

# MPLS VPN CONFIGURATION AND DESIGN GUIDE

**By Munther Louis Antoun**

Munther Antoun is a Consulting SE working with the Service Provider Engineering Group. He has an extensive background in networking services and has been with Cisco for 8 years. Munther is a valuable member of the Service Provider Solutions Team. To contact Munther Antoun - email: [antoun@cisco.com](mailto:antoun@cisco.com)

## SERVICE PROVIDER IP-BASED VPN ARCHITECTURE DOCUMENT SUMMARY

THIS DOCUMENT, PUT FORTH BY THE SERVICE PROVIDER SOLUTIONS TEAM, DISCUSSES THE CONFIGURATION AND DESIGN ASPECTS OF MULTI-PROTOCOL LABEL SWITCHING-VIRTUAL PRIVATE NETWORK (MPLS-VPN).

FIRST, AN INTRODUCTION TO VPN CONCEPTS IS PRESENTED, FOLLOWED BY AN OVERVIEW OF THE MPLS-VPN PARADIGM. SECURITY, QUALITY OF SERVICE, AND TRAFFIC ENGINEERING CONCEPTS ARE ADDRESSED NEXT, FOLLOWED BY ARCHITECTURAL DETAILS OF MPLS-VPN. MPLS-VPN CONFIGURATIONS ARE PURSUED NEXT, FOLLOWED BY NEED-TO-KNOWS AND POTENTIAL FUTURE FEATURES. ONE MOVES ON NEXT TO THE PROVISIONING ASPECTS OF MPLS-VPN WITH "EUREKA." SCALABILITY DISCUSSIONS ENSUE AFTER THAT, WHILE APPENDICES OF FURTHER VPN CONFIGURATIONS; STANDARDS PROGRESS; PLATFORM SUPPORT; AND AN OVERVIEW OF RRR CLOSE OUT THE DOCUMENT.

THIS DOCUMENT REPRESENTS WORK IN PROGRESS. THE AUTHOR HAS TRIED, AS MUCH AS POSSIBLE, TO ENGAGE THE READER IN ASPECTS OF THE MPLS-VPN TECHNOLOGY THAT WILL REMAIN UNCHANGED. THERE ARE INSTANCES, HOWEVER, WHERE REFERENCES ARE MADE TO VERSION-SPECIFIC LIMITATIONS OR FEATURES. THE READER SHOULD EXPECT CERTAIN FEATURES TO BE ADDED IN THE FUTURE, WHILE DEVELOPMENT ENGINEERING SIMULTANEOUSLY ADDRESSES LIMITATIONS.

READERS INTERESTED IN LEARNING DETAILS OF MPLS OR ATM MPLS DESIGN ISSUES ARE ENCOURAGED TO REFER TO JEREMY LAWRENCE'S "DESIGNING ATM MPLS NETWORKS" DESIGN GUIDE.

*Note: As the MPLS-VPN technology is deployed by more service providers, and as they engage our field, support, and development-engineering organizations in larger and more complex environments, we shall be able to provide more accurate performance statistics and characteristics. The reader is strongly encouraged to contact the author of this document to provide any design or scalability feedback regarding MPLS-VPN. The reader can also contact Product Marketing for pricing and other marketing information, and the "tag-vpn" e-mail alias for technical questions. Our Consulting, Systems, and Consulting Systems Engineers are encouraged to utilize the "tag-vpn" electronic-mail alias for features that Service Provider customers and prospects are looking for.*

## Acknowledgements

I am grateful to several individuals who were kind enough to review this document, making sure that it is as free of inaccuracies as possible.

I would like to recognize Yakov Rekhter for reviewing and suggesting changes to the architecture section. Ranjeet Sudan (MPLS-VPN Product Manager) and Robert Raszuk (NSA) were always available to handle my questions, as were several individuals from the “tag-vpn” e-mail alias, such as Dan Tappan and Eric Rosen. I am indebted to Ripin Checker for providing test information as well as patiently reducing my confusion about the functionality of MSSBU products.

My thanks also go out to David Phillips for reviewing the MPLS-PPP sections. J-F Deschênes helped me get started with some good write-ups and diagrams. Alain Fiocco was kind enough to point me to some valuable information that Riccardo Casiraghi and Simon Spraggs have gathered. A multitude of excellent presentations on anything dealing with MPLS and MPLS-VPN has been very helpful. Last, but not least, I am grateful to my manager, Joe Wojtal, who made sure I had time to spend on this document. I hope I did not leave anybody out. If I did, my apologies.

### Document Development Chronology

Revision	Date	Originator	Comments
0.1	3/8/1999	Munther Antoun	Original Guide
0.2	4/11/1999	J-F Deschênes	Edited original guide
0.3	4/16/1999	Munther Antoun	Edited JFD's draft version
0.4	7/5/1999	Munther Antoun	Edited final draft for version 1

## Table Of Contents

1	Virtual Private Networks	10
1.1	VPN Overview	10
1.2	VPN Architecture	11
1.2.1	The Overlay Model	11
1.2.1.1	Types of Shared Backbone VPNs/Overlay Networks	11
1.2.1.1.1	Circuit-switched VPN	11
1.2.1.1.2	Frame relay or ATM VPN	11
1.2.1.1.3	IP VPN	11
1.2.1.2	Disadvantages of the Overlay Model	11
1.2.2	The “Peer Model” VPNs	12
1.2.2.1	Who is Peering with Whom?	12
1.2.2.2	Advantages of the Peer Model	13
1.2.2.3	Difficulties in Providing the Peer Model	13
1.2.2.3.1	Routing information overload in the P routers	13
1.2.2.3.2	What Contiguous Address Space!	13
1.2.2.3.3	Private Addressing in the C Networks	13
1.2.2.3.4	Access Control	14
1.2.2.3.5	Encryption	14
1.3	MPLS-VPNs	14
1.3.1	MPLS-VPN Overview	14
1.3.2	MPLS VPN Requirements	14
1.3.3	MPLS-VPN Pre-requisites	15
1.3.4	MPLS-VPN - The True Peer Model	15
1.3.5	New Address Family	16
1.3.6	Thou Shalt Not Have to Carry 50,000 Routing Entries	16
1.3.7	Route Reflectors	16
1.3.8	Packet Forwarding - PEs Utilize BGP While Ps use LDP	17
1.3.9	Take Two Labels Before Delivery	17
1.3.10	Intranets and Extranets	17
1.3.11	Security	18
1.3.12	Quality of Service in MPLS-Enabled Networks	20
1.3.12.1	DiffServ	20
1.3.12.2	Design Approach For Implementing QoS	20
1.3.12.3	Cisco IOS“ QoS/CoS Toolkit	20
1.3.12.3.1	IP Precedence	21
1.3.12.3.2	Committed Access Rate (CAR)	21
1.3.12.3.3	Differential Discard and Scheduling Policies	21
1.3.12.3.4	Modified Deficit Round Robin	23
1.3.12.4	Proper QoS Tool Placement in the Network	23
1.3.12.4.1	QoS At the Edge	23
1.3.12.4.2	QoS In the Core	23
1.3.12.5	ATM-based MPLS and QoS/CoS	24
1.3.12.5.1	ATM MPLS-VPN CoS/QoS Mechanisms	24
1.3.13	MPLS Traffic Engineering	26
1.3.13.1.1	RRR Requirements	27
1.4	Detailed MPLS-VPN Functional Characteristics	27
1.4.1	Per-Site Forwarding Tables in the PEs	28
1.4.1.1	Internet Connectivity	28
1.4.1.2	My VPN Doesn’t Talk to Your VPN	28
1.4.1.3	Virtual Sites	29
1.4.2	Same VPN, Different Routes to the Same Address	29
1.4.3	MPLS-VPN Backbone	29
1.4.4	A Set of Sites Inter-connected via a MPLS-VPN Backbone	30
1.4.5	CE-PE Routing Exchange	30
1.4.6	Backdoor Connections	30
1.4.7	Per-site VRFs on PEs	31

1.4.7.1	Development of VRF Entries	31
1.4.7.2	Default	31
1.4.7.3	Traffic Isolation	32
1.4.8	PEs Re-distribute Customer Routes to One Another	32
1.4.8.1	VPN-IPv4 Address Family	32
1.4.8.2	Import & Export Route Policy	33
1.4.8.2.1	Target VPN Attribute	33
1.4.8.3	Route Re-distribution	34
1.4.8.4	Building VPNs with Extended Community Attributes	34
1.4.8.5	Packet Forwarding across the Backbone	35
1.4.9	PEs Learn Routes from CEs	36
1.4.9.1	PEs Redistribute VPN-IPv4 Routes into IPv4 VRFs	36
1.4.9.2	PE-CE Routing Protocol Options	37
1.4.10	CEs Learn Routes from PEs	38
1.4.11	ISP as a Stub VPN	38
1.4.11.1	Encoding VPN-IPv4 Address Prefixes in BGP	38
1.4.11.2	Filtering Based on Attributes	38
1.4.11.2.1	Site of Origin	38
1.4.11.2.2	VPN of Origin	39
1.4.11.2.3	Target VPN/Route Target	39
1.4.12	BGP Amongst PE Routers	39
1.4.12.1	Ordinary BGP Routes	40
1.4.12.2	Internet Filtering	40
1.4.12.3	Route Aggregation	40
1.4.13	Security	40
1.4.13.1	Cisco's Support of IPSec on CEs Today	40
1.4.13.2	IPSec Work in Progress	40
1.4.14	MPLS VPN Functional Summary	40
1.5	MPLS-VPN Configuration	41
1.5.1	Summary of MPLS-VPN Configuration Steps	41
1.5.2	MPLS-VPN Configuration Entities	41
1.5.2.1	VRF instances	41
1.5.2.1.1	IOS Configuration Command for a VRF Instance	41
1.5.2.1.2	VRF Configuration Sub-mode	41
1.5.2.2	Router Address Family	42
1.5.2.2.1	Backwards Compatibility	42
1.5.2.2.2	Address Family Components	42
1.5.2.2.3	Address Family Configuration	42
1.5.2.2.4	Address Family Usage	42
1.5.2.3	VPN-IPv4 NLRIs	43
1.5.2.4	Route Target (RT) Communities	43
1.5.2.4.1	CUG VPN	43
1.5.2.4.2	Hub-and-Spokes VPN	43
1.5.2.4.3	Controlled Access to Servers	43
1.5.3	MPLS-VPN Configuration, Next Steps:	44
1.5.4	Global-level versus sub-command VRF Commands	46
1.5.5	First Configuration Example	46
1.5.6	Second MPLS VPN Configuration Example	48
1.5.7	Third Configuration Example - Hub-and-Spokes	50
1.5.7.1	Configuration from CE: A-3620-mpls	50
1.5.7.2	Configuration from CE: B-3620-mpls	50
1.5.7.3	Configuration from: C-2611-mpls	51
1.5.7.4	Configuration from: D-1720-mpls	52
1.5.7.5	Configuration from: E-1720-mpls	53
1.5.7.6	Configuration from: H-7204-mpls	53
1.5.7.7	Configuration from: I-7204-mpls	54
1.5.7.8	Configuration from: J-7204-mpls	54
1.5.7.9	Configuration from: K-7204-mpls	56
1.5.7.10	Configuration from: L-7204-mpls	57
1.5.8	Fourth Configuration Example –Default Routing	58

1.5.9	PPP + MPLS-VPN Configurations (Cisco IOS 12.0(5)T)	58
1.5.9.1	Diagram of PPP + MPLS-VPN European Testing	58
1.5.9.2	Configuration and Monitoring of PPP + MPLS-VPN/European Testing	58
1.5.10	MPLS Traffic Engineering (TE) Configuration	73
1.5.10.1	New Command Syntax	73
1.5.10.2	MPLS TE Issues	74
1.5.10.3	MPLS TE Lab Configuration Scenarios	74
1.5.10.3.1	MPLS TE Lab Scenario One - Basic TE Environment	74
1.5.10.3.2	MPLS TE Lab Scenario Two - Basic Tunnel Configuration	75
1.5.10.3.3	MPLS TE Lab Scenario Three - Path Options	76
1.5.11	Performance and Management Characteristics	76
1.5.11.1	Scalability of MPLS-VPNs	76
1.5.11.2	MPLS Network Management	77
1.5.11.2.1	MPLS MIBs	77
1.5.11.2.2	Ping and RTR MIBs	77
1.5.12	MPLS-VPN Must-knows	77
1.6	MPLS-VPN (Uncommitted) Future Features	79
1.6.1	PPP/VPN - Today	79
1.6.2	PPP/VPN Integration – Multi-FIB VPNs	80
1.6.3	PPP MPLS-VPN Integration – Scaling PPP	80
1.6.4	PPP MPLS-VPN Without Tunnels	80
1.6.5	Proposed MPLS/VPN Multicast Support	81
1.6.6	MPLS/VPN Route-Map Support	81
1.6.7	Proposed DSL Interaction with MPLS-VPN	81
2	Cisco Service Management for MPLS-VPN (aka “Eureka”)	81
2.1	Platform Requirements	81
2.2	Eureka 1.0 Features	82
2.2.1	Service Provisioning	82
2.2.2	Provisioning Components	82
2.2.3	Eureka Administrative Console	83
2.2.4	Provisioning Steps	83
2.2.4.1	Defining Networks and Targets	83
2.2.4.2	Defining Provider and Customer Device Structure	83
2.2.4.3	Defining the Customer Edge Routers	83
2.2.4.4	Defining Customer VPNs	83
2.2.4.5	Downloading CE & PE Configurations	83
2.2.4.6	Other Steps	83
2.2.5	Service Requests	84
2.2.6	Generating configlets	84
2.2.7	Download	84
2.2.8	Auditing	84
2.2.9	Reports Generated with Eureka 1.0	85
2.2.9.1	Maximum Round Trip Time (RTT)	85
2.2.9.2	Percentage Connectivity of Devices	85
2.2.9.3	Delay Threshold Connectivity of Devices	85
2.2.9.4	Netflow Statistics and Accounting	85
2.2.9.4.1	Overview of the Netflow Collector	85
2.2.9.4.2	Netflow Reports within Eureka	86
2.2.10	Eureka 1.0 Status	86
3	Appendices - Standards; References; and Monitoring and Debugging Information	86
3.1	Appendix A – Cisco’s MPLS Efforts	86
3.1.1	MPLS Availability	86
3.1.2	To CR-LDP or not to CR-LDP	86
3.1.3	Is MPLS a Standard Yet?	86
3.1.3.1	Last Call for WG or IESG	86
3.1.3.2	MPLS Core Specifications	87
3.1.3.3	When is a Standard a Standard?	87
3.1.4	Cisco’s MPLS Efforts - Summary	87
3.2	Appendix B – References	88
3.3	Appendix C – MPLS-VPN Platforms	88

3.3.1	MPLS-VPN Functionality - Available Platforms	88
3.3.2	GSR MPLS-VPN Support	89
3.3.3	MPLS Support in MSSBU Platforms	89
3.3.3.1	General MSSBU MPLS Support	89
3.3.3.2	The VSI Interface	89
3.3.3.3	VSI Resource Partitioning	90
3.3.3.4	The BPX 8650	90
3.3.3.5	MGX 8850 with the Route Processor Module	90
3.3.3.5.1	MGX Today - Edge LSR Functionality without the LSC	91
3.3.3.5.2	MGX Futures - LSC Support	92
3.3.4	12.0T and 12.0S Code Paths	92
3.4	Appendix D – Architecture of RRR	92
3.4.1	Introduction	93
3.4.2	Traffic Engineering Case Study	93
3.4.3	RRR Requirements	95
3.4.3.1	MPLS	95
3.4.3.2	RSVP Extensions	95
3.4.3.3	OSPF and IS-IS Extensions	95
3.4.4	Traffic Trunks and other RRR Traffic Engineering Paradigms	95
3.5	Appendix E – Application Note: MSSBU's Demo Lab @ SP Base Camp Wk 2	96
3.5.1	Software Versions	96
3.5.1.1	LS1010	96
3.5.1.2	4700	96
3.5.1.3	2611	96
3.5.2	Configuration Examples	96
3.5.2.1	LS1010-A	96
3.5.2.2	LS1010-B	97
3.5.2.3	4700-A	97
3.5.2.4	4700-B	98
3.5.2.5	4700-C	99
3.5.2.6	2611-A	101
3.5.2.7	2611-B	101
3.5.2.8	2611-C	102
3.5.2.9	2611-D	103

## Table of Figures

Figure 1 - MPLS-VPN Architectural components	12
Figure 2 - VPN Peer Model	15
Figure 3 - VPN Forwarding Information Example	17
Figure 4 - Stack of Labels	17
Figure 5 Using MPLS to Build VPNs	19
Figure 6 - Example of Class-Based Weighted-Fair Queuing	21
Figure 7 - CAR Sets Service Classes at the Edge of the network (Edge LSR)	23
Figure 8 - ATM Forum PVC Mode	25
Figure 9 - Multi-VC Mode	25
Figure 10 - Multi-VC Mode, Application of Cisco IOS QoS @Egress/Core	25
Figure 11 - Single ABR VC-Mode	26
Figure 12 - Implementations of Single-VC Mode	26
Figure 13 - CE Backdoor Scenario	31
Figure 14 - MPLS Traffic Engineering Scenario 1, Basic TE	75
Figure 15 - MPLS Traffic Engineering Scenario 2, Basic Tunnel Configuration	75
Figure 16 - MPLS Traffic Engineering Scenario 3, Path Options	75
Figure 17 - PPP + MPLS/VPNs in Cisco IOS 12.0T	79
Figure 18 - Potential PPP/MPLS-VPN Integration	80
Figure 19 - Scalability	80
Figure 20 - Long-term Potential PPP/VPN Integration	80
Figure 21 - Proposed MPLS/Multicast Support	81
Figure 22 - Proposed MPLS/Multicast Support, the Next Steps	82
Figure 23 - Eureka 1.0 Functional Components	82
Figure 24 - The Eureka Service Model	82
Figure 25 - Administrative Console Graphical User Interface for Eureka	82
Figure 26 - Service Auditing	85
Figure 27 - "Formula" for BPX 8650 ATM LSR	89
Figure 28 - The MGX 8850 IP+ATM Switch	89
Figure 29 - VSI & End-to-End MPLS Signaling	90
Figure 30 - Typical RPM Deployment	90
Figure 31 - PVP/PVC Connection between a pair of RPM ELSRs	91
Figure 32 - PVP connection between an RPM Edge LSR and a BPX 8650 with an LSC	91
Figure 33 - RPM Functionality without LSC	91
Figure 34 - MGX with LSC Support	92
Figure 35 - PVP connection between an RPM Edge LSR and an RPM LSC	92
Figure 36 -The Traffic Engineering Problem	93
Figure 37 -Traffic Engineering Example Topology	93
Figure 38 - MSSBU's Demo Setup, SP Bootcamp for SE's, March 22-26, 1999	96

## Definitions

This section defines words, acronyms, and actions that may not be readily understood.

<b>AXSM</b>	ATM Switch Service Module. A serial-bus-based Service Module supported on the MGX 8850 beginning in Release 2, expected in CQ1, 2000. The AXSM card supports a variety of broadband ATM interfaces.
<b>MPLS</b>	Multi-Protocol Label Switching. The IETF equivalent of Tag Switching.
<b>C Network</b>	Customer or enterprise network, consisting of Customer routers, which are maintained and operated by the enterprise customer or by the Service Provider as part of a managed service.
<b>P Network</b>	Service Provider network, consisting of Provider routers, which are maintained and operated by the Service Provider.
<b>CE Router</b>	Customer Edge router - an edge router in the Customer network, defined as a C router which attaches directly to a P router, and is a routing peer of the P router.
<b>P Router</b>	Provider router (aka MPLS-VPN Backbone Router) - a router in the Provider network, defined as a P router which may attach directly to a PE router, and is a routing peer of other P routers. P Routers perform MPLS label switching.
<b>PE Router</b>	Provider Edge router - an edge router in the Provider network, defined as a P router which attaches directly to a C router, and is a routing peer of the C router. PE Routers translate IPv4 addresses into VPN-IPv4 12-byte quantities. Please see the appropriate definitions below.
<b>VPN-IPv4</b>	12-byte quantity. The first eight bytes are known as the Route Distinguisher (RD); the next four bytes are an IPv4 address.
<b>RD</b>	The Route Distinguishers (RD) are structured so that every service provider can administer its own “numbering space” (i.e., can make its own assignments of RD’s), without conflicting with the RD assignments made by any other service provider. The RD consists of a two-byte Type field, and a six-byte Value field. The interpretation of the Value field depends on the value of the Type field. At the present time, we define only two values of the type field: 0 and 1.
<b>Border router</b>	A router at the edge of a provider network which interfaces to another provider’s Border router using EBGp procedures. E.g., a PE router that interfaces via IBGP to its PE peers, as well as an EBGp peer to a public Internet router.
<b>VRF</b>	VPN Routing/Forwarding. It is the set of routing information that defines a customer VPN site that is attached to a single PE router. A VRF Instance consists of an IP routing table; a derived forwarding table; a set of interfaces that use the forwarding table; and a set of rules and routing protocols that determine what goes into the forwarding table (From “Approved_Draft 2 Final Tappan VPN”). There are three pieces to VRFs. The first is multiple routing protocol contexts. The second is multiple VRF routing tables. And the third is multiple VRF forwarding tables using FIB (CEF) forwarding tables. One can have only one VRF configured per (sub-)interface.
<b>VRF Routing Table</b>	Table which contains the routes which should be available to a particular set of sites. This is analogous to the standard IP routing table, which one may see with the “show ip route” Cisco IOS EXEC command, and it supports exactly the same set of redistribution mechanisms. MPLS-VPN code in Cisco IOS has routing information in CONFIG and EXEC modes with a VRF context. For example, one can issue a “show ip route vrf vrf_name.”
<b>VPN0</b>	A future feature that will allow the re-distribution of the public Internet BGP tables into MPLS-VPN tables to be exchanged amongst PEs, if so desired. Contrast that with the ability of the software (in the future) to refer to the global routing table if a route lookup fails inside a particular VRF.
<b>Global Routing Table</b>	The standard Cisco IOS IP routing table that traditional commands like “show ip route” utilize.

<b>VRF Forwarding Table</b>	Contains the routes that can be used from a particular set of sites. This uses the FIB forwarding technology. FIB must be enabled in order to support VPN.
<b>VSI</b>	Virtual Switch Interface. A protocol that allows for a common control interface to some of Cisco's ATM switches, for example, the MGX and BPX products. VSI is a protocol through which a master controls a slave. The Label Switch Controller is the master that, based on the MPLS information that it has, controls the operation of the slave ATM switch, which has no knowledge about MPLS. All the switch knows is that it understands VSI and how to respond to the requests from the master. VSI was invented by Cisco and implemented first with the LSC. It has recently been submitted to the Multi-Service Switching Forum for consideration as an open standard.
<b>Label Switching</b>	The IETF equivalent of Tag Switching, or the act of switching labels/tags.
<b>Label</b>	Header used by an LSR to forward packets. The header format depends upon network characteristics. In non-ATM networks, the label is a separate, 32-bit header, and QoS is applied using the ToS field in IP headers. In ATM networks, the label is the same length, but an unlimited number of labels can represent different levels of service. They are placed into the Virtual Channel Identifier/Virtual Path Identifier (VCI/VPI) cell header. In the core, LSRs read only the label, not the packet header. One key to the scalability of MPLS is that labels have only local significance between two devices that are communicating.
<b>LDP</b>	Label Distribution Protocol. The IETF equivalent of Tag Distribution Protocol (TDP)
<b>LSR</b>	Label Switch Router. The IETF equivalent of a Tag Switch Router (TSR). The core device that switches labeled packets according to pre-computed switching tables. It can be a router, or an ATM switch plus LSC.
<b>Edge LSR</b>	The IETF equivalent of a Tag Edge Router (TER). The edge device that performs initial packet processing and classification, and applies the first label. i.e., the role of an Edge LSR is to turn unlabeled packets into labeled ones. This device can be either a router, such as the Cisco 7500, or a Cisco IP+ATM switch that has a routing entity/LSC.
<b>LSC</b>	Label Switch Controller. IETF equivalent of Tag Switch Controller (TSC). An LSC is an MPLS router, with the unique characteristic that it also controls the operation of a separate ATM switch in such a way that the two of them together function as a single ATM Label Switch Router. From the outside, the combination of the LSC and ATM switch are viewed as a single high performance MPLS router. It's important to note that the LSC capability is an extension of the basic Label Switch router capability. LSC functionality is a superset of the functionality of an ATM Label Switch Router. This paradigm allows a Cisco BPX to be converted to also an MPLS LSR. The MGX will have that functionality, but with the introduction of the PXM 2 switch controller, expected out around June of 2000.
<b>LSP</b>	Label Switch Path. Path defined by all labels assigned between end points. An LSP can be dynamic or static. It is the IETF equivalent to TSP.
<b>LFIB</b>	Label Forwarding Information Base. IETF equivalent of Tag FIB (TFIB).
<b>Label</b>	The IETF equivalent of Tag.
<b>LVC</b>	Label VC. IETF equivalent of Tag VC (TVC).
<b>VPN</b>	Virtual Private Network

## 1 Virtual Private Networks

### 1.1 VPN Overview

A Virtual Private Network is defined as network whereby customer connectivity amongst multiple sites is deployed on a shared infrastructure with the same policies as a private network. Examples of Virtual Private Networks are the ones built using traditional Frame-Relay and ATM technologies. Service Providers have been very successful with these services and double-digit growth rates are expected to continue for a number of years.

An IP VPN is simply a VPN that provides IP connectivity on an intra- as well as inter-company basis. In other words, the VPN infrastructure is IP-aware.

Cisco has a number of strategies to address this emerging market for IP intra-networking as well as extra-networking<sup>1</sup>. The hub-and-spokes pattern common to existing VPNs is being replaced with any-to-any mesh patterns. Moreover, conventional VPNs are based on creating and maintaining a full mesh of tunnels or permanent virtual circuits among all sites belonging to a particular VPN, using IPsec, L2TP, L2F, GRE, Frame Relay or ATM. To provision and manage these overlay schemes is not supportable in a network that requires thousands or tens of thousands of VPNs, and hundreds, thousands, and tens of thousands of sites in those VPNs.

MPLS-based VPNs, which are created in Layer 3, are based on the peer model, and therefore substantially more scalable and easier to build and manage than conventional VPNs. In addition, value-added services, such as application and data hosting, network commerce, and telephony services, can easily be targeted and deployed to a particular MPLS VPN because the Service Provider backbone will recognize each MPLS VPN as a secure, connectionless IP network.

MPLS-based VPNs offer these benefits:

- MPLS VPNs provide a platform for rapid deployment of additional value-added IP services, including Intranets, Extranets, voice, multimedia, and network commerce.
- MPLS VPNs provide privacy and security equal to Layer-2 VPNs by constraining the distribution of a VPN's routes to only those routers that are members of that VPN<sup>2</sup>, and by using MPLS for forwarding.
- MPLS VPNs offer seamless integration with customer intranets.
- MPLS VPNs have increased scalability over current VPN implementations, with thousands of sites per VPN and hundreds of thousands of VPNs per service provider.
- MPLS VPNs provide IP Class of Service (CoS), with support for multiple classes of service within a VPN, as well as priorities amongst VPNs, as well as a flexible way of selecting a particular class of service (e.g., based on a particular application).
- MPLS VPNs offer easy management of VPN membership and easy provisioning of new VPNs for rapid deployment.
- MPLS VPNs provide scalable any-to-any connectivity for extended intranets and extranets that encompass multiple businesses.

Service Providers will utilize the functionality of MPLS-VPN to offer an IP service. However, the MPLS-VPN focus is not on providing VPNs over the public Internet<sup>3</sup>. Customer requirements for public Internet connectivity can be accomplished through the injection of external or default routes into CPE routers. Furthermore, a Service Provider can optionally provision data encryption services for their customers, through the overlaying of IPsec tunnels on top of MPLS-VPN.

<sup>2</sup>As of Cisco IOS 11.0(5)T, the MPLS-VPN PE routers that exchange VPN-IPv4 routes via IBGP, receive all routes for all VPNs. They then accept into the appropriate VPN routing tables only the routes that pertain to the respective VPNs. Development Engineering currently has experimental code that does this more efficiently by performing inbound filtering before importing all the routes into the global BGP table. The reader should consult with Product Marketing or the "tag-vpn" e-mail alias as to availability of that feature in a supported release. There is also work for Outbound Route Filtering (ORF), which is a dynamic way to exchange outbound filters between BGP speakers. The ORF draft, which is not published yet, considers one ORF-type today (NLRI) but it will be extended in order to use the route-target (ExtComm) attributes, which will make an IBGP PE router send to an IBGP peer only the routes that it is interested in (i.e., routes for VPNs it has been configured with.)

<sup>1</sup>An extranet is a network connecting IP systems within a company as well as at least one other independent entity. The public Internet can substitute for the "other independent entity."

<sup>3</sup>Although a Service Provider that offers MPLS-VPN services can also utilize that infrastructure to offer global Internet connectivity.

## 1.2 VPN Architecture

In order to properly understand the scalability improvements afforded by MPLS-based VPN's let us first examine the various VPN models available today. We first examine the limitations of the overlay model and then approach the significant design advantages resulting from a peer-model implementation.

### 1.2.1 The Overlay Model

A Service Provider provides an enterprise customer with the technology to inter-connect many sites by utilizing a private WAN IP network. Each site requiring connectivity will receive a router that needs to be peered through an appropriate IGP, to at least the head-end router. In this case, the SP has supplied the enterprise customer with a private network backbone.

If the enterprise actually owns all the transmission media and switches which constitute the backbone, then we have a truly private network. More commonly though, the transmission media, and at least some of the backbone switches, are owned by a Service Provider (SP), and are actually shared amongst multiple enterprise networks. Then each enterprise network is not really a private network, but a Virtual Private Network.

#### 1.2.1.1 Types of Shared Backbone VPNs/Overlay Networks

##### 1.2.1.1.1 Circuit-switched VPN

Here, the routers at the various sites of an enterprise can be inter-connected either by leased lines or by dial-up lines. In either case, the backbone is most likely a shared telephone network.

##### 1.2.1.1.2 Frame relay or ATM VPN

In this environment, the routers at the various sites of an enterprise can be inter-connected by virtual circuits. Like real circuits, virtual circuits provide point to point connections.

##### 1.2.1.1.3 IP VPN

Point-to-point connections amongst the enterprise routers can be provided by means of some sort of IP tunneling, such as IPsec, or GRE.

In private or virtual private networks like these, the design and operation of the backbone topology is the responsibility of the enterprise or of the Service Provider if managed services are involved. Routers

located at the enterprise sites are adjacent to one another via the point-to-point connections, and routing information is exchanged directly via the point-to-point connections.

To the Service Provider's backbone network, this routing information is merely data, and it is handled transparently. Similarly, the enterprise routers have no knowledge or control over the routing functions of the backbone. That is the domain of the Service Provider.

We say that the enterprise IP network is overlaid on top of the Service Provider backbone. The enterprise network can be called the higher layer network, the backbone network the lower layer network. Both networks exist, but independently of each other. This way of building a higher layer network on top of a lower layer network is called the overlay model.

#### 1.2.1.2 Disadvantages of the Overlay Model

For the enterprise network to obtain optimal routing through the backbone, it is necessary for the enterprise network to be fully meshed. That is, each site in the enterprise network must have a router that is an adjacency of some enterprise router in all other sites.

If the enterprise network is not fully meshed, then there will be cases in which traffic goes from one enterprise router, through the SP backbone, to the enterprise's backbone (head-end) router, back into the SP backbone, and finally onto the destination enterprise router (destination remote site). Since remote site routers are attached to the common (SP) backbone, having the data leave the backbone, traverse a second router, and re-enter the backbone is inefficient.

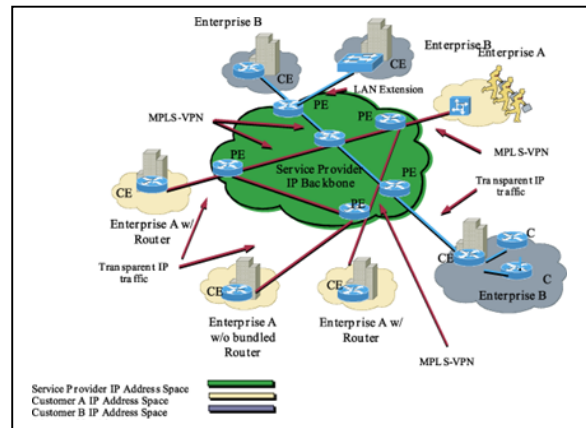
If the enterprise network is fully meshed, this situation is avoided, but other problems arise. The enterprise has to pay for, and the provider has to provision, a number of virtual circuits, which grows as the square

of the number of sites<sup>4</sup>. Apart from the cost, the IP routing algorithms scale poorly<sup>5</sup> as the number of direct connections amongst routers grows, which causes additional problems.

In the overlay model the enterprise customer needs to come to an agreement with the Service Provider as to who is responsible for designing and operating the “backbone” that inter-connects the customer sites. Neither alternative of the Service Provider versus the customer designing and operating the backbone is attractive. If it is the customer’s responsibility, then the staff for that customer needs to have IP routing expertise, and most customers do not have the luxury of affording such knowledgeable staff. So, this doesn’t scale to a large number of customers. The second alternative, which calls for the Service Provider to design and support each and every one of its VPN customers, does not scale either. That endeavor is fairly expensive, and doesn’t scale to a large number of customers. So neither alternative scales to a large number of VPNs.

## 1.2.2 The “Peer Model” VPNs

But why does the enterprise have to design and operate a backbone network at all, even a virtual backbone network, and engage its staff in properly designing and supporting one or more IGP’s? The SP, which is already providing the backbone infrastructure, can certainly design and operate the backbone. Then each site won’t require peering with a head-end router, and, in the case of partial or full meshing, more neighbor relations. The peer model VPN will merely require that a router attach to one of the SP’s routers. From the point of view of a particular site administrator, every IP address that isn’t located at one’s own site is reachable via the SP’s backbone network. How the SP’s backbone decides to route the traffic is the SP’s concern.



### 1.2.2.1 Who is Peering with Whom?

In the peer model VPN, two C routers are routing peers of each other only if they are at the same site. That is, Customer router C1 does not have a peering (neighbor) relationship with router C2, belonging to the same customer, in a different site. Rather, each site has at least one CE router, which is peered to at least one PE router.

In the peer model, the SP backbone routers/switches will themselves be IP networks. Contrast that to a public X.25, Frame Relay, or ATM network, where the provider’s backbone is a collection of Data Link Layer devices that communicate amongst themselves with a common, usually proprietary, protocol. Since CE routers do not exchange routing information with one another, there is never any need for data to travel through transit CE routers<sup>6</sup>. Data goes from an ingress CE router, through a sequence of one or more P (backbone) routers, to an egress CE router. Hence we get optimal routing.

Since CE routers do not directly<sup>7</sup> exchange routing information with other CEs, there is no virtual backbone for the enterprise to manage. It is of course possible to use an IP backbone as if it were a Frame Relay network, setting up “virtual circuits” of a sort amongst CE routers. This is commonly

Figure 1 - MPLS-VPN Architectural components

<sup>4</sup> Actually, the number of connections is  $[(N-1) * N] / 2$ , where N is the number of sites. So, four fully-meshed sites require  $[4*3]/2 = 6$  connections. Five sites stipulate 10 links, and so on.

<sup>5</sup> One cannot envisage an IGP like EIGRP, OSPF, or ISIS with several hundred or thousand peers. Amongst the many problems with this design is the CPUs of the routers will be overwhelmed, while the routing overhead will occupy a good portion of the WAN bandwidth.

<sup>6</sup> Versus for example CE routers in a non-fully-meshed Frame Relay environment.

<sup>7</sup> CE routers do exchange routing information with one another, but indirectly, via PEs.

done by means of some form of IP tunneling. This is still the overlay model though, and has all the problems of that model. The peer model is very different.

### 1.2.2.2 Advantages of the Peer Model

The peer model has many advantages:

- The amount of work the Service Provider needs to do in order to provision a new enterprise customer site is  $O(1)$  – independent of the number of sites in the VPN. In contrast, amount of work is  $O(n)$  in the overlay model, where  $n$  is the number of sites in the VPN.
- The peer model allows optimal routing of customer data through the Service Provider's backbone, as there will not be the need for transit CEs.
- The enterprise customer does not have a virtual backbone to manage. The customer just plugs in a CE router at each site.
- The peer model makes it simple for a service provider to provide server hosts that can be accessed from multiple VPNs. With the overlay model, this requires a virtual circuit from each VPN.

Thus the peer model provides advantages to producer and consumer - less work for the SP, and more value for the enterprise customer.

### 1.2.2.3 Difficulties in Providing the Peer Model

While the peer model has many advantages over the overlay model, there are a number of problems that must be solved before the peer model can be used.

#### 1.2.2.3.1 Routing information overload in the P routers

The peer model requires that routing information from the C network flow into the P network. One of the main problems in large IP backbones is the amount of resources (memory, processing, bandwidth) needed to store the routing information. If one takes an IP backbone and then adds routing information from a whole set of enterprise networks, the P routers will never be able to handle it.

So, to make the peer model successful, the amount of routing information which the backbone routers must maintain, has to scale well as the number of VPNs supported by the backbone grows.

#### 1.2.2.3.2 What Contiguous Address Space!

Topologically, Internet Service Providers (ISPs) generally try to assign addresses in a meaningful way. That is, the address a system has should be related to where it attaches to the ISP's network. This sort of addressing scheme allows routing information to be aggregated, reducing the routing load on the P routers<sup>8</sup>.

However, many enterprise networks have addressing schemes that will not necessarily map well to the backbone topology of any SP. Addresses in the enterprise network will have been assigned to the various sites without regard to where in the SP's network the site will eventually be attached.

This reduces the opportunities for route aggregation, with more enterprise routing information passed into the P network.

Expecting the enterprise customer to re-address all its IP hosts is unrealistic, due to administrative burdens and hence the costs of such an endeavor.

#### 1.2.2.3.3 Private Addressing in the C Networks

Another problem is that many enterprise networks use non-unique addresses. That is, the addresses are unique within the particular enterprise, but not amongst enterprises. If a single IP backbone is shared as the backbone for two different enterprise networks, and those enterprise networks had non-unique addresses, the P routers will have no way of ensuring that packets get to their intended destinations.

#### 1.2.2.3.4 Access Control

If an enterprise buys IP backbone service from an SP, it wants some assurance that packets which enter their enterprise network come from that enterprise network, and that packets which originate in the enterprise network do not leave the enterprise network "by accident." If enterprise network routing

<sup>8</sup> In fact, this IP route aggregation, referred to as Classless Inter-Domain Routing, or CIDR, has allowed Service Providers to slow down the growth in the size of the Internet routing tables. Please refer to the appropriate CIDR RFCs for further information.

information is passed into the P network, how can this sort of inter-enterprise communication be controlled?

Of course, two enterprises may wish to communicate directly, or over the Internet. But they want such communication to occur through firewalls. However, they do not want intra-enterprise communication to occur through firewalls. Yet they may want to use the same ISP backbone for all these purposes.

#### 1.2.2.3.5 Encryption

To ensure privacy, one should set up point-to-point encrypted tunnels between every pair of CE routers (this is the IPsec model). This particular solution lends itself nicely to the overlay model, since the overlay model already uses a point-to-point tunnel between each pair of CE routers<sup>9</sup> that are “routing adjacencies”. It lends itself less nicely to the peer model, since in the peer model, a given CE router has no way of knowing the identity of the next CE router for a given packet.

### 1.3 MPLS-VPNs

#### 1.3.1 MPLS-VPN Overview

MPLS-VPN is a “true peer VPN” model that performs traffic separation at Layer 3, through the implementation of separate IP VPN forwarding tables.

MPLS-VPN enforces traffic separation amongst customers by assigning a unique VRF to each customer’s VPN.

This delivers the same level of privacy as ATM or Frame Relay, because users in a specific VPN cannot see traffic outside their VPN. The same level of privacy is provided because of the following factors:

- (1) forwarding within the Service Provider backbone is based on labels,
- (2) LSPs within the Service Provider infrastructure begin and terminate at the PE routers,
- (3) it is the incoming interface on a PE router’s interface, that determines which forwarding table to use when handling a packet, and
- (4) each incoming interface on a PE router is associated (at the provisioning time) with a

<sup>9</sup> That is, the IP address of the other end of the point-to-point tunnel is reachable from the source.

particular VPN. Therefore, a packet can enter a VPN only through an interface on the PE that is associated (via provisioning) with that VPN.

Traffic separation occurs without tunneling or encryption, because it is built directly into the network itself<sup>10</sup>.

Briefly, MPLS-VPN has the following characteristics:

- Multiprotocol BGP extensions are used to encode customer IPv4 address prefixes into unique VPN-IPv4 NLRIs.
- Extended BGP community attributes are used to control the distribution of customer routes.
- Associated with each customer route is an MPLS label. The PE router that originates the route assigns this. The label is then used to direct data packets to the correct egress CE router.
- MPLS forwarding is used across the provider backbone based on either dynamic IP paths, or Traffic Engineered paths.
- When a data packet is forwarded across the backbone, two labels are used. The top label directs the packet to the appropriate egress PE router. The second label indicates how that egress PE should forward the packet.
- Cisco MPLS CoS/QoS mechanisms provide service differentiation amongst customer data packets.
- Standard IP forwarding is used between the PE and CE routers. The PE associates each CE with a per-site forwarding table that contains only the set of routes available to that CE router.

#### 1.3.2 MPLS VPN Requirements

There are four major technologies that provide the ability to actuate MPLS-VPN. The first is Multi-Protocol BGP. The second is route filtering based on the “route target” extended BGP community attribute. We use MPLS forwarding to carry the packets across the backbone. Finally, Provider Edge

<sup>10</sup> Please refer to the Security sub-section at the end of the detailed section on MPLS-VPN.

routers utilize multiple routing and forwarding instances.

MPLS-VPN utilizes BGP amongst PE routers to facilitate Customer routes. This is facilitated through extensions to BGP to carry addresses other than IPv4<sup>11</sup>. In particular, we've defined a new address family of the VPN-IPv4 address. It consists of a 96-bit address, which has a 64-bit prefix that we call the Route Distinguisher, that makes the address unique in the backbone. The MPLS Label is carried as part of a BGP routing update. The routing update also carries the addressing/reachability information. So long as the 96-bit entity is unique across the MPLS-VPN network, proper connectivity is transacted even if different enterprise customers used non-unique IP addresses.

### 1.3.3 MPLS-VPN Prerequisites

On the appropriate router platforms, one has to have the PLUS image set. It is also a requirement to have CEF or FIB<sup>12</sup> switching on the PEs.

On the P side, MPLS has to be configured.

There are no requirements on the CE router, except IP static or dynamic forwarding. If so desired, RIP II or EBGp is required if either one of these protocols is spoken in common with the Service Provider equipment.

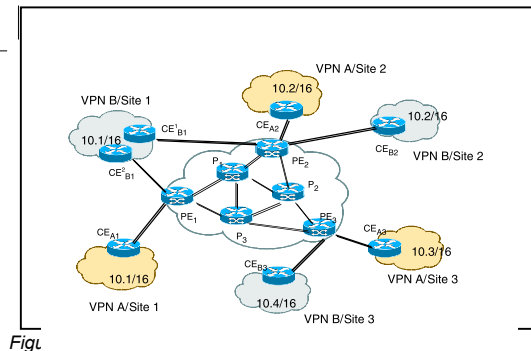


Fig.

<sup>11</sup> RFC2283: "Multi-protocol Extensions for BGP-4," makes it possible for BGP to carry routing information other than just IPv4 – multiple network layer protocols, one of which is MPLS. The extensions are backward compatible – a router that supports the extensions can inter-operate with a router that doesn't support the extensions. It will utilize "classical BGP-4."

<sup>12</sup> Cisco Express Forwarding, also referred to internally as Forwarding Information Base.

### 1.3.4 MPLS-VPN - The True Peer Model

In this VPN paradigm, MPLS is used for forwarding packets over the backbone, and BGP is used for distributing routes over the backbone. The primary goal of this method is to support the outsourcing of IP backbone services for enterprise networks. It does so in a manner that is simple for the enterprise, while still scalable and flexible for the Service Provider, and while allowing the Service Provider to add value. These techniques can also be used to provide a VPN which itself provides IP service to customers.

The CE router is a routing peer of the PE(s) to which it is attached<sup>13</sup>, but is not a routing peer of CE routers at other sites. Routers at different sites do not directly exchange routing information with one another; in fact, they do not even need to know of other CEs at all (except in the case where this is necessary for security purposes). As a consequence, very large VPNs (i.e., VPNs with a very large number of sites) are easily supported, while the routing configuration for each individual site is greatly simplified.

The True Peer Model maintains proper administrative boundaries between the C network and the P network. Solely the SP should administer the PE and P routers, and the SP's customers should not have any management access to it. Solely the customer should administer the CE devices (unless the customer has contracted the management services out to the SP).

In the True Peer Model, each site in a particular C network can interface to the Service Provider backbone via RIP II, static, or EBGp routing. When configuring an EBGp peering relationship between the CE and PE, the C network is modeled as an Autonomous System; the CE router(s) at a site use External BGP to exchange routing information with the PE router(s) at that site. RIP II or static routing are current alternatives to EBGp. The C network's interior routing protocol (i.e., its IGP) runs

<sup>13</sup> Statically, via RIP II, or EBGp. OSPF support will occur in the future.

independently at each site, and does not run in the P network. In other words, the True Peer paradigm models each VPN as an internet, with the backbone, the P network(s), connecting the sites together.

### 1.3.5 New Address Family

IPv4 addresses from a particular C network are mapped, by the PE<sup>14</sup> routers, into corresponding addresses in a new address family, the VPN-IPv4 address family<sup>15</sup>. A VPN-IPv4 address is a twelve-octet quantity. The first eight bytes are known as the Route Distinguisher (RD); the original IPv4 address occupies the next four bytes. If two C networks attach to the same P network, and a given IP address is used in both C networks, the PE routers which attach to the C networks will translate the IPv4 address into two different VPN-IPv4 addresses (by using a different RD), depending on which C network the address belongs to. Thus even when two C networks use the same IPv4 address, the corresponding VPN-IPv4 addresses will be different. Within the P network, routes to addresses that are within C networks are maintained as routes to VPN-IPv4 addresses<sup>16</sup>. Hence the fact that there is overlap between the address spaces of the two C networks does not cause any ambiguity in the P network. As long as a given end system has an address which is unique within the scope of the VPNs that it belongs to, the end system itself does not need to know anything about VPNs.

### 1.3.6 Thou Shalt Not Have to Carry 50,000 Routing Entries

The Internet needs to be a default-free zone for the Service Providers that are carrying the IP prefixes in full. Hence when a Service Provider needs to carry those routes via BGP4 across their backbone, all their IBGP peers need to have full IP routes. This causes scaling problems as the number of routes gets very large. However, hierarchical label switching provides a forwarding mechanism which allows one to maintain exterior routes only at border routers. Although BGP<sup>17</sup> is used to distribute VPN routing information, one does not require that the

interior routers of the backbone receive routes for VPN-IPv4 addresses.

When providing VPNs in this manner, the border routers are the PE routers. In the public Internet, all routes known to any border router must be known to all, or else complete end-to-end connectivity may not be possible. However, when providing multiple VPNs over a shared backbone, it is neither necessary nor desirable to provide complete end-to-end connectivity. End-to-end connectivity should be provided only amongst systems which are in the same VPN. VPN-IPv4 routing information for a particular C network is exchanged, using BGP, only by the PE routers that attach to that C network. PE routers that do not attach to a particular C network will not receive the routing information for that network. Hence the amount of routing information stored in a PE router is not proportional to the total number of VPNs supported by the P network, but only to the number of VPNs to which that P router is directly attached.

### 1.3.7 Route Reflectors

If a particular C network is attached to a large number of PE routers, the need to have each one distribute routing information to all the others can cause a scalability problem. However, this problem can be addressed by means of well known techniques, such as the use of BGP Route Reflectors. That is, rather than having a PE distribute the routes directly to another PE, the two PEs can be clients of a common route reflector. A given route reflector need not handle routes from all VPNs; the set of VPNs using a particular backbone can be partitioned, and each set of VPNs can be assigned to a different Route Reflector. In no case is there ever any one system that needs to know all the routes. This fact makes it possible to scale the system virtually without limit.

Before a PE router distributes routing information (about other sites in the C network) to a CE router, it translates the VPN-IPv4 addresses into IPv4 addresses, by stripping off the first eight bytes. Thus the CE routers see only ordinary IPv4 addresses; only in the P network is the longer

<sup>14</sup> This mapping or translation, from IPv4 to VPN-IPv4 is referred to as MPLS-VPN edge function.

<sup>15</sup> RFC2547, Section 4.1

<sup>16</sup> That is, a P router (which is actually an MPLS LSR) has an VPN-IPv4 route that the appropriate PE – the one directly attached to the CE router that is originating the C route – will translate to/from IPv4 and VPN-IPv4.

<sup>17</sup> That is, BGP with Multi-protocol extension support, as discussed elsewhere in this document.

addressing form used. The C routers do not need to support the VPN-IPv4 address family.

### 1.3.8 Packet Forwarding - PEs Utilize BGP While Ps use LDP

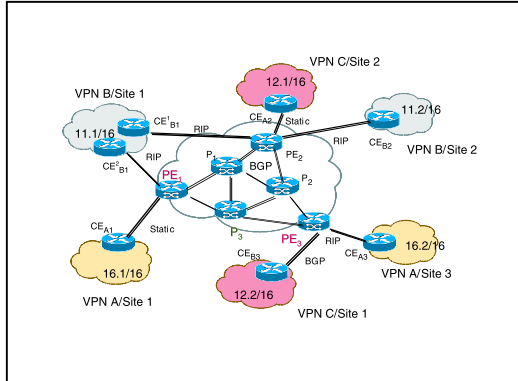


Figure 3 - VPN Forwarding Information Example

A customer's IP packet arrives at that customer's CE, which forwards the packet to its PE using conventional IP packet delivery means. Once the packet arrives at the PE, verification is made of the appropriate input interface; the packet's VPN is identified; and the VPN-specific FIB is located. The PE's FIB lookup provides the outgoing interface and two labels - The first label is to get across the P backbone to the egress PE router, while the second label controls handling of the packet by the egress PE router. From that egress PE, the packet is delivered to the correct CE destination.

In the MPLS-VPN connectivity paradigm, an ingress PE (PE1 in figure 3) router must maintain a separate forwarding table for each C network to which it is attached (customers CEA1 and CE2B1 in figure 3). This forwarding table is populated with routing information that pertains only to the C network. This information will have been gathered, via IBGP, from other PE nodes that attach to the same C network (PE3 for VPN A in figure 3). The PE routers for a particular VPN collect routing information from their respective CE peers (statically or dynamically), and re-distribute that into IBGP, to their PE peers for that VPN.

When a packet arrives from a particular directly attached C network onto the appropriate PE router interface, its destination address is looked up in that PE's corresponding forwarding table, to determine its egress PE router.

### 1.3.9 Take Two Labels Before Delivery

As is indicated in figure 4, an ingress PE receives "normal" IP Packets from its CE router, at which point the PE does an "IP Longest Match" from VPN\_B FIB, finds iBGP next hop PE2, and imposes a stack of labels: EXTERIOR label L2 + INTERIOR label L8.

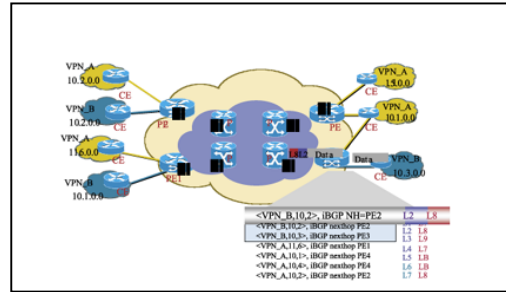


Figure 4 - Stack of Labels

All Subsequent P routers do switch the packet, solely on the Interior Label. The Egress PE router, removes the Interior Label Egress, if the penultimate<sup>18</sup> hop is an ATM-LSR. If the penultimate hop is a router-based LSR, then the interior label is removed by that LSR (the penultimate hop). The Egress PE routers then uses the Exterior Label to select which VPN/CE to forward the packet to. The Exterior Label is removed and the packet routed to the connected CE router. The interior label is removed by the egress PE only if penultimate hop is an ATM-LSR. If the penultimate hop is a router-based LSR, then the interior label is removed by that LSR (by the penultimate hop).

The route a packet must traverse between its ingress and egress PE routers will usually include one or more intermediate P routers (e.g., P3 in figure 3 for traffic between PE1 and PE3). The intermediate P routers do not maintain routing information for the VPNs, so they cannot forward the packet by looking up its IP destination address. Proper forwarding through the P network is achieved by means of label switching. Once

<sup>18</sup> Please refer to "draft-ietf-mpls-arch-05.txt" for further information on the penultimate hop concept.

the egress PE router for a given packet is determined, label switching is used to route the packet to the chosen egress PE router. The ingress PE router wraps the packet in a label switching header, where the label corresponds to a route (through the P network) to the egress PE router. Intermediate P routers forward the packet based on the label, not based on the IP destination address. Therefore the intermediate P routers do not need to know anything about C network routing. Nor do they need to know anything at all about VPN-IPv4 addresses. In fact, the P routers can simultaneously support MPLS-VPN as well as non MPLS-VPN Edge LSRs.

As stated earlier, the ingress PE router applies two labels to the packet. When PE1 sends, via BGP, a VPN-IPv4 route to PE3, it also specifies a label for the route. If this route belongs to a particular C network, PE3 enters this route into the forwarding table it uses for packets from that C network. When PE3 receives a packet from the CEA3 router in the network, it looks up the packet's destination address in this forwarding table. As a result, it determines the packet's BGP next hop (i.e., PE1), and the label assigned by that next hop. This label is pushed onto the packet's label stack. Then PE3 looks up, in its "regular" forwarding table (i.e., in the forwarding table containing routes through the P network), the address of PE1. The P router which is PE3's next hop to PE1 (i.e., P3 in figure 3) will have used LDP to bind a label to the route to PE1. This label is then pushed on the packet's label stack, and the packet sent to P3.

The topmost label, used for routing the packet through the P network, corresponds to a route to the egress PE router. The bottom label is used by the egress PE router to determine the particular output port (or sub-interface) on which it should transmit the packet. Thus the egress PE router avoids the need to look up the packet's destination address at all.

The MPLS-VPN True Peer connectivity model allows a P network to support any number of VPNs while not stipulating a large amount of routing information that needs to be stored in any one P router. It prevents data from flowing amongst VPNs, since it maintains separate forwarding information for each VPN. Furthermore, it does not assume that VPNs use addresses that are unique. Thus it avoids the problems of the overlay model, while also avoiding the problems of the Virtual Peer model.

In the True Peer Model, each enterprise network becomes an Internet, with the P network taking the role of backbone SP.

### 1.3.10 Intranets and Extranets

The procedures described above allow an SP to provide extranets, as well as intranets. An intranet is simply a collection of one customer's set of sites that are interconnected via one particular technology – in this case, MPLS-VPN. When customer C1 wishes to communicate with customer C2 via this MPLS-VPN technology, one has to construct an extranet.

To provide an intranet, the PE routers ensure that the forwarding table for the C network contains only routes learned from other sites of the C network. To provide an extranet, the PE routers allow the C network's forwarding table to contain selected routes from other C networks (or from the P network itself).

### 1.3.11 Security

So far, we have shown that MPLS-VPN functionality provides a level of security which is equivalent to that provided by overlay VCs based on Frame Relay or ATM networks.

Security in MPLS-enabled VPN networks is delivered through a combination of BGP and IP address resolution.

BGP is a routing information distribution protocol that defines who can talk to whom using multi-protocol extensions and community attributes. VPN membership depends upon logical ports entering the VPN, where BGP assigns a unique RD. RDs are unknown to end users, making it impossible to enter the network on another access port and spoof a flow. Only pre-assigned ports are allowed to participate in the VPN. In an MPLS-enabled VPN, BGP distributes forwarding information base (FIB) tables about VPNs to only members of the same VPN, providing native security via logical VPN traffic separation. Furthermore, IBGP PE routing peers can perform TCP segment protection using the

MD5 Signature Option<sup>19</sup>, when establishing IBGP peering relationships, further reducing the likelihood of introducing spoofed TCP segments into the IBGP connection stream, amongst PE routers.

The provider, not the customer, associates a specific VPN with each interface when provisioning the VPN<sup>20</sup>. Users can only participate in an intranet or extranet if they reside on the correct physical or logical port and have the proper RD. This setup makes a Cisco MPLS-enabled VPN virtually impossible to enter. As is the case with Frame Relay and other VPN technologies, mis-configurations by the Service Provider may increase the chances of data spoofing.

Within the core, a standard Interior Gateway Protocol (IGP) such as OSPF or IS-IS distributes routing information. Provider edge LSRs set up paths amongst one another using LDP to communicate label-binding information. Label binding information for external (customer) routes is distributed amongst PE routers using BGP multi-protocol extensions instead of LDP, because they easily attach to VPN-IP information already being distributed. The BGP community attribute constrains the scope of reachability information. BGP maps FIB tables to provider edge LSRs belonging to only a particular VPN, instead of updating all edge LSRs in the provider network.

### IP Address Resolution

MPLS-enabled IP VPN networks are easier to integrate with IP-based customer networks. Subscribers can seamlessly inter-connect with a provider service without changing their intranet applications, because MPLS-enabled networks have built-in application-awareness. Customers can even transparently use their existing IP address space without NAT because each VPN has a unique identifier.

MPLS VPNs remain unaware of one another. Traffic is separated amongst VPNs using a logically distinct forwarding table and RD for each VPN. Based on the incoming interface, the PE selects a specific forwarding table, which lists only valid destinations in the VPN, thanks to BGP. To create

extranets, a provider explicitly configures reachability amongst VPNs.

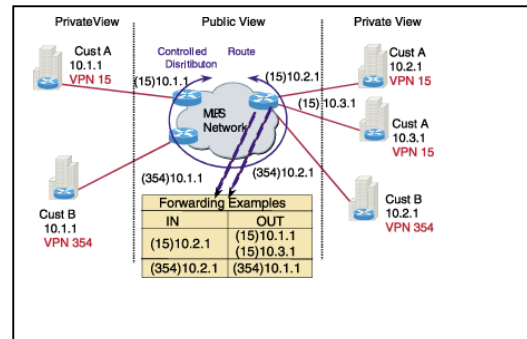


Figure 5 Using MPLS to Build VPNs

In Figure 5, those who are in VPN 15 never learn about the existence of VPN 354. As one can see in the forwarding table for the indicated router, it only contains address entries for members of the same VPN. It rejects requests for addresses not listed in its forwarding table. By implementing a logically separate forwarding table for each VPN, each VPN itself becomes a private, connectionless network built on a shared infrastructure, and we have attained an IP VPN-aware network.

IP limits the size of an address to 32 bits in the packet header. The VPN IP address adds 64 bits in front of the header, creating an “extended” address in routing tables that classical IP cannot forward. MPLS solves this problem by forwarding traffic based on labels, so one can use MPLS to bind VPN IP routes to label-switched paths (LSPs). LSRs need to be concerned with reading labels and not packet headers. We have already discussed how the edge LSR (i.e., PE)

- identifies the appropriate VPN for a packet it needs to deliver on behalf of its customer
- indexes it to the forwarding table for that VPN
- obtains the corresponding label and
- applies the label to the packet.

From there on, MPLS manages forwarding through the LSR core. Since labels only exist for valid destinations, this is how MPLS delivers both security and scalability.

<sup>19</sup> RFC2385, "Protection of BGP Sessions via the TCP MD5 Signature Option."

<sup>20</sup> See the sub-section titled "MPLS-VPN Overview" for details on how MPLS-VPNs facilitate data privacy.

When a VC is provided using the overlay model, the egress interface for any particular data packet is a function solely of the packet's ingress interface; the IP destination address of the packet does not determine its path in the backbone network<sup>21</sup>. Thus unauthorized communication into or out of a VPN is prevented.

In MPLS-VPNs, a packet received by the backbone is first associated with a particular VPN, by stipulating that all packets received on a certain interface (or sub-interface) belong to a certain VPN. Then its IP address is looked up in the forwarding table associated with that VPN. The routes in that forwarding table are specific to the VPN of the received packet. So the ingress interface determines a set of possible egress interfaces, and the packet's IP destination address is used to choose from among that set. This prevents unauthorized communication into and out of a VPN.

### 1.3.12 Quality of Service in MPLS-Enabled Networks

Quality of Service (QoS) and Class of Service (CoS) enable the Service Provider to offer differentiated IP-based service levels and tiered pricing. QoS refers to the overall service levels that a network can deliver. CoS refers to a specific level category in which a user or other model is classified, such as Gold, Silver, and Best-Effort service classes.

In order to properly deploy QoS, enforcement of QoS measurements and policies needs to occur all the way through the network, from the first inter-network forwarding device (like a layer 2 switch or router) to the last entity that front-ends the ultimate IP destination station. QoS requires an end-to-end approach as it requires mechanisms at the edge and in the core.

To Service Providers, QoS is desirable because it has the potential of helping them support many types of traffic (data, voice and video) over the same network infrastructure. It allows them to offer business-quality IP VPN services, and the end-to-end service level agreements (SLAs) that customers demand.

<sup>21</sup> For example, a Frame Relay switch is concerned with only input interface and input DLCI, which then gets mapped to the appropriate output interface and output DLCI. Frame Relay switching is independent of the C network's IP infrastructure.

In an MPLS environment, one needs to consider both packet and cell routers. In a packet environment, MPLS Class of Service is fairly straightforward. An MPLS LSR simply copies the IP precedence to the MPLS Class of Service field. The CoS field can then be used as input to Weighted RED as well as Weighted Fair Queuing. The challenge is to provide MPLS CoS in environments where LSRs are connected to ATM. Class of Service is more involved on ATM interfaces and within the ATM LSRs themselves. Quality of Service concepts in ATM MPLS environments are discussed later.

QoS is discussed in-depth in other resources available from Cisco. The emphasis in this section is engage the reader in investigating differentiated services in MPLS Intranet and Extranet VPN environments.

The next few pages will engage the reader in Quality-of-Service concepts, with information on the tools available from Cisco Systems. Following that, the proper QoS paradigms are highlighted in the Edge as well as the Core of a network. ATM-based MPLS and non-MPLS networks are then discussed.

#### 1.3.12.1 DiffServ

DiffServ is an emerging IETF QoS standard that will increase the available ToS bits in a packet header from the three used by IP Precedence to six enabling up to 64 classes of service. This offers Providers the ability to support very granular traffic handling. Cisco is actively participating in the development of the DiffServ standard, and plans to support it in the future.

#### 1.3.12.2 Design Approach For Implementing QoS

In mega-scale VPNs, applying QoS on a flow-by-flow basis is not practical because of the number of IP traffic flows in carrier-sized networks. The key to QoS in large-scale VPNs is implementing controls on a set of service classes that applications are grouped into. For example, a Service Provider network may implement three

service classes: a high-priority, low-latency “premium” class; a guaranteed-delivery “mission-critical” class; and a low-priority “best-effort” class. Each class of service is priced appropriately, and subscribers can buy the mix of services that suits their needs. For example, subscribers may wish to buy guaranteed-delivery, low-latency service for their voice and video conferencing applications, and best-effort service for e-mail traffic and bulk file transfers.

Because QoS requires intensive processing, the Cisco model distributes QoS duties between edge and core LSRs. This approach assumes a lower-speed, high-touch<sup>22</sup> edge and a high-speed, lower-touch core for efficiency and scalability.

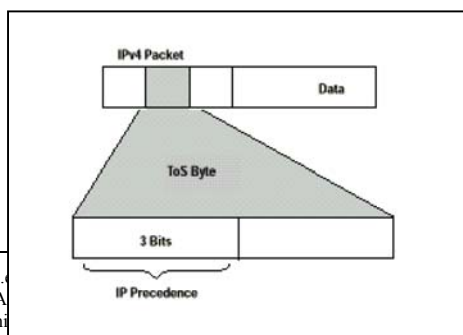
### 1.3.12.3 Cisco IOS“ QoS/CoS Toolkit

Cisco IOS Software includes several Layer 3 QoS features that are particularly applicable to VPN provisioning and management. MPLS-enabled networks make use of the following Cisco IOS QoS features to build an end-to-end QoS architecture:

- IP Precedence/DiffServ<sup>23</sup>
- Committed Access Rate (CAR)<sup>24</sup>
- Weighted Random Early Detection (WRED)
- Weighted Fair Queuing (WFQ)
- Class-Based Weighted Fair Queuing (CBWFQ)
- Modified Deficit Round Robin (M-DRR)

#### 1.3.12.3.1 IP Precedence

IP Precedence utilizes the three precedence bits in the IPv4 header Type-of-Service field to specify class of service for each packet, as shown in the figure below. One can partition traffic in up to six classes of service using IP Precedence (two others are reserved for internal network use). Queuing technologies throughout the network can use this signal to provide the appropriate expedited handling as discussed further in subsequent sections.



<sup>22</sup> Low speed.  
<sup>23</sup> Also known as DiffServ.  
<sup>24</sup> Also known as weighted rate limiting (WRL).

#### 1.3.12.3.2 Committed Access Rate (CAR)

Committed Access Rate is Cisco’s traffic policing tool for instituting a QoS policy at the edge of a network. CAR allows one to identify packets of interest for classification with or without rate limiting.

CAR allows one to define a traffic contract in routed networks. One can classify and police traffic on an incoming interface, and set policies for handling traffic that exceeds a certain bandwidth allocation. CAR can be used to set IP precedence based on extended access list classification. This allows considerable flexibility for precedence assignment, including allocation by application, port, or by source destination address, and so on. As a rule-based engine, CAR classifies traffic based on flexible rules, including IP Precedence, DiffServ (future), IP access lists, incoming interface, or MAC address. It limits the rate to the defined ingress thresholds to help allay congestion through the core.

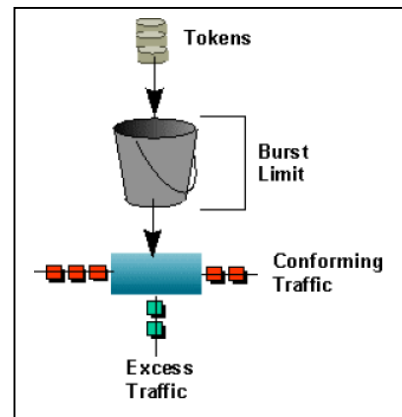


Figure 6 - CAR Sets Service Classes at the Edge of the network (Edge LSR)

The reader is encouraged to refer to the myriad of Cisco documents available on these QoS technologies. The focus here is on pertinent QoS paradigms at the edge and core of MPLS-VPNs.

### 1.3.12.3.3 Differential Discard and Scheduling Policies

Weighted Random Early Discard (WRED) is a differential discard policy applied to packets that are backing up in a queue during outbound congestion. WRED is the differentially-oriented counterpart to simple “tail drop” drop policy.

On the other hand, Weighted Fair Queuing (WFQ) is a differential scheduling policy that results in packets of different classes getting different amounts of link bandwidth during outbound congestion. WFQ is the differentially-oriented counterpart to “FIFO” scheduling policy.

#### 1.3.12.3.3.1 WRED

WRED provides congestion avoidance. This technique monitors network traffic load in an effort to anticipate and avoid congestion at common network bottlenecks, as opposed to congestion management techniques that operate to control congestion once it occurs.

WRED is designed to avoid congestion in internetworks before it becomes a problem. It leverages the flow monitoring capabilities of TCP. It monitors traffic load at points in the network and discards packets if the congestion begins to increase. The result is that the source detects the dropped traffic and slows its transmission. WRED interacts with other QoS mechanisms to identify class of service in packet flows. It selectively drops packets from low-priority flows first, ensuring that high-priority traffic gets through.

WRED is supported on the same interface as WFQ<sup>25</sup>. One needs to run both of these queueing algorithms on every interface where congestion is likely to occur. One applies WRED by IP precedence and WFQ by service class in the core.

#### 1.3.12.3.3.2 WFQ

WFQ addresses situations where it is desirable to provide consistent response time to heavy and light network users alike without adding excessive bandwidth. WFQ is a flow-based queueing algorithm that does two things simultaneously: it schedules interactive traffic to the front of the queue to reduce response time, and it fairly shares the remaining bandwidth amongst lower-priority flows.

WFQ ensures that queues are not starved for bandwidth, and that traffic achieves predictable service, so that mission-critical traffic receives highest priority to ensure guaranteed delivery and latency. Lower-priority traffic streams share the remaining capacity proportionally amongst them.

The WFQ algorithm also addresses the problem of round-trip delay variability. If multiple high-volume conversations are active, their transfer rates and inter-arrival periods are made much more predictable. Algorithms such as the Transmission Control Protocol (TCP) congestion control and slow-start features are much enhanced by WFQ. The result of WFQ is more predictable throughput and response time for each active flow.

#### 1.3.12.3.3.3 Cooperation between WFQ and IP Precedence

WFQ is IP Precedence-aware, that is, it is able to detect higher priority packets marked with precedence by the IP Forwarder and schedule them faster, providing superior response time for this traffic. The IP Precedence field has values between 0 (the default) and 7. As the precedence value increases, the algorithm allocates more bandwidth to that conversation to make sure that it gets served more quickly when congestion occurs. WFQ assigns a weight to each flow, which determines the transmit order for queued packets. It provides the ability to re-order packets and control latency at the edge and in the core. By assigning different weights to different service classes, a switch can manage buffering and bandwidth for each service class. This mechanism constrains delay bounds for time-sensitive traffic such as voice or video.

#### 1.3.12.3.3.4 Class-Based Weighted Fair Queuing

Class-based Weighted Fair Queuing (CBWFQ) provides the ability to guarantee service levels and maximize bandwidth utilization.

CBWFQ is a more sophisticated version of Cisco’s Custom Queueing feature that has been in existing in IOS for several years. It

<sup>25</sup> At least in Cisco IOS 12.0(5)T and higher. The author is not sure about other IOS revisions.

allows application service classes to be mapped to a portion of network link. For example, a QoS class can be configured to occupy at most 35% of an OC3 link.

Figure 6 provides an example of three service classes:

- “Gold”, with guaranteed latency and delivery
- “Silver”, with guaranteed delivery
- “Bronze”, a best effort service

In Service Provider MPLS-VPN CBWFQ environments, bandwidth is configured per class, not per connection.

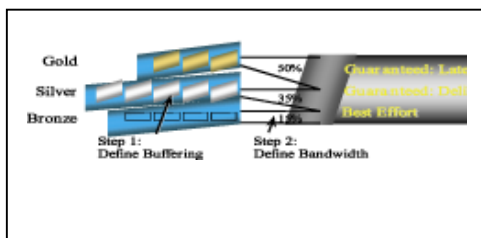


Figure 7 – Example of Class-Based Weighted-Fair Queuing

By separately allocating bandwidth and buffering space, Service Providers can tailor each class to the specific service needs of their customers. For example, a Service Provider can offer a “Gold class” for voice traffic. Here, a large bandwidth allocation policy ensures that sufficient bandwidth is available for all the calls in the voice queue while a moderately-sized buffer limits the potential cell delay. Since these shares are relative weights, allocating a large share to Gold means that a minimum is guaranteed. If the gold class is underutilized, the bandwidth will be shared by the remaining classes in proportion to their weights. This ensures maximum efficiency and that paying customer traffic will be sent if bandwidth is available.

#### 1.3.12.3.4 Modified Deficit Round Robin

Modified Deficit Round Robin (MDRR) is an mechanism in development for use in routed cores based on the Cisco 12,000 GSR. It provides CoS-based queue scheduling to assign priority to traffic based on its ToS value (as defined at the edge by CAR). A single special queue can be set to provide either Alternate or Strict priority.

Alternate priority queues are scheduled to alternate with other queues. Alternate priority assigns

different “deficit counters” to queues. Then all queues are emptied in an alternating, round-robin fashion. How much is emptied at each stop is determined by the value of the deficit counter, and this varies per-queue. For instance, if there are three service classes, there are three active queues in a GSR. Queue 1 is the special queue. In alternate priority, the GSR empties part of Queue 1 until it meets the value kept in the deficit counter. The GSR then empties Queue 2 until its deficit value is consumed, then alternates to Queue 1 again. Then it takes packets from Queue 3 and goes back to Queue 1. How much traffic is taken at a given pass is determined by the value of the deficit counter, and that is set by the administrator to reflect service class and bandwidth requirements.

Strict priority queues does not use the deficit counter, but all other queues do. Strict priority queues have absolute priority over all other traffic. The GSR always empties this queue before attending to other queues. This mechanism may cause bandwidth starvation in other queues during busy periods. Other queues are emptied in a round-robin fashion, and how much traffic is forwarded is determined by their deficit counter values, as with Alternate priority queuing.

#### 1.3.12.4 Proper QoS Tool Placement in the Network

CoS/QoS application is easy to implement in a non-ATM MPLS environment. As one needs to utilize QoS in an end-to-end fashion, two areas of implementation need to be looked at – Ingress/Egress (Edges) of the network, as well as the core.

Briefly,

- at the edges of the network traffic enforcement/policing need to be present. Therefore, at the edges of the network, Cisco’s Committed Access Rate (CAR) is required, and
- in the core of the network, concepts such as Weighted Random Early Detection (WRED); Weighted Fair Queuing (WFQ);

Class-Based WFQ (CBWFQ); and finally Modified Deficit Round Robin (MDRR<sup>26</sup>) need to be considered.

#### 1.3.12.4.1 QoS At the Edge

The next few sections simply point to QoS tools to be deployed at the Edge of a network. Details of those tools have already been covered.

##### 1.3.12.4.1.1 IP Precedence

At is at the Ingress of a network that IP Precedence setting, if policy calls for it, is modified. It is also possible, for certain environments, for the IP Precedence field to get adjusted at the Egress of the network. However, this document focuses on Ingress-based IP Precedence adjustments.

##### 1.3.12.4.1.2 Committed Access Rate

##### 1.3.12.4.2 QoS In the Core

The next few sections refer to Cisco IOS QoS tools to be deployed in the core of a network. As was the case with the Edge concepts, details of those tools have already been covered.

##### 1.3.12.4.2.1 Weighted Random Early Detection (WRED)

##### 1.3.12.4.2.2 Weighted Fair Queuing (WFQ)

##### 1.3.12.4.2.3 Class-Based WFQ (CBWFQ)

##### 1.3.12.4.2.4 Modified Deficit Round Robin (M-DRR)

### 1.3.12.5 ATM-based MPLS and QoS/CoS

Within an ATM LSR, there are two challenges. First, there's no WRED in the ATM switch. Also, because the VC is actually the label, there is no CoS field in an ATM label.

Service Providers can mark the IP packets using CAR either in the CPE<sup>27</sup> or PE routers prior to the packets getting label-switched. When the packets are MPLS-forwarded, the IP Precedence is copied to the MPLS COS field. Enabling WFQ in the backbone should be sufficient to preserve the QoS.

Cisco IOS 12.0(5)T will support CAR on non-MPLS IP packets only, but that should be sufficient for marking packets at the edge. WRED and WFQ are supported on Labelled packets on output packet interfaces; individual ATM PVCs on the "ATM

Deluxe" Port Adapters; and at the interface level on the predecessor of "PA-A3" : the "PA-A1." Interface-level queueing on the PA-A3 will be supported in a maintenance release.

MPLS CoS Phase 2 (MCP2) is expected to be available in Calendar Quarter (CQ)3, 1999. MCP2 will permit CAR to mark the Label CoS field directly (during label imposition), so that the original IP Precedence is preserved end-to-end. This is known as "CoS Transparency" and has been requested by some Service Providers. The reader is encouraged to keep in touch with engineering regarding availability of this feature.

#### 1.3.12.5.1 ATM MPLS-VPN CoS/QoS Mechanisms

For ATM LSR environments Cisco supports the following modes:

1. ATM Forum PVC
2. Single-VC LSP
3. Multi-VC LSP

Within ATM LSRs, there are three modes that a Service Provider can select in order to perform MPLS CoS. The first one would be used on an ATM Forum PVC where there are actually non-LSR ATM switches. One is able to only set up PVCs through the core. In the PVC mode, one does not actually use MPLS on the ATM switches.

The second CoS mode is using a single VC, which is a label switch path (LSP), where ABR control algorithms are used on that VC. The third mode is known as Multi-VC mode where there are also LSP VCs, but multiple VCs are set up, each having a different class associated with it. Those multiple VCs are set up in parallel along the LSP.

##### 1.3.12.5.1.1 ATM Forum-based PVC

As mentioned earlier, this is usually used in a non-MPLS enabled ATM core. The PVC looks like a packet interface and per-VC WRED and per-VC WFQ are used in a similar manner to algorithms that are

<sup>26</sup> Sometimes referred to as DRR+

<sup>27</sup> Realistically, if the CPE router is managed by the Service Provider. It is unlikely that the SP will accede to customers setting QoS knobs themselves.

applied in IP-only packet environments. In addition, one is able to choose PVC parameters, including bandwidth, whatever's available within the core on that PVC. A drawback of this mode is that there's a significant amount of configuration that's required, usually a full mesh of PVCs with all the associated configurations.

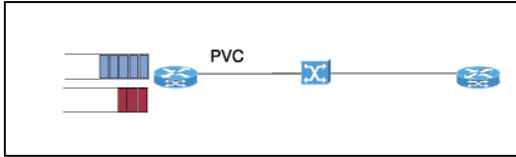


Figure 8 - ATM Forum PVC Mode

#### 1.3.12.5.1.2 Multi-VC Mode

In this environment, one can configure each PE to support multiple classes.

In Multi-VC mode the MPLS ATM core provides CoS at each link. There are multiple VCs that are established along the Label Switched Path. LSPs are automatically established, which simplifies the configuration process. In multi-VC mode, there are up to four different Label VCs (LVCs) to each destination on each ATM link, assuming VC Merge is being used. Parallel VCs are automatically established, and one can assign a weight to each class on a per-link basis. One can do that based on the expected load and desired performance of each class, just like provisioning.

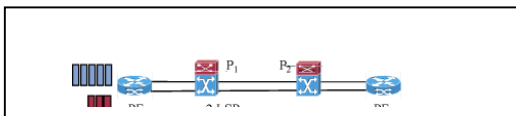


Figure 9 - Multi-VC Mode

In the Multi-VC mode, multiple labels exist per route, established by LDP. For each class of service, as well as on a route-basis, there will exist an LSP. One utilizes CAR for classifying and policing the traffic.

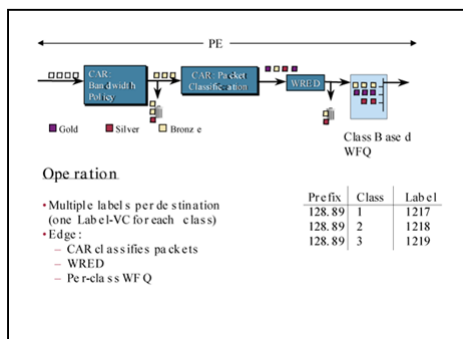


Figure 10 - Multi-VC Mode, Application of Cisco IOS QoS @Egress/Core

#### 1.3.12.5.1.3 Single-VC Mode

In a Single-VC mode, ABR service is enabled on the LSRs. In Single-VC ABR mode, there will be one LVC per destination on the link with class-based queuing at the edge feeding into the LVC. Congestion is pushed back to the edge of the ATM LSR cloud. The edge ATM LSRs respond to this feedback and manage the per-VC queues using WRED. The main benefit here is that the core becomes lossless and drop decisions are made where MPLS CoS is visible, at the Edge LSR, outside of the ATM cloud.

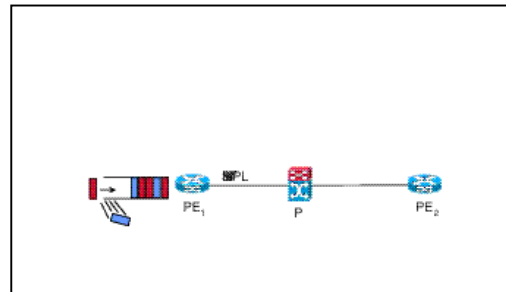
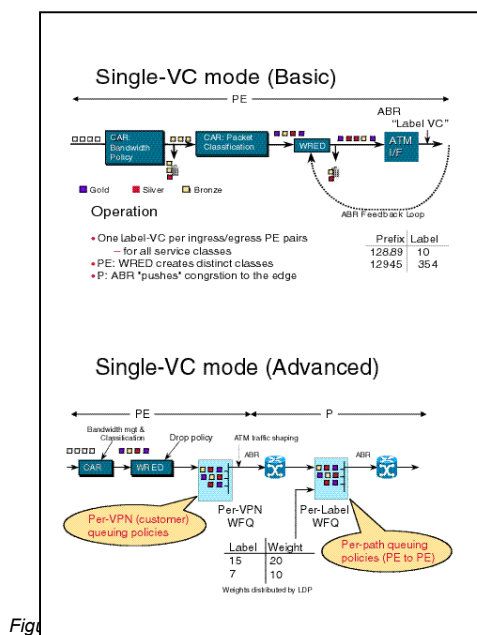


Figure 11 - Single ABR VC-Mode

One label is assigned to each destination. This label is used for all service classes.

At the Edge of this Service Provider's MPLS-VPN network, CAR is used to implement L3 bandwidth policies and stratify packets into classes. All packets are placed in a single egress interface queue. Each label implements a separate Label VC (LVC) that utilizes ABR. As in the ATM Forum ABR case, "RM" cells will be received to adjust the delivery rate. In effect, congestion is "pushed" to the edge. As congestion occurs at the interface, WRED is utilized to discard packets (before they are queued) based on service class. In this model, the core is "lossless" and WRED is used to queue packets based on priority.



Fig

#### 1.3.12.5.1.4 Single- vs Multi-VC Modes

The Multi-VC paradigm works well with VC Merge, which saves VC space, reducing the number of VCs used by SPs in a large network. Multi-VC mode will allow the different classes to be engineered separately. For example, one set of VCs may have short buffers, allowing guarantees of end-to-end delay across the network.

ABR mode, on the other hand, uses fewer VCs in small networks, but does not work with VC Merge. It's quite difficult to implement with VC Merge. And it can introduce large delays for certain classes in the network.

### 1.3.13 MPLS Traffic Engineering

It is also possible to use MPLS for traffic engineering, routing certain flows along pre-defined layer 3 pathways. At the present time, traffic engineering is manually configured on the Cisco devices. Cisco is pioneering work on an open standard for dynamically performing traffic engineering based not only on network policy, but also on congestion and link availability throughout the network. This is the Routing for Resource

Reservation<sup>28</sup> (RRR or R3) that has yet to become a standard. Other camps, lead by Nortel are advocating, for Traffic Engineering, a specification in the draft stage, called Constraint-based Routing with Label Distribution Protocol. Please refer to the Appendix titled "Cisco's MPLS Efforts," for further information.

Layer 3 traffic engineering is the ability to control specific routes across a network to reduce congestion and improve the cost-efficiency of carrying IP traffic in routed networks. The goal of Layer 3 traffic engineering is maximizing utilization of network resources. IP networks typically have multiple pathways that traffic can take to reach its destination. Relying solely on routing protocols, some paths in a service provider network may become congested, while others are underutilized.

Around the same timeframe of the release of MPLS-VPN support, MPLS will provide an elegant traffic engineering mechanism called Routing for Resource Reservation (RRR<sup>29</sup>). RRR is currently undergoing EFT<sup>30</sup> testing. It provides a flexible, scalable way to support traffic engineering at Layer 3 in service provider environments. MPLS lets managers map traffic flows to explicitly configure paths, sending selected traffic along pre-calculated routes by engineering the network to deliver specific capacity to individual VPN customers. RRR allows network operators to dynamically apply traffic-engineering rules that override the traditional IP forwarding mechanisms, providing for more efficient link utilization. RRR creates one or more explicit paths with bandwidth assurances for each traffic trunk. It takes into consideration the policy constraints associated with trunks, the physical network resources, and network topology. This way, packets are no longer routed based solely on destination address, but also on resource availability and traffic classification policy, as specified by the network administrator. It also provides fast, Layer 3 restoration and protection

<sup>28</sup> Cisco has recently decided to refer to the use of RRR in MPLS environments as "MPLS Traffic Engineering."

<sup>29</sup> Also referred to as "R3."

<sup>30</sup> Early Field Trial.

mechanisms that are not available in traditional IP networks.

### 1.3.13.1.1 RRR Requirements

RRR leverages several foundation technologies:

- CEF,
- MPLS,
- Link State IGP with extensions (OSPF with “Opaque” LSAs, as well as IS-IS with new TLV<sup>31</sup>s),
- An enhanced version of RSVP.

The reader will note that Cisco does not currently support Opaque OSPF as an IGP in MPLS TE environments.

RSVP sessions are created on a per-traffic-trunk basis for high scalability.

In summary, future releases of MPLS will implement constraint-based routing to automatically establish explicit paths for balancing traffic loads to conform to specified policy.

Next, interested reader can explore the details of □MPLS-VPN.

## 1.4 Detailed MPLS-VPN Functional Characteristics

MPLS is a standardized version of Cisco’s original Tag Switching technology. MPLS and Tag Switching are identical in principle, and nearly identical in operation. Cisco’s TDP and the MPLS’s LDP are nearly identical in general function, but use incompatible message formats and some different procedures. Cisco has a leadership position in the advancement of MPLS-VPN, as MPLS software is largely based on Cisco’s Tag Switching technology. In the future, as certain components of MPLS, like LDP, become standards, Cisco will introduce software features that will adhere to those specifications.

The reader is encouraged to consider Cisco’s value-add in the MPLS packet switching area:

- VPNs using MPLS + Multi-protocol BGPv4. That is, MPLS-VPN technology is a value-add itself
- CoS/QoS

In MPLS-VPN, a PE router is considered attached to a particular VPN if it is attached to a CE device

which is in that VPN. Similarly, a PE router is considered attached to a particular site if it is attached to a CE device which is in that site. A set of IP hosts constitutes a site if those entities communicate with one another without the use of the backbone.

In general, a site will consist of a set of IP hosts that are in geographic proximity. However, this is not always true. Take, for example, two geographic locations connected via a leased line, over which an IP routing protocol<sup>32</sup> like OSPF is running. That environment will constitute a single site, because communication between the two locations does not involve the use of the backbone.

A CE device is always regarded as being in a single site<sup>33</sup>. A site, however, may belong to multiple VPNs.

The CE router is a routing peer of the PE(s) to which it is attached, but is not a routing peer of CE routers at other sites. Routers at different sites do not directly exchange routing information with one another. As a consequence, very large VPNs (i.e., VPNs with a very large number of sites) are easily supported, while the routing strategy for each individual site is greatly simplified.

The MPLS-VPN architecture has the following characteristics:

- Conventional IP addressing is used within the Provider backbone. All provider addresses are unique, globally valid, IPv4 addresses. Furthermore, provider addresses are also unique across all customer VPNs<sup>34</sup>. This is the only restriction placed on customer address assignment.
- Routing information for provider addresses is distributed amongst provider routers using an interior routing protocol. MPLS switching is used across the provider backbone, using IGP labels based on the provider address space.

<sup>32</sup> Or even static routing.

<sup>33</sup> Although a site may consist of multiple “virtual sites”, as we shall see later in the document.

<sup>34</sup> However, we do allow to use customers’ addresses on the PE interfaces that connect the PE to the CE router. These addresses may not be globally unique.

<sup>31</sup> Type, Length, Value tuple, or parameters utilized by ISIS.

- Provider edge routers connect to one or more customer edge routers, in one or more customer VPNs.

PE routers may learn customer routing information in several ways:

- Through static configuration.
- Through EBGp. MPLS-VPN provides the flexibility as to whether or not customer address are carried forward to other VPNs, as well as to the Internet.
- Through other dynamic routing protocols that support classless routing<sup>35</sup>.

Controlled access is provided through the desired leakage of routing information. A customer can select, either through configuration or dynamically, which addresses are exported to and imported from other VPNs. In order to support firewall access, the same address can be exported from different sites internally and externally.

IBGP is used amongst PE routers to distribute customer address information. Customer addresses are exchanged using the Multi-protocol Extensions to BGP<sup>36</sup>. Addresses are encoded in "VPN-IPv4" format, creating a globally unique value by combining a VRF with the customer's IP address.

Existing BGP mechanisms, such as Route Reflectors, can be used to improve the scalability of the IBGP information exchange within a provider's backbone.

VPN-IPv4 addresses for a given VPN need only be distributed amongst the set of PE routers which connect to, or import addresses from, that VPN. This means that each PE router need only contain a subset of the full IPv4-VPN addressing table.

The VPN-IPv4 addressing information includes second-level labeling information. When a PE router forwards a unicast VPN packet to another PE router, it utilizes the first level label to select the IBGP destination - the target PE router. The second level label (VPN label) comes from the VPN-IPv4 information distributed by the target PE router, and selects the outgoing CE link.

In the future, VPN multicast forwarding across the provider backbone will be achieved, perhaps

through multicast label switching and sparse-mode PIM by using multicast VPN-IPv4 addresses in PIMv2 messages.

### 1.4.1 Per-Site<sup>37</sup> Forwarding Tables in the PEs

Each PE router maintains one or more VRF tables. A particular packet's IP destination address is looked up in the appropriate VRF table only if that packet has arrived directly through an interface which is associated with that table.

A PE generally maintains only one forwarding table per site, even if it is multiply connected to that site.

#### 1.4.1.1 Internet Connectivity

Suppose a PE router receives a packet from a particular interface, and the packet's destination address does not match any entry in the appropriate VRF table<sup>38</sup>. If the SP is providing Internet access for that site, then the PE's Internet forwarding table will be consulted.

At this point, Cisco's MPLS-VPN code supports VRF default routing through the injection of the default route into each VPN that the PE participates in<sup>39</sup>. In a future release, Cisco PE routers will provide the ability to define a "default VPN" (or VPN0), to simplify the configuration of default routing.

#### 1.4.1.2 My VPN Doesn't Talk to Your VPN

To maintain proper isolation of one VPN from another, it is important that P router not accept a labeled packet from any adjacent PE router unless

- the label at the top of the label stack was actually distributed by the P router to the PE device, and
- the P router can determine that use of that label will cause the packet to leave the backbone before any labels lower in the

<sup>35</sup> Therefore RIP I and IGRP are NOT appropriate IGP's. At the present time, Cisco supports RIP II. Support for OSPF will occur in the future.

<sup>36</sup> RFC2283, "Multi-protocol Extensions for BGP-4."

<sup>37</sup> We shall be using the terms "per-site forwarding table" and "VRF table" interchangeably.

<sup>38</sup> That is, the forwarding table associated with that site.

<sup>39</sup> That is, for every site that requires Internet connectivity.

stack will be inspected, and before the IP header will be inspected. These restrictions are necessary in order to prevent packets from entering a VPN where they do not belong.

The per-VRF tables in a PE router are only used for packets arriving from a CE router that is directly attached to the PE device. They are not used for routing packets arriving from other routers that belong to the SP backbone. As a result, there may be multiple different routes to the same system, where the route followed by a given packet is determined by the site from which the packet enters the backbone. So one may have one route to a given IP network for packets from the extranet (where the route leads to a firewall), and a different route to the same network for packets from the intranet.

#### 1.4.1.3 Virtual Sites

In some cases, a particular site may be divided by the customer into several virtual sites, perhaps by the use of VLANs. Each virtual site may be a member of a different set of VPNs. The PE then needs to contain a separate forwarding table for each virtual site. For example, if a CE supports ISL or 802.1Q VLANs, and wants each VLAN mapped to a separate VPN, the packets sent between CE and PE will be contained in the site's VLAN encapsulation, and this can be used by the PE router to assign the packet to a particular virtual site.

Support for VRF on a VLAN trunking port has not ☐ yet been tested, but is available in the code on the platforms that already have CEF (FIB) support on those VLAN ports.

CEF support for 802.1Q on the GSR's Gigabit Ethernet Line Card is expected to be made available in the 12.0 S code path in July. The author of this document does not currently know whether that CEF switching on 802.1Q trunks has been committed for the 7500 family yet.

Only one CE router is ever needed per site, even if there are multiple virtual sites. Of course, a different CE router can be used for each virtual site, if that is desired. One can then design a customer VPN that has a CE router with multiple VPNs configured on a VLAN trunking port, with packets from the customer site funneling through a Catalyst LAN switch with the proper VLAN encapsulation.

#### 1.4.2 Same VPN, Different Routes to the Same Address

Although a customer site may be in multiple VPNs, MPLS-VPN provides the controls for differentiated routes to a given host at that site. Suppose, for example, assume one has an intranet consisting of sites A, B, and C, and an extranet consisting of A, B, C, and the external site D. Suppose that at site A there is a server, and that the goal is for clients from B, C, or D to be able to use that server. Suppose also that at site B there is a firewall. One needs the ability to direct all the traffic from site D to the server to pass through the firewall, so that traffic from the extranet can be access controlled. On the other hand, traffic from C need not pass through the firewall on the way to the server, since this is intranet traffic.

MPLS-VPN provides the flexibility to set up two routes to the server. One route, used by sites B and C, takes the traffic directly to site A. The second route, used by site D, takes the traffic instead to the firewall at site B. If the firewall allows the traffic to pass, it then appears to be traffic coming from site B, and follows the route to site A.

#### 1.4.3 MPLS-VPN Backbone

The SP's backbone is comprised of the PE and P routers.

The MPLS-VPN paradigm provides the ability that the routing information about a particular VPN be present ONLY in those PE routers which attach to that VPN. In particular, the P routers do not need to have ANY per-VPN routing information whatsoever.

It is possible for VPNs to span multiple service providers. At this stage, Cisco's MPLS-VPN does not support inter-provider connectivity. The author of this document will publish availability information regarding this feature, as soon as it is obtained from Product Marketing.

Once supported, the path between PE routers that represent two different Service Providers will be part of a private peering

arrangement between the two providers. At that point, there will exist mutual trust between the two providers. In particular, each provider must trust the other to pass it only correct routing information, and to pass it MPLS-labeled packets only if those packets have been labeled by trusted sources.

#### 1.4.4 A Set of Sites Inter-connected via a MPLS-VPN Backbone

If all the sites in a VPN are owned by the same enterprise, the VPN is a corporate “intranet.” If the various sites in a VPN are owned by different enterprises, the VPN is an “extranet.” A site can be in more than one VPN; e.g., in an intranet and several extranets. Both intranets and extranets are regarded as VPNs.

While the basic unit of inter-connection is the site, the MPLS-VPN architecture allows a finer degree of granularity in the control of inter-connectivity. For example, at a given site, it may be desirable to allow only certain specified systems to connect to certain other sites. That is, certain systems at a site may be members of an intranet as well as members of one or more extranets, while other systems at the same site may be restricted to being members of the intranet only.

We regard both intranets and extranets as VPNs. In general, when we use the term VPN we will not be distinguishing between intranets and extranets.

A CE router can be in multiple VPNs, although it can only be in a single site. When a CE router is in multiple VPNs, one of these VPNs will be considered its primary VPN. In general, a CE router’s primary VPN will be the intranet which includes the CE router’s site.

A PE router may attach to CE routers in any number of different sites, whether those CE routers are in the same or in different VPNs. A CE router may, for robustness, attach to multiple PE routers, of the same or of different service providers.

A PE router attaches to a particular VPN if it is a router adjacency of a CE router which is in that VPN.

#### 1.4.5 CE-PE Routing Exchange

The PE routers which attach to a particular VPN need to know, for each of that VPN’s CE routers, which addresses in that VPN are at each CE’s site.

There are three possible ways that a PE router can obtain this information from the CE router:

1. The PE and CE routers may be BGP peers, and the CE router may use BGP to tell the PE router the set of address prefixes which are at the CE router’s site. This interaction is discussed in more detail in a later section<sup>40</sup>. Great care has been taken to simplify the use of BGP in the most common scenarios. When BGP is used between a CE and a PE router, it is always EBGP.
2. The PE and CE routers may be RIPv2 peers, and the CE may use RIPv2 to tell the PE router the set of address prefixes which are reachable at the CE router’s site. This interaction is discussed in more detail in the section titled “Alternatives to BGP for CE/PE Routing Exchange.”
3. Static routing may be used. That is, the PE router may be configured with the set of address prefixes reachable via a particular CE router.<sup>41</sup>

Support for OSPF as an alternative IGP between the CE and PE routers will be incorporated in the future.

To prevent loops, the routes which a PE learns from a CE should be routes to systems which are at the CE’s site. If a PE learns from a particular CE about a route which is not at that CE’s site, some special procedure must be used to ensure that the PE can determine that the route leads off the site.<sup>42</sup>

A CE router is a routing peer of only the PE router(s) to which it attaches; and, via static, or typically an IGP, other CE routers which are at the same site as the CE router.

#### 1.4.6 Backdoor Connections

Suppose one has the customer environment depicted in figure 13 below. There are three CE routers – CE<sub>B1</sub>, CE<sub>B2</sub>, and CE<sub>B3</sub>, connected respectively to PE<sub>1</sub>, PE<sub>2</sub>, and

<sup>40</sup> The section titled “BGP Between CE and PE Routers.”

<sup>41</sup> See section “Alternatives to BGP for CE/PE Routing Exchange.”

<sup>42</sup> See the sections titled “BGP Between CE and PE Routers,” and “Alternatives to BGP for CE/PE Routing Exchange.”

PE<sub>3</sub>. Suppose further that CE<sub>B1</sub><sup>1</sup> has a “backdoor” IP connection with CE<sub>B1</sub><sup>2</sup> (e.g., the two CE routers are speaking OSPF to each other on a non-MPLS-VPN interface). What are the routing implications?

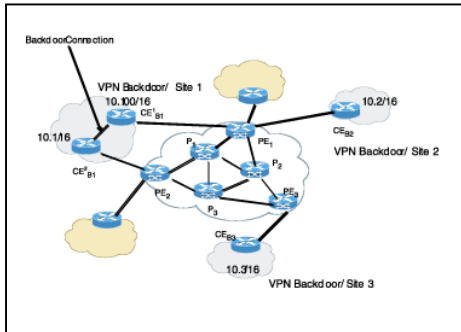


Figure 13 – CE Backdoor Scenario

PE<sub>2</sub> receives IBGP routes for CE<sub>B1</sub><sup>1</sup>'s networks via PE<sub>1</sub>. PE<sub>2</sub> also receives EBGP routes for the same networks<sup>43</sup>, from neighbor CE<sub>B1</sub><sup>2</sup>. Assuming normal route-weighting criteria<sup>44</sup>, then PE<sub>2</sub> will select routes via CE<sub>B1</sub><sup>Y</sup>.

BGP “multipath” functions in an EBGP<sup>45</sup> only environment, and does not apply to IBGP. Therefore from PE<sub>3</sub>'s perspective, the preferred route will be, let's say, the one via PE<sub>1</sub>. Selection criteria are based on standard BGP route selection processes.

One must be careful in this and other backdoor environments:

- With RIP or IGRP<sup>46</sup> as the routing protocol between the two CE routers, one needs to have some route filtering in the CEs to ensure that only intra-site routes get distributed to the PEs.
- If CE<sub>B1</sub><sup>1</sup> and CE<sub>B1</sub><sup>2</sup> are speaking BGP to each other, then looping is prevented by the fact that a route distributed by PE<sub>1</sub> carries PE<sub>1</sub>'s ASN in its AS-path. However, one must make sure not to combine it with any tricks that cause PE<sub>1</sub> to ignore the occurrence of its ASN in the AS-path.
- OSPF protocol between CE<sub>B1</sub><sup>1</sup> and CE<sub>B1</sub><sup>2</sup> is less likely to cause routing loops, as it inherently distinguishes between interior and exterior routes. However, it is still recommended that route filtering

be applied at the ASBR boundaries to not accept intra-site routes from the PE neighbors.

### 1.4.7 Per-site VRFs on PEs

Each PE router maintains one or more “per-site forwarding tables”. Every site to which the PE router is attached is associated with one of these tables. A particular packet's IP destination address is looked up in a particular per-site forwarding table only if that packet has arrived directly from a site which is associated with that table.

#### 1.4.7.1 Development of VRF Entries

To illustrate how a PE develops its forwarding table(s), let us offer an example. Let PE<sub>1</sub>, PE<sub>2</sub>, and PE<sub>3</sub> be three PE routers, and let CE<sub>1</sub>, CE<sub>2</sub>, and CE<sub>3</sub> be three CE routers. Suppose that PE<sub>1</sub> learns, from CE<sub>1</sub>, the routes that are reachable at CE<sub>1</sub>'s site. If PE<sub>2</sub> and PE<sub>3</sub> are attached respectively to CE<sub>2</sub> and CE<sub>3</sub>, and there is some VPN V containing CE<sub>1</sub>, CE<sub>2</sub>, and CE<sub>3</sub>, then PE<sub>1</sub> uses BGP to distribute to PE<sub>2</sub> and PE<sub>3</sub> the routes, which it has learned from CE<sub>1</sub>. PE<sub>2</sub> and PE<sub>3</sub> use these routes to populate the forwarding tables that they associate, respectively with the sites of CE<sub>2</sub> and CE<sub>3</sub>. Routes from sites that are not in VPN V do not appear in these forwarding tables, which means that packets from CE<sub>2</sub> or CE<sub>3</sub> cannot be sent to sites that are not in VPN V.

If a site is in multiple VPNs, the forwarding table associated with that site can contain routes from the full set of VPNs of which the site is a member.

A PE generally maintains only one forwarding table per site, even if it is multiply connected to that site. Also, different sites can share the same forwarding table if they are meant to use exactly the same set of routes.

#### 1.4.7.2 Default

Suppose a PE router from a particular directly attached site receives a packet, but the packet's destination address does not match any entry in the forwarding table

<sup>43</sup> If so configured.

<sup>44</sup> That is, using default administrative distances.

<sup>Y</sup> Default Cisco routing administrative distances for EBGP and IBGP are 20 and 200, respectively.

<sup>45</sup> And for paths coming from the same neighboring AS.

<sup>46</sup> RIP and IGRP are singled out since they have no inherent mechanism for distinguishing between interior and exterior routes.

associated with that site. If the SP is not providing Internet access for that site, then the packet is discarded as undeliverable. If the SP is providing Internet access for that site, then the PE's Internet forwarding table will be consulted. This means that in general, only one forwarding table per PE need ever contain routes from the Internet, even if Internet access is provided. Cisco IOS does not currently support the ability to revert to the "global" routing table (i.e., the router's "standard" or "non-VRF" IP routing table that everyone is used to seeing in Cisco IOS), if there is a route lookup failure in a VRF. So one has to inject default routes into each VRF that requires that kind of connectivity. So it is appropriate, using the current IOS MPLS-VPN functionality, to inject default into the VRF routing table, for example, from a centralized server site with Internet connectivity. In this case, the PE router peered with the CE device connected to the centralized server can inform its PE peers of the default route that it either originates or learns dynamically from the CE neighbor. It is also possible in this centralized environment to server multiple VPNs.

As an alternative to the "global routing table" capability mentioned above, there is also a plan to have a VPN0 on a PE router. The future VPN0 capability will allow PE IBGP peers to exchange the global Internet routes amongst themselves or with a Route Reflector.

### 1.4.7.3 Traffic Isolation

To maintain proper isolation of one VPN from another, it is important that no router in the backbone accept a labeled packet from any adjacent non-backbone device unless (a) the label at the top of the label stack was actually distributed by the backbone router to the non-backbone device, and (b) the backbone router can determine that use of that label will cause the packet to leave the backbone before any labels lower in the stack will be inspected, and before the IP header will be inspected. These restrictions are necessary in order to prevent packets from entering a VPN where they do not belong.

The per-site forwarding tables in a PE are ONLY used for packets that arrive from a site that is directly attached to the PE. They are not used for routing packets that arrive from other routers that belong to the SP backbone. As a result, there may be multiple different routes to the same address,

where the route followed by a given packet is determined by the site from which the packet enters the backbone. For example, one may have one route to a given IP address for packets from the extranet (where the route leads to a firewall), and a different route to the same IP address for packets from the intranet.

### 1.4.8 PEs Redistribute Customer Routes to One Another

PE routers use IBGP to distribute VPN routes to one another, or rather, to cause VPN routes to be distributed to one another.

A BGP speaker can only install and distribute one route to a given address prefix. The inclusion of the VPN-IPv4 Address Family allows a Service Provider to set up each customer VPN to have its own IP address space, which means that the same address can be used in any number of VPNs, where in each VPN the address denotes a different system. MPLS-VPN makes it possible for BGP to install and distribute multiple routes to a single IP address prefix. Utilization of policy route maps helps determine which sites can use which routes. One must however ensure that when several such routes are installed by BGP, only one such must appear in any particular VRF forwarding table.

These goals are met by the use of a new address family, the VPNv4 or VPN-IPv4 Address Family.

#### 1.4.8.1 VPN-IPv4 Address Family

The BGP Multi-protocol Extensions allow BGP to carry routes from multiple "address families." MPLS-VPN utilizes the "VPN-IPv4 address family." A VPN-IPv4 address is a 12-byte quantity, beginning with an 8-byte "Route Distinguisher (RD)" and ending with a 4-byte IPv4 address. If two VPNs use the same IPv4 address prefix, the PEs translate these into unique VPN-IPv4 address prefixes. This ensures that if the same address is used in two different VPNs, it is possible to install two completely

different routes to that address, one for each VPN.

The RD does not by itself impose any semantics. It contains no information about the origin of the route or about the set of VPNs to which the route is to be distributed. The purpose of the RD is solely to allow one to create distinct routes to a common IPv4 address prefix. Other means are used to determine where to redistribute the route<sup>47</sup>.

The RD can also be used to create multiple different routes to the very same host or subnet. Earlier, we gave an example where the route to a particular server had to be different for intranet traffic than for extranet traffic. Creating two different VPN-IPv4 routes that have the same IPv4 part, but different RDs can achieve this. This allows BGP to install multiple different routes to the same system, and allows policy routing to be used<sup>48</sup> to decide which packets use which route.

The RDs are structured so that every service provider can administer its own “numbering space” (i.e., can make its own assignments of RDs), without conflicting with the RD assignments made by any other service provider. An RD consists of a two-byte type field, an administrator field, and an assigned number field. The value of the type field determines the lengths of the other two fields, as well as the semantics of the administrator field. The administrator field identifies an assigned number authority, and the assigned number field contains a number, which has been assigned, by the identified authority, for a particular purpose. For example, one can have an RD whose administrator field contains an Autonomous System number (ASN), and whose (4-byte) number field contains a number assigned by the SP to whom IANA has assigned that ASN. RDs are given this structure in order to ensure that an SP, which provides VPN backbone service, can always create a unique RD when it needs to do so. However, the structuring provides no semantics. When BGP compares two such address prefixes, it ignores the structure entirely.

If the Administrator sub-field and the Assigned Number sub-field of a VPN-IPv4 address are both set to all zeroes, the VPN-IPv4 address is considered to have exactly the same meaning as the corresponding globally-unique IPv4 address. In particular, this VPN-IPv4 address and the corresponding globally unique IPv4 address will be

considered comparable by BGP. In all other cases, a VPN-IPv4 address and its corresponding globally unique IPv4 address will be considered non-comparable by BGP.

A given VRF table will only have one VPN-IPv4 route for any given IPv4 address prefix. When a packet’s destination IP address is matched against a VPN-IPv4 route, only the IPv4 part is actually matched.

A PE needs to be configured to associate routes, which lead to a particular CE with a particular RD. The PE may be configured to associate all routes leading to the same CE with the same RD, or it may be configured to associate different routes with different RDs, even if they lead to the same CE.

#### 1.4.8.2 Import & Export Route Policy

In this section, we discuss the way in which the distribution of the VPN-IPv4 routes is controlled.

##### 1.4.8.2.1 Target VPN Attribute

Every VRF table is associated with one or more “Target VPN” attributes.

When a VPN-IPv4 route is created by a PE router, it is associated with one or more “Target VPN” attributes. These are carried in BGP as attributes of the route.

Any route associated with a target VPN must be distributed to every PE router that has a forwarding table associated with that target VPN. When a PE router receives such a route, it is eligible to be installed in each of the PE’s per-site forwarding tables<sup>49</sup> that are associated with that target VPN. (Whether it actually gets installed depends on the outcome of the BGP decision process.)

In essence, a Target VPN Attribute identifies a set of sites. Associating a particular Target VPN attribute with a route allows that route to be placed in the VRF tables that are used for routing traffic which is received from the corresponding sites.

<sup>47</sup> See section “The Target VPN Attribute.”

<sup>48</sup> Refer to the section “The VPN of Origin Attribute.”

<sup>49</sup> Or VRF tables.

There is a set of Target VPNs that a PE router attaches to a route received from site S (export function, into the IBGP routing table to distribute to other peer PEs for that VPN). And there is a set of Target VPNs that a PE router uses to determine whether a route received from another PE router could be placed in the forwarding table associated with site S (import function). The two sets are distinct, and need not be the same. In a majority of customer VPN environments, however, one expects to have the import and exports functions to be the same.

The function performed by the Target VPN attribute is similar to that performed by the BGP Communities Attribute. However, the format of the latter is inadequate, since it allows only a two-byte numbering space. It would be fairly straightforward to extend the BGP Communities Attribute to provide a larger numbering space. It should also be possible to structure the format, similar to what we have described for RDs, so that a type field defines the length of an administrator field, and the remainder of the attribute is a number from the specified administrator's numbering space.

When a BGP speaker has received two routes to the same VPN-IPv4 prefix, it chooses one, according to the BGP rules for route preference.

A route can only have one RD, but it can have multiple Target VPNs. In BGP, scalability is improved if one has a single route with multiple attributes, as opposed to multiple routes. One can eliminate the Target VPN attribute by creating more routes (i.e., using more RDs), but the scaling properties would be less favorable.

#### 1.4.8.3 Route Re-distribution

If two sites of a VPN attach to PEs which are in the same Autonomous System, the PEs can distribute VPN-IPv4 routes to each other by means of an IBGP connection between them. Alternatively, each can have an IBGP connection to a route reflector. When a PE router distributes a VPN-IPv4 route via BGP, it uses its own address as the "BGP next hop". It also assigns and distributes an MPLS label. (Essentially, PE routers distribute not VPN-IPv4 routes, but Labeled VPN-IPv4 routes<sup>50</sup>) When the PE processes a received packet that has this label at the top of the stack, the PE will pop the stack, and

send the packet directly to the site from to which the route leads. This will usually mean that it just sends the packet to the CE router from which it learned the route.

In most cases, the label assigned by a PE will cause the packet to be sent directly to a CE, and the PE that receives the labeled packet will not look up the packet's destination address in any forwarding table.

The MPLS label that is distributed in this way is only usable if there is a label-switched path (LSP) between the router that installs a route and the BGP next hop of that route. We do not make any assumptions about the procedure used to set up that label switched path. It may be set up on a pre-established basis, or it may be set up when a route that would need it is installed. It may be a "best effort" route, or it may be a traffic-engineered route. Between a particular PE router and its BGP next hop for a particular route there may be one LSP, or there may be several, perhaps with different QoS characteristics. All that matters for the VPN architecture is that some label switched path between the router and its BGP next hop exists.

All the usual techniques for using Route Reflectors<sup>51</sup> to improve scalability, e.g., RR hierarchies, are available. If route reflectors are used, there is no need to have any particular Route Reflector know all the VPN-IPv4 routes for all the VPNs supported by the backbone. One can have separate route reflectors, which do not communicate with each other, each of which supports a subset of the total set of VPNs.

#### 1.4.8.4 Building VPNs with Extended Community Attributes

By setting up the Target VPN and VPN of Origin attributes properly, one can construct different kinds of VPNs.

<sup>50</sup> Please refer to Rekhter and Rosen, "Carrying Label Information in BGP4", Work in Progress.

<sup>51</sup> Bates, T. and R. Chandrasekaran, "BGP Route Reflection: An alternative to full mesh IBGP", RFC 1966, June 1996. Reader can also refer to Sam Halabi's textbook on routing in the Internet, which provides BGP4 details.

If a customer requests that an SP create a Closed User Group (CUG) which contains a particular set of sites, it will be possible to do so by creating a particular Target VPN attribute value to represent the CUG. This value then needs to be associated with the VRF tables for each site in the CUG, and it needs to be associated with every route learned from a site in the CUG. Any route which has this Target VPN attribute will need to be redistributed so that it reaches every PE router attached to one of the sites in the CUG.

Alternatively, suppose one desired, for whatever reason, to create a “hub and spokes” kind of VPN. This can be done by the use of two Target Attribute values, one meaning “Hub” and one meaning “Spoke”. Then routes from the spokes could be distributed to the hub, without causing routes from the hub to be distributed to the spokes.

Suppose one has a number of sites that are in an intranet and an extranet, as well as a number of sites, which are in the intranet only. Then there may be both intranet and extranet routes, which have a Target VPN identifying the entire set of sites. The sites which are to have intranet routes only can filter out all routes with the “wrong” VPN of Origin.<sup>52</sup>

These two attributes allow great flexibility in allowing one to control the distribution of routing information among various sets of sites, which in turn provides great flexibility in constructing VPNs.

#### 1.4.8.5 Packet Forwarding across the Backbone

MPLS-VPN utilizes a two-level label stack to allow intermediate routers in the backbone, that do not have any information about the routes to the VPNs, to forward packets from one VPN site to another.

PE routers need to insert /32 IP host address prefixes for themselves into the IGP routing tables of the backbone. This enables MPLS, at each node in the backbone network, to assign a label corresponding to the route to each PE router.

PE routers (and, in the future, ASBRs, which redistribute VPN-IPv4 addresses<sup>53</sup>) need to insert /32 address prefixes for themselves into the IGP

routing tables of the backbone. This enables MPLS, at each node in the backbone network, to assign a label corresponding to the route to each PE router.

When a PE receives a packet from a CE device, it chooses a particular VRF table in which to look up the packet’s destination address. If a match is found, then if the packet is destined for a CE device attached to this same PE, the packet is sent directly to that CE device.

If the packet is not destined for a CE device attached to this same PE, the packet’s “BGP Next Hop” is found, as well as the label which that BGP next hop assigned for the packet’s destination address. This label is pushed onto the packet’s label stack, and becomes the bottom label. Then the PE looks up the IGP route to the BGP Next Hop, and thus determines the IGP next hop, as well as the label assigned to the address of the BGP next hop by the IGP next hop. This label gets pushed on as the packet’s top label, and the packet is then forwarded to the IGP next hop. (If the BGP next hop is the same as the IGP next hop, the second label may not need to be pushed on, however.)

At this point, MPLS will carry the packet across the backbone and into the appropriate CE device. That is, all forwarding decisions by P routers and PE routers are now made by means of MPLS, and the packet’s IP header is not looked at again until the packet reaches the CE device. The final PE router will pop the last label from the MPLS label stack before sending the packet to the CE device, thus the CE device will just see an ordinary IP packet.

When a packet enters the backbone from a particular site via a particular PE router, the packet’s route is determined by the contents of the forwarding table, which that PE router associated with that site. The forwarding tables of the PE router where the packet leaves the backbone are not relevant. As a result, one may have multiple routes to the same system, where the particular route chosen for a particular packet is based on

<sup>52</sup> As stated elsewhere in this document, VPN-of-Origin support in Cisco IOS software is slated for a future release.

<sup>53</sup> Current Cisco MPLS-VPN software does not support inter-Service Provider connectivity

the site from which the packet enters the backbone.

### 1.4.9 PEs Learn Routes from CEs

Before a PE can redistribute a VPN-IPv4 route learned from a site, it must assign certain attributes to the route. There are three such attributes<sup>54</sup>:

#### 1. Site of Origin

This attribute uniquely identifies the site of the corresponding CE router from which the PE router learned the route.

The same Site of Origin attribute must be used for all CE routers that are at the same site, whether or not those CEs are attached to the same PE.

Distinct Site of Origin attributes must be used for CE routers which are at distinct sites.

This attribute will be used, when distributing the routes among the PEs, to prevent certain sorts of loops.

A route must be associated with at most one attribute of this type.

#### 2. VPN of Origin

This attribute uniquely identifies the primary VPN of the corresponding CE router. It will typically be used to identify the CE router's intranet, if there is one. When attached to a route, this attribute can then be thought of as identifying the enterprise owning the system or systems to which the route corresponds.

In situations in which it is necessary to identify the source of a route, it is this attribute, not the RD, which must be used.

A route must be associated with at most one attribute of this type.

#### 3. Target VPN

This attribute uniquely identifies a VPN or a set of VPNs to which the route should be distributed. It is the value of this attribute (a) which determines the set of PE routers that will receive this route, and (b) for each such PE, determines the set of CE routers which will receive this route.

A route may be associated with multiple attributes of this type.

If a route is to be distributed only within its intranet, then a single Target VPN attribute should be associated with the route, and its value field (see section "Route Distinguishers (RDs)") should be identical to the value field of the route's VPN of Origin attribute. If the route is also to be distributed to one or more extranets, additional Target VPN attributes will be used.

The use of a single Target VPN attribute value which identifies a set of VPNs can be used to provide a Closed User Group concept (see the "Closed User Groups" section). In this case, the VPN of Origin will not be necessary.

These attributes are to be encoded as BGP Extended Communities Attributes<sup>55</sup>.

#### 1.4.9.1 PEs Redistribute VPN-IPv4 Routes into IPv4 VRFs

How does a PE know which set of routes needs to be distributed to which set of VPNs. The architecture supports three different models:

1. Manual configuration of the PE router.
2. When the CE and the PE talk BGP to each other, one may allow the CE to specify the Target VPN attribute (but not the Site of Origin or VPN of Origin attributes). The PE will respect the Target VPN attribute specified by the CE.
3. A hybrid mode, in which the PE is configured with a list of acceptable Target VPNs, and the CE specifies from among this list. The PE may also be configured with a set of mandatory Target VPNs. In this mode, the PE removes any Target VPN attributes which the CE specifies but which aren't in the configured list of acceptable Target VPNs; the PE then adds any additional Target VPNs which are in the mandatory category.

<sup>54</sup> At the present time, Cisco's MPLS-VPN code supports neither the VPN of Origin nor the Site of Origin Extended Attributes.

<sup>55</sup> Section "BGP Extended Community Attributes."

The PE will assign a label to each route that it learns from the CE. It will know that if it receives any labeled packets from another P router that have this label value, the label stack must be popped and the packet forwarded to the particular CE router. While it is allowable for the PE router to assign the same label to all routes it learns from a given CE, it is highly recommended that the PE router assign a different label to each route. This will allow a particular route to migrate from one CE router to another without requiring the PE router to redistribute any labels to its BGP peers.

The PE router translates IPv4 addresses received from the CE router, into VPN-IPv4 addresses, using the RD configured on the interface through which the IPv4 packet came. The PE then treats these VPN-IPv4 routes as input to BGP. In no case will routes from a site ever be leaked into the backbone's IGP.

#### 1.4.9.2 PE-CE Routing Protocol Options

Exactly which PE-CE route re-distribution techniques are possible depends on whether a particular CE is in a "transit VPN" or not. A "transit VPN" is one which contains a router that receives routes from a "third party" (i.e., from a router which is not in the VPN, but is not a PE router), and that redistributes those routes to a PE router. A VPN that is not a transit VPN is a "stub VPN". The vast majority of VPNs, including just about all corporate enterprise networks, are expected to be "stubs" in this sense.

The possible PE/CE distribution techniques are:

1. Static routing (i.e., configuration) may be used. (This is likely to be useful only in stub VPNs.)
2. PE and CE routers may be RIP peers, and the CE may use RIP to tell the PE router the set of address prefixes which are reachable at the CE router's site. When RIP is configured in the CE, care must be taken to ensure that address prefixes from other sites (i.e., address prefixes learned by the CE router from the PE router) are never advertised to the PE. More precisely: if a PE router, say PE1, receives a VPN-IPv4 route R1, and as a result distributes an IPv4 route R2 to a CE, then R2 must not be distributed back from that CE's site to a PE router, say PE2, (where PE1 and PE2 may be the same router or different routers), unless PE2 maps R2 to

a VPN-IPv4 route which is different than (i.e., contains a different RD than) R1.

3. The PE and CE routers may be OSPF peers. In this case, the site should be a single OSPF area, the CE should be an ABR in that area, and the PE should be an ABR which is not in that area. Also, the PE should report no router links other than those to the CEs which are at the same site. (As mentioned earlier, Cisco's software does not currently support OSPF as a CE-PE IGP)
4. The PE and CE routers may be BGP peers, and the CE router may use BGP - in particular, EBGP to tell the PE router the set of address prefixes which are at the CE router's site. (This technique can be used in stub VPNs or transit VPNs.)

From a purely technical perspective, this is by far the best technique:

- a) Unlike the IGP alternatives, this does not require the PE to run multiple routing algorithm instances in order to talk to multiple CEs
- b) BGP is explicitly designed for just this function - passing routing information between systems run by different administrations
- c) If the site contains "BGP backdoors", i.e., routers with BGP connections to entities other than PE routers, this procedure will work correctly in all circumstances. The other procedures may or may not work, depending on the precise circumstances.
- d) Use of BGP makes it easy for the CE to pass attributes of the routes to the PE. For example, the CE may suggest a particular Target for each route, from among the Target attributes that the PE is authorized to attach to the route.

On the other hand, using BGP is likely to be something new for the CE administrators, except in the case where the customer itself is already an Internet Service Provider (ISP). If a site is not in a transit VPN, note that it need not have a unique Autonomous System Number (ASN). Every CE whose site which is not in a transit VPN can use the same ASN. This can be chosen from the private ASN space, and the PE will strip it out.

Routing loops are prevented by use of the Site of Origin Attribute.

#### 1.4.10 CEs Learn Routes from PEs

In general, a PE may distribute to a CE any route which the PE has placed in the forwarding table which it uses to route packets from that CE. There is one exception: if a route's Site of Origin attribute identifies a particular site, that route must never be redistributed to any CE at that site.

In most cases, however, it will be sufficient for the PE to simply distribute the default route to the CE. (In some cases, it may even be sufficient for the CE to be configured with a default route pointing to the PE.) This will generally work at any site which does not itself need to distribute the default route to other sites. (For example, if one site in a corporate VPN has the corporation's access to the Internet, that site might need to have default distributed to the other site, but one could not distribute default to that site itself.)

#### 1.4.11 ISP as a Stub VPN

If a particular VPN is actually an ISP, but its CE routers support MPLS, then the VPN can actually be treated as a stub VPN. The CE and PE routers need only exchange routes which are internal to the VPN. The PE router would distribute to the CE router a label for each of these routes. Routers at different sites in the VPN can then become BGP peers. When the CE router looks up a packet's destination address, the routing lookup always resolves to an internal address, usually the address of the packet's BGP next hop. The CE labels the packet appropriately and sends the packet to the PE.

##### 1.4.11.1 Encoding VPN-IPv4 Address Prefixes in BGP

The procedures defined in RFC 2283, Multi-protocol Extensions for BGP-4, will be used to carry Labeled VPN-IPv4 address prefixes amongst PE routers. The Labeled VPN-IPv4 address prefix appears in the Network Layer Reachability Information (NLRI) field.

Before two BGP peers can make use of the Labeled VPN-IPv4 address family, they must both agree to do so by means of a Capability Negotiation.

The NLRI is encoded as one or more triples of the form <label, length, prefix>:

##### 1. Length: 1 octet

The Length field indicates the length of the label (24 bits) plus the length in bits of the address prefix, including the length of the RD. Thus its minimum value is 88, corresponding to a label and an RD, followed by an IPv4 prefix of length 0. This will indicate a prefix that matches all VPN-IPv4 addresses that begin with the specified RD. The maximum value of this field is 120, corresponding to a labeled VPN-IPv4 host address.

##### 2. Label: 3 octets

The Label field carries one or more labels (that correspond to the stack of labels). Each label is encoded as 3 octets, where the high-order bit contains "Bottom of Stack". The following high-order three bits must be zero. The remaining 20 bits contain the label value.

##### 3. Prefix:

- Route Distinguisher: 8 octets
- IPv4 address prefix

This contains the IPv4 prefix, padded out to as many bits as needed to make it end on an octet boundary. (The value of any such padding bits is irrelevant.)

The encoding described above allows a single BGP Update message to carry multiple VPN-IPv4 routes, each with its own label(s). The label(s) specified for a particular route (and associated with its prefix) must be assigned by the router which is identified by the value of the Next Hop attribute of the route.

In the future, it may be possible to support the simultaneous installation of multiple routes to the same VPN-IPv4 address prefix, as long as each route is associated with a different Label value. Currently, however, BGP will consider two Labeled VPN-IPv4 addresses to be equal if they have the same VPN-IPv4 address prefix part, even if they have different Tag values.

### 1.4.11.2 Filtering Based on Attributes

#### 1.4.11.2.1 Site of Origin

The Site of Origin Extended Attribute is a feature that is expected in a future release of Cisco IOS. Once incorporated into IOS, this Attribute will allow a CE router CE1, if associated with the Site of Origin attribute value O1, and if a particular VPN-IPv4 route R1 has this attribute; then R1 must not be distributed to CE1. Furthermore, it must not be placed into the per-VPN forwarding table associated with CE1.

Since all CE routers at the same site must be associated with the same Site of Origin attribute, this ensures that no route from a particular site is ever advertised back into the site. It also ensures that a packet will never be sent from a site to the P network, and then directly back into the site via a different CE router.

#### 1.4.11.2.2 VPN of Origin

The Cisco IOS MPLS-VPN software does not currently have support for the VPN of Origin Extended Attribute. It may be a future implementation that will facilitate the distribution of routes to a particular CE router, but only if those routes originated in a particular VPN or set of VPNs. For example, a particular CE may be allowed to accept only intranet routes.

Alternatively, it may be desired to prevent the distribution of routes to a particular CE if those routes originated in a particular VPN or set of VPNs.

The presence of the VPN of Origin attribute allows one to configure a PE router to only distribute routes to a particular CE router, and only places those routes in a particular VRF, if those routes originated from a particular set of VPNs.

Alternatively, it allows one to configure a PE router so that it never distributes routes to a particular CE, or places routes in a particular per-VPN forwarding table, if those routes originated from a particular set of VPNs.

#### 1.4.11.2.3 Target VPN/Route Target

Every CE router is associated with one or more Target VPN attributes. Each of the per-VPN forwarding tables maintained by the PE is also associated with one or more Target VPN attributes. The PE must be configured to know which of its attached CEs, and which of its per-VPN forwarding tables, is associated with which of these attributes.

When a PE learns a VPN-IPv4 route from another PE router or Route Reflector, it must determine whether that route needs to be placed in a particular per-VPN forwarding table. This will be done if, and only if, the per-VPN forwarding table and the route have a Target VPN attribute in common.

When a PE learns a VPN-IPv4 route from an IBGP peer, it must determine whether that route needs to be redistributed to a particular CE router (after being translated to an IPv4 address). This will be done if, and only if, the CE router and the route have a Target VPN attribute in common.

### 1.4.12 BGP Amongst PE Routers

Each PE router, before distributing a route, will also assign a label for that route. Using the Multi-protocol Extensions of BGP, the route will be distributed as a route to an address in the MPLS-VPN-IPv4 address family, and both the label and the VRF will be encoded as part of the NLRI.

When a PE router uses BGP to redistribute a route received from a CE router, the PE router will always specify its own address as the value of the BGP Next Hop Attribute to ensure that the next hop is always reachable in the P network's IGP (i.e., it doesn't require routes to all the CE routers to be injected into the P network's IGP). It is also necessary to ensure proper interpretation of the label which has been assigned to the route; the label associated with a route must be a label assigned by the corresponding next hop.

For the purpose of supporting VPNs, PE routers, and other P routers which are BGP peers of PE routers, need to support the following capabilities:

- Label distribution via BGP, so that labels can be assigned to the VPN-IPv4 addresses.
- Label distribution via LDP, so that label switching can be used to transport packets from one PE router to another.

- MPLS-VPN-IPv4 Address Family, so that labeled VPN-IPv4 addresses can be distributed amongst PE routers.
- BGP Capability Negotiation should be used to determine whether a BGP peer has the appropriate capabilities.

#### 1.4.12.1 Ordinary BGP Routes

PE routers may also have “ordinary” EBGp and IBGP connections which have nothing to do with VPNs. On such ordinary connections, IPv4 routes, rather than VPN-IPv4 routes, are distributed; routes learned from CE routers will not be sent on such connections, unless those routes are to be exported to the public Internet.

#### 1.4.12.2 Internet Filtering

Any router with a BGP connection to the Internet must ensure, through proper filtering, that it doesn't leak any routes to the Internet that are not part of the P network's AS, or of the AS of some client network of the P network. When routes are leaked to the Internet, all private AS numbers must be removed (via outbound filtering) from the AS-path.

VPN-IPv4 routes will not be accepted from connections to the internet.

#### 1.4.12.3 Route Aggregation

It is possible for a PE router that distributes Labeled VPN-IPv4 routes to do a certain amount of route aggregation. Two VPN-IPv4 routes with the same RD may be aggregated, and the aggregate may be distributed as a Labeled VPN-IPv4 route to BGP peers.

Then when a packet arrives with a particular incoming label, and the label corresponds to an aggregated VPN-IPv4 route, the P router will not be able to choose the next hop based solely on the label. However, the label will identify the RD. The P router must pre-pend the RD to the IP destination address of the packet, in order to get the packet's VPN-IPv4 route. The VPN-IPv4 address can then be looked up in the routing table.

This aggregation procedure may be useful whenever □ a single PE router needs to be able to route to a very large number of VPNs. If each VPN is assigned a single RD, then the PE router will need as little as one route per VPN, rather than one route per address prefix. The BGP next hop will be a P

router which has the full routing table for the VPN, and which can choose the route to the egress PE router.

### 1.4.13 Security

Under the following conditions:

- labeled packets are not accepted by backbone routers from untrusted or unreliable sources, unless it is known that such packets will leave the backbone before the IP header or any labels lower in the stack will be inspected, and
- labeled VPN-IPv4 routes are not accepted from untrusted or unreliable sources.

The security provided by this architecture is identical to that provided by Frame Relay- or ATM-based VPNs.

It is also possible for a VPN user to provide themselves with enhanced security by making use of IPSec.

#### 1.4.13.1 Cisco's Support of IPSec on CEs Today

Currently, one has to manually configure the crypto-maps in each of the CE routers. In that mode, MPLS-VPN packet delivery and IPSec encryption are independent. In order to manually configure the crypto-maps, you has to have a priori knowledge of the next CE router that a given packet will traverse, and hence static routing needs to be introduced for that IPSec tunnel destination.

#### 1.4.13.2 IPSec Work in Progress

Work is under way to dynamically develop the route to the next CE – precisely the idea behind routing in the first place!

RFC 2547<sup>56</sup> considers a feature in which a PE uses BGP to tell a CE, say, CE1, what the next hop CE, say, CE2, is for a particular address prefix. Then this information from routing can be used by CE1 to choose the IPSEC tunnel that leads to CE2. Cisco has not implemented this feature.

<sup>56</sup> Section 9, "Security."

The Service Providers that are interested in MPLS-VPN are supportive of Cisco's IPSEC/MPLS-VPN integration story, as they think that IPSEC/MPLS-VPN integration may be needed in the future, but realistically don't know exactly what the requirements are.

#### 1.4.14 MPLS VPN Functional Summary

Up until today, all VPN solutions have used some kind of tunneling or overlay technology, be they VCs, L2TP, L2F, GRE or IP set tunnels. These technologies are all based on creating overlays, not networks. These solutions are inherently unscalable as the number of VCs provisioned grows exponentially as the number of sites in those VPNs grows. In addition, each end customer requires a new network to be engineered and managed. So the ability to roll out these solutions and support these various VPNs is not scalable.

In contrast, MPLS-based VPNs are based on the peer model, not the overlay paradigm. A VPN should be thought of as a set of inter-connected sites. An MPLS-VPN consists of a set of sites that are inter-connected via an MPLS or label switching Provider core. At each site, there are one or more CE routers, which attach (via a data link of some sort) to one or more PE routers. PE routers dynamically communicate to one another utilizing IBGP. It is not required that the set of IPv4 addresses used in any two VPNs be mutually exclusive, as the PEs translate IPv4 addressees into IPv4VPN entities, utilizing BGP with extended community attributes.

The set of addresses used in a VPN must be exclusive of the set of addresses used in the P network. More specifically, it is required that every CE router be able to address the PE router(s) to which it is directly attached. This means that the addresses of the PE routers must not be duplicated in any VPN.

### 1.5 MPLS-VPN Configuration

#### 1.5.1 Summary of MPLS-VPN Configuration Steps

- Define VRF(s)
- Define RD for Each VRF Instance
- Configure Import and Export VPN route-target lists
- Assign interfaces to VRFs

- Configure Routing Protocol Families
- Configure IBGP for distribution of VPN routes

#### 1.5.2 MPLS-VPN Configuration Entities

In this section, we highlight the concepts that IOS relies upon in order to configure MPLS-VPN on a PE router. We point to VRFs, Router Address Family, VPN-IPv4 NLRIs<sup>57</sup>, and Route-target Communities<sup>58</sup>.

##### 1.5.2.1 VRF instances

At the heart of MPLS-VPN functionality are VPN Routing/Forwarding (VRF) instances:

A VRF consists of:

- An IP Routing table. This is directly analogous to the single central routing table in previous versions of IOS. In fact, the central routing table can be considered to be the VRF routing table for the provider backbone.
- A derived forwarding table, based on the CEF (FIB) □ forwarding technology.
- A set of interfaces which use the derived forwarding table.
- Rules which control route injection into and out of the VRF routing table.
- A set of routing protocols and routing peers which inject information into the VRF routing table.
- Router variables associated with the VRF routing instance.

In this document "per-site forwarding table" will be used interchangeably with VRF table.

##### 1.5.2.1.1 IOS Configuration Command for a VRF Instance

```
(config)# ip vrf vrf_name
```

For example, "ip vrf CustomerA" will initiate a VPN Routing table, and associated CEF table, named CustomerA. It then enters vrf configuration sub-mode to configure variables associated with the VRF.

<sup>57</sup> Network Layer Reachability Information. NLRIs were introduced with BGP, consisting of a prefix and length, e.g., 47/8 (network 47.0.0.0/One Octet); 204.10.1/24, etc.

<sup>58</sup> Extended BGP communities.

### 1.5.2.1.2 VRF Configuration Sub-mode

The router then changes the config-mode prompt to VRF configuration sub-mode:

```
(config-vrf)#RD RD_Value
```

This is where one enters the 8-byte Route Descriptor that will be pre-pended to the IPv4 routes by the PE, prior to re-distributing the route into the MPLS-VPN backbone.

In that vrf sub-mode, one also enters the route-target information for that VRF. Route targets are explained in detail below.

```
(config-vrf)#route-target import | export | both
<community>
```

### 1.5.2.2 Router Address Family

As mentioned earlier in this document, one of the building blocks of MPLS-VPN are the Multi-protocol Extensions to BGP. Address Families were introduced in order to support the multi-protocol extensions to BGP-4.

#### 1.5.2.2.1 Backwards Compatibility

##### 1.5.2.2.1.1 Parser Level

The Cisco IOS CLI command parser will accept BGP commands in either the (standard, pre-MPLS-VPN) router protocol configuration mode; or in the new address-family mode for 'ipv4 unicast.' So all BGP configuration commands that are supported in previous versions of Cisco IOS are valid for address-family IPv4 unicast.

##### 1.5.2.2.1.2 BGP Capabilities

When an MPLS-BGP router is attempting to establish a TCP peering relationship with another BGP speaker, it will revert to "standard" BGP-4, if the other end does not support the RFC2283 Multi-protocol Extensions. This is done through the Capabilities Exchange between them.

An MPLS-VPN BGP speaker will be able to apply EBGp exchanges in a VRF context with one TCP peer, while speaking standard BGP-4 with another peer, even on the same interface. In fact, two TCP peers can maintain an IPv4 classical BGP-4 peering relationship, as well as a VPNv4 connection.

##### 1.5.2.2.1.3 IPv4 NLRI Caveat

One must enter the "no bgp default ipv4-activate" command<sup>59</sup> under the definition of the EBGp CE neighbor, while one utilizes the "neighbor <neighbor-ip-address> activate" with that same CE neighbor, in the VRF configuration context. A BGP session that will be used for the exchange of IPv4 NLRIs with that VRF (CE) neighbor must not be activated for the exchange of IPv4 NLRIs with any non-VRF neighbor (i.e., for Internet connectivity). An alternative to the "no bgp default ipv4-activate" command is the issuance of the "no neighbor <CE-neighbor-ip-address> activate" configuration command under the configuration of that CE peer<sup>60</sup>.

#### 1.5.2.2.2 Address Family Components

An address family consists of a main family (also referred to as an Address Family Identifier<sup>61</sup>, or AFI) and a modifier, or SubAddress-Family-Identifier (SUBAFI).

#### 1.5.2.2.3 Address Family Configuration

Configuring a routing protocol that supports multiple address families consists of the following steps:

1. Configuring global variables for the routing protocol
2. Configuring variables specific to each address family
3. Defining and activating sessions with other routing peers carrying this address family

The address families supported vary with the routing protocol. For example, the current<sup>62</sup> BGP code supports the following address families:

<sup>59</sup> That same "activate" command can be utilized in a different context, whereby two routers are utilizing the same TCP peering session to transact two different BGP exchanges - an IBGP VPNv4 routing exchange, as well as an EBGp IPv4 Public Internet routing exchange. This will make sense in environments where reducing the TCP peering overhead is critical on a particular router.

<sup>60</sup> It turns out that the Cisco IOS CLI provides a multitude of ways to configure the routing-protocol configuration aspects of MPLS-VPN. Another example is the ability to configure VRF-specific commands from the vrf sub-mode context, as well as from the global configuration context (i.e., "ip vrf VPN-Example rd 1000:2", versus "ip vrf VPN-Example" and then "rd 1000:2"). The reader can also refer to the configuration examples below for further "flexibility."

<sup>61</sup> RFC2283

<sup>62</sup> That is, the BGP code that already supports MPLS-VPNs.

1. IPv4 unicast<sup>63</sup>
2. VPN-IPv4 unicast

#### 1.5.2.2.4 Address Family Usage

The VPN-IPv4 concept can be used in one of two ways:

1. Distinguish amongst different customer subnets that have non-unique IPv4 addresses.
2. Set up multiple (non-comparable) routes to a single customer subnet or network, while utilizing policy to decide which packets follow which route. This will allow, for example, firewall interception for packets under one policy, versus bypassing the router for packets coming from certain hosts.

#### 1.5.2.3 VPN-IPv4 NLRIs

The prefix portion of the VPN-IPv4 NLRI consists of:

1. an 8-byte value known as a Route Distinguisher, or RD and
2. an IPv4 prefix (32 bits, and a mask length).

The VPNv4 prefix is created by pre-pending the RD to the VPN IPv4 prefix. Therefore, even if two or more customers were using non-unique IP addresses, MPLS-VPN will still be able to distinguish amongst the different customer destinations, as long as a unique RD value is chosen for each customer.

Each RD is a BGP-4 Extended Community Attribute comprised of a:

1. 2-byte Type Field, and a
2. 6-byte Value Field.

An RD can be entered in two formats, depending on the setting of the high-order byte (HOB) setting of the Type Field to either a zero or one.

- Type HOB set to 0x00 provides us with a 16 bit ASN<sup>64</sup>, followed by an arbitrary 32 bit number, for example 551:1000
- Type HOB set to 0x01 generates a 32 bit IPv4 address, followed by an arbitrary 16 bit number, for instance, 131.108.14.3:1000

#### 1.5.2.4 Route Target (RT) Communities

The mechanism by which BGP/MPLS VPN controls distribution of VPN routing information is through the use of VPN route-target extended BGP communities. An extended BGP community<sup>65</sup> is an eight-octet structured value, as stated above.

A BGP speaker may use the “Route Origin” and “Route Target” community attributes to control which routing information it accepts, prefers or distributes to its peers.

BGP/MPLS VPN uses VPN route-target communities as follows:

- When a VPN route is injected into BGP, it is associated with a list of VPN route-target communities. Typically this is set through an export list of community values associated with the VRF from which the route was learned.

- Associated with each VRF is an import list of route-target communities. This list defines the values that should be matched against to decide whether a route is eligible to be imported into this VPN routing instance. For example, if the import list for a particular VRF is {A, B, C}, then any VPN route that carries community value A, or B, or C will be imported into the VRF.

For examples of how these are used, consider the scenarios in the following sections.

##### 1.5.2.4.1 CUG VPN

The simplest scenario for a VPN is a closed user group (CUG) of sites. Every site can communicate directly with every other site. To support this scenario we need to do the following:

- define a single VPN route-target community, call it Cclosed
- Set the export list of each VRF connected to a site in the VPN to contain only Cclosed
- Set the import list of each VRF connected to a site in the VPN to contain only Cclosed

<sup>63</sup> That is, good-old BGP-4 that has been supported since Cisco IOS 10.0

<sup>64</sup> Autonomous System Number.

<sup>65</sup> Please refer to draft-ramachandra-bgp-ext-communities-00.txt for details.

Thus, every route exported into IBGP from one of these VRFs will contain Cclosed. Every route received that contains C closed will be eligible for import into these VRFs.

#### 1.5.2.4.2 Hub-and-Spokes VPN

This is a more complex VPN. In this case we have one central site, the “hub”, and a set of “spoke” sites. Spokes cannot communicate directly with each other. Instead, all traffic between “spoke” sites will be routed through the “hub.” The “hub” site can communicate directly to all “spokes.” Note: the current code does not support this configuration optimally. The current code requires that routes exported from a site be present in that site’s VRF, otherwise MPLS forwarding information is not created correctly<sup>66</sup>. Therefore, with the current code, additional route-target communities must be defined in order to cause the spokes routes to be imported into their own VRFs. This example documents the optimal configuration, which will be supported in the future.

To implement this scenario we do the following:

- Define two VPN route-target communities: C\_hub and C\_spoke .
- Set the export list of each VRF connected to a “spoke” site to C\_spoke
- Set the export list of the “hub” site to C\_hub
- Set the import list of each VRF connected to a “spoke” site to C\_hub
- Set the import list of the “hub” site to C\_spoke

Thus, every route exported into IBGP from a “spoke” site is imported only into the VRF for the “hub” site.

Conversely, routes exported from the “hub” site are imported into the VRFs for each “spoke” site.

This scenario can be generalized to multiple hub sites. Through the definition of additional route-target communities it can be generalized to support arbitrary overlap between the “spoke” sites.

<sup>66</sup> In the current implementation (IOS 12.0(5)T) , for a route to be imported into a VRF, the extended community attributes for the route must have a non-null intersection with the import extended community attributes associated with the VRF, regardless of how that route is learned (i.e., via IBGP from a PE or RR, EBGp from a CE, RIP from a CE, static configuration on the PE). The “export” extended community attributes associated with a VRF are added as attributes to routes learned from the VRF CE (via EBGp, RIP or static configuration).

#### 1.5.2.4.3 Controlled Access to Servers

As a third scenario, let us consider a case where a Service Provider wishes to control the access to a set of content servers. For example, assume there are three sites A, B, and C, and three servers S1, S2, and S3. Sites have subscribed to differing sets of content. In particular, A and B are allowed to receive data from S1; B and C from S2; and only C from S3. A, B, and C may or may not be allowed to communicate directly.

To implement this scenario, one does the following:

- Define six Route Target Extended Community Attributes: C\_S1Imp, C\_S1Exp, C\_S2Imp, C\_S2Exp, C\_S3Imp, C\_S3Exp.
- Routes from server S1 should be exported using community C\_S1Exp. The VRF connected to server S1 should be configured to import routes with community C\_S1Imp.
- Routes S2 export with community C\_S2Exp. The S2 VRF imports routes with community C\_S2Imp.
- Similarly for the S3 server.
- Routes for site A should be exported with community C\_S1Imp.
- Routes for site B should be exported with communities C\_S1Imp and C\_S2Imp.
- Routes for site C should be exported with communities C\_S2Imp and C\_S3Imp.
- The VRF connected to site A should import routes with community C\_S1Exp.
- The VRF connected to site B should import routes with communities C\_S1Exp and C\_S2Exp.
- The VRF connected to site C should import routes with communities C\_S2Exp and C\_S3Exp.

### 1.5.3 MPLS-VPN Configuration, Next Steps:

So one first configures the name of the VPN(s), or VRF(s) to handle. Next, one defines the 64-bit Route Distinguisher value for this VRF. One then sets up the route-target lists, which is a community that will be matched in order to cause a route to be

imported into this VRF. One also can define the export route target community, which is the community that shall be added on all routes exported into the backbone BGP from this VRF.

Next, while still in configuration mode, one enters the command:

```
ip vrf forwarding <vpn_name67>
```

This command will attach an interface to a VRF. This will cause all packets forwarded as well as all packets received on this interface to be transacted using the forwarding table for this VRF, and all routing protocols which use this interface to pick the appropriate context.

Continuing on, one enters the following configuration commands:

```
address-family ipv4 vrf <name>
```

```
<router configuration commands>
```

```
neighbor <address> activate
```

Other routing-protocol-specific configuration commands...

Router ‘address-family’ sub-mode is used to configure BGP across the provider backbone for transport of addresses other than IPv4. In order to configure routing protocols, Cisco IOS has an address family command. One can use address family IPv4, in a context of a particular VRF. Once in the sub-mode for this address family, one can use existing (and familiar) routing commands, such as “neighbor”, “offset list”, “redistribute”, and the commands will apply to the group context for this VRF.

Let us go through a few examples, to allow the reader to better understand the configuration steps required for MPLS-VPN functionality.

---

<sup>67</sup> or vrf\_name

### 1.5.4 Global-level versus sub-command VRF Commands

It appears that the “ip vrf” global-level configuration commands are either going to be removed from the parser in the shipping versions of Cisco IOS, or that those commands will become hidden. The same functions can be enabled in the vrf sub-command mode (i.e., one will issue the configuration command “ip vrf vrf-name” and then from that vrf mode, enter the appropriate commands).

#### 1.5.5 First Configuration Example

!

! Remember. CEF switching is a prerequisite for MPLS and MPLS-VPN...

!

ip cef distributed

!

! Define two VPN Routing instances, named ‘CustomerA’ and ‘CustomerB’

!

ip vrf CustomerA rd 100:1

ip vrf CustomerB rd 100:2

!

! Configure the import and export VPN route-target list for each VRF

!

ip vrf CustomerA route-target both 100:1

ip vrf CustomerB route-target both 100:2

ip vrf CustomerB route-target import 100:1

!

! Note that one can arrive at the same result of specifying the route-target

! extended community string in the last five configuration commands,

! through entering vrf config sub-mode, as follows:

!

! ip vrf CustomerA

! rd 100:1

!

! Import into and export out of VRF/VPN the NLRIs that have the extended

! route-target community value 100:1

!

! route-target both 100:1

!

! ip vrf CustomerB

! rd 100:2

! route-target both 100:2

! route-target import 100:1

!

!

! Configure an import route-map for CustomerB

!

ip vrf CustomerB import map CustomerB\_import

!

! ‘CustomerB’ should not install PE-CE addresses in the

! “global routing table.” So another site in this VPN can have

! a similar address on its CE end. If the following command is

! not issued, then all CE routers in a particular VPN must have

! unique addresses.

!

no ip vrf CustomerB global-connected-addresses

!

interface loopback0

```

ip address 10.13.0.13 255.255.255.255
!
! Set up an Ethernet interface as a VRF link to a CE
router
!
interface Ethernet5/0/1
!
! Accept CustomerA's traffic into and out of E5/0/1
!
ip vrf forwarding CustomerA
ip address 10.20.0.13 255.255.255.0
!
! Set up a Frame-Relay PVC sub-interface a link to
another CE router
!
interface hssi 10/1/0
encapsulation frame-relay
frame-relay lmi-type ansi
!
interface hssi 10/1/0.16 point-to-point
!
! CustomerB's traffic is expected on this interface...
!
ip vrf forwarding CustomerB
ip address 10.20.1.13 255.255.255.0
frame-relay interface-dlci 16
!
! Configure BGP sessions
!
router bgp 1
!
! Define an IBGP session with another PE
!
neighbor 10.15.0.15 remote-as 1
!
neighbor 10.15.0.15 update-source lo0
no synchronization
!
! Define some VRF (CE) sessions.
!
neighbor 10.20.1.11 remote-as 65535
neighbor 10.20.1.11 update-source
h10/1/0.16
!
! Deactivate the default IPv4 session.
! This was discussed earlier. A PE router
communicating with a CE peer
! via EBGP must be told explicitly to not
speak standard IPv4 BGP, but
! rather to use IPv4 in a VRF context.
!
no neighbor 10.20.1.11 activate
!
neighbor 10.20.0.60 remote-as 65535
neighbor 10.20.0.60 update-source e5/0/1
no neighbor 10.20.0.60 activate
!
! Activate PE peer for exchange of VPNv4
NLRIs
!
address-family vpnv4 unicast
neighbor 10.15.0.15 activate
exit-address-family
!
! Define router variables for VRFs. That is,
in this case,
! define BGP parameters for PE-CE
sessions...

```

```

! Activate sessions to peers within those VRFs
!
! Remember, we're speaking good-old ebgp, using
IPv4 protocols, albeit

! in a VRF context, between the CE and PE, if that's
the routing protocol

! we choose.
!
address-family ipv4 unicast vrf CustomerA
neighbor 10.20.0.60 activate
no auto-summary
redistribute static
exit-address-family

address-family ipv4 unicast vrf CustomerB
neighbor 10.20.1.11 activate
no auto-summary
redistribute static
exit-address-family
!
! Define a VRF static route.

! Currently, static VRF routes must point out an
interface, versus

! point to a next hop IP address.
!
ip route vrf CustomerA 12.0.0.0 255.0.0.0 e5/0/1

```

### 1.5.6 Second MPLS VPN Configuration Example

```

Router(config)# ip vrf RED
Router(config-vrf)# rd 102:1234
Router(config-vrf)# route-target both 102:1234
Router(config-vrf)# no global-connected-addresses

Router(config)# interface E5/0/2

```

```

Router(config-if)# ip vrf forwarding RED

Router(config-if)# ip address 10.1.1.1
255.255.255.0

```

```

Router(config-if)# interface Hssi10/1/0

Router(config-if)# ip vrf forwarding RED

Router(config-if)# ip unnumbered lo100

```

First the VRF is defined, in this case VPN “RED”<sup>68</sup>. Next the VPN “RED” is given an RD. This is followed by the definition of the route target. This is a simple VPN. It has the same RD and import/export (and hence the keyword both) route target. Specifying the same value for RD and route targets is a useful convention in very simple VPNs. The no global-connected-addresses command controls whether or not the address between the provider router and customer router, shows up in the vrf and global routing tables. The “no” version of the command ensures that the connected address does not get carried into the IBGP routing table and get re-distributed to other PE peers in that VPN. This will make it possible to have non-unique local CE addresses in the same VPN.

The next configuration command sets up interfaces to be connected to this VRF. One can apply an IP address to the interface participating in VPN, or one can use unnumbered IP addressing. If one chooses to support the latter, then it needs to be unnumbered on an interface that is also attached to this VRF.

```

Router# show ip vrf

Name      Default RD   Interfaces
RED       102:1234    Ethernet5/0/2
          Hssi10/1/0

```

The above EXEC command indicates that the VRF “red” has a route distinguisher and two interfaces connected to it.

Next, one performs BGP configuration:

<sup>68</sup> m IOS's perspective, VRFs are case-sensitive.

```

Router(config)# router bgp 1
!
! Let us configure a PE peer...
!
! Remember, PEs speak IBGP amongst them.
!
Router(config-router)# neighbor 192.168.72.1
remote-as 1
Router(config-router)# neighbor 192.168.72.1
update-source lo0
!
! Next, we set up a CE neighbor.
! SP and customer decided that it makes sense to
speak EBGP between them.
!
Router(config-router)# neighbor 10.100.1.2 remote-
as 65535
Router(config-router)# neighbor 10.100.1.2 update-
source lo100
!
! Let's make sure we don't speak standard IPv4
unicase BGP-4 between us (the PE) and the CE,
! but rather IPv4 in a VRF context, as we shall see
in the commands following that...
!
Router(config-router)# no neighbor 10.100.1.2
activate
!
! So switching to IPv4, but in a VRF context...
!
Router(config-router)# address-family ipv4 vrf RED
Router(config-router-af)# neighbor 10.100.1.2
activate
Router(config-router-af)# redistribute static
!

```

If instead of EBGP, we used RIP ii, then:

```

!
Router(config)# router rip
Router(config-router)# address-family ipv4
vrf RED
Router(config-router-af)# version 2
Router(config-router-af)# network 10.0.0.0
! Moving on to the VPN-IPv4 config
portion of our PE peer...
!
Router(config-router)# address-family
vpnv4
Router(config-router-af)# neighbor
192.168.72.1 activate
!
! Customer told the SP that they need to
carry their IGP routes to other sites in the
VPN...
!
Router(config-router-af)# redistribute rip

```

Here, we are speaking RIP II in the context of VRF RED. SP then configures the familiar Cisco IOS RIP-specific commands.

Next, one can look at the routing table in this VRF's context. One will note that it looks very much like the existing Cisco IOS routing table.

```

Router# show ip route vrf RED

```

```

....

```

Gateway of last resort is not set

```

B 2.0.0.0/8 [20/0] via 10.100.1.2, 2d00h,
Hssi10/1/0

```

10.0.0.0/24 is variably subnetted, 4 subnets, 2 masks

B 10.2.2.0/24 [200/0] via 192.168.72.1, 2d00h

C 10.1.1.0/24 is directly connected, Ethernet5/0/2

C 10.100.1.0/24 is directly connected, Loopback100

C 10.100.1.2/32 is directly connected Hssi10/1/16

12.0.0.0/32 is subnetted, 1 subnets

R 12.0.0.1 [120/1] via 10.1.1.2, 00:00:05, Ethernet5/0/2

The reader will note that NLRI 2/8 was obtained from our CE neighbor on the HSSI interface (since the administrative distance [AD] for this route is 20. We are, of course, assuming default setting for AD.)

One can also note the 10.2.2/24 NLRI obtained from a PE peer, for a customer subnet at another site (note the AD value of 200).

The reader will also note the RIP route, which was also learned from a customer router.

Router#show ip bgp vpnv4 vrf RED

BGP table version is 11, local router ID is 10.13.0.13

Status codes: s suppressed, d damped, h history, \* valid,

> best, i - internal

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
---------	----------	--------	--------	--------	------

Route Distinguisher: 102:1234 (RED)

\*> 2.0.0.0 10.100.1.2 0 0 65535 ?

\*>i12.0.0.0 10.1.1.2 0 0 ?

\*>i10.2.2.0/24 192.168.72.1 0 100 0 ?

\*>i10.1.1.0/24 0.0.0.0 0 32768 ?

\*>i10.100.1.0/24 0.0.0.0

Router# show ip bgp vpnv4 vrf RED tags

Network	Next Hop	In tag/Out tag
---------	----------	----------------

Route Distinguisher: 102:1234 (RED)

2.0.0.0 10.100.1.2 36/notag

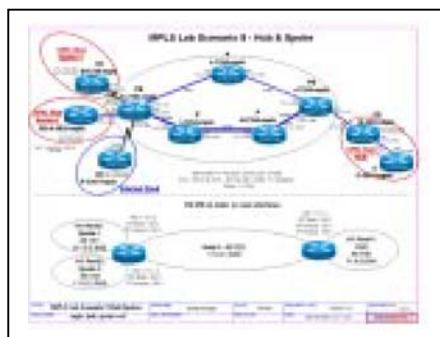
10.1.1.0/24 0.0.0.0 35/aggregate(RED)

10.2.2.0/24 192.168.72.1 notag/27

12.0.0.0 10.1.1.1 34/notag

The command above lets the Service Provider see the labeling information that was learned or distributed. One can see a connected route which is 10.1.1.0. Since this is a connected route, it has a distributed aggregate label of 35.

## 1.57 Third Configuration Example - Hub-and-Spokes<sup>69</sup>



### 1.5.7.1 Configuration from CE: A-3620-mpls

```
!
version 12.0
!
hostname a-3620-mpls
!
interface Loopback0
ip address 222.2.3.1 255.255.255.255
no ip directed-broadcast 0 32768 ?
!
interface Loopback1
```

<sup>69</sup> From Robert Raszuk's NSA MPLS pages.

```
ip address 111.0.3.1 255.255.255.255
no ip directed-broadcast
!
interface Loopback2
ip address 111.0.3.2 255.255.255.255
no ip directed-broadcast
!
interface Ethernet0/0
ip address 172.16.69.34 255.255.255.224
no ip directed-broadcast
!
interface Ethernet0/2
ip address 111.0.1.106 255.255.255.252
no ip directed-broadcast
!
ip classless
ip route 0.0.0.0 0.0.0.0 111.0.1.105
ip route 171.0.0.0 255.0.0.0 172.16.69.33
!
end
```

### 1.5.7.2 Configuration from CE: B-3620-mpls

```
!
version 12.0
!
hostname b-3620-mpls
!
interface Loopback0
ip address 222.2.5.1 255.255.255.255
no ip directed-broadcast
!
```

```
interface Loopback1
ip address 111.0.5.1 255.255.255.255
no ip directed-broadcast
!
interface Loopback2
ip address 111.0.5.2 255.255.255.255
no ip directed-broadcast
!
interface Ethernet0/0
ip address 172.16.69.35 255.255.255.224
no ip directed-broadcast
!
interface Ethernet0/1
ip address 111.0.5.9 255.255.255.252
no ip directed-broadcast
!
interface Ethernet0/2
ip address 111.0.1.118 255.255.255.252
no ip directed-broadcast
!
router ospf 100
passive-interface Ethernet0/2
network 111.0.5.0 0.0.0.255 area 0
network 222.2.5.1 0.0.0.0 area 0
default-information originate
!
ip classless
ip route 0.0.0.0 0.0.0.0 111.0.1.117
ip route 171.0.0.0 255.0.0.0 172.16.69.33
!
end
```

### 1.5.7.3 Configuration from: C-2611-mpls

```
!  
version 12.0  
!  
hostname c-2611-mpls  
!  
interface Loopback0  
ip address 222.2.5.2 255.255.255.255  
no ip directed-broadcast  
!  
interface Ethernet0/0  
ip address 172.16.69.36 255.255.255.224  
no ip directed-broadcast  
!  
interface Ethernet1/0  
ip address 111.0.5.10 255.255.255.252  
no ip directed-broadcast  
!  
router ospf 100  
network 111.0.5.0 0.0.0.255 area 0  
network 222.2.5.2 0.0.0.0 area 0  
!  
ip classless  
ip route 171.0.0.0 255.0.0.0 172.16.69.33  
!  
end
```

### 1.5.7.4 Configuration from: D-1720-mpls

```
!  
version 12.0  
!
```

```
hostname d-1720-mpls  
!  
interface Loopback0  
ip address 222.2.2.1 255.255.255.255  
no ip directed-broadcast  
!  
interface Loopback1  
ip address 111.0.2.1 255.255.255.255  
no ip directed-broadcast  
!  
interface Loopback2  
ip address 111.0.2.2 255.255.255.255  
no ip directed-broadcast  
!  
interface Loopback3  
ip address 20.1.1.1 255.0.0.0  
no ip directed-broadcast  
!  
interface Serial0  
ip address 111.0.1.102 255.255.255.252  
no ip directed-broadcast  
!  
interface FastEthernet0  
ip address 172.16.69.37 255.255.255.224  
no ip directed-broadcast  
!  
ip classless  
ip route 0.0.0.0 0.0.0.0 111.0.1.101  
ip route 171.0.0.0 255.0.0.0 172.16.69.33  
!  
end
```

### 1.5.7.5 Configuration from: E-1720-mpls

```

!
version 12.0
!
hostname e-1720-mpls
!
interface Loopback0
 ip address 222.2.4.1 255.255.255.255
 no ip directed-broadcast
!
interface Loopback1
 ip address 111.0.4.1 255.255.255.255
 no ip directed-broadcast
!
interface Loopback2
 ip address 111.0.4.2 255.255.255.255
 no ip directed-broadcast
!
interface Serial0
 ip address 111.0.1.114 255.255.255.252
 no ip directed-broadcast
 clockrate 4000000
!
interface FastEthernet0
 ip address 172.16.69.38 255.255.255.224
 no ip directed-broadcast
!
router bgp 64700
 no synchronization
 network 111.0.4.1 mask 255.255.255.255
 network 111.0.4.2 mask 255.255.255.255
 network 222.2.4.1 mask 255.255.255.255

```

```

network 222.2.4.1
neighbor 111.0.1.113 remote-as 222
 no auto-summary
!
ip classless
ip route 171.0.0.0 255.0.0.0 172.16.69.33
!
end

```

### 1.5.7.6 Configuration from: H-7204-mpls

```

!
version 12.0
!
hostname h-7204-mpls
!
ip cef
!
interface Loopback0
 ip address 222.2.1.5 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet1/0
 ip address 172.16.69.41 255.255.255.224
 no ip directed-broadcast
!
interface FastEthernet2/0
 ip address 111.0.1.13 255.255.255.252
 no ip directed-broadcast
 tag-switching ip
!
interface FastEthernet2/1
 ip address 111.0.1.10 255.255.255.252

```

```

no ip directed-broadcast
tag-switching ip
!
router ospf 222
 network 111.0.1.0 0.0.0.255 area 0
 network 222.2.1.5 0.0.0.0 area 0
!
ip classless
ip route 171.0.0.0 255.0.0.0 172.16.69.33
!
access-list 110 deny ip any 224.0.0.0
15.255.255.255
!
end

```

#### 1.5.7.7 Configuration from: I-7204-mpls

```

!
version 12.0
!
hostname i-7204-mpls
!
ip cef
!
interface Loopback0
 ip address 222.2.1.4 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet1/0
 ip address 172.16.69.42 255.255.255.224
 no ip directed-broadcast
!
interface FastEthernet2/0
 ip address 111.0.1.6 255.255.255.252

```

```

no ip directed-broadcast
tag-switching ip
!
interface FastEthernet2/1
 ip address 111.0.1.9 255.255.255.252
 no ip directed-broadcast
tag-switching ip
!
router ospf 222
 network 111.0.1.0 0.0.0.255 area 0
 network 222.2.1.4 0.0.0.0 area 0
!
ip classless
ip route 171.0.0.0 255.0.0.0 172.16.69.33
!
end

```

#### 1.5.7.8 Configuration from: J-7204-mpls

```

!
version 12.0
!
hostname j-7204-mpls
!
ip cef
!
ip vrf RedH1
 rd 111:1
 route-target export 100:1
 route-target import 101:1
 route-target import 102:1
 route-target import 100:1
!

```

```

interface Loopback0
ip address 222.2.1.2 255.255.255.255
no ip directed-broadcast
!
interface Ethernet1/0
ip address 172.16.69.43 255.255.255.224
no ip directed-broadcast
!
interface Ethernet1/1
ip address 111.0.1.18 255.255.255.252
no ip directed-broadcast
tag-switching ip
!
interface Ethernet1/3
ip vrf forwarding RedH1
ip address 111.0.1.117 255.255.255.252
no ip directed-broadcast
!
interface FastEthernet2/0
ip address 111.0.1.14 255.255.255.252
no ip directed-broadcast
tag-switching ip
!
interface Serial4/0
ip address 111.0.1.113 255.255.255.252
no ip directed-broadcast
!
router ospf 222
passive-interface Ethernet1/3
passive-interface Serial4/0
network 111.0.1.0 0.0.0.255 area 0
network 222.2.1.2 0.0.0.0 area 0

!
router bgp 222
no synchronization
no bgp default ipv4-unicast
neighbor 222.2.1.1 remote-as 222
neighbor 222.2.1.1 update-source
Loopback0
!
address-family ipv4 vrf RedH1
redistribute connected
redistribute static
no auto-summary
no synchronization
exit-address-family
!
address-family vpnv4
neighbor 222.2.1.1 activate
neighbor 222.2.1.1 send-community
extended
exit-address-family
!
ip classless
ip route 171.0.0.0 255.0.0.0 172.16.69.33
ip route vrf RedH1 111.0.5.1
255.255.255.255 Ethernet1/3 111.0.1.118
ip route vrf RedH1 111.0.5.2
255.255.255.255 Ethernet1/3 111.0.1.118
ip route vrf RedH1 111.0.5.8
255.255.255.252 Ethernet1/3 111.0.1.118
ip route vrf RedH1 222.2.5.1
255.255.255.255 Ethernet1/3 111.0.1.118
ip route vrf RedH1 222.2.5.2
255.255.255.255 Ethernet1/3 111.0.1.118
!
end

```

**1.5.7.9 Configuration from: K-7204-mpls**

```

!
version 12.0
!
hostname k-7204-mpls
!
ip cef
!
ip vrf RedS1
rd 111:2
route-target export 101:1
route-target import 100:1
route-target import 101:1
!
ip vrf RedS2
rd 111:3
route-target export 102:1
route-target import 100:1
route-target import 102:1
!
interface Loopback0
ip address 222.2.1.1 255.255.255.255
no ip directed-broadcast
!
interface Ethernet1/0
ip address 172.16.69.44 255.255.255.224
no ip directed-broadcast
!
interface Ethernet1/1
ip address 111.0.1.1 255.255.255.252
no ip directed-broadcast
tag-switching ip

```

```

!
interface Ethernet1/3
ip vrf forwarding RedS2
ip address 111.0.1.105 255.255.255.252
no ip directed-broadcast
!
interface FastEthernet2/0
ip address 111.0.1.5 255.255.255.252
no ip directed-broadcast
tag-switching ip
!
interface Serial4/0
ip vrf forwarding RedS1
ip address 111.0.1.101 255.255.255.252
no ip directed-broadcast
clockrate 56000
!
router ospf 222
redistribute connected
network 111.0.1.0 0.0.0.255 area 0
network 222.2.1.1 0.0.0.0 area 0
default-metric 25
!
router bgp 222
no synchronization
no bgp default ipv4-unicast
neighbor 222.2.1.2 remote-as 222
neighbor 222.2.1.2 update-source Loopback0
!
address-family ipv4 vrf RedS2
redistribute connected
redistribute static

```

```

no auto-summary
no synchronization
exit-address-family
!
address-family ipv4 vrf RedS1
redistribute connected
redistribute static
no auto-summary
no synchronization
exit-address-family
!
address-family vpv4
neighbor 222.2.1.2 activate
neighbor 222.2.1.2 send-community extended
exit-address-family
!
ip classless
ip route 171.0.0.0 255.0.0.0 172.16.69.33
!
ip route vrf RedS1 111.0.2.1 255.255.255.255
Serial4/0 111.0.1.102
ip route vrf RedS1 111.0.2.2 255.255.255.255
Serial4/0 111.0.1.102
ip route vrf RedS1 222.2.2.1 255.255.255.255
Serial4/0 111.0.1.102
ip route vrf RedS2 111.0.3.1 255.255.255.255
Ethernet1/3 111.0.1.106
ip route vrf RedS2 111.0.3.2 255.255.255.255
Ethernet1/3 111.0.1.106
ip route vrf RedS2 222.2.3.1 255.255.255.255
Ethernet1/3 111.0.1.106
!
end

```

### 1.5.7.10 Configuration from: L-7204-mpls

```

!
version 12.0
!
hostname l-7204-mpls
!
ip cef
!
interface Loopback0
ip address 222.2.1.3 255.255.255.255
no ip directed-broadcast
!
interface Ethernet1/0
ip address 172.16.69.45 255.255.255.224
no ip directed-broadcast
!
interface Ethernet1/2
ip address 111.0.1.2 255.255.255.252
no ip directed-broadcast
tag-switching ip
!
interface Ethernet1/3
ip address 111.0.1.17 255.255.255.252
no ip directed-broadcast
tag-switching ip
!
router ospf 222
network 111.0.1.0 0.0.0.255 area 0
network 222.2.1.3 0.0.0.0 area 0
ip classless
ip route 171.0.0.0 255.0.0.0 172.16.69.33
!

```

end

### 1.5.8 Fourth Configuration Example – Default Routing

Here, the reader will encounter solely the set of routing commands necessary to provide default routing to a particular VPN, “VPN\_Internet.”

```
address-family ipv4 vrf VPN_Internet
```

```
redistribute static
```

```
default-information originate
```

```
exit-address-family
```

```
address-family vpnv4
```

```
neighbor 193.0.10.5 activate
```

```
neighbor 193.0.10.5 next-hop-self
```

```
neighbor 193.0.10.5 send-community extended
```

```
no auto-summary
```

```
network 144.254.82.0 mask 255.255.255.0
```

```
exit-address-family
```

```
!
```

```
ip route vrf VPN_Internet 0.0.0.0 0.0.0.0  
FastEthernet0/0 144.254.82.11 tag 4070
```

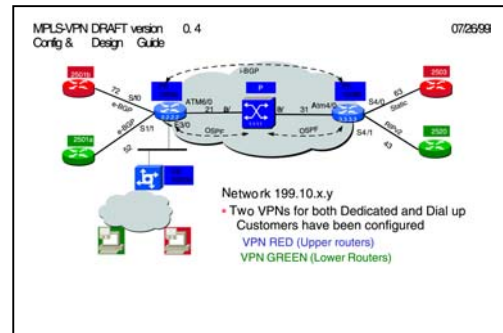
```
ip route vrf VPN_Internet 20.20.20.0 255.255.255.0  
FastEthernet0/0 144.254.82.1
```

The reader is encouraged to refer to the latest MPLS-VPN EFT documentation, for further details on the static VRF command. For example, the capability exists for specifying that the next hop to a static VRF route is to be found in the non-VRF (the router’s “standard”) routing table.

<sup>70</sup> The optional tag 40 arguments in this command pertain to a tag value that can be used for controlling redistribution of routes via route maps. This keyword is not associated with the labels in MPLS.

### 1.5.9 PPP + MPLS-VPN Configurations (Cisco IOS 12.0(5)T)

#### 1.5.9.1 Diagram of PPP + MPLS- VPN European Testing



#### 1.5.9.2 Configuration and Monitoring of PPP + MPLS-VPN/European Testing

```
milab-7206a#sh running-config
```

```
Building configuration...
```

```
Current configuration:
```

```
!
```

```
version 12.0
```

```
!
```

```
hostname milab-7206a
```

```
!
```

```
username richy@cisco.com password
```

```
username nas-cisco password
```

```
username milab-5300b password
```

```
username pe1 password
```

```
username paolo@albacom.com password
```

```
username nas-albacom password
```

```
!
```

```
ip subnet-zero
```

```
no ip domain-lookup
```

```
!
```

```

ip vrf RED
rd 100:1
route-target export 100:1
route-target import 100:1
!
ip vrf GREEN
rd 100:2
route-target export 100:2
route-target import 100:2
!
ip cef
!
vpdn enable
!
vpdn-group 1
accept dialin l2f virtual-template 1 remote nas-cisco
local name pe1
!
vpdn-group 2
accept dialin l2f virtual-template 2 remote nas-
albacorn
local name pe1
!
interface Loopback0
ip address 2.2.2.2 255.255.255.255
no ip directed-broadcast
!
interface Serial1/0
ip vrf forwarding RED
ip address 199.10.72.2 255.255.255.0
no ip directed-broadcast
no fair-queue
clockrate 64000
!
interface Serial1/1
ip vrf forwarding GREEN
ip address 199.10.52.2 255.255.255.0
no ip directed-broadcast
no fair-queue
clockrate 64000
!
interface Ethernet3/0
ip address 172.17.238.77 255.255.255.252
no ip directed-broadcast
!
interface Ethernet3/1
no ip address
no ip directed-broadcast
!
interface Ethernet3/2
no ip address
no ip directed-broadcast
!
interface ATM6/0
no ip address
no ip directed-broadcast
no atm ilmi-keepalive
!
interface ATM6/0.1 tag-switching
ip address 199.10.21.2 255.255.255.0
no ip directed-broadcast
tag-switching ip
!
interface Virtual-Template1
ip vrf forwarding GREEN

```

```

ip address 10.0.0.1 255.0.0.0
no ip directed-broadcast
peer default ip address pool CISCO
ppp authentication chap
!
interface Virtual-Template2
ip vrf forwarding RED
ip address 10.0.0.1 255.0.0.0
no ip directed-broadcast
ip mroute-cache
peer default ip address pool CISCO
ppp authentication chap
!
router ospf 100
network 2.2.2.2 0.0.0.0 area 0
network 199.10.21.0 0.0.0.255 area 0
!
router bgp 100
no synchronization
no bgp default ipv4-unicast
neighbor 3.3.3.3 remote-as 100
neighbor 3.3.3.3 update-source Loopback0
no auto-summary
!
address-family ipv4 vrf GREEN
redistribute connected route-map conn2bgp
neighbor 199.10.52.5 remote-as 555
neighbor 199.10.52.5 activate
no auto-summary
no synchronization
no bgp default ipv4-unicast
network 10.0.0.0

```

```

exit-address-family
!
address-family ipv4 vrf RED
redistribute connected
neighbor 199.10.72.7 remote-as 777
neighbor 199.10.72.7 activate
no auto-summary
no synchronization
no bgp default ipv4-unicast
network 10.0.0.0
exit-address-family
!
address-family vpnv4
neighbor 3.3.3.3 activate
neighbor 3.3.3.3 next-hop-self
neighbor 3.3.3.3 send-community extended
no auto-summary
no bgp default ipv4-unicast
exit-address-family
!
ip local pool CISCO 10.0.0.10 10.0.0.30
no ip classless
ip route 172.17.238.0 255.255.255.0
172.17.238.78
ip route 172.17.238.80 255.255.255.240
172.17.238.78
!
access-list 1 permit 10.0.0.0 0.255.255.255
access-list 1 permit 4.0.0.0 0.255.255.255
access-list 2 deny 199.10.52.0 0.0.0.255
access-list 2 permit any
access-list 20 permit 199.10.52.0 0.0.0.255
route-map conn2bgp permit 10

```

```

match ip address 2
!
end
milab-7206a#sh ip vrf
Name      Default RD   Interfaces
RED       100:1       Serial1/0
GREEN     100:2       Serial1/1
           Virtual-Template1
milab-7206a#sh ip v det
VRF RED; default RD 100:1
Interfaces:
Serial1/0
Connected addresses are not in global routing table
Export VPN route-target communities
RT:100:1
Import VPN route-target communities
RT:100:1
No import route-map
VRF GREEN; default RD 100:2
Interfaces:
Serial1/1   Virtual-Template1
Connected addresses are not in global routing table
Export VPN route-target communities
RT:100:2
Import VPN route-target communities
RT:100:2
No import route-map

milab-7206a#sh ip ro v GREEN
Codes: C - connected, S - static, I - IGRP, R - RIP,
M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA
- OSPF inter area

```

```

N1 - OSPF NSSA external type 1, N2 -
OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF
external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS
level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static
route, o - ODR
      P - periodic downloaded static route, T -
traffic engineered route

Gateway of last resort is not set
      4.0.0.0/32 is subnetted, 1 subnets
B   4.4.4.4 [200/1] via 3.3.3.3
      5.0.0.0/32 is subnetted, 1 subnets
B   5.5.5.5 [20/0] via 199.10.52.5, Serial1/1
C 199.10.52.0/24 is directly connected,
Serial1/1
milab-7206a#sh ip bgp v v GREEN
BGP table version is 21, local router ID is
2.2.2.2
Status codes: s suppressed, d damped, h
history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? -
incomplete
      Network  Next Hop    Metric LocPrf
Weight Path
Route Distinguisher: 100:2 (GREEN)
*>i4.4.4.4/32  3.3.3.3      1 100  0 ?
*> 5.5.5.5/32 199.10.52.5   0    0 555 i
milab-7206a#
milab-7206a#
milab-7206a#sh ip ro v RED
Codes: C - connected, S - static, I - IGRP, R
- RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O -
OSPF, IA - OSPF inter area

```

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP

i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area

\* - candidate default, U - per-user static route, o - ODR

P - periodic downloaded static route, T - traffic engineered route

Gateway of last resort is not set

6.0.0.0/32 is subnetted, 1 subnets

B 6.6.6.6 [200/0] via 3.3.3.3

7.0.0.0/32 is subnetted, 1 subnets

B 7.7.7.7 [20/0] via 199.10.72.7, Serial1/0

C 199.10.72.0/24 is directly connected, Serial1/0

milab-7206a#sh ip bgp v v red

%Bad vpnv4 spec

!

! VRFs are case-sensitive

!

milab-7206a#sh ip bgp v v RED

BGP table version is 21, local router ID is 2.2.2.2

Status codes: s suppressed, d damped, h history, \* valid, > best, i - internal

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
---------	----------	--------	--------	--------	------

Route Distinguisher: 100:1 (RED)

*>i6.6.6.6/32	3.3.3.3	0	100	0	?
---------------	---------	---	-----	---	---

*>7.7.7.7/32	199.10.72.7	0	0	777	i
--------------	-------------	---	---	-----	---

milab-7206a#telnet 7.7.7.7 /vrf RED

Trying 7.7.7.7 ... Open

milab-2501b#sh runn

Building configuration...

Current configuration:

!

version 11.3

!

hostname milab-2501b

!

interface Loopback0

ip address 7.7.7.7 255.255.255.255

no ip route-cache

no ip mroute-cache

!

interface Ethernet0

no ip address

no ip route-cache

no ip mroute-cache

!

interface Serial0

ip address 199.10.72.7 255.255.255.0

no ip route-cache

no ip mroute-cache

!

router bgp 777

no synchronization

network 7.7.7.7 mask 255.255.255.255

neighbor 199.10.72.2 remote-as 100

no auto-summary

!

ip classless

end

milab-2501b#sh ip r

Codes: C - connected, S - static, I - IGRP, R - RIP,  
M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA  
- OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF  
NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external  
type 2, E - EGP

i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, \* -  
candidate default

U - per-user static route, o - ODR

Gateway of last resort is not set

6.0.0.0/32 is subnetted, 1 subnets

B 6.6.6.6 [20/0] via 199.10.72.2, 4d00h

7.0.0.0/32 is subnetted, 1 subnets

C 7.7.7.7 is directly connected, Loopback0

C 199.10.72.0/24 is directly connected, Serial0  
milab-2501b#exit

[Connection to 7.7.7.7 closed by foreign host]

milab-7206a#telnet 5.5.5.5 /vrf GREEN

Trying 5.5.5.5 ... Open

milab-2501a#sh running-config

Building configuration...

Current configuration:

!

version 11.3

!

hostname milab-2501a

!

ip subnet-zero

!

interface Loopback0

ip address 5.5.5.5 255.255.255.255

!

interface Serial0

ip address 199.10.52.5 255.255.255.0

no ip route-cache

no ip mroute-cache

no fair-queue

!

interface Serial1

no ip address

no ip route-cache

no ip mroute-cache

!

router bgp 555

no synchronization

network 5.5.5.5 mask 255.255.255.255

neighbor 199.10.52.2 remote-as 100

no auto-summary

!

no ip classless

!

end

milab-2501a#sh ip rou

Codes: C - connected, S - static, I - IGRP, R  
- RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O -  
OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 -  
OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF  
external type 2, E - EGP

i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS  
level-2, \* - candidate default

U - per-user static route, o - ODR

Gateway of last resort is not set

```

4.0.0.0/32 is subnetted, 1 subnets
B   4.4.4.4 [20/0] via 199.10.52.2, 4d00h
5.0.0.0/32 is subnetted, 1 subnets
C   5.5.5.5 is directly connected, Loopback0
C 199.10.52.0/24 is directly connected, Serial0

```

```

milab-2501a#exit

```

```

[Connection to 5.5.5.5 closed by foreign host]

```

```

milab-7206a#1.1.1.1

```

```

Trying 1.1.1.1 ... Open

```

```

milab-ls1010#sh run

```

```

Building configuration...

```

```

Current configuration:

```

```

!
version 12.0
!
hostname milab-ls1010
!
ip subnet-zero
ip host-routing
!
atm lecs-address-default
47.0091.8100.0000.0010.11be.b401.0010.0db0.203
3.00 1
atm lecs-address-default
47.0091.8100.0000.0010.29a6.9e01.0010.29a6.9a1
3.00 2
atm address
47.0091.8100.0000.0010.11be.b401.0010.11be.b40
1.00

```

```

atm router pnni
no aesa embedded-number left-justified
node 1 level 56 lowest
redistribute atm-static
!
interface Loopback0
ip address 1.1.1.1 255.255.255.255
no ip directed-broadcast
!
interface ATM0/0/0
no ip address
no ip directed-broadcast
!
interface ATM0/0/1
no ip address
no ip directed-broadcast
!
interface ATM0/0/2
ip address 199.10.21.1 255.255.255.0
no ip directed-broadcast
no atm auto-configuration
atm uni version 3.1
atm maxvci-bits 10
tag-switching ip
!
interface ATM0/0/3
ip address 199.10.31.1 255.255.255.0
no ip directed-broadcast
tag-switching ip
!
interface ATM0/1/0
no ip address

```

```

no ip directed-broadcast
!
interface ATM1/0/0
no ip address
no ip directed-broadcast
!
interface ATM2/0/0
ip address 172.17.238.22 255.255.255.240
no ip directed-broadcast
atm maxvp-number 0
lane config auto-config-atm-address
lane client ethernet elan1
!
interface ATM3/0/0
no ip address
no ip directed-broadcast
!
interface ATM3/0/1
no ip address
no ip directed-broadcast
!
interface ATM3/0/2
no ip address
no ip directed-broadcast
!
interface ATM3/0/3
no ip address
no ip directed-broadcast
!
interface ATM3/1/0
no ip address
no ip directed-broadcast
!
interface ATM3/1/1
no ip address
no ip directed-broadcast
!
interface ATM3/1/2
no ip address
no ip directed-broadcast
!
interface ATM3/1/3
no ip address
no ip directed-broadcast
!
router ospf 100
network 1.1.1.1 0.0.0.0 area 0
network 199.10.0.0 0.0.255.255 area 0
!
ip default-gateway 172.17.238.17
no ip classless
!
snmp-server community public RO
snmp-server community private RW
!
end

milab-ls1010#sh ip rou
Codes: C - connected, S - static, I - IGRP, R
- RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O -
OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 -
OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF
external type 2, E - EGP

```

i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, \* - candidate default

U - per-user static route, o - ODR

T - traffic engineered route

Gateway of last resort is not set

1.0.0.0/32 is subnetted, 1 subnets

C 1.1.1.1 is directly connected, Loopback0

2.0.0.0/32 is subnetted, 1 subnets

O 2.2.2.2 [110/2] via 199.10.21.2, 4d00h, ATM0/0/2

3.0.0.0/32 is subnetted, 1 subnets

O 3.3.3.3 [110/2] via 199.10.31.3, 4d00h, ATM0/0/3

199.10.31.0/24 is variably subnetted, 2 subnets, 2 masks

S 199.10.31.3/32 is directly connected, ATM0/0/3

C 199.10.31.0/24 is directly connected, ATM0/0/3

172.17.0.0/28 is subnetted, 1 subnets

C 172.17.238.16 is directly connected, ATM2/0/0

199.10.21.0/24 is variably subnetted, 2 subnets, 2 masks

C 199.10.21.0/24 is directly connected, ATM0/0/2

S 199.10.21.2/32 is directly connected, ATM0/0/2

milab-ls1010#exit

[Connection to 1.1.1.1 closed by foreign host]

milab-7206a#

milab-7206a#3.3.3.3

Trying 3.3.3.3 ... Open

milab-7206b#sh running-config

Building configuration...

Current configuration:

```
!
version 12.0
!
hostname milab-7206b
!
ip subnet-zero
ip cef
no ip domain-lookup
ip vrf RED rd 100:1
ip vrf RED route-target export 100:1
ip vrf RED route-target import 100:1
!
ip vrf GREEN rd 100:2
ip vrf GREEN route-target export 100:2
ip vrf GREEN route-target import 100:2
!
!
interface Loopback0
ip address 3.3.3.3 255.255.255.255
no ip directed-broadcast
!
interface ATM1/0
no ip address
no ip directed-broadcast
!
interface ATM1/0.1 tag-switching
ip address 199.10.31.3 255.255.255.0
no ip directed-broadcast
tag-switching ip
```

```

!
interface Serial4/0
 ip vrf forwarding RED
 ip address 199.10.63.3 255.255.255.0
 no ip directed-broadcast
 no ip mroute-cache
 no fair-queue
 clockrate 64000
!
interface Serial4/1
 description Serial line to 2520
 ip vrf forwarding GREEN
 ip address 199.10.43.3 255.255.255.0
 no ip directed-broadcast
 no fair-queue
 clockrate 64000
!
router ospf 100
 network 3.3.3.3 0.0.0.0 area 0
 network 199.10.31.0 0.0.0.255 area 0
!
router rip
 version 2
!
 address-family ipv4 vrf GREEN
 redistribute bgp 100 metric 2
 network 199.10.43.0
 no auto-summary
 exit-address-family
!
router bgp 100
 no synchronization
 no bgp default ipv4-unicast
 neighbor 2.2.2.2 remote-as 100
 neighbor 2.2.2.2 update-source Loopback0
 no auto-summary
!
 address-family ipv4 vrf GREEN
 redistribute rip route-map rip2bgp
 no auto-summary
 no synchronization
 no bgp default ipv4-unicast
 exit-address-family
!
 address-family ipv4 vrf RED
 redistribute static
 no auto-summary
 no synchronization
 no bgp default ipv4-unicast
 exit-address-family
!
 address-family vpnv4
 neighbor 2.2.2.2 activate
 neighbor 2.2.2.2 next-hop-self
 neighbor 2.2.2.2 send-community extended
 no auto-summary
 no bgp default ipv4-unicast
 exit-address-family
!
 ip classless
 ip route vrf RED 6.6.6.6 255.255.255.255
 Serial4/0 199.10.63.6
!
 access-list 10 deny 199.10.43.0 0.0.0.255
 access-list 10 permit any

```

```
route-map rip2bgp permit 10
```

```
match ip address 10
```

```
!
```

```
end
```

```
milab-7206b#sh ip v
```

```
Name      Default RD  Interfaces
```

```
RED       100:1     Serial4/0
```

```
GREEN     100:2     Serial4/1
```

```
milab-7206b#
```

```
milab-7206b#sh ip v det
```

```
VRF RED; default RD 100:1
```

```
Interfaces:
```

```
Serial4/0
```

```
Connected addresses are in global routing table
```

```
Export VPN route-target communities
```

```
RT:100:1
```

```
Import VPN route-target communities
```

```
RT:100:1
```

```
No import route-map
```

```
VRF GREEN; default RD 100:2
```

```
Interfaces:
```

```
Serial4/1
```

```
Connected addresses are in global routing table
```

```
Export VPN route-target communities
```

```
RT:100:2
```

```
Import VPN route-target communities
```

```
RT:100:2
```

```
No import route-map
```

```
milab-7206b#sh ip rou v GREEN
```

```
Codes: C - connected, S - static, I - IGRP, R - RIP,
M - mobile, B - BGP
```

```
D - EIGRP, EX - EIGRP external, O -
OSPF, IA - OSPF inter area
```

```
N1 - OSPF NSSA external type 1, N2 -
OSPF NSSA external type 2
```

```
E1 - OSPF external type 1, E2 - OSPF
external type 2, E - EGP
```

```
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS
level-2, * - candidate default
```

```
U - per-user static route, o - ODR
```

```
T - traffic engineered route
```

```
Gateway of last resort is not set
```

```
C 199.10.43.0/24 is directly connected,
Serial4/1
```

```
4.0.0.0/32 is subnetted, 1 subnets
```

```
R 4.4.4.4 [120/1] via 199.10.43.4,
00:00:08, Serial4/1
```

```
5.0.0.0/32 is subnetted, 1 subnets
```

```
B 5.5.5.5 [200/0] via 2.2.2.2, 4d00h
```

```
milab-7206b#sh ip bgp v v GREEN71
```

```
BGP table version is 31, local router ID is
3.3.3.3
```

```
Status codes: s suppressed, d damped, h
history, * valid, > best, i - internal
```

```
Origin codes: i - IGP, e - EGP, ? -
incomplete
```

```
Network  Next Hop  Metric LocPrf
Weight Path
```

```
Route Distinguisher: 100:2 (GREEN)
```

```
*> 4.4.4.4/32 199.10.43.4 1 32768 ?
```

```
*>i5.5.5.5/32 2.2.2.2 0 100 0 555 i
```

```
milab-7206b#sh ip rou v RED
```

<sup>71</sup> Or, "show ip bgp vpnv4 vrf GREEN."

Codes: C - connected, S - static, I - IGRP, R - RIP,  
M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA  
- OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF  
NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external  
type 2, E - EGP

i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, \* -  
candidate default

U - per-user static route, o - ODR

T - traffic engineered route

Gateway of last resort is not set

6.0.0.0/32 is subnetted, 1 subnets

S 6.6.6.6 [1/0] via 199.10.63.6, Serial4/0

7.0.0.0/32 is subnetted, 1 subnets

B 7.7.7.7 [200/0] via 2.2.2.2, 4d00h

C 199.10.63.0/24 is directly connected, Serial4/0

milab-7206b#sh ip bgp v v RED

BGP table version is 31, local router ID is 3.3.3.3

Status codes: s suppressed, d damped, h history, \*  
valid, > best, i - internal

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
---------	----------	--------	--------	--------	------

Route Distinguisher: 100:1 (RED)

\*> 6.6.6.6/32 199.10.63.6 0 32768 ?

\*>i7.7.7.7/32 2.2.2.2 0 100 0 777 i

milab-7206b#

milab-7206b#telnet 6.6.6.6 /vrf RED

Trying 6.6.6.6 ... Open

milab-2503#

milab-2503#sh runn

Building configuration...

Current configuration:

!

version 11.3

service timestamps debug uptime

service timestamps log uptime

no service password-encryption

!

hostname milab-2503

!

username milab-2520 password 0 paolo

ip subnet-zero

isdn switch-type basic-net3

!

interface Loopback0

ip address 6.6.6.6 255.255.255.255

!

interface Loopback1

no ip address

!

interface Ethernet0

no ip address

!

interface Serial0

ip address 199.10.63.6 255.255.255.0

no ip mroute-cache

no fair-queue

!

interface BRI0

ip unnumbered Loopback0

```

encapsulation ppp
dialer map ip 10.0.0.0 1231243
dialer-group 1
ppp authentication chap
ppp chap hostname milab-2503
ppp chap password 7 051B07002D43
ppp multilink
!
ip classless
ip route 0.0.0.0 0.0.0.0 Serial0
ip route 10.0.0.0 255.0.0.0 BRI0
!
dialer-list 1 protocol ip permit
!
end
milab-2503#sh ip rou

Codes: C - connected, S - static, I - IGRP, R - RIP,
M - mobile, B - BGP

    D - EIGRP, EX - EIGRP external, O - OSPF, IA
- OSPF inter area

    N1 - OSPF NSSA external type 1, N2 - OSPF
NSSA external type 2

    E1 - OSPF external type 1, E2 - OSPF external
type 2, E - EGP

    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * -
candidate default

    U - per-user static route, o - ODR

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

    6.0.0.0/32 is subnetted, 1 subnets
C   6.6.6.6 is directly connected, Loopback0
C  199.10.63.0/24 is directly connected, Serial0
S  10.0.0.0/8 is directly connected, BRI0
S*  0.0.0.0/0 is directly connected, Serial0

```

```

milab-2503#exit
[Connection to 6.6.6.6 closed by foreign
host]
milab-7206b#telnet 6.6.6.6 /vrf RED
Trying 6.6.6.6 ... Open
Password:
[Connection to 6.6.6.6 closed by foreign
host]
milab-7206b#
milab-7206b#telnet 4.4.4.4 /vrf GREEN
Trying 4.4.4.4 ... Open
milab-2520#sh run
Building configuration...

Current configuration:
!
version 11.3
!
hostname milab-2520
!
username milab-2503 password 0 paolo
ip subnet-zero
multilink virtual-template 1
isdn switch-type basic-net3
!
!
interface Loopback0
ip address 4.4.4.4 255.255.255.255
!
interface Ethernet0
ip address 10.0.1.1 255.255.255.0
!
interface Serial0
description Serial line to 7206b

```

```

ip address 199.10.43.4 255.255.255.0
no ip mroute-cache
load-interval 30
no fair-queue
!
interface Serial1
no ip address
encapsulation ppp
load-interval 30
no fair-queue
ppp authentication chap
ppp chap hostname milab-2520
ppp chap password 7 0014120908544B46
ppp multilink
!
interface BRI0
no ip address
encapsulation ppp
dialer pool-member 1
isdn switch-type basic-net3
ppp authentication chap
ppp chap hostname milab-2520
ppp chap password 7 06160E2E4041
ppp multilink
!
router rip
version 2
network 4.0.0.0
network 199.10.43.0
no auto-summary
!
ip classless

```

```

ip route 199.10.52.0 255.255.255.0
199.10.43.3
dialer-list 1 protocol ip permit
!
end
milab-2520#exit
[Connection to 4.4.4.4 closed by foreign
host]
milab-7206b#exit
[Connection to 3.3.3.3 closed by foreign
host]
milab-7206a#
milab-7206a#exit
milab-5300b#
milab-5300b#sh runn
Building configuration...
Current configuration:
!
! Last configuration change at 11:17:31
UTC Sun Apr 11 1999
! NVRAM config last updated at 17:44:47
UTC Mon Apr 5 1999
!
version 11.3
!
hostname milab-5300b
!
username pe1 password 0 cisco
username nas-cisco password 0 cisco
username nas-albacom password 0 albacom
no ip domain-lookup
ip host modem1 2001 100.100.100.100
vpdn enable
!

```

```

vpdn search-order domain

vpdn-group 1
    request dialin l2f ip 172.17.238.77 domain
    cisco.com
    local name nas-cisco
    !
vpdn-group 2
    request dialin l2f ip 172.17.238.77 domain
    albacom.com
    local name nas-albacom
    !
isdn switch-type primary-net5
    !
controller E1 0
    clock source line primary
    pri-group timeslots 1-16
    !
controller E1 1
    clock source line secondary
    !
controller E1 2
    clock source internal
    channel-group 1 timeslots 1-31
    !
controller E1 3
    clock source internal
    !
interface Loopback0
    ip address 100.100.100.100 255.255.255.255
    !
interface Ethernet0
    ip address 172.17.238.89 255.255.255.240
    !

```

```

interface Serial0:15
    no ip address
    dialer rotary-group 1
    dialer-group 1
    isdn switch-type primary-net5
    isdn incoming-voice modem
    !
interface Serial2:1
    no ip address
    !
interface Group-Async1
    ip unnumbered Loopback0
    encapsulation ppp
    ip tcp header-compression passive
    dialer in-band
    async dynamic address
    async dynamic routing
    async mode interactive
    peer default ip address pool ippool
    ppp authentication pap
    group-range 1 30
    hold-queue 10 in
    !
interface Dialer1
    no ip address
    !
router eigrp 100
    network 10.0.0.0
    network 172.17.0.0
    !
ip local pool ippool 10.10.10.1 10.10.10.31
ip classless

```

```
!
dialer-list 1 protocol ip permit
!
end
```

## 1.5.10 MPLS Traffic Engineering<sup>72</sup> (TE) Configuration<sup>†</sup>

NOTE: MPLS TE is a developing technology. The reader is encouraged to keep up to date on the technology, Cisco platform deployment timelines, as well as any changes to Cisco IOS command syntax. Recent versions of IOS have apparently replaced tag-switching references to MPLS. The following configuration reflects the new syntax for traffic engineering in an MPLS environment.

### 1.5.10.1 New Command Syntax

Cisco has recently decided to refer to the utilization of RRR in MPLS as “MPLS Traffic Engineering.”

Readers that have dealt with the older EFT documentation of RRR will now<sup>73</sup> notice that a few config-mode and EXEC-mode commands have changes, as highlighted in the table below.

Previous Command Syntax	Newer Command Syntax
tag-switching tsp-tunnels <sup>74</sup>	mpls traffic-eng tunnels
tag-switching rrr router id ip- address	mpls traffic-eng router-id □ ip- address
tunnel mode tag-switching	tunnel mode mpls traffic-eng
tunnel rrr bandwidth bandwidth bandwidth	tunnel mpls traffic-eng bandwidth bandwidth

Currently, Cisco supports only the IS-IS IGP with the new TLVs. Although Cisco has OSPF with Opaque LSA support, it cannot currently be used as an IGP in MPLS TE environments. The reader is encouraged to contact the “cs-rrr” e-mail alias for updates on OSPF support in with MPLS TE.

### 1.5.10.2 MPLS TE Issues<sup>75</sup>

- RSVP does not support policing or WFQ parameters. Therefore there is no correlation between MPLS TE signaling (RSVP with extensions) and, for example, CAR.
  - Currently, there is no per-tunnel Class of Service support. So one can’t set things up so that one CoS goes over one TE tunnel while another class utilizes a different tunnel.
  - MPLS tunnel signaling in RRR is incompatible with MPLS tunnel signalling in the first release of tag switching. Tag Switching was released in 11.1CT and 12.0, with pre-standard RSVP extensions to support TSP signaling. RSVP extensions for LSP signalling in 12.0(5)S/12.0(6)T, which are also pre-standard, do not inter-operate with those in 12.0/12.0(4)S/12.0(5)T/11.1CT.
  - MPLS TE is in 12.0(5)S. However, MPLS-VPN is not in 12.0S. In order to get both MPLS TE and MPLS-VPN features in a released image, one has to await the arrival of IOS 12.0(6)T
  - Unicast only
  - Limited to a single IGP area. Hence, one cannot go across Service Provider networks
  - Current MPLS TE implementation {Cisco IOS 12.0(5)S} does not have support for TE over OSPF. Support for OSPF as an IGP in TE environments is expected in Cisco IOS 12.0(6)S and 12.0(6)T. The reader should consult with Engineering or Product Marketing regarding availability
- MPLS Traffic Engineering is not the same as Guaranteed Bandwidth for VPN (GBVPN). The latter is a future capability that will synchronize Cisco IOS QoS tools, such as CAR, with Traffic Engineering. Until then, MPLS TE software cannot restrict traffic to a particular rate

<sup>72</sup> It appears that Cisco’s version of Constraint-based routing will be referred to as MPLS Traffic Engineering versus the older name: RRR. The reader needs to keep in mind that, technically speaking, MPLS TE and RRR are not the same thing. MPLS TE is an application of RRR.

<sup>†</sup> Thanks to Robert Raszuk for pointing to George Swallow’s excellent demo scenarios with those MPLS TE configurations.

<sup>73</sup> The syntax change will occur in Cisco IOS 12.0(6)T.

<sup>74</sup> At both the global and interface subcommand levels.

<sup>75</sup> As of Cisco IOS 12.0(6)T

### 1.5.10.3 MPLS TE Lab Configuration Scenarios

NOTE: MPLS Traffic Engineering is a new paradigm in inter-network traffic engineering. The reader is cautioned to verify command syntax as well as default settings, as future versions of Cisco IOS software are introduced. It may be possible for particular commands and command parameters to change default values in the future.

#### 1.5.10.3.1 MPLS TE Lab Scenario One - Basic TE Environment

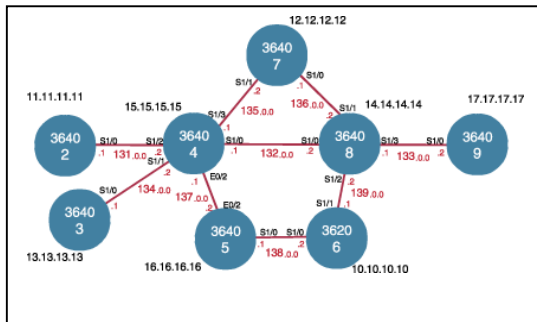


Figure 14- MPLS Traffic Engineering Scenario 1, Basic TE

```
3640-4#sh run
```

```
Building configuration...
```

```
Current configuration:
```

```
!
```

```
version 12.0
```

```
!
```

```
hostname 3640-4
```

```
!
```

```
mpls traffic-eng tunnels
```

```
!
```

```
int s1/0
```

```
mpls traffic-eng tunnels
```

```
ip rsvp bandwidth 1000
```

```
mpls traffic-eng administrative-weight 10
```

```
!
```

```
int s1/1
```

```
mpls traffic-eng tunnels
```

```
ip rsvp bandwidth 1000
```

```
mpls traffic-eng administrative-weight 10
```

```
!
```

```
int s1/2
```

```
mpls traffic-eng tunnels
```

```
ip rsvp bandwidth 1000
```

```
mpls traffic-eng administrative-weight 10
```

```
!
```

```
int s1/3
```

```
mpls traffic-eng tunnels
```

```
ip rsvp bandwidth 1000
```

```
mpls traffic-eng administrative-weight 10
```

```
!
```

```
int e0/2
```

```
mpls traffic-eng tunnels
```

```
ip rsvp bandwidth 1000
```

```
mpls traffic-eng administrative-weight 10
```

```
router isis
```

```
mpls traffic-eng router-id Loopback0
```

```
mpls traffic-eng level-1
```

```
!
```

```
end
```

#### 1.5.10.3.2 MPLS TE Lab Scenario Two - Basic Tunnel Configuration

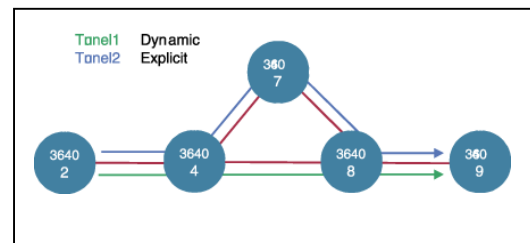


Figure 15- MPLS Traffic Engineering Scenario 2, Basic Tunnel Configuration

```

3640-2#sh run
Building configuration...
Current configuration:
!
version 12.0
!
hostname 3640-2
!
mpls traffic-eng tunnels
!
! dynamic tunnel to 3640-9
!
int tunnel1
 ip unnum lo0
 tunnel destination 17.17.17.17
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng bandwidth 100
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng path-option 1 dynamic
! Create an explicit path
 ip explicit-path identifier 2
 next-address 131.0.0.2
 next-address 135.0.0.2
 next-address 136.0.0.2
 next-address 133.0.0.2
! Second Tunnel is added to same destination with
an explicit path
int tunnel2
 ip unnum lo0
 tunnel destination 17.17.17.17
 tunnel mode mpls traffic-eng

```

```

tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng bandwidth 100
tunnel mpls traffic-eng priority 1 1
tunnel mpls traffic-eng path-option 1
explicit identifier 2

```

### 1.5.10.3.3 MPLS TE Lab Scenario Three -Path Options

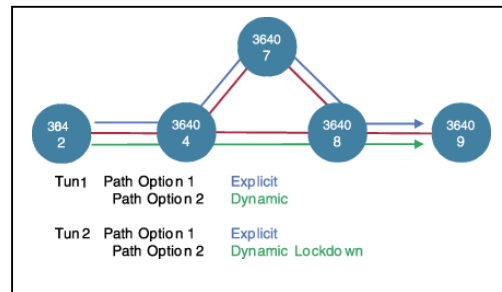


Figure 16-MPLS Traffic Engineering Scenario 3, Path Options

```

3640-2#sh run
Building configuration...
Current configuration:
!
version 12.0
!
hostname 3640-2
!
mpls traffic-eng tunnels
mpls traffic-eng reoptimize
!
! Configure a dynamic tunnel to 3640-9...
!
! Reconfigure Tunnel1 like Tunnel2, but
Lockdown 2nd path
!
int tunnel1
 ip unnum lo0

```

```
tunnel destination 17.17.17.17
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng bandwidth 100
tunnel mpls traffic-eng priority 1 1
tunnel mpls traffic-eng path-option 1 explicit id 2
tunnel mpls traffic-eng path-option 2 dynamic
lockdown
```

! Create an explicit path

```
ip explicit-path identifier 2
next-address 131.0.0.2
next-address 135.0.0.2
next-address 136.0.0.2
next-address 133.0.0.2
```

! Second Tunnel is added to same destination with an explicit path

```
int tunnel2
ip unnum lo0
tunnel destination 17.17.17.17
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng bandwidth 100
tunnel mpls traffic-eng priority 1 1
tunnel mpls traffic-eng path-option 1 explicit
identifier 2
!
! Add backup option to Tunnel 2
!
tunnel mpls traffic-eng path-option 2 dynamic
```

## 1.5.11 Performance and Management Characteristics

### 1.5.11.1 Scalability of MPLS-VPNs

In the design of MPLS-VPNs, there are a number of different scalability variables that the reader will hopefully be mindful of, until more scalability information becomes available from Engineering and real-world networks. Cisco does not currently have any specific numbers to point the SP field to, so far as design limitations of MPLS-VPN entities are concerned.

The reader will need to keep in mind that current limitations may be eased, with the introduction of a few MPLS-VPN features that are on Development Engineering's plate.

Here are some of the variables that engineers need to look at, before "blessing" an MPLS-VPN design:

1. The number of VRFs per PE is tied into the IDB limit on a particular PE router. In most Cisco IOS images in the field today, this value is 255, and will later be changed to higher numbers that are platform-specific. One needs to contact Product Marketing or Engineering for updates on the platform(s) being envisaged.
2. The total number of VPN routes. This is mostly a matter of how much memory there is on the PE. Preliminary tests done by Engineering point to the possibility of supporting about 100,000 VPN routes on a 128MByte RSP4/7500 or a 7200 routers with a 200 or 300 MHz NPE. It appears that the VRF overhead is low. This design variable boils down to the number of routes one can support on the router in question.<sup>76</sup>
3. Depending on connection type, one needs to consider the number of sub-interfaces supported. For example, the number of sub-interfaces (and hence the number of VRFs on that physical interface) on an ATM interface, will be limited to the number of VCs that the interface's hardware can support.

<sup>76</sup> Once Outbound Route Filtering and/or inbound route target filtering is available, this number is expected to rise.

4. The number of routing peers. This is a function of:
  - The number of PE IBGP sessions the router is supporting
  - The number of CE sessions
  - The routing protocol used to the CE

The number depends on the Cisco IOS release, and the routing protocol. This is not really a VPN-specific issue. The number is expected to be similar to the non-VPN case.

5. The total forwarding bandwidth of the router in question, in terms of the packets-per-second (pps) and Mbps rates.
6. The forwarding features (e.g., Traffic Shaping or WFQ) that are enabled on the PE-CE links, which will most likely reduce the forwarding bandwidth of the router. Again, this is not a VPN specific issue.

As the field develops more experience with MPLS-VPN deployments, and as Engineering conducts more performance testing, one will be able to provide more accurate scalability information.

### 1.5.11.2 MPLS Network Management

NOTE: The reader is cautioned to check with Product Marketing and other resources for up-to-date information on what will be available.

#### 1.5.11.2.1 MPLS MIBs

The MPLS MIB undertaking involves the introduction of the following potential MIBs:

- MPLS-LDP
- MPLS TE
- VSI-controller
- Cisco MPLS
- Other

The goal of the development engineering is for reaching EFT status for the MPLS MIBs in CQ3'99.

#### 1.5.11.2.2 Ping and RTR MIBs

In certain VPN environments, a Service Provider is interested in managing PE-CE links from the PEs. They would like to use the Ping MIB in order to perform management. However, the Ping and RTR MIBs are not VPN aware. Eureka, discussed elsewhere in this document, does the correlation to the VPN service.

### 1.5.12 MPLS-VPN Must-knows

MPLS-VPN, like any new technology or paradigm, takes getting used to, before one becomes familiar with the appropriate monitoring and debugging commands, as well as overall how it functions.

- It is important to keep in mind that the global routing table<sup>77</sup> and the per-VRF routing table are independent entities. It is important to understand that the familiar Cisco IOS commands apply to IP routing in a global routing table context. For example, “show ip route,” and other EXEC-level show commands; as well as utilities such as ping, traceroute, and telnet; all invoke the services of the Cisco IOS routines that deal with the global IP routing table.
- It is also important to become familiar with the fact that a local VRF interface on a PE is not considered a directly-connected interface in a traditional sense. When one configures, for example, a Fast Ethernet interface on a PE to participate in a particular VRF/VPN, the interface no longer shows up as a directly-connected interface when one issues a “show ip route.” One needs to issue a “show ip route vrf vrf-name” command in order to see that interface in a routing table.
- One will also notice that it possible to issue a standard “telnet” command from a CE router to connect to PE router. However, from that PE router, one needs to issue a “telnet CERouterName /vrf RF-NAME” command in order to connect from the PE to CE router. Similarly, one needs to utilize “traceroute” and “ping” commands in a VRF context.
- It will also be important to realize that the MPLS-VPN backbone relies on the appropriate IGP that is configured for MPLS, e.g., EIGRP or OSPF. When one issues a “show ip route” on a PE routers, one should see the IGP-derived routes connecting the PEs together. Contrast that with the issuance of the “show ip route vrf VRF-NAME” which will display IBGP

<sup>77</sup> I find this term confusing and mis-leading, but nevertheless, one will encounter this term on different e-mail aliases' exchanges and some engineering documentation.

- routes connecting C sites in a particular VPN.
- One can get around the lack of support for inter-Service Provider MPLS-VPN connectivity, through the utilization of GRE tunnels between two PE routers. So in a typical RBOC/ILEC environment (which is different than European and other PTTs), an ILEC's PE needs to have a GRE tunnel connecting it to another PE from that same ILEC, across an IXC network. Whether that IXC's network supports MPLS-VPN on its own or not, doesn't really matter.
- In Hub-and-Spokes MPLS-VPN environments, the Spoke routers have to have unique Route Distinguishers. In order to use the Hub site as a transit point for connectivity in such an environment, the Spoke sites export their routes to the Hub. Meanwhile, the Hub router (through a second interface) will re-export Spoke routes to the other Spokes.

It appears at first, that the easiest way to handle the Hub-and-Spokes topology is to assign the same RD to all spoke VRFs, since the Spoke VRFs do not exchange routes with one another, and therefore, there is no need to make IP VPN routes unique. Such routes are already unique by the logic of the topology.

However, due to the current MPLS-VPN implementation, one must use a different RD for each spoke VRF. The BGP selection process applies to all the routes that have to be imported into the same VRF plus all routes that have the same RD of such a VRF. Once the selection process is done, only the best routes are imported. In this case this can result in a best route which is not imported. To illustrate, suppose one has a Hub-and-Spokes environment whereby there are CE routers H, S(poke)1, and S2. Suppose further that S1 advertises a route to 192.1.1.0/24, via its PE router, to sites H and S2. Using RTs, H's PE will import this route to this VPN. Now suppose H's PE has been configured to re-advertise this back up. Through its PE router, S2 receives the route to 192.1.1.0/24 from H. If the RDs are unique, then S2 now has two routes to 192.1.1.0/24 in the VPNv4 BGP table. The two routes will not be compared. If the two Spoke sites are utilizing the same RD, then the two afore-mentioned routes to 192.1.1.0/24 will be compared. This may cause the version from S1 to be preferred over the one from H, but the version from S1 is not a candidate for inclusion in S2's VRF, and so S2 gets no route to 192.1.1.0/24 at all.

It doesn't appear that this stipulation will change in the near future. As always, one needs to check with the Product Marketing or the "tag-vpn" e-mail alias.

In this environment, there are two possible workarounds:

1. One is to use unique RDs for each Spoke site, while
2. another would be to get H to perform summarization (dynamically, or statically).

Summarization is possible, but may be difficult when deploying Hub-and-Spokes on an existing network.

If none of the Spokes have the need for independent Public Internet connections, then the easiest workaround is to get the Hub PE router to advertise 0/0 and not any other routes. If that's not possible, then some summarization of the known addressing space of the sites is the next best alternative. If all else fails, dynamic summarization at the Hub may be appropriate.

These options are appropriate if Spoke IP addresses do not have to be changed.

The main aspect of this (I think) is that it's now clear to anyone that the RD doesn't play a role on the route distribution between PEs (route import/export are based on RTs, versus RDs). However, due to the current MPLS-VPN implementation, it just so happens that the RD may play a role on the route selection and influence the import process.

So, to summarize, customers must have different RDs per spoke-VRF. Cisco can provide workarounds to those Hub-and-Spokes environments, with summarization and default routing so long as those workarounds are acceptable to the Service Provider's customer.

- Must Have MPLS Connectivity amongst PE routers (dynamic and/or TE tunnel). MPLS must be enabled across the backbone, utilizing either dynamic label switching or a traffic-engineered tunnel between the Provider Edge routers.

- Must have a /32 route for all PEs in IGP. This is because MPLS-VPN, as stated earlier, uses two-level labeling and so one must have a label that will directly reach the router that assigned the second-level label.
- Changing VRF forwarding on an interface removes the IP address. This surprises people is that when one changes the VRF forwarding on an interface, the IP address associated with that interface is removed and has to be configured. This is done on purpose! If a Service Provider is moving the existing address from one customer to another, it doesn't make sense to preserve the address on the one that has just had VRF forwarding disabled. Also by removing the address on the interface, it makes sure that the routing protocols configured on that interface properly adjust their routing information.
- Must have CEF enabled on PE router.
- To ping (traceroute) between CE routers, must either use extended ping, or 'redistribute connected' into VPNv4 IBGP on PE.
- Traceroute between two CEs will display provider nodes, unless 'no tag ip propagate-ttl' on PEs.
- Support for overlapping private ASNs. In today's implementation of MPLS-VPN, the BGP code will not strip the private ASN if it is equal to the neighboring ASN. This precludes the use of dummy ASNs (same private ASN at different sites). However, so long as all ASNs of a VPN are unique, one is okay with this configuration. One needs to keep in mind that the AS\_PATH Attribute is essential for detecting routing loops in BGP. If one strips private ASNs (and if the ASN happens to be the neighboring one), then the site is not able to detect IP routing loops. That task will fall on the shoulders of the PE router. One needs to be even more careful in environments that happen to also have backdoor connections amongst some customer routers. The overlapping ASN scenario presents itself in environments where independent customers have built networks with the same private ASNs. These customers then request intranet connectivity from a Service Provider utilizing MPLS-VPN. These independent customers connect to the PE routers via eBGP, but they would like not to re-number their ASN.

## 1.6 MPLS-VPN (Uncommitted<sup>78</sup>) Future Features

NOTE: This section discusses possible future features for MPLS-VPN. This section was included to help answer some of the questions that have arisen in VPN discussions. The reader should not discuss these features with anyone unless expectations have been set properly. None, some, or all of the features may be incorporated in the future.

### 1.6.1 PPP/VPN - Today<sup>79</sup>

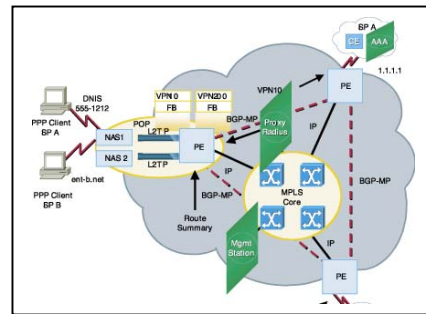


Figure 17 - PPP + MPLS/VPNs in Cisco IOS 12.0T

Currently, there are a few stipulations to configuring PPP + MPLS-VPN

- One has to define a virtual template per VPDN in the Access Server<sup>80</sup>.
- One can define only a single AAA server per access server. Proxy is needed to be able to redirect the AAA requests to customers servers.
- All the AAA customer servers must be in the global routing table thus they must have a unique IP address (NAT is not an alternative).
- One is also supposed to number the virtual templates in the Access MPLS VPN configuration. Previously one was not supposed to number virtual templates - they were always unnumbered to some other LAN or Loopback interface<sup>81</sup>.

<sup>78</sup> As of the writing of this document (middle of June 1999).

<sup>79</sup> In Cisco IOS 12.0(5)T and higher.

<sup>80</sup> In 12.0T, one can have up to 25 Virtual Templates per dial-in server. That is not a limitation in a wholesale dial environment.

<sup>81</sup> This was because the Virtual Access interfaces that are cloned from the Virtual Template took the IP address setting

## 1.6.2 PPP/VPN Integration – Multi-FIB VPNs

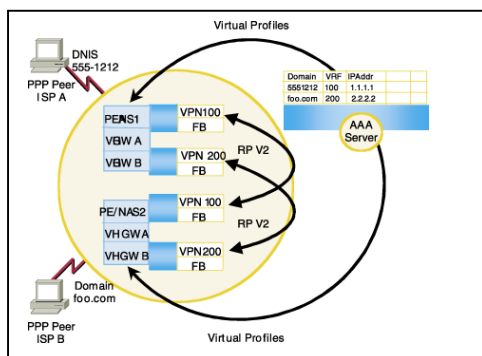


Figure 18 - Potential PPP/MPLS-VPN Integration

A Cisco as5300/5800 will provide PPP termination, and L2F/L2TP tunnel initiation with a Cisco 7200, with the latter performing full-fledged MPLS-VPN functionality.

There are no commitments to provide full MPLS-VPN functionality on current access server platforms.

Enhancements to SGBP are contemplated (no commitments yet) to provide a more scalable L2F/L2TP/MPLS-VPN functionality. If this project is committed, we shall provide details.

## 1.6.3 PPP MPLS-VPN Integration – Scaling PPP

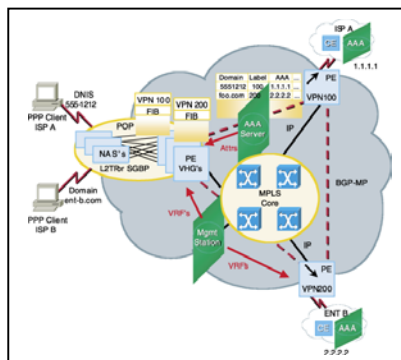


Figure 19 - Scalability

Should we proceed forward with this feature, PE's will be pre-configured to have the appropriate VRFs. AAA will be used to download the per-VPN attributes for the CPE Radius, DHCP, etc. Different PE's will be able to host different VPNs. Simple NAS load-balancing will be an option. SGBP extensions may allow dynamic instantiation and better load balancing.

## 1.6.4 PPP MPLS-VPN Without Tunnels

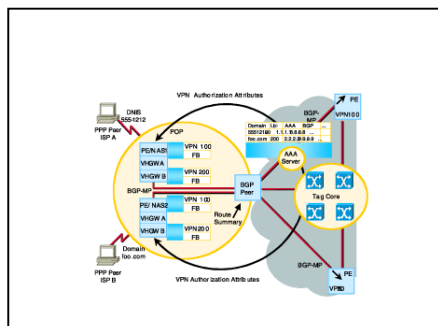


Figure 20- Long-term Potential PPP/VPN Integration

Should we proceed forward with this feature (which would occur after we commit to the previous features first), Cisco Access Servers will have the ability to house multiple Virtual Home Gateways.

from the Template. If one numbered the Template they'd end up with a number of Virtual Access interfaces with the same IP address. Development Engineering made a change to the way Virtual Access interfaces are cloned so that one can now have numbered Templates and the Virtual Access interfaces will be unnumbered to the Template.

### 1.6.5 Proposed MPLS/VPN Multicast Support

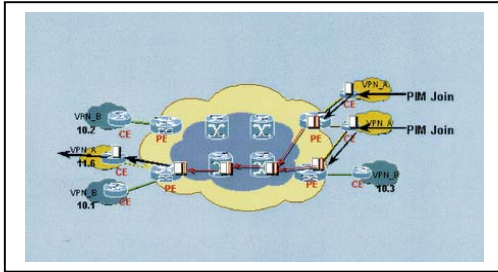


Figure 21 - Proposal MPLS/Multicast Support

- Ingress PE router controls that the requested multicast group is in the VPN
- Ingress PE adds the Label and VRF into the PIM JOIN message
- P routers forward the PIM message
- PIM JOIN message are forwarded only to sites belonging to the same VPN

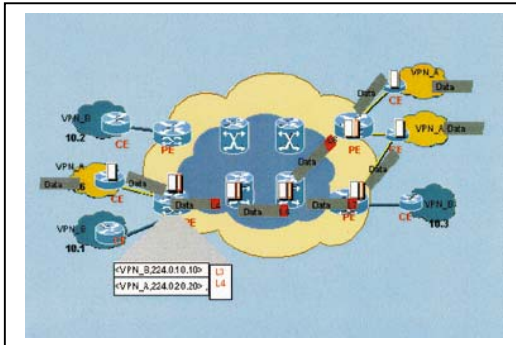


Figure 22 - Proposed MPLS/Multicast Support, the Next Steps

- Egress PE router will remove the Label and VRF information in PIM JOIN message
- Multicast destination address (224.X.X.X) is looked up in the VPN Multicast FIB
- A Multicast Label is imposed in front of the Packet
- Packet are transacted on the VPN multicast tree

### 1.6.6 MPLS/VPN Route-Map Support

Route map support for extended bgp communities, whereby one is able to set certain characteristics on a route based on finding a particular extended attribute, is on the roadmap. Similarly for the ability to also set a particular extended bgp community attribute.

### 1.6.7 Proposed DSL Interaction with MPLS-VPN<sup>82</sup>

It is expected that the 6400 platform will provide, around the September timeframe, a “front-end” function for MPLS-VPN, versus MPS-VPN Edge functionality. It will be similar to what a Cisco access server is able to support today, with Cisco IOS 12.0(5)T. The 6400 will terminate, as it does today, remote PPP sessions (2000 today, while aiming for 8000 in the near future), and will be VPN-aware through the mapping of virtual interfaces into VRFs.

Further out, in the second half of 2000, there is a possibility of having a DSL platform perform full-fledged MPLS-VPN functionality.

## 2 Cisco Service Management for MPLS-VPN (aka “Eureka”)

VPN Solutions Center: MPLS Solutions 1.0, (code named “Eureka” – and will be referred to as Eureka in this document)<sup>83</sup>, is a modular suite of network and service management applications, that defines and monitors VPN services for Service Providers. It is that part of the operations management that addresses flow-through provisioning, service auditing, and SLA measurement of IP MPLS-VPN environments.

Eureka will also integrate with Cisco IP Manager (CIPM) for element management. Other features include, CoS provisioning, VPN-aware Netflow accounting, and Service Level Agreement (SLA) monitoring. Eureka provides a complete set of provisioning, accounting and SLA monitoring API's.

### 2.1 Platform Requirements

- Sun UltraSparc 2 or higher-performing hardware, with one or more processors
- Solaris 2.6

<sup>82</sup> Please contact Product Marketing for confirmation on this, as well as any changes.

<sup>83</sup> Also known as Cisco Service Management (CSM) for IP MPLS VPNs.



### 2.2.3 Eureka Administrative Console

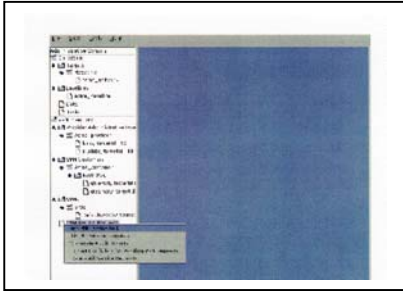


Figure 25-Administrative Console Graphical User Interface for Eureka

The GUI administrative console is used to initiate service requests for defining VPNs, sites, services, etc. It provides wizards for VPN administration tasks; allows the definition of PE and CE objects; and facilitates the launching of VPN provisioning wizards.

The administrative console presents graphical views of Provider and VPN topologies; as well as information and reports on devices, VPNs, and services.

### 2.2.4 Provisioning Steps

In the next few pages, the steps a Service Provider goes through to establish a VPN for a customer are discussed.

#### 2.2.4.1 Defining Networks and Targets

In this initial step, the operator must define the Network Name (namespace) of all the routers that will be used. The network name is a concept that allows one to partition the PE-CE space. The system collects data from these routers so that all the parameters that are needed to access this router (users, passwords, IP addresses, etc.) are specified here.

#### 2.2.4.2 Defining Provider and Customer Device Structure

The operator identifies the entities from the previous step that are Provider Edge versus Customer Edge routers. The PEs are grouped into objects called Provider Administrative Domains (PADs). The CEs are grouped into objects called Customer Sites.

PADs can be organized into regions, in much the same way that customers divide into sites. The primary objective for having regions is to allow different IP pools to be used in specific large regions (for example, Europe vs. Asia). But the same feature can be extended to mark CEs with regions, thus simplifying PE selection (for example, only presenting Europe PEs when adding service to a Europe CE).

The GUI steps the administrator through entering the PAD information through a screen to enter Provider-specific information. Besides entering the Provider name, this is the place where the administrator also specifies the BGP Autonomous System Number (ASN).

#### 2.2.4.3 Defining the Customer Edge Routers

As part of the VPN object, the network administrator defines CE routing communities (CERCs), which are descriptions of the topological structure in the VPN. At this stage, the administrator is just describing the VPNs and their intended topological structures.

#### 2.2.4.4 Defining Customer VPNs

Since in MPLS VPNs, customers join the VPN rather than just create point-to-point tunnels, as is the case in the overlay model; the administrator must define VPNs that each customer will join.

#### 2.2.4.5 Downloading CE & PE Configurations

Eureka will validate the configuration that it creates before it is scheduled for download to the actual network elements (i.e., CEs & PEs). Such a download may occur in the middle of the night along with all the other new service requests. The scheduling of these configuration downloads is fairly flexible.

#### 2.2.4.6 Other Steps

In addition to defining Customer and Provider space, the administrator may import router configuration files into Eureka.

These configurations can then be identified to Eureka as a CE or PE router's. The routers' definitions are now stored within the Eureka repository as network objects. Eureka also allows the administrator to define address pools which may be used to provision the PE to CE links, when a service request comes in.

### 2.2.5 Service Requests

Service Operators take all service request information from the customer. The product assists the operator in making entries since it already has been initialized with information on the customers, like the CE and PE routers that will be affected, the VPN etc.

GUI wizards step the operator through to accept the customer name, the CE router (if this definition exists within Eureka), as well as the VPN name. Eureka tries to simplify the task of provisioning the PE and the CE by automating some of the tasks required in setting up an MPLS VPN. It does so by using a CERC. Since MPLS VPNs are restrained routing communities rather than point-to-point connections, this concept allows Eureka to define the topology of this restrained community with respect to the CE that is joining the VPN. A group of CERCs forms a VPN. Eureka allows the operator to define whether the CE router, that is now joining the CERC (part of the VPN), is joining it as a hub or a spoke entity. Based on this information, Eureka is able to generate values for the Route Targets (RTs) that are required in the Cisco IOS MPLS-VPN configuration.

The service provisioning wizards will also allow definition of CoS requirements for the service. Eureka will support in its initial release, the ability to traffic shape the link between the CE and the PE in both directions and police at the PE into the core of the provider.

### 2.2.6 Generating configlets

Eureka generates configlets using the service object model. Configlets can be defined as the delta configuration needed to provision the service request on top of the existing router configuration. This obviously implies some knowledge of the pre-existing router configuration by Eureka. For this, Eureka schedules collections from the network and obtains the required configuration files from the

respective routers. A process within Eureka is able to convert the IOS configurations into object models for modeling and analysis. This eases Eureka's task which now compares two object models – the service object and the IOS configuration models, to evaluate the additional configuration needed to implement the service request. This additional configuration, the configlet is then generated.

### 2.2.7 Download

Eureka has a robust scheduler that allows for scheduling downloads to the routers. This allows batch downloads of service requests that can be scheduled for a non-peak period to be downloaded onto the routers. Eureka uses IP Manager to download<sup>84</sup> the configurations to routers. If any problems occur in the actual download, IP Manager will report the problem back to Eureka, which can then show a report of all service requests that were flagged because of download problems.

### 2.2.8 Auditing

Once the configurations have been downloaded, Eureka performs audit checks, which will verify the proper state of a service request through the audit reports it generates. The various states of a service request can be as depicted in Fig 32. A service is in the Pending state immediately after it has been created. The next step is to deploy the configuration that implements the service request. If this has been achieved, the service state is Deployed. Routes will appear in the appropriate VRF table and connectivity between the different VPN sites will exist. The service is now in the Functional state. If a problem occurs while trying to download the service request configlet, the state of the service request will continue to be Pending instead of Deployed. Additionally, based on a series of tests done by the auditing components, a configlet that was Functional that happens to not be exhibiting the connectivity it should, will

<sup>84</sup> Via TFTP.

thereby cause the audit sub-system to transition the service into a Broken state.

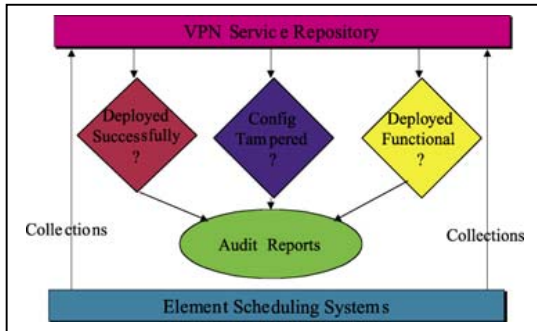


Figure 26-Service Auditing

Audit Reports can be viewed on a web browser. Auditing is achieved by first performing collections from the network. Information on network configuration and route tables is collected. By comparing the object model of the service requests against the object model of the network configurations, Eureka can verify the state of the each service. If the configuration deltas that implement the service exist in their correct form in the configuration collected, then the service has been deployed properly. If the VRF tables reflect connectivity between the various VPN sites then the service request is functional. These reports can be scheduled as tasks to run at different times and be organized by customers VPNs. Operators can take corrective action, based on these reports, in the event that a service request has a problem.

Once the service has been provisioned, there is a need to audit the services on an on-going basis – to keep state information on these services, and be able to provide Operations Management reports. Eureka 1.0 will perform the collection of a variety of data from the network elements. This information is analyzed against the services that are to be implemented. Each service state is then established with this audit.

Collection of data from the network is in the form of detailed Netflow information (which can yield information in the form of the Internet Data Records – equivalent to the Call Data Records [CDR] in the Old Telephony world). Performance data such as Round Trip Response Time (RTR) is also collected from the network elements. This will allow Eureka

to provide the relevant router performance data to third-party vendors such as Concord or Portal, so that Service Providers can obtain sophisticated reports on SLA and performance, which are important in a billing system. In addition, the system performs the collection of other data from the network elements that are of use in the analysis of the Service-to-Network correlation. Down-the-road, other products such as Infocenter may be used in conjunction with Eureka to perform fault correlation between the service and the affected network elements.

Service auditing has the ability to perform a routing audit, by verifying reachability amongst VPN sites. That is, it checks the appropriate VRF.

## 2.2.9 Reports Generated with Eureka 1.0<sup>85</sup>

### 2.2.9.1 Maximum Round Trip Time (RTT)

- (a) Per VPN
- (b) Per source router
- (c) Per customer
- (d) Per SLA ID
- (e) Per class of Service
- (f) In contract and out of contract

### 2.2.9.2 Percentage Connectivity of Devices

- (g) Per VPN
- (h) Per source router
- (i) Per customer
- (j) Per SLA ID
- (k) Per class of Service
- (l) In contract and out of contract

### 2.2.9.3 Delay Threshold Connectivity of Devices

- (m) Per VPN
- (n) Per source router
- (o) Per customer

<sup>85</sup> The author cautions the reader to not assume that all of these features will be available when Eureka 1.0 ships. Please check proper documentation, e-mail aliases, and/or with Product Marketing for more up-to-date information.

- (p) Per SLA ID
- (q) Per class of Service
- (r) In contract and out of contract

### 2.2.9.4 Netflow Statistics and Accounting

#### 2.2.9.4.1 Overview of the Netflow Collector

Netflow Collector is the software that gathers flow statistics from Cisco IOS devices. It is used for data collection, filtering and aggregation. The Netflow data is stored on the Netflow workstations in flat files.

Netflow Collector version 3.0 needs to be installed on a separate workstation and the workstation is connected directly to the PE device.

#### 2.2.9.4.2 Netflow Reports within Eureka

Netflow data will be periodically procured from the Netflow Collector workstation using Eureka. The data will be analyzed to create the following reports:

- (a) Accounting Summary Report
- (b) TOS Summary Report
- (c) PE to PE
  - (i) Traffic Between PE and Connected CEs
  - (ii) Traffic to CE by endpoints (ie CE to CE)
- (d) PE to Customer
  - (i) By Application
  - (ii) By Site
  - (iii) By CE
- (e) Customer
  - (i) Site to site
  - (ii) By application
  - (iii) By TOS

### 2.2.10 Eureka 1.0 Status

The product is undergoing Early Field Trials (EFT). The First Customer Shipment (FCS) is expected in September 1999.

## 3 Appendices - Standards; References; and Monitoring and Debugging Information

### 3.1 Appendix A – Cisco's MPLS Efforts

It is important to realize that Cisco is at the forefront of the standardization movement. Cisco invented

“tag switching” and then worked with the standards bodies and the networking community to develop an open standard based on tag switching.

### 3.1.1 MPLS Availability

- MPLS is identical to Tag Switching in all but the finest details. Specifically, Cisco's current MPLS implementation uses the Cisco Tag Distribution Protocol (TDP) instead of the Label Distribution Protocol (LDP).
- LDP is based on TDP, and is identical to it in purpose and general operation. LDP and TDP differ only in message formats, and in a few of the protocol procedures. TDP is openly published. Cisco is fully committed to LDP. An LDP implementation is being developed now, for field trials before the end of 1999.
- Once LDP ships, a Cisco router will be able to run LDP on some links while running TDP on others. This will provide full interoperability with standards while also providing backwards compatibility with prior releases of Cisco software. Since TDP and LDP differ so little, a fully functional MPLS network can readily operate with TDP running on some links, and LDP running on others.
- There is a smooth transition between TDP to LDP in a network. TDP or LDP runs independently on each link, so links may be changed from TDP to LDP one by one.
- At the May 1999 InterOp show in Las Vegas, Cisco demonstrated MPLS interoperability, utilizing LDP.

### 3.1.2 To CR-LDP or not to CR-LDP

Cisco is at the forefront of developing a standard for dynamically developing constraint-based traffic paths. Cisco put forth Routing for Resource Reservation (RRR) allowing Service Providers to take advantage of Traffic-Engineered paths as well as adjust to changing link conditions in a network. In fact, Cisco has software running in Service Provider trials performing that. RRR utilizes RSVP with extensions to perform the signaling for the

development of traffic paths. Cisco does not feel that it makes sense to develop a signaling protocol from scratch (CR-LDP), since it will require a lot more engineering effort and time for development, versus taking a proven protocol (RSVP), which is already a standard, and applying extensions to it to operate properly in a traffic-engineered environment, supporting Constraint-based routing.

### 3.1.3 Is MPLS a Standard Yet?

#### 3.1.3.1 Last Call for WG or IESG

There is not really a single “MPLS standard”—there are several different specifications which are the core parts of MPLS. These documents are either at Working Group Last Call, or at Internet Engineering Steering Group (IESG) Last Call now. They should pass Last Call by the middle of 1999. Once they have passed the Last Calls, all technical discussion are completed, and the documents become “Proposed Standard RFCs”. A Proposed Standard RFC, and sometimes just the earlier Internet Draft, are used as the basis for interoperability testing amongst vendors. The Proposed Standard RFC is changed only if issues arise during Inter-operability testing. Consequently, a Proposed Standard does have weight as “a standard”.

#### 3.1.3.2 MPLS Core Specifications

A. There are four main documents. These are Internet Drafts, and are available from “<http://www.ietf.org/html.charters/mpls-charter.html>.”

The documents are:

1. “MPLS Architecture”, draft-ietf-mpls-arch-04.txt
2. “MPLS Label Stack Encodings”, draft-ietf-mpls-label-encaps-03.txt
3. “MPLS using ATM VC Switching”, draft-ietf-mpls-atm-01.txt
4. “Label Distribution Protocol”, draft-ietf-mpls-ldp-03.txt

Three of these documents have already passed Working Group Last Call and are expected to turn into Proposed Standard RFCs as soon as they are cleared by the IESG. The Label Distribution Protocol is in Working Group Last Call now, and will soon go to IESG Last Call also.

In addition, the working group is working on other documents which describe optional extensions or

applications of MPLS. These are generally less advanced in the standards process.

Once three inter-operable implementations have been demonstrated, each MPLS document will become a Draft Standard. Once a technology reaches Draft Standard status, standardization of it is effectively complete.

#### 3.1.3.3 When is a Standard a Standard?

The IETF has a unique standards process which is different from other standards organizations. The IESG in the IETF grants “Internet Standard” status to a technology only after it has been widely deployed in the Internet and used for some years. So, a technology must be a standard for some years before it is finally granted “Internet Standard” status.

Many widely-used and effectively standardized protocols have never achieved Internet Standard status, or even Draft Standard status. The IS-IS IP routing protocol (RFC1195), for example, has been widely used for many years, but has never become a “Draft Standard,” let alone an “Internet Standard.” It is still a “Proposed Standard.” Another example of a “Proposed Standard” in wide use is Classless Inter-Domain Routing (RFC1518, RFC1519), which is used throughout the Internet, and has helped slow down the growth rate of the global Internet table.

### 3.1.4 Cisco’s MPLS Efforts - Summary

Let us separate fact from fiction! Cisco is the leading vendor of MPLS-based forwarding technologies, including MPLS-VPN and Traffic Engineering over MPLS. Other vendors, like Nortel and Lucent have responded to customer and market pressures to support MPLS. In the next few months, Cisco will officially come out with products that are MPLS- and LDP-compliant.

### 3.2 Appendix B – References

Reference Number	Title	Author(s)
RFC2547	<i>BGP/MPLS VPNs</i>	Eric Rosen, Yakov Rekhter
ENG-23056	<i>Tag-VPN Architecture</i>	Eric Rosen
RFC2283	<i>Multiprotocol Extensions for BGP-4</i>	T. Bates, R. Chandra, D. Katz, Y. Rekhter
Draft-ramachandra-bgp-ext-communities-01.txt	<i>BGP Extended Communities Attribute</i>	Srihari Ramachandra, Daniel Tappan
rrr.doc	<i>Routing with Resource Reservation (RRR)</i>	Jonathan Jiang
ENG-29568, Rev. E	<i>IOS CLI for BGP/MPLS VPN</i>	Dan Tappan
EFT DRAFT Doc	<i>VPNs for Tag Switching</i>	
ENG-37750 Rev 1.1	<i>RPM (MPLS) for MGX8850 Release 2.0+ Software Functional Specification</i>	Siu-Man Leung, Richard Lam, Loanne Cheung
Internal Document	<i>MPLS Standardization &amp; Cisco Q&amp;A</i>	Jeremy Lawrence

### 3.3 Appendix C – MPLS-VPN Platforms

#### 3.3.1 MPLS-VPN Functionality - Available Platforms<sup>86</sup>

Cisco IOS version 12.0(5)T “p” and “js” images are expected to FCS in late July of 1999 for the platforms below.

Platform	Image Name	Required DRAM
Cisco 7500 Series	rsp-jsv-mz	128 MB
	rsp-pv-mz	
Cisco 7200 Series	c7200-js-mz	128 MB
	c7200-p-mz	
Cisco 4500/4700 Series	c4500-js-mz	TBD
	c4500-p-mz	
Cisco 3640	c3640-js-mz	TBD
	c3640-p-mz	
Cisco 3620	c3620-js-mz	TBD
	c3620-p-mz	

<sup>86</sup> The reader is (here we go again) cautioned to verify release and feature availability with Product Marketing or the appropriate e-mail aliases.

A Cisco 7200 (in a standalone or bundled fashion), as well as the Cisco 7500 (in a standalone configuration) will also be available around the same time for the VSI and LSC functionality. In other words, in late July, the BPX will obtain MPLS functionality.

It is also expected that the RPM on the MGX platform will have LER functionality in CQ3'99, while LSR functionality on that integrated router module is anticipated to proceed to EFT status in that same time period.

With the exception of Fast Reroute, MPLS TE is expected in Cisco IOS 12.0(5)S, as well as 12.0(6)T.

Opaque OSPF as an IGP for MPLS TE support, is expected to be available in Cisco IOS 12.0(6)S.

### 3.3.2 GSR MPLS-VPN Support

It is expected that certain GSR line cards (LCs) will have MPLS-VPN functionality around the October timeframe. One definitely needs to contact the GSR Product Marketing team to inquire about MPLS-VPN availability on which LCs. With some LCs, a hardware limitation exists that prevents the creation of more than four FIBs/CEFs.

Generally speaking, newer LCs will have the capability to support MPLS-VPN, not just in terms of a higher number of FIBs supported, but also in terms of hardware-based QoS capabilities, whereby one does not pay a performance penalty when turning some QoS features on.

The author of this document was not able to obtain more information regarding which LCs are going to support MPLS versus MPLS-VPN, and in what timeframe.

### 3.3.3 MPLS Support in MSSBU Platforms

MPLS and MPLS-VPN support on the MSSBU platforms involves utilizing a Label Switch Controller and Routing control software.

#### 3.3.3.1 General MSSBU MPLS Support

The BPX 8650 is a bundle that includes the BPX 8620 ATM WAN switch and a Label Switch Controller based on a 7204/NPE 150 router. This combination turns the BPX 8650 into an ATM Label Switch Router.

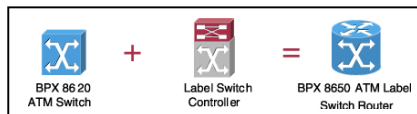


Figure 27 - "formula" for BPX 8640 ATM LSR

On the other hand, the MGX 8850 is a new ATM switch which is primarily intended to be an edge switch. It incorporates a multiservice access switch that incorporates a Cisco IOS full-featured router which runs the Edge LSR function. The first customer ship (FCS) of the MPLS function is expected in the first calendar quarter of 2000. Of course, engineering trials (EFT) and beta will occur earlier.

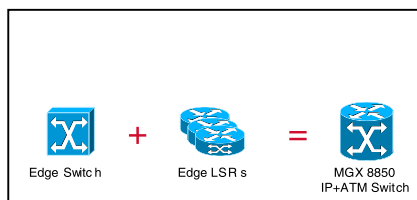


Figure 28 - MGX 8850 IP+ATM Switch

#### 3.3.3.2 The VSI Interface

VSI is an interface between the network-layer and platform connection software. It allows cross-connects to be established at a switch. VSI connects routing and signaling software, for example, PNNI or MPLS; to the MGX or BPX platform software, which actually establishes cross-connects and manages resources. So the network-layer software utilizes VSI to form cross-connects in the switch.

Figure 29 shows MPLS and IP routing processes with signaling (via VCs) between them. MPLS and IP routing handle the end-to-end signaling, deciding which connections need to be set up. Furthermore, at each point through the network, these control processes use VSI to set up connections, requesting that cross-connects and VCs be set up end-to-end across the network.

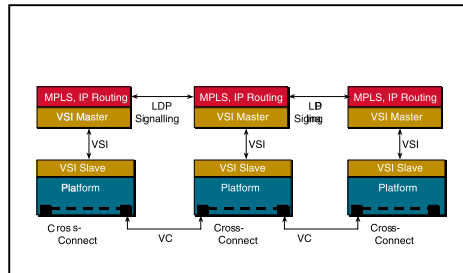


Figure 29 - VSI & End-to-End MPLS Signaling

### 3.3.3.3 VSI Resource Partitioning

The VSI is involved in allocating resources to different controllers. For example, if one has PNNI and MPLS as two different control planes, then PNNI sets up SVCs while MPLS establishes MPLS Label VCs (LVCs). This facilitates a certain range of cross-connect space to be allocated to SVCs, allowing it to set up a certain number of VCs on each link, while a certain range of that space will be allocated to MPLS LVCs, permitting a certain number of LVCs to be established on that link. Bandwidth as well as VPI and VCI space are also partitioned.

### 3.3.3.4 The BPX 8650

As indicated earlier, the BPX 8650 is currently available with a packaged Cisco 7204 router<sup>87</sup>. The bundled router acts as the Label Switch Controller and provide VSI Master support functions.

Although theoretically, one can utilize the BPX 8650 IP+ATM Switch as an MPLS-VPN PE entity, it is not recommended. The 7204 CPU can get easily overwhelmed having to handle VSI and LSC control traffic; MPLS label-handling and IGP protocol support; and data traffic also.

The BPX 8650 will serve very well as an MPLS-VPN backbone or P router, performing MPLS switching.

From a design perspective, one needs to also be aware of the bandwidth of the lone control link over which VSI runs, between the router and the BPX switch. It is strongly recommended that the link between the external router and the BPX 8620 be an ATM OC3 connection, and not a lower-bandwidth

connection<sup>88</sup>. All control and data traffic between the router and the switch has to pass over that single link. On that single link, labels have to be assigned for all prefixes for which traffic passes from frame-based to switch-based interfaces. And the number of labels and prefixes one can support is limited by the number of VCs that can actually be created on that control link. Depending upon the port adapter used, this is either 2K or 4K VCs.

### 3.3.3.5 MGX 8850 with the Route Processor Module

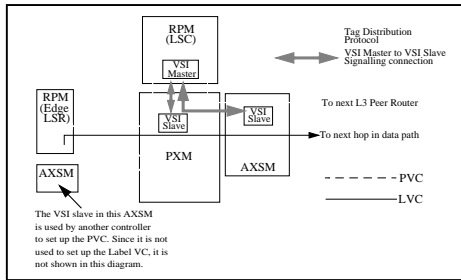
Typically an RPM functioning as an Edge LSR will have Frame Relay/ATM data from other Service Modules via a PVC, terminate on an RPM. The RPM will then translate the data and transport it as an LVC, through the PXM and on to the next hop in the data path (as highlighted in figure 30).

In this environment, the AXSM associates the data with a PVC. The other end of the PVC terminates at the RPM switch port, which is the interface between the router blade and the cell bus. This PVC is provisioned independently from MPLS.

The RPM receives packets and provides Layer 3 services. Based on the Layer 3 destination address, the RPM forwards the packet to an MPLS LVC. In this case, forwarding the packet involves segmenting the packet into ATM cells, and applying the label as the cell address (VPI/VCI). The LVC is established by the LSC (a separate RPM, which is the recommended design). In this scenario, the AXSM port is used for user access to the network.

<sup>87</sup> It is also possible to utilize an existing 7200 or 7500 router running the appropriate IOS image (e.g., IOS 12.0(5)T for MPLS-VPN and IOS 12.0(6)T for both MPLS-VPN and MPLS TE).

<sup>88</sup> This, in fact, is the setup in the BPX 8650. The recommendation pertains to customers procuring a 7200 or 7500 Series router separately and connecting it to the BPX 8620.



*Figure 30 - Typical RPM Deployment*

In both MGX Releases 1 and 2.0+, RPMs can be used as Edge LSRs to receive and label IP packets. Labeled packets can be forwarded to the other RPM Edge LSRs via PVCs or PVPs as shown in figure 31.

For ELSRs connected via a PVP, the label is transacted in the VCI field of an ATM cell. VCI 32 is used for the connection to run LDP. For PVC-connected ELSRs, the label is carried in the ATM payload.

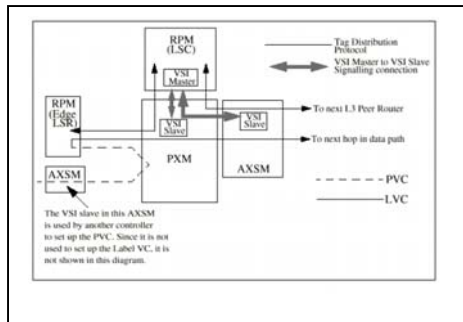
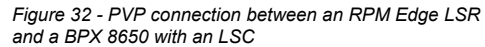
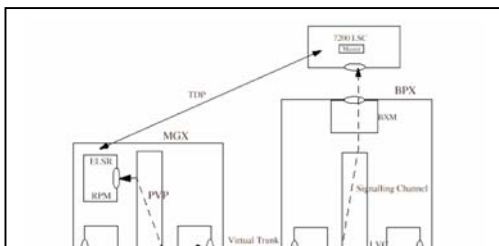


Figure 31 - PVP/PVC Connection between a pair of RPM ELSRs

Labeled packets can also be forwarded to a BPX 8650 with an LSC via Permanent Virtual Path (PVP) connection as shown in figure 32.



In Figure 32, the LSC software on the 7204 is configured to use a fixed VPI on the BXM port connected to the MGX to communicate with the MGX ELSR. The label is carried in the VCI field of an ATM cell and VCI 32 is used for the connection to run TDP. Since the 7204 is configured as an LSC, it can establish cross-connects at the VCI level so that LVCs can be switched directly between two MPLS-enabled ATM interfaces.

Unlabelled IP traffic can enter the RPM via the RPM back card or any ATM AXSM cards in the shelf.

The reader will note that the ability to run MPLS traffic over the RPM back-card ports (non-ATM MPLS) is supported in both MGX Releases 1 and Release 2.0+.

### 3.3.3.5.1 MGX Today - Edge LSR Functionality without the LSC

Figure 33 shows user data entering an MGX service module (Frame Relay), flowing on a PVC to an RPM acting as an Edge LSR, and then on to a PVP or PVC and on to the next hop in the data path.

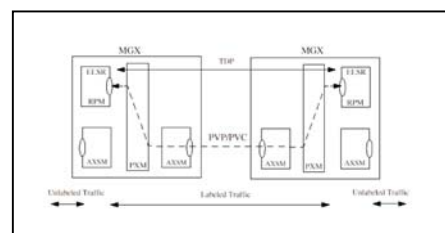


Figure 33 - RPM Functionality without FSC

In this case, the FRSM associates the data with a PVC. The other end of the PVC terminates at the RPM switch port. The RPM receives the packets and provides Layer 3 services. Based on the Layer 3 destination address, the RPM forwards the packet to a PVP or a PVC.

In the case where a PVP is used, the Edge LSR uses the VCI field in the ATM cell header for MPLS labels. The VPI value is specified statically when the PVP is provisioned.

In the case where a PVC is used, the Edge LSR labels the packet, and then segments it in to ATM cells. The VPI/VCI values are specified statically when the PVC is provisioned. Therefore, the label exists only in the “data” portion of the ATM cell.

It is advantageous to use a PVP rather than a PVC, to take advantage of the inter-working capability with a BPX 8650 (running the appropriate version of software).

In this case, the LSC at the BPX 8650 and the RPM Edge LSR are using LDP to negotiate labels. Because the connection between the 8650 and the Edge LSR is a PVP, the VPI is static, and the VCI is the negotiated label. The LSC establishes cross-connects in the 8650 so that the connections in the PVP are broken out and individually-switched. In this manner, the RPM Edge LSR acts as an MPLS “feeder” to the BPX 8650. This is not possible when a PVC is used, because the label does not exist in the ATM cell header.

### 3.3.3.5.2 MGX Futures - LSC Support

MGX Release 2.0+ will permit an RPM to function as an LSC. LSC will implement the Cisco VSI protocol to dynamically set up or tear down cross-connects in the MGX switch. This enables data traffic carried in a LVC to be switched between two MPLS-enabled ATM interfaces transparently without RPM’s involvement in the data path.

MGX Release 2.0+ is expected by April 2000. The MGX, as highlighted in figure 34, will then have

functionality similar to the external LSC that connects to the BPX today. Unlike the BPX 8650 however, the MGX will be appropriate for MPLS-VPN PE functionality, so long as one utilizes at least two RPMs - one for LSC functionality and another for the PE role.

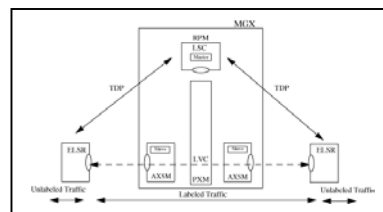


Figure 34 - MGX with LSC Support

RPM LSC will implement the Master side of the VSI protocol. Each MPLS-enabled interface will be represented by its associated VSI slave. There will be one VSI slave process per AXSM, and only one VSI slave process in the PXM45 representing all RPM switch ports within an MGX shelf.

The Edge LSR previously connected to a BPX 8650 using a PVP connection can now be connected to a LSC-controlled MGX.

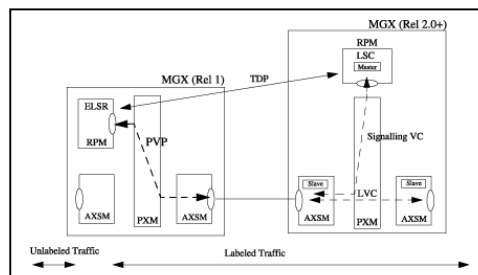


Figure 35 - PVP connection between an RPM Edge LSR and an RPM LSC

An LSC will coexist with other controllers on the same shelf. Resources will be partitioned among different controllers. For example, the resources of an ATM interface (AXSM/VISM) will be partitioned and assigned to different LSCs and PNNI independently.

Each LSC will run independently of the other LSCs.

### 3.3.4 12.0T and 12.0S Code Paths

MPLS-VPN features on the above-mentioned platforms will be available in Cisco IOS 12.05(T). MPLS-VPN along with MPLS Traffic Engineering<sup>89</sup> support will be available in Cisco IOS 12.0(6)T. Please check with Product Marketing regarding support in other Cisco IOS images.

## 3.4 Appendix D – Architecture of RRR<sup>†</sup>

This section provides a brief discussion on Routing with Resource Reservations (RRR, or R3). Traffic Engineering is an application that takes advantage of RRR. Moving forward, Cisco IOS commands will utilize the MPLS traffic engineering syntax versus “rrr.”

### 3.4.1 Introduction

The goal of traffic engineering is to maximize the utilization of network resources. In a large Service Provider network today, available network bandwidth is inefficiently utilized because for each destination, an intra-domain routing protocol (e.g., OSPF, IS-IS) finds a single “least-cost” route<sup>90</sup>. But what if this least-cost route is not the only possible one? Further, what if this link is over-subscribed, at least during certain times of the day? For example, in figure 36, there are two paths between the San Francisco router and the New York router: San Francisco-Chicago-New York; and San Francisco-Dallas-Atlanta-New York. The routing protocol decides that the former path is preferred and therefore all packets between these two points take the former path<sup>91</sup>. Even when the San Francisco-Chicago-New York path is congested, packets are not routed to the San Francisco-Dallas-Atlanta-New York path, which is not congested.

Figure 36 - The Traffic Engineering Problem

Due to this limitation, one often has the situation where a part of the network is over-utilized while another part is under-utilized. Traffic engineering attempts to address this issue<sup>92</sup>.

### 3.4.2 Traffic Engineering Case Study

Consider the following network:

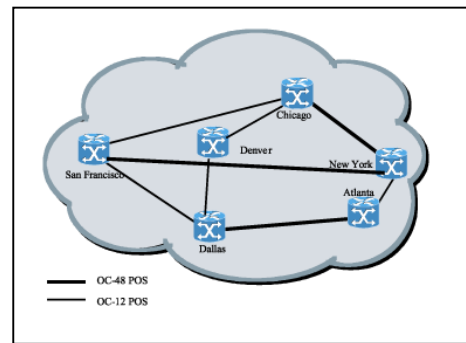
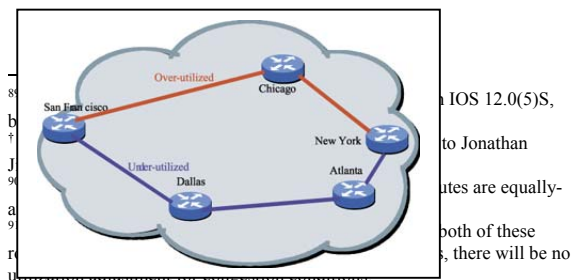


Figure 37 - Traffic Engineering Example Topology

As shown in Figure 37, this network has three OC-48 connections, between Chicago and New York; San Francisco and New York; and Dallas and Atlanta. The rest of the connections are OC-12 and OC-3. In this example, consider traffic engineering at the San Francisco node. One has the following traffic distribution information from the San Francisco node:



<sup>92</sup> Traffic engineering for a large IP network, in fact, has many more requirements. The reader is encouraged to read "draft-ietf-mpls-traffic-eng-00.txt"

Destination	Required Bandwidth	Path	Comments
Chicago	200 Mbps	SF-Chicago	SF-Chicago link – size: 655 Mbps, required: 700 Mbps
Denver	500 Mbps	SF-Chicago-Denver	
Dallas	300 Mbps	SF-Dallas	SF-Dallas link – size: 655 Mbps, required: 900 Mbps
Atlanta	600 Mbps	SF-Dallas-Atlanta	
New York	1 Gbps	SF-New York	SF-New York link – size: 2.2 Gbps, required: 1 Gbps

**Table 1 Traffic Distribution**

In the above table, the destination column lists the city pair (e.g., SF-Chicago, SF-Denver, etc.) under study. “Required bandwidth” refers to the amount of traffic expected to traverse the city pair. “Path” refers to the path chosen by the IGP. As indicated in the above table, without traffic engineering, the SF-Chicago link and the SF-Dallas link will experience congestion, while the SF-New York link is underutilized.

RRR can be the tool to re-engineer this network. At the San Francisco node, one configures the traffic trunk using the city pair bandwidth requirements in Table 1. In addition, one statically configures the path between SF and Denver to be SF-Chicago-Denver. The rest of the traffic trunks are dynamically calculated. The following table presents the traffic distribution after the data traffic has been engineered.

Destination	Required Bandwidth	Path	Comments
Chicago	200 Mbps	SF-New York-Chicago	SF-Chicago link – size: 655 Mbps, required: 500 Mbps
Denver	500 Mbps	SF-Chicago-Denver	
Dallas	300 Mbps	SF-Dallas	SF-Dallas link – size: 655 Mbps, required: 300 Mbps
Atlanta	600 Mbps	SF-New York -Atlanta	
New York	1 Gbps	SF-New York	SF-New York link – size: 2.2 Gbps, required: 1.8 Gbps

**Table 2. Result of Traffic Engineering**

As shown in Table 2, the SF-Chicago and SF-Atlanta traffic now traverse New York. As a result, there is no longer any congestion on the network. The SF-New York link is better utilized.

### 3.4.3 RRR Requirements

RRR makes use of several foundation technologies: MPLS, OSPF or IS-IS, and RSVP with extensions. Rather than explaining these technologies in detail<sup>93</sup>, the next few sections will briefly describe their roles in RRR environments while highlighting the protocol extensions required for traffic engineering.

#### 3.4.3.1 MPLS

MPLS provides:

- an efficient and graceful method to override destination-based forwarding,
- the mechanism to support nested explicit routes, and
- the ability to bind resources to paths, so that packets forwarded along LSPs are able to utilize the resources bound to LSPs.

Label switching paths are established by the traffic engineering procedures described above. Once IP packets enter the LSPs, they are guaranteed to be forwarded along the pre-determined path regardless of their IP headers. Readers are encouraged to review “draft-ietf-mpls-arch-02.txt” and “draft-ietf-mpls-framework-02.txt.”

#### 3.4.3.2 RSVP Extensions

The usage of RSVP in traffic engineering deviates from the original design goals of RSVP. There are several key differences:

- In traffic engineering, the sender, instead of the receiver determines the bandwidth required for each RSVP sessions.
- RSVP was designed to support both unicast and multicast, but the strong emphasis was placed on supporting multicast. In traffic engineering, only unicast is supported.
- The original intention of RSVP was to enable hosts to reserve bandwidth for applications such as multimedia. In the traffic engineering model, RSVP is used only by edge routers. In addition, maintaining an RSVP session for each application

on each host (i.e., a micro-flow) is clearly not scalable in a large network. In the traffic engineering model, RSVP sessions are created on a per-traffic-trunk basis. The number of traffic trunks will be manageable when reasonable micro-flow aggregation strategies are used.

- RSVP has no mechanism for routing signaling messages differently from normal IP packets. New RSVP objects are introduced in order to support traffic engineering. These objects are:
- Label object - used to carry the label information necessary for LSP creation.
- Explicit route object - used to specify the hop-by-hop path the PATH and RESV messages should follow.

The reader is encouraged to review “draft-ietf-mpls-rsvp-lsp-tunnel-02.txt,” and Jonathan Jiang’s RRR document.

#### 3.4.3.3 OSPF and IS-IS Extensions

Either OSPF or IS-IS needs to be able to carry resource information in their routing updates. In the case of OSPF, one uses opaque LSAs to carry the resource attributes<sup>94</sup>. In the case of IS-IS, a new type-length-value (TLV) entity is introduced to carry the resource attributes in the link state packets (LSPs). The reader is referred to “draft-ietf-isis-traffic-00.txt” for details.

So, to summarize, RRR uses:

- “Explicit” routing (aka “source routing”)
- MPLS as the forwarding mechanism
- RSVP(with extensions) as the mechanism for establishing Label Switched Paths (LSPs)
- Extensions to OSPF/IS-IS, to overcome limitations of the single IGP metric

<sup>93</sup> There are several excellent resources within Cisco discussing details of those inter-networking paradigms.

<sup>94</sup> FRC2370 defines the use of opaque LSAs. Traffic engineering extensions to OSPF are described in “draft-katz-yeung-ospf-traffic-00.txt”

### 3.4.4 Traffic Trunks and other RRR Traffic Engineering Paradigms

RRR traffic (more appropriate within a Service Provider environment) is a collection of traffic trunks with known bandwidth requirements.

Traffic trunks are aggregated micro-flows<sup>95</sup> that share a common path. In the context of this document, a “common path” does not refer to the end-to-end path of the flows, but a portion of the end-to-end path within the Service Provider’s network. Typically, the common path originates between the ingress and egress of the service provider’s Wide Area Network.

For example, all traffic originating from an IP address in San Jose and destined for an address in New York City may constitute a traffic trunk, while all traffic between an address in Palo Alto and an address in Washington D.C. another. Optionally, one may require that all packets within a traffic trunk have the same Class of Service. For example, all FTP and Telnet (priority 1) traffic between San Francisco and New York City may be considered a trunk, and all VoIP (priority 5) traffic between San Francisco and New York City another one.

In a nutshell, RRR creates one or more explicit paths with bandwidth assurances for each traffic trunk. It takes into consideration the policy constraints associated with the traffic trunks, and the physical network resources, as well as the topology of the network. This way, packets are no longer routed just based on destination, but also based on resource availability, and policy.

### 3.5 Appendix E – Application Note: MSSBU’s Demo Lab @ SP Base Camp Wk 2<sup>96</sup>

<sup>95</sup> A micro-flow refers to uni-directional packet transactions traveling from a source to a destination using the same transport protocol and the same port number. For example, an ftp session between two IP hosts constitutes two micro-flows - one from the client to the servers, and the other from the server to the client. Cisco’s original “NetFlow”

<sup>96</sup> Ripin Checker was kind enough to supply this information

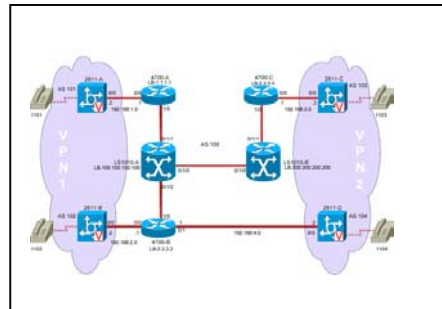


Figure 38 - MSSBU's Demo Setup, SP Bootcamp for SE's, March 22-26, 1999

### 3.5.1 Software Versions

#### 3.5.1.1 LS1010

IOS (tm) LS1010 W5-5 Software (LS1010-WP-M), Version 12.0(1a)W5(5b), RELEASE

SOFTWARE

#### 3.5.1.2 4700

IOS (tm) 4500 Software (C4500-JS-M), Experimental Version 12.0(19990211:021737) [BLD-bgp\_reorg.990210 114]

#### 3.5.1.3 2611

IOS (tm) C2600 Software (C2600-IS-M), Version 11.3(7)T, RELEASE SOFTWARE (fc1)

### 3.5.2 Configuration Examples

#### 3.5.2.1 LS1010-A

version 12.0

!

hostname LS1010-A

!

ip subnet-zero

!

atm address

47.0091.8100.0000.0050.e209.b801.0050.e209.b801.00

atm router pnni

```

no aesa embedded-number left-justified
node 1 level 56 lowest
redistribute atm-static
!
interface Loopback0
ip address 100.100.100.100 255.255.255.255
no ip directed-broadcast
!
interface ATM0/1/0
ip unnumbered Loopback0
no ip directed-broadcast
tag-switching ip
!
interface ATM0/1/1
ip unnumbered Loopback0
no ip directed-broadcast
tag-switching ip
!
interface ATM0/1/2
ip unnumbered Loopback0
no ip directed-broadcast
tag-switching ip
!
router ospf 100
network 100.100.100.100 0.0.0.0 area 100
!
ip classless
!
end

```

### 3.5.2.2 LS1010-B

```

version 12.0
!
hostname LS1010-B
!
ip subnet-zero
!
atm address
47.0091.8100.0000.0050.e20a.5b01.0050.e2
0a.5b01.00
atm router pnni
no aesa embedded-number left-justified
node 1 level 56 lowest
redistribute atm-static
!
interface Loopback0
ip address 200.200.200.200
255.255.255.255
no ip directed-broadcast
!
interface ATM0/1/0
ip unnumbered Loopback0
no ip directed-broadcast
tag-switching ip
!
interface ATM0/1/1
ip unnumbered Loopback0
no ip directed-broadcast
tag-switching ip
!
router ospf 100
network 200.200.200.200 0.0.0.0 area 100
!

```

```
ip classless
!  
end
```

### 3.5.2.3 4700-A

```
version 12.0
!  
hostname 4700-A
!  
ip subnet-zero
ip cef
!  
ip vrf vpn1 rd 100:1
ip vrf vpn1 route-target export 100:1
ip vrf vpn1 route-target import 100:1
!  
interface Loopback0
ip address 1.1.1.1 255.255.255.255
no ip directed-broadcast
!  
interface Ethernet0
ip vrf forwarding vpn1
ip address 192.168.1.1 255.255.255.0
no ip directed-broadcast
media-type 10BaseT
tag-switching ip
!  
interface ATM0
no ip address
no ip directed-broadcast
!  
interface ATM0.1 tag-switching
```

```
ip unnumbered Loopback0
no ip directed-broadcast
tag-switching ip
!  
router ospf 100
network 1.1.1.1 0.0.0.0 area 100
!  
router bgp 100
no synchronization
no bgp default ipv4-unicast
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 update-source Loopback0
neighbor 192.168.1.2 remote-as 101
!  
address-family ipv4 vrf vpn1
neighbor 192.168.1.2 activate
no auto-summary
no synchronization
no bgp default ipv4-unicast
exit-address-family
!  
address-family vpnv4
neighbor 2.2.2.2 activate
neighbor 2.2.2.2 send-community extended
no bgp default ipv4-unicast
exit-address-family
!  
ip classless
!  
end
```

### 3.5.2.4 4700-B

```

version 12.0
!
hostname 4700-B
!
ip subnet-zero
ip cef
ip vrf vpn1 rd 100:1
ip vrf vpn1 route-target export 100:1
ip vrf vpn1 route-target import 100:1
!
ip vrf vpn2 rd 100:2
ip vrf vpn2 route-target export 100:2
ip vrf vpn2 route-target import 100:2
!
interface Loopback0
ip address 2.2.2.2 255.255.255.255
no ip directed-broadcast
!
interface Ethernet0
ip vrf forwarding vpn1
ip address 192.168.2.1 255.255.255.0
no ip directed-broadcast
no ip mroute-cache
media-type 10BaseT
tag-switching ip
!
interface Ethernet1
ip vrf forwarding vpn2
ip address 192.168.4.1 255.255.255.0
no ip directed-broadcast
no ip mroute-cache

```

```

media-type 10BaseT
tag-switching ip
!
interface ATM0
no ip address
no ip directed-broadcast
no ip mroute-cache
!
interface ATM0.1 tag-switching
ip unnumbered Loopback0
no ip directed-broadcast
tag-switching ip
!
router ospf 100
network 2.2.2.2 0.0.0.0 area 100
!
router bgp 100
no synchronization
no bgp default ipv4-unicast
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 update-source Loopback0
neighbor 3.3.3.3 remote-as 100
neighbor 3.3.3.3 update-source Loopback0
neighbor 192.168.2.2 remote-as 102
neighbor 192.168.4.2 remote-as 104
!
address-family ipv4 vrf vpn1
neighbor 192.168.2.2 activate
no bgp default ipv4-unicast
exit-address-family
!
address-family ipv4 vrf vpn2

```

```
neighbor 192.168.4.2 activate
no bgp default ipv4-unicast
exit-address-family
!
address-family vpnv4
neighbor 1.1.1.1 activate
neighbor 1.1.1.1 send-community extended
neighbor 3.3.3.3 activate
neighbor 3.3.3.3 send-community extended
no bgp default ipv4-unicast
exit-address-family
!
ip classless
!
end
```

### 3.5.2.5 4700-C

```
version 12.0
!
hostname 4700-C
!
ip subnet-zero
ip cef
ip vrf vpn2 rd 100:2
ip vrf vpn2 route-target export 100:2
ip vrf vpn2 route-target import 100:2
!
interface Loopback0
ip address 3.3.3.3 255.255.255.255
no ip directed-broadcast
!
interface Ethernet0
```

```
ip vrf forwarding vpn2
ip address 192.168.3.1 255.255.255.0
no ip directed-broadcast
media-type 10BaseT
tag-switching ip
!
interface ATM0
no ip address
no ip directed-broadcast
!
interface ATM0.1 tag-switching
ip unnumbered Loopback0
no ip directed-broadcast
tag-switching ip
!
router ospf 100
network 3.3.3.3 0.0.0.0 area 100
!
router bgp 100
no synchronization
no bgp default ipv4-unicast
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 update-source Loopback0
neighbor 192.168.3.2 remote-as 103
!
address-family ipv4 vrf vpn2
neighbor 192.168.3.2 activate
no auto-summary
no synchronization
no bgp default ipv4-unicast
exit-address-family
!
```

```

address-family vpnv4
neighbor 2.2.2.2 activate
neighbor 2.2.2.2 send-community extended
no bgp default ipv4-unicast
exit-address-family
!
ip classless
!
end

```

### 3.5.2.6 2611-A

Using 490 out of 29688 bytes

```

!
version 11.3
!
hostname 2611-A
!
dial-peer voice 1000 pots
destination-pattern 14085551101
port 1/0/0
!
dial-peer voice 200 voip
destination-pattern 14085551102
session target ipv4:192.168.2.2
!
dial-peer voice 300 voip
destination-pattern 14085551103
session target ipv4:192.168.3.2
!
dial-peer voice 400 voip
destination-pattern 14085551104
session target ipv4:192.168.4.2
!

```

```

num-exp 1101 14085551101
num-exp 1102 14085551102
num-exp 1103 14085551103
num-exp 1104 14085551104
!
voice-port 1/0/0
!
voice-port 1/0/1
!
voice-port 1/1/0
!
voice-port 1/1/1
!
interface Ethernet0/0
ip address 192.168.1.2 255.255.255.0
!
router bgp 101
network 192.168.1.0
neighbor 192.168.1.1 remote-as 100
!
ip classless
!
end

```

### 3.5.2.7 2611-B

Using 510 out of 29688 bytes

```

!
version 11.3
!
hostname 2611-B
!
dial-peer voice 1000 pots
destination-pattern 14085551102

```

```
port 1/0/0
!
dial-peer voice 100 voip
destination-pattern 14085551101
session target ipv4:192.168.1.2
!
dial-peer voice 300 voip
destination-pattern 14085551103
session target ipv4:192.168.3.2
!
dial-peer voice 400 voip
destination-pattern 14085551104
session target ipv4:192.168.4.2
!
num-exp 1101 14085551101
num-exp 1102 14085551102
num-exp 1103 14085551103
num-exp 1104 14085551104
!
voice-port 1/0/0
!
voice-port 1/0/1
!
voice-port 1/1/0
!
voice-port 1/1/1
!
interface Ethernet0/0
ip address 192.168.2.2 255.255.255.0
!
router bgp 102
no synchronization
```

```
network 192.168.2.0
neighbor 192.168.2.1 remote-as 100
!
ip classless
!
end
```

### 3.5.2.8 2611-C

```
version 11.3
!
hostname 2611-C
!
dial-peer voice 1000 pots
destination-pattern 14085551103
port 1/0/0
!
dial-peer voice 100 voip
destination-pattern 14085551101
session target ipv4:192.168.1.2
!
dial-peer voice 200 voip
destination-pattern 14085551102
session target ipv4:192.168.2.2
!
dial-peer voice 400 voip
destination-pattern 14085551104
session target ipv4:192.168.4.2
!
num-exp 1101 14085551101
num-exp 1102 14085551102
num-exp 1103 14085551103
num-exp 1104 14085551104
!
```

```
voice-port 1/0/0
!
voice-port 1/0/1
!
voice-port 1/1/0
!
voice-port 1/1/1
!
interface Ethernet0/0
ip address 192.168.3.2 255.255.255.0
!
router bgp 103
network 192.168.3.0
neighbor 192.168.3.1 remote-as 100
!
ip classless
!
end
```

### 3.5.2.9 2611-D

```
version 11.3
!
hostname 2611-D
!
dial-peer voice 1000 pots
destination-pattern 14085551104
port 1/0/0
!
dial-peer voice 100 voip
destination-pattern 14085551101
session target ipv4:192.168.1.2
!
dial-peer voice 200 voip
```

```
destination-pattern 14085551102
session target ipv4:192.168.2.2
!
dial-peer voice 300 voip
destination-pattern 14085551103
session target ipv4:192.168.3.2
!
num-exp 1101 14085551101
num-exp 1102 14085551102
num-exp 1103 14085551103
num-exp 1104 14085551104
!
voice-port 1/0/0
!
voice-port 1/0/1
!
voice-port 1/1/0
!
voice-port 1/1/1
!
interface Ethernet0/0
ip address 192.168.4.2 255.255.255.0
!
router bgp 104
network 192.168.4.0
neighbor 192.168.4.1 remote-as 100
!
ip classless
!
end
```