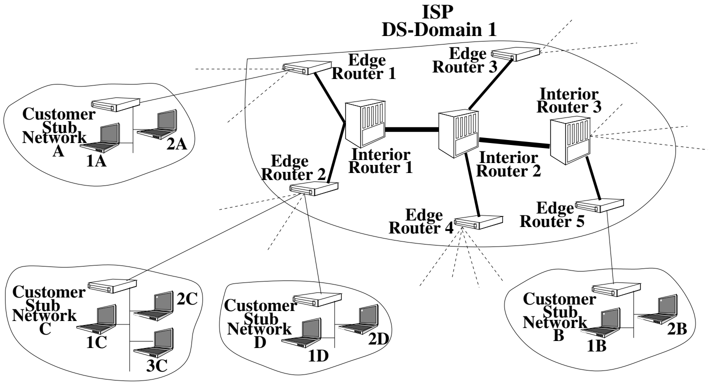

Fig. 1. VPN Diffserv deployment scenario

II. Provisioning requirements for VPN-Diffserv Networks: A Model

Provisioning in Diffserv Networks does not only mean determination and allocation of resources necessary at various points in the network, but also modification of existing resources to be shared dynamically among various VPN classes (i.e. groups). Both quantitative, as it is the case with VPNs, and qualitative traffic (some assured service) are required to be provisioned at the network boundaries and in the network interior.

Determination of resources required at each node for quantitative traffic needs the estimation of volume of traffic that will traverse each network node. While an ISP naturally knows from the SLA the amount of VPN quantitative traffic that will enter the transit network through a specific edge node, this volume cannot be estimated with exact accuracy at various interior nodes that will be traversed by VPN connections if we do not know the path of such connections [1]. However, if the routing topology is known, this figure can be almost accurately estimated. If the default path doesn't meet the requirements of an incoming connection, alternate and various QoS routing [6], [5] can also be used to find a suitable path and enforced by MPLS techniques [8].

A. Role of Bandwidth Broker for Automated Provisioning

Based on the basic needs of provisioning a VPN-Diffserv network to support quantitative service we consider the provisioning as a two layered model - the top layer responsible for edge provisioning and driving the bottom layer which is in charge of interior provisioning (Figure 2). As we seek to provide a system where VPN services are available on demand, we find that Bandwidth Broker [15],[17] is the right choice, because it is not only capable of performing dynamic end-to-end admission control to setup a leased line like VPN by maintaining the topology as well as policies and states of all nodes in the network, but also capable of managing and provisioning network resources of a separately administered DS domain and cooperating with other similar domains.

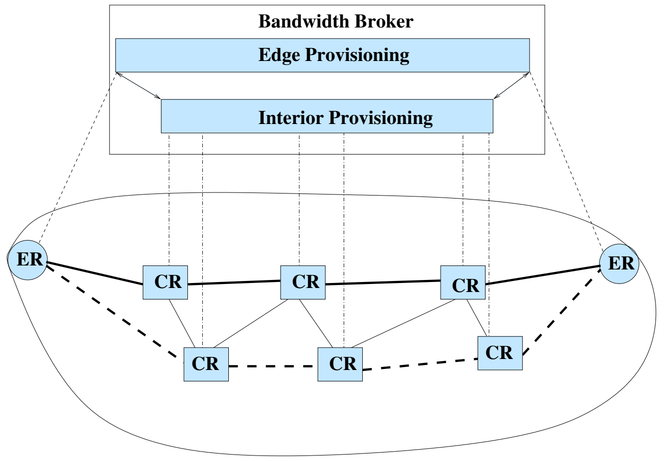

Fig. 2. Layered Provisioning view of VPN-Diffserv Networks

B. A Novel Approach: Bandwidth Specified as an Interval

To overcome users' difficulty in specifying the exact amount of quantitative bandwidth required while outsourcing the VPN service to ISP our model supports a flexible way to express SLAs where users specify a range of quantitative amounts rather than a single value. Although it has several advantages, this also makes the edge and interior provisioning difficult. This complexity can be explained with a simple example. Referring to Figure 1 once again, assume that edge router 1 has been provisioned to provide 20 Mbps quantitative resources to establish VPN connections elsewhere in the network and ISP has provided two options via a web interface to the VPN customers to select the rate of the connections dynamically: 1 Mbps or 2 Mbps. It is easy to see that at any time there can be 20 connections each having 1 Mbps, or 10 connections each enjoying 2 Mbps, or even a mixture of the two (e.g. 5 connections with 2 Mbps, 10 connections with 1 Mbps). When a new connection is accepted or an active connection terminates, maintaining the network state is simple and doesn't cause either reductions or forces re-negotiations to existing connections. If there are 20 connections of 1 Mbps, and one connection leaves then there will be simply 19 connections of 1 Mbps. Admission process is equally simple.

Now if the ISP provides a new option (for example, as shown in Figure 6(b)) by which users can select a range 1Mbps - 2 Mbps (where 1 and 2 are the minimum and maximum offered guaranteed bandwidth), maintaining the state and admission control can be difficult. A detailed example can be found in section III-B.When there are up to 10 users each connection would get the maximum rate of 2 Mbps, but as new connections start arriving, the rate of existing connections would decrease. For example, when there are 20 connections this rate would be $\frac{20}{20} = 1$ Mbps and then at that stage if an active connection terminates the rate of every single connection would be expanded from 1 Mbps to $\frac{20}{19} = 1.05$ Mbps. This is a simple case when we have a single resource group supporting a range 1Mbps-2 Mbps. In reality, we might have several such groups as shown in Figure 6(b). In such cases, renegotiation for possible expansion of existing connections, admission control and maintenance of network states will not be simple. The idea presented here is illustrated in figure 3.

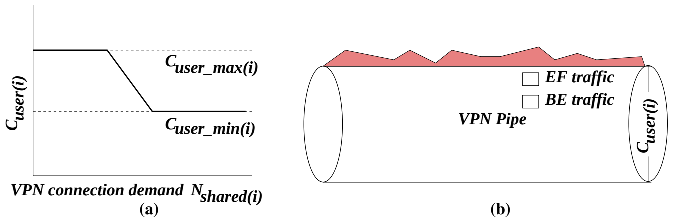

Fig. 3. The SLA approach: (a) Bandwidth is specified as an interval of $C_{user\_min(i)}$ and $C_{user\_min(i)}$ for any group i. Actual rate of a VPN connection $C_{user(i)}$ varies between this range but never gets below $C_{user\_min(i)}$. (b) $C_{user(i)}$ is the rate that is configured in the edge router as the policing rate. Traffic submitted at a rate higher than this rate is marked as best effort traffic or dropped depending on the policy

C. The Model and Notations

In our model, we address this novel approach to SLA and provide policies and algorithms for automated resource provisioning and admission control. However, to support such provisioning, we first start by allocating a certain percentage of resources at each node (edge and interior) to accommodate quantitative traffic. At the edge this quantitative portion is further logically divided between dedicated VPN tunnels (i.e. require 1Mbps or 2 Mbps explicitly) and those connections that wish to have rates defined by a range (i.e 0.5-1 Mbps or 1-2 Mbps etc.). This top level bandwidth apportionment is shown in Figure 4. The notations are :

- C_T is the total capacity of a node interface.
- C_{ded} is the capacity to be allocated to VPN connections requiring absolute dedicated service
- C_{shared} is the capacity apportioned for those VPN connections who describe their requirement as a range.
- C_{qual} is the remaining capacity for qualitative traffic.
- C_{quan} is the capacity provisioned for quantitative traffic and is equal to (C_{ded}+ C_{shared}).

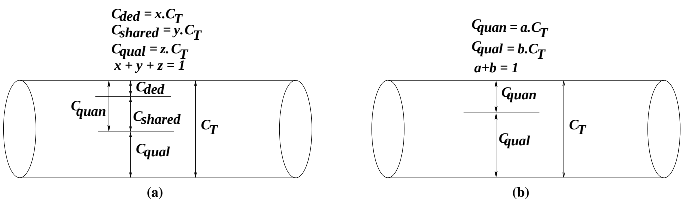

Fig. 4. Top level Bandwidth Apportionment: (a) logical partitioning at the edge, (b) logical partitioning at an interior

While at the edge C_{quan} is rate controlled by policing or shaping, at the interior this C_{quan} indicates that this amount of capacity will be allocated (actually protected) to quantitative traffic if need arises. All the values can be different at different nodes. This kind of logical partitioning is helpful because capacity is never wasted even if portions of resources allocated to quantitative traffic are not used by VPN connections. Unused capacity naturally goes to qualitative portion and enhances the best effort and other qualitative service. This is true both at the edge and in the interiors. C_{shared}, as shown in Figure 4, can be logically divided to multiple groups where each group supports a different range (Figure 5). As there might be multiple of such groups, for any group i we define the following notations:

- $C_{base(i)}$ is the the base capacity for group i which is shared by the VPN connections belonging to that group.
- $C_{user\_min(i)}$ is the ISP offered minimum guaranteed bandwidth that a user can have for a VPN connection.
- $C_{user\_max(i)}$ is the ISP offered maximum guaranteed bandwidth that a user can have for a VPN connection.
- $N_{shared(i)}$ is the current number of shared VPN connections in group i
- $C_{shared(i)}$ is the amount of capacity currently used by group *i*.
- $C_{user(i)}$ is the actual rate of active connections in group i and is equal to $\frac{C_{shared(i)}}{N_{shared(i)}}$ (in section 3).
- C_{shared_unused} is the total unused bandwidth from all shared service groups.

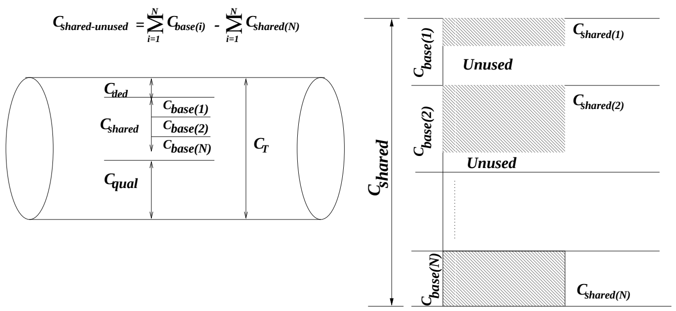

Fig. 5. Microscopic View of Bandwidth Apportionment at Edge

There are numerous sharing policies that we can apply to these shared service groups. We call them shared service groups because in reality the base capacity is shared by a certain number of VPN connections and sharing policy might allow a group to share it's resources not only among it's own connections, but also share with other groups' VPN connections in case there is some unused capacity. This may also apply to dedicated capacity. Priority can be given to certain groups while allocating unused resources. Actually, fair sharing is a challenging problem, and we will address all these issues in the following sections while developing provisioning mechanisms.

III. EDGE PROVISIONING POLICIES: ANALYSIS AND ALGORITHMS

Based on the model described in section II, various allocation policies could be adopted by the ISPs at the ingress point to allocate capacity dynamically to maintain and guarantee the quality of service of various types of incoming and existing VPN connections as we will have multiple classes of VPNs each supporting different bandwidth specifications. Some suitable policies are :

- Policy I: Capacity unused by one group cannot be used by any other groups. This means that if we have multiple shared service groups, one group whose resources have been exhausted while supporting numerous connections doesn't borrow resources from others even when those groups have unused capacity.Also, none of the groups are allowed to use unused capacity of dedicated service group.
- Policy II: Capacity unused by one shared service group can be borrowed by another shared service group. However, like the previous policy, they are not supposed to borrow from the dedicated service group.
- Policy III: Capacity unused by dedicated service group can be borrowed by tunnels of shared service groups. Also, these groups can share resources among themselves.

In this section, we will start with VPN Connection Acceptance algorithms at Ingress point where all admission complexities lie. This complexities are introduced because of the need to partition and share resources to support our model and policies presented above. Further analysis with examples of algorithms for Policy I,II and III clarify those in detail.

A. VPN Call Acceptance at Ingress

The job of admission control is to determine whether a VPN connection request is accepted or rejected. If the request is accepted, the required resources must be guaranteed. For any group i a new VPN establishment request is admitted only if at least the minimum bandwidth as stated in the offer can be satisfied while also retaining at least the minimum requirements for the existing users, i.e. if $\left(N_{shared(i)} \leq \frac{C_{base(i)}}{C_{user\_min(i)}}\right)$ a new VPN connection request can be accepted. This ensures that, an admitted VPN connection will always receive at least the minimum offered bandwidth $C_{user\_min(i)}$ in group i by restricting the number of maximum connections that can join the group. How much capacity the accepted connection will actually have is decided by connection state in that group and sharing policies that we are going to discuss in the next subsections.

B. Capacity Allocation with no sharing among groups: Policy I

The base capacity allocated to a group is solely used by the VPN connections belonging to that group only. Under no circumstance resources assigned to one group can be borrowed by others, even if that capacity is unused. This makes allocation simple not only at the edges, but also in the interior and from an implementation point of view it is simple. Since the unused capacity is not used by any other groups, qualitative services, as we mentioned earlier, are also enhanced.

If a VPN connection is accepted the system checks if that connection can be allocated the maximum rate. This is possible if the base capacity $C_{base(i)}$ is enough to assign all the existing connections the maximum rate $C_{user\_max(i)}$. Otherwise, the base capacity is shared among all the existing and new VPN connection. Therefore, we can express this admission policy as follows:

$$C_{shared(i)} = min\left(C_{base(i)}, C_{user\_max(i)}.N_{shared(i)}\right)$$
$$C_{user(i)} = \frac{C_{shared(i)}}{N_{shared(i)}}$$

Example 1: For the following example assume that the total link bandwidth $C_T = 100$ Mbps, $C_{shared} = 0.3C_T = 30$ Mbps and there is only one ($N = 1$) shared user group .Also assume that ISP offers this group as $C_{user\_min(1)} = 1$ Mbps and $C_{user\_max(1)} = 2$ Mbps. Base capacity $C_{base(1)}$ allocated to this group is 20 Mbps.

$N_{shared(1)} = 1$, $C_{shared(1)} = 2$ Mbps, $C_{user(1)} = 2$ Mbps

.

$N_{shared(1)} = 10$, $C_{shared(1)} = 20$ Mbps, $C_{user(1)} = 2$ Mbps

$N_{shared(1)} = 11$, $C_{shared(1)} = 20$ Mbps, $C_{user(1)} = \frac{20}{11}$ Mbps

.

$N_{shared(1)} = 20$, $C_{shared(1)} = 20$ Mbps, $C_{user(1)} = \frac{20}{20}$ Mbps

Calls are accepted as long as the condition $\left(N_{shared(i)} \leq \frac{C_{base(i)}}{C_{user\_min(i)}}\right)$ of section III-A is met. When the number of calls exceed $\frac{C_{base(i)}}{C_{user\_min(i)}}$ a new arriving call is rejected. For example, if the 21st call in the example were accepted then $C_{user(1)}$ would have been $\frac{20}{21}$, and the minimum bandwidth could no longer be guaranteed. Therefore, the call is rejected.

C. Capacity Allocation with sharing among groups: Policy II

If the capacity allocated to a group is not fully used by VPN connections, then this capacity can be borrowed by connections of other shared service groups if needed. However, borrowed capacity must be relinquished when needed by the group from which capacity was borrowed. Although this borrowing and deallocation adds some complexity in edge provisioning, connections from various groups, however, have better chances of enjoying higher rates. In the following we present algorithms regarding VPN connection arrival, termination and possible expansion of existing connections as a result of the termination of a connection from a shared service group.

C.1 VPN Connection Arrival

Like the previous case, VPN connection arrival essentially involves checking the availability of resources that can be used by the new connection, and if available, allocating this capacity to an incoming call. Even if the base capacity of a certain group allows the new connection belonging to that group to assign maximum ISP offered rate (i.e. $\left(C_{base(i)} - C_{shared(i)}\right) \geq C_{user\_max(i)}$), because of the resource sharing among various groups it might happen that resources from that group has been borrowed by other group(s) not leaving the required resources (i.e. $C_{shared\_unused} < C_{user\_max(i)}$). In such a case resource must be relinquished from the appropriate groups(s). Any such deallocation from existing connections leads to rearrangement of capacity of those connections. It should be noted that capacity should be relinquished the way it was borrowed. There are numerous ways unused capacity can be borrowed by competing groups which we will see in sections III-C.3 and III-D. For the sake of simplicity, the group which has the maximum excess bandwidth, $C_{excess(i)} = C_{shared(i)} - C_{base(i)}$, should release first, and then the next, and so on.

/ if the group has enough base capacity to support a new connection with max. offered rate. */*

$if\left[\left(C_{base(i)} - C_{shared(i)}\right) \geq C_{user\_max(i)}\right]$

$\{$

/ if the shared unused capacity is also enough to support the new connection with max. offered rate. See Example 2 */*

$if\left(C_{shared\_unused} \geq C_{user\_max(i)}\right)$

$\{$

$C_{shared(i)} = C_{user\_max(i)} \cdot N_{shared(i)}$

$C_{user(i)} = C_{user\_max(i)}$

$\}$

/ if the shared unused capacity has been borrowed then capacity is relinquished from borrower(s). See Example 3 */*

else

$\{$

relinquish $C_{user\_max(i)}$ from group(s) which has max excess bw rearrange bandwidth of that group(s)

$C_{shared(i)} = C_{user\_max(i)} \cdot N_{shared(i)}$

$C_{user(i)} = C_{user\_max(i)}$

$\}$

$\}$

We have just mentioned that capacity can be borrowed from one group by the others. When does one group borrows resources? Naturally, when the base capacity is less than what is needed i.e $\left(C_{base(i)} - C_{shared(i)}\right) \leq 0$. How much can one group borrow? This depends on how much unused resources are available. If this is at least equal to the maximum offered rate $C_{user\_max(i)}$, then that amount is allocated, otherwise (i.e. $C_{shared\_unused} < C_{user\_max(i)}$) the whole unused resource goes to the group in question and is then divided among all the connections in that group

/ if the shared capacity is equal to or has exceeded the base capacity */*

$if\left[\left(C_{base(i)} - C_{shared(i)}\right) \leq 0\right]$

$\{$

/ but the unused capacity can still support the new connection with max rate. Capacity is then borrowed. See Example 4 */*

$if\left(C_{shared\_unused} \geq C_{user\_max(i)}\right)$

$\{$

$C_{shared(i)} = C_{shared(i)} + C_{user\_max(i)}$

$C_{user(i)} = \frac{C_{shared(i)}}{N_{shared(i)}} = C_{user\_max(i)}$

$\}$

*/*if the unused capacity is less than the max. rate. Capacity is then shared by existing and the new connection. See Example 5 */*

else

$\{$

$C_{shared(i)} = C_{shared(i)} + C_{shared\_unused}$

$C_{user(i)} = \frac{C_{shared(i)}}{N_{shared(i)}}$

$\}$

$\}$

We will now consider several numerical examples in this section to clarify the algorithms and analysis presented above. For all the following examples we assume that the total link bandwidth $C_T = 100$ Mbps, $C_{shared} = 0.3C_T = 30$ Mbps and there are only two shared users groups i.e. $i = 1, 2$. For group 1 $C_{user\_min(1)} = 0.5$ Mbps and $C_{user\_max(1)} = 1$ Mbps, and for group 2 $C_{user\_min(2)} = 1$ Mbps and $C_{user\_max(2)} = 2$ Mbps.

Example 2 : Prior to VPN connection request in group 1:
$N_{shared(1)} = 5, C_{shared(1)} = 5 \times 1 = 5$ Mbps
$N_{shared(2)} = 10, C_{shared(2)} = 10 \times 2 = 20$ Mbps

Here, for group 1, $C_{base(1)} - C_{shared(1)} = 10 - 5 = 5$ Mbps and $C_{user\_max(1)} = 1$ Mbps. Therefore, $C_{base(1)} - C_{shared(1)} > C_{user\_max(1)}$. Also, $C_{shared\_unused} = 30 - (5 + 20) = 5$ Mbps, which is greater than $C_{user\_max(1)}$. Hence, $C_{user(1)} = 1$ Mbps.

Example 3 : Prior to VPN connection request in group 1:
$N_{shared(1)} = 6, C_{shared(1)} = 6 \times 1 = 6$ Mbps
$N_{shared(2)} = 12, C_{shared(2)} = 12 \times 2 = 24$ Mbps

In this example, $C_{base(1)} - C_{shared(1)} = 10 - 6 = 4$ Mbps, which is greater than $C_{user\_max(1)} = 1$ Mbps. This means that group 1 hasn't used all it's base bandwidth and a new connection can have the maximum offered bandwidth 1 Mbps. However, C_{shared_unused} at the time of request arrival is $C_{shared} - \sum_{i=1}^{2} C_{shared(i)} = 30 - (6 + 24) = 0$ Mbps. This indicates that another group has has borrowed capacity from group 1. If that group had left at least $C_{user\_max(1)} = 1$ Mbps then the request could have been assigned the desired amount of resource. Therefore, the only option left is to relinquish 1 Mbps from the group that has borrowed it. Searching the table we find that the only other group 2 has taken that bandwidth. Therefore, we need to deduct 1 Mbps from group 2 and recompute the individual share of a VPN connection as $C_{user(2)} = \frac{C_{shared(2)} - C_{user\_max(1)}}{N_{shared(2)}} = \frac{24-1}{12} = 23/12$ Mbps. Obviously, $C_{user(1)} = 1$ Mbps and $C_{shared(1)} = 6 + 1 = 7$ Mbps.

Example 4 : Prior to VPN connection request in group 2:
$N_{shared(1)} = 5, C_{shared(1)} = 5 \times 1 = 5$ Mbps
$N_{shared(2)} = 10, C_{shared(2)} = 10 \times 2 = 20$ Mbps

This is a case where one group has used it's full allocated base capacity but can borrow resources from the other group which has left some spare capacity. Here, $C_{base(2)} - C_{shared(2)} = 20 - 20 = 0$ Mbps, but the total spared capacity $C_{shared\_unused} = 30 - (5 + 20) = 5$ Mbps, and this value is greater than $C_{user\_max(2)}$ (i.e 2 Mbps). Therefore, the new VPN connection request can be allocated the maximum offered value (i.e. 2 Mbps) by even exceeding the base capacity of group 2.

Example 5 : Prior to VPN connection request in group 2:
$N_{shared(1)} = 8, C_{shared(1)} = 8 \times 1 = 8$ Mbps
$N_{shared(2)} = 11, C_{shared(2)} = 11 \times 2 = 22$ Mbps

The example here depicts a scenario where one group which has already exceeded it's base capacity and has to accommodate a new connection request when there is no unused resource left by other group(s).Here, even before the new call arrival, Group 2 has borrowed $C_{shared(2)} - C_{base(2)} = 22 - 20 = 2$ Mbps and $C_{shared\_unused} = 30 - (8 + 22) = 0$ Mbps. So, the current capacity allocated to group 2 will have to be equally distributed among all the existing and the new arriving VPN connection. Therefore, $C_{user(2)} = \frac{C_{shared(2)}}{N_{shared(2)}} = \frac{22}{11+1} = \frac{22}{12}$ Mbps.

C.2 VPN Connection Termination

When a VPN connection terminates, resources might have to be released from the relevant group depending on the current rate every connection is enjoying in that group. If the rate is less than or equal to the maximum offered rate then no capacity is released from the groups current share and as a result all the connections in that group increases equally. This is because the same capacity is shared by a lower number of connections. If, however, the current rate of every connection is already equal to the maximum offered rate, then this termination would trigger a deduction of $C_{user\_max(i)}$ from the shared resource $C_{shared(i)}$. If all the connections were already enjoying $C_{user\_max(i)}$, no rate change occurs in any of the existing connections.The algorithm is stated as follows:

$$if\left(\frac{C_{shared(i)}}{N_{shared(i)}} \leq C_{user\_max(i)}\right) \text{ /*See Example 6 */}$$
$$\{$$
$$C_{shared(i)} = C_{shared(i)}$$
$$C_{user(i)} = \frac{C_{shared(i)}}{N_{shared(i)}}$$
$$C_{shared\_unused} = C_{shared\_unused}$$
$$\}$$
$$if\left(\frac{C_{shared(i)}}{N_{shared(i)}} = C_{user\_max(i)}\right) \text{ /* Example 7 */}$$
$$\{$$
$$C_{shared(i)} = C_{shared(i)} - C_{user\_max(i)}$$
$$C_{user(i)} = \frac{C_{shared(i)}}{N_{shared(i)}} = C_{user\_max(i)}$$
$$C_{shared\_unused} = C_{shared\_unused} + C_{user\_max(i)}$$
$$\}$$

To clarify the VPN connection termination process will now consider similar examples as presented in the previous section.

Example 6: Before VPN connection termination from group 1:
$N_{shared(1)} = 11$, $C_{shared(1)} = 10$ Mbps
$N_{shared(2)} = 10$, $C_{shared(2)} = 20$ Mbps

Here, $\frac{C_{shared(1)}}{N_{shared(1)}} < C_{user\_max(1)}$ since $\frac{10}{11} < 1$. This means that the capacity used by this group before the connection termination will remain unchanged even after the termination. So, the new value of $C_{shared(1)}$ is also 10 Mbps and each VPN connection will equally share this capacity which is $\frac{C_{shared(1)}}{N_{shared(1)}} = \frac{10}{10} = 1$ Mbps. Since no capacity is deducted from this group, the total unused shared capacity will also remain unchanged.

Example 7: Before VPN connection departure from group 1:
$N_{shared(1)} = 10$, $C_{shared(1)} = 10$ Mbps
$N_{shared(2)} = 10$, $C_{shared(2)} = 20$ Mbps

In this example, $\frac{C_{shared(1)}}{N_{shared(1)}} = C_{user\_max(1)}$ since $\frac{10}{10} = 1$. This states the fact that prior to this departure all active VPN connections were using the maximum possible offered bandwidth $C_{user\_max(1)} = 1$ Mbps and in total were having $C_{shared(1)} = 1 \times 10 = 10$ Mbps. Hence, the departure should trigger a deduction of $C_{user\_max(1)} = 1$ Mbps from the total capacity used by this group prior to the departure as the capacity even after the deduction will be good enough to satisfy $N_{shared(1)} = 10 - 1 = 9$ active connections offering highest possible rate of 1 Mbps. Therefore, $C_{shared(1)} = 10 - 1 = 9$ Mbps and each VPN connection will receive $\frac{C_{shared(1)}}{N_{shared(1)}} = \frac{9}{9} = 1$ Mbps. Since the termination process triggers deduction of $C_{user\_max(1)}$ from the capacity used by group 1, the unused shared capacity will increase by the same value. So, $C_{shared\_unused} = 0 + 1 = 1$ Mbps.

C.3 VPN Capacity Expansion

Unused shared capacity left by some groups can be distributed among others. Priority can be given to certain groups while allocating unused capacity. In the next section we will present various policies to allocate unused dedicated capacity and those might apply here as well. Here we consider only one case where preference is given to the needy groups where need is determined from the ratio $\frac{C_{user(i)}}{C_{user\_max(i)}}$. So, we order the groups according to this ratio where in reordered groups the first one has the lowest $\frac{C_{user(i)}}{C_{user\_max(i)}}$ and the last one has the highest $\frac{C_{user(i)}}{C_{user\_max(i)}}$. Once reordering has been done the expansion algorithm starts allocating unused bandwidth to the first group, then the next, and so on based on the availability of resources. This can be stated as :

$$if\left(\frac{C_{shared(i)}+C_{shared\_unused}}{N_{shared(i)}} > C_{user\_max(i)}\right)$$
$$\{$$
$$C_{shared(i)} = N_{shared(i)}.C_{user\_max(i)}$$
$$C_{user(i)} = \frac{C_{shared(i)}}{N_{shared(i)}}$$
$$C_{shared\_unused(i)} =$$
$$C_{shared\_unused} - [N_{shared(i)}.C_{user\_max(i)} - C_{shared(i)}]$$
$$\} \text{ /* See Example 8 */}$$
$$if\left(\frac{C_{shared(i)}+C_{shared\_unused}}{N_{shared(i)}} \leq C_{user\_max(i)}\right)$$
$$\{$$
$$C_{shared(i)} = C_{shared(i)} + C_{shared\_unused}$$
$$C_{user(i)} = \frac{C_{shared(i)}}{N_{shared(i)}}$$
$$C_{shared\_unused} = 0$$
$$\} \text{ /* See Example 9 */}$$

Example 8: Before VPN connection termination from group 2:
$N_{shared(1)} = 11$, $C_{shared(1)} = 10$ Mbps
$N_{shared(2)} = 10$, $C_{shared(2)} = 20$ Mbps

After the termination of a VPN connection from group 2, $C_{shared\_unused} = 2$ Mbps. If there is need of resources by other group(s), this capacity can be used partly or fully. We find that group 1 has need for this resource since $\frac{C_{user(1)}}{N_{user\_max(1)}} < 1$. Now it remains to be seen to what extent we could use this unused capacity. Here, $\frac{C_{shared(1)}+C_{shared\_unused}}{N_{shared(1)}} = \frac{10+2}{11} = \frac{12}{11}$ and is greater than $C_{user\_max(1)}$ which is 1 Mbps. Therefore, capacity for group 1 can be expanded to $N_{shared(1)}.C_{user\_max(1)} = 11 \times 1 = 11$ Mbps allocating each existing connection $C_{user\_max(1)} = 1$ Mbps. The remaining unused capacity will be reduced to $C_{shared\_unused} - [N_{shared(1)}.C_{user\_max(1)} - C_{shared(1)}] = 2 - (11 \times 1 - 10) = 1$ Mbps.

Example 9: Before VPN connection departure from group 2:
$N_{shared(1)} = 14$, $C_{shared(1)} = 10$ Mbps
$N_{shared(2)} = 10$, $C_{shared(2)} = 20$ Mbps

Unlike the previous example where group 1 only needed to use a portion of the unused resources, all the remaining capacity can be allocated to existing group 1 VPN connections in order to enhance the service. $C_{shared(1)}$ will be increased to 10+ 2=12 Mbps and each existing connection will receive $\frac{C_{shared(1)}}{N_{shared(1)}} = \frac{12}{14}$ Mbps.

D. Fair Allocation of Unused Dedicated Resources: Policy III

In the previous section we have discussed methods where one shared service group can borrow resources from another similar group. In this section, we will discuss the possibilities of sharing the unused dedicated resources among various shared service groups. If the shared service groups are allowed to borrow resources from unused dedicated resources, we then define a new term:

$$C^{+}_{shared} = C_{shared} + C_{ded\_unused}$$

The question here is how we can allocate the unused dedicated resources fairly among the competing groups. If all VPN tunnels want the maximum bandwidth as offered in ISP policy offer, then it is possible that at some point:

$$\sum_{i=1}^{N} N_{shared(i)}.C_{user\_max(i)} > C^{+}_{shared}$$

If $\left[\sum_{i=1}^{N} N_{shared(i)}.C_{user\_max(i)} - C^{+}_{shared}\right]$, the quantity that is needed to allocated the maximum possible offered rates to all connections even after allowing the unused dedicated resources to be used by shared service groups, is greater than 0, we need to define a fair set of user throughput values (i.e. $C_{user(i)}$) given the set of maximum offered loads $C_{user\_max(i)}$ and C^{+}_{shared}. In other words, we need to divide this extra capacity C_{ded_unused} among all the needy groups in a fair manner. However, fair sharing of extra resources is not a trivial issue and was addressed by others for different network situations [20], [11], [18], [19] . Some proposals [11] are in favour of sharing the bottleneck capacity equally among users independent of their requirements , and others [20], [18] advocate to penalize users causing overloads.

While we do share the resources among VPN connections in each group, equal sharing of unused dedicated capacity will not help much to some groups where connections are already enjoying rates close to $C_{user\_max(i)}$. At the same time it also doesn't alleviate the problem of other groups having rates above $C_{user\_min(i)}$ but much less than $C_{user\_max(i)}$. The fairness criterion of [20] also doesn't fit here as that would deprive the heavy user groups to gain share from unused dedicated resources even when they are enjoying rates much below $C_{user\_max(i)}$. Our case is further complicated by the fact that while penalizing heavy user groups we cannot reduce their current share, and this is what might happen in certain cases while trying to maximize the rates of lower user groups. In the following sections we will discuss various fair sharing methods at the edges.

D.1 Allocation of unused resources to lower user groups first

In this case, we first need to order the user groups based on their $C_{user\_max(i)}$ values. The objective is to satisfy the lower user groups first by trying to allocate maximum offered values while higher user groups have less chances to acquire resources left by dedicated service group. The rationale behind this is that more VPN users can be satisfied and allocating to higher user groups might bring little changes in many cases if sufficient extra resource is not available.

If the ordering leads to service groups $1, 2, 3, ..., K-1, K, K+1, ...N-1, N$, it is possible that if we expand K groups the VPN tunnels belonging to those group will enjoy the maximum offered bandwidth, $(K+1)$ th group receives rest of unused dedicated resource, and other tunnels remain unchanged. The total enhanced shared capacity can then be computed as follows:

$$\begin{aligned} C^{+}_{shared} &= \sum_{i=0}^{K} N_{shared(i)}.C_{user\_max(i)} \\ &+ C_{shared(k+1)} + \Bigg[C_{ded\_unused} \\ &- \sum_{i=1}^{K}[N_{shared(i)}.C_{user\_max(i)} - C_{shared(i)}]\Bigg] \\ &+ \sum_{i=K+2}^{N} C_{shared(i)} \end{aligned}$$

The above computation helps us to view how C^{+}_{shared} is shared by different groups. However, this general case is true when $K \geq 1, (N-K) \geq 2$. The other cases are:

$$C^{+}_{shared} = \begin{cases} C_{shared(1)} + C_{ded\_unused} & \text{if } K=0, (N-K)=1 \\ \left[C_{shared(1)} + C_{ded\_unused}\right] + \sum_{i=2}^{K} C_{shared(i)} & \text{if } K=0, (N-K)\geq 2 \\ \sum_{i=1}^{K} N_{shared(i)}.C_{user\_max(i)} + C_{shared(k+1)} + C_{ded\_unused} - \sum_{i=1}^{K}[N_{shared(i)}.C_{user\_max(i)} - C_{shared(i)}] & \text{if } K \geq 1, (N-K)=1 \end{cases}$$

In practice, when there is unused dedicated capacity the process starts by asking the first group if the unused capacity is enough to satisfy all the VPN connections. If so, each connection receives a maximum value $C_{user\_max(i)}$ and then queries the second group. Otherwise, the whole amount of capacity is allocated to the first group and divided among the competing connections. The process continues as long as unused capacity is a positive figure.

Example 10 : Assume a situation where we have 3 groups where VPN connections in each of them were having capacity below their respective $C_{user\_max(i)}$. Also, $C_{shared} = 30$ Mbps, and for group 1: $C_{base(1)} = 5$ Mbps, $C_{user\_max(1)} = 0.5$ Mbps, $C_{user\_min(1)} = 0.25$ Mbps, for group 2: $C_{base(2)} = 10$ Mbps, $C_{user\_max(2)} = 1$ Mbps, $C_{user\_min(2)} = 0.5$ Mbps, and for group 3: $C_{base(3)} = 15$ Mbps, $C_{user\_max(3)} = 2$ Mbps, $C_{user\_min(3)} = 1$ Mbps. Prior to the availability of $C_{ded\_unused} = 7$ Mbps we had :
$N_{shared(1)} = 15, C_{shared(1)} = 5$ Mbps $C_{user(1)} = 0.333$ Mbps
$N_{shared(2)} = 12, C_{shared(2)} = 10$ Mbps $C_{user(2)} = 0.833$ Mbps
$N_{shared(3)} = 15, C_{shared(3)} = 15$ Mbps $C_{user(3)} = 1.00$ Mbps

Here the groups are already ordered. Applying the algorithms we see that the first two groups can be allocated the maximum rates. Therefore, they are both expanded to $15 \times (0.5) = 7.5$ Mbps and $12 \times 1 = 12$ Mbps respectively. Rest of the unused capacity $C_{ded\_unused} - \sum_{i=1}^{2}[N_{shared(i)}.C_{user\_max(i)} - C_{shared(i)}] = 7 - (7.5 - 5 + 12 - 10) = 2.5$ Mbps goes to the third group.

D.2 Allocation of unused resources to highest needy groups first

This is much like the process as described above with the only difference that groups are ordered based on their needs. Apportionment mechanisms and algorithms remain the same. Here, need is determined from the ratio of $\frac{C_{user(i)}}{C_{user\_max(i)}}$. So, groups with lower ratios get preference over groups with higher ratios. Therefore, the process starts feeding the most needy group and continues as long as it has some unused capacity.

Example 11 : From example 10 of previous section: $\frac{C_{user(3)}}{C_{user\_max(3)}} = 0.5$, $\frac{C_{user(1)}}{C_{user\_max(1)}} = 0.67$, and $\frac{C_{user(2)}}{C_{user\_max(2)}} = 0.83$. Clearly, group 3 is the most needy group. If we have $C_{ded\_unused} = 5$ Mbps then that can serve the the most needy group 3 and enhance it's service. The new $C_{user(3)} = \frac{20}{15} = 1.33$ Mbps and $\frac{C_{user(3)}}{C_{user\_max(3)}} = 0.67$. In the previous examples, this group never had the chance to grab portion of the unused bandwidth, but the new policy here allows it to improve the service substantially.

D.3 Allocation of unused resources based on proportional needs

Although the above mechanism seems to be fair since it allocates based on the group's need, but in many cases there will be several needy groups with little differences in their needs, and in such cases the apportionment might not be always fair if unused dedicated resources are exhausted while trying to feed first few groups and other remain deprived to get a share. In this section, we therefore, present a way to allocate unused resources based on proportional need. Any group that is in need of resource, i.e, having ratio $\frac{C_{user(i)}}{C_{user\_max(i)}} < 1$ receives a portion of unused resource that is proportional to the group's need. Therefore, any group i, after receiving the extra resource based on this proportional need, is expanded to $C_{shared(i)} = \frac{C_{ded\_unused} \cdot C_{shared\_excess(i)}}{C_{shared\_excess}} + C_{shared(i)}$. Here, need for group i $C_{shared\_excess(i)}$, is actually excess quantity that is needed to offer all connections in that group the maximum value $C_{user\_max(i)}$. Therefore, $C_{shared\_excess(i)} = [C_{user\_max(i)} - C_{user(i)}]N_{shared(i)}$

Example 12: Once again, let us consider example 10 to illustrate the use of proportional need. No ordering is needed here as allocation of extra capacity is solely based on proportional need. Here for group 1: $\frac{C_{user(1)}}{C_{user\_max(1)}} = 0.67$, for group 2: $\frac{C_{user(2)}}{C_{user\_max(2)}} = 0.83$, and for group 3: $\frac{C_{user(3)}}{C_{user\_max(3)}} = 0.5$.Application of this allocation policy will expand the capacity of group 1 to: $C_{shared(1)} = \frac{7[(0.5)15-5]}{[(0.5)15-5]+[(1)12-10]+[(2)15-15]} + 5 = 5.897$ Mbps.As a result, connections are improved with new $C_{user(1)} = 0.393$ Mbps, $\frac{C_{user(1)}}{C_{user\_max(1)}} = 0.79$. Similarly, for group 2: $C_{shared(2)} = 10.71$ Mbps, $C_{user(2)} = 0.89$ Mbps, $\frac{C_{user(2)}}{C_{user\_max(2)}} = 0.89$ and for group 3: $C_{shared(3)} = 20.39$ Mbps, $C_{user(3)} = 1.36$ Mbps, $\frac{C_{user(3)}}{C_{user\_max(3)}} = 0.68$. This clearly shows that proportional sharing fairly enhances the rate of most needy group 3. This wouldn't have been the case had we applied other fairness methods.

IV. Implementation of Bandwidth Broker for Dynamic Configuration

As the underlying network may provide different classes of service to satisfy various VPN customers, by identifying the generic functionality provided by any resource and policy options, we present the BB with a standard WEB interface as shown in Figure 6(b). The Bandwidth Broker manages the outsourced VPNs for corporate customers that have Service Level Agreements (SLAs) with their ISPs and allows one such user to specify demand through a WWW interface to establish a VPN with certain QoS between two endpoints.

A. The Essential Components of Bandwidth Broker

The BB needs to keep track of existing connections and available resources and update relevant databases to reflect the most recent network state. The BB interacts with specialized configuration daemons (CD) when a certain user request arrives to setup a tunnel and the BB has to decide whether it can allocate enough resources to meet the demand of that tunnel. While the BB invokes a SLA database to check the validity of the user request, it essentially needs to maintain a connection database that contains a list of currently active VPNs and an edge resource database to keep track of records of quantitative resource available (base capacity) and current resource consumption of various router interfaces.

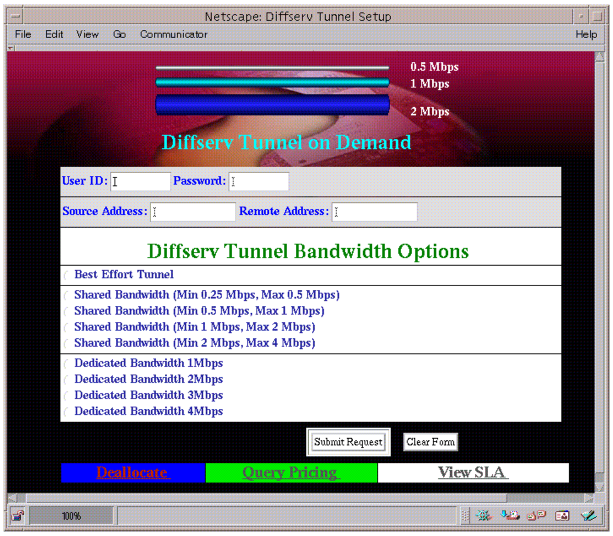

Fig. 6. BB WEB interface for Users

B. Examples of Dynamic Configuration

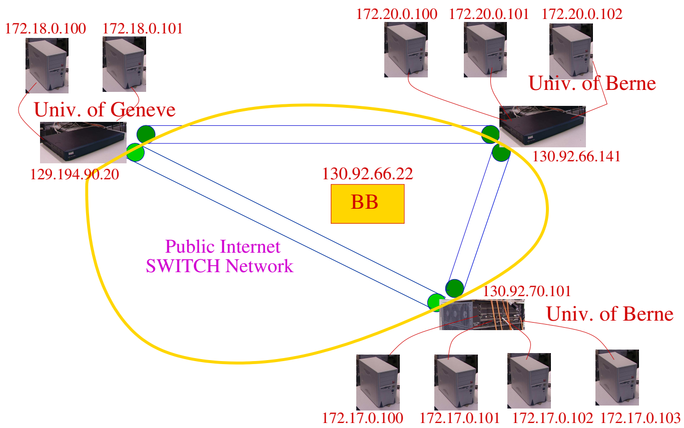

Fig. 7. Experimental Setup of VPN

A resource controller in the Bandwidth Broker checks resource and connection databases whenever there is any new connection arrival or departure that might trigger modification of rates of existing connections. For better understanding of how edge routers are dynamically configured to meet the user demand and conform SLA we will now demonstrate some examples of dynamic rate allocations of VPN connections in commercial Cisco routers. By considering similar examples as detailed in section III we will see how the simple algorithms are really applied to the edge devices. Let us consider an experimental setup (Figure 7) of Difserv-VPNs where we have three VPN and QoS capable edge routers each having private network behind them.

Configuration 1: User 'A' wants to establish a VPN connection for source 172.17.0.100 and destination 172.20.0.100 and chooses a menu (1-2 Mbps) from ISP provided website and submits his request. The resource group definition and edge resource database entries are as shown in Figure 8. Applying algorithm presented in section 3, the policing rate $C_{user(1)}$ that is configured in edge router 130.92.70.101 is $C_{user(1)} = C_{user\_max(1)} = 2$ Mbps. If user 'B' chooses the same menu he also gets $C_{user(1)} = 2$ Mbps since capacity in group 1 has the ability to support that. Assume that two more users 'C' and 'D' decide to have VPN connection with capacity varying between 0.5 and 1 Mbps. Group 2 can support both the connections with the maximum available rate of 1 Mbps. Therefore, $C_{user(2)} = C_{user\_max(2)} = 1$ Mbps is also configured in the router for these connections as we see in the following:

*/*policing individual VPN connection at the inbound with $C_{user(1)} = 2$ Mbps */*
for users 'A' and 'B' and $C_{user(2)} = 1$ Mbps for users 'C' and 'D'/*

```
rate-limit input access-group 140 2000000 2000000 8000000
  conform-action set-prec-transmit 1 exceed-action set-prec-transmit 2
rate-limit input access-group 141 2000000 2000000 8000000
  conform-action set-prec-transmit 1 exceed-action set-prec-transmit 2
rate-limit input access-group 142 1000000 2000000 8000000
  conform-action set-prec-transmit 1 exceed-action set-prec-transmit 2
rate-limit input access-group 143 1000000 2000000 8000000
  conform-action set-prec-transmit 1 exceed-action set-prec-transmit 2
```

*/*Classifying the requested VPN traffic/*

```
access-list 140 permit ip host 172.17.0.100 host 172.20.0.100
access-list 141 permit ip host 172.17.0.101 host 172.20.0.100
access-list 142 permit ip host 172.17.0.102 host 172.20.0.100
access-list 143 permit ip host 172.17.0.103 host 172.20.0.100
```

Here, we show only the ingress router policing and marking since diffserv is unidirectional. We assume that bit precedence 1 is used for EF traffic marking and traffic that exceed the specified rate are marked as best effort (bit precedence 2). Users not familiar with Cisco routers, should only notice the first of the traffic rate parameters (for example `2000000` in '`2000000 2000000 8000000`') in `rate-limit` policing and marking commands. This is the rate that we refer to as $C_{user(i)}$ for any group i. The other two are burst parameters.

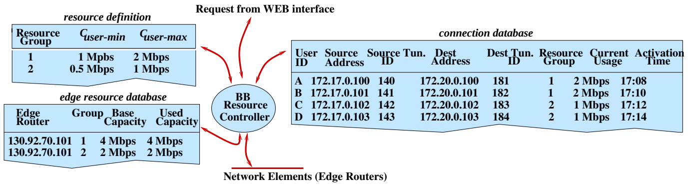

Fig. 8. Partial entries of Connection and Resource Databases.A scenario when all connections receive the maximum offered value

Configuration 2: Now if users 'A' and 'B' also want to establish connections from the same sources to 172.18.0.100 and 172.18.0.101 respectively and choose a menu (0.5 - 1 Mbps) i.e. group 2, we see that capacity is exhausted in group 2, and therefore, these two new connections and other two existing connections share the base capacity of 2 Mbps and each connection is configured with $C_{user(2)} = C_{user\_min(2)} = 0.5$ Mbps. This is shown in Figure 9 and the new configuration script that is used at this point is as follows:

```
rate-limit input access-group 140 2000000 2000000 8000000
 conform-action set-prec-transmit 1 exceed-action set-prec-transmit 2
rate-limit input access-group 141 2000000 2000000 8000000
 conform-action set-prec-transmit 1 exceed-action set-prec-transmit 2
rate-limit input access-group 142 500000 2000000 8000000
 conform-action set-prec-transmit 1 exceed-action set-prec-transmit 2
rate-limit input access-group 143 500000 2000000 8000000
 conform-action set-prec-transmit 1 exceed-action set-prec-transmit 2
rate-limit input access-group 144 500000 2000000 8000000
 conform-action set-prec-transmit 1 exceed-action set-prec-transmit 2
rate-limit input access-group 145 500000 2000000 8000000
 conform-action set-prec-transmit 1 exceed-action set-prec-transmit 2
access-list 144 permit ip host 172.17.0.100 host 172.18.0.100
access-list 145 permit ip host 172.17.0.101 host 172.18.0.101
```

V. Summary and Conclusion

In this paper, we have proposed a novel range based SLA that allows customers to specify their requirements as a range of quantitative service for VPN connections since they are unable or unwilling to predict load between the VPN endpoints. To support such services we have proposed and developed a prototype Bandwidth Broker (BB) that can logically partition the capacity at the edges to various service classes (or groups) of VPNs and manage them efficiently to allow resource sharing among the groups in a dynamic and fair manner. Various algorithms with examples and analysis have been presented to provision resource dynamically at the edges to support QoS for VPN connections.

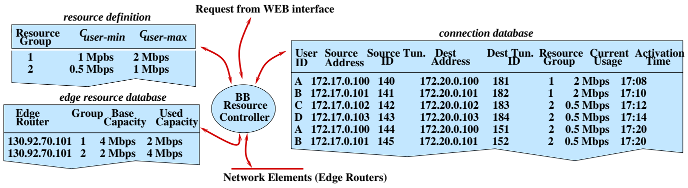

Fig. 9. A scenario when rate of existing connections are reduced to accommodate new connections

One obvious advantage of our system is the pricing gain. The price that customers have to pay is higher than one pays for the lower bound capacity but lower than what is normally needed to be paid for upper bound capacity. During low load it is possible that users might enjoy the upper bound rate without paying anything extra. Such pricing might be attractive to users and ISPs can take advantage of that to attract more customers. With all these advantages we believe that our model can be quite attractive to the ISPs willing to deploy it in a real world scenario.

VI. Acknowledgement

The work described in this paper is part of the work done in the project Charging and Accounting Technologies for the Internet (CATI) [4] funded by the Swiss National Science Foundation (Project no. 5003-054559/1 and 5003-054560/1), the SNF R' Equip project no. 2160-053299.98/1 and the foundation Förderung der wissenschaftlichen Forschung an der Universität Bern

References

[1] Gerald R. Ash. Routing Guidelines for Efficient Routing Methods. Internet Draft `draft-ash-itu-sg2-routing-guidelines-00.txt`, October 1999. work in progress.
[2] Y. Bernet, J. Binder, M. Carlson, B. E. Carpenter, S. Keshav, E. Davies, B. Ohlman, D. Verma, Z. Wang, and W. Weiss. A Framework for Differentiated Services. Internet Draft `draft-ietf-diffserv-framework-02.txt`, February 1999. work in progress.
[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weis. An Architecture for Differentiated Services, December 1998. RFC 2475.
[4] CATI. Charging and Accounting Technologies for the Internet. http://www.tik.ee.ethz.ch/~cati/.
[5] S. Chen and K. Nahrstedt. An Overview of Quality of Service Routing for Next-Generation High-Speed Networks: Problems and Solutions. *IEEE Network Magazine*, November/December 1998.
[6] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick. A framework for qos-based routing in the internet, August 1998. RFC 2386.
[7] N.G. Duffield, Pawan Goyal, Albert Greenberg, Partho Mishra, K.K. Ramakrishnan, , and Jacobus E. Van der Merwe. A Flexible Model for Resource Management in Virtual Private Networks. *SIGCOMM'99 Conference*, August 1999.
[8] F. L. Faucheur, L. Wu, B. Davie, S. Davari, P. Vaananen, R. Krishnan, and P. Cheval. MPLS Support of Differentiated Services. Internet Draft `draft-ietf-mpls-diff-ext-02.txt`, October 1999. work in progress.
[9] B. Gleeson, A. Lin, J. Heinanen, G. Armitage, and A. Malis. A Framework for IP Based Virtual Private Networks. Internet Draft `draft-gleeson-vpn-framework-03.txt`, 1999. work in progress.
[10] V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding phb, June 1999. RFC 2598.
[11] J.M. Jaffe. Bottleneck Flow Control. *IEEE Transactions on Communications*, 29(7), 1981.
[12] I. Khalil and T. Braun. Implementation of a Bandwidth Broker for Dynamic End-to-End Resource Reservation in Outsourced Virtual Private Networks. *The 25th Annual IEEE Conference on Local Computer Networks (LCN)*, November 9-10 2000.
[13] Ibrahim Khalil, Torsten Braun, and M. Günter. Implementation of a Service Broker for Management of QoS enabled VPNs. In *IEEE Workshop on IP-oriented Operations & Management (IPOM'2000)*, September 2000.
[14] Karthik Muthukrishnan and Andrew Malis. Core MPLS IP VPN Architecture. Internet Draft `raft-muthukrishnan-mpls-corevpn-arch-00.txt`, 2000. work in progress.
[15] K. Nichols, Van Jacobson, and L. Zhang. A Two-bit Differentiated Services Architecture for the Internet, July 1999. RFC 2638.
[16] QBONE. The Internet2 QBone Bandwidth Broker, 2000. http://www.internet2.edu/qos/qbone/QBBAC.shtml.
[17] Benjamin Teitelbaum and et al. Internet2 QBone: Building a Testbed for Differentiated Services. *IEEE Network*, September/October 1999.
[18] F. Wong and J.R.B. deMarca. Fairness in Window Flow Controlled Computer Networks. *IEEE Transactions on Communications*, 37(5), 1989.
[19] J.W. Wong, J.P. Sauve, and J.A. Field. A study of fairness in packet switching networks. *IEEE Transactions on Communications*, 30(2), 1982.
[20] Moshe Zukermann and Sammy Chan. Fairness in ATM networks. *Computer Networks and ISDN Systems*, 26, 1993.