Journal of Network and Systems Management, Vol. 10, No. 1, March 2002 (© 2002)

Edge Provisioning and Fairness in VPN-DiffServ Networks[1]

Ibrahim Khalil[2,3] and Torsten Braun[2]

Customers of Virtual Private Networks (VPNs) over Differentiated Services (DiffServ) infrastructure are most likely to demand not only security but also guaranteed Quality-of-Service (QoS) in pursuance of their desire to have leased-line-like services. However, expectedly they will be unable or unwilling to predict the load between VPN endpoints. This paper proposes that customers specify their requirements as a range of quantitative services in the Service Level Agreements (SLAs). To support such services Internet Service Providers (ISPs) would need an automated provisioning system that can logically partition the capacity at the edges to various classes (or groups) of VPN connections and manage them efficiently to allow resource sharing among the groups in a dynamic and fair manner. While with edge provisioning a 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 analyses 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.

KEY WORDS: Virtual Private Network (VPN); Differentiated Services (DiffServ); Quality-of-Service (QoS); Bandwidth Broker (BB); Service Level Agreement (SLA); Connection Admission; Resource Provisioning; Fairness; Dynamic Configuration.

[1]The work done here is part of the project Charging and Accounting Technologies for the Internet (CATI) [24] 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 Berne.
[2]Institute for Computer Science and Applied Mathematics (IAM), University of Berne, Neubrückstrasse 10, CH-3012 Bern, Switzerland. E-mail: {*ibrahim; braun*}*@iam.unibe.ch*
[3]To whom correspondence should be addressed.

1064-7570/02/0300-0011/0

1. INTRODUCTION

Since private networks built on using dedicated lines offer guaranteed bandwidth and latency, a growing demand urges similar guarantees being provided in IP-based Virtual Private Networks (VPNs) [1–3]. While the Internet has not been designed to deliver the performance guarantees, with the advent of DiffServ [4], the IP backbones can now provide various QoS levels [5]. Recently proposed Expedited Forwarding (EF) [6] Per Hop Behavior (PHB) is the recommended method of building such Virtual Leased Line (VLL)-type point-to-point connections for VPNs.

To provide such services we [7–9], along with others [10, 11], have implemented Bandwidth Brokers [12] that allow users to specify a single quantitative value (e.g., 1 Mbps or 2 Mbps) and based on this specification, the edge routers establish VPN connections dynamically. However, it is apprehended that users will be unable or unwilling to predict the load between VPN endpoints [13]. Also, from the provider's point of view, guaranteeing exact quantitative service might be difficult at the beginning of VPN-DiffServ deployment [5]. We, therefore, propose that users specify their requirements as a range of quantitative services. For example, a user who wants to establish a VPN between stub networks A and B (Fig. 1), not sure whether 0.5 Mbps or 0.6 Mbps or 1 Mbps is needed, and only knows the lower and upper bounds of the requirements approximately, can specify a range 0.5–1 Mbps as the user's requirement from the ISP (Internet Service Provider). An ISP can offer multiple such options via a website (Fig. 6) to help customers select any suitable option to activate services dynamically.

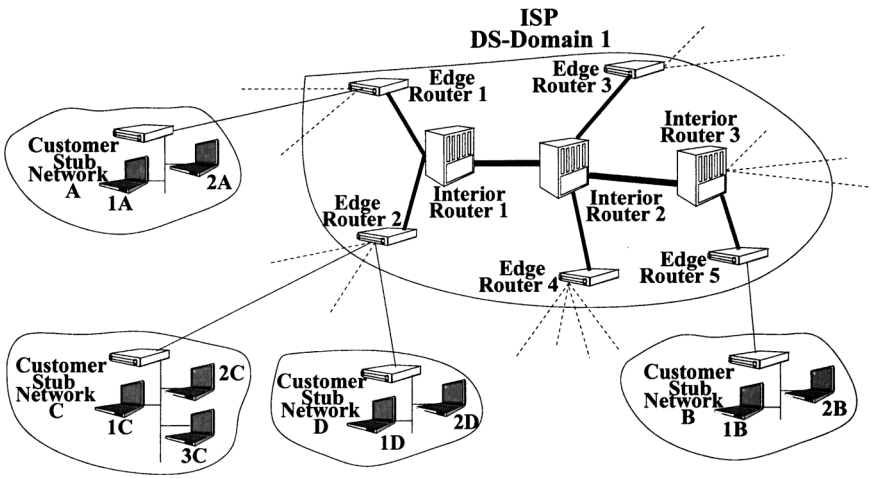

Fig. 1. VPN DiffServ deployment scenario.

This approach has several advantages. Users do not need to specify the exact capacity, but it gives them 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 when the range 0.5–1 Mbps is chosen) without paying anything extra. This kind of pricing might be attractive to users, and ISPs can take advantage of this to attract more customers without breaking the commitment.

This, however, poses a significant challenge to the ISPs, as they would need to deploy automated provisioning systems that can logically partition the capacity at the edges to various classes or groups of VPN connections and manage them efficiently to allow resource sharing among the groups in a dynamic and fair manner. Here, each group is identified from what it offers. For example, one group could represent the range 0.5–1 Mbps, another 1–2 Mbps. 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 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 BB.

We have restricted this paper only to edge provisioning because most of the complexities lie at the boundaries of the network and is the main driving force for overall provisioning. Section 2 describes the model for provisioning, and in Section 3 various algorithms with examples and analyses have been presented to provision and allocate resources dynamically at the edges. Fairness issues while allocating resources to connections of various VPN groups have also been addressed in this Section. A prototype BB performing the required provisioning and connection admission is described in Section 4. Section 5 concludes the paper with a summary of our contributions.

2. 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 includes modification of the existing resources to be shared dynamically among various VPN classes (i.e., groups). Both quantitative, as 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 the traffic volume that will traverse each network node. While an ISP naturally knows from the SLA the amount of quantitative VPN traffic that will enter the transit network through a specific edge node, this volume cannot be

estimated with exact accuracy at various interior nodes being traversed by VPN connections, if we do not know the path of such connections [14]. However, if the routing topology is known, this figure can almost be accurately estimated. If the default path does not meet the requirements of an incoming connection, alternate and various QoS routings [15, 16] can also be used to find a suitable path and enforced by MPLS techniques [17].

2.1. The Role of Bandwidth Broker for Automated Provisioning

Based on the basic needs of provisioning a VPN-DiffServ network to support quantitative services, 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 (Fig. 2). As we seek to provide a system where VPN services are available on demand, we find that the BBs [11, 12] are the right choice, because they are not only capable of performing dynamic end-to-end admission control to set up a leased line like VPN by maintaining the topology as well as policies and the states of all nodes in the network, but are also capable of managing and provisioning network resources of a separately administered DiffServ domain and cooperating with other similar domains.

2.2. A Novel Approach: Bandwidth Specified as an Interval

To overcome the difficulties faced by users in specifying the exact amount of quantitative bandwidth required while outsourcing the VPN service to ISPs,

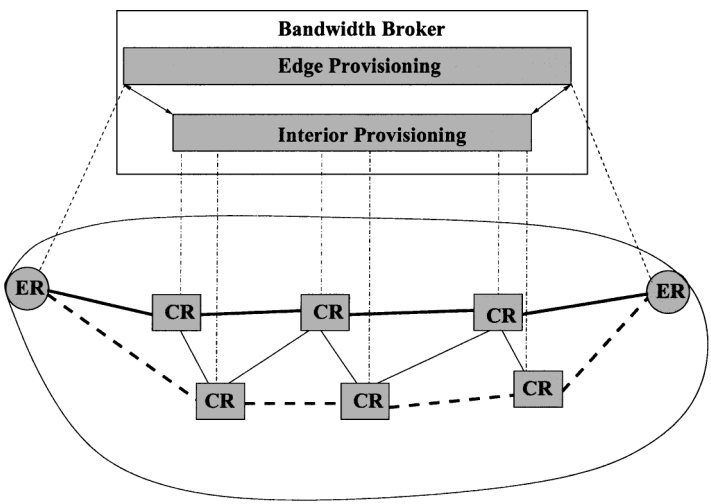

Fig. 2. Layered Provisioning view of VPN-DiffServ Networks.

our model supports a flexible way to express SLAs where a range of quantitative amounts, rather than a single value, can be specified. Although it has several advantages, this also makes the edge and the interior provisioning difficult. This complexity can be explained with a simple example. Referring to Fig. 1, assume that the edge router *R2* has been provisioned to provide 20 Mbps quantitative resources to establish VPN connections elsewhere in the network with the ISP has providing 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 does not cause either reductions or force any renegotiations 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 (Fig. 6) allowing users to select the range 1–2 Mbps, where 1 and 2 are the minimum and maximum offered guaranteed bandwidth, maintaining the state and admission control can be difficult. 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 the existing connections would decrease. For example, when there are 20 connections this rate would be $\frac{20}{20} = 1$ Mbps. At this 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 the range 1 Mbps–2 Mbps. In reality, we might have several such groups to support users requiring varying bandwidth. In such cases, renegotiation for possible expansion of the existing connections, admission control, and maintenance of network states will not be simple. Figure 3 illustrates the idea of range-based SLA. Bandwidth is specified as an interval of $C_{user\_min(i)}$ and $C_{user\_min(i)}$ for any group i. The Actual rate of a VPN connection $C_{user(i)}$ varies between this range but never gets below $C_{user\_min(i)}$. $C_{user(i)}$ is the

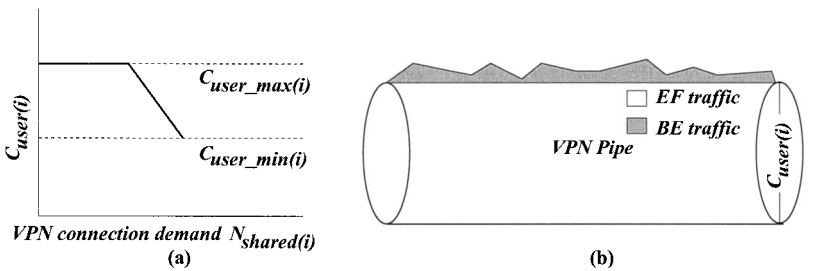

Fig. 3. The range-based SLA approach.

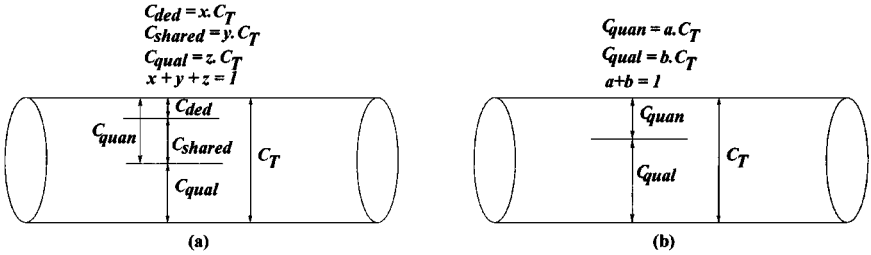

Fig. 4. Top level bandwidth apportionment: (A) logical partitioning at the edge; (B) logical partitioning at an interior.

rate that is configured in the edge router as the policing rate. Traffic submitted at a rate higher than this is marked as best effort traffic or dropped depending on the policy.

2.3. The Model and Notations

In our model, we address this novel SLA approach 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 1 Mbps 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 Fig. 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 VPN connections describing their requirement as a range.
- C_{quan} is the capacity provisioned for quantitative traffic and is equal to $(C_{ded} + C_{shared})$.
- C_{qual} is the remaining capacity for qualitative traffic.

While at the edge C_{quan} is rate controlled by policing or shaping, at the interior this C_{quan} indicates the amount of capacity 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. The unused capacity naturally goes to the qualitative portion and enhances the best effort and other qualitative services. This is true at both the edge

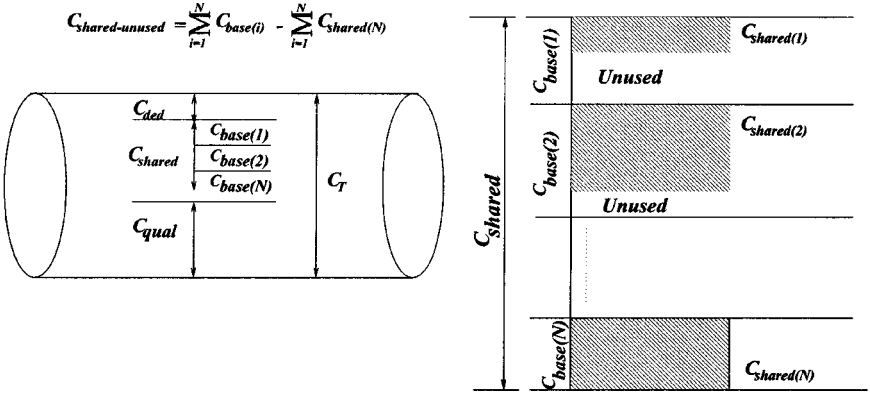

Fig. 5. Microscopic view of bandwidth apportionment at edge.

and in the interiors. C_{shared}, as shown in Fig. 4, can be logically divided to multiple groups where each group supports a different range (Fig. 5). As there might be multiple of such groups, for any group i we define the following notations:

- $C_{base(i)}$ is 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 $C_{shared(i)}/N_{shared(i)}$.
- C_{shared_unused} is the total unused bandwidth from all shared service groups.

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. A sharing policy might allow a group to share its resources not only among its own connections, but also share with other groups' VPN connections in case of some unused capacity left. 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.

3. EDGE PROVISIONING POLICIES: ANALYSIS AND ALGORITHMS

Based on the model described in Section 2, 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 from multiple classes of VPNs. 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, the group whose resources have been exhausted while supporting numerous connections does not borrow resources from others even when they 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 the dedicated service group can be borrowed by tunnels of the shared service groups. Also, these groups can share resources among themselves.

In this section, we will start with VPN connection acceptance algorithms at network ingress point where all admission complexities lie. These complexities are introduced because of the need to partition and share resources to support our model and policies presented above. Further analyses with examples of algorithms for Policy I, II and III clarify them.

3.1. VPN Connection 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 $(N_{shared(i)} \leq C_{base(i)}/C_{user\_min(i)})$, 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 hold is decided by the connection states in that group and sharing policies that we are going to discuss in the following sections.

3.2. 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 circumstances resources assigned to one

group can be borrowed by others, even if that capacity remains 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, the qualitative services mentioned earlier are also enhanced.

If a VPN connection is accepted, the system checks whether 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 at 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. Also assume that ISP offers a 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.

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

$$\vdots$$

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

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

$$\vdots$$

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

Connections are accepted as long as the condition ($N_{shared(i)} \leq C_{base(i)}/C_{user\_min(i)}$) of Section 3.1 is met. When the number of connections exceed $C_{base(i)}/C_{user\_min(i)}$ a new arriving connection is rejected. For example, if the 21st connection in the example is accepted then $C_{user(1)}$ would be $\frac{20}{21}$, and the minimum bandwidth could no longer be guaranteed. Therefore, the connection request is rejected.

3.3. Capacity Allocation with Sharing Among Groups: Policy II

If the capacity allocated to a group is not fully used by VPN connections, this capacity can be borrowed by connections of the other shared service groups

if needed. However, the 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 sections, we present algorithms regarding VPN connection arrival, termination, and possible expansion of the existing connections as a result of the termination of a connection from a shared service group.

3.3.1. VPN Connection Arrival

Similar to 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 connection. Even if the base capacity of a certain group allows the new connection belonging to that group to assign the maximum ISP offered rate (i.e. $(C_{base(i)} - C_{shared(i)}) \geq C_{user\_max(i)}$) because of the resource sharing among various groups, it might happen that the resources from that group would be borrowed by other group(s) not leaving the required resources (i.e. $C_{shared\_unused} < C_{user\_max(i)}$). In such a case resources must be relinquished from the appropriate groups(s). Any such deallocation from the existing connections leads to rearrangement of capacity of those connections. This capacity should be relinquished the way it was borrowed. The unused capacity can be borrowed numerous ways by competing groups which we will see in sections 3.3.3 and 3.4. For the sake of simplicity, the group having 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) . 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 borrow resources? Naturally, when the base capacity is less than what is needed, i.e., $(C_{base(i)} - C_{shared(i)}) \leq 0$. 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 divided among all the connections in that group.

```
/* if the shared capacity is equal to or has exceeded the base capacity */
if [ (C_base(i) - C_shared(i)) ≤ 0 ]
 {
/* but the unused capacity can still support the new connection
with max rate. Capacity is then borrowed. See Example 4 */
  if (C_shared_unused ≥ C_user_max(i))
   {
   C_shared(i) = C_shared(i) + C_user_max(i)
   C_user(i) = 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 here. 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 $C_{base(1)} = 10$ Mbps, $C_{user\_min(1)} = 0.5$ Mbps and $C_{user\_max(1)} = 1$ Mbps, and for group 2 $C_{base(2)} = 20$ Mbps, $C_{user\_min(2)} = 1$ Mbps and $C_{user\_max(2)} = 2$ Mbps.

Example 2. Prior to VPN connection request in group 1:

$$N_{shared(1)} = 5, C_{shared(1)} = 5 \times 1 = 5 \text{ Mbps}$$

$$N_{shared(2)} = 10, C_{shared(2)} = 10 \times 2 = 20 \text{ Mbps}$$

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:

$$N_{shared(1)} = 6, C_{shared(1)} = 6 \times 1 = 6 \text{ Mbps}$$

$$N_{shared(2)} = 12, C_{shared(2)} = 12 \times 2 = 24 \text{ Mbps}$$

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 has not used all its 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

$$C_{user(2)} = \frac{C_{shared(2)} - C_{user\_max(1)}}{N_{shared(2)}} = \frac{24-1}{12} = 23/12 \text{ Mbps}.$$

Obviously, $C_{user(1)} = 1$ Mbps and $C_{shared(1)} = 6 + 1 = 7$ Mbps.

Example 4. Prior to VPN connection request in group 2:

$$N_{shared(1)} = 5, C_{shared(1)} = 5 \times 1 = 5 \text{ Mbps}$$

$$N_{shared(2)} = 10, C_{shared(2)} = 10 \times 2 = 20 \text{ Mbps}$$

This is a case where one group has used its full allocated base capacity but could 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; 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:

$$N_{shared(1)} = 8, C_{shared(1)} = 8 \times 1 = 8 \text{ Mbps}$$

$$N_{shared(2)} = 11, C_{shared(2)} = 11 \times 2 = 22 \text{ Mbps}$$

The example here depicts a scenario where one group that has already exceeded its 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 connection 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 connections. Therefore,

$$C_{user(2)} = \frac{C_{shared(2)}}{N_{shared(2)}} = \frac{22}{11+1} = \frac{22}{12} \text{ Mbps}.$$

3.3.2. VPN Connection Termination

When a VPN connection terminates, the resources might have to be released from the relevant group depending on the current rate every connection enjoying in that group. If the rate is less than or equal to the maximum offered rate, no capacity is released from the group's current share. As a result, all the connections in that group will increase 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, 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 would occur in any of the existing connections. The algorithm stated follows:

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

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:

$$N_{shared(1)} = 11, C_{shared(1)} = 10\,\text{Mbps}$$

$$N_{shared(2)} = 10, C_{shared(2)} = 20\,\text{Mbps}$$

Here, $C_{shared(1)}/N_{shared(1)} < C_{user\_max(1)}$ 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 $C_{shared(1)}/N_{shared(1)} = \frac{10}{10} = 1$ 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:

$$N_{shared(1)} = 10, C_{shared(1)} = 10\,\text{Mbps}$$

$$N_{shared(2)} = 10, C_{shared(2)} = 20\,\text{Mbps}$$

In this example, $C_{shared(1)}/N_{shared(1)} = C_{user\_max(1)}$ since $\frac{10}{11} = 1$. Thus, 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 the highest possible rate of 1 Mbps. Therefore, $C_{shared(1)} = 10 - 1 = 9$ Mbps, and each VPN connection will receive $C_{shared(1)}/N_{shared(1)} = \frac{9}{9} = 1$ 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.

3.3.3. VPN Capacity Expansion

Unused shared capacity left by some groups can be distributed among others with priority given to certain groups while allocating the unused capacity. In the next section we will present various policies to allocate the 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 $C_{user(i)}/C_{user\_max(i)}$. So, we order the groups according to this ratio so that the first one has the lowest and the last one has the highest value of $C_{user(i)}/C_{user\_max(i)}$. Once reordered, 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:

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

Example 8. Before VPN connection termination from group 2:

$$N_{shared(1)} = 11, C_{shared(1)} = 10\,\text{Mbps}$$
$$N_{shared(2)} = 10, C_{shared(2)} = 20\,\text{Mbps}$$

After the termination of a VPN connection from group 2, $C_{shared\_unused} = 2\,\text{Mbps}$. If there is a 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 $C_{user(1)}/N_{user\_max(1)} < 1$. 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 $N_{shared(1)}.C_{user\_max(1)} = 11 \times 1 = 11\,\text{Mbps}$ allocating each existing connection $C_{user\_max(1)} = 1\,\text{Mbps}$. The remaining unused capacity will be reduced to $C_{shared\_unused} - [N_{shared(1)}.C_{user\_max(1)} - C_{shared(1)}] = 2 - (11 \times 1 - 10) = 1\,\text{Mbps}$.

Example 9. Before VPN connection departure from group 2:

$$N_{shared(1)} = 14, C_{shared(1)} = 10\,\text{Mbps}$$
$$N_{shared(2)} = 10, C_{shared(2)} = 20\,\text{Mbps}$$

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 the existing group 1 VPN connections in order to enhance the service. $C_{shared(1)}$ will be increased to $10 + 2 = 12\,\text{Mbps}$ with each existing connection will receiving $C_{shared(1)}/N_{shared(1)} = \frac{12}{14}\,\text{Mbps}$.

3.4. Fair Allocation of Unused Dedicated Resources: Policy III

In the previous section we discussed methods where one shared service group could 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 the 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, 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 needed to allocate the maximum possible offered rates to all connections even after allowing the unused dedicated resources to be used by the 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 the maximum offered rates $C_{user\_max(i)}$ and C^{+}_{shared}. In other words, we need to fairly divide this extra capacity C_{ded_unused} among all the needy groups. However, fair sharing of extra resources is not a trivial issue and was addressed by others for different network situations [18–21]. Some proposals [19] are in favor of sharing the bottleneck capacity equally among users independent of their requirements, and others [18, 20] 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 does not 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 [18] also does not fit here as that would deprive the heavy user groups to gain share from the 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 the heavy user groups we cannot reduce their current share. 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.

3.4.1. 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 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 the dedicated service group. The rationale behind this is that more VPN users can be satisfied and allocating to the higher user groups might bring little changes in many cases if sufficient extra resources are not available.

If the ordering leads to service groups $1, 2, 3, \ldots, K-1, K, K+1, \ldots N-1, N$, 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 the rest of the unused dedicated resource, and other tunnels remain unchanged.

The total enhanced shared capacity can then be computed as follows:

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

This 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:

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

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 the unused capacity is a positive figure.

Example 10. Assume a situation where we have 3 groups with VPN connections in each of them 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:

$$N_{shared(1)} = 15, C_{shared(1)} = 5\,\text{Mbps}\; C_{user(1)} = 0.333\,\text{Mbps}$$

$$N_{shared(2)} = 12, C_{shared(2)} = 10\,\text{Mbps}\; C_{user(2)} = 0.833\,\text{Mbps}$$

$$N_{shared(3)} = 15, C_{shared(3)} = 15\,\text{Mbps}\; C_{user(3)} = 1.00\,\text{Mbps}$$

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. The 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.

3.4.2. Allocation of Unused Resources to Highest Needy Groups First

This is much like the process as previously described with the only difference that the groups are ordered based on their needs. Apportionment mechanisms and algorithms remain the same. Here, the need is determined from the ratio of $C_{user(i)}/C_{user\_max(i)}$. So, the groups with lower ratios get preference over the 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:

$$\frac{C_{user(3)}}{C_{user\_max(3)}} = 0.5, \frac{C_{user(1)}}{C_{user\_max(1)}} = 0.67, \text{ and } \frac{C_{user(2)}}{C_{user\_max(2)}} = 0.83.$$

Clearly, group 3 is the most needy group. If we have $C_{ded\_unused} = 5$ Mbps, then it can serve the most needy group 3 and enhance its service. The new $C_{user(3)} = \frac{20}{15} = 1.33$ Mbps and $C_{user(3)}/C_{user\_max(3)} = 0.67$. In the previous example, 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.

3.4.3. Allocation of Unused Resources Based on Proportional Needs

Although this mechanism seems to be fair since it allocates based on the group's need, in many cases there will be several needy groups with little differences in their needs. In such cases, the apportionment might not be always fair if the unused dedicated resources are exhausted while trying to feed the first few groups and the others remain deprived to get a share. In this section, we present a way to allocate unused resources based on proportional need. Any group that is in need of resource, i.e., having the ratio $C_{user(i)}/C_{user\_max(i)} < 1$ receives a portion of the unused resource proportional to the group's need. Therefore, any group i, after receiving the extra resource based on this proportional need, is expanded to

$$C_{shared(i)} = \frac{C_{ded\_unused}.C_{shared\_excess(i)}}{C_{shared\_excess}} + C_{shared(i)}.$$

Here, the need for group i $C_{shared\_excess(i)}$, is actually the excess quantity 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 the allocation of extra capacity is solely based on the proportional need. Here, for group 1: $C_{user(1)}/C_{user\_max(1)} = 0.67$; for group 2: $C_{user(2)}/C_{user\_max(2)} = 0.83$; and for group 3: $C_{user(3)}/C_{user\_max(3)} = 0.5$.

Application of this allocation policy will expand the capacity of group 1 to:

$$C_{shared(1)} = \frac{7[(0.5)15 - 5]}{[(0.5)15 - 5] + [(1)12 - 10] + [(2)15 - 15]} + 5 = 5.897\,\text{Mbps}.$$

As a result, connections improve with new $C_{user(1)} = 0.393$ Mbps, $C_{user(1)}/C_{user\_max(1)} = 0.79$. Similarly, for group 2: $C_{shared(2)} = 10.71$ Mbps, $C_{user(2)} = 0.89$ Mbps, $C_{user(2)}/C_{user\_max(2)} = 0.89$; and for group 3: $C_{shared(3)} = 20.39$ Mbps, $C_{user(3)} = 1.36$ Mbps, $C_{user(3)}/C_{user\_max(3)} = 0.68$. This clearly shows that proportional sharing fairly enhances the rate of the most needy group 3. This would not have been the case had we applied other fairness methods.

4. IMPLEMENTATION OF BB FOR DYNAMIC CONFIGURATION

A prototype BB has been implemented which optimally configures network resources and supports call admission based on user preferences and SLA. 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 Fig. 6. The BB manages the outsourced VPNs for corporate customers that have SLAs

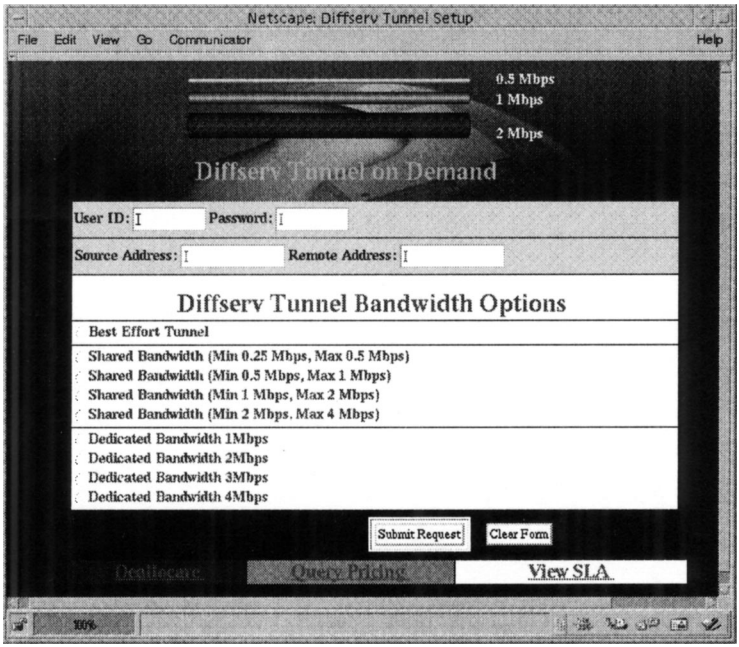

Fig. 6. BB WEB interface for users.

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. Here, we will not present the implementation details but briefly discuss the relevant parts that are mostly responsible for dynamic resource allocation at the edge devices. Readers are encouraged to refer to [7–9] for further details of the implementation, operation and examples of dynamic VPN establishment. We will also present some examples of the dynamic rate allocations of VPN connections in commercial Cisco IOS routers [22] to illustrate the methods presented in earlier sections.

4.1. The Essential BB Components

While admission process might merely involve checking resource availability at the edge (assuming enough resource is available in the interior), it might also trigger modification of the existing connections. To do this, the BB keeps track of the 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 has to decide whether it can allocate enough resources to meet the demand of that tunnel. The CDs are intelligent provisioning agents that are able to translate user requests and policy data to device specific configurations. These agents also remotely configure the network devices with translated configurations without any human intervention. While the BB invokes an *SLA database* to check the validity of the user request, it essentially needs to maintain a *connection database* containing 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.

The basic operation (Fig. 7) of our system is as follows: based on request parameters (step 1) provided by the user, the BB first contacts a SLA database (step 2,3) to check the validity of the user and its request parameters. It then checks the CD's availability (steps 4,5) and the connection (step 6,7) database whether a similar requested connection already exists or not. If this is not the case, the BB looks at its resource database (8,9) to identify the possibility of tunnel establishment. A positive answer would then lead to a tunnel establishment by the CD (rest of the steps).

4.2. Examples of Dynamic Configuration

A resource controller in the BB checks resource and connection databases whenever there is any new connection arrival or departure that might trigger the modification of rates of the existing connections. For a better understanding of how the edge routers are dynamically configured to meet the user demand and conform the SLA, we will now demonstrate some examples of dynamic rate allocations

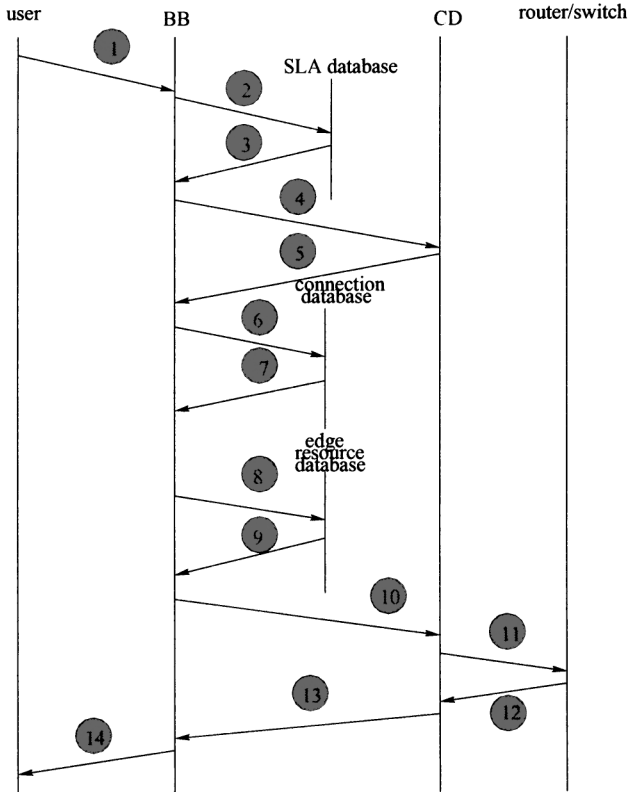

Fig. 7. Successful VPN connection establishment.

of VPN connections in commercial Cisco IOS routers. By considering similar examples, as detailed in Section 3, we will see how the simple algorithms are really applied to the edge devices. Let us consider an experimental setup (Fig. 8) of Diffserv-VPNs where we have three VPN and QoS capable edge routers each having a 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 option (1–2 Mbps) from ISP provided website and submits a request. Figure 9 shows the resource group definition and edge resource database entries. Applying algorithm presented in Section 3, the policing rate $C_{user(1)}$ 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 option, the same rate $C_{user(1)} = 2$ Mbps is allocated since capacity in group 1 has the ability to support that. Assume that two more users 'C' and 'D' decide to have VPN connections (for sources and destinations specified in the connection database of

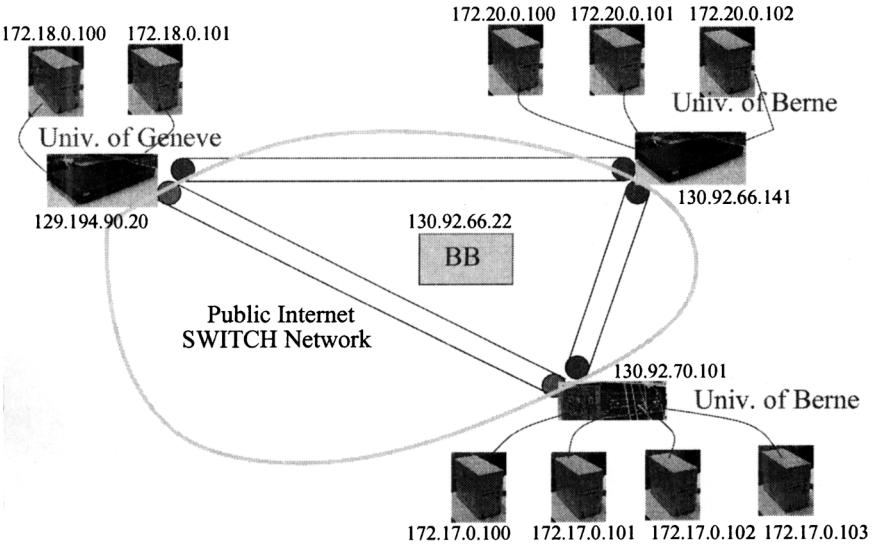

Fig. 8. Experimental setup of VPN.

Fig. 8) 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
```

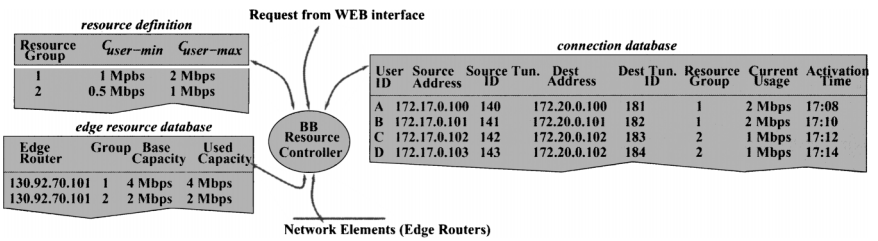

Fig. 9. Partial entries of connection and resource databases. A scenario when all connections receive the maximum offered value.

```
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 172.20.0.101
access-list 142 permit ip host 172.17.0.102 host 172.20.0.102
access-list 143 permit ip host 172.17.0.103 host 172.20.0.102
```

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 with traffic that exceed the specified rate marked as the best effort (bit precedence 2). The users not familiar with Cisco IOS 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 we refer to as $C_{user(i)}$ for any group i. The other two are burst parameters.

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 an option (0.5–1 Mbps) i.e., group 2, we see that group 2 is exhausted of its capacity. Therefore, these two new connections along with the 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 Fig. 10 and the new configuration commands loaded to the router at this point is as follows:

```
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
```

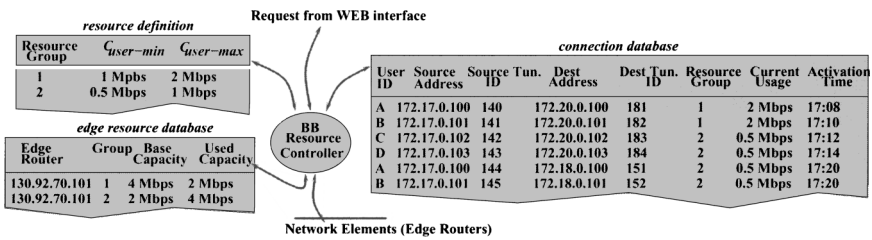

Fig. 10. A scenario when rates of the existing connections are reduced to accommodate new connections.

```
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
```

5. CONCLUSIONS

In this paper, we have proposed a novel range-based SLA that allows customers to specify their requirements as a range of quantitative services for VPN connections since they are unable or unwilling to predict the load between the VPN endpoints. To support such services, we have proposed and developed a prototype 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 analyses have been presented to provision resource dynamically at the edges to support QoS for VPN connections.

We have restricted this paper to edge provisioning only considering the fact that most of the complexities lie at the boundaries of the network, and that it is the main driving force for overall provisioning. However, the ISPs must provision the interior nodes in the network to meet the assurance offered at the boundaries of the network. Core provisioning that work in unison with the proposed edge resource allocation policies here has been addressed in [23].

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 these advantages we believe that our model can be quite attractive to the ISPs willing to deploy it in a real world scenario.

REFERENCES

1. R. Callon, M. Suzuki, B. Gleeson, A. Malis, K. Muthukrishnan, E. Rosen, C. Sargor, and J. J. Yu, A framework for provider provisioned virtual private networks, Internet draft *draft-ietf-ppvpn-framework-01.txt*, work in progress, July 2001.
2. J. D. Clercq, O. Paridaens, M. Iyer, and A. Krywaniuk, A framework for provider provisioned CE-based virtual private networks using IPsec, Internet draft *draft-ietf-ppvpn-ce-based-00.txt*, work in progress, July 2001.

3. K. Muthukrishnan, C. Kathirvelu, A. Malis, T. Walsh, F. Ammann, J. Sumimoto, and J. M. Xiao, Core MPLS IP VPN architecture, Internet draft `draft-ietf-ppvpn-rfc2917bis-00.txt`, work in progress, July 2001.
4. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weis, An architecture for differentiated services, RFC 2475, December 1998.
5. 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.
6. V. Jacobson, K. Nichols, and K. Poduri, An expedited forwarding, RFC 2598, June 1999.
7. 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.
8. T. Braun, M. Günter, and I. Khalil, Management of quality-of-service enabled VPNs. *IEEE Communications Magazine*, Vol. 39, No. 5, pp. 90–98, May 2001.
9. I. Khalil, T. Braun, and M. Günter, Implementation of a service broker for management of QoS enabled VPNs, *IEEE Workshop on IP-Oriented Operations and Management (IPOM'2000)*, September 2000.
10. QBONE, The Internet2 QBone bandwidth broker, 2000. *http://www.internet2.edu/qos/qbone/QBBAC.shtml.*
11. B. Teitelbaum, S. Hares, L. Dunn, R. Neilson, R. Narayan, and F. Reichmeyer, Internet2 QBone: Building a testbed for differentiated services, *IEEE Network*, Vol. 13, No. 5, pp. 8–16, 1999.
12. K. Nichols, Van Jacobson, and L. Zhang, A two-bit differentiated services architecture for the Internet, RFC 2638, July 1999.
13. N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan, and J. E. van der Merwe, A flexible model for resource management in virtual private networks. *SIGCOMM'99 Conference*, August 1999.
14. Gerald R. Ash, Routing guidelines for efficient routing methods, Internet draft `draft-ash-itu-sg2-routing-guidelines-00.txt`, October 1999.
15. E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick, A framework for QoS-based routing in the Internet, RFC 2386, August 1998.
16. S. Chen and K. Nahrstedt, An overview of quality-of-service routing for next-generation high-speed networks: Problems and solutions, *IEEE Network Magazine*, Vol. 12, No. 6, pp. 64–79, 1998.
17. F. L. Faucheur, L. Wu, B. Davie, S. Davari, P. Vaananen, R. Krishnan, P. Cheval, and J. Heinanen, MPLS support of differentiated services, Internet draft `draft-ietf-mpls-diff-ext-09.txt`, work in progress, April 2001.
18. Moshe Zukermann and Sammy Chan, Fairness in ATM networks, *Computer Networks and ISDN Systems*, Vol. 26, pp. 109–117, 1993.
19. J. M. Jaffe, Bottleneck flow control, *IEEE Transactions on Communications*, Vol. 29, No. 7, pp. 954–962, 1981.
20. F. Wong and J. R. B. deMarca, Fairness in window flow controlled computer networks, *IEEE Transactions on Communications*, Vol. 37, No. 5, pp. 954–962, 1989.
21. J. W. Wong, J. P. Sauve, and J. A. Field, A study of fairness in packet switching networks, *IEEE Transactions on Communications*, Vol. 30, No. 2, pp. 346–353, 1982.
22. Cisco web site, last modified October 2001. *http://www.cisco.com.*
23. I. Khalil and T. Braun, A range-based sla and edge driven virtual core provisioning in diffservvpns, *The 26th Annual IEEE Conference on Local Computer Networks (LCN)*, November 15–16, 2001.
24. CATI, Charging and accounting technologies for the Internet. *http://www.tik.ee.ethz.ch/∼cati/.*

Ibrahim Khalil has an M.S. degree in computer engineering from University Putra Malaysia, and a B.S. in electrical engineering from BIT Rajshahi, Bangladesh. He was a graduate research assistant with the ATM research group in electronics and computer engineering, University Putra Malaysia, between 1993 and 1996. Prior to joining the research group of Prof. Braun in 1998 as a Junior Researcher, and Ph.D. Student, he briefly worked as a research assistant with the LTS and C3i groups of EPFL, Switzerland. His research interests are dynamic network provisioning, QoS issues of VPN, MPLS, and network pricing. He is currently working with Ponte Communications, Mountain View, California, USA.

Torsten Braun received his Ph.D. from the University of Karlsruhe, Germany in 1993. From 1994 to 1995, he was a guest scientist at INRIA Sophia-Antipolis, France. From 1995 to 1997, he worked at the IBM European Networking Center in Heidelberg, Germany, and became the project leader and senior consultant. Since 1998, he has been a full professor of computer science at the Institute of Computer Science and Applied Mathematics at the University of Bern, Switzerland, where he leads the Computer Networks and Distributed Systems research group. He was elected a member of the Swiss Academic Research Network (SWITCH) committee in 2000.