

Solution Brief

# How Different VPN Approaches Affect Site-to-Site Scalability and Connectivity

---

*Comparison of Rule-based, Route-based and Dynamic Route-based VPNs*

Sarah Sorensen  
Product Marketing Manager



Juniper Networks, Inc.  
1194 North Mathilda Avenue  
Sunnyvale, CA 94089 USA  
408 745 2000 or 888 JUNIPER  
[www.juniper.net](http://www.juniper.net)

Part Number: 351037-001

---

---

## Contents

Introduction.....	3
Rule-Based VPN Connectivity.....	4
Rule-Based Setup Approach Method #1: .....	4
Rule-based Setup Approach Method #2.....	7
Connectivity Problems with Rulebases .....	9
Rule-based Approach Summary .....	9
Static Route-Based VPN Connectivity .....	10
Static Route-based Approach Summary.....	12
Dynamic Route-Based VPN Connectivity.....	12
Dynamic Route-Based Approach Summary .....	14
Conclusion .....	14

## Introduction

There are many different approaches to setting up a site-to-site IPSec VPN. Each approach has implications for the overall connectivity of the network and the amount of time needed to administer and maintain that connectivity. The returns diminish with a solution that requires a lot of manual intervention to keep the private network running, so it is important that the IPSec VPN:

- Be easy to manage for all types of network configurations and administratively scale
- Automatically learn and incorporate network topology changes
- Minimize the need for human resources
- Leverage the dynamic nature of the network to increase connectivity

This paper will go into detail on the three different approaches to site-to-site VPNs, including rule-based, static route-based and Juniper Networks dynamic route-based. It will demonstrate how each approach affects the overall scalability and, ultimately, connectivity of the solution. Following is a brief definition of each approach:

**Rule-based VPNs** work by defining the network topology (IP addresses), in one way or another, and then dictating, based on that topology, who can talk to whom in a secure manner. While this approach does simplify some VPN deployments, the coupling of the network topology (IP addresses) with the transport of the VPN connection means the network cannot easily accommodate changes or the complex needs of widely distributed networks.

**Static route-based VPNs** separate the physical network from the abstract VPN network to simplify deployment and management. With static route-based VPNs, organizations define the VPN overlay links and then define the static routes that will be used for transport, allowing the route, rather than a policy, to determine which traffic goes through the VPN. This provides some flexibility over a rule-based approach, but still requires resources to make changes to the route tables any time networks are added, deleted or changed.

**Juniper's Dynamic route-based VPNs** leverage the dynamic nature of the network and enable dynamic routing protocols over the VPN tunnels to completely separate the transport, forwarding decision from the VPN connection, giving enterprises the flexibility they need to efficiently manage the constant changes inherent in complex networks. Dynamic route-based VPNs provide Layer 2-like network operation, but with more security and flexibility.

## Rule-Based VPN Connectivity

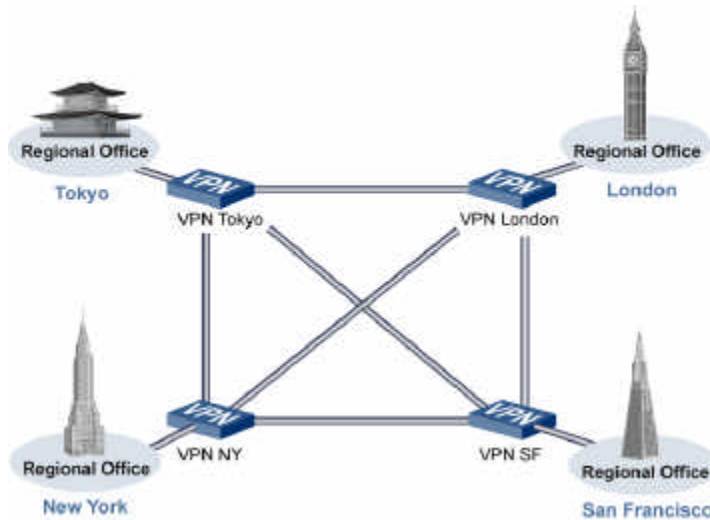
The rule-based approach that has worked so well for firewalls has its limitations when applied to large-scale networks, generating a lot of work for organizations during the deployment and ongoing management of the network. The reason is that rule-based VPNs tie specific traffic and services, or source and destination groups, to one particular IPSec tunnel, essentially binding the VPN connection to a static route. This has resiliency and scalability consequences, which are important to understand. This section will look at the two different ways to set up a rule-based VPN and what that means for the scalability of the solution. They are:

- Method #1: Define the network topology and then define the policy.
- Method #2: Define the topology within the policy

### Rule-Based Setup Approach Method #1:

The first is to define the network topology and then define the policy.

Step 1) The administrator defines the VPN and then the network for which that VPN is responsible, which means all of the IP addresses or the IP address ranges, from servers to work stations.



1

VPN Device	Encryption Domain
VPN-Tokyo	Tokyo-Net (10.0.0.0/24)
VPN-London	London -Net (10.1.0.0/24)
VPN-NY	NY-Net (10.2.0.0/24)
VPN-SF	SF-Net (10.3.0.0/24)

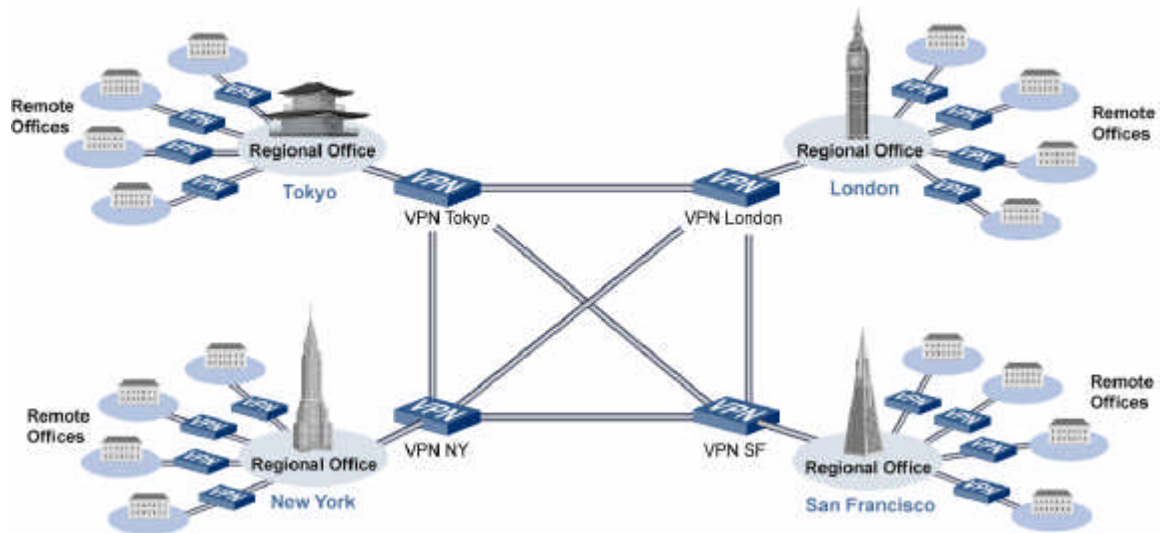
2

Policy			
Source	Destination	Service	Action
Tokyo Net, London Net, NY Net, SF Net	Tokyo Net, London Net, NY Net, SF Net	Any	Encrypt

Step 2) The administrator then defines the policy, which sets up who can talk to whom through the tunnel and what services they can run, most likely defined as “Any” service. The diagram below illustrates what the two steps might look like and the resulting configuration:

Assuming that the organization can easily understand and define its network topology, this approach can be fairly simple to set up. However, as a network grows, it becomes harder to scale, adding complexity to the deployment and ongoing management of the solution. Any change in the network has to be changed in the topology definition. For example, if an IP subnet mask gets changed or a new subnet is added to or removed from the network, the change needs to be made in the network definitions.

Other limitations also exist. What if an organization wants to do a full mesh at the core and a hub and spoke configuration for satellite offices to take advantage of regional controls or performance SLAs? This hybrid configuration, which would look something like the diagram below, is very common.

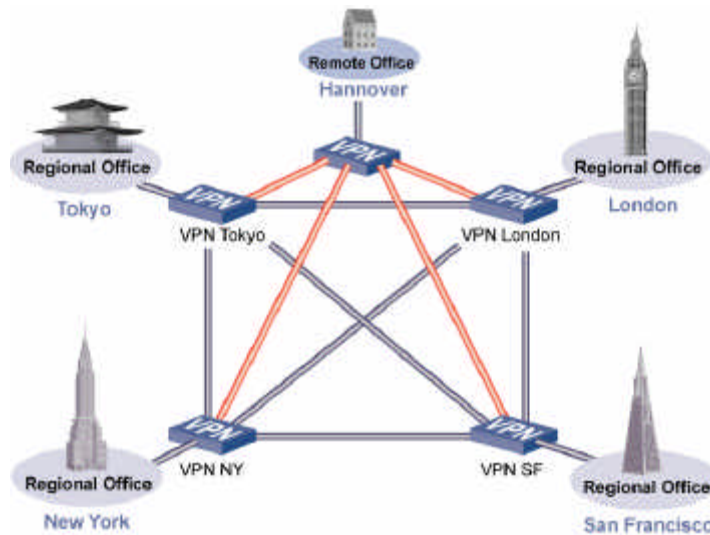


This rule-based approach, however, cannot support it. This is because each gateway has to see the exact same topology as all of the other gateways. What does this really mean? Remember in step one in the VPN set up an administrator defines the VPN device and the network for which that VPN device is responsible. If an administrator wants to add a satellite office, for example Hannover, and have all traffic to and from that network go through the London regional office it can't. Why, because to add a Hannover VPN the administrator will need to define it and, in the process, make it responsible for the Hannover network. The administrator can't make the London VPN responsible for the Hannover network because, in order for this to happen, Hannover would have to see London as responsible for the rest of the network, and the rest of the network

VPN Device	Encryption Domain
<b>VPN-Tokyo</b>	<b>Tokyo-Net (10.0.0.0/24)</b>
<b>VPN-London</b>	<b>London-Net (10.1.0.0/24)</b>
<b>VPN-NY</b>	<b>NY-Net (10.2.0.0/24)</b>
<b>VPN-SF</b>	<b>SF-Net (10.3.0.0/24)</b>
<b>VPN-Hannover</b>	<b>Hannover-Net (10.4.0.0/24)</b>

would have to see London as responsible for Hannover.

Instead, Hannover sees each of the VPNs as responsible for their own networks, and the rest of the network sees Hannover as responsible for the Hannover network, forcing the organization to create a direct connection to each gateway, or a full mesh, such as the one that follows:



1

VPN Device	Encryption Domain
VPN-Tokyo	Tokyo-Net (10.0.0.0/24)
VPN-London	London -Net (10.1.0.0/24)
VPN-NY	NY-Net (10.2.0.0/24)
VPN-SF	SF-Net (10.3.0.0/24)
VPN-Hannover	Hannover-Net (10.4.0.0/24)

2

Policy			
Source	Destination	Service	Action
Tokyo Net, London Net, NY Net, SF Net, Hannover Net	Tokyo Net, London Net, NY Net, SF Net, Hannover Net	Any	Encrypt

Full mesh might be fine for a limited number of sites, but quickly becomes unmanageable for a large distributed network.

## Rule-based Setup Approach Method #2

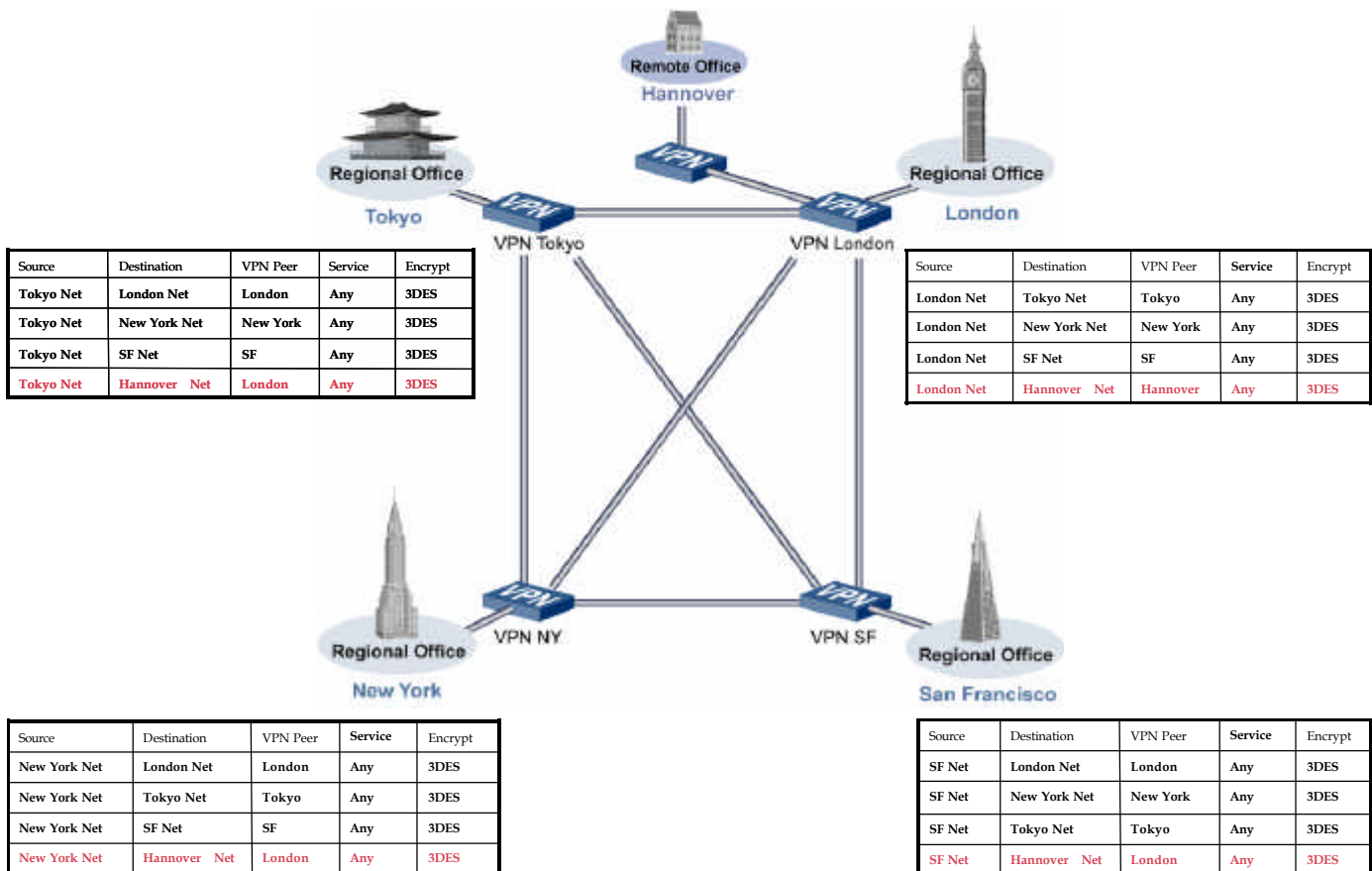
The second approach to rule-based VPNs defines the topology within the policy. This means that an administrator defines, in the policy, which network resources need to talk in a secure manner and to which peer gateway they should send the traffic. For example, the Tokyo rulebase would look something like the following:

Source	Destination	VPN Peer	Service	Encrypt
<b>Tokyo Net</b>	<b>London Net</b>	<b>London</b>	<b>Any</b>	<b>3DES</b>
<b>Tokyo Net</b>	<b>New York Net</b>	<b>New York</b>	<b>Any</b>	<b>3DES</b>
<b>Tokyo Net</b>	<b>SF Net</b>	<b>SF</b>	<b>Any</b>	<b>3DES</b>

So, any traffic from the Tokyo network going to the London network would be sent to the London VPN (peer), which is responsible for that network. As a result, the administrator is inherently defining which VPN device is responsible for which network and what IP ranges are included in that network.

This is more resource intensive up front because the topology is defined in the policy, however, it gives the organization more flexibility in the long run. This approach makes it easier to define, at a more granular level, which services are allowed through each tunnel and how the traffic should be forwarded. This approach also enables organizations to more easily set up hybrid configurations, such as the below mixed mesh and hub and spoke topology.

Source	Destination	VPN Peer	Service	Encrypt
Hannover Net	London Net	London	Any	3DES
Hannover Net	Tokyo Net	London	Any	3DES
Hannover Net	SF Net	London	Any	3DES
Hannover Net	New York Net	London	Any	3DES



By defining the topology within the policy, the organization can dictate, for example, that all VPN traffic going from Hannover to New York has to go through the London office. This enables a mixed topology for better scalability than the first method, but still requires iterating through the policies any time a change is made to any network on each gateway. So when a network gets added, removed or changed, it needs to be added, removed or changed in the policies. As a result, this approach is very difficult to configure and maintain for large, complex networks.

## Connectivity Problems with Rulebases

What happens if something within the Internet goes down or a network connection becomes unavailable? With both rule-based approaches, if something happens to a particular connection the VPN connection also goes down and the enterprise suffers lost connectivity and productivity. The reason is because the network topology is defined within the policy, tying the VPN connection to a static route. For example, traffic

Source	Destination	VPN Peer	Service	Encrypt
<b>Tokyo Net</b>	<b>London Net</b>	<b>London</b>	<b>Any</b>	<b>3DES</b>
<b>Tokyo Net</b>	<b>New York Net</b>	<b>New York</b>	<b>Any</b>	<b>3DES</b>
<b>Tokyo Net</b>	<b>SF Net</b>	<b>SF</b>	<b>Any</b>	<b>3DES</b>
<b>Tokyo Net</b>	<b>Hannover Net</b>	<b>London</b>	<b>Any</b>	<b>3DES</b>

from the Tokyo network to the London network gets “routed” to the London VPN (peer), as dictated in the rule. If something happens and that route is not available, the VPN goes down. Even if other routes are available the VPN cannot automatically use those alternatives. For instance, if Tokyo could potentially reach London by going through the New York VPN, it will not be able to do so until the administrator redefines the VPN peer (static route) in the rulebase to reroute the traffic through New York. As a result, site-to-site connectivity is lost until either the problem fixes itself or an administrator figures out what is wrong and makes a change to the rulebase. This might be adequate for small deployments, but for large, distributed organizations where VPN availability is of the essence, this approach cannot deliver the connectivity required.

## Rule-based Approach Summary

Both rule-based approaches require that the organization define its network topology in the rule-based policy, with any changes to the network resulting in necessary changes to the policy and on each and every device. Often the people responsible for the network are not the same people responsible for its security, so changes to the network, such as adding a server or changing an IP address, may go unknown until someone tries to send traffic and can't because it has not been added on the VPN gateway or in the corresponding policy. In addition, if something happens to the Internet connection or one of the VPN gateways goes down, the organization has to manually figure out what happened and then manually make a change to get the VPN up and running again. In the meantime, the VPN is down and site-to-site communication stopped, representing loss of productivity and money. With rule-based VPNs, there is a lot of time and effort needed to configure and manage the connectivity, with the effort growing exponentially with the complexity of the network. As a result, rule-based solutions may be fine for small deployments, but don't scale to meet the site-to-site connectivity requirements of large distributed networks.

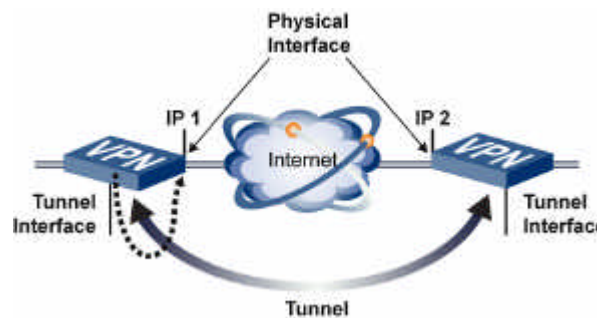
## Static Route-Based VPN Connectivity

Static route-based VPNs separate the physical network from the logical VPN network and allow routing, versus a policy, to determine what gets transported through the VPN tunnel. Route-based VPNs create logical VPN tunnels between destinations to establish the private network overlay and then the route table determines how the traffic gets there. In this way, an organization doesn't have to define the network topology with firewall-type rule sets, just the tunnel and the route.

To set up a route-based VPN, the organization needs to look at the basic topology of the network and determine which locations need to talk to each other, determining the ultimate configuration of the VPN. Assuming that the organization wants to do the configuration from the previous example, with a full mesh at the core and hub and spoke at the regional offices, the organization would go through two steps. First, it would define the gateways and second, define the route table.

To set up the VPN gateways, several things need to be defined. First, a tunnel interface on each VPN for each tunnel needs to be defined, then each of these interfaces needs to be associated with one or more actual physical interfaces where the traffic will be sent to get out to the Internet. Second, the VPN parameters need be defined, including such things as which encryption algorithm (3DES, AES) to use, which message integrity algorithm (MD5, SHA1), whether certificates or a pre-shared secret will be used, etc. Third, the IP address(es) or logical name (that can be resolved to an address) of the peer gateway's physical interface(s) needs to be provided. The last step in setting up the VPN is to configure the peer device reciprocally to ensure the tunnel is set up correctly. This, by the way, is similar to how the gateways would be defined for rule-based VPNs.

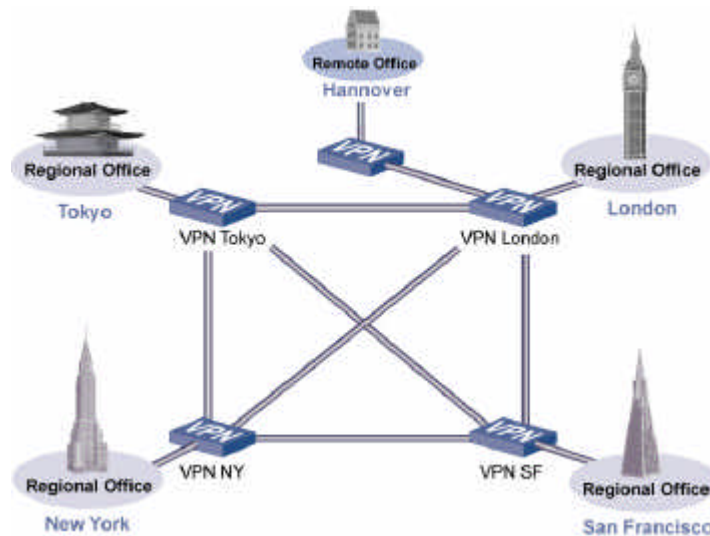
Sends traffic to virtual tunnel interface, but actually goes out encrypted through the physical interface, taking whatever route is best to



Once the VPN gateway is set up, a static routing table needs to be established for each VPN to define how traffic should be routed between the gateways. As opposed to rule-based VPNs that can only have a single static route associated between VPN peers, static route-based VPNs enable multiple routes to be defined between gateways. As a result, these alternate VPN connections offer a level of flexibility and resiliency over the rule-based approach. Below, is an example of static routes that have been defined for the VPN traffic coming from Tokyo to New York.

**Some Entries in Tokyo’s Route Table**

Source	Destination	Next Hop	Cost	Send to
Tokyo	New York	New York	10	Tunnel Interface L1
Tokyo	New York	London	20	Tunnel Interface L2
Tokyo	New York	San Francisco	23	Tunnel Interface L3



A “cost” is associated with each route, in relation to the directness of the connection. The “cheapest” route will always be the route of choice for the VPN. A mechanism identifies when a route becomes unavailable, and then removes that route from the routing table, so the VPN will use the next (lowest cost) tunnel to transport the VPN traffic.

While removing the tedious iteration of policies that needs to occur with rule-based VPNs every time the network changes, static route-based VPNs still require some manual intervention to maintain the connectivity of the network. For example, if a route changes, it needs to be manually updated in the route table for the VPN and its peer gateway. This can pose a lot of work for an administrator of a large distributed network. Another problem is that the administrator responsible for the network may not be the same as the administrator responsible for the VPN, so routing changes may not be made in the VPN in a timely manner; only getting called to the administrator’s attention when the VPN goes down. If a route becomes unavailable and no alternative route has been defined or is available, the site-to-site connectivity will be lost until an administrator is able to manually identify a new viable route and add it to the route table.

## Static Route-based Approach Summary

Static route-based VPNs build tunnels between VPN gateways without worrying about the networks for which they are responsible. This approach simplifies deployment, eliminating the need to define the network topology with firewall rule sets. However, it requires manual route statements be entered into a route table for each gateway, which can be time consuming and tedious for an administrator to create and maintain. Plus, an administrator needs to manually ensure that there is always an available route defined in the route table to maintain the connectivity of the VPN, which allows the solution to scale, but can be tedious.

## Dynamic Route-Based VPN Connectivity

Juniper Networks introduced the concept of dynamic route-based VPNs to the market to meet the requirements of a truly scalable solution. Dynamic route-based VPNs separate the physical network from the logical VPN network and allow routing, versus a policy, to automate the transport decisions. As a result, dynamic route-based VPNs are able to automatically learn network topology and available routes, minimizing the need for human intervention. Dynamic route-based VPNs can find an alternate path, when necessary, to maintain the connectivity of the network to match that offered by traditional solutions.

### Dynamic Routing

The Internet is made up of routers, which are continually “talking” to each other “advertising” the routes for which they are responsible. If a connection or route goes down for some reason, the routers will learn that the particular route is no longer available and look for an alternate path to get to the destination.

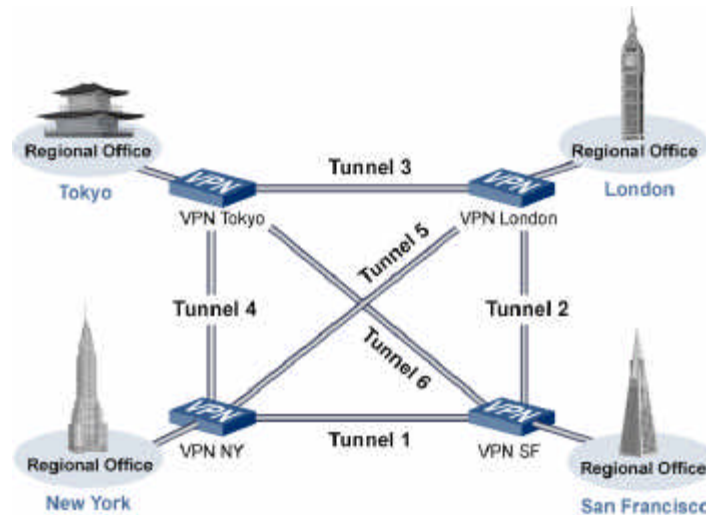
The initial set up of dynamic VPNs is similar to rule-based and route-based VPNs, with the defining of the VPN gateway. But instead of defining the network topology or manually building route tables, an administrator will enable dynamic routing on each of the gateways, binding the dynamic route protocol to the virtual “tunnel” interfaces. Juniper Networks’ Dynamic VPNs turn the IPSec Security Association into a routable interface, so tunnels are treated essentially as links. This is what enables the solution to run dynamic routing protocols through the tunnels, just as traditional frame relay links did, giving organizations the same ease of use they had with legacy private line and router solutions.

---

**NOTE** For more information on all of the redundancy features in a VPN, please refer to Juniper Networks’ white paper “Dynamic VPNs Achieving Scalable, Secure Site-to-Site Connectivity: How to cost-effectively replace WAN connections with a more reliable communication infrastructure.”

---

The below illustration shows how a dynamic VPN would be set up. The administrator would determine what they want the VPN topology to look like, define the VPN gateways and then set up the tunnels to enable a site to securely communicate with other sites. Notice that redundant tunnels (full mesh) can be easily set up to ensure the connectivity can persist in the event of a failure.



Once dynamic routing is enabled, Juniper Networks' dynamic VPNs leverage the protocol to automatically learn the network topology, saving organizations the time and resources required to define each and every machine on a network and iterating through the policy every time something is added or changed. New networks are accessible from any tunnel endpoint and are dynamically learned by the other endpoints via dynamic routing protocols. This also reduces the likelihood of mistakes due to human error.

More importantly, Juniper Networks' dynamic VPNs are able to automatically survive failures within the network to keep the connection available. If a tunnel is no longer a viable path for a route, or if it is removed from the routing table, a new route will be automatically learned. If there is a path open to get from point A to point B, Juniper Networks' dynamic VPNs will find it to ensure the communication persists. This makes the ongoing management and maintenance of the VPN possible without much human intervention at all, saving time and resources and providing a scalable solution that achieves the connectivity requirements of large, distributed organizations.

## Dynamic Route-Based Approach Summary

Juniper Networks' dynamic route-based VPNs separate the network topology from the VPN and enable dynamic routing to make the transport decision. This greatly simplifies the set-up and ongoing management of the VPN because the gateways are able to automatically learn the networks behind each peer. When changes are made, those changes are learned and incorporated into the network without any manual intervention. In addition, dynamic route-based VPNs enable the private network to maintain connectivity, enabling the solution to find an available route to ensure the connection can persist, even in the event of failures. As a result, Juniper Networks' dynamic route-based VPNs minimize the need for human resources and maximize the connectivity of the private network, making it a truly scalable site-to-site solution.

## Conclusion

A Dynamic Route-based VPN provides significant management benefits over a Rule-based VPN or a Route-based VPN. This paper discussed how the topology of the network is integral to the VPN policy in a Rule-based VPN approach and, therefore, creates a significant management burden. When the network changes or there is a link failure, the policy needs to be manually changed to reflect the new network. A Route-based VPN, on the otherhand, decouples the network route from the policy and, therefore, offers a slight improvement over the Rule-based VPN approach. However, because Route-based VPNs use static network routes as the foundation for the VPN links, they still require manual intervention, forcing the administrator to manually change the network route in the route table when a site changes or if an available route has not been defined. Only Juniper Networks' Dynamic Route-based VPN is designed to leverage dynamic routing to minimize the need for human intervention and easily scale to achieve the connectivity required by large site-to-site networks.

---

Copyright © 2004 Juniper Networks, Inc. All rights reserved.

Juniper Networks, the Juniper Networks logo, NetScreen, NetScreen Technologies, GigaScreen, and the NetScreen logo are registered trademarks of Juniper Networks, Inc. NetScreen-5GT, NetScreen-5XP, NetScreen-5XT, NetScreen-25, NetScreen-50, NetScreen-100, NetScreen-204, NetScreen-208, NetScreen-500, NetScreen-5200, NetScreen-5400, NetScreen-Global PRO, NetScreen-Global PRO Express, NetScreen-Remote Security Client, NetScreen-Remote VPN Client, NetScreen-IDP 10, NetScreen-IDP 100, NetScreen-IDP 500, GigaScreen ASIC, GigaScreen-II ASIC, and NetScreen ScreenOS are trademarks of Juniper Networks, Inc. All other trademarks and registered trademarks are the property of their respective companies.

Information in this document is subject to change without notice.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without receiving written permission from:

Juniper Networks, Inc.  
1194 N. Mathilda Ave. Sunnyvale, CA 95014 ATTN: General Counsel