

# Optimal Customer Provisioning in Network-Based Mobile VPNs

Katherine Guo Sarit Mukherjee Sanjoy Paul Sampath Rangarajan  
Center for Networking Research  
Bell Laboratories  
101 Crawfords Corner Road, Holmdel, NJ 07733  
{kguo, sarit, sanjoy, sampath}@bell-labs.com

## Abstract

A virtual private network (VPN) is an overlay network that uses the public network to carry data traffic between corporate sites and users, maintaining privacy through the use of tunnelling protocols and security procedures. In the network-based model, VPN-aware network elements are placed within the network to set up concatenated tunnels between the user/site and enterprise resources to offer intranet VPN and remote access VPN.

This paper identifies the important differences between a traditional VPN and the mobile VPN and proposes a hierarchical network architecture to efficiently realize network-based mobile VPNs. We address the problem of optimally provisioning VPN-aware devices, called IP Service Gateways (IPSGs), in the hierarchical network architecture for mobile VPNs, while taking into account of (1) the cost of links over which VPN tunnels are established, (2) the cost of provisioning a VPN customer on an IPSG, and (3) redundancy in IPSG provisioning for fault tolerance. We develop generic yet powerful problem formulations for different scenarios described above while considering practical requirements of the network elements and business requirements of the VPN service provider. The formulation becomes a set of integer programming problems. We solve several instances of the problem for a few practical cases and discuss their applications in the overall network design.

## 1. Introduction

A Virtual Private Network (VPN) [15] is a cost effective and secure way of extending enterprise network resources over a shared public data network. Most popular uses of VPNs are to interconnect multiple geographically dispersed sites of an enterprise (known as intranet/extranet VPN) and to provide remote users access to the enterprise resources (known as remote access VPN).

The basic method of setting up a VPN from a user or a site to secure enterprise resources is to set up a secure data connection between them over the underlying insecure shared network. A VPN can be categorized into two models. In the *end-to-end* model, the user/site connects to the enter-

prise resources over a secure tunnel using the underlying network as a simple data conduit. In the *network-based* model, the network service provider (NSP) implements VPN-aware routers [6, 13] within the network. This router usually sets up two secure tunnels, one from the user/site to the router itself and the other from the router to the enterprise. The data flows end-to-end through the two tunnels concatenated together at the VPN router. Moreover, a VPN router can enable other value-added services from the tunnel concatenation point. Examples include better QoS guarantees for VPN tunnels, service differentiation among users, offloading of Internet traffic from the enterprise intranet, etc. All these benefits come at the cost of trusting the NSP to maintain the security associations with the end points at the VPN router.

Although the end-to-end model makes VPN service access facilities independent of the service provider, it cannot support scalable growth and requires large investments by the enterprise. The network-based model, on the other hand, can perform traffic aggregation at the tunnel concatenation points for better scalability and network resource usage and, therefore, can cost-effectively offer VPN services. Thus, a network-based VPN is preferred as VPN usage grows [5]. The rest of the paper deals with network-based remote-access VPNs only.

In a network-based VPN, the VPN routers are capable of handling VPNs with different types of tunnels and security mechanisms. Examples include L2TP [17], MPLS [16] and IPSec [10]. Several existing service switches, [13] [6], support such services. Such a service switch is called the *IP Services Gateway (IPSG)*. An IPSG can be provisioned to serve a number of enterprise VPN *customers* each with a number of *end users*. Throughout the paper, we use *customers* to refer to enterprises and *users* to refer to end users of enterprises. The provisioning process creates virtual instances of routing mechanisms for each of the customers facilitated in the IPSG. Each routing instance requires a considerable amount of computing resources. Since all the instances share common resources of the IPSG, the number of VPN customers that can be provisioned on an IPSG is *limited*. There is a similar restriction on the number of tunnels an IPSG can support. Moreover, due to physical resource constraints, configuring an IPSG with increased number of provisions reduces the number of tunnels that can be handled, and vice versa. Because of the complexity of the process, IPSG provisioning per

customer is usually carried out statically and is not changed frequently.

In the network-based VPN model, a remote access VPN is created by tunnelling the remote user's connection to an IPSG provisioned for the enterprise that the user belongs to. The IPSG then tunnels the connection to the appropriate CPE using tunnel concatenation, as described above.

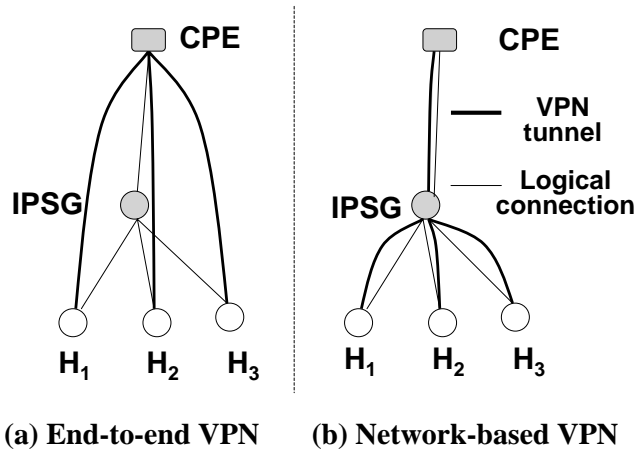


Figure 1. Two types of VPNs.

Figure 1 shows the difference between an end-to-end VPN and a network-based VPN with reference to remote access VPNs. In this example, users  $H_1$ ,  $H_2$  and  $H_3$  belong to the same enterprise VPN customer. The CPE resides inside the customer's intranet. In the end-to-end VPN model, three VPN tunnels are established, one from each user to the CPE as shown in Figure 1(a). The CPE must maintain three different tunnels one with each of the users. In the network-based VPN model in Figure 1(b), four VPN tunnels are established, one from each user to the IPSG, and one from the IPSG to the CPE. Here the CPE maintains only one tunnel with the IPSG. The IPSG maintains tunnels with the users. Therefore, the load on the CPE is reduced considerably. Because all the traffic from the three users is going to the same CPE, IPSG serves as a tunnel concatenation point and a traffic aggregation point. This allows the IPSG to enable value-added services for the customer.

A network-based VPN solution is a well-suited candidate to implement remote access VPNs to support stationary users. But with the growth in the number of mobile users, an important issue to explore is whether the existing network-based VPN architecture is suited for mobile users. We raise this issue in this paper, and propose a solution to optimally use network resources to provide network-based remote access VPN services to mobile users.

The rest of the paper is organized as follows. Section 2 discusses the main problems associated with mobile network-based VPN model, and proposes a novel architecture. Section 3 formalizes the problem using integer programming and performance results are discussed in Section 4. Finally Sec-

tion 5 presents our conclusion and future work.

## 2. Mobile VPN

At present, remote access VPNs are mostly limited to end users connecting to the enterprise from remote locations using wireline access. With the emergence of high-speed wireless data services in 2.5G and 3G wireless technologies [1, 3], VPN usage from mobile nodes (that is, mobile VPN services) will grow exponentially [8]. We believe that network-based VPNs are the best solution to satisfy the expected growth of mobile VPN services, and we examine how a scalable network-based mobile VPN can be built using existing VPN enabling routers like IPSGs.

In order to enable mobile data services, an NSP installs wireless access devices at the edge of its network. Radio to packet network gateways, that are referred to as *Mobile Access Points (MAPs)* in this paper, connect the access devices to the data network. A PDSN in the CDMA 2000 architecture [3] and a GGSN/SGSN in the UMTS architecture [1] are candidate MAPs. To set up a data session, a mobile end user, whom we refer to as a *mobile node (MN)*, must first connect to a MAP, which then routes the session towards the destination CPE through an appropriately provisioned IPSG.

A mobile data session originating from an MN to a MAP, then routed through an IPSG to the enterprise CPE is the basis of a network-based mobile VPN service.

### 2.1. Mobile VPN Design

The NSP has several choices in designing and provisioning a network-based mobile VPN. One easy approach is to collocate the IPSG and the MAP within a single device. We will also refer to such a device as an IPSG when there is no ambiguity. In this scenario, an IPSG performs radio to packet network gateway functions to terminate MN's connection and conducts other IPSG specific functions. We note here that such gateway devices exist today [14] and its all wireline counterpart, which does nothing more than terminating different wireline interfaces at the IPSG, also exists today [6, 13].

In such a scenario, the MN is not free to choose an IPSG; its data sessions are anchored to the IPSG that provides the MAP functionality for the MN's current roaming region and if this IPSG is not provisioned to provide VPN services for this MN, it cannot be offered these services until it moves into the region of a co-located IPSG/MAP that can provide such service. This problem has two obvious solutions. In the first solution named *uniform-provision*, the NSP provisions every IPSG in the network for all customers. This is required because an MN belonging to any customer can roam into the region served by any IPSG and request service. In the second solution referred to as *tunnel-switching*, an IPSG is provisioned for only a subset of customers. But when a MN moves into the region of a co-located IPSG/MAP that is not provisioned for this mobile, it is required that this IPSG tunnel the traffic to an IPSG that is provisioned for this MN. This requires each IPSG to be aware of the provisions made by other

IPSGs, detect the identity of the MN and tunnel the session to an appropriate IPSG.

In the uniform-provision solution, suppose the NSP has  $N$  IPSGs and each can support at most  $M$  different provisions. The total number of different provisions the NSP can provide is therefore  $M \times N$ . In practice, every VPN customer must be provisioned on every IPSG, and this limits the total number of supported VPN customers to merely  $M$ . Clearly this solution does not scale with the number of subscribed VPN customers. The tunnel-switching solution supports mobility through tunnel switching the MN’s data sessions from the IPSG in the MN’s roaming area to the appropriately provisioned IPSG. To handle more VPN customers, the IPSGs must support more tunnels, which in turn will reduce the number of provisions that can be made per IPSG. Thus, this solution does not scale with the number of subscribed VPN customers either. Moreover, tunnel switching among IPSGs leads to undesirable redirection of connections (commonly known as “dog-legging”) within the NSP’s network resulting in an inefficient usage of network links.

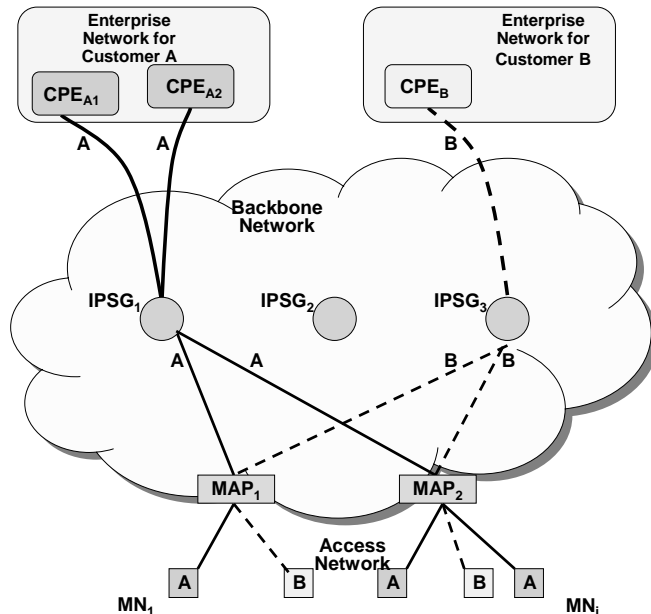
Contrasting to these two existing design, our solution *hierarchical mobile VPN architecture* is much more scalable and is described in Section 2.2.

## 2.2. Hierarchical Mobile VPN Architecture

Our solution for scalable network-based remote-access VPN is to separate the MAP functionality and IP services functionality and hierarchically locate MAPs and IPSGs as shown in Figure 2. A MAP serves a region and all MNs within that region connect to the MAP to initiate data sessions. Each IPSG is statically provisioned for only a subset of the VPN customers. The subset of customers per IPSG is chosen so that at least one IPSG is provisioned for each customer. An IPSG maintains the virtual routing instance and the security association corresponding to each provisioned VPN customer. Each MAP maintains a simple and fairly static list of customer-to-IPSG mappings. When an MN requests a VPN connection to its CPE, the MAP identifies the customer the MN belongs to, and routes and/or tunnel switches the connection to the appropriate IPSG provisioned for the customer. The MN/user identification method is discussed in Section 3.1. In the example shown in Figure 2,  $IPSG_1$  is provisioned for VPN customer A and  $IPSG_3$  is provisioned for customer B. Mobile traffic destined to customer A and B is directed by MAPs to  $IPSG_1$  and  $IPSG_2$  respectively. Each IPSG only needs to support a subset of the two VPN customers.

The key novelty of this approach is to separate mobility from services, where a MAP deals with mobility of users while an IPSG offers VPN services. This is a natural division of functions because IPSGs are designed to support services for stationary locations, while MAPs are designed to handle mobility by providing dynamic switching and routing.

Such a hierarchical solution provisions a subset of IPSGs per customer (instead of provisioning each customer on every IPSG) thereby offering much improved scalability. The need for tunnel switching between IPSGs and the associated dog-legging problems are eliminated as well. In addition, the



**Figure 2. Hierarchical network-based mobile VPN architecture: separate MAP and IPSG.**

MAPs are able to separate intranet VPN traffic from internet traffic, direct VPN traffic to the appropriate IPSGs, and direct internet traffic to appropriate internet proxies in the NSP network. This value-added internet traffic offloading service effectively saves bandwidth for the NSP and its customers over the existing architecture where MAPs and IPSGs are collocated.

In order to design the NSP’s network, we need to map each VPN customer to a subset of IPSGs. One extreme and easy solution is to map/provision all customers on one IPSG, and use the next IPSG only when the current one is full. This method, of course, does not utilize the resources fully. It creates hot spots and degrades the overall performance of the network. For better utilization, a subset of IPSGs must be chosen in an optimal fashion for each customer, so that all the IPSGs are equally provisioned/utilized and there is room for inclusion of future customers. This paper addresses the issue of determining the best set of IPSGs to provision for each customer while taking into account the following three factors: (1) the cost of links over which VPN tunnels are established, (2) the cost of provisioning a VPN customer on an IPSG, and (3) redundancy in IPSG provisioning for fault tolerance. In the following sections, we formulate the decision process as a set of integer programming problems. We solve several instances of the problem for a few practical cases and discuss related design considerations.

## 3. Selection of IPSGs

This section investigates and proposes a solution to the MN-IPSG-CPE connectivity problem of selecting a subset of

IPSGs to provision for each VPN customer. From the NSP’s point of view, provisioning each VPN customer in its network produces certain revenue while incurring certain cost for the service. In this paper we consider the sum of the link cost over which the VPN tunnels are established and the IPSG provision cost as the total cost for each customer. The profit is defined as the difference between the revenue and the total cost. Our objective is to optimally provision each IPSG so that (1) any MN belonging to the customer can connect to its CPE from any region within the NSP, and (2) the NSP can maximize the number of subscribed customers, under the constraints that (a) the number of VPN customers provisioned per IPSG is limited and (b) the overall profit is maximized.

### 3.1. Assumptions

We assume that the IPSGs and MAPs are already deployed by the NSP and that the NSP’s network provides connectivity between these MAPs and IPSGs. We consider the general case where a VPN customer may have one or more CPEs serving as VPN endpoints for its users. First, we resolve which set of IPSGs should be provisioned for the customer. These IPSGs establish pre-configured VPN tunnels to the corresponding CPEs. We assume that MNs are identified using the popular methods of Network Access Identifier (NAI) [4] and/or Access Point Name (APN) [2]. A MAP extracts the NAI/APN of the MN during connection setup time with the MN. Directly from the NAI/APN, it can then identify the destination CPE, if there is only one CPE. If there is more than one CPE, the MAP can determine the MN’s preferred CPE from an Authentication, Authorization and Accounting (AAA) Server [7].

### 3.2. The Network Model

In our model we consider multiple VPN customers in a batch and determine the best set of IPSGs to provision for the batch that would maximize the profit. For each customer, we consider the CPEs in the customer’s intranet, and all of the IPSGs and MAPs in the NSP network. We model the network of IPSGs, MAPs, and the customer’s CPEs as an undirected graph  $G = (V, E)$  where  $V$  is the set of nodes and  $E$  is the set of links. We assume that there are  $I$  MAPs,  $J$  IPSGs in the network and  $K$  CPEs for a given customer, denoted by  $p_i$ ,  $q_j$  and  $c_k$ , respectively.

Graph nodes in  $V$  correspond to CPEs, IPSGs and MAPs only. Graph links in  $E$  fall in the following two categories: (1) a link between a MAP and an IPSG corresponds to the chosen path between the corresponding MAP and the IPSG, and (2) a link between an IPSG and a CPE corresponds to the chosen path between them. The chosen paths are computed by the routing algorithm based on the routing objective. It could be the shortest path based on hop counts, or the lowest-cost path based on the cost assigned to network links, both of which can be computed by OSPF [12]. It could also be a traffic-engineered path such as an ATM VC or an MPLS Label Switched Path. In the hierarchical architecture, traf-

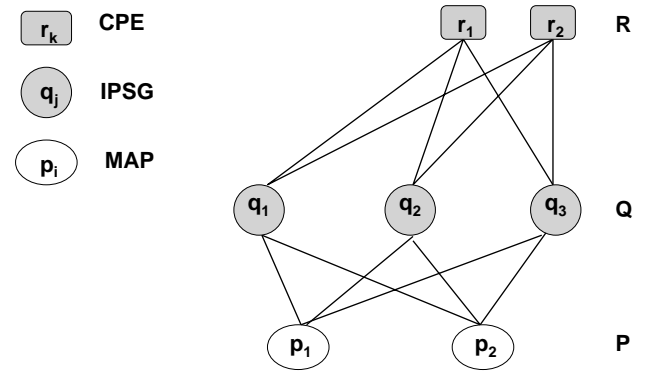


Figure 3. Example graph model for one customer.

fic flows from MNs to CPEs through the MAPs and IPSGs. Therefore, only links between them are considered.

Figure 3 shows an example graph for a customer. There are two MAPs,  $p_1$  and  $p_2$ , three IPSGs,  $q_1$ ,  $q_2$ , and  $q_3$  in the network, and two CPEs  $r_1$  and  $r_2$  for the customer.

The establishment of VPN tunnels over a physical network link incurs a certain cost associated with the link. In practice, depending on the requirement of the VPN customer, the link cost may be the number of hops in the underlying physical network or a fraction of the bandwidth capacity of the physical links, etc. In this paper, as we are interested only in an optimal *connectivity* between the MAPs and the CPEs, we do not consider the bandwidth capacity of a physical link in computing link costs.

Traffic for all the users of a customer received at a MAP, that is destined to any of the CPEs of the customer, and redirected through a specific IPSG, are sent over the same link from that MAP to that IPSG. Hence, the cost of a link from a MAP to an IPSG is considered only once per customer. Similarly, traffic for a specific customer that is destined to a specific CPE received at an IPSG from all the MAPs is aggregated and sent over the same link between the IPSG and the CPE. Hence, the cost of a link from an IPSG to a CPE is considered only once per customer.

Based on the network model described above we now formulate the IPSG selection problem. We first consider the case where provisioning does not take IPSG fault tolerance into account. Afterwards we bring in IPSG fault tolerance into the formulation. In the latter case, multiple IPSGs are provisioned per customer so that should one IPSG fail, traffic can be directed through other IPSGs already provisioned for the customer.

### 3.3. Selection of IPSGs without Fault Tolerance

For every session from a user of a customer to a CPE, a VPN tunnel is dynamically established from the user through the corresponding MAP to an IPSG. However, the traffic from the IPSG to the CPE will be aggregated over one statically

pre-configured tunnel. We refer to them as *dynamic tunnel* and *static tunnel*, respectively, because of the way they are usually created in practice. We first formulate the problem for a single customer, and then generalize it for multiple customers.

**3.3.1. Single Customer Formulation** The goal of the optimization problem is to maximize the total profit by optimally provisioning the customer(s). Of course, profit is the difference between revenue and cost. We assume that the revenue for a customer is a fixed value if it can be provisioned. The cost has several components as described below.

Let  $P$  be the set of all MAPs,  $Q$  be the set of all IPSGs, and  $R$  be the set of all CPEs for the customer as shown in Figure 3. We denote by the binary variable  $x_{ijk} \in \{0, 1\}$  whether a dynamic tunnel between node  $i \in P$  and node  $j \in Q$  is used for the traffic from MAP  $i$  to CPE  $k \in R$ . We use binary variable  $z_{jk} \in \{0, 1\}$  to denote whether a static tunnel from IPSG  $j$  to CPE  $k$  is established. Assume that the cost of sending traffic from node  $i$  to node  $j$  is  $c_{ij}$ , and the cost of sending traffic from node  $j$  to node  $k$  is  $d_{jk}$ .

The binary variable  $y_j \in \{0, 1\}$  is 1 if IPSG  $j$  is provisioned for the customer to send traffic to at least one of its CPEs, and it is 0 otherwise. We use  $f_j$  as the current cost of using IPSG node  $j$ . For a given customer, at most one provision is considered at any IPSG. Therefore  $f_j$  has a fixed value when only one customer is considered at a time. As a result, for a customer, the cost of the links over which a dynamic tunnel is established (which is the cost from MAPs to IPSGs) is  $C_{C1} = \sum_{i \in P, j \in Q, k \in R} c_{ij} x_{ijk}$ . Similarly, the cost of the links over which a static tunnel is established (which is the cost from IPSGs to CPEs) is  $C_{C2} = \sum_{j \in Q, k \in R} d_{jk} z_{jk}$ . Therefore, the total connection cost is  $C_C = C_{C1} + \beta C_{C2}$ , where  $\beta$  is the relative weight on the cost of the links over which a static tunnel is established. The total provisioning cost is  $C_V = \sum_{j \in Q} f_j y_j$ . Thus, the total cost is  $C = C_C + \alpha C_V$  where  $\alpha$  is the relative weight on the provision cost. Without loss of generality, we assume the revenue is 1. Therefore, the profit for provisioning the customer is  $G = \gamma - C$  where  $\gamma$  is the relative weight on revenue compared to total cost. The optimization problem formulation can then be specified as

$$\max G = \gamma - C \quad (1)$$

$$C = \left( \sum_{i \in P, j \in Q, k \in R} c_{ij} x_{ijk} + \beta \sum_{j \in Q, k \in R} d_{jk} z_{jk} \right) + \alpha \sum_{j \in Q} f_j y_j \quad (2)$$

$$x_{ijk} \in \{0, 1\}, \forall i \in P, \forall j \in Q, \forall k \in R \quad (3)$$

$$z_{jk} \in \{0, 1\}, \forall j \in Q, \forall k \in R \quad (4)$$

$$y_j \in \{0, 1\}, \forall j \in Q \quad (5)$$

$$\sum_{j \in Q} x_{ijk} = 1, \forall i \in P, \forall k \in R \quad (6)$$

$$x_{ijk} \leq z_{jk}, \forall i \in P, \forall j \in Q, \forall k \in R \quad (7)$$

$$z_{jk} \leq y_j, \forall j \in Q, \forall k \in R \quad (8)$$

Note that condition (6) specifies that exactly one link out of a MAP is chosen to go to one CPE, implying that traffic from a MAP to a CPE is sent to only one IPSG. Condition (7) specifies that only one tunnel is established between an IPSG and

a CPE even if traffic from multiple MAPs are going through the IPSG to reach the CPE. That is

$$z_{jk} = \begin{cases} 1 & \text{if } \sum_{i \in P} x_{ijk} > 0, \forall j \in Q, \forall k \in R \\ 0 & \text{otherwise.} \end{cases}$$

This is equivalent to condition (7) since  $z_{jk}$  is in the objective function  $G$ , and when  $x_{ijk} = 0, \forall i \in P$ , to maximize  $G$ ,  $z_{jk} = 0$  must be chosen.

Condition (8) specifies that even if an IPSG is provisioned to send traffic to more than one CPE, for the purpose of computing provision cost, it should be considered as only one provision. That is,

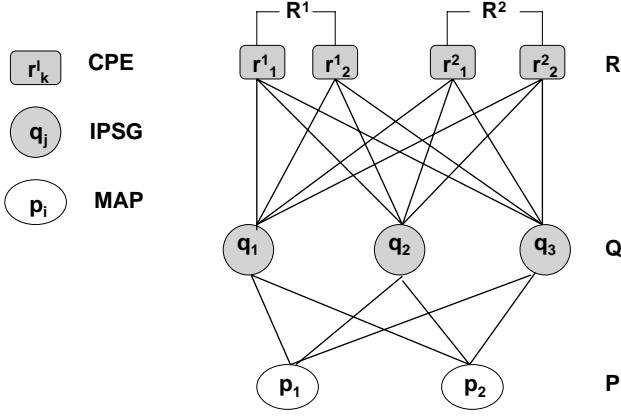
$$y_j = \begin{cases} 1 & \text{if } \sum_{k \in R} z_{jk} > 0, \forall j \in Q \\ 0 & \text{otherwise.} \end{cases}$$

This is equivalent to condition (8) since  $y_j$  is in the objective function  $G$ , and when  $z_{jk} = 0, \forall k \in R$ , to maximize  $G$ ,  $y_j = 0$  must be chosen.

**3.3.2. Multiple Customer Formulation** In the multiple customers case, we maximize the sum of the profit obtained by provisioning all customers (or a subset of customers as we discuss below), where the profit for each customer is calculated exactly the same way as in the single customer case in Section 3.3.1. All MAPs and IPSGs in the network are shared among all customers. However, each customer has its distinct set of CPEs.

In the single customer case, the provision cost  $f_j$  at each IPSG  $j$  has a fixed value, and an IPSG that has reached its provision capacity is not considered, which is equivalent to setting  $f_j = \infty$ . When multiple customers are considered,  $f_j$  is assigned a fixed value for all customers provisioned on IPSG  $j$ , however, because multiple customers can be provisioned at each IPSG, care must be taken to ensure that the number of customers provisioned does not exceed the provision capacity of each IPSG. Moreover, when multiple customers are considered at the same time, not every customer should be provisioned in the network. Priorities should be given to customers providing maximum profit. There are two cases where a customer is rejected. One case is when there is no more provision capacity left on any IPSG in the network, the other case is when provisioning this customer results in negative profit, meaning a loss. Essentially to maximize the total profit, a subset of the customers are provisioned. The rest of the customers are rejected because either the provision capacity is reached or they produce a loss instead of profit.

The optimization problem for multiple VPN customers can be described as follows. Let  $T$  be the set of VPN customers to consider and  $|T| = L$ . Let  $P$  be the set of all MAPs,  $Q$  be the set of all IPSGs, and  $R$  be the set of all CPEs for all customers where  $R = \{R_1, R_2, \dots, R_l, \dots, R_L\}$  and  $R_l$  is the set of CPEs for customer  $l \in T$ . Let  $w^l$  be the binary variable specifying if customer  $l$  should be provisioned in the network. For each customer  $l$  provisioned, every node  $i$  in  $P$  and every node  $k$  in  $R_l$ , choose an IPSG node  $j$  in  $Q$ , to forward traffic through a dynamic tunnel between  $i$  and  $j$ , and a static



**Figure 4. Example graph model for multiple customers.**

tunnel between  $j$  and  $k$ , such that the total profit for all customers is maximized. Needless to say, for a customer not provisioned, the cost is 0.

An example of the graph for multiple customers is shown in Figure 4. There are two MAPs,  $p_1$  and  $p_2$ , three IPSGs,  $q_1$ ,  $q_2$ , and  $q_3$  in the network, and two customers. For customer 1, there are two CPEs  $r_1^1$ , and  $r_2^1$ , and for customer 2, there are two CPEs  $r_1^2$ , and  $r_2^2$ .

For customer  $l \in T$ , we denote by the binary variable  $x_{ijk}^l \in \{0, 1\}$  whether a dynamic tunnel between node  $i \in P$  and node  $j \in Q$  is used for the traffic from MAP  $i$  to CPE  $k \in R_l$ . We use binary variable  $z_{jk}^l \in \{0, 1\}$  to denote whether a static tunnel from IPSG  $j$  to CPE  $k$  is established. The cost of sending traffic from node  $i$  to node  $j$  is  $c_{ij}$ . Notice that the cost is the same for all customers, therefore index  $l$  is not needed. The cost of sending traffic from node  $j$  to node  $k$  is  $d_{jk}^l$ .

The binary variable  $y_j^l \in \{0, 1\}$  is 1 if IPSG  $j$  is provisioned for customer  $l$  to send traffic to at least one of its CPEs, and it is 0 otherwise. We use  $P_{cap}$  as the maximum number of customers that can be provisioned on each IPSG, and  $f_j$  as the cost for customer  $l$  to use node  $j$ . As long as the provision capacity of IPSG  $j$  has not been reached, the provision cost for each customer is the same, therefore index  $l$  is not needed.

For a single customer  $l \in T$  under consideration, the cost of links over which a dynamic tunnel is established (which is the cost from MAPs to IPSGs) is  $C_{C1}^l = \sum_{i \in P, j \in Q, k \in R_l} c_{ij} x_{ijk}^l$ . The cost of links over which a static tunnel is established (which is the cost from IPSGs to CPEs) is  $C_{C2}^l = \sum_{j \in Q, k \in R_l} d_{jk}^l z_{jk}^l$ . The total connection cost is therefore,  $C_C^l = C_{C1}^l + \beta C_{C2}^l$ , where  $\beta$  is the relative weight on the cost of links over which a static tunnel is established. The total provisioning cost for customer  $l$  is  $C_V^l = \sum_{j \in Q} f_j y_j^l$ . Thus, the total cost is  $C^l = C_C^l + \alpha C_V^l$  where  $\alpha$  is the relative weight on the provision cost. Without loss of generality, we assume the revenue for each customer provisioned to be the same. Naturally, both the rev-

enue and cost are zero for each customer not provisioned. The profit is therefore  $G^l = \gamma w^l - C^l$ , where  $\gamma$  is the relative weight on revenue compared to cost. The optimization problem formulation can then be specified as

$$\max G = \sum_{l \in T} G^l \quad (9)$$

$$G^l = \gamma w^l - C^l, \forall l \in T \quad (10)$$

$$C^l = \left( \sum_{i \in P, j \in Q, k \in R_l} c_{ij} x_{ijk}^l + \beta \sum_{j \in Q, k \in R_l} d_{jk}^l z_{jk}^l \right) + \alpha \sum_{j \in Q} f_j y_j^l \quad (11)$$

$$w^l \in \{0, 1\}, \forall l \in T \quad (12)$$

$$x_{ijk}^l \in \{0, 1\}, \forall l \in T, \forall i \in P, \forall j \in Q, \forall k \in R_l \quad (13)$$

$$z_{jk}^l \in \{0, 1\}, \forall l \in T, \forall j \in Q, \forall k \in R_l \quad (14)$$

$$y_j^l \in \{0, 1\}, \forall l \in T, \forall j \in Q \quad (15)$$

$$\sum_{j \in Q} x_{ijk}^l = w^l, \forall l \in T, \forall i \in P, \forall k \in R_l \quad (16)$$

$$x_{ijk}^l \leq z_{jk}^l, \forall l \in T, \forall i \in P, \forall j \in Q, \forall k \in R_l \quad (17)$$

$$z_{jk}^l \leq y_j^l, \forall l \in T, \forall j \in Q, \forall k \in R_l \quad (18)$$

$$\sum_{l \in T} y_j^l \leq P_{cap}, \forall j \in Q \quad (19)$$

Compared with the formulation for one customer, Condition (19) is added to specify that the total number of provisions on each IPSG  $j$  cannot exceed its capacity  $P_{cap}$ . Moreover, a new variable  $w^l$  is introduced to specify if a customer  $l$  is provisioned or not, and Condition (16) is modified to specify that if customer  $l$  is provisioned, exactly one link out of every MAP is chosen to go to one CPE for this customer through one IPSG, otherwise no link out of any MAP is chosen and no IPSG is provisioned for this customer.

### 3.4. Selection of IPSGs with Fault Tolerance

We describe two different formulations for fault tolerance, one where we put a minimum bound on the number of IPSGs on which each customer should be provisioned and the other where we put an exact bound on the number of IPSGs on which each customer should be provisioned.

**3.4.1. Minimum Bound on Number of IPSGs** In order to provide fault tolerance, for every customer, each traffic session from a MAP to a CPE should have the option of going through  $N > 1$  IPSGs. In case  $N - 1$  IPSGs fail, traffic sessions can still be established using the functioning IPSG. The only modification to the formulation without fault tolerance consideration in Section 3.3.2 is to substitute Condition (16) with

$$\sum_{j \in Q} x_{ijk}^l = N w^l, \forall l \in T, \forall i \in P, \forall k \in R_l \quad (20)$$

Condition (20) specifies for each customer  $l \in T$  that is provisioned, there must be  $N$  connections established between a MAP and a CPE each going through a separate IPSG. Because each pair of MAP and CPE requires the use of a set of

$N$  IPSGs, and these IPSG sets can overlap, therefore the total number of IPSGs used for customer  $l$  is greater than or equal to  $N$ . In other words, this formulation specifies the minimum number of IPSGs provisioned for each customer.

**3.4.2. Exact Bound on Number of IPSGs** Another way to consider fault tolerance is to require that each customer can only use exactly  $N$  IPSGs for all its connections. This would require one more condition to be added to the formulation in Section 3.4.1 as follows:

$$\sum_{j \in Q} y_j^l = Nw^l, \forall l \in T \quad (21)$$

Condition (21) specifies exactly  $N$  IPSG nodes can be used for all the connections for a provisioned customer  $l$ .

## 4. Performance Evaluation

We study the IPSG selection problem presented in Section 3 under a representative network topology and parameter setting. We evaluate the effect of relative weight  $\alpha$  on provision cost,  $\beta$  on the cost of links over which a static tunnel is established, and  $\gamma$  on revenue.

### 4.1. Cost Computation

In order to solve the integer programming problem in Section 3, connection cost  $c_{ij}$ ,  $d_{jk}$  and provision cost  $f_j$  need to be assigned appropriate values. The cost computation can be adapted to fit the NSP's design objectives. This makes our formulation quite general and can be used for different scenarios in addition to guaranteeing connectivity for VPN customers.

Connection cost is a function of the parameters that the NSP wants to control. Suppose the NSP wants to satisfy a special requirement from a VPN customer such that the users of this customer are not switched to a remote lightly loaded IPSG even if that reduces the total cost for the NSP. To elaborate, an MN on the east coast trying to access corporate intranet on the east coast should not be switched to an IPSG on the west coast even if the total cost is minimized with this solution. To take the constraint into account, we restrict the number of hops allowed from a MAP to a CPE and modify the link cost of the graph as

$$c_{ij} = \infty, \text{ if } c_{ij} > L1_{max}, \forall i \in P, \text{ and } \forall j \in Q \quad (22)$$

$$d_{jk} = \infty, \text{ if } d_{jk} > L2_{max}, \forall j \in Q, \text{ and } \forall k \in R \quad (23)$$

where  $L1_{max}$  and  $L2_{max}$  are the maximum number of hops allowed for the tunnel between MAP and IPSG and the tunnel between IPSG and CPE respectively.

When a single customer is considered at a time, we have the option of setting provision cost to reflect the existing number of provisions at each IPSG. For example, we can use  $f_j = cap_j/avail_j$ , where  $cap_j$  is the capacity of IPSG  $j$  and  $avail_j$  is the number of available provisions left. This cost assignment will result in even distribution of the number of provisions per IPSG across all IPSGs.

However, when multiple customers are considered at the same time, the provision cost for different customers has to be the same to be a valid input to the integer programming program. Without loss of generality, we set  $f_j = 1$  for IPSG  $j$  for all customers.

The cost computation phase accounts for customer specific requirements. After the cost refinement, we use the integer programming packages CPLEX [9] to solve the IPSG selection problem.

### 4.2. Network Topology Selection

We use the Tier random topology generator [11] to produce a representative network topology for the NSP. On the top level, the structure is a Wide Area Network (WAN) of 10 nodes, connecting 30 Metropolitan Area Networks (MANs). Each MAN consists of 15 nodes and interconnects 30 Local Area Networks (LANs). For simplicity, we create each LAN with one node. The network therefore has a total of  $10 + 30 \times (15 + 30) = 1360$  nodes. We assign the degree of intranetwork and internetwork redundancy on this network structure to create the representative topology. The degree of intranetwork redundancy is defined as the degree (number of links) from a node to other nodes within the same network, while the degree of internetwork redundancy is defined as the number of connections between the two networks. For the representative topology, the degrees of intranetwork redundancy in a WAN, MAN and LAN are 3, 2 and 1, respectively; the degree of internetwork redundancy between a MAN and the WAN is 2, and between a LAN and a MAN is 1.

We randomly assign IPSGs to the  $30 \times 15 = 450$  MAN nodes, and MAPs and CPEs to the  $30 \times 30 = 900$  LAN nodes. Additionally, we make sure that no single node can host more than one server be it an IPSG, a MAP or a CPE. In our simplified model with one node per LAN, by requiring that no node can host more than one server, we easily guarantee that MAPs and CPEs are not located in the same LAN. Furthermore, because we do not consider the actual Mobile Nodes (MNs) in our IPSG selection problem, using one node per LAN is sufficient.

When an enterprise customer signs up for the VPN service, it provides the NSP the location of its CPEs. The NSP can process customer requests in two ways. In the simple approach, it can process customer requests one at a time using our formulation in Section 3.3.1. This approach cannot guarantee that all customers are optimally provisioned to maximize the profit for NSP. The other approach is to consider multiple customers at a time as in Section 3.3.2. Because the formulation with multiple customers achieves a globally optimal solution, and the single customer formulation is a special case of the multi-customer one, we focus on the multi-customer formulation.

We select a reasonable sized network. We assign 5 IPSGs and 10 MAPs in the network and 1 CPE for each customer. We assume each IPSG can make at most 20 provisions, the total number of provisions available in the network is therefore  $5 \times 20 = 100$ .

We assign the number of hops from MAP node  $i$  to IPSG node  $j$ , from IPSG node  $j$  to CPE node  $k \in R_l$  for customer  $l$  in the network to cost  $c_{ij}$  and cost  $d_{jk}^l$  respectively. We also assign  $f_j^l = 1$  for IPSG node  $j$  and customer  $l$ . For the fault tolerance case we choose  $N = 2$ .

In reality, the number of Mobile-VPN customers served by a service provider is expected to be in the hundreds. The number of virtual routing instances (and hence the number of customers) that IPSGs can handle is in the order of hundreds as well [6, 13]. This means, the number of IPSGs that will be deployed in a service provider network will not be more than a handful. These considerations lead us to the numbers we use for our results shown below.

We find that the commercial integer programming solver CPLEX [9] running on a SUN UltraSPARC workstation Ultra-80 running Solaris 8 is able to generate the optimal solution in a very short time (in the order of minutes) for the realistic problem sizes that are used below. For example, it takes only a few seconds to solve the 20 customer case, less than 10 minutes to solve the 100 customer case, and less than 60 minutes for the 500 customer case. In general, the network provisioning problem is solved off-line when customers sign up with a service provider for Mobile VPN service and thus the solution does not have to be generated in real-time. Therefore, we can find optimal solutions to realistic problem sizes with CPLEX without using heuristics.

### 4.3. Case with 20 customers

To study the effect of  $\alpha$  and  $\beta$ , we consider 20 customers as a group. Because each IPSG can provision upto 20 customers, each customer has the option of making provisions on all 5 IPSGs. Therefore we can see a broad range of provisions used by each customer as  $\alpha$  and  $\beta$  changes. In order to completely eliminate the effect of  $\gamma$ , we set  $\gamma$  to be a relatively large number. Therefore, all the 20 customers are accepted because the profit for a customer is always positive.

In Figures 5 and 6 we plot the average number of provisions used by each customer as  $\alpha$  and  $\beta$  varies and fault tolerance is not considered. Observe from Figure 5 that for a fixed value of  $\beta$ , as  $\alpha$  increases the average number of provisions per customer decreases to 1 and stays at 1 when  $\alpha$  reaches 7. Note that  $\alpha$  is the relative weight on the provision cost. So when  $\alpha = 0$ , provision cost is not considered in the formulation at all, and the customers use multiple provisions so long as the connection cost is minimized, which in turn maximizes the profit. The extreme case happens when  $\alpha = 0$  and  $\beta = 0$ . The only cost is the cost of links over which a dynamic tunnel is established (which is from MAPs to IPSGs). To minimize this cost, each customer would choose the shortest path connection from each MAP to an IPSG, and each customer uses 4 provisions in this particular network setting. For a fixed  $\beta$ , as  $\alpha$  increases, provision cost increases, therefore each customer tends to use less number of provisions to keep total cost minimized.

Now let us examine the effect of  $\beta$  which is the relative weight on the cost of links over which a static tunnel is es-

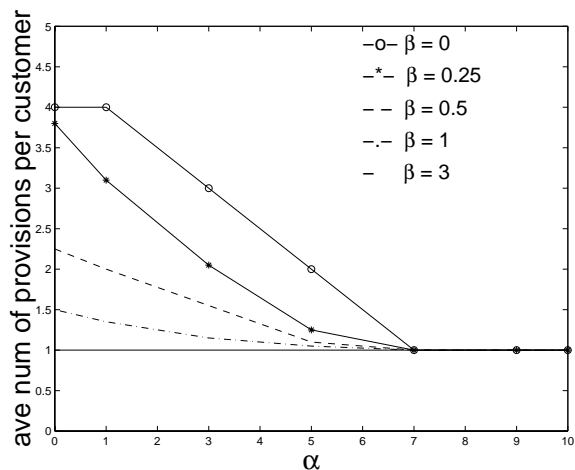


Figure 5. Effect of  $\alpha$  without fault tolerance (with 20 customers).

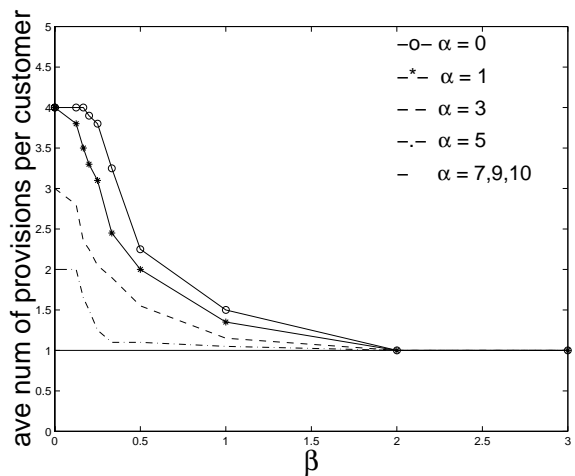
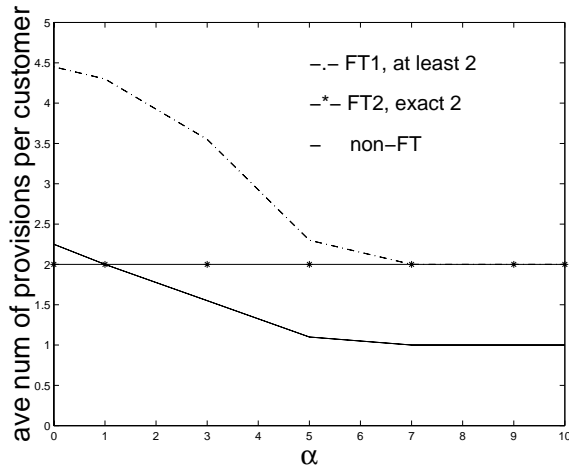


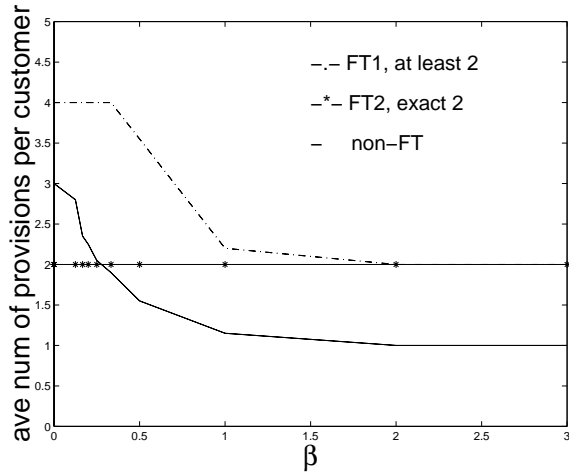
Figure 6. Effect of  $\beta$  without fault tolerance (with 20 customers).

tablished (which is from IPSGs to CPEs). Note that the cost of links over which a dynamic tunnel is established (from MAPs to IPSGs) represent paths in wireless access networks, whereas the cost of links over which a static tunnel is established (from IPSGs to CPEs) represent paths in the backbone networks. Therefore it is natural to use  $\beta$  to distinguish their relative importance. For a fixed  $\alpha$ , as  $\beta$  increases, the cost of links over which a static tunnel is established increases. Therefore the solution reduces the cost by using smaller number of tunnels from IPSGs to the CPEs. As a result, the average number of provisions per customer decreases. This can be seen in both Figure 5 and Figure 6.

To study the effect of fault tolerance, we compare the average number of provisions per customer for the two fault tolerance formulations and the non-fault-tolerance formula-



**Figure 7. Effect of  $\alpha$  and fault tolerance trade-offs (with 20 customers and  $\beta = 0.5$ ).**



**Figure 8. Effect of  $\beta$  and fault tolerance trade-offs (with 20 customers and  $\alpha = 3$ ).**

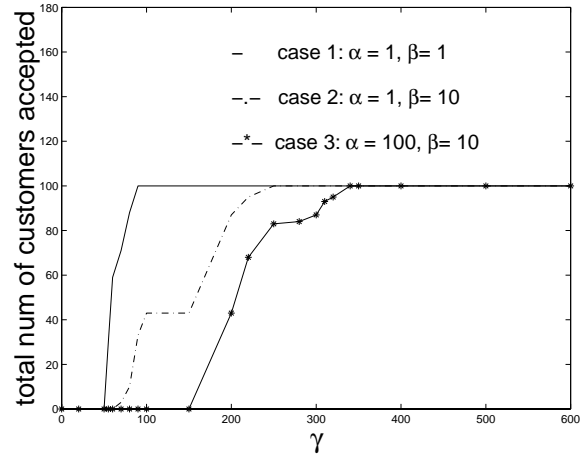
tion. Let FT1 and FT2 denote the minimum bound and exact bound on the number of IPSGs, respectively, for the fault tolerance formulation. Figure 7 shows for a fixed  $\beta$ , as  $\alpha$  increases, the average number of provisions stays the same at 2 in FT2 case where each customer is required to have  $N = 2$  provisions, and decreases in the other two cases. In FT1 case, each MAP for each customer needs to be provisioned at two IPSGs, therefore the number of provisions per customer for all MAPs is at least two. Compared with the non-FT case, for a fixed  $\alpha$  and  $\beta$  value, FT1 uses roughly twice as many provisions as the non-FT case. This is because each MAP needs to be provisioned at two different IPSGs in the FT1 case.

Similarly, Figure 8 shows for a fixed  $\alpha$ , as  $\beta$  increases, the average number of provisions decreases in the FT1 and non-FT cases, and FT1 uses roughly twice as many provisions as

the non-FT case.

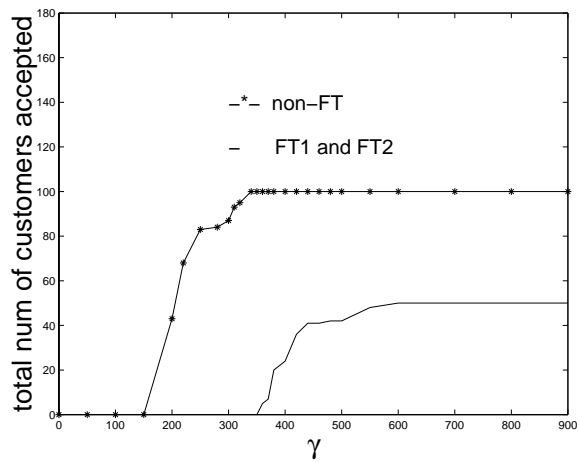
#### 4.4. Case with 100 customers

Intuitively, if  $\gamma$  is greater than the total cost for a customer, this customer should be accepted. On the other hand, if there is no room to provision a customer with total cost less than  $\gamma$ , it should be rejected. When the number of customers considered (100 in this case) is greater than or equal to the total number of provisions available in the system (100 in this case), the solution to the integer programming formulation will accept as many customers as possible while keeping the number of provisions per customer minimum. Compare two possible solutions: the first one accepts  $m$  customers, and one of them, customer  $l$  uses multiple provisions. The second solution accepts  $m + 1$  customers, and every one uses 1 provision. If the profit from provisioning the  $(m + 1)$ -th customer is more than the cost savings from using multiple provisions for customer  $l$ , the  $(m + 1)$ -th customer is provisioned while customer  $l$  is provisioned using one less IPSG than before. This is due to the objective of maximizing the total profit from all customers. Because of this, only one IPSG is provisioned for every customer and in the non-FT case, and only two IPSGs are provisioned for every customer for both FT1 and FT2 cases in this section.



**Figure 9. Effect of  $\gamma$  (with 100 customers).**

To study the effect of  $\gamma$ , we consider 100 customers and plot the number of accepted customers for a different combinations of  $\alpha$  and  $\beta$  values. Figure 9 shows for fixed sets of  $\alpha$  and  $\beta$  values, when  $\gamma$  is below some low threshold value, no customer is accepted, and when  $\gamma$  is above some high threshold value, all 100 customers are accepted. As  $\gamma$  increases from the low threshold to high threshold value, the number of accepted customers increases from 0 to 100. The low threshold value is the smallest total cost of any customer, and the high threshold value is largest total cost of any customer. Naturally the set of threshold values is different for different sets of  $\alpha$  and  $\beta$  values.



**Figure 10. Effect of  $\gamma$  and fault tolerance trade-offs (with 100 customers,  $\alpha = 100$  and  $\beta = 10$ ).**

When fault tolerance is considered (see Figure 10), for both FT1 and FT2 cases, the number of accepted customers increases from 0 to 50 as  $\gamma$  increases from the low threshold to high threshold. Because each customer requires 2 IPSGs in both FT1 and FT2 cases, the maximum number of accepted customers is 50. As the total cost of provisioning a customer using FT1 and FT2 is roughly twice as much as the non-FT case, the low threshold value for FT1 and FT2 is roughly twice as much as the one for the non-FT case. So is the case for the high threshold value.

#### 4.5. Case with more than 100 customers

When we consider more than 100 customers, the same trend in Section 4.4 is observed as the solution picks the lowest cost customers to provision to maximize total profit.

### 5. Conclusion and Future Work

This paper identifies the major differences between the traditional VPN and the mobile VPN, both based on the network-based VPN model, and proposes a hierarchical architecture using two network elements, namely the MAPs and IPSGs, to provide mobile VPN services. In order to optimally use the network elements, we identify several cost parameters that play important roles in designing the network. In particular, we study the problem of provisioning IPSGs for mobile VPN customers in order to minimize the total connection cost of links over which VPN tunnels are established and the cost of provisioning IPSGs for a set of customers.

We propose a generic and powerful problem formulation considering a number of practical requirements such as fault tolerance. In our on-going work, we are expanding the formulation to consider additional concerns such as guaranteeing bandwidth for VPN connections.

### References

- [1] 3rd generation partnership project. <http://www.3gpp.org>.
- [2] 3rd Generation Partnership Project. Combined GSM and Mobile IP Mobility Handling in UMTS IP CN. *Technical Specification Group: Services and System Aspects, 3GPP*, May 2000.
- [3] 3rd Generation Partnership Project 2. Developing the Next Generation CDMA 2000 Wireless Communications. <http://www.3gpp2.org>.
- [4] B. Aboba and M. Beadles. The Network Access Identifier. *RFC-2486*, Jan 1999.
- [5] R. Cohen and G. Kaempfer. On the Cost of Virtual Private Networks. *IEEE/ACM Transactions on Networking*, Dec 2000.
- [6] CoSine Communications. IP Service Generators. [http://www.cosinecom.com/library/ip95\\_ipsg\\_ds.html](http://www.cosinecom.com/library/ip95_ipsg_ds.html).
- [7] T. Hiller et al. CDMA2000 Wireless Data Requirements for AAA. *RFC-3141*, Jun 2001.
- [8] The Yankee Group. Wireless VPN: The Mobile Enterprise Solution. Technical Report 5, May 2000.
- [9] ILOG. Cplex. <http://www.ilog.com/products/cplex/>.
- [10] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. *RFC-2401*, Nov 1998.
- [11] M.B.Doar. A Better Model for Generating Test Networks. In *Proc. of Globecom'96*, 1996.
- [12] J. Moy. OSPF Version 2. *RFC-1583*, Mar 1994.
- [13] Nortel Networks. Shasta Portfolio. <http://www.nortelnetworks.com/products/01/shasta/>.
- [14] Starent Networks. Intelligent Mobile Gateway. <http://www.starentnetworks.com>.
- [15] B. Perlmutter. *Virtual Private Networking: A View from the Trenches*. Prentice Hall PTR, 2000.
- [16] E. Rosen and Y. Rekhter. BGP/MPLS VPNs. *RFC-2547*, Mar 1999.
- [17] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, and B. Palter. Layer Two Tunneling Protocol "L2TP". *RFC-2661*, Aug 1999.