

**CISCO SYSTEMS**



# **Cisco ISP Essentials**

***Essential IOS Features Every ISP  
Should Consider***

***Lessons from people who have been operating  
backbones since the early days of the Net***

*Version 2.9*

Wednesday, June 06, 2001

---



---

# TABLE OF CONTENTS

---



---

<b>TABLE OF CONTENTS .....</b>	<b>3</b>
<b>LIST OF FIGURES .....</b>	<b>9</b>
<b>ACKNOWLEDGEMENTS .....</b>	<b>10</b>
<b>CHANGES FROM THE PREVIOUS PUBLISHED VERSION .....</b>	<b>11</b>
SOFTWARE AND ROUTER MANAGEMENT .....	11
GENERAL FEATURES.....	11
SECURITY.....	11
ROUTING.....	12
REFERENCES & APPENDICES.....	12
<b>INTRODUCTION.....</b>	<b>13</b>
<b>SOFTWARE AND ROUTER MANAGEMENT.....</b>	<b>14</b>
WHICH IOS VERSION SHOULD I BE USING? .....	14
<i>Where to Get Information on 12.0S.....</i>	15
<i>Further Reference on IOS Software Releases .....</i>	15
IOS SOFTWARE MANAGEMENT .....	16
Flash Memory .....	16
System Memory .....	17
When and How to Upgrade?.....	18
Copying new images to FLASH Memory .....	18
Copying using TFTP .....	18
Copying using FTP.....	20
CONFIGURATION MANAGEMENT .....	20
NVRAM, TFTPserver and FTPserver .....	20
Large Configurations.....	21
DETAILED LOGGING.....	22
Syslog Topologies.....	24
Analysing Syslog Data.....	24
NETWORK TIME PROTOCOL (NTP).....	24
NTP Architecture.....	25
Client/Server Models and Association Modes.....	26
Implementing NTP on an ISP's Routers.....	26
NTP Deployment Examples .....	27
NTP in a PoP (Example).....	27
Further NTP References.....	29
SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP).....	29
SNMP in read-only mode.....	30
SNMP in read-write mode.....	31
SNMP and commercial network management software.....	31
HTTP SERVER .....	32
CORE DUMPS.....	32
<b>GENERAL FEATURES .....</b>	<b>34</b>
COMMAND LINE INTERFACE.....	34
Editing Keys .....	34
CLI String Search.....	34

Wednesday, June 06, 2001

IOS AND LOOPBACK INTERFACES.....	35
Background.....	35
BGP Update-Source.....	36
Router ID .....	36
IP Unnumbered Interfaces.....	36
Exception Dumps by FTP.....	37
TFTP-SERVER Access .....	37
SNMP-SERVER Access .....	37
TACACS/RADIUS-Server Source Interface.....	38
NetFlow Flow-Export.....	38
NTP Source Interface.....	38
SYSLOG Source Interface.....	39
Telnet to the Router.....	39
RCMD to the router .....	40
INTERFACE CONFIGURATION.....	40
Description.....	40
Bandwidth.....	40
IP Unnumbered.....	41
Caveats .....	41
A Full Example .....	41
INTERFACE STATUS CHECKING.....	42
“Show Interface Switching”.....	42
“Show Interface Stats”.....	43
CISCO EXPRESS FORWARDING (CEF) .....	44
NET FLOW.....	44
NetFlow Feature Acceleration.....	45
Netflow Statistics – Basics.....	45
Netflow Data Export.....	46
TURN ON NAGLE.....	47
DNS AND ROUTERS .....	47
Mapping IP addresses to Names.....	47
DNS Resolver in IOS .....	48
<b>SECURITY.....</b>	<b>49</b>
SECURITY FOR AN ISP.....	49
SECURING THE ROUTER.....	50
Global Services that are not needed or are a security risk.....	50
Interface Services That Are Not Needed or Are A Security Risk.....	51
Cisco Discovery Protocol.....	51
Login Banners.....	52
Use enable secret.....	53
The Ident Feature.....	54
Router Access – Controlling who can get into the Router.....	54
Principles .....	54
VTY and Console Port Timeouts.....	54
Access List on the VTY Ports.....	55
VTY Access and SSH .....	56
User authentication.....	56
Using AAA to Secure the Router.....	57
Router Command Auditing .....	58
Full Example .....	59
SECURING THE ROUTING PROTOCOL.....	60
Authenticating Routing Protocol Updates.....	60
Benefits of Neighbour Authentication .....	60
Protocols That Use Neighbour Authentication .....	60
When to Configure Neighbour Authentication .....	61

How Neighbour Authentication Works.....	61
Plain Text Authentication.....	62
MD5 Authentication.....	62
Routing Protocol Authentication Summary .....	62
<b>SECURING THE NETWORK.....</b>	<b>63</b>
<i>Egress and Ingress Filtering.....</i>	<i>63</i>
<i>Route Filtering.....</i>	<i>64</i>
Route Filtering: Networks that should not be advertised on the Internet.....	64
Effects of CIDRisation.....	66
<i>Packet Filtering.....</i>	<i>69</i>
<i>Access Control List – General Sequential Based ACLs.....</i>	<i>70</i>
<i>Access Control Lists – Turbo ACLs .....</i>	<i>71</i>
Turbo ACLs Configuration Details and References .....	71
<i>Using ACLs for Egress Packet Filtering.....</i>	<i>72</i>
<i>Using ACLs for Ingress Packet Filtering.....</i>	<i>74</i>
<i>Black Hole Routing as a Packet Filter (Forwarding to Null0).....</i>	<i>76</i>
<i>Unicast RPF – (Reverse Path Forwarding).....</i>	<i>77</i>
RPF Configuration Details (as of IOS Version 12.0(10)S1) .....	79
ACL Option (added in IOS Version 12.0(10)S1).....	80
debug ip cef drops rpf.....	82
Routing Tables Requirements .....	82
Unicast RPF Exceptions.....	83
<i>Implementing Unicast RPF.....</i>	<i>83</i>
<i>Unicast RPF for Service Providers and ISPs.....</i>	<i>83</i>
Single Homed Lease Line Customers .....	83
NAS Application – Applying Unicast RPF in PSTN/ISDN PoPs .....	84
Multihomed Lease Line Customers (one ISP) .....	85
Multihomed Leased Line Customers (two ISPs).....	87
<i>Unicast RPF for Enterprise Networks.....</i>	<i>88</i>
Single Homed Enterprise Networks – Filtering Incoming Traffic.....	88
Multi-Homed Enterprise Networks to One Upstream ISP – Filtering Incoming Traffic.....	89
Multi-Homed Enterprise Networks to Multiple Upstream ISP – Filtering Incoming Traffic.....	90
Where not to use Unicast RPF.....	91
<i>Unicast RPF Examples – Putting it all together.....</i>	<i>91</i>
<i>Other Considerations.....</i>	<i>92</i>
<b>COMMITTED ACCESS RATE (CAR) TO RATE LIMIT OR DROP PACKETS.....</b>	<b>92</b>
<i>Why use a QoS Tool for Security?.....</i>	<i>92</i>
<i>Passive SMURF Defences.....</i>	<i>93</i>
<i>Active SMURF Defences .....</i>	<i>93</i>
Rate Limiting with CAR .....	93
<b>REACTING TO SECURITY INCIDENTS.....</b>	<b>95</b>
<b>ISP SECURITY REFERENCES AND DEEPENING MATERIALS .....</b>	<b>96</b>
<b>ROUTING.....</b>	<b>97</b>
<b>HOT STANDBY ROUTING PROTOCOL.....</b>	<b>97</b>
<b>CIDR FEATURES .....</b>	<b>98</b>
<b>SELECTIVE PACKET DISCARD.....</b>	<b>98</b>
<b>IP SOURCE ROUTING.....</b>	<b>99</b>
<b>CONFIGURING ROUTING PROTOCOLS.....</b>	<b>100</b>
<i>Router ID .....</i>	<i>100</i>
<i>Choosing an IGP .....</i>	<i>101</i>
<i>Configuring an IGP.....</i>	<i>101</i>
<i>Putting Prefixes into the IGP .....</i>	<i>101</i>
The Network Statement.....	101
Redistribute Connected into an IGP.....	102
Redistribute Static into an IGP.....	102
Redistribute <anything> into an IGP .....	102

Wednesday, June 06, 2001

<i>IGP Summarisation</i> .....	102
<i>IGP Adjacency Change Logging</i> .....	102
<i>Putting Prefixes into BGP</i> .....	103
The Network Statement.....	103
Redistribute Connected into BGP.....	103
Redistribute Static into BGP.....	103
Redistribute <anything> into BGP.....	104
THE BGP PATH SELECTION PROCESS.....	104
<i>The BGP Best Path Algorithm for IOS</i> .....	105
BGP FEATURES AND COMMANDS.....	106
<i>Stable iBGP Configuration</i> .....	107
<i>BGP Auto Summary</i> .....	107
<i>BGP Synchronisation</i> .....	108
<i>BGP Community Format</i> .....	108
<i>BGP Neighbour Shutdown</i> .....	108
<i>BGP Dynamic Reconfiguration</i> .....	109
Soft-Reconfiguration.....	109
Route Refresh.....	109
<i>BGP Route Reflectors (RR) and the BGP cluster-id</i> .....	110
<i>Next-Hop-Self</i> .....	111
External connections.....	111
Aggregation routers.....	112
<i>BGP Route Flap Damping</i> .....	112
Background.....	112
Command Syntax.....	112
Implementation.....	113
Designing Flap Damping Parameters.....	114
BGP Flap Statistics.....	115
<i>BGP Neighbour Authentication</i> .....	116
<i>BGP MED not set</i> .....	116
<i>BGP Deterministic MED</i> .....	116
<i>Comparing Router-IDs</i> .....	117
<i>BGP Network Statement</i> .....	117
<i>Removing Private ASes</i> .....	117
<i>BGP local-as</i> .....	118
Configuration.....	118
Motivation.....	119
<i>BGP Neighbour Changes</i> .....	119
<i>Limiting the Number of Prefixes from a Neighbour</i> .....	119
<i>BGP Fast External Fallover</i> .....	120
<i>BGP Peer-group</i> .....	120
Background.....	120
Requirements.....	120
Historical Limitations.....	120
Typical Peer-group Usage.....	121
BGP Peer-Group Examples.....	121
<i>Using Prefix-list in Route Filtering</i> .....	121
Introduction.....	121
Configuration Commands.....	122
Command Attributes.....	122
Configuration Examples.....	123
How Does Match Work.....	124
Show and Clear Commands.....	124
Using Prefix-list with BGP.....	124
Using Prefix-list in Route-map.....	124
Using Prefix-list in Other Routing Protocols.....	125
<i>BGP filter processing order</i> .....	126
<i>BGP Conditional Advertisement</i> .....	126

Example.....	127
<i>BGP Outbound Route Filter Capability</i> .....	128
Configuration.....	129
Pushing out a Prefix-list ORF .....	129
Displaying Prefix-list ORF.....	129
<i>BGP Policy Accounting</i> .....	130
Overview .....	130
Configuration.....	130
Displaying BGP Policy Accounting status.....	131
Displaying BGP Policy Accounting statistics.....	131
<b>FURTHER STUDY AND TECHNICAL REFERENCES .....</b>	<b>132</b>
SOFTWARE AND ROUTER MANAGEMENT .....	132
GENERAL FEATURES.....	133
SECURITY.....	134
ROUTING.....	135
OTHER REFERENCES AND RECOMMENDED READING.....	135
<b>GLOSSARY .....</b>	<b>137</b>
<b>APPENDIX 1 – ACCESS LIST AND REGULAR EXPRESSIONS .....</b>	<b>139</b>
ACCESS LIST TYPES .....	139
IOS REGULAR EXPRESSIONS .....	140
<b>APPENDIX 2 – CUT AND PASTE TEMPLATES .....</b>	<b>141</b>
GENERAL SYSTEM TEMPLATE.....	141
GENERAL INTERFACE TEMPLATE.....	141
GENERAL SECURITY TEMPLATE.....	141
GENERAL BGP TEMPLATE.....	141
MARTIAN AND RFC1918 NETWORKS TEMPLATE.....	142
<i>IP Access-List Example</i> .....	142
<i>IP Prefix-List Example</i> .....	142
<b>APPENDIX 3 – EXAMPLE CONFIGURATIONS .....</b>	<b>143</b>
SIMPLE NETWORK PLAN.....	143
CONFIGURATIONS.....	143
<i>ISP Addressing Plan</i> .....	144
<i>Border Router</i> .....	144
<i>Core Router</i> .....	148
<i>Aggregation Router</i> .....	151
<i>Service Router</i> .....	154
<i>NOC Router</i> .....	157
<i>Access Server</i> .....	160
<i>Out-of-band Console Server</i> .....	163
<b>APPENDIX 4 – ROUTE FLAP DAMPING.....</b>	<b>167</b>
BGP FLAP DAMPING CONFIGURATION .....	167
<i>IP Access-List Example</i> .....	167
<i>IP Prefix-List Example</i> .....	168
<b>APPENDIX 5 – TRAFFIC ENGINEERING TOOLS .....</b>	<b>170</b>
INTERNET TRAFFIC AND NETWORK ENGINEERING TOOLS.....	170
<i>CAIDA</i> .....	170
<i>NetScarf/Sicon</i> .....	170
<i>NeTraMet/NetFlowMet</i> .....	170

Wednesday, June 06, 2001

<i>Cflowd</i> .....	170
<i>MRTG</i> .....	171
<i>RRDTool</i> .....	171
<i>Vulture</i> .....	171
<i>CMU SNMP</i> .....	171
<i>UCD SNMP (the successor to CMU SNMP)</i> .....	172
<i>Gnuplot</i> .....	172
<i>NETSYS</i> .....	172
<i>SysMon</i> .....	173
<i>Treno</i> .....	173
<i>Scotty – Tcl Extensions for Network Management Applications</i> .....	173
<i>NetSaint</i> .....	173
<i>Big Brother</i> .....	173
OTHER USEFUL TOOLS TO MANAGE YOUR NETWORK.....	174
<i>Traceroute</i> .....	174
<i>Whois</i> .....	174
<i>RTRMon – A Tool for Router Monitoring and Manipulation</i> .....	174
<i>Cisco’s MIBs</i> .....	174
<i>Replacement SYSLOG Daemons</i> .....	174
<i>SYSLOG-NG</i> .....	174
<i>SECURE SYSLOG (ssyslog)</i> .....	175
OVERALL INTERNET STATUS AND PERFORMANCE TOOLS.....	175
<i>NetStat</i> .....	175
WHAT OTHER ISPS ARE DOING... ..	175
<b>APPENDIX 6 – EXAMPLE ISP ACCESS SECURITY MIGRATION PLAN .....</b>	<b>178</b>
PHASE ONE – CLOSE OFF ACCESS TO EVERYONE OUTSIDE YOUR CIDR BLOCK.....	178
PHASE TWO – ADD ANTI-SPOOFING FILTERS TO YOUR PEERS.....	179
<i>Where to place the anti-spoofing packet filters?</i> .....	180
PHASE THREE – CLOSE OFF NETWORK EQUIPMENT TO UNAUTHORISED ACCESS.....	181

---



---

## LIST OF FIGURES

---



---

Figure 1 – IOS Roadmap.....	16
Figure 2 – Tailing a Centralised syslog file for a Cisco Router.....	24
Figure 3 – Typical Internet POP Built for Redundancy and Reliability using the core routers as NTP servers.....	28
Figure 4 – Netflow in it's capacity planning role.....	45
Figure 5 – ISP's Security Threats come from all Directions.....	49
Figure 6 – Command Auditing on the Router via Cisco Secure and TACACS+.....	59
Figure 7 – Router Authentication.....	61
Figure 8 – How Routing Protocol Authentication Operates.....	62
Figure 9 – Ingress and Egress Filtering.....	64
Figure 10 – URLs to the RIR's Default Allocation Sizes.....	68
Figure 11 – Standard Access List – Incoming Packets.....	70
Figure 12 – Extended ACLs – Logical Flow for Every Packet.....	71
Figure 13 – Turbo ACLs – Processing takes five steps no matter what the size.....	72
Figure 14 – Turbo ACL's Performance Consistency.....	72
Figure 15 – Egress Packet Filtering on the Upstream Gateway Router.....	73
Figure 16 – Egress Packet Filtering on the Customer Edge.....	74
Figure 17 – Ingress Packet Filtering on the Upstream Gateway Router.....	75
Figure 18 – Ingress Packet Filtering on the ISP ↔ Customer Edge (IETF BCP Recommendation).....	76
Figure 19 – Using static host routes to null 0 for black list filtering.....	77
Figure 20 – Unicast RPF validating IP source addresses.....	78
Figure 21 – Unicast RPF dropping packets which fail verification.....	78
Figure 22 – Unicast RPF Drop Counter.....	80
Figure 23 – Unicast RPF applied to Leased Line Customer Connections.....	84
Figure 24 – Unicast RPF applied to PSTN/ISDN Customer Connections.....	84
Figure 25 – Multihomed Lease Line Customer & Unicast RPF.....	85
Figure 26 – Enterprise Customer Multihomed to two ISPs.....	88
Figure 27 – Enterprise Network using Unicast RPF for Ingress Filtering.....	89
Figure 28 – Multi-Homed Enterprise Networks.....	90
Figure 29 – How asymmetrical routing would not work with Unicast RPF.....	91
Figure 30 – How SMURF uses amplifiers.....	94
Figure 31 – Dual gateway LAN.....	97
Figure 32 – BGP Route Reflector Cluster.....	110
Figure 33 – BGP Route Reflector Cluster with two RRs.....	111
Figure 34 – BGP Route Flap Damping.....	114
Figure 35 – BGP Conditional Advertisement – Steady State.....	127
Figure 36 – BGP Conditional Advertisement – Failure Mode.....	128
Figure 37 – Network Used for Configuration and Filtering Sample.....	143
Figure 38 – ISP Network Example.....	179
Figure 39 – Applying Anti-Spoofing Filters.....	180
Figure 40 – Closing off access to everyone except the NOC Staff.....	182

Wednesday, June 06, 2001

---

---

## ACKNOWLEDGEMENTS

---

---

We would like to thank the many people who have contributed to this document, making numerous suggestions, contributions, clarifications, and who have provided their deep real world operational experience with the Internet. Their willingness to help others *do the right thing* is one of the reasons for the Internet's success.

Any comments, questions, updates, or corrections should be sent to:

Barry Raveendran Greene  
Philip Smith

[bgreene@cisco.com](mailto:bgreene@cisco.com)  
[pfs@cisco.com](mailto:pfs@cisco.com)

---

---

## CHANGES FROM THE PREVIOUS PUBLISHED VERSION

---

---

The previous published version of IOS Essentials was 2.84. This section lists the major changes and additions which have been made since the previous version.

### *Software and Router Management*

1. The “Which IOS” section (page 14) has been updated to be consistent with Cisco recommendations (using 12.0+ software), and also our experience with service providers several of our colleagues have been working with.
2. Copying files to and from the router using FTP has been added (page 20). Using the FTPserver has also been added for router configuration management (page 20).
3. The syslogging section has been extended to include an example of a Unix system configuration (page 22). The section also discusses syslog topologies and strategies to ensure continuous collection of syslog information (page 24).
4. NTP section has been extended to discuss deployment of NTP in a PoP (page 27).
5. SNMP section has been extended to discuss commercial network management software and it’s interaction with network equipment (page 31).

### *General Features*

1. New section on checking the status of router interfaces (page 42).
2. New section describing Cisco Express Forwarding (CEF) (page 44).
3. New section describing NetFlow Feature acceleration (page 45).
4. Discussion of NAGLE has been moved from the Security section (it was misplaced in an earlier version) to this section (page 47).

### *Security*

The section on security has undergone a major rewrite and we encourage all ISP network and operations engineers to reread the security section in its entirety. The major changes are:

1. Discussion of proxy -arp (page 51).
2. How to secure the routing protocol (neighbour authentication, etc.) (page 60).
3. Major rewrite and expansion of the section describing how to secure the network (page 63). This includes the new TurboACL feature now available in IOS (page 71).
4. New section describing how to implement Unicast Reverse Path Forwarding as a network security tool (page 77). This includes configuration details and deployment examples.
5. Updated sections on CAR and how to deal with security incidents in the network (page 92).

Wednesday, June 06, 2001

## ***Routing***

1. Detailed documentation of the BGP Path Selection Process as found in IOS (page 104).
2. Rewrite and expansion of the description of the BGP Flap Damping process (page 112), including several configuration examples.
3. Description of the BGP Deterministic MED feature (page 116).
4. Description of the BGP compare-routerid “knob” (page 117).
5. The historical limitations of the BGP network statement (which still seems to be stuck in folklore) (page 117).
6. Section describing the order of processing of BGP filters (page 126).

## ***References & Appendices***

1. A completely new section has been written (page 132) gathering on one place all the reference and technical reading advised throughout IOS Essentials. Also, all the URLs have been updated (as content on CCO tends to move around fairly frequently).
2. A new appendix has been written (page 143) giving a working example of typical implementation of IOS commands in a small ISP network. Hopefully this is useful for ISPs who are newer to IOS, and would like a “quick start” to getting their network up and running.
3. Appendix 4 has had the route flap damping configuration updated (due to an error in the original parameters) (page 167).
4. Appendix 5 has been updated (page 170).

---

---

# INTRODUCTION

---

---

Cisco Systems have built a tremendous range of features into the IOS. The extensive feature set is excellent for public network service providers, giving their network engineers a large number of options and capabilities which can be designed into the network. While a huge number of features maybe desirable, it also poses a problem – network engineers busy running their networks have a difficult time keeping up with all the new IOS features. Many do not know how, when, and where to deploy the various features in their network. Experienced network engineers building the Internet are not exempt. This document has been written to highlight many of the key IOS features in everyday use in the major ISP backbones of the world. Judicious study and implementation of these *IOS pearls* will help to prevent problems, increase security, improve performance, and ensure the operational stability of the Internet.

***NOTE: This document and its recommendations focus on Internet Service Providers – not the general Internet population. This point of view needs to be understood by the person using the techniques described in this document for their network.***

This paper has four major sections as well as several appendixes that give the reader further information, tips, and templates relating to what has been covered in the paper. These major sections are:

- Software and Router Management
- General Features
- Security
- Routing

If you have questions on any of the materials in this document, please refer to the following:

- Cisco System's Documentation. (available free via <http://www.cisco.com/univered/>)
- Cisco Connection On-line. (<http://www.cisco.com>)
- Local Cisco Systems' support channels,
- Public discussion lists. The list that specifically focuses on ISPs who use Cisco Systems equipment is *Cisco NSP* – hosted by Jared Mauch at Nether.Net<sup>1</sup>.

An up to date copy of this document can be found at <http://www.cisco.com/public/cons/isp>. This page also has related material referenced in this document, plus other information which may be useful for Internet Service Providers.

---

<sup>1</sup> CISCO NSP is a mailing list which has been created specifically by Internet Service Providers to discuss Cisco Systems products and their use. To subscribe, send an e-mail to: [majordomo@puck.nether.net](mailto:majordomo@puck.nether.net) with a message body containing: *subscribe cisco-nsp*

---

---

## SOFTWARE AND ROUTER MANAGEMENT

---

---

### *Which IOS version should I be using?*

ISPs and NSPs operate in an environment of constant change, exponential growth, and unpredictable threats to the stability of their backbone. The last thing an Internet backbone engineer needs is buggy or unstable code on their routers. As in any commercial grade service providing infrastructure, the equipment forming that infrastructure requires stable operating software. Stable software needs to have rapid updates with fixes to bugs that have been identified. This stable code needs to have the latest features – critical to their operations – added long before the rest of Cisco’s enterprise customers see them. (Herein is a key difference between enterprise and Internet service provision. The former demands stability and change only when necessitated, while the latter demands stability, yet market leadership when it comes to new features.) ISPs need to access this stable code via the Internet with out the traditional hassles of obtaining a software upgrade. Bottom line is that ISPs require an IOS code train specific to their needs.

This is exactly what has happened. Mid way through the life of the 10.3 software train, Cisco created a branch of IOS that catered specifically to ISPs’ requirements. The key characteristics were IP-centric code base, stability, quick bug fixes, easy access to the key development engineers, and rapid feature additions. The so-called “isp-geeks” software started life as an unofficial ISP software issue, but with the arrival of the 11.1 software train, has matured and developed into a release system specifically targeted at Internet Service Providers. As IOS becomes more and more feature rich, this ISP software train has been further developed and enhanced and now provides a very well developed and stable platform for all Internet Service Providers.

Along with the development of specific IOS images for ISPs, the Service Provider image was added to all Cisco IOS software released. This software is based on the IP-only image but with additional features for Service Providers. Such software can be recognised by the “-p-” in the image name. This image is usually all that any ISP needs to run. These images cannot be ordered at time of router purchase but can be downloaded from CCO prior to deployment of the router in service. For example, a 7200 which an ISP orders may come with the **c7200-i-mz.120-6** image – this image should be replaced with the Service Provider equivalent, **c7200-p-mz.120-6**. These Service Provider “-p-” images are built for all supported router platforms unlike the more limited platform support available on the ISP release trains.

At the time of writing, the recommended IOS branches for ISPs are:

- ✓ **12.0S** – supporting the 7200, RSP7000, 7500 and 12000 series routers.
- ✓ **12.0** – supporting 2500, 2600, 3600, and 4000/4x00 series routers.<sup>2</sup>
- ✓ **12.1** – supporting the new hardware platforms which are not supported by 12.0 (such as 3660).

At the time of writing, releases 11.1CC and 11.2GS are no longer recommended for service provider backbones as they do not support the current generation of hardware in use, nor will they be enhanced to support new hardware or software features. For example, 11.1CC has gained no new features since 11.1(26)CC. Releases prior to 12.0 are now coming to the end of their life, although still supported by Cisco, they will not gain any new features. Migration from these old releases should be part of the ongoing upgrade planning process in all service provider networks at the moment.

**Cisco Systems’ most up to date recommendations on which IOS branch an ISP should be using will be on our Product Bulletin page available via CCO:**

<http://www.cisco.com/warp/public/cc/general/bulletin/index.shtml>

---

<sup>2</sup> Yes, there are many ISPs in the world whose entire backbone is built on 2500s!

## Where to Get Information on 12.0S

12.0S is now available via CCO's Software Library <http://cco.cisco.com/kobayashi/sw-center/sw-ios.shtml>. The following URLs have some additional details on the features included in 12.0S, migration options, and how to download.

**Cisco IOS Software Release 12.0S New Features**

[http://www.cisco.com/warp/public/cc/pd/iosw/iore/iomjre12/prodlit/934\\_pb.htm](http://www.cisco.com/warp/public/cc/pd/iosw/iore/iomjre12/prodlit/934_pb.htm)

**Cisco IOS Software Release 12.0S Ordering Procedures and Platform Hardware Support**

[http://www.cisco.com/warp/public/cc/pd/iosw/iore/iomjre12/prodlit/935\\_pb.htm](http://www.cisco.com/warp/public/cc/pd/iosw/iore/iomjre12/prodlit/935_pb.htm)

**Cisco IOS Software Release Notes for Release 12.0S**

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/relnote/7000fam/rn120s.htm>

**Cisco IOS 12.0S Migration Guide**

[http://www.cisco.com/warp/public/cc/pd/iosw/iore/iomjre12/prodlit/940\\_pb.htm](http://www.cisco.com/warp/public/cc/pd/iosw/iore/iomjre12/prodlit/940_pb.htm)

## Further Reference on IOS Software Releases

Figure 1 provides a visual map of IOS releases and how the different versions and trains inter-relate. The following URLs on CCO will have more detailed and possibly up to date information on IOS release structure:

**Cisco IOS Releases**

<http://www.cisco.com/warp/public/732/Releases/>

**Types of Cisco IOS Software Releases**

[http://www.cisco.com/warp/customer/cc/pd/iosw/iore/prodlit/537\\_pp.htm](http://www.cisco.com/warp/customer/cc/pd/iosw/iore/prodlit/537_pp.htm)

**Release Designations Defined - *Software Lifecycle Definitions***

<http://www.cisco.com/warp/customer/417/109.html>

**Software Naming Conventions for IOS**

<http://www.cisco.com/warp/customer/432/7.html>

**IOS Reference Guide**

<http://www.cisco.com/warp/public/620/1.html>

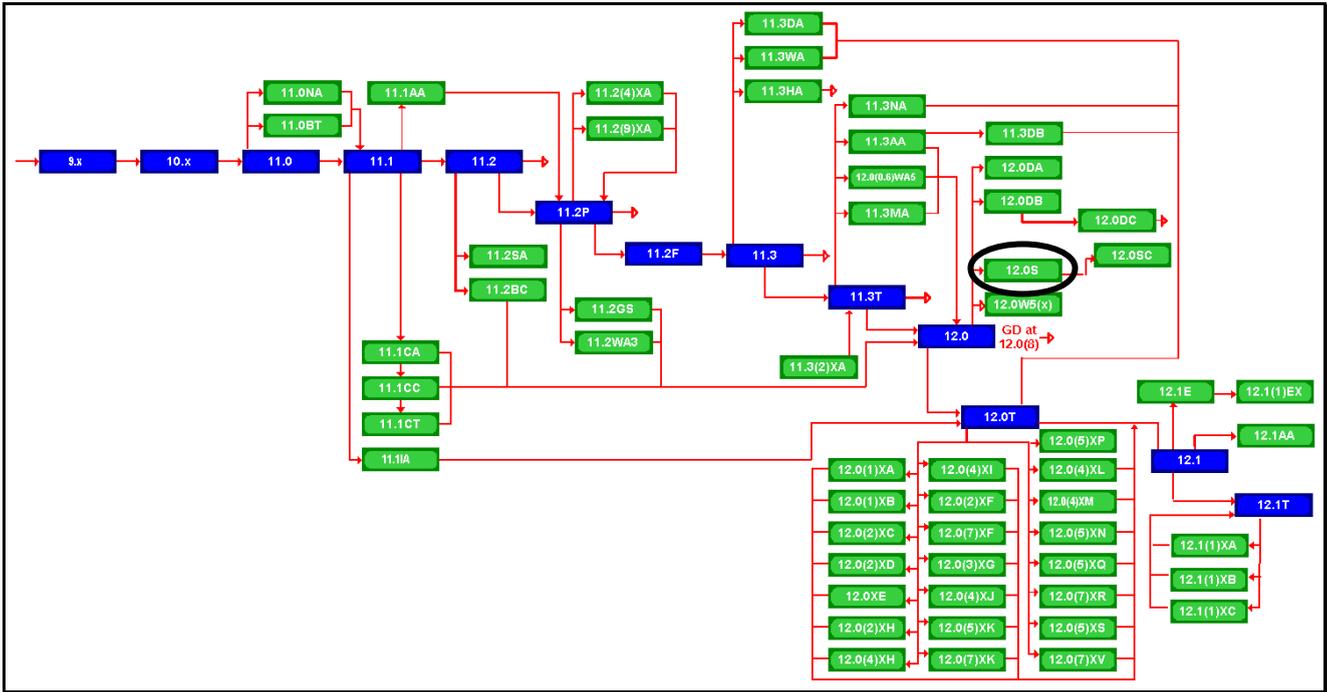


Figure 1 – IOS Roadmap<sup>3</sup>

## IOS Software Management

Most router platforms used in ISP backbone networks have very flexible RAM and FLASH memory configurations. For private, enterprise, or campus networks, the number of changes required in software, new features, or even the network infrastructure is small. The Internet is changing daily, growing daily, and a common mistake by new ISPs is to under specify the equipment they purchase. This should not be taken as a recommendation to buy what may seem like unneeded memory. It is recognition of the fact that having to upgrade a router every few months due to “unforeseen” growth in the Internet causes disruption to the network, and can potentially affect the reliability of hardware. Many Internet engineers support the view that “the more often humans touch a piece of hardware, the less reliable the hardware becomes”.

## Flash Memory

The flash memory on a router is where the IOS software images are stored. When a new router is purchased, it will have the IOS image specified at time of ordering installed in flash. Flash memory is usually built into the router, and some platforms have expansion slots where PCMCIA flash cards can be installed.

It is good practice to supplement the built-in flash with another area of flash memory. This can be done in at least two ways:

1. Partition the built-in flash memory. This can be done using the configuration command, for example:

```
partition flash 2 8 8
```

which will partition the flash into two areas of 8Mbyte each (assuming 16Mbytes of installed flash memory, and the hardware supports this type of partitioning).

2. Install a flash card in one or both of the external flash slots.

<sup>3</sup> Check <http://www.cisco.com/warp/public/620/roadmap.shtml> for updates to this roadmap.

Having more than one flash memory area ensures that the router IOS image can be upgraded without affecting the existing image. For example, if there is only room for one image in flash, and it is the image the router is running, the existing image would have to be removed before a new one could be installed. What would happen, say, if the router crashed during the image upgrade? Recovery is possible with the boot image, but this is significantly harder than if proper precautions had been taken. By copying the new image into the other area of flash memory, the ISP ensures that the network functionality is minimally affected in the event of a crash or other unforeseen problems during image upgrade.

The new image in the second area of flash memory can easily be selected, as shown in the following example for the 7x00 series routers:

```
boot system flash slot1:rsp-pv-mz.120-5.S
boot system flash slot0:rsp-pv-mz.111-25.CC
boot system flash
```

which tells the router to boot `rsp-pv-mz.120-5.S` from slot1 flash first. If that image cannot be found or the flash card is not in slot1, it looks for `rsp-pv-mz.111-25.CC` in slot0. If that image cannot be found, the router boot software looks for the first image in any of the system flash.

Or this example, on the 36x0 series routers where the main 16Mb flash has been partitioned:

```
boot system flash flash:1:c3640-p-mz.120-5.S
boot system flash flash:2:c3640-p-mz.112-19.P
boot system flash
```

which tells the router to boot `c3640-p-mz.120-5.S` from the first flash partition. If the router cannot find that image, it will try to look for `c3640-p-mz.112-19.P` in the second flash partition. Failing that, it looks for the first usable IOS image in flash memory.

This type of arrangement ensures that in the event of image corruption, or a problem with the operating image, or router crash, there is always some backup image available on the router. Proper precautions ensure minimal network down time during maintenance periods or emergency occasions. Downloading a new image during a network down situation with customers and management exerting pressure is unnecessary extra stress which could easily have been avoided with a little precaution.

Common practice is for ISPs to leave the known working image on one of the flash cards or flash partitions of the router. In the event of deployment of a new release (which has passed tests in the lab environment) exhibiting some problem or defect later in operation, it is easy to back track to the old image.

Finally, it makes no commercial or operational sense to skimp on size of flash memory. As more and more of the features requested by ISPs are added to IOS, the image grows larger and larger. Sizing flash to the current image size is a false economy because it is more than likely that in the near future a larger image with new features may require flash memory larger than has been installed in the router.

## System Memory

Another common practice amongst the Tier 1 and Tier 2 ISPs in all regions of the world is maximising the memory on every router. Cisco recommends the necessary amount of memory required to run each IOS image. Downloading a new image from CCO includes a question, which has to be answered, asking the user if they are fully aware of the memory requirements for the chosen image. Ignore the minimum recommendations at your peril!

For example, at the time of writing, it is recognised that 128Mbytes of memory is the minimum requirement to operate a router carrying the full Internet routing table. And any ISP requesting IOS 12.0S is required to acknowledge this fact. IOS 12.0S will still operate inside 32Mbytes of memory on a 7200 router, and will carry majority of the Internet routes with 64Mbytes memory (with minimal configuration and all features turned off). For example, the BGP algorithms will use more memory if it is available to improve their efficiency. Skimping on memory means affecting the performance of the routers, and the end result which the customer experiences.

Wednesday, June 06, 2001

Indeed many ISPs now simply specify maximum memory when they purchase new routing hardware. They recognise that sending an engineer to remove processor cards costs money through downtime, extra human resources, potential service disruption, and shortens the lifetime of the equipment through the human interaction. "Fit and Forget" is the norm amongst many of the largest ISPs today.

## When and How to Upgrade?

Several ISPs upgrade their router software almost every time Cisco releases a new image. However, the only time any ISP should be upgrading software is when they require to fix bugs, support new hardware, or implement new software features. In many other industries, changing core operating software is seen as a major event, not undertaken lightly. Yet for some reason, some ISPs seem to think that a fortnightly upgrade is good practice.

Again, based on what most Tier 1 and Tier 2 ISPs now do, software upgrades are carried out only when they are required. Extensive testing is carried out in their test lab (how many ISPs have a test network which looks like one of their PoPs, or portion of their network?). Deployment only happens after extensive testing, and even then new images are implemented with caution on a quieter part of the network. Caution is of paramount importance on a commercial grade network. Even when upgrades are carried out, remember the recommendations above. IOS makes it easier by giving back-out paths via alternative images. Never attempt an upgrade without being aware of potential side-effects due to unforeseen problems; never attempt an upgrade without a back-out plan.

Another practice implemented by most Tier 1 and Tier 2 ISPs is to minimise the number of different versions of IOS images running on their network's routers. This is almost always due to administrative and management reasons. Apart from reducing the number of potential interoperability issues due to bugs and new features, it is easier to train operations staff on the features and differences between few images than it is to train them on the differences between many images. Typically ISPs aim to run no more than two different IOS releases. One image is the old release, the other is the one which they are doing the blanket upgrade on their backbone. Upgrades tend to be phased, not carried en-masse overnight. If the ISPs have access equipment, such as the AS5x00 series, or the cable/xDSL aggregation devices they will deploy different IOS images on these devices. But again, if one dial box needs to be upgraded, ISPs tend to upgrade them all to ensure a consistent IOS release on that network.

## Copying new images to FLASH Memory

Copying a new image into Flash memory in itself isn't a complicated process, but there are a few good practice points to be aware of. The most important point here is to re-emphasise that leaving a back out image somewhere on the router is good practice and plain common sense. So many network outages have been prolonged because a new router image failed and the ISP hadn't left a back out image on the device.

New images should be loaded into flash during maintenance periods, not when the router is live carrying full traffic load. While the activity of copying an image to flash won't impact the router's operation, it is advisable to avoid enhancing the possibility of an accident while the network is in production. At least an operational error during a maintenance period won't cause customer anger as they were expecting downtime during the maintenance period in any case (assuming the customers were informed in the first place, another key point several ISPs seem to forget!).

## Copying using TFTP

The commands to copy software into flash memory have been refined in releases from 12.0, making the mechanics of getting software to and from the router simpler and more consistent in style. The "copy" command has been enhanced to support a URL appearance covering all system devices in a consistent format, for example as:

```
beta7200#copy tftp ?
bootflash:      Copy to bootflash: file system
disk0:          Copy to disk0: file system
disk1:          Copy to disk1: file system
flash:          Copy to flash: file system
ftp:            Copy to ftp: file system
```

```

lex:          Copy to lex: file system
null:        Copy to null: file system
nvram:       Copy to nvram: file system
rcp:         Copy to rcp: file system
running-config Update (merge with) current system configuration
slot0:       Copy to slot0: file system
slot1:       Copy to slot1: file system
startup-config Copy to startup configuration
system:      Copy to system: file system
tftp:        Copy to tftp: file system

```

This is somewhat improved over the rather inconsistent and platform dependent format used in previous releases.

Before copying the image to flash, make sure the flash has enough space for the new image. Preferably install two flash devices, or partition the existing flash device, as has been described previously. Use “cd”, “delete”, “erase” and “squeeze” (depending on platform) to clear enough space on the flash file system. Make sure that the partition/flash holding the currently running image is not touched. If there is any problem with the upgrade, a back out path is available. And don’t forget to set up the “*boot system xxxxx*” IOS configuration command so that the router is told to boot the currently running image.

Once the flash partition is ready, a copy command can be issued to copy the new IOS image from the remote device (could be any of those listed above) to the partition. An example of the copy command is given below:

```

beta7200#copy tftp slot1:
Address or name of remote host []? noc1
Source filename []? 12.0S/c7200-p-mz.120-6.S
Destination filename [c7200-p-mz.120-6.S]?
Accessing tftp://noc1/12.0S/c7200-p-mz.120-6.S...
Loading 12.0S/c7200-p-mz.120-6.S from 192.168.3.1 (via Serial3/1):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 5708964/11417600 bytes]

5708964 bytes copied in 330.224 secs (17299 bytes/sec)
beta7200#

```

This will copy the image *c7200-p-mz.120-6.S* from the tftp server to the flash device in slot1. Or the command can be shortened to:

```

beta7200#copy tftp://noc1/12.0S/c7200-p-mz.120-6.S slot1:
Destination filename [c7200-p-mz.120-6.S]?
...etc...

```

which will achieve the same thing. Notice that the router will attempt to work out the file name from the URL string entered – this can be helpful and save typing. When successfully verified, set up the router to boot the new image (use the “no boot system” and “boot system” commands described earlier) – it is also a good idea to configure another boot system command pointing to the backup image (as in the example in the earlier section). If desirable, the router can even be configured to do a timed/delayed reboot at some time in the future. Use that feature with care though; it is perfectly feasible to do timed reboots on several routers and completely break a portion of the ISP network! An example:

```
beta7200# reload at 17:05
```

will reboot the router at 17:05 router time.

When the new image has been booted successfully, it should be put through acceptance tests during the maintenance slot (could be as simple as “does the routing work?”, for example), and then monitored during operation. Don’t delete the previous image – you know it works so if it is left on the other flash partition a back out path is available in case of future problems. The old image can be deleted once a decision has been made to further upgrade the IOS. The benefit of configuring two flash devices/partitions is clear to see – ease of maintenance!

Wednesday, June 06, 2001

## Copying using FTP

FTP client support has been added in 12.0 images of IOS. TFTP has well known limitations – hence the requirement for FTP client support on the router. This allows for FTPing images to and from a FTP server. The following is an example of a software download directly from Cisco’s FTP site – something that cannot be done via TFTP (the password is XXX out):

```
7206-AboveNet-SJ2#copy ftp://bgreene:XXX@ftp.cisco.com slot0:
Source filename []? /cisco/ios/12.0/12.0.9S/7200/c7200-k3p-mz.120-9.S.bin
Destination filename [c7200-k3p-mz.120-9.S.bin]?
Accessing ftp://bgreene:XXX@ftp.cisco.com //cisco/ios/12.0/12.0.9S/7200/c7200-k3p-mz.120-
9.S.bin...Translating "ftp.cisco.com"...domain server (207.126.96.162) [OK]

Loading /cisco/ios/12.0/12.0.9S/7200/c7200-k3p-mz.120-9.S.bin
```

**CAVEAT:** It’s important to be aware that on the 2500 series routers, the image runs from flash, so a reload of the router to run the BOOTROM image is required to upgrade. The BOOTROM image does not support ftp copies of the IOS image onto the router, even though it is possible to enter the command sequences above – the BOOTROM attempts to upload the image using TFTP, it’s only supported functionality.

**Summary: upgrade only when bug fixes, or new hardware support, or new software features are required. Otherwise, do not touch that router! “If it isn’t broken, don’t fix it.”**

## Configuration Management

### NVRAM, TFTPserver and FTPserver

The onboard router NVRAM is used to store the router’s active configuration. Most ISPs keep an off-router copy of this configuration too. In the unlikely event of the configuration being lost on the router, they can quickly recover the system with the off-router backup. There are several options for off-router backup of the running configuration:

- Write configuration to a tftp server using the “write net” command. (In 12.0 and more recent software the “write net” command has been superseded with the more sophisticated copy function. The equivalent command is “copy running tftp:”.)
- Configurations saved by an operator’s “write net” command are kept under automated revision control. Combined with TACACS+ authentication (see later) it is possible to track whom has changed what and when. Important for accountability and configuration back-out in case of problems.
- An automated (e.g. Unix cron) job collects the configuration from the router every few hours or so. The collected copy is kept under revision control. Changes could be flagged to the network monitoring system, or to the NOC, or to the operations engineers.
- Router configurations are built from a master configuration database. The running configuration is only a copy, with the master configuration kept off-router. Updates to the running configuration are made by altering the master files (under revision control) and implementing the new master configuration during maintenance periods.

**Note:** See the Chapter discussing loopback interfaces for best practice for configuring the router for tftp services.

The IOS command prompts to save the configuration are given in the following examples. The syntax has been significantly changed starting with IOS 12.0, mainly to make the commands used to transfer configuration files and IOS software between operator/NOC and the router more consistent. The IOS command prior to 12.0 is given in the following example:

```
alpha7200#write network
Remote host []? noc-pc
Name of configuration file to write [alpha7200-config]?
Write file router2-config on host 192.168.3.1? [confirm]
```

```
Building configuration...

Writing alpha7200-config !!! [OK]
alpha7200#
```

From 12.0 onwards, the command to do the same thing is *copy*, as given in the following example:

```
beta7200#copy running tftp:
Remote host[]? noc-pc
Destination filename [beta7200-config]?
Write file tftp://noc-pc/beta7200-config? [confirm]
!!! [OK]
beta7200#
```

**Note:** the “write network” command is still supported in 12.0, although may be withdrawn in a future release.

Since 12.0, FTP can also be used to copy configurations to an FTP server. This provides more security for the configurations, since the FTP server will require a username/password. Since a username/password is required for FTP to work, Cisco provides two ways of provide the username/password to the FTP client.

The first way puts the username and password as part of the IOS configuration. With *service password-encryption* turned on, the ftp password would be stored with encryption type 7

```
ip ftp source-interface Loopback 0
ip ftp username user
ip ftp password quake
```

This allows the FTP command transparently insert the username/password when connection to the FTP server.

```
Excalabur#copy running-config ftp:
Address or name of remote host []? 1.13.13.13
Destination filename [excalabur-config]?
Writing excalabur-config !!
3803 bytes copied in 3.848 secs (1267 bytes/sec)
Excalabur#
```

The alternative is to include the username/password in a standard URL format:

```
Excalabur#copy running-config ftp://user:quake@1.13.13.13
Address or name of remote host [1.13.13.13]?
Destination filename [excalabur-config]?
Writing excalabur-config !!
3800 bytes copied in 4.152 secs (950 bytes/sec)
Excalabur#
```

Either way works.

## Large Configurations

When the NVRAM is not large enough to store the router configuration there is an option which allows the configuration to be compressed (using a gzip like compression algorithm):

```
service compress-config
```

Only use this if there is a requirement to. If the existing NVRAM can hold the configuration uncompressed, do not use this feature. Some ISPs have extremely large configurations and this feature was introduced to assist them.

Furthermore, if the router configuration has become very large it is worth checking whether some of the newer IOS features can be used. One example would be using prefix-lists instead of access-lists; the former is more space efficient in NVRAM, and also is more efficient in operation.

Wednesday, June 06, 2001

## Detailed Logging

Keeping logs is a common and accepted operation practice. Interface status, security alerts, environmental conditions, CPU process hog and many other events on the router can be captured and analysed via UNIX syslog. Cisco System's IOS has the capability to do UNIX logging to a UNIX syslog server. The router syslog format is compatible with BSD UNIX syslog (now found as part of most Unix and Linux systems deployed today). The follow is a typical logging configuration for ISPs:

```
no logging console          <-- Don't send logs to the router console
logging buffered 16384      <-- 16Kbyte history buffer on router
logging trap debugging     <-- Catch debugging level traps (i.e. everything)
logging facility local7    <-- Syslog facility on syslog server
logging 169.223.32.1       <-- IP address of your first syslog server
logging 169.223.45.8      <-- IP address of your second syslog server
```

To set up the syslog daemon on a UNIX system, include a line such as the following in the file /etc/syslog.conf:

```
local7.debugging    /var/log/cisco.log
```

It is considered good practice to reserve a syslog facility on the Unix log host for each type of network device. So, for example, backbone routers may use "local7", access servers may use "local5", tacacs+ may use "local3", etc. An example of a working syslog configuration file is given below:

```
# Log all kernel messages to the console.
kern.*                                /var/log/kern.log

# Log everything apart from the specific entries given after this line
*.debug;kern.none;mail.none;authpriv.none;local7.none\
  local5.none;local4.none;local3.none  /var/log/messages

# The authpriv file has restricted access.
authpriv.*                             /var/log/secure

# Log all the mail messages in one place.
mail.*                                  /var/log/maillog

# Everybody gets emergency messages.
*.emerg                                 *

# Save mail and news errors of level err and higher in a special file.
uucp,news.crit                          /var/log/spooler

# Cisco Access server log
local7.*                                /var/log/cisco-core-log

# Cisco Access server log
local5.*                                /var/log/cisco-access-log

# NetFlowLog
local4.*                                /var/log/netflowlog

# TACACS+
local3.*                                /var/log/tacacs+

# Save boot messages to boot.log
local1.*                                /var/log/boot.log
```

Putting all the logs in one huge file simply makes system management hard, and debugging problems by searching the log file next to impossible. More modern Unix platforms will require network support in the syslog daemon to be enabled by a runtime option (network support is now disabled by default to try and avoid security problems). Check that you see the "-r"

in the syslog command line as in the following example when you list the process on a Unix system (the example is for RedHat Linux):

```
pfs-pc$ ps ax | grep syslog
 496 ?      S        0:04 syslogd -r
20853 pts/5  S        0:00 grep  syslog
pfs-pc$
```

By default, log messages are not time stamped. If you do configure your routers for UNIX logging, you will want detailed timestamps of for each log entry:

```
service timestamps debug datetime localtime show-timezone msec
service timestamps log datetime localtime show-timezone msec
```

which will produce a syslog message looking something similar to:

```
Jul 27 15:53:23.235 AEST: %SYS-5-CONFIG_I: Configured from console by philip on console
```

The command line options in the timestamps command are as follows:

- debug: all debug info is time stamped
- log: all log info is time stamped
- datetime: the date and time is include in the syslog message
- localtime: the local time is used in the log message (as opposed to UTC)
- show-timezone: the timezone defined on the router is included (useful if the network crosses multiple time-zones)
- msec: time accuracy to milliseconds – useful if NTP is configured.

By default, a syslog message contains the IP address of the interface it uses to leave the router. You can require all syslog messages to contain the same IP address, regardless of which interface they use. Many ISPs use the loopback IP address. This keeps their syslogs consistent plus allows them to enhance the security of their SYSLOG server host (by the use of TCP wrappers or router filters for example).

```
logging source-interface loopback0
```

**Note:** See the Chapter discussing loopback interfaces for best practice for configuring the log hosts for syslog services.

Another command to consider is the *no logging console* command. There are times when logging generates a tremendous amount of traffic on the console port. Of course this happens just when you really need to connect to the console port to troubleshoot what is causing the tremendous load of messages. Hence, it is good practice to turn console logging off. This will keep the console port free for maintenance.

In review, with all the options used, a typical ISP logging configuration would look like the following:

```
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
!
no logging console
logging buffered 16384
logging trap debugging
logging facility local7
logging 169.223.32.1
logging 169.223.45.8
logging source-interface loopback0
!
```

Wednesday, June 06, 2001

## Syslog Topologies

ISPs use one of two syslog topologies in their networks. The first is a traditional centralised syslog infrastructure. The centralised topology has the syslog servers in a central location supporting all the logging flows from the network devices in the network. This is located in one of the ISP's major data centres or the ISP's NOC. The advantage with this topology is that all the raw syslog data is quickly available in one location. The disadvantage of this topology is one of scaling the amount of syslog data coming to one location.

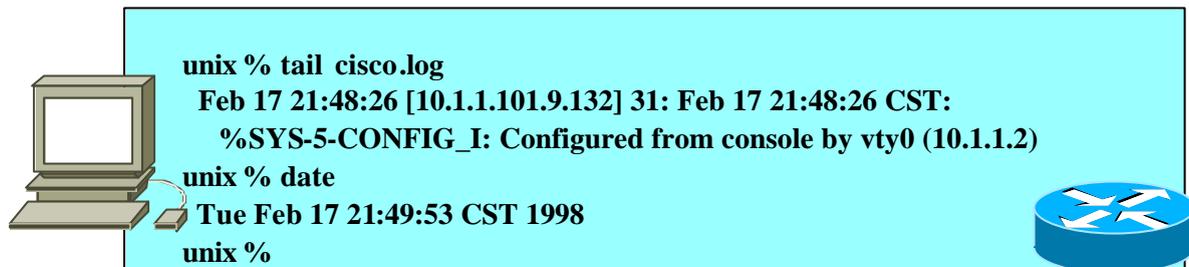


Figure 2 – Tailing a Centralised syslog file for a Cisco Router

The second topology is a distributed topology. Syslog servers are pushed out to the edges of the network and/or spread in several data centres. For example, each major POP would have a syslog server for the all the devices in the POP. Syslog data is either pre-processed on the remote server and/or pulled from the distributed servers to centralise server for processing.

Either of these two topologies (and other topologies) will work. What is important is that the syslog information is collected and used to maintain your network.

## Analysing Syslog Data

Configuring the routers to export syslog data is one step. The next step is to take the data, store it, analyse it, and use it in the day to day operations. Interface status, security alerts, and debugging problems are some of the most common events ISPs monitor from the collected syslog data. Some use custom written PERL scripts to create simple reports. Others use more sophisticated software to analyse the syslog data and create html reports, graphs, and charts.

The following is a list of known available software that analyses syslog data. Even if you are going to write your own scripts, it's worth checking out the commercial packages to see what can be done with syslog data.

Cisco Resource Manager	<a href="http://www.cisco.com/warp/public/cc/pd/wr2k/rsmn/index.shtml">http://www.cisco.com/warp/public/cc/pd/wr2k/rsmn/index.shtml</a>
Private I	<a href="http://www.opensystems.com/index.asp">http://www.opensystems.com/index.asp</a>
Crystal Reports	<a href="http://www.seagatesoftware.com/crystalreports/">http://www.seagatesoftware.com/crystalreports/</a>
Netforensics	<a href="http://www.netforensics.com/">http://www.netforensics.com/</a>

One item to remember with the ISP's syslog infrastructure – **time synchronisation is critical!** Two routers in two parts of the network need to have synchronised time with the syslog server. Hence, the ISP must take the effort to deploy NTP in their network – insuring the entire network and systems infrastructure is in time sync.

## Network Time Protocol (NTP)

Time synchronization across the ISP's network is one of those least talked about, yet critical pieces of the network. Without some mechanism to insure that all devices in the network are synchronized to exactly the same time source, functions like accounting, event logging, fault analysis, security incident response, and network management would not be

possible on more than one network device. Whenever an ISP's System or Network Engineer needs to compare two logs from two different systems, each system needs a frame of reference to match the logs. That frame of reference is synchronized time.

Network Time Protocol (NTP) is **THE** most overlooked feature on an ISP's network. NTP is a hierarchical protocol designed to synchronize the clocks on a network of computing and communication equipment. It is a dynamic, stable, redundant protocol used to keep time synchronised between network devices to a granularity of 1 millisecond. First defined in RFC 958, NTP has since been modified to add more redundancy and security. Other RFCs for time synchronization include:

- RFC-1128, "Measured Performance of the Network Time Protocol in the Internet System", 1989.
- RFC-1129, "Internet Time Synchronization: The Network Time Protocol", 1989.
- RFC-1165, "Network Time Protocol (NTP) over the OSI Remote Operations Service", 1990.
- RFC-1305, "Network Time Protocol (Version 3) Specification", 1992 (Draft Standard).

An NTP network usually gets its time from an authoritative time source, such as a radio clock, or a Global Positioning System (GPS) device, or an atomic clock attached to a timeserver. NTP then distributes this time across the network. NTP is a hierarchical with different timeservers maintaining authority levels. This highest authority is stratum 1. Levels of authority then descend from 2 to a maximum of 16. NTP is extremely efficient; no more than one packet per minute is necessary to synchronize two machines to within a millisecond of one another.

## NTP Architecture<sup>4</sup>

In the NTP model, a number of primary reference sources, synchronised by wire, GPS, or radio to national standards, are connected to widely accessible resources, such as backbone gateways, and operated as primary time servers. NTP provides a protocol to pass timekeeping information from these servers to other time servers via the Internet and to cross-check clocks and correct errors arising from equipment or propagation failures. Local-net hosts or gateways, acting as secondary time servers, use NTP to communicate with one or more of the primary servers. In order to reduce the protocol overhead, the secondary servers distribute time to the remaining local-net hosts. For reliability, selected hosts are equipped with less accurate (and less expensive) radio clocks. These hosts are used for backup in case of failure of the primary and/or secondary servers or the communication paths between them.

The NTP "network" consists of a multiple redundant hierarchy of servers and clients, with each level in the hierarchy identified by a stratum number. This number specifies the accuracy of each server, with the topmost level (primary servers) assigned as 1 and each level downward (secondary servers) in the hierarchy assigned as one greater than the preceding level. Stratum 1 is populated with hosts with bus or serial interfaces to reliable sources of time, such as radio clocks, GPS satellite timing receivers, or atomic clocks. Stratum 2 servers might be company or campus servers that obtain time from some number of primary servers over Internet paths, and provide time to many local clients. The stratum 2 servers may be configured to peer with each other, comparing clocks and generating a synchronised time value.

NTP performs well over the non-deterministic path lengths of packet-switched networks, because it makes robust estimates of three key variables in the relationship between a client and a time server. These three variables are: network delay, dispersion of time packet exchanges (a measure of maximum clock error between the two hosts), and clock offset (the correction to apply to a client's clock to synchronise it). Clock synchronisation at the 10-millisecond level over long distance (2000 km) WANs, and at the 1-millisecond level for LANs, is routinely achieved.

There is no provision for peer discovery or virtual-circuit management in NTP. Data integrity is provided by the IP and UDP checksums. No flow-control or retransmission facilities are provided or necessary. Duplicate detection is inherent in the processing algorithms.

---

<sup>4</sup> This section was written for Cisco's DNS/DHCP Manager. Sections of the documentation on NTP have been included in this document. The complete document can be found at: <http://www.cisco.com/univercd/cc/td/doc/product/iaabu/cddm/cddm111/adguide/ntp.htm>

Wednesday, June 06, 2001

NTP uses a system call on the local host to “slew” the local system clock by a small amount in order to keep the clock synchronised. If the local clock exceeds the “correct” time by pre-set threshold, then NTP uses a system call to make a step adjustment of the local clock.

NTP is careful to avoid synchronising to a system whose time may not be accurate. It avoids doing so in two ways. First of all, NTP will never synchronise to a system that is not itself synchronised. Secondly, NTP will compare the time reported by several systems, and will not synchronise with a system whose time is significantly different from the others, even if its stratum is lower.

## Client/Server Models and Association Modes

There are a number of modes in which NTP servers can associate with each other. The mode of each server in the pair indicates the behaviour the other server can expect from it. An “association” is formed when two peers exchange messages and one or both of them create and maintain an instantiation of the protocol machine. The association can operate in one of several modes: server, client, peer, and broadcast/multicast. The modes are further classified as active and passive. In active modes, the host continues to send NTP messages regardless of the reachability or stratum of its peer. In passive modes, the host sends NTP messages only as long as its peer is reachable and operating at a stratum level less than or equal to the host; otherwise, the peer association is dissolved.

- **Server Mode** – By operating in server mode, a host (usually a LAN time server) announces its willingness to synchronise, but not to be synchronised by a peer. This type of association is ordinarily created upon arrival of a client request message and exists only in order to reply to that request, after which the association is dissolved. Server mode is a passive mode.
- **Client Mode** – By operating in client mode, the host (usually a LAN workstation) announces its willingness to be synchronised by, but not to synchronise the peer. A host operating in client mode sends periodic messages regardless of the reachability or stratum of its peer. Client mode is an active mode.
- **Peer Mode** – By operating in peer mode (also called “symmetric” mode), a host announces its willingness to synchronise and be synchronised by other peers. Peers can be configured as active (symmetric-active) or passive (symmetric-passive).
- **Broadcast/Multicast Mode** – By operating in broadcast or multicast mode, the host (usually a LAN time server operating on a high-speed broadcast medium) announces its willingness to synchronise all of the peers, but not to be synchronised by any of them. Broadcast mode requires a broadcast server on the same subnet, while multicast mode requires support for IP multicast on the client machine, as well as connectivity via the MBONE to a multicast server. Broadcast and multicast modes are active modes.

Normally, one peer operates in an active mode (symmetric-active, client or broadcast/multicast modes), while the other operates in a passive mode (symmetric-passive or server modes), often without prior configuration. However, both peers can be configured to operate in the symmetric-active mode. An error condition results when both peers operate in the same mode, except for the case of symmetric-active mode. In this case, each peer ignores messages from the other, so that prior associations, if any, will be demobilised due to reachability failure.

## Implementing NTP on an ISP's Routers

The time kept on a machine is a critical resource, so we strongly recommend that you use the security features of NTP to avoid the accidental or malicious setting of incorrect time. Two mechanisms are available: an access list-based restriction scheme and an encrypted authentication mechanism. The example below highlights both NTP security options.

Cisco's implementation of NTP does not support stratum 1 service; in other words, it is not possible to connect a router running IOS directly to a radio or atomic clock. It is recommended that time service for your network is derived from the public NTP servers available in the Internet. If the network is isolated from the Internet, Cisco's implementation of NTP allows a system to be configured so that it acts as though it is synchronised via NTP, when in fact it has determined the

time using other means. Other systems then synchronise to that system via NTP. The command to set up a router in this way is:

```
ntp master 1
```

which tells the router that it is the master time source, and running at Stratum 1.

The example below is a NTP configuration on a router getting a Stratum 2 server connection from 192.36.143.150 and peering with 169.223.50.14. The peered IP addresses are the loopback addresses on each router. Each router is using the loopback as the source. This makes security easier (note the access-list).

```
clock timezone SST 8
!
access-list 5 permit 192.36.143.150
access-list 5 permit 169.223.50.14
access-list 5 deny any
!
ntp authentication-key 1234 md5 104D000A0618 7
ntp authenticate
ntp trusted-key 1234
ntp source Loopback0
ntp access-group peer 5
ntp update-calendar
ntp server 192.36.143.150
ntp peer 169.223.50.14
!
```

## NTP Deployment Examples

ISPs use several designs to deploy NTP on their backbones. It is hard to recommend a best current practice, but this section lists a few examples:

- **Flat peer structure.** Here all the routers peer with each other, with a few geographically separate routers configured to point to external systems. From experience, this is very stable, but convergence of time will be longer with each new member of the NTP mesh. The larger the mesh, the longer it takes for time to converge.
- **Hierarchy.** Here the BGP route reflector hierarchy is “copied” for the NTP hierarchy. Core routers (route reflectors) have a client/server relationship with external time sources, the reflector clients have a client/server relationship with the core routers, the customer routers have a client/server relationship with the reflector clients ... and so on down the tree. Hierarchy scales. A simple hierarchy that matches the routing topology provides consistency, stability, and scalability – hence the author’s favourite technique.
- **Star.** Here all the ISP routers have a client/server relationship with a few time “servers” in the ISP’s backbone. The dedicated timeservers are the centre of the *star*. The dedicated timeservers are usually Unix systems synchronised with external time sources, or their own GPS receiver. This set up is also reported to be very stable.

Undoubtedly there are other possibilities too. The main aim is to go for stability as time synchronisation is a key tool within the ISP backbone.

## NTP in a PoP (Example)

Devices in an ISP PoP do not need to part of the backbone NTP mesh. Instead, the devices in the PoP (routers, NAS, switches, and workstations) use the two core *PoP Gateway* routers as the NTP Servers for the PoP. All devices will use both routers a NTP sources – simplifying the NTP configuration and decreasing the NTP convergence time in the POP.

As can be seen in Figure 3, devices in a PoP all need time synchronisation. Accounting on the Radius server needs to be synchronised with the NAS equipment which needs to be synchronised with the Syslog server, which needs to be synchronised with the Access routers, which needs to be synchronised with the NetFlow Collectors, etc. Having all devices use the same two servers (one primary, one backup) assures time synchronisation between all devices.

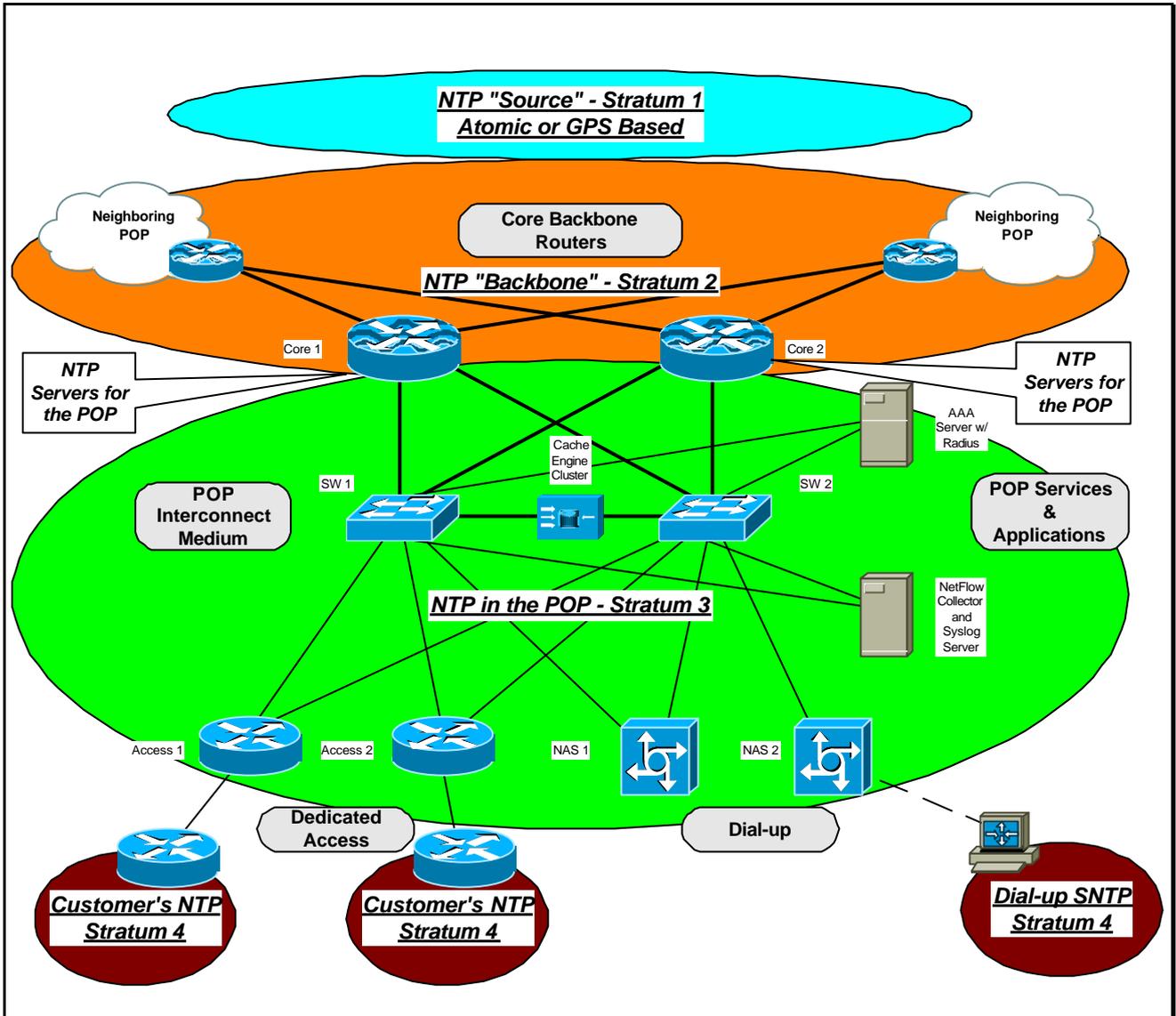


Figure 3 – Typical Internet POP Built for Redundancy and Reliability using the core routers as NTP servers

Configuration is simplified with only two servers. For example, NAS 1 – a Cisco 3640 with 96 built in modems would have a configuration highlighted in Example 2. POP Gateway router *Core 1* and *Core 2* have two configuration options. First, each device in the POP can be manually configured via an *ntp peer* command. Even though the peer commands open the gateway routers to have the ability to allow synchronisation, the *ntp server* commands on the POP devices will make this unlikely. Yet, there is always the chance of *MIT*<sup>5</sup> – mis-configuration on the POP Gateway or on one of the POP network devices. Hence, a second option offers more protection. This second option uses the *ntp access-group* to limit what can query, serve, and be an NTP peer. Example 1 demonstrates how the *ntp access-group* is used to add an extra layer of security for all the NTP Peers on the ISP's backbone while allowing a general access list to cover all the devices in the POP. If the ISP is following a logical addressing plan, then the whole POP will be assigned one block of IP addresses for all the infrastructure and loopback addresses. This makes the *ntp access-group serve-only* ACL easier to create – with one ACL covering the entire POP.

<sup>5</sup> Maintenance Induced Trouble – *MIT*

```

! POP Gateway Router
!
ntp authentication-key 4235 md5 ISPWorkshop
ntp authenticate
ntp trusted-key 4235
!
! Lock NTP source to the uplink
!
ntp source Loopback0
ntp update-calendar
!
! List of NTP Peers - Adding an additional Security Layer
!
ntp access-group peer 99
!
! Allow POP Devices to use this router as an NTP Server
!
ntp access-group serve-only 42
!
! Loopback Addresses of the Backbone Routers
ntp peer 200.200.1.1
ntp peer 200.200.1.2
ntp peer 200.200.1.3
ntp peer 200.200.1.4
! etc ...

```

#### Example 1 – NTP Config for the POP Gateway Routers

```

! POP Devices
ntp authentication-key 4235 md5 ISPWorkshop
ntp authenticate
!
ntp trusted-key 4235
!
! Lock NTP source to the uplink
ntp source Loopback0
ntp update-calendar
!
! IP Addresses to Routers Core 1 and Core 2
ntp server 192.135.248.249
ntp server 192.135.248.250
!

```

#### Example 2 – NTP Devices in the POP

## Further NTP References

Here are some URLs with further pointers to NTP information, software, and hardware:

Network Time Protocol (NTP) Master Clock for the US	<a href="http://tycho.usno.navy.mil/">http://tycho.usno.navy.mil/</a>
Datum Inc, Bancomm Timing Division	<a href="http://www.datum.com/">http://www.datum.com/</a>
True Time Inc.	<a href="http://www.truetime.com">http://www.truetime.com</a>
The Time Web Server (Time Sync) by Dave Mills	<a href="http://www.eecis.udel.edu/~ntp/">http://www.eecis.udel.edu/~ntp/</a>
Coetanian Systems Time Synchronisation Server 100	<a href="http://www.coetanian.com/tss/tss100.htm">http://www.coetanian.com/tss/tss100.htm</a>

## *Simple Network Management Protocol (SNMP)*

Keeping data on the health of an ISP's network is critical to its survival as a business. For example, an ISP must know the load on their backbone circuits, and the loading on their customer circuits. They also need to keep track of the packets lost on routers at various points of the network. And they need to be aware of long term trends on the overall growth of the

Wednesday, June 06, 2001

network. Simple Network Management Protocol (SNMP) can collect and process all of this data – hence it is a very critical utility for Network Engineers. Given the wide range of freeware, shareware, and commercially available SNMP tools, all ISPs should be able to collect SNMP data, process it, graph it, and analyse it for proper traffic engineering. Appendix 5 – Traffic Engineering Tools – lists pointers to various software and tools on the Net.

Yet, remember that SNMP, especially version 1, has very weak security! If SNMP is not going to be used, turn it off! On the other hand, ensure that there is sufficient configuration information present which will control the use of SNMP such that it doesn't become a security risk. Most importantly, never leave a configuration which includes “public” or “private” as the community string – these strings are so well known, and are common defaults on hardware shipping from so many vendors that they are open invitations to abuse, filters or not.

## SNMP in read-only mode

If SNMP is used in a read-only scenario, ensure that it is set up with appropriate access controls. The following is an example:

```
access-list 98 permit 215.17.34.1
access-list 98 permit 215.17.1.1
access-list 98 deny any
!
snmp-server community 5nmc02m RO 98
snmp-server trap-source Loopback0
snmp-server trap-authentication
snmp-server enable traps config
snmp-server enable traps envmon
snmp-server enable traps bgp
snmp-server enable traps frame-relay
snmp-server contact Barry Raveendran Greene [bgreene@cisco.com]
snmp-server location Core Router #1 in City Y
snmp-server host 215.17.34.1 5nmc02m
snmp-server host 215.17.1.1 5nmc02m
snmp-server tftp-server-list 98
!
```

Note the application of *access-list 98* above. The community string *5nmc02m* is not encrypted, hence the need to use an access-list to control access. This is too often forgotten, and if the community string is known outside the ISP, it can easily lead to compromise of a router. In fact, there are scripts available on the Internet which allow *script kiddies*<sup>6</sup> to probe a router and crack the community name. Unless BOTH SNMP is set up to send a trap on community name authentication failure AND the SNMP management device is configured to react to the authentication failure trap, the *script kiddies* will most likely discover the community name.

ISPs should always remember that accepting SNMP only from “known good” IP addresses does not guarantee their security. Unless the ISP has some very serious anti-spoofing measures in place, you **cannot** completely rely on IP addresses for the primary security of any system. IP addresses are frequently spoofed. Layered security where the system relies on several mechanisms has proven to be more effective.

The *snmp-server host* configuration lists the hosts to which SNMP information is sent – if there is no means of collecting SNMP traps, don't configure *snmp-server host*, saving CPU cycles and network bandwidth. ISPs should ensure that the *snmp-server host* is configured to receive and respond to SNMP traps. For instance, a PERL script on a PC running Unix or Linux with UCD SNMP<sup>7</sup> could receive an SNMP environmental trap relating to high router internal temperature, e-mail it to the NOC alias, send an alphanumeric page to the on-duty engineer, and open a trouble ticket.

---

<sup>6</sup> Script Kiddies are amateur crackers who use scripts to break into and cause damage to networks and systems on the Internet.

<sup>7</sup> CMU SNMP has not been updated in a while and the project has now been taken over by UCD. UCD SNMP contains a port and modified code of the CMU 2.1.2.1 snmp agent. It has been modified to allow extensibility quickly and easily. It is by far the best and most configurable system – and it's free!

<http://ucd-snmp.ucdavis.edu/> for the source

<http://net-snmp.sourceforge.net/> for the UCD-SNMP project home page

SNMP can be set up with more than one community, with a different access-list. This is quite useful, say if different groups in the company require different SNMP access to the router, or maybe a transit ISP offers SNMP read access to its ISP customers on its aggregation router. An example might look like this:

```
access-list 97 permit 220.3.20.1
access-list 97 deny any
!
access-list 98 permit 215.17.34.1
access-list 98 permit 215.17.1.1
access-list 98 deny any
!
snmp-server community 5nmc02m RO 98
snmp-server community lspxs RO 97
!
```

## SNMP in read-write mode

If SNMP is going to be used in read/write mode, think very very carefully about the configuration and why there is a requirement to do this, as configuration errors in this scenario could leave the router very vulnerable.

If possible, put an ACL at the edge of your network that will prevent outside parties from probing your network via SNMP. There are many publicly and commercially available tools which will scan ANY network on the Internet via SNMP. This could map out your entire network and/or discover a device that has had SNMP left open.

**In short, our recommendation is not to use SNMP in read-write mode on any ISP router – the operational risk is simply not worth it.**

## SNMP and commercial network management software

One thing to be aware of is that some commercial network management software likes to “take over” the network by doing auto-discovery of devices on the backbone. Many ISP engineers don’t approve of this style of network management, so tend to build their own tools which are suitable for monitoring the backbone. For those ISPs who rely on commercial packages, such as HP OpenView, it is worth remembering about and understanding the impact the auto-discovery function these systems have. Auto-discovery fits very nicely into the campus network where the network operators have little control over the backbone and what is added to it. However, the network operations staff should be in control of an ISP backbone, and all additions and removals from the backbone should be premeditated.

One example follows. A service provider was using a commercial package to monitor their backbone, and was being severely troubled by very high CPU utilisation. After a lot of research, it was discovered that the auto-discovery function of their management system was downloading the entire routing table from each of their routers. This service provider was, of course, carrying all 80k prefixes which were present at the time. Given the CPU hit dealing with a full BGP routing table, it is little wonder that an SNMP poll of the routing table was causing high CPU. The workaround was the following SNMPv2 configuration:

```
snmp-server view cutdown internet included
snmp-server view cutdown ipRouteTable excluded
snmp-server view cutdown ipNetToMediaTable excluded
snmp-server view cutdown at excluded
snmp-server community public view cutdown RO
```

For routers which run IOS versions which don’t take the above object names, the following will also achieve the same result:

```
snmp-server view cutdown internet included
snmp-server view cutdown ip.21 excluded
snmp-server view cutdown ip.22 excluded
snmp-server view cutdown mib-2.3 excluded
snmp-server community public view cutdown RO
```

Wednesday, June 06, 2001

This will basically stop the router from responding to SNMP queries regarding the routing and ARP tables.

## *HTTP Server*

The *http* server is a new feature in 11.1CC and 12.0 software which, when configured and enabled, allows the network operator to view and configure the router through a convenient and easy to use Web interface (via common browsers such as Netscape or Internet Explorer). If there is no intention of using the built-in HTTP server in the running configuration, it is worth checking that it has not been enabled by default, or in error, or during system installation. The configuration command:

```
no ip http server
```

will ensure that the server is not running.

If there is a need to configure the http server because Web based configuration of the router is desired or desirable, then we strongly advise that the server is configured with the appropriate security. For example:

```
ip http server
ip http port 8765          ! use a non-standard port
ip http authentication aaa ! use the AAA authentication method which has been configured
ip http access-class <1-99>! access-list to protect the HTTP port
```

Notice the suggestion of a non-standard port. This adds a little obscurity to the Web server on the router, making potential attack a little more difficult. Also notice the access-list used, and the authentication type (AAA is discussed in the section on router security). This ensures that only the permitted administrative users of the router get access to the device from the authorised IP address range.

**In short, our recommendation is not to use the HTTP server on any ISP router or switch – the operational risk is simply not worth it.**

## *Core Dumps*

A *core dump* facility has been part of IOS for several years and many software releases. The core dump facility operates like the Unix variant – when a programme crashes, the memory image is stored in a “core” file. When a router crashes, a copy of the core memory is kept. Before the memory is erased on reboot, the router can be set up to copy the core dump out to a UNIX server. An account (ftp, tftp, or rcp) and sufficient disk space (equal to the amount of memory on the router per dump) needs to be set up and allocated.

Here is an example using ftp:

```
ip ftp source-interface Loopback0
ip ftp username cisco
ip ftp password 7 045802150C2E
!
exception protocol ftp
exception dump 169.223.32.1
!
```

Note the use of the loopback interface as a source interface. It is recommended that access to the “cisco” account above be made as secure as possible. For example, do not send core dumps to the same ftp server as the one used to provide generic anonymous or user ftp accounts. Use a wrapper for the ftp daemon, and make sure that only the loopback interfaces are listed in any system filter lists.

Be aware that **rcp** is inherently insecure and its use cannot be recommended over a public network. Also **tftp** core dumps (which are the default in IOS) only support system memory sizes up to 16Mbytes. Generally it is recommended that **ftp** core dumps be configured whatever the situation or router hardware configuration.

More detailed information for configuring core dumps on a Cisco IOS based system is located on the Cisco Documentation CD. It is publicly available via the Web at:

*Creating Core Dumps* – [http://www.cisco.com/univerred/cc/td/doc/cisintwk/itg\\_v1/tr19aa.htm](http://www.cisco.com/univerred/cc/td/doc/cisintwk/itg_v1/tr19aa.htm)

It includes information needed to troubleshoot problems using the *core dump* and *show stacks* commands.

---

---

## GENERAL FEATURES

---

---

This section covers general features which ISPs should consider for their routers and network implementations. Most are good design practices rather than leveraging particular unique IOS features, but each demonstrates how IOS can aid the smooth operation of an ISP's business.

### *Command Line Interface*

The IOS Command Line Interface (CLI) is the traditional (and favoured) way of interacting with the router, to enter and change configuration, and to monitor the router's operation. The CLI is now very well documented in the Cisco UniverCD documentation set, e.g. at [http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/fun\\_r/index.htm](http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/fun_r/index.htm). There are, however, a few tips and tricks which are regularly used by ISPs and are worth mentioning here.

### Editing Keys

There are several keys which are very useful as shortcuts for editing the IOS configuration. While these are documented in detail in the IOS 12.0 documentation set, it is useful to point out the most commonly used here.

- The TAB key while typing in an IOS configuration command will complete the command being typed in. This saves typing effort and is especially useful when the operator is still learning the IOS command set.
- The “?” key while typing in an IOS configuration command will list the available commands starting with the characters entered so far.
- Up arrow and down arrow respectively allows the operator to scroll up and down through the history buffer.
- CTRL-A moves the cursor to the beginning of the line
- CTRL-E moves the cursor to the end of the command line
- CTRL-K deletes all characters from the cursor to the end of the command line
- CTRL-W deletes the word to the left of the cursor
- CTRL-X deletes all characters from the cursor to the beginning of the command line
- ESC-B moves the cursor back one word
- ESC-F moves the cursor forward one word

The complete list of commands can be found in the IOS documentation:

[http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/fun\\_r/frprt1/frui.htm](http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/fun_r/frprt1/frui.htm)

### CLI String Search

A new feature in IOS from 11.1CC and 12.0 is a Unix grep-like function allowing operators to search for common expressions in configuration and other terminal output. See the following IOS document for complete information:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t1/cliparse.htm>

The function is invoked by using a vertical bar “|”, like the Unix “pipe” command. For example:

```
beta7200#show configuration | ?
  begin      Begin with the line that matches
  exclude    Exclude lines that match
  include    Include lines that match
beta7200#show configuration _
```

Following one of these three options the operator should enter a regular expression to get a pattern match on the configuration, as in this example above. The regular expressions can be single or multiple characters or a more complex construction, in a similar style to Unix regular expressions.

```
Defiant#show running-config | begin router bgp
router bgp 200
no synchronization
neighbor 4.1.2.1 remote-as 300
neighbor 4.1.2.1 description Link to Excalabur
neighbor 4.1.2.1 send-community
neighbor 4.1.2.1 version 4
neighbor 4.1.2.1 soft-reconfiguration inbound
neighbor 4.1.2.1 route-map Community1 out
maximum-paths 2
--More--
```

During the displaying of configuration or file contents, the screen pager “—More—“ will be displayed if the output is longer than the current terminal length setting. It is possible to do a regular expression search at this prompt too. The “/” key matches the “begin” keyword above, the “-” key means exclude, and the “+” key means include.

Finally, in enable mode it is possible to use the “more” command to display file contents. Regular expressions as in the example shown above can be used with “more”. The options available with more are given in this example:

```
beta7200#more ?
/ascii      Display binary files in ascii
/binary     Force display to hex/text format
/ebcdic     Display binary files in ebcdic
bootflash:  File to display
disk0:      File to display
disk1:      File to display
flash:      File to display
ftp:        File to display
null:       File to display
nvram:      File to display
rcp:        File to display
slot0:      File to display
slot1:      File to display
system:     File to display
tftp:       File to display
beta7200#
```

By using the “|” after the more command and its option above, it is possible to search within the file for the strings of interest in the same way as discussed previously.

## *IOS and Loopback interfaces*

There are many instances throughout this document where the use of the loopback interface is mentioned. While this is not a feature unique to IOS, there are many and considerable advantages in making full use of the capability the loopback interface allows. This section brings together all the occasions the loopback interface is mentioned in this document, and describes how they can be useful to the ISP Network Engineer.

## **Background**

ISPs endeavour to minimise the unnecessary overhead present in their networks. This unnecessary overhead can be number of networks carried in the IGP, the number of skilled engineering staff to operate the network, or even network security. The utilisation of one feature, the loopback interface on the router, goes a long way to help with each of the three scenarios mentioned here.

Wednesday, June 06, 2001

The size of the IGP is attended to by summarisation of point-to-point addresses at PoP or regional boundaries, the use of IP unnumbered on static WAN interfaces, and a carefully designed network addressing plan. ISP Network security is of paramount importance, and any techniques which make the management simpler are usually welcomed. For example, when routers access core servers for whatever reasons, ISPs apply filters or access-lists to these servers so that the risk of compromise from the outside is reduced. The loopback interface is helpful here too.

It is very common to assign all the IP addresses used for loopback interfaces from one address block. For example, an ISP with around 200 routers in their network may assign a /24 network (253 usable addresses) for addressing the loopback interface on each router. If this is done, all dependent systems can be configured to permit this address range access the particular function concerned, be it security, unnumbered WAN links, or the iBGP mesh. Some examples follow in the rest of this section.

## BGP Update-Source

Here the iBGP mesh is built using the loopback interface on each router. The loopback doesn't disappear, ever, so results in a more stable iBGP, even if the underlying physical connectivity is less than reliable. Sample configuration:

```
hostname gateway1
!
interface loopback 0
 ip address 215.17.1.34 255.255.255.255
!
router bgp 200
 neighbor 215.17.1.35 remote-as 200
 neighbor 215.17.1.35 update-source loopback 0
 neighbor 215.17.1.36 remote-as 200
 neighbor 215.17.1.36 update-source loopback 0
!
```

## Router ID

If a loopback interface is configured on the router, its IP address is used as the router ID. This is important to ensure stability and predictability in the operation of the ISP's network.

OSPF chooses the Designated Router on a LAN as the device which has the highest IP address. If routers are added or removed from the LAN, or a router gains an interface with a higher address than that of the existing DR, it is likely that the DR will change during the next network event – this may or may not be undesirable. It can be avoided by ensuring that the loopback interface is configured and in use on all routers on the LAN.

The loopback interface is used for the BGP router ID. If the loopback isn't configured, then BGP uses the lowest IP address on the router. Again due to the ever-changing nature of an ISP network, this value may change, possibly resulting in operational confusion. Configuring and using a loopback interface ensures stability.

**Important Note:** If the router has two or more loopback interfaces configured, the router ID is the highest IP address of the configured loopback interfaces at the time of booting the router.

## IP Unnumbered Interfaces

IP addresses do not need to be used on static WAN links to customers. IP unnumbered saves /30 of address space, and one entry in the IGP routing table, a significant saving for a large number of customers. IP unnumbered makes use of the loopback interface on the ISP's backbone router, the same loopback interface used for iBGP etc. An example configuration:

ISP's router:

```
interface loopback 0
 description Loopback interface on Gateway Router 2
```

```

    ip address 215.17.3.1 255.255.255.255
    !
interface Serial 5/0
  description 128K HDLC link to Galaxy Publications Ltd [galpub1] WT50314E R5-0
  bandwidth 128
  ip unnumbered loopback 0
    !
ip route 215.34.10.0 255.255.252.0 Serial 5/0
    !

```

Customer's router:

```

interface Ethernet 0
  description Galaxy Publications LAN
  ip address 215.34.10.1 255.255.252.0
    !
interface Serial 0
  description 128K HDLC link to Galaxy Internet Inc WT50314E C0
  bandwidth 128
  ip unnumbered ethernet 0
    !
ip route 0.0.0.0 0.0.0.0 Serial 0
    !

```

## Exception Dumps by FTP

Cisco routers can be configured to dump core memory to an FTP server as part of the diagnostic and debugging process. However, this core dump should be to a system not running a public ftp server, but one heavily protected by filters (tcp wrapper even) which only allows the routers access. If the loopback interface address is used as source address from the router, and is part of one address block, the filter is very easy to configure. A 200 router network with 200 disparate IP addresses makes for a very large filter list on the ftp server. Sample IOS configuration:

```

ip ftp source-interface Loopback0
ip ftp username cisco
ip ftp password 7 045802150C2E
exception protocol ftp
exception dump 169.223.32.1

```

## TFTP-SERVER Access

TFTP is the most common tool to upload and download configurations. The TFTP Server's security is critical. That means using security tools with a IP source addresses. IOS allows tftp to be configured to use a specific IP interfaces address. This allows a fixed ACL on the TFTP Server based on a fixed address on the router (i.e. the loopback interface).

```

ip tftp source-interface Loopback0

```

## SNMP-SERVER Access

If SNMP is used in the network, the loopback interface can again be brought into use for security access issues. If SNMP traffic from the router is sourced from its loopback interface, it is easy to protect the SNMP management station in the NOC. Sample IOS configuration:

```

access-list 98 permit 215.17.34.1
access-list 98 permit 215.17.1.1
access-list 98 deny any
    !
snmp-server community 5nmc02m RO 98
snmp-server trap-source Loopback0
snmp-server trap-authentication
snmp-server host 215.17.34.1 5nmc02m
snmp-server host 215.17.1.1 5nmc02m

```

Wednesday, June 06, 2001

## TACACS/RADIUS-Server Source Interface

Most ISP uses TACACS+ or Radius for user authentication. Very few define accounts on the router itself as this offers more opportunity for the system to be compromised. A well-protected TACACS+ server accessed only from the router's loopback interface address block offers more security of user and enable accounts. Sample configuration for standard and enable passwords:

```
aaa new-model
aaa authentication login default tacacs+ enable
aaa authentication enable default tacacs+ enable
aaa accounting exec start-stop tacacs+
!
ip tacacs source-interface Loopback0
tacacs-server host 215.17.1.2
tacacs-server host 215.17.34.10
tacacs-server key CKr3t#
!
```

When using RADIUS, either for user administrative access to the router, or for dial user authentication and accounting, the router configuration to support loopback interfaces as the source address for RADIUS packets originating from the router looks like this:

```
radius-server host 215.17.1.2 auth-port 1645 acct-port 1646
radius-server host 215.17.34.10 auth-port 1645 acct-port 1646
ip radius source-interface Loopback0
!
```

## NetFlow Flow-Export

Exporting traffic flows from the router to a NetFlow Collector for traffic analysis or billing purposes is quite a common activity. Using the loopback interface as the source address for all exported traffic flows from the router allows for more precise and less costly filtering at or near the server. A configuration example:

```
ip flow-export destination 215.17.13.1 9996
ip flow-export source Loopback0
ip flow-export version 5 origin-as
!
interface Fddi0/0/0
description FDDI link to IXP
ip address 215.18.1.10 255.255.255.0
ip route-cache flow
ip route-cache distributed
no keepalive
!
```

Here interface FDDI0/0/0 has been configured to capture flow records. The router has been configured to export version 5 style flow records to the host at IP address 215.17.13.1 on UDP port 9996, with source address being the router's loopback interface.

## NTP Source Interface

NTP is the means of keeping the clocks on all the routers on the network synchronised to within a few milliseconds. If the loopback interface is used as the source interface between NTP speakers, it makes filtering and authentication somewhat easier to maintain. Most ISPs only wish to permit their customers to synchronise with their time servers and not everyone else in the world. A configuration example:

```
clock timezone SST 8
!
access-list 5 permit 192.36.143.150
access-list 5 permit 169.223.50.14
!
```

```

ntp authentication-key 1234 md5 104D000A0618 7
ntp authenticate
ntp trusted-key 1234
ntp source Loopback0
ntp access-group peer 5
ntp update-calendar
ntp peer 192.36.143.150
ntp peer 169.223.50.14
!
```

## SYSLOG Source Interface

Syslog servers also require careful protection on ISP backbones. Most ISPs prefer to see only their own systems' syslog messages, not anything from the outside world. Denial of service attacks on syslog devices are not unknown either. Protecting the syslog server is again made easier if the known source of sysbg messages comes from a well-defined set of address space, for example that used by the loopback interfaces on the routers. A configuration example:

```

logging buffered 16384
logging trap debugging
logging source-interface Loopback0
logging facility local7
logging 169.223.32.1
!
```

## Telnet to the Router

This may seem to be an odd example in a document dedicated to IOS Essentials. However, remember that a loopback interface on a router never changes its state, and has rarely any need to change IP address. Physical interfaces may be physically swapped out or renumbered, address ranges may change, but the loopback interface will always be there. So if the DNS is set up so that the router name maps to the loopback interface address, there is one less change to worry about during operational and configuration changes elsewhere in the ISP backbone. ISP backbones are continuously developing entities. Example from the DNS forward and reverse zone files:

```

; net.galaxy zone file
net.galaxy.      IN      SOA      ns.net.galaxy. hostmaster.net.galaxy. (
1998072901 ; version == date(YYYYMMDD)+serial
10800      ; Refresh (3 hours)
900        ; Retry (15 minutes)
172800     ; Expire (48 hours)
43200 )    ; Mimimum (12 hours)
                IN      NS       ns0.net.galaxy.
                IN      NS       ns1.net.galaxy.
                IN      MX       10 mail0.net.galaxy.
                IN      MX       20 mail1.net.galaxy.
;
localhost      IN      A        127.0.0.1
gateway1       IN      A        215.17.1.1
gateway2       IN      A        215.17.1.2
gateway3       IN      A        215.17.1.3
;
;etc etc

; 1.17.215.in-addr.arpa zone file
;
1.17.215.in-addr.arpa.  IN      SOA      ns.net.galaxy. hostmaster.net.galaxy. (
1998072901 ; version == date(YYYYMMDD)+serial
10800      ; Refresh (3 hours)
900        ; Retry (15 minutes)
172800     ; Expire (48 hours)
43200 )    ; Mimimum (12 hours)
                IN      NS       ns0.net.galaxy.
                IN      NS       ns1.net.galaxy.
1           IN      PTR     gateway1.net.galaxy.
2           IN      PTR     gateway2.net.galaxy.
```

Wednesday, June 06, 2001

```
3          IN      PTR      gateway3.net.galaxy.  
;  
;etc etc
```

On the router, set the telnet source to the loopback interface:

```
ip telnet source-interface Loopback0
```

## RCMD to the router

RCMD requires the operator to have the Unix rlogin/rsh clients to allow access to the router. Some ISPs use RCMD for grabbing interface statistics, uploading or downloading router configurations, or the taking a snapshot of the routing table. The router can be configured so that RCMD connections use the loopback interface as the source address of all packets leaving the router:

```
ip rcmd source-interface Loopback0
```

## Interface Configuration

Configuring interfaces is more than simply plugging in the cable and activating the interface with the IOS command “no shutdown”. Attention should be applied to details such as whether it is WAN or LAN, a routing protocol is running across the interface, addressing and masks to be used, and operator information.

### Description

Use the “*description*” interface command to document details such as the circuit bandwidth, the customer name, the database entry mnemonic, the circuit number the circuit supplier gave you, and the cable number. Sounds like overkill, especially if there is a customer database within the ISP organisation. However, it is very easy to pick up all the relevant details from the router “*show interface*” command if and when an engineer needs to be on site, or is away from the database system, or the database happens to be unavailable. There can never be too little documentation, and documentation such as this ensures that reconstructing configurations and diagnosing problems are made considerably easier.

### Bandwidth

Don’t forget the “*bandwidth*” interface command. It is used by interior routing protocols to decide optimum routing and especially important to be set properly in the case of backbone links using only a portion of the available bandwidth support by the interface. For example a serial interface (Serial0/0) on a router supports speeds up to 4Mbps, but has a default bandwidth setting of 1.5Mbps. If the backbone has different size links from 64Kbps to 4Mbps and the bandwidth command is not used, the interior routing protocol will assume that all the links have the same cost, and calculate optimum paths accordingly – this may be less than ideal.

On customer links it may seem that this setting is superfluous, as an interior routing protocol is never run over a link to a customer. However, it never the less provides very useful online documentation for what the circuit bandwidth is. Furthermore, the bandwidth on the circuit is used to calculate the interface load variable – some ISPs monitor their customer interfaces loading by SNMP polls so that they can get prior warning of problems, or congestion, or proactively inform customers of necessary upgrades. (Some ISPs look at the load variable; other ISPs look at the 5-minute average, inbound and outbound. If you monitor the load variable, you need to set the bandwidth so that it matches the true circuit bandwidth, not the default configured on the router.)

## IP Unnumbered

Traditionally ISPs have used IP addresses for the point-to-point links on leased line circuits to customers. Indeed, several years ago, prior to the advent of CIDR, it was not uncommon to see a /26 or even a /24 used for simple point to point link addresses. With the advent of CIDR, /30 networks have been used instead (/30 is a block of four addresses, two of which can be used for physical interfaces). However, this has started to lead to problems too as IGPs of some of the larger ISPs are starting to carry several thousand networks, affecting convergence time, and becoming an administrative and documentation nightmare.

To avoid problems with large numbers of /30s floating around the ISP's internal routing protocol, and avoid the problems of keeping internal documentation consistent with network deployment (especially true in larger ISPs), many are now using "unnumbered" point to point links.

An unnumbered point-to-point link is one which requires no IP addresses. The configuration is such that traffic destined for one network from another is simply pointed at the serial interface concerned. "*ip unnumbered*" is an essential feature applicable to point-to-point interfaces such as Serial, HSSI, POS, etc. It allows the use of a fixed link (usually from ISP to customer) without consuming the usual /30 of address space, thereby keeping the number of networks routed by the IGP low. The "*ip unnumbered*" directive specifies that the point-to-point link should use an address of another interface on the router, typically a LAN, or more usually a Loopback interface. Any networks, which require to be routed to the customer, are pointed at the serial interface rather than the remote address of the point-to-point link, as would be done in normal instances.

### Caveats

There are some situations which ISPs need to consider before implementing an IP unnumbered system for their customer point-to-point connections. These are considerations only – bear in mind that many ISPs have used IP unnumbered for several years, mainly so that they can control the size of the IGP running in their backbone network.

- **Pinging the Customer.** Many ISPs use monitoring systems which use "ping" to check the status of the leased line (customer connectivity). Even if the customer unplugs the LAN, an alarm will not be raised on the ISP's management system. This is because the customer router still knows that the LAN IP address is configured on the system and "useable". So long as the IP address is configured on the LAN there will be no reachability issues with using "ip unnumbered".
- **Routing Protocols.** If a routing protocol needs to be run over this link, it is operationally much easier to use IP addresses. Don't use "*ip unnumbered*" if the customer is peering with you using BGP across the link, or if the link is an internal backbone link. Simply use a network with a /30 address mask. (Routing will work over unnumbered links but the extra management and operational complexity probably outweighs the small address space advantage gained.)
- **Loopback Interfaces on the Customer's Router.** These offer no advantage to addressing the "ping" problem, and unnecessarily consume address space (not to mention adding complexity to the customer router configuration).

## A Full Example

Using the above configuration commands, a typical configuration on the ISP's router would be as follows:

```
interface loopback 0
  description Loopback interface on Gateway Router 2
  ip address 215.17.3.1 255.255.255.255
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
!
interface Serial 5/0
  description 128K HDLC link to Galaxy Publications Ltd [galpub1] WT50314E R5-0
  bandwidth 128
  ip unnumbered loopback 0
```

Wednesday, June 06, 2001

```
no ip redirects
no ip directed-broadcast
no ip proxy-arp
!
ip route 215.34.10.0 255.255.252.0 Serial 5/0
```

The customer router configuration would look something like:

```
interface Ethernet 0
description Galaxy Publications LAN
ip address 215.34.10.1 255.255.252.0
no ip redirects
no ip directed-broadcast
no ip proxy-arp
!
interface Serial 0
description 128K HDLC link to Galaxy Internet Inc WT50314E C0
bandwidth 128
ip unnumbered ethernet 0
no ip redirects
no ip directed-broadcast
no ip proxy-arp
!
ip route 0.0.0.0 0.0.0.0 Serial 0
```

In this example the Regional or Local Registry has allocated the customer the network block 215.34.10.0/22. This is routed to the customer site with the static route pointing to Serial 5/0 above. The customer router simply needs a default route pointing to its serial interface to ensure a connection.

With this configuration, there are no /30s from point to point links present in the IGP, and the ISP does not need to document the link address, or keep a table/database up to date. It all makes for easier configuration, and easier operation of the ISP's business.

Note the contents of the description field. This example has included:

- the bandwidth of the circuit, **128K**
- the encapsulation, **HDLC**
- the name of the company, **Galaxy Publications Ltd**
- the database mnemonic in the ISP's internal database, **[galpub1]**
- the telco's circuit ID, **WT50314E**
- the cable number **R5-0**

All of these are online documentation, seemingly superfluous, but so necessary/essential to ensure smooth and efficient operations. All the information pertinent to the customer's connection from the cabling to the IP values is contained in the interface configuration. Should the ISP's database be down, or unavailable, any debug information required by operators or engineers can be found on the router itself.

## *Interface Status Checking*

There are some useful hidden IOS commands which allows the operator to check the status of the interfaces in IOS. Two useful commands are "*show interface switching*" and "*show interface stats*".

### **“Show Interface Switching”**

The IOS command **show interface switching** provides useful information about the switching status of the router's interfaces, either on an individual interface basis, or over the whole router. The full command format is **show interface [int n/n] switching**, where an optional argument is the specific interface in question. Command completion cannot be used for "switching" – it needs to be typed in up to and including the final "i". Sample output might be like:

```

gw>show interface FastEthernet 1/0 switching
FastEthernet1/0 Production LAN
  Throttle count          0
  Drops                   RP          0          SP          0
  SPD Flushes             Fast          0          SSE          0
  SPD Aggress             Fast          0
  SPD Priority             Inputs        2421       Drops          0

  Protocol                Path        Pkts In    Chars In    Pkts Out    Chars Out
  Other                   Process     0          0           74633       4477980
  Cache misses
  Fast                    0          0           31653       2957994
  Auton/SSE               0          0           0            0
  IP                       Process     5339594    516613071   5622371     851165330
  Cache misses
  Fast                    256289350  1125491757  257803747   2058541849
  Auton/SSE               0          0           0            0
  ARP                     Process     16919     1015300     34270       2056200
  Cache misses
  Fast                    0          0           0            0
  Auton/SSE               0          0           0            0
  CDP                     Process     12449     4083272     12440       4142520
  Cache misses
  Fast                    0          0           0            0
  Auton/SSE               0          0           0            0

```

```
gw>
```

which shows SPD<sup>8</sup> activity, as well as other activity on that particular interface on the router. Note the references to autonomous/SSE switching – this is applicable only to the Cisco 7000 series with Silicon Switch Engine only (product now discontinued, but a significant part of the Internet core in the mid 1990s). Fast switching refers to all packets which have not been process switched, so would include Optimum switching, NetFlow and CEF.

## “Show Interface Stats”

The IOS command **show interface stats** is the second useful command to show interface status. It shows the number of packets and characters inbound and outbound on an individual or all the router interfaces. The full command format is **show interface [int n/n] stats**, where an optional argument is the specific interface in question. Command completion cannot be used for “stats” – at least “st” needs to be typed in at the command prompt. Sample output might be like:

```

gw>show interface stats
Interface FastEthernet0/0 is disabled

FastEthernet1/0
  Switching path    Pkts In    Chars In    Pkts Out    Chars Out
  Processor         5371378    521946816   5746126     862068168
  Route cache       256413200  1149405512  257960291   2072462774
  Total             261784578  1671352328  263706417   2934530942

```

```
gw>
```

As for “interface switching”, the output differentiates between packets which go via the processor and those which have been processed via the route cache. This is useful to determine the level of process switching which is taking place on the router.

On a router which supports distributed switching (e.g. 7500 with VIP interfaces), the output will look like:

```

gw>show interface stats
FastEthernet0/1/0
  Switching path    Pkts In    Chars In    Pkts Out    Chars Out
  Processor         207745     14075132    270885     21915788
  Route cache       0          0           0           0

```

<sup>8</sup> SPD is discussed in detail in its own section on Page 98

Wednesday, June 06, 2001

Distributed cache	93	9729	0	0
Total	207838	14084861	270885	21915788

Notice that packets which have been processed via the distributed cache are counted separately from those handled via the central route cache and the processor.

## Cisco Express Forwarding (CEF)

CEF is now the recommended forwarding/switching path for Cisco routers in an ISP environment. CEF adds increased performance, scalability, resilience, and enabling new functionality over the older optimum switching. Details on the operation and functionality of CEF are covered in detail by the IOS Documentation and in several whitepapers describing CEF (see references in the Technical Reference and Recommend Reading section at the end of this document).

Implementation is simple with either of the following commands (depending on the platform):

```
ip cef
ip cef-distributed
```

The key issue for ISPs is ensuring CEF is turned on. On most platforms, CEF is not turned on by default. So ISP Engineers need to be mindful to ensure CEF is turned on. Table 1 provides a list of the default CEF configurations for various Cisco platforms. Service Providers should check their configuration scripts to insure CEF/dCEF is turned on – especially the 7200 based edge platforms – uBR, 6400, and 5800 NAS.

On this platform....	The default is....
2600/3600	CEF is not enabled
4500/4700	CEF is not enabled
7000 series with RSP 7000	CEF is not enabled
7200	CEF is not enabled
7500	CEF is enabled – but not Distributed CEF
12000	Distributed CEF is enabled

Table 1 – Default Configuration for CEF on Various Platforms

## NetFlow

Enabling NetFlow on routers provides network administrators with access to “packet flow” information from their network. Exported NetFlow data can be used for a variety of purposes, including security monitoring, network management, capacity planning, customer billing, and Internet traffic flow analysis.

NetFlow is available on all router platforms from the 2600 series upwards from the 12.0 software release onwards. It was first introduced in 11.1CC on the 7200 and 7500 platforms. It can be enabled on a per-interface basis on the routers as in the following example:

```
interface serial 5/0
ip route-cache flow
!
```

If CEF is not configured on the router, this will turn off the existing switching path on the router and enable NetFlow switching (basically modified optimum switching). If CEF is configured on the router, NetFlow simply becomes a “flow information gatherer” and feature accelerator – CEF remains operational as the underlying switching process.

### Public Routers 1, 2, 3 Month of September Outbound Traffic

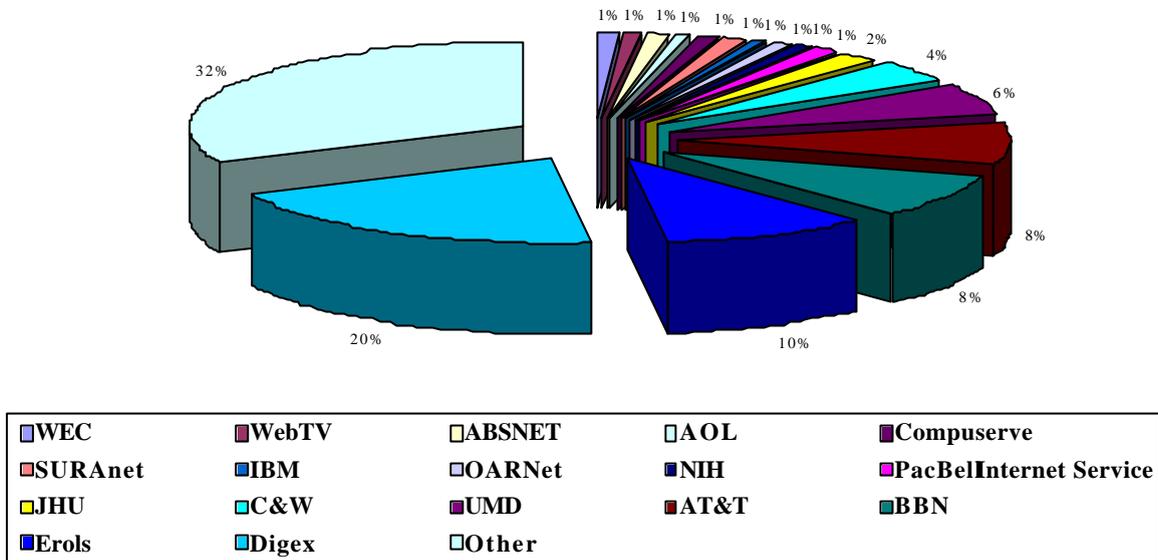


Figure 4 – Netflow in it's capacity planning role.

## NetFlow Feature Acceleration

The *NetFlow Feature Acceleration* works for a limited set of features which can take advantage of flow process short cuts. NFFA reserves space in the flow cache for state information belonging to features converted to use the flow acceleration. The features can then attach per flow state to the cache entry – using NetFlow as a quick way to access information which is flow based. For example, NetFlow Policy Routing (NPR) uses flow-acceleration to eliminate route-map checks on a per-packet basis. NetFlow feature acceleration is turned on with the following command:

```
ip flow-cache feature-accelerate
```

As of 12.0(11)S, the following features have been converted to work with NetFlow Feature-Acceleration:

- Numbered Access Lists
- Named Access Lists
- IP Accounting
- Crypto decrypt
- Crypto encrypt
- Policy Routing
- WCCP redirection

## Netflow Statistics – Basics

To view NetFlow information on the router, simply enter the command “show ip cache flow”. This will display the current flow cache on the terminal screen. An example might be:

```
gw.test.int>sh ip cache flow
IP packet size distribution (437938623 total packets):
 1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
 .000 .059 .527 .007 .258 .063 .049 .007 .001 .012 .000 .000 .000 .000 .000

 512   544   576 1024 1536 2048 2560 3072 3584 4096 4608
 .000 .000 .001 .002 .005 .000 .000 .000 .000 .000 .000
```

Wednesday, June 06, 2001

```
IP Flow Switching Cache, 4456448 bytes
1911 active, 63625 inactive, 131132988 added
1466401980 aged polls, 0 flow alloc failures
last clearing of statistics never
```

Protocol	Total	Flows	Packets	Bytes	Packets	Active(Sec)	Idle(Sec)
-----	Flows	/Sec	/Flow	/Pkt	/Sec	/Flow	/Flow
TCP-Telnet	8399	0.0	28	64	0.1	12.5	15.7
TCP-FTP	13410	0.0	3	101	0.0	1.3	9.9
TCP-FTPD	780	0.0	173	704	0.0	20.0	3.9
TCP-WWW	829901	0.5	9	450	5.3	6.3	5.6
TCP-SMTP	170303	0.1	8	226	1.0	5.6	8.7
TCP-X	72	0.0	58	278	0.0	28.7	15.2
TCP-BGP	41217	0.0	5	281	0.1	13.5	15.9
TCP-NNTP	18	0.0	1	72	0.0	3.8	16.0
TCP-Frag	22003	0.0	1	32	0.0	1.3	16.0
TCP-other	2480106	1.7	6	139	10.8	3.1	4.2
UDP-DNS	98444883	68.1	3	100	230.3	4.8	16.0
UDP-NTP	136223	0.0	1	84	0.1	1.0	16.0
UDP-TFTP	45	0.0	2	155	0.0	2.0	16.0
UDP-Frag	43	0.0	1	494	0.0	0.1	15.9
UDP-other	28243323	19.5	2	161	51.3	4.2	16.0
ICMP	694753	0.4	6	138	3.3	10.0	16.0
IP-other	47897	0.0	6	115	0.2	17.5	15.9
Total:	131133376	90.7	3	119	303.0	4.7	15.7

SrcIf	SrcIPAddress	DstIf	DstIPAddress	Pr	SrcP	DstP	Pkts
Se2/0	192.169.33.3	Fa0/0	203.37.255.97	11	C4A8	0035	23
Fa0/0	203.37.255.97	Se2/0	192.169.33.3	11	0035	C4A8	23
Fa0/0	203.37.255.97	Se2/0	192.94.123.24	11	0035	9F05	1
Se2/0	192.94.123.24	Fa0/0	203.37.255.97	11	9F05	0035	1
Se2/0	148.184.176.31	Fa0/0	203.37.255.97	11	0035	0035	1
Fa0/0	203.37.255.97	Se2/0	148.184.176.31	11	0035	0035	1
Fa0/0	203.37.255.97	Se2/0	203.23.1.50	11	0035	ED76	1
Se2/0	203.23.1.50	Fa0/0	203.37.255.97	11	ED76	0035	1
Fa0/0	203.37.255.97	Se2/0	199.203.98.25	11	0035	82E5	2
Se2/0	199.203.98.25	Fa0/0	203.37.255.97	11	82E5	0035	2
Se2/0	202.96.237.81	Fa0/0	203.37.255.97	11	0035	0035	1

The first part of the output displays the packet size distribution of the traffic flows **into** the interfaces that NetFlow is configured on. The next portion of the output displays the flows, packet size, activity etc, for the flows per well known protocol. And the final section displays the source and destination interfaces/addresses/ports for the currently active traffic flows.

It is also possible to export this collected data to a system which will collect the data allowing the ISP to carry out further analysis. There is public domain software available (cflowd from Caida, NetFlowMet from the University of Auckland, for example), as well as commercial products such as Cisco's NetFlow Collector and Analyser packages.

## Netflow Data Export

The greatest benefits of Netflow are found when its data is exported to collection systems, analysed, and processed. Cisco has adopted a broad approach to facilitate this activity. These include donations for freeware collection/analysis software, Cisco's own commercial software, tools for others to create their own software, and partnerships with companies which make commercial grade billing systems based on Netflow export.

To export the data, the following configuration commands are required:

```
ip flow-export version 5 [origin-as|peer-as]
ip flow-export destination x.x.x.x udp-port
```

The first command sets the export version to 5 (basically this includes BGP information such as AS number), and has options to include origin-as or peer-as in the exported records. The second command configures the IP address of the destination system, the NetFlow collector system, and the UDP port that the collector is listening on. Note that because the flow records use UDP it is important to design the infrastructure such that the flow collector is not too far away from the

originating router. Some ISPs who use NetFlow for billing purposes build a separate management network simply to support this function.

A new feature as from 12.0(5)S is NetFlow aggregation where summarisation/aggregation of the Flow records is carried out on the router prior to the data being exported to the collecting system. The aim here is to reduce the amount of data going across the network from router to flow collector, thereby improving the reliability of the collecting system. Flow aggregation is enabled by the following commands:

```
ip flow-aggregation cache as|destination-prefix|prefix|protocol-port|source-prefix
enabled
export destination x.x.x.x UDP-port
```

Subcommands required include “enabled” which switches on the flow aggregation, and “export destination” which lists the host which will gather the aggregated records. The collector host needs to support NetFlow type 8 records to be able to read the aggregated information.

## Turn on Nagle

The Nagle congestion control algorithm is something that many ISPs turn on to improve the performance of their telnet session to and from the router. When using a standard TCP implementation to send keystrokes between machines, TCP tends to send one packet for each keystroke typed. On larger networks, many small packets use up bandwidth and contribute to congestion.

John Nagle’s algorithm (RFC 896) helps alleviate the small-packet problem in TCP. In general, it works this way: The first character typed after connection establishment is sent in a single packet, but TCP holds any additional characters typed until the receiver acknowledges the previous packet. Then the second, larger packet is sent and additional typed characters are saved until the acknowledgement comes back. The effect is to accumulate characters into larger chunks, and pace them out to the network at a rate matching the round-trip time of the given connection. This method is usually a good for all TCP-based traffic, and helps when connectivity to the router is poor or congested, or the router itself is busier than normal. However, do not use the *service nagle* command if you have XRemote users on X Window sessions.

```
service nagle
```

Note: Without *service nagle* on a Cisco router, each character in a telnet session is a separate CPU interrupt. Hence, a command like *show tech* will force a lot of CPU interrupts – impacting the performance of the router. From a Cisco point of view, the Nagle service not only helps to optimise the telnet session, but it lessens the load on the router.

## DNS and Routers

### Mapping IP addresses to Names

Mapping Domain Names to IP addresses and vice-versa is one of those commonly overlooked areas in a new ISP’s operations. Doing a trace from Australia across the backbones in the US to a site in the UK gives you some thing like this:

```
traceroute to www.pipex.net (158.43.128.176): 1-30 hops, 38 byte packets
 1  brisbane-gw-fa20.cisco.com (144.254.153.1)  1.4 ms  0.911 ms  0.732 ms
 2  sydney-gb2.cisco.com (144.254.159.9)  29.2 ms  32.4 ms  35.9 ms
 3  sydney-wall-1.cisco.com (144.254.153.244)  35.8 ms  32.4 ms  36.4 ms
 4  telstra-gw.cisco.com (203.41.198.241)  35.2 ms  46.4 ms  32.8 ms
 5  Serial15-1-3.pad20.Sydney.telstra.net (139.130.34.205)  83.7 ms  37.2 ms  35.7 ms
 6  FastEthernet0-0-0.pad8.Sydney.telstra.net (139.130.249.228)  133 ms  43.9 ms  71.6 ms
 7  bordercore4-hssi0-0.SanFrancisco.cw.net (166.48.19.249)  260 ms  254 ms  276 ms
 8  core7.SanFrancisco.cw.net (204.70.4.93)  303 ms  260 ms  267 ms
```

Wednesday, June 06, 2001

```
 9 Hssi2-1-0.BR1.SCL1.Alter.Net (206.157.77.74) 284 ms 345 ms 433 ms
10 105.at-5-0-0.XR4.SCL1.ALTER.NET (152.63.48.182) 404 ms 496 ms 429 ms
11 195.at-1-0-0.TR4.SCL1.ALTER.NET (152.63.48.130) 324 ms 415 ms 308 ms
12 207.ATM6-0.TR2.NYC1.ALTER.NET (152.63.3.201) 343 ms 342 ms 335 ms
13 198.ATM6-0.XR2.NYC1.ALTER.NET (146.188.178.193) 333 ms 380 ms 359 ms
14 194.ATM3-0.GW1.NYC5.ALTER.NET (146.188.177.229) 347 ms 345 ms 333 ms
15 421.ATM4-0.BR1.NYC5.ALTER.NET (137.39.30.118) 384 ms 341 ms 337 ms
16 225.ATM6-0-0.CR1.LND2.Alter.Net (146.188.7.69) 407 ms 399 ms 409 ms
17 311.ATM2-0-0.GW2.LND2.Alter.Net (146.188.3.114) 423 ms 414 ms 418 ms
18 pos0-1.BR2.LND2.gbb.uu.net (146.188.5.50) 422 ms 415 ms 415 ms
19 srp4-0.cr1.uk2.london.pipex.net (158.43.233.1) 416 ms 409 ms 407 ms
20 pos0-2.cr1.doc.london.pipex.net (158.43.254.25) 408 ms 415 ms 422 ms
21 pos0-1.cr2.doc.london.pipex.net (158.43.254.66) 409 ms 414 ms 421 ms
22 pos4-0-0.cr2.cbg1.gbb.uk.uu.net (158.43.254.1) 405 ms 423 ms 408 ms
23 www.pipex.net (158.43.128.176) 408 ms * 414 ms
```

Notice that each router IP address has a corresponding DNS entry. These very descriptive DNS names help Internet users and operators understand what is happening with their connections and which route the outbound traffic is taking. The descriptive names are an invaluable aid to troubleshooting problems on the Net.

Here are some examples of descriptive DNS formats used by various ISPs:

<b>C&amp;W</b>	<b>bordercore4-hssi0-0.SanFrancisco.cw.net</b>
<b>BBN Planet</b>	<b>p2-0.paloalto-nbr2.bbnplanet.net</b>
<b>Concert</b>	<b>core1-h1-0-0.uk1.concert.net</b>
<b>Sprint</b>	<b>sl-bb6-dc-1-1-0-T3.sprintlink.net</b>
<b>DIGEX</b>	<b>sjc4-core5-pos4-1.atlas.digex.net</b>
<b>Verio</b>	<b>p0-0-0.cr1.mtvwca.pacific.verio.net</b>
<b>IiJ</b>	<b>otemachi5.iij.net</b>
<b>Qwest</b>	<b>sfo-core-03.inet.qwest.net</b>
<b>Telstra BigPond</b>	<b>Pos5-0-0.cha-core2.Brisbane.telstra.net</b>
<b>UUNET</b>	<b>ATM2-0.BR1.NYC5.ALTER.NET</b>
<b>Teleglobe</b>	<b>if-8-0.core1.NewYork.Teleglobe.net</b>
<b>VSNL</b>	<b>E3-VSB1-LVSB.Bbone.vsnl.net.in</b>

## DNS Resolver in IOS

You can specify a default domain name that the Cisco IOS software will use to complete domain name requests for functions like telnet, tftp, and other instances of name completion (e.g. **ip ospf domain-lookup**). You can specify either a single domain name or a list of domain names. Any IP host name that does not contain a domain name will have the domain name you specify appended to it before being added to the host table.

```
ip domain-name name
ip domain-list name
```

It is also advisable to include a name server for the router to resolve DNS request:

```
ip name-server server-address1 [[server-address2]...server-address6]
```

Remember that the current practice on the Internet is to quote at least two DNS resolvers. The reason is the same as for any other situation – redundancy! If one DNS server disappears, the other one can take over.

---

## SECURITY

---

*This section on IOS Security Features assumes that the ISP Engineer has a working grasp of the fundamentals of system security. If not, please review the materials listed in the reference section to help gain an understanding of some of the fundamentals. In addition, the sections below are intended to supplement the Cisco Documentation. It is assumed that the ISP Engineer will Read The Fantastic Manuals (RTFM ©) in parallel with this document.*

### Security for an ISP

Securing an enterprise network from Internet threats is easy when compared with the problems of security facing an ISP. When an enterprise network connects to the Internet, there is essentially one Internet security problem – protecting your network from outside intrusion. To achieve their Internet security objectives, an enterprise will balance tradeoffs with connectivity, accessibility, performance, and security.

An ISP's security concerns are much broader. The ISP business is all about transparent, cost effective and high performance Internet connectivity. Security measures will affect the ISP's network operation, yet security threats are real and need to be protected against. ISPs are very visible targets for malicious, vindictive, and criminal attacks, so must protect themselves, help protect their customers, and minimise the risk of their customers from becoming problems to others on the Internet.

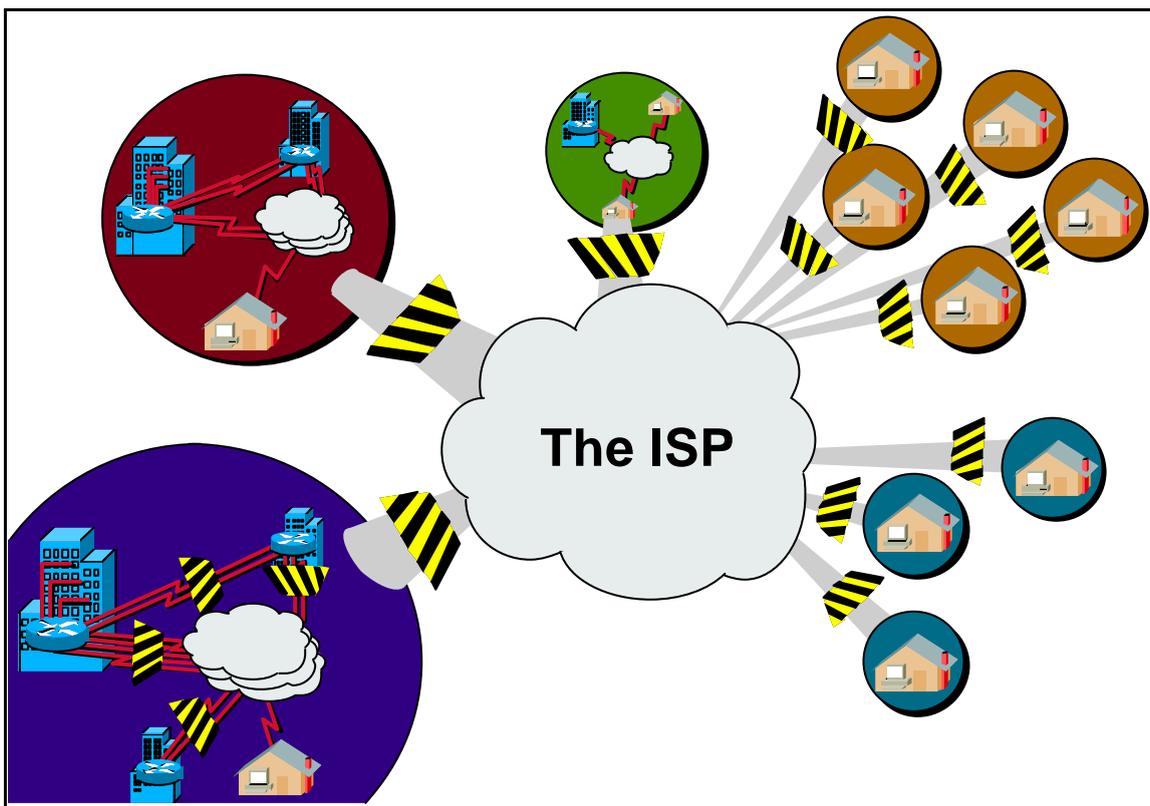


Figure 5 – ISP's Security Threats come from all Directions

Wednesday, June 06, 2001

This section describes tools that all ISPs should consider for their overall security architecture. Most of these tools are *passive* tools. Once configured, they will help prevent security problems from happening and make it more difficult to cause mischief on the ISP's network.

This section is divided into the following parts:

- Securing the Router
- Securing the Routing Protocol
- Securing the Network
- Tracking Attacks through an ISP's Network

Subdividing the section in this way allows for a hierarchy of what an ISP should work on to minimise the effects of a security incident.

**Note:** No network is ever fully *secure* or *protected*. There is always a RISK factor – especially for Service Providers whose job it is to move other people's packets across their networks (other people's packets are customer's packets). What Service Providers can expect are tools to build *resistance*. Service Providers need to resist attacks and intrusion attempts to their network – resisting long enough for internal security reaction procedures to be activated to track the incident and apply counters. This section of Cisco ISP Essentials will consistently refer to tools that *build resistance* and help build security.

## *Securing the Router*

Ensuring each device on the network is as secure as possible is one of the first security tasks. This means the mode of access, features/services turned on, and the configuration on the router all need to be reviewed from a mindset of security. Simple security principles like – “if you are not using it, do not turn it on” – can be applied to routers and switches just as easily as it can be applied to a UNIX server. Issues with how people gain access to the network devices and how their access is audited need to be considered. In today's Internet, attacks directed at the network device itself are a real threat, hence tools to help ride out the direct attack must be considered. Securing the router/switch is the first thing that needs to be completed before any tools used to defend the network are implemented. Otherwise, the router/switch becomes the back door or *Achilles Heel* of the network.

### **Global Services that are not needed or are a security risk**

Many of the built in services in IOS are not needed in an ISP backbone environment. These features should be turned off in your default configuration. Only turn them on if there are explicit requirements.

```
no service finger
no service pad
no service udp-small-servers
no service tcp-small-servers
no ip bootp server
```

Some of these will be pre-configured in IOS (depending on release) to be turned off by default, but ISPs should ensure they are explicitly turned off in the master configuration files.

The whitepaper/field alert *Defining Strategies to Protect Against UDP Diagnostic Port Denial of Service Attacks* describes the security risk and provides pointers to public discussion on the ISP Operations forums. This whitepaper is posted publicly at <http://www.cisco.com/warp/public/707/3.html>.

- *no service finger* disables the process which listens for “finger” requests from remote hosts. Only ISP personnel normally access the backbone routers, and there are other and better means of tracking who is logged in. Besides, “finger” is a known security risk in the Internet, due to its divulgence of detailed information of people logged into a system. (In 12.0 and later releases, this command has been changed to “no ip finger”.)

- *service pad* is not required. It refers back to the days of X.25 networking, and in recent versions of IOS “no service pad” has become the default.
- The small TCP and UDP servers are those with port numbers below 20 – typical services include “echo” and “discard” ports, the former echoing all packets sent to it, the latter throwing away all packets sent to it. If they are enabled and active, they could be used to carry out successful denial of service attacks – their use will divert CPU resources away from other processes which will cause problems for the connected networks and Internet service dependent on that router.
- The *bootp* service provides support for systems which find their configuration using the bootp process. This is commonly used in LANs (X-terminals commonly use bootp for example), and never on the WAN. It should be disabled.

## Interface Services That Are Not Needed or Are A Security Risk

Some IP features are great for Campus LANs, but do not make sense on an ISP backbone. Abuse of these functions by “cyberpunks” increases the ISP’s security risk. All interfaces on an ISP’s backbone router should have the following configured by *default*:

```
no ip redirects
no ip directed-broadcast
no ip proxy-arp
```

The configuration “no ip redirects” means that the router will not send redirect messages if the IOS is forced to resend a packet through the same interface on which it was received.

The configuration command “no ip directed-broadcast” means that the translation of directed broadcast to physical broadcasts is disabled. If enabled, a broadcast to a particular network could be directed at a router interface, producing effects which may be undesirable and potentially harmful. An example of the ill effects of directed broadcasts being enabled is the so-called SMURF attack. For more information about SMURF, see Craig Huegen’s SMURF website at <http://www.pentics.net/>. As from IOS 12.0, “no ip directed-broadcast” has become the default on all router interfaces.

Proxy ARP is defined in RFC 1027 and is used by the router to help hosts with no routing capability determine the MAC addresses of hosts on other networks or subnets. For example, if the router receives an ARP request for a host that is not on the same interface as the ARP request sender, and if the router has all of its routes to that host through other interfaces, then it generates a proxy ARP reply packet giving its own local MAC address. The host that sent the ARP request then sends its packets to the router, which forwards them to the intended host. This is basically the router saying that it knows how to get to the host being requested by the ARP request sender. This configuration could be undesirable on an ISP backbone as ISP networks carry explicit routing information for all destinations on the Internet. Relying on proxy ARP could result in an Internet backbone router carrying a huge MAC address table, potentially hindering the router’s performance. The configuration “no ip proxy-arp” disables the proxy-arp function.

## Cisco Discovery Protocol

The Cisco Discovery Protocol (CDP) is used for some network management functions on campus LANs etc. It allows any system on a directly connected segment to discover that the equipment is manufactured by Cisco (could be a router, switch, etc), and determine information such as the model number and the software version running. This is very useful in some instances but does not make very much sense on an ISP’s backbone where the ISP should be completely aware of what is installed and what software versions are running!

The information available from CDP does not threaten security as such, but attackers could CDP as an intelligence tool. Known bugs and are targeted through the discovery of the specific version of software on the network device. As can be seen in Example 3, CDP easily highlights the specific IOS version on the router.

Wednesday, June 06, 2001

```
Defiant#show cdp neighbors detail
-----
Device ID: Excalabur
Entry address(es):
  IP address: 4.1.2.1
Platform: cisco RSP2, Capabilities: Router
Interface: FastEthernet1/1, Port ID (outgoing port): FastEthernet4/1/0
Holdtime : 154 sec

Version :
Cisco Internetwork Operating System Software
IOS (tm) RSP Software (RSP-K3PV-M), Version 12.0(9.5)S, EARLY DEPLOYMENT MAINTEN
ANCE INTERIM SOFTWARE
Copyright (c) 1986-2000 by cisco Systems, Inc.
Compiled Fri 03-Mar-00 19:28 by htseng

Defiant#
```

### Example 3 – Information that can be gained from Cisco Discovery Protocol (CDP)

CDP may be disabled using the global command:

```
no cdp run
```

If CDP is required on an ISP's network, for what ever reason, it is possible to leave CDP running, but disable the protocol on a per interface basis. The interface configuration command:

```
no cdp enable
```

will disable CDP on a particular interface. It is strongly recommended that CDP be disabled on all public-facing interfaces, whether those face exchange points, upstream ISP, or even customers.

Note that CDP is enabled by default on 11.1CA, 11.1CC, and more recent software.

## Login Banners

Much overlooked, but important in the age of the commercial ISP is the *banner login* command. This feature is part of the "banner" command set, which displays text when users connect to the router. *Banner login* displays text when a user first initiates a telnet session to the router.

It may be seemingly trivial, but a lack of banner is as effective as a security device as a banner telling connected sessions that only those who are authorised to are permitted to connect. Some ISPs are now using banners with message content similar to the one below. Any ISP should consider whether their interest is served best by including a banner with an official warning, or nothing at all. It is good practice not to identify too much about the system itself in the banner. (Things like **joes-router** may not be such a good idea as they may give a hint about the user/owner of the system, and any user-ids or passwords on it.)

```
banner login ^
Authorized access only

This system is the property of Galactic Internet

Disconnect IMMEDIATELY if you are not an authorised user!

Contact noc@net.galaxy +99 876 543210 for help.

^
```

Another type of banner available is the "exec" banner, displayed at the time a user has successfully authenticated and logged in. For example, a note to all engineering staff on a backbone router:

```

banner exec ^
PLEASE NOTE - THIS ROUTER SHOULD NOT HAVE A DEFAULT ROUTE!

It is used to connect paying peers. These `customers' should not be able to default to us.

The configuration of this router is NON-STANDARD

Contact Network Engineering +99 876 543234 for more information.

^

```

## Use enable secret

Use *enable secret* in lieu of the *enable password* command. The encryption algorithm type 7 used in *enable password* and *service password-encryption* is reversible. The *enable secret* command provides better security by storing the enable secret password using a non-reversible cryptographic function. The added layer of security encryption it provides is useful in environments where the password crosses the network or is stored on a TFTP server.

```

service password-encryption
enable secret <removed>
no enable password9

```

Almost all passwords and other authentication strings in Cisco IOS configuration files are encrypted using the weak, reversible scheme used for user passwords. To determine which scheme has been used to encrypt a specific password, check the digit preceding the encrypted string in the configuration file. If that digit is a 7, the password has been encrypted using the weak algorithm. If the digit is a 5, the password has been hashed using the stronger MD5 algorithm. Even though *enable secret* is used for the enable password, do not forget *service password-encryption* so that the remaining passwords are stored in the configuration with type 7 encryption rather than in plain text. Weak encryption is better than none at all.

For example, in the configuration command:

```
enable secret 5 $1$iUjJ$cDZ03KKGh7mHfX2RSbDqP.
```

The *enable secret* has been hashed with MD5, whereas in the command:

```
username jbash password 7 07362E590E1B1C041B1E124C0A2F2E206832752E1A01134D
```

the password has been encrypted using the weak reversible algorithm. Since there are several versions of code designed to break the weak encryption on a Cisco, ISPs are strongly encouraged to use other strategies for passwords that are not protected by strong encryption. Cisco IOS supports Kerberos, TACACS+, and RADIUS authentication architectures, so the option is open to use AAA to get into the router versus having usernames on the router itself.

A Cisco Technical Tip, *Cisco IOS Password Encryption Facts*, explains the security model behind Cisco password encryption, and the security limitations of that encryption. It is publicly available via Cisco Connection On-Line (CCO):

<http://www.cisco.com/warp/public/701/64.html>

<sup>9</sup> A caveat though. Do not remove the *enable password* as above if the boot ROMs or boot image of the router does not support the *enable secret* configuration. The use of *secret* is supported in IOS 11.0 and later. With an older boot ROM and no *enable password* it is possible to gain access to the router without supplying any password should the router end up running the boot image due to some network problem, or malfunction. A network's first line of defense are the routers used, and anyone wishing to compromise a network will more than likely start with the router rather than any system behind that router (where configurations might be stored).

Wednesday, June 06, 2001

## The Ident Feature

Identification (*ident*) support allows you to query a Transmission Control Protocol (TCP) port for identification. This feature enables an insecure protocol, described in RFC 1413, to report the identity of a client initiating a TCP connection and a host responding to the connection. No attempt is made to protect against unauthorised queries. This command should only be enabled if the consequences and the advantages in the local situation are understood.

```
ip ident
```

Some ISP Backbone engineers like IDENT. The rest do not. New ISP Engineers are recommended to look into IDENT, read the RFC, try it, and see if it fits as a security tool on their backbone.

## Router Access – Controlling who can get into the Router

### Principles

Access to routers to carry out administrative functions can be achieved either physically through the console port, or remotely through a VTY (virtual terminal). The console port is generally only used as last resort access, with the common set-up being that the console is plugged into a “console server” which gives the ISP what is called out of band access. Most versions of IOS have support for 5 VTY ports on the router – these are the most common way of accessing the device, operating across the network and supporting multiple protocols. ISP’s commonly use telnet, with support for SSH (Secure Shell) support added from the 12.0S software release.

The following configuration guidelines are common sense:

1. Use Access Control Lists (ACLs) to restrict telnet attempts to be from source networks you trust. This is not foolproof, but it adds a layer of difficulty. It is also recommended that you include *anti-spoofing* filters on the edge of your network to prevent spoof attempts from outside your network.
2. Implement *username/password* pairs instead of the traditional password only technique of logging into a router. Using both a username and password increases the level of effort need to use brute force to crack the password. Ideally, an AAA protocol (Radius, TACACS+, or Kerberos) should be used. If an AAA protocol is used, the *username/password* pair could be used as a backup in case the AAA protocol is not working.
3. Include shorter inactivity timeouts. The inactivity timeout minimises some of the risk when the careless operator leaves their terminal logged into the router.

Each of these will now be looked at in more detail in the following sections.

### VTY and Console Port Timeouts

By default the timeout applied to all connections to the VTY, Console and AUX ports on a router is 10 minutes. This timeout is controlled by the `exec-timeout` command, as in this example:

```
line con 0
exec-timeout 5 0
line aux 0
exec-timeout 10 0
line vty 0 4
exec-timeout 5 0
```

Here the router has been told to disconnect console port and VTY connections which have been idle for more than 5 minutes and 0 seconds. The auxiliary port timeout has been set to 10 minutes.

Notice that setting the idle timeout to 0 means that the session will be left connected indefinitely. This is generally regarded as bad practice as this will hog the few available ports on the router, and could cause maintenance access problems in case of emergencies.

Furthermore, enabling TCP keepalives on incoming connections will ensure that any sessions left hanging by a remote system crash or disconnection will not block or use up the available router VTY ports. The configuration command will ensure that session are not left hanging:

```
service tcp-keepalives-in
```

### *Access List on the VTY Ports*

It is important to secure the vty ports used for telnet access with a standard ACL. By default there are no access controls on any of the VTY ports. If left this way and a password is applied<sup>10</sup> to the VTY port the router would be wide open to all comers to attempt a brute force crack against the password. The configuration below with *access-list 3* is typical:

```
aaa new-model
aaa authentication login Cisco-Lab local
!
username Cisco1 password 7 11041811051B13
!
access-list 3 permit 215.17.1.0 0.0.0.255
access-list 3 permit 215.17.34.0 0.0.0.255
access-list 3 deny any
!
line vty 0 4
access-class 3 in
exec-timeout 5 0
transport input telnet ssh
transport output none
transport preferred none
login authentication Cisco-Lab
history size 256
!
```

*Access-list 3* defines a network 215.17.1/24 and 215.17.34/24 as the only networks with access to these vtys (these networks could be the administration or NOC networks at two locations, for example). An ACL that will provide more accurate auditing is listed below:

```
access-list 199 permit tcp 215.17.1.0 0.0.0.255 any log
access-list 199 permit tcp 215.17.34.0 0.0.0.255 any log
access-list 199 deny tcp any any range 0 65535 log
access-list 199 deny ip any any log
```

This Extended ACL (xACL) will do everything that ACL 3 does above. In addition, this xACL will log TCP and IP attempts to the syslog file. The advantage with this xACL is that attempts to scan or break into the router will be logged with the TCP/IP information (i.e. the source address). Both valid and unauthorised attempts will be logged and this is one way of checking if people are attempting a brute force break-in to the router. Each attempt will be sent to the log file. If the log files are exported to a syslog server, they can be analysed for abnormal login attempts.

```
Dec 28 17:29:34.917: %SEC-6-IPACCESSLOGP: list 199 permitted tcp 144.254.193.62(1749) ->
0.0.0.0(23), 1 packet
```

See the next section for more sophisticated and strongly recommended means of protecting access.

A *timeout* of 5 minutes is applied to the interface. The second field is for “seconds”, for finer granularity. ISPs generally pick the best timeout values according to experience and the operating environment. Also, all unnecessary *transports* are removed – users of VTYS only require character access to the router, nothing else. (Other available transports such as pad,

<sup>10</sup> Telnet is disabled to any VTY port that does not have a password or some other authentication configured.

Wednesday, June 06, 2001

rlogin, and V120, not required on an ISP backbone router.) It is good practice to configure necessary transports on a per interface application basis – dialup users will only require IP transport, for example. In the case above, only telnet has been permitted to the VTY port, and no outbound connections are permitted.

If the router supports more than 5 VTYS, do not forget them! IP only software (-i- and -p- code releases) only support 5, but other feature sets can support 64 or as many as 1024 VTYS. Be sure to apply access lists to all of them if they are configured. The command `line vty 0 4` will cover the first five VTY ports. To find out how many VTYS the router will support, in configuration mode enter the following command:

```
beta7200(config)#line vty 0 ?
<1-4> Last Line number
<cr>
beta7200(config)#line vty 0 _
```

The router will print the next two lines above, listing the VTY range supported. It will then give you the configure command prompt back as above. Use this to ensure that all the VTYS on the router are actually protected by an access-list.

## ***VTY Access and SSH***

Before 12.0S software, the only method really used to access the VTY ports was telnet. Rlogin has been used by some ISPs, especially for executing one-off commands but the protocol is insecure and can't be recommended. SSH version 1 support has now been added, giving ISPs greater flexibility and some security when accessing their equipment across the Internet.

Before SSH can be configured, the router needs to be running a cryptographic image which supports SSH. The standard Service Provider (-p-) images do not have SSH support as the US Government restricts the export of strong encryption. Images supporting single DES are freely available for download – in 12.0S they might be called something like **c7200-k3p-mz.120-14.S**. 3DES images are available on CCO after an approval application has been submitted. The approval application form can be found at [http://www.cisco.com/cgi-bin/crypto/crypto\\_main.pl](http://www.cisco.com/cgi-bin/crypto/crypto_main.pl). Once the application has been approved, permission will be granted to download the necessary images.

Once the appropriate cryptographic image is running, SSH needs to be set up on the router. The following sequence of configuration commands gives an example of how this may be achieved:

```
beta7200(config)#crypto key generate rsa
```

and select a key size of at least 1024 bits. After this, add `ssh` as the input transport on the vtys:

```
line vty 0 4
transport input telnet ssh
```

It is now possible to use SSH to access the router. If the IOS image supporting only single DES is being used, the SSH client on the operator's system needs to support DES (most SSH clients have this disabled by default in preference of 3DES as single DES is no longer considered very secure). This may require the client to be recompiled (if this is an option).

**Note:** a username/password pair **must** be configured on the router before SSH access will work. However, it is strongly recommended that AAA is used to authenticate users (see below) as this is the preferred way of securing the router.

```
ssh beta7200 -l philip
```

is the Unix command to connect to the router `beta7200` using `ssh` with the username "philip".

## ***User authentication***

It is good practice to register each individual user with a separate user-id. If a generic account is set up, it is easier for it to fall into the wrong hands, and there is virtually no accountability, resulting in abuse of access, and potential malfunction of

the network. In addition, if the default password only login is used, it becomes very easy to use a brute force crack utility to get the password. A *username/password* pair makes brute force techniques harder, but not impossible.

This example shows one way of configuring user-ids. It is practical for networks of a few routers, but does not scale, and suffers from the weak type-7 encryption. (It should be avoided but is given here for completeness.)

Configuring:

```
username joe password 7 045802150C2E
username jim password 7 0317B21895FE
!
line vty 0 4
  login local
!
```

on the router will change the login prompt sequence from:

```
Password:
to:
Username:
Password:
```

where the username requested is from those listed above. Each user will have to supply the password on request.

### *Using AAA to Secure the Router*

The preferred and recommended method is to use an AAA protocol like TACACS+, Radius, or Kerberos. Here all the users who have access to the routers have their usernames and passwords held at a central location, off the router. This has several advantages:

- Recall that the encryption method 7 is reversible. Anyone who has access to the router configuration could potentially work out the password and gain access to the system.
- Scalability. If there is a new user, or a user leaves the ISP, it is easy to change the password database once. Changing it on many different routers becomes a considerable task.
- Passwords are held in Unix encrypted format on the central AAA server. The algorithm for Unix password encryption is not reversible, so is more secure.
- All accesses are logged to the AAA server. In fact, some AAA software will allow all actions on the router to be logged.

Commercial Cisco ACS software is available for Windows NT and Sun Solaris systems. A TACACS+ server for Unix platforms is available on Cisco's Engineering ftp site at [ftp://ftp-eng.cisco.com/pub/tacacs/tac\\_plus.F4.0.4.alpha.tar.Z](ftp://ftp-eng.cisco.com/pub/tacacs/tac_plus.F4.0.4.alpha.tar.Z). On the router, a typical tacacs+ configuration would be:

```
aaa new-model
aaa authentication login default tacacs+ enable
aaa authentication enable default tacacs+ enable
aaa accounting exec start-stop tacacs+

ip tacacs source-interface Loopback0
tacacs-server host 215.17.1.2
tacacs-server host 215.17.34.10
tacacs-server key CKr3t#
```

To explain these commands, the `authentication login` states that TACACS+ should be used for login authentication – if the TACACS+ servers are not reachable, the local enable secret is used. The `authentication enable` command states that TACACS+ should be used to authenticate the use of the enable command. The enable password should be taken from the TACACS+ server before using the local enable secret.

Wednesday, June 06, 2001

Note the use of the Loopback interface as the source of TACACS+ requests (reasons as in previous examples), and the use of two TACACS+ servers for redundancy and resilience.

If the router is running an IOS version which does not support TACACS+ (pre 11.0), it is strongly advised that it is upgraded to at least 11.0, as the more recent software has more features appropriate to ISP's as discussed in this document.

## **Router Command Auditing**

*It is a bad day at the office. You have had to fire an engineer on the operations team. It was a rough experience with words exchanged. You have briefed your team and people are starting to changing things on the network – removing the employees access. When suddenly, alarms trigger on all sorts of equipment in the NOC. A network wide outage is spreading fast. Is it an attack? Is it a routing protocol collapse? What is happening!?!? Devices are failing through out the network and customer lines are ringing off the hook. Someone from your team uses the out-of-band access to get into one of the effected switches. Horror! The entire config has been erased along with the software image. Other switches and routers are checked, they also have their configs erased and software images erased. It is sabotage! You know who did it, but how are you going to prove it!*

Yes, sabotage from disgruntled employees has happened and will happen again. If for no other reason, the two of the primary advantages of using TACACS+ – centralised account control and command auditing – provide enough motivation for ISPs to protect themselves from situations like the previous story. With centralised control, the employee's account can be removed before he was given notice. If the employee had backdoor access via another account, then any commands done on the switches and routers would be audited, providing records that can be used in a court of law. In fact, from a security point of view, we should view AAA as Authentication, Authorisation, and Auditing – which of course is one of the strengths TACACS+ has over Radius in the security domain.

AAA on the router and TACACS+ server can be configured to track all commands or a limited set of commands typed into the router. AAA Command accounting provides information about the EXEC shell commands for a specified privilege level that are being executed on a router. Each command accounting record includes a list of the commands executed for that privilege level, as well as the date and time each command was executed, and the user who executed it.

The following example shows the information contained in a TACACS+ command accounting record for privilege level 1:

```
Wed Jun 25 03:46:47 1997      172.16.25.15      fgeorge  tty3      5622329430/4327528  stop
task_id=3      service=shell      priv-lvl=1      cmd=show version <cr>
Wed Jun 25 03:46:58 1997      172.16.25.15      fgeorge  tty3      5622329430/4327528  stop
task_id=4      service=shell      priv-lvl=1      cmd=show interfaces Ethernet 0 <cr>
Wed Jun 25 03:47:03 1997      172.16.25.15      fgeorge  tty3      5622329430/4327528  stop
task_id=5      service=shell      priv-lvl=1      cmd=show ip route <cr>
```

The following example shows the information contained in a TACACS+ command accounting record for privilege level 15:

```
Wed Jun 25 03:47:17 1997      172.16.25.15      fgeorge  tty3      5622329430/4327528  stop
task_id=6      service=shell      priv-lvl=15      cmd=configure terminal <cr>
Wed Jun 25 03:47:21 1997      172.16.25.15      fgeorge  tty3      5622329430/4327528  stop
task_id=7      service=shell      priv-lvl=15      cmd=interface Serial 0 <cr>
Wed Jun 25 03:47:29 1997      172.16.25.15      fgeorge  tty3      5622329430/4327528  stop
task_id=8      service=shell      priv-lvl=15      cmd=ip address 1.1.1.1 255.255.255.0 <cr>
```

The next example shown in Figure 6 is taken from the Cisco Secure AAA Server and demonstrates how useful command auditing can be.

Configuration control and audit of who has done what and when on the routers is the key objective for using AAA command accounting on an ISP's backbone.

User-Name	Group-Name	cmd	priv-lvl	service	NAS-Portname	task_id	NAS-IP- reason
bgreene	NOC	enable <cr>	0	shell	tty0	4	210.210.51.224
bgreene	NOC	exit <cr>	0	shell	tty0	5	210.210.51.224
bgreene	NOC	no aaa accounting exec Workshop	0	shell	tty0	6	210.210.51.224
bgreene	NOC	exit <cr>	0	shell	tty0	8	210.210.51.224
pfs	NOC	enable <cr>	0	shell	tty0	11	210.210.51.224
pfs	NOC	exit <cr>	0	shell	tty0	12	210.210.51.224
bgreene	NOC	enable <cr>	0	shell	tty0	14	210.210.51.224
bgreene	NOC	show accounting <cr>	15	shell	tty0	16	210.210.51.224
bgreene	NOC	write terminal <cr>	15	shell	tty0	17	210.210.51.224
bgreene	NOC	configure <cr>	15	shell	tty0	18	210.210.51.224
bgreene	NOC	exit <cr>	0	shell	tty0	20	210.210.51.224
bgreene	NOC	write terminal <cr>	15	shell	tty0	21	210.210.51.224
bgreene	NOC	configure <cr>	15	shell	tty0	22	210.210.51.224
bgreene	NOC	aaa new-model <cr>	15	shell	tty0	23	210.210.51.224
bgreene	NOC	aaa authorization commands 0 de	15	shell	tty0	24	210.210.51.224
bgreene	NOC	exit <cr>	0	shell	tty0	25	210.210.51.224
bgreene	NOC	ping <cr>	15	shell	tty0	32	210.210.51.224
bgreene	NOC	show running-config <cr>	15	shell	tty66	35	210.210.51.224
bgreene	NOC	router ospf 210 <cr>	15	shell	tty66	45	210.210.51.224
bgreene	NOC	debug ip ospf events <cr>	15	shell	tty66	46	210.210.51.224

Figure 6 – Command Auditing on the Router via Cisco Secure and TACACS+

```

aaa new-model
aaa authentication login default tacacs+ enable
aaa authentication enable default tacacs+ enable
aaa accounting command 15 start-stop tacacs+
aaa accounting exec start-stop tacacs+

ip tacacs source-interface Loopback0
tacacs-server host 215.17.1.2
tacacs-server host 215.17.34.10
tacacs-server key CKr3t#

```

**Note:** When command accounting is enabled as above, all commands (i.e. keystrokes) sent to the router in enabled mode will be logged in the accounting file on the accounting host. Be aware that when changing sensitive configurations on the router that these changes will be recorded in the accounting host log file. One example is where the last resort password (enable secret) is changed during an online session on the router – the new password will be recorded in full in the accounting file. Of course, the recommended way to change such a last resort password is to use tftp to copy the necessary configuration (including the first line disabling the accounting command, and the last line re-enabling it again!) from a tftp-server, and never to make such changes live on the router.

### Full Example

The following is a full example putting all the techniques together.

```

service password-encryption
!
aaa new-model
aaa authentication login default tacacs+ enable
aaa authentication login Cisco-Lab local enable
aaa authentication enable default tacacs+ enable
aaa accounting exec start-stop tacacs+
!
username Cisco1 password 7 11041811051B13
enable secret <removed>
!
access-list 199 permit tcp 215.17.1.0 0.0.0.255 any
access-list 199 permit tcp 215.17.34.0 0.0.0.255 any
access-list 199 deny tcp any any range 0 65535 log
access-list 199 deny ip any any log
!

```

Wednesday, June 06, 2001

```
ip tacacs source-interface Loopback0
tacacs-server host 215.17.1.2
tacacs-server host 215.17.34.10
tacacs-server key CKr3t#
!
line vty 0 4
access-class 199 in
exec-timeout 5 0
transport input telnet ssh
transport output none
transport preferred none
login authentication Cisco-Lab
history size 256
!
```

## ***Securing the Routing Protocol***

How do you know the routing updates from one of your internal backbone routers *really* came from a neighbouring router on your backbone? You do not – unless neighbour authentication is used. Theoretically, routing information can be spoofed and injected into an ISP's backbone. The horror of seeing a normal 95000 entry routing table jump to 200000 or 500000 entries, then seeing these propagate all over the Internet, has been quietly talked about in the halls of Internet operations meetings. ISPs are strongly encouraged to prevent their routers from receiving fraudulent route updates by protecting the routing protocol updates.

Cisco IOS has several tools to help protect the routing protocols from intentional or unintentional attacks. The first and primary mechanism to protect route protocol updates is through configuring router authentication. MD5 is a valuable tool that will validate the authentication of a routing update. The second tool is the extended ACL. ACLs should be used strategically through out the network to validate the source / destination address of packets headed for the IGP and EGP ports. ACLs make it more difficult for DOS/DDOS attacks from targeting the routing protocol. Finally, there are specific commands and BCPs in the routing protocols that help protect them from attack. For example, BGP's *maximum-prefix* command alerts the operators and optionally shuts down the BGP session whenever the maximum prefix limit is reached. This protects the router from being overwhelmed by the number of updates or having its memory completely consumed – potentially crashing the router. This section will review these tools and techniques so ISP Engineers will have a better understanding of how to protect their routing protocols.

## **Authenticating Routing Protocol Updates**

Neighbour router authentication is part of an ISP's total security plan. This section describes what neighbour router authentication is, how it works, and why it should be used to increase overall network security. Documentation details can be found at:

[http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed\\_cr/secur\\_c/seprt5/scrouter.htm](http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed_cr/secur_c/seprt5/scrouter.htm)

## ***Benefits of Neighbour Authentication***

When configured, neighbour authentication occurs whenever routing updates are exchanged between neighbour routers. This authentication ensures that a router receives reliable routing information from a trusted source. Without neighbour authentication, unauthorised or deliberately malicious routing updates could compromise the security of your network traffic. A security compromise could occur if an unfriendly party diverts or analyses your network traffic. For example, an unauthorised router could send a fictitious routing update to convince your router to send traffic to an incorrect destination. This diverted traffic could be analysed to learn confidential information of your organisation, or merely used to disrupt your organisation's ability to effectively communicate using the network. Neighbour Authentication prevents any such fraudulent route updates from being received by your router.

## ***Protocols That Use Neighbour Authentication***

Neighbour authentication can be configured for the following routing protocols:

---

Cisco Systems Inc  
170 West Tasman Drive.  
San Jose, CA 95134-1706  
Phone: +1 408 526-4000  
Fax: +1 408 536-4100

- Border Gateway Protocol (BGP)
- Dynamic Routing Protocol (DRP) Server Agent
- Intermediate System-to-Intermediate System (IS-IS)
- IP Enhanced Interior Gateway Routing Protocol (EIGRP)
- Open Shortest Path First (OSPF)
- Routing Information Protocol version 2 (RIPv2)

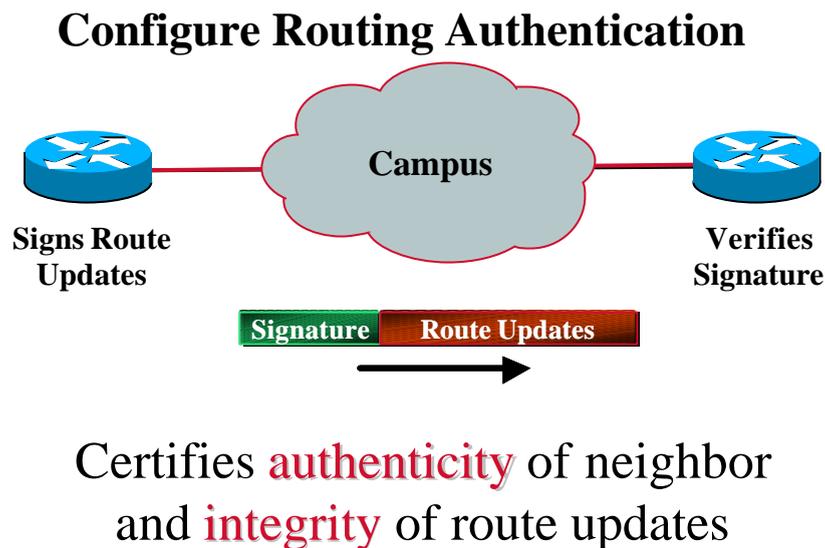


Figure 7 – Router Authentication

### *When to Configure Neighbour Authentication*

You should configure any router for neighbour authentication if that router meets all of these conditions:

- The router uses any of the routing protocols previously mentioned.
- It is conceivable that the router might receive a false route update.
- If the router were to receive a false route update, your network might be compromised.
- If you configure a router for neighbour authentication, you also need to configure the neighbour router for neighbour authentication.

### *How Neighbour Authentication Works*

When neighbour authentication has been configured on a router, the router authenticates the source of each routing update packet that it receives. This is accomplished by the exchange of an authenticating key (sometimes referred to as a password) that is known to both the sending and the receiving router. There are two types of neighbour authentication used: plain text authentication and Message Digest Algorithm Version 5 (MD5) authentication. Both forms work in the same way, with the exception that MD5 sends a “message digest” instead of the authenticating key itself. The message digest is created using the key and a message, but the key itself is not sent, preventing it from being read while it is being transmitted. Plain text authentication sends the authenticating key itself over the wire.

**Note:** Plain text authentication is not recommended for use as part of your security strategy. Its primary use is to avoid accidental changes to the routing infrastructure. Using MD5 authentication, however, is a recommended security practice.

**CAUTION: As with all keys, passwords, and other security secrets, it is imperative that you closely guard the keys used in neighbour authentication. The security benefits of this feature are reliant upon your keeping all authenticating keys confidential. Also, when performing router management tasks via Simple Network Management Protocol (SNMP), do not ignore the risk associated with sending keys using non-encrypted SNMP.**

### Plain Text Authentication

Each participating neighbour router must share an authenticating key. This key is specified on each router during configuration. Multiple keys can be specified with some protocols; each key must be identified by a key number. In general, when a routing update is sent, the following authentication sequence occurs:

**Step 1:** A router sends a routing update with a key and the corresponding key number to the neighbour router. For protocols that can have only one key, the key number is always zero.

**Step 2:** The receiving (neighbour) router checks the received key against the same key stored in its own memory.

**Step 3:** If the two keys match, the receiving router accepts the routing update packet. If the two keys did not match, the routing update packet is rejected.

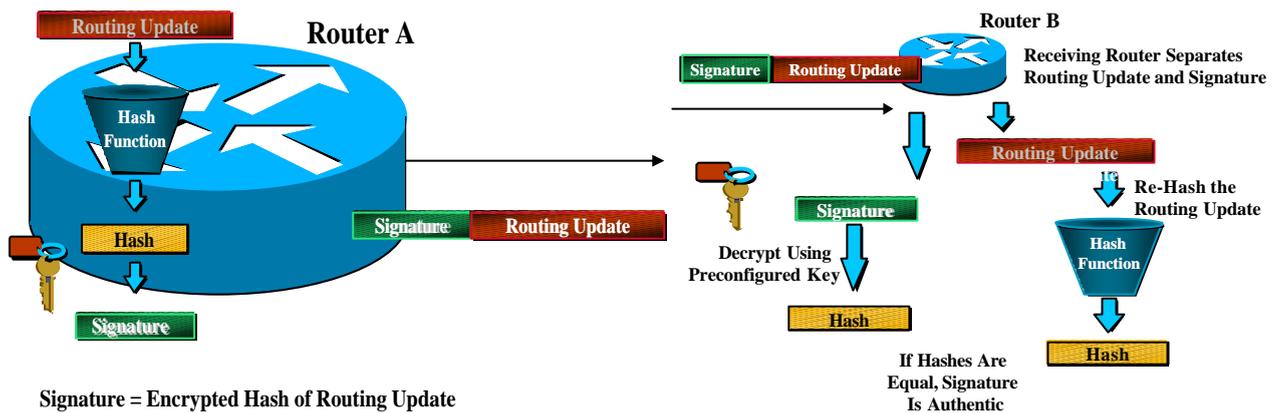


Figure 8 – How Routing Protocol Authentication Operates

### MD5 Authentication

MD5 authentication works similarly to plain text authentication, except that the key is never sent over the wire. Instead, the router uses the MD5 algorithm to produce a “message digest” of the key (also called a “hash”). The message digest is then sent instead of the key itself. This ensures that nobody can eavesdrop on the line and learn keys during transmission. These protocols use MD5 authentication:

- OSPF
- RIP version 2
- BGP
- EIGRP
- ISIS

### Routing Protocol Authentication Summary

The following are some examples of how router authentication is configured in OSPF, ISIS, and BGP:

## OSPF

```
interface ethernet1
ip address 10.1.1.1 255.255.255.0
ip ospf message-digest-key 100 md5 cisco
!
router ospf 1
network 10.1.1.0 0.0.0.255 area 0
area 0 authentication message-digest
```

## ISIS

```
interface ethernet0
ip address 10.1.1.1 255.255.255.0
ip router isis
isis password cisco level-2
```

## BGP

```
router bgp 200
no synchronization
neighbor 4.1.2.1 remote-as 300
neighbor 4.1.2.1 description Link to Excalabur
neighbor 4.1.2.1 send-community
neighbor 4.1.2.1 version 4
neighbor 4.1.2.1 soft-reconfiguration inbound
neighbor 4.1.2.1 route-map Community1 out
neighbor 4.1.2.1 password cisco
```

## Securing the Network

The next step to ISP security is building resistance to attacks and intrusions into the network. For an ISP, this is divided into three major areas:

- Route Filtering – Controlling which routes are advertised and received from sources outside of your autonomous system (AS).
- Packet Filtering – Ensuring that packets that enter the network are valid and have a source address matching the origin.
- Rate Limiting – Limiting the amount of some types of traffic (like ICMP) that has no valid reason to consume huge chunks of bandwidth.

These three techniques are the building blocks for securing an ISP's network against attacks.

## Egress and Ingress Filtering

Egress and ingress filtering are a critical part of an ISP's router configuration strategy. These terms are relative to the network ingress/egress filtering are applied:

- *Ingress Filtering* applies filters to traffic coming **into** a network from outside (see Figure 9). This can be from an ISP's customers and/or from the Internet at large.
- *Egress Filtering* applies a filter for all traffic **leaving** an ISP's networks (see Figure 9). It is applied to information leaving the network to the Internet and/or customer networks.

Be mindful that these terms are relative to the specific network's point of view. For example, ISP B's egress traffic is ISP A's ingress traffic.

Ingress/Egress filters help protect an ISP's resources, its customers' networks, allows it to enforce policy, and minimises the risk of being the network chosen by hackers to launch an attack on other networks. ISPs are strongly encouraged to

Wednesday, June 06, 2001

develop strategies using egress and ingress filtering to protect themselves from their customers and the Internet at large. By protecting themselves, the ISPs are working towards protecting the Internet in general.

BCP38/RFC2827 *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*: P. Ferguson, D. Senie, May 2000, <http://www.ietf.org/rfc/rfc2827.txt> provides general guidelines for all ISPs on Ingress and Egress filtering. The Internet community now considers this document Best Current Practice, and all service providers should follow its recommendations.

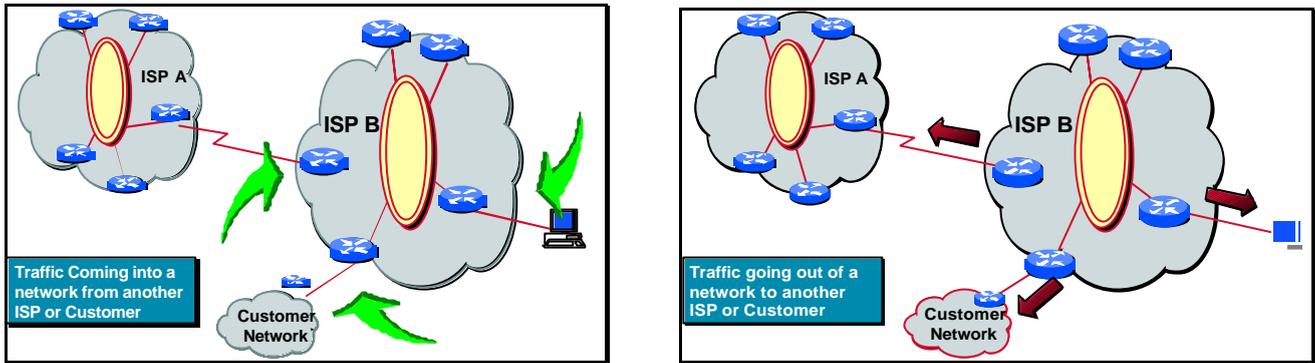


Figure 9 – Ingress and Egress Filtering

## Route Filtering

Route filtering removes specific routes from routing protocol advertisements. From the context of the Internet, route filtering is applied to the BGP ingress and egress advertisements with an ISP's customers and peers. In today's Internet, route filtering is done to meet two objectives. The first objective is to keep specific networks that should not be advertised on the Net from propagating to other networks. For example, RFC 1918 (Private address space), host subnet (127.0.0.0/8), and multicast addresses (if multicast BGP is not used) should be filtered from an ISP's ingress and egress advertisements<sup>11</sup>. The second objective is to filter smaller network advertisements out of larger CIDR blocks. Some ISPs *leak* out a more specific route from a larger CIDR block (sometimes intentionally for operational reasons, more often accidentally or through lack of knowledge). These leaks increase the rate of growth of the Internet Route Table. To discourage route leaking, some ISPs route filter on the prefix boundaries of the Regional Internet Registry's (RIR's) minimal IPv4 allocation block.<sup>12</sup> The ISPs who do this are often referred to a *Net Police* – hence the name of this type of routing filter is called **Net Police Filtering**. Examples of a routing filter to keep unregistered blocks out and Net Police Filtering are provided in this section.

### *Route Filtering: Networks that should not be advertised on the Internet*

As mentioned earlier, some networks are reserved for special functions on the Internet. These networks should not appear in the Internet Route Table. Some examples of which networks should not show up in the Internet Route Table are:

- 0.0.0.0/0 & 0.0.0.0/8 – Default and network 0 (unique and now historical properties)
- 127.0.0.0/8 – Host Loopback
- 192.0.2.0/24 – TEST-NET generally used for examples in vendor documentation
- 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16 – RFC 1918 Private Addresses
- 169.254.0.0/16 – End node auto-config for DHCP
- Any advertisements less than /24
- Any network from an Internet eXchange Point (IXP) medium (i.e. like the ethernet segment)

<sup>11</sup> Experiments with Multicast BGP (MBGP) have now begun on some ISP sites. Deployment of MBGP will require a re-think and re-design of the egress/ingress route filters.

<sup>12</sup> Currently, the minimal allocation block for the RIRs are /20s.

These last two could be controversial:

- A /24 (the historical Class C address) has been the smallest block of IPv4 addresses ever allocated from an IP registry. Hence, there should be no advertisements less than a /24 on the Internet. Yet, for whatever reason, networks below /24 do appear in the Internet Routing Table. So it is up to the ISP to decide whether they wish to filter them.
- Filtering the IP networks of the IXP's – is an option many ISPs select. IXPs are generally not locations where some ISP provide transit to the Internet. The IXP network should not be advertised out to the rest of the Internet, so many ISPs filter this prefix on the ingress and egress.

The networks which should not be advertised have been documented by Bill Manning in his Internet Draft, the current version being <http://www.ietf.org/internet-drafts/draft-manning-dsua-06.txt>.

The following is a list of some of the US IXP networks in ACL format.<sup>13</sup> If anyone wants to implement this, they should first check that these are still valid IXP network blocks, and that they are prepared to put in the effort to maintain them. Also, it might be worth extending this list to include all the exchange points in the world ([www.ep.net](http://www.ep.net)), a potentially harder job.

```

! MAE-East primary IP block
access-list 150 deny ip 192.41.177.0 0.0.0.255 0.0.0.0 255.255.255.0
!
! MAE-East secondary IP block
access-list 150 deny ip 198.32.186.0 0.0.0.255 0.0.0.0 255.255.255.0
!
! MFS Portion of MAE-West
access-list 150 deny ip 198.32.136.0 0.0.0.255 0.0.0.0 255.255.255.0
!
! NASA/AMES - NASA Portion of MAE-West
access-list 150 deny ip 198.32.184.0 0.0.0.255 0.0.0.0 255.255.255.0
!
! Ameritech Advanced Data Services - AADS
access-list 150 deny ip 198.32.130.0 0.0.0.255 0.0.0.0 255.255.255.0
!
! Digital's Palo Alto Internet eXchange - PAIX
access-list 150 deny ip 198.32.176.0 0.0.0.255 0.0.0.0 255.255.255.0
!
! Sprint NAP - Pennsauken
access-list 150 deny ip 192.157.69.0 0.0.0.255 0.0.0.0 255.255.255.0
!
! Pac*Bell NAP
access-list 150 deny ip 198.32.128.0 0.0.0.255 0.0.0.0 255.255.255.0
!
! MAE-LA
access-list 150 deny ip 198.32.146.0 0.0.0.255 0.0.0.0 255.255.255.0
!
! MAE-Dallas
access-list 150 deny ip 198.32.138.0 0.0.0.255 0.0.0.0 255.255.255.0
!
! MAE-Houston
access-list 150 deny ip 198.32.150.0 0.0.0.255 0.0.0.0 255.255.255.0
!

```

Thus far, we have been talking about applying the router filters on the ingress and egress. Why apply the filter to both ingress and egress? Ingress route filtering prevents routes from entering your network. This would keep the information from causing problems in your network. Egress routing filtering protects others – just in case something goes wrong with your network advertisements. Applying both ingress and egress is the logical way to protect your network while keeping any local problems isolated from the rest of the Internet. Hence, the recommendation that ISPs apply ingress and egress filters the edge of the network.

There are two techniques used in the Internet today to perform route filtering – BGP Distribute List and BGP Prefix list. The performance difference between the two is negligible. The essential difference is personal preference and usability. Distribute lists are widely used in the ISP community. So many ISPs used them for route filtering because they work (remember, *if it ain't broken, don't fix it*). Prefix Lists are increasingly being used by ISPs for route filtering. They are

<sup>13</sup> Patrick W. Gilmore of PRIORI NETWORKS for supplying the list.

Wednesday, June 06, 2001

somewhat easier to configure (using natural network/prefix length representation) and have the nice feature of names to help keep straight which list does what on a router.

Here is an example configuration and filter using BGP Prefix-List:

```
router bgp 200
  no synchronization
  neighbor 220.220.4.1 remote-as 210
  neighbor 220.220.4.1 version 4
  neighbor 220.220.4.1 prefix-list rfc1918-sua in
  neighbor 220.220.4.1 prefix-list rfc1918-sua out
  neighbor 222.222.8.1 remote-as 220
  neighbor 222.222.8.1 version 4
  neighbor 222.222.8.1 prefix-list rfc1918-sua in
  neighbor 222.222.8.1 prefix-list rfc1918-sua out
  no auto-summary
!
ip prefix-list rfc1918-sua deny 0.0.0.0/0 ! Default
ip prefix-list rfc1918-sua deny 0.0.0.0/8 le 32 ! Network Zero
ip prefix-list rfc1918-sua deny 10.0.0.0/8 le 32 ! RFC1918
ip prefix-list rfc1918-sua deny 127.0.0.0/8 le 32 ! Hostnet
ip prefix-list rfc1918-sua deny 169.254.0.0/16 le 32 ! Non-DHCP Autoconfig
ip prefix-list rfc1918-sua deny 172.16.0.0/12 le 32 ! RFC1918
ip prefix-list rfc1918-sua deny 192.0.2.0.0/24 le 32 ! TESTNET
ip prefix-list rfc1918-sua deny 192.168.0.0/16 le 32 ! RFC1918
ip prefix-list rfc1918-sua deny 224.0.0.0/3 le 32 ! Multicast
ip prefix-list rfc1918-sua deny 0.0.0.0/0 ge 25 ! prefixes >/24
ip prefix-list rfc1918-sua permit 0.0.0.0/0 le 32 ! Everything Else
!
```

## *Effects of CIDRisation*

One of the most critical issues that could threaten the stability of the Internet is the size of the global Internet Route Table. The table's growth influences the scalability, increases operational/capital cost, and has posed a security risk to the Internet. For a variety of reasons, ISPs through out the world have injected all sorts of networks into the Internet ranging from /8s (old class As) to /32s (host routes). The result is rapid growth of the Internet Route Table.

CIDR – Classless Inter-Domain Routing – has had a noticeable and significant impact on the growth of the table. CIDR is a combination of revised IPv4 allocation policies (i.e. today's norm of provider based addressing), updates to routing protocols (RIPv2, OSPF, ISIS, and BGP), and new address aggregation techniques to ensure that only the large allocated blocks of addresses get advertised into the global Internet Route Table. The impact of CIDR moved the growth rate from an exponential curve to linear growth.

Even though the introduction of CIDR did have an effect, it did not have as longterm an effect as many providers and industry observers hoped it would. The linear growth rate was still fairly steep. Some operators on the Internet began to feel that the real problem was lazy ISPs who either did not care or did not know how to do proper aggregation. Two approaches were tried to encourage ISPs to do the right thing. The first approach was the weekly posting of the Internet CIDR Report to the core operational mailing lists. The second approach was to implement filters on the routes announced by each ISP, basically only accepting the major allocated CIDR blocks and no more specifics. Both approaches are in use today and are items that all ISPs need to be aware of and consider.

The CIDR Report from Tony Bates ([tbates@cisco.com](mailto:tbates@cisco.com)) is a weekly analysis of the rate of growth of the Internet Route Table. The report – sent to all the major operations mailing list – provides the name of ISPs injecting the most prefixes, aggregated and un-aggregated. Everyone saw who was causing the growth and could now apply peer pressure to have the corrections made. Table 2 shows the growth of the Internet Route Table. Between late 1995 and 1998, peer pressure helped to stabilise the growth, and indeed helped slow down the growth rate by late 1998 as can be seen in Table 2. As newer routers with faster processors and more memory capacity were deployed, ISPs became less concerned with the growth in the routing table and hence lost interest in providing or accepting peer pressure. The CIDR reports are interesting, but losing their earlier impact, and arguably no longer serving their original function.

**List of sites collecting data on the Internet Route Table:**

Tony Bates' Weekly/Daily CIDR Report - <http://www.employees.org:80/~tbates/cidr-report.html>

Philip Smith's Weekly Routing Analysis - <http://www.apnic.net/stats/bgp/>

Telstra's AS1221 BGP Table Data - <http://www.telstra.net/ops/bgp.html>

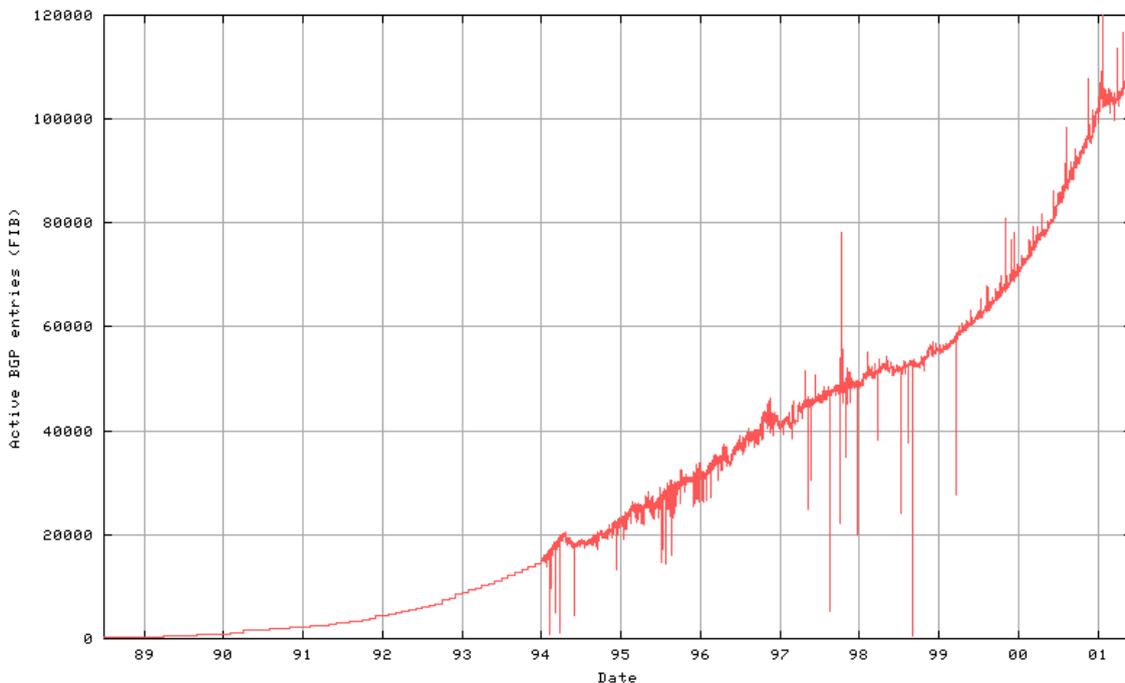
The second approach to encouraging ISPs to use proper aggregation was to install inbound prefix filters on route announcements. Sean Doran <[smd@clock.org](mailto:smd@clock.org)> was one of the first to notice and act on the fact that the new CIDR based allocation policies from the three RIRs were not affecting the ISP's announcements to the Internet. In Sean's words:

*"What triggered the filter was the observation that the blocks freshly allocated by all three registries were very poorly aggregated. More annoyingly, those allocated to Sprint's principal peers (most notably Internet MCI) demonstrated the worst aggregation; in one case a /14 was announced almost exclusively as prefixes no shorter than 19 bits."*<sup>14</sup>

In essence, CIDR was not being implemented as originally intended. Sprint placed one of the first filters that enforced the RIR's default allocation size (see Figure 10). This filter blocked any announcement from any ISP which was longer than their allocated address block. For example, an ISP was allocated a /19 from RIPE NCC's 62/8 CIDR block; this is all that should be announced to the Internet. The RIPE NCC had published their minimum allocation size, so Sprint could construct a filter that would only accept a /19 or less from 62/8. So a /18 would pass the filter, a /20, /21, /22, /23, /24, etc would not pass.

The operational effects of these filters have been variable. Several ISPs joined Sprint in implementing such filters. All three RIRs have assisted the ISP community by publishing their default allocation size for all of their CIDR blocks (see Figure 10 for list of URLs). Today, all three RIRs have a minimal allocation size of /20 on their new CIDR blocks making it potentially easier for any ISP to filter on the registry allocation sizes if they so wish.

**Table 2 – BGP Route Table (5 June 2001) from Telstra's AS 1221 Point of View**



<sup>14</sup> NANOG Mailing List - Subject: Re: too many routes - From: "Sean M. Doran" <[smd@clock.org](mailto:smd@clock.org)>  
Date: 10 Sep 1997 20:33:09 -0400 (see [www.nanog.org](http://www.nanog.org) for mailing list archives).

Wednesday, June 06, 2001

## Do Net Police Filters help secure a Network?

There have been incidents on the Internet where a rogue router (intentionally or unintentionally) *deaggregated* the address blocks and started advertising lots of small – more specific – networks. In one incident, the Internet Route Table grew from 50K routes to more than 120K routes over a period of several minutes. ISP Security people have been mindful that this sort of attack is theoretically feasible. Yet, it has also proven that the Net Police filter is a deterrent to this sort of attack. For this sort of attack to work, a router attached to the Internet speaking BGP must take a large CIDR block (say a /19) and advertise many more specifics (say all possible /32s – 8192 networks). A Net Police filter ensuring that a /19 or less (i.e. /18) would pass the filter and all other more specifics are dropped. The result is that the attack would have minimal effect<sup>15</sup> on the ISP with the Net Police filter, while other ISPs with out the Net Police filter will get a huge influx of new announcements – overloading the memory of their routers and potentially creating chaos on the Net.

**Where to get the list information on each RIR's minimal allocation size for each of their macro address blocks:**

APNIC – <http://www.apnic.net/docs/add-manage-policy.html#6.9>

ARIN – <http://www.arin.net/regserv/IPStats.html>

RIPE NCC – <ftp://ftp.ripe.net/ripe/docs/ripe-211.txt>

**Figure 10 – URLs to the RIR's Default Allocation Sizes**

## Negative Impact of Net Police Filters

The major downside, if not showstopper, for Net Police Filters is their impact on legitimate multihoming. This is especially felt in developing parts of the Internet where new service providers have received their first allocation from the regional registry, and have a basic requirement to be multihomed as part of their business. Their /20 network block is all they have, and if they subdivide this, they fall foul of the Net Police filters which have been put in place. It is for this reason that many ISP operators outside of the US frown upon the concept – many hours have been wasted trying to make multihoming work in the face of Net Police filters by an ISP in the US.

As for all multihoming scenarios, the ISP should always announce their network block. This will ensure connectivity to the Internet. However, announcing two /21s, the first step to basic inbound loadsharing of their traffic, will be impacted by their upstreams implementing RIR minimum allocation size filters. Encountering any situation like this will need cooperation of upstream providers, or worst case, considering moving custom elsewhere.

## Creating your own Net Police Filter

ISPs who wish to create their own Net Police Filter are strongly encouraged to do the following:

- Consider the impact. Shutting the door doesn't make the storm go away – it shuts everything out, good and bad.
- Accurate List. Consult each of the RIR's published CIDR Blocks for their default allocations. Ensure you have an up to date list and create a process to validate your Filter with future updates of the RIR's list.
- Consult with Colleagues. Consult with your peers on the list. If possible, compare what others have done with what you wish to accomplish.
- Publish Policy. A Web page with the filter policy is extremely helpful. Refer to the NANOG list of filter policies as an example (<http://www.nanog.org/filter.html>). And put a comment in your AS-Object.
- Keep it up to date! This is the most important part, and often the hardest to do. Be sensitive to complaints, especially about reachablity from your customers, or their customers – without them, you have no business.

The following is just one example of a Net Police Filter. ISPs can use this as a starting point, but are strongly encouraged to validate that everything in the list will work in their environment.

---

<sup>15</sup> Depending on the number of routes advertised, there might be additional load on the BGP process as it drops all these unwanted networks. Also, if BGP soft-reconfiguration is turned on, each of the dropped networks would still get saved.

```

!
!! RIPE
ip prefix-list FILTER permit 62.0.0.0/8 ge 9 le 20
ip prefix-list FILTER permit 80.0.0.0/7 ge 9 le 20
ip prefix-list FILTER permit 193.0.0.0/8 ge 9 le 20
ip prefix-list FILTER permit 194.0.0.0/7 ge 9 le 20
ip prefix-list FILTER permit 212.0.0.0/7 ge 9 le 20
ip prefix-list FILTER permit 217.0.0.0/8 ge 9 le 20
!
!! APNIC
ip prefix-list FILTER permit 61.0.0.0/8 ge 9 le 20
ip prefix-list FILTER permit 202.0.0.0/7 ge 9 le 20
ip prefix-list FILTER permit 210.0.0.0/7 ge 9 le 20
ip prefix-list FILTER permit 218.0.0.0/8 ge 9 le 20
!
!! ARIN
ip prefix-list FILTER permit 24.0.0.0/8 ge 9 le 20
ip prefix-list FILTER permit 63.0.0.0/8 ge 9 le 20
ip prefix-list FILTER permit 64.0.0.0/6 ge 9 le 20
ip prefix-list FILTER permit 199.0.0.0/8 ge 9 le 20
ip prefix-list FILTER permit 200.0.0.0/8 ge 9 le 20
ip prefix-list FILTER permit 204.0.0.0/6 ge 9 le 20
ip prefix-list FILTER permit 208.0.0.0/7 ge 9 le 20
ip prefix-list FILTER permit 216.0.0.0/8 ge 9 le 20
!
!! General - Filter anything greater than /24
ip prefix-list FILTER permit 0.0.0.0/0 ge 25
!
!! Other filters...

```

Notice that announcements between /9 and /20 are permitted. No registry will allocate their entire /8 block to one provider (at least not so far), so a /8 announcement shouldn't be seen from these blocks. Likewise, as a /20 is the minimum allocation by the registries, nothing smaller should be seen. Some ISPs argue that the above ranges should be /9 thru /21 or /22 to allow multihoming to be implementable for the smaller ISPs – providers who are implementing these filters should consider what they reasonably want to permit in their network.

## Packet Filtering<sup>16</sup>

Denial of service, spoofing, and other forms of attacks are on the increase on the Internet. Many of these attacks can be thwarted through the judicious use of ingress (packet originating from your network) and egress (packets arriving from the Internet) packet filtering. Packet filtering is where the router/switch inspects the packet's source and/or destination information and either pass or drop a packet based on a rule set. IP addresses and port numbers are the most common information used in the rule set. Cisco has several ways of inspecting and dropping packets:

- **Access Control List (ACLs)** – Access List is used to match on the source/destination IP address, protocol, or TOS to drop a packet.
- **Black Hole Filtering (Forwarding to Null0)** – The forward table routes the packet based on destination IP address to the adjacency *Null0* – dropping the packet.
- **Unicast RPF** – An MTRIE<sup>17</sup> look-up into the Forwarding Information Base (FIB) based on the source IP address. If the source IP address does not match the information in the forward table, then the packet is dropped.
- **Committed Access Rate (CAR)** – When the committed rate is set to zero, all packets are then dropped. CAR can match packets based on ACLs or MTRIE look-ups into the FIB's QOS-ID field.

Each of these packet-dropping techniques has various packet per second (PPS) performance impacts. Like any security tool, risks need to be weighted against the benefits.

<sup>16</sup> CAVEAT: This section covers the simple case single homed downstream customers. More thought is required when applying packet filtering for ISPs who are multihomed. For example, when the customer ISP's link to your network goes down, you will see packets from that network coming from "the Internet". Also be aware that a multihomed customer may require special routing to implement loadsharing – in that case you will again see that ISP's traffic coming from "the Internet".

<sup>17</sup> An MTRIE is a storage mechanism made up of nodes and leaves and used in CEF.

**NOTE: There could be a performance impact on the forwarding speed of the router when many filters are applied. Cisco's newer switching technologies help minimise the performance impact. TurboACLs (compiled access-lists) are available from 12.0(6)S and give superior performance for access-lists more than 10 entries long. Due care and consideration should be taken whenever the access list starts getting beyond 50 entries. However, it should be stated that trading a few microseconds of IP forwarding speed for the safety of minimising the impact of denial of service attacks may prove to be worth while.**

## Access Control List – General Sequential Based ACLs

Access Lists (ACLs) are the first option for a lot of organizations looking to enforce policy and provide a layer of security. At the same time ACLs have the greatest scaling, performance, and security risk. Generally, traditional ACLs are sequentially processed (see Figure 11). Any router processing a packet through an ACL will look for a match on a list of ACL entries, each line of the ACL would need to be checked until there is a match.<sup>18</sup> If the ACL were an extended version then the IP payload, ports and application would also need to be checked sequentially (see Figure 12). Since many policy and security filters would be an ACL with a lot of deny statements (see Config Example 1), a very long ACL increases the packet latency and consumes more CPU cycles.

```
access-list 25 deny 165.21.10.10
access-list 25 deny 171.68 34.1
access-list 25 deny 192.34.5.10
.
. many deny entries
.
access-list 25 deny 141.43.10.100
access-list 25 permit all
```

Config Example 1 – Standard Access List with lots of deny statements

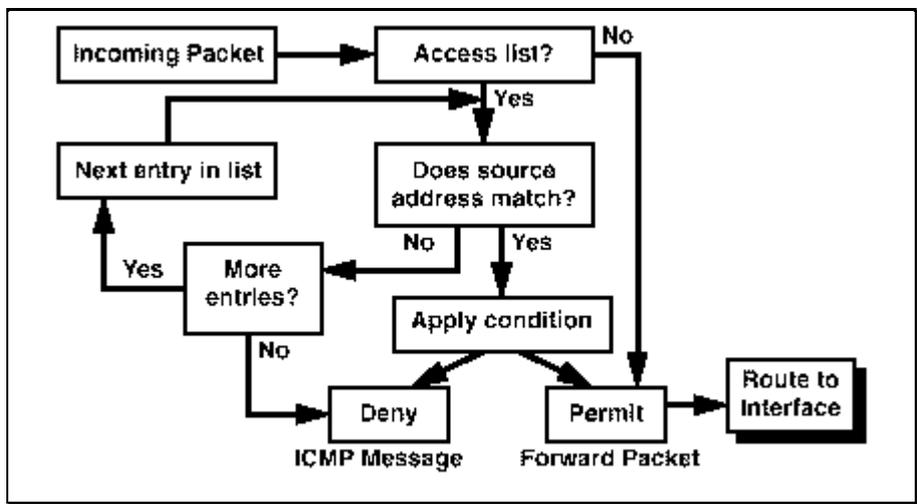


Figure 11 – Standard Access List – Incoming Packets

As the length of the policy filter increases, the burden on the processing power of the router reaches a point where it's at 99% CPU utilization. Added to this is the 100% to 200% growth rate of the Internet. This means that as the PPS load on the router increases from the natural growth of the network – the CPU load and packet latency from ACL processing will also increase, affecting the overall end-to-end bandwidth performance. To keep the ACL a relevant tool on the Internet improvements need to be added to allow ACLs to scale. Faster processors distributed processing, dedicated ASIC/TCAMs,

<sup>18</sup> The general principle of ACLs and the processing involved is an issue for all routers – not just Cisco routers. Cisco has tools like NetFlow and Optimum switching which increases the performance of the ACL checks – yet even this has its limits.

and special flow based switching technologies have been added to various Cisco products to allow ACLs to continue to keep pace with the Internet's growth. Yet, while keeping pace with the growth of the Internet, it is understood that sequential ACLs have limitations. One-way around these limitations is to use a new way to process packets through an ACL – using Turbo ACLs.

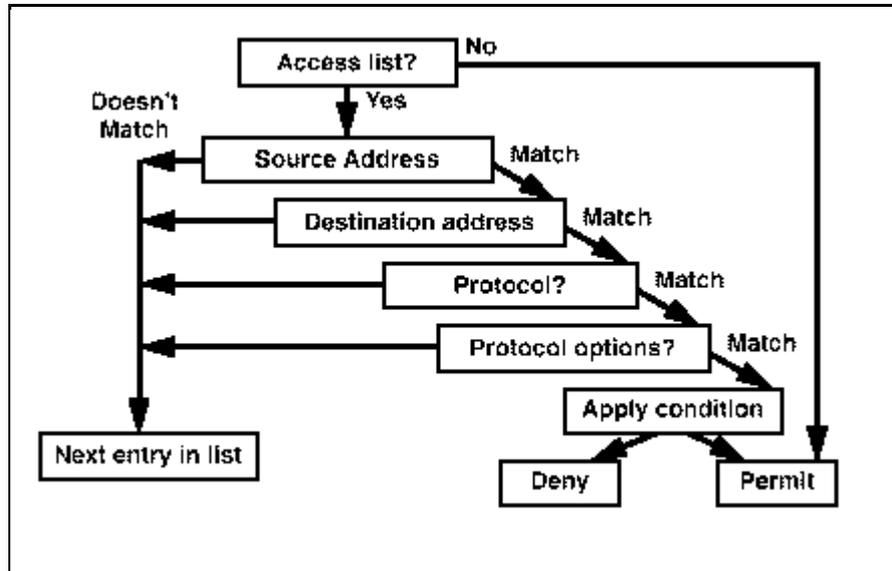


Figure 12 – Extended ACLs– Logical Flow for Every Packet

## Access Control Lists – Turbo ACLs

Turbo ACLs use a technique that takes a standard or extended ACL, creates a set of data tables and compiles them for runtime processing. For this reason, Turbo ACLs are also referred to as *compiled ACLs*. Turbo ACLs do not change the first match wins characteristics of all ACLs. What they will do is reduce the number of CPU operations to find a match – allowing for larger ACLs to be used without an increase in packet latency. This provides ISPs with a tool to have large ACLs without a significant performance impact on the router.

As seen in Figure 13, when Turbo ACLs are activated in a router, it will take the standard ACL input, create tables based on the ACLs entries, and compile the tables to allow an arrayed match. **The result is that a match will be achieved in 5 steps – no matter the size of the ACL.** This also means that Turbo ACL's advantages only become clear when the ACL is longer than five entries. So ACLs with 3 or 5 line would out perform Turbo ACLs, but Turbo ACLs with 300 to 500 lines would out perform sequentially searched ACLs. Figure 14 show the result of one study on the performance difference between sequential ACLs and Turbo ACLs. As the length of the ACL increases, the time it takes a sequential ACL to match increases (assuming last line match). Turbo ACLs provide consistency throughout the length of the ACL. The limit in the length of a Turbo ACL has more to do with the hardware performance envelope – memory, TCAM size, ASIC size, buffering, and CPU cycles are all factors that would limit the maximum size of a Turbo ACL.

### *Turbo ACLs Configuration Details and References*

To activate Turbo ACLs in a Cisco router, the following global command is used:

```
access-list compiled
```

This will apply Turbo ACLs to **all** ACLs on the router – no matter their size. Turbo ACLs were introduced in IOS version 12.0(6)S. Key word searches on CCO or the following URL will provide configuration information on Turbo ACLs:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/120s6/turboacl.htm>

Wednesday, June 06, 2001

Full details of how Turbo ACLs work were given by the author, Andrew McRae ([amcrae@cisco.com](mailto:amcrae@cisco.com)), in the paper *High Speed Packet Classification*, presented at AUUG national conference September 1999.

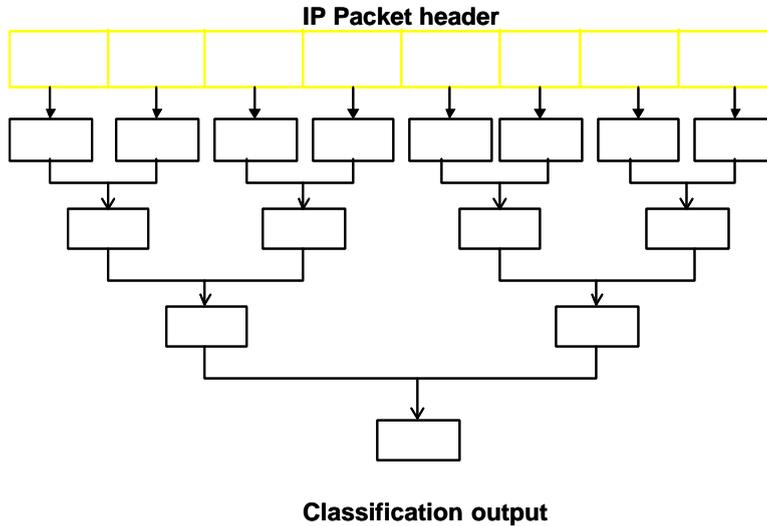


Figure 13 – Turbo ACLs – Processing takes five steps no matter what the size

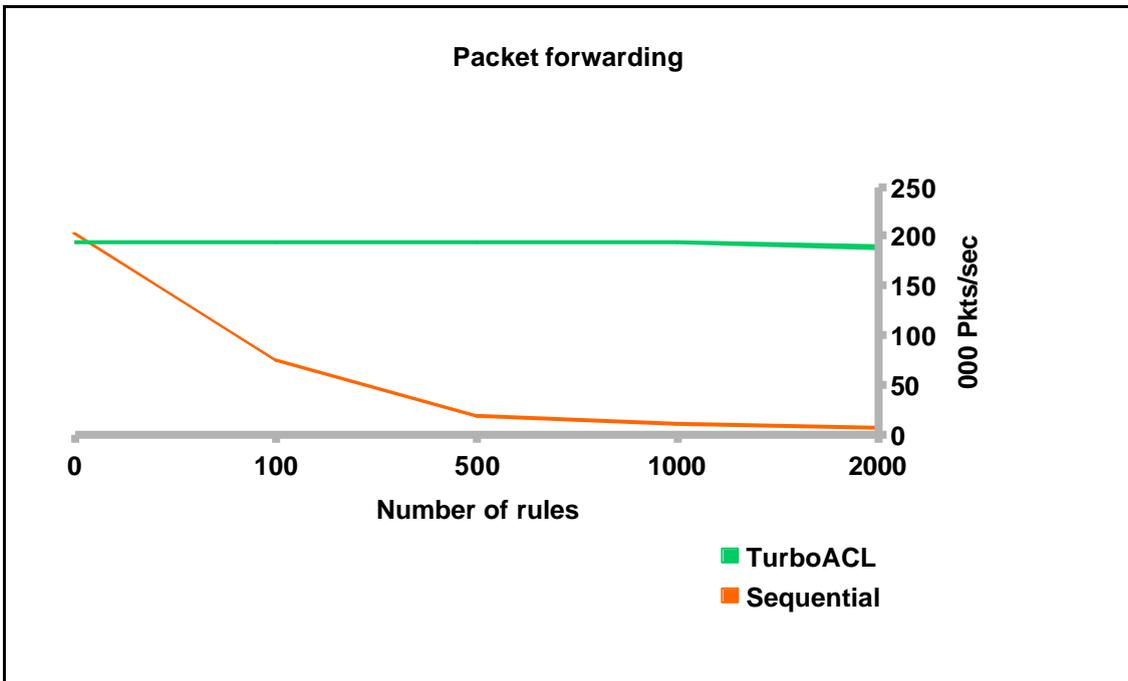


Figure 14 – Turbo ACL's Performance Consistency

## Using ACLs for Egress Packet Filtering

Egress packet filtering ensures that the packets that you send out to other networks (ISPs or customers) are valid. It can be applied either on the gateway (upstream – peering routers) or on the customer edge. By filtering packets on your routers

that connect your network to the Internet (Figure 15), you can permit only packets with valid source IP addresses to leave your network and get into the Internet. For example, if your network consists of network 165.21.0.0, and your router connects to your ISP using a serial 0/1 interface, you can apply the access-list as follows:

```
access-list 110 permit ip 165.21.0.0 0.0.255.255 any
access-list 110 deny ip any any log
!
interface serial 0/1
description Upstream Connection to ISP A
ip access-group 110 out
!
```

The last line of the access-list determines if there is any traffic with an invalid source address entering the Internet. If there are any matches, they will be logged. It is not crucial to have this line, but it will help locate the source and extent of the possible attacks.

Some ISPs consider the option of egress filtering on the customer edge of their network (see Figure 16). In this case, the objective is to ensure that packets the bound for the customer do not have a source IP address within the range of addresses allocated by the ISP to the customer. For example, a packet with a destination address of 165.21.10.1 should not have a source address of 165.21.10.1. These are spoofed packets and should be dropped. The following is an example of an Egress Packet Filter ACL on the customer's interface:

```
access-list 121 deny ip 165.21.0.0 0.0.255.255 any
access-list 121 permit ip any any log
!
interface serial 1/1/1.3
description T1 Link to Customer Acme Computer systems
ip access-group 121 out
!
```

The key factor of that limits this technique is the magnitude of the implementation. ISPs with 10000 business customers do not see the benefits gained by managing 10000 ACLs. Especially when the customer is encouraged to implement ingress filtering on their gateway router.

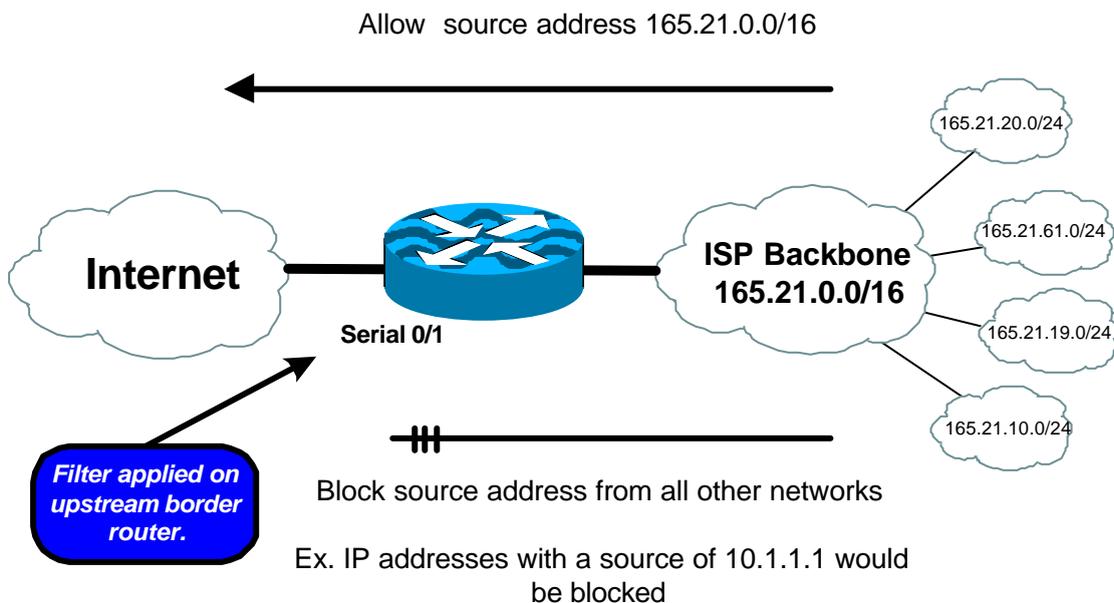


Figure 15 – Egress Packet Filtering on the Upstream Gateway Router

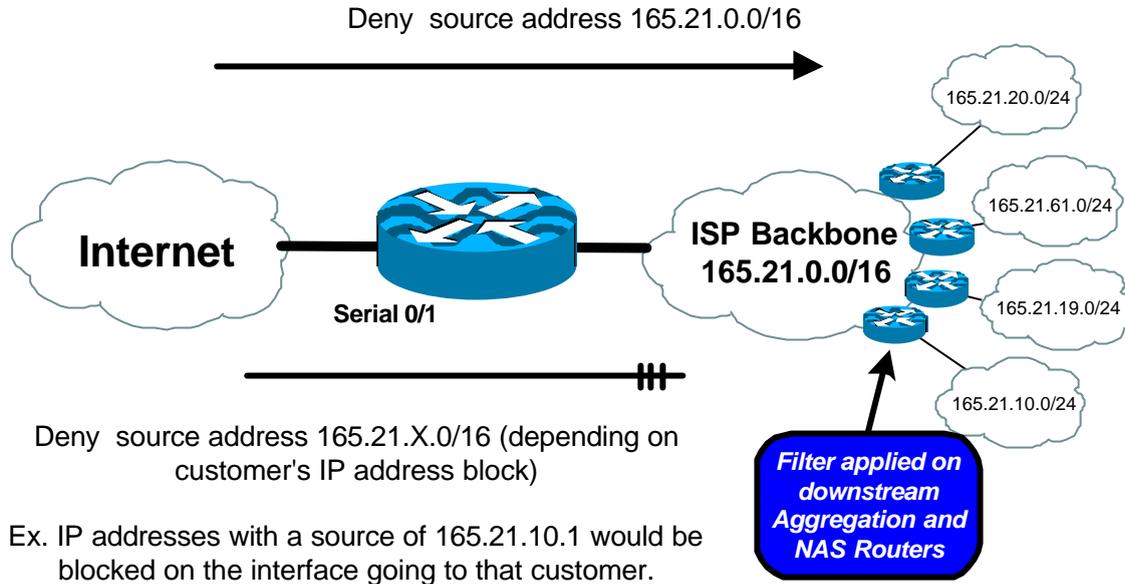


Figure 16 – Egress Packet Filtering on the Customer Edge

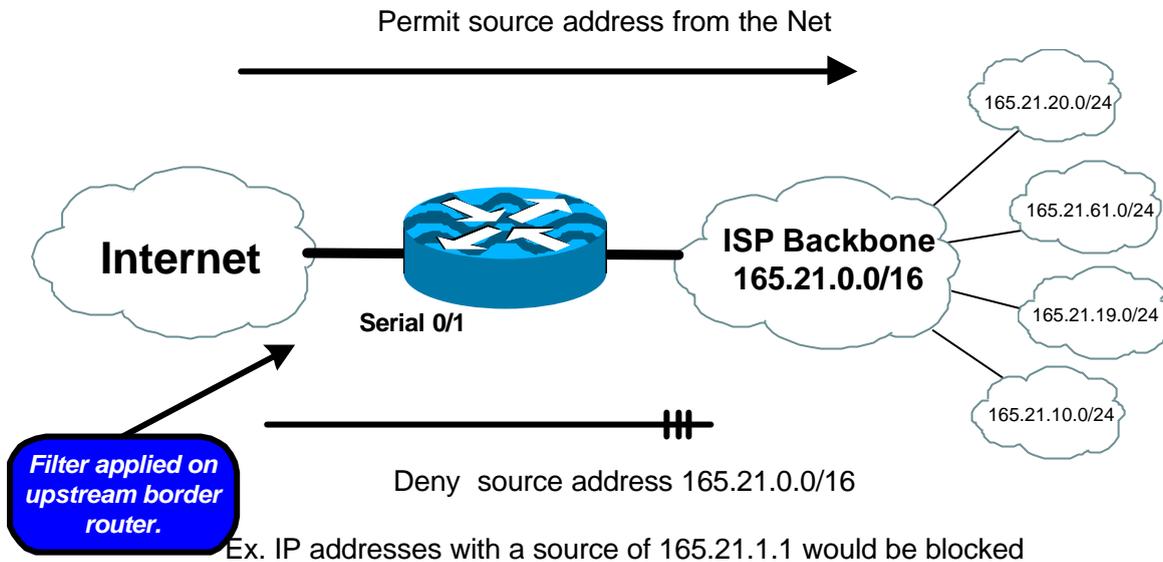
## Using ACLs for Ingress Packet Filtering

Ingress packet filtering validates the packets from the outside world (ISPs and Customers) into and across your network. For an ISP, the *outside world* is any place from outside the ISP's control. Obviously, packets from other ISPs are from the *outside world* and are not to be implicitly trusted. It also means packets from an ISP's customers are from the *outside world*. Just because a network is a customer of an ISP does not make it a trusted network.

For ISPs who provide service to end networks, we highly recommend the validation of incoming packets from your clients. This can be accomplished by the use of inbound packet filters on your border routers. Figure 17 is an example where an ISP has one gateway to their upstream network. Customers with a network number of 165.21.10.0/24, should not see any packets coming from the Internet with 165.21.10.1 as the source address. These packets are attempts at spoofing and should be dropped. The following example shows a sample filter for network 165.21.0.0 with filters for private and rogue routes:

```
access-list 111 deny ip host 0.0.0.0 any log
access-list 111 deny ip 127.0.0.0 0.255.255.255 any log
access-list 111 deny ip 10.0.0.0 0.255.255.255 any log
access-list 111 deny ip 172.16.0.0 0.15.255.255 any log
access-list 111 deny ip 192.168.0.0 0.0.255.255 any log
access-list 111 deny ip 165.21.0.0 0.0.255.255 any log
access-list 111 permit ip any any
!
interface serial 1/0
 ip access-group 111 in
!
```

All the “anti spoof”, private address, and rogue filters have *log any* matches. It is not crucial to have this line, but it will help locate the source and extent of the possible probes or attacks. Figure 17 provides an illustration of ingress filtering on the upstream gateway router.



**Figure 17 – Ingress Packet Filtering on the Upstream Gateway Router**

While the example just given will work for all customers connecting to the Internet and many small to medium sized ISPs, it is in addition to what is recommended in the IETF's Best Common Practice (BCP) document – **BCP 38/ RFC 2827 Title: Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing Author(s): P. Ferguson, D. Senie.** BCP 38 recommends ingress filtering on the ISP↔Customer edge of the network. This filtering is checking to ensure that a source address of a packet matches the IP address block allocated to the customer. For example, a customer who has been allocated an IP address block of 165.21.10.0/24 should not send any packets with a source address of 192.168.1.1. The following is the example of an Ingress Packet Filter ACL for the customer:

```
access-list 122 permit ip 165.21.0.0 0.0.255.255 any
access-list 122 deny ip any any log
!
interface serial 1/1/1.3
description T1 Link to Customer Acme Computer systems
ip access-group 121 in
```

Notice that the ACL is one line long and only permits the authorised packets. The second line of the ACL – *deny any any log* – is optional. The one or two line ACL is easier to implement. The second deny any line is optional – but does capture the packets that are denied and logs them in the router's syslog. The objective is to log packets that are in violation of the access-list, allowing for the detection of compromised systems. The limitation of this technique is when there are 1000s of leased line customers; how does one manage 1000s of ACLs. The answer is that you do not. You use Unicast RPF to do the ingress packet filtering on the ISP ↔ Customer edge.

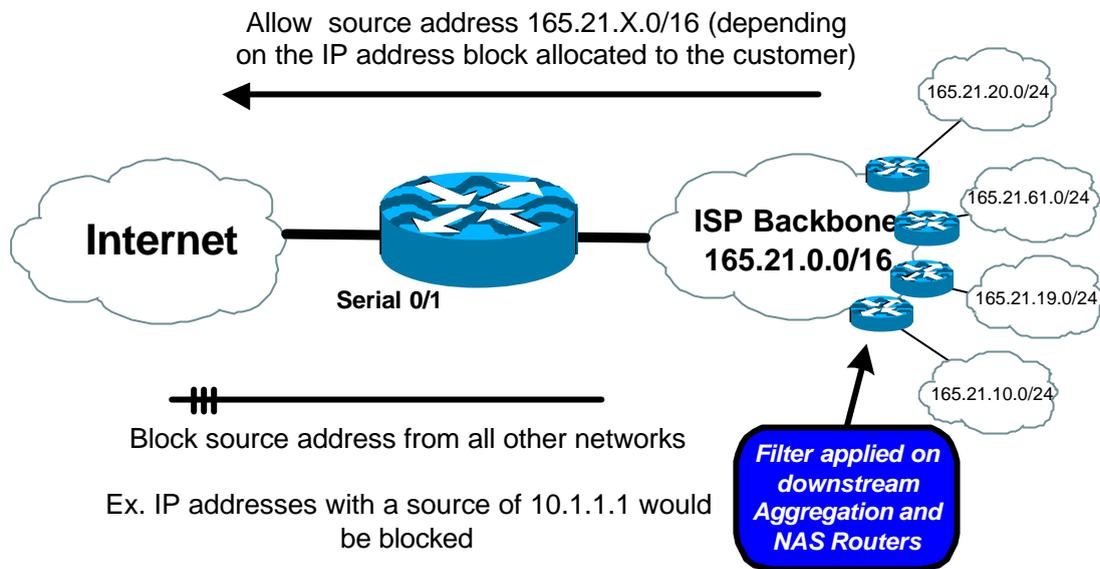


Figure 18 – Ingress Packet Filtering on the ISP <- Customer Edge (IETF BCP Recommendation)

## Black Hole Routing as a Packet Filter (Forwarding to Null0)

Another way of implementing destination based packet filtering on a router is creating a specific list of static host routes and pointing them to the pseudo interface *Null0*. This technique is commonly referred to as **black hole routing**. Null0 is a pseudo-interface, which functions similarly to the null devices available on most operating systems. This interface is always up and can never forward or receive traffic. While Null0 is a pseudo interface, within CEF, it is not a *valid* interface. Hence, whenever a route is pointed to Null0, it will be dropped via CEF's and DCEF's forwarding process with no processor overhead.

The null interface provides an alternative method of filtering traffic. You can avoid the overhead involved with using access lists by directing undesired network traffic to the null interface. The following example configures a null interface for IP route 127.0.0.0/16 and the specific host 171.68.10.1 (subnet mask 255.255.255.255):

```
interface Null0
  no icmp unreachable
  !
ip route 127.0.0.0 255.0.0.0 null 0
ip route 171.68.10.1 255.255.255.255 null 0
```

The *no icmp unreachable* command is used to prevent unnecessary replies whenever traffic is passed to the Null0 interface. Figure 19 give a graphic example of how this black list filtering technique works.

By using the black hole routing technique to implement black list filtering (no pun intended), the strength of the router is utilized to drop packets bound for forbidden sites. A router's primary function is to forward packets – not filter packets. The black hole routing technique uses the packet forwarding power to drop all packets bound for sites on the black list.

Yet, there are two main drawbacks to this technique. First, access to all services at a given site must be restricted. The attraction of Extended ACLs is the fine granularity given to filter at the application level. Black hole routing does not have this granularity. It filters access for all packets bound to a specific host or subnet. Second, it is hard to bypass the black hole routing technique. Any organisation that wishes to by-pass the black list must actually find a way to by-pass the filtering router's forward table. Compensating for either limitation is no trivial task, yet options are available for both.

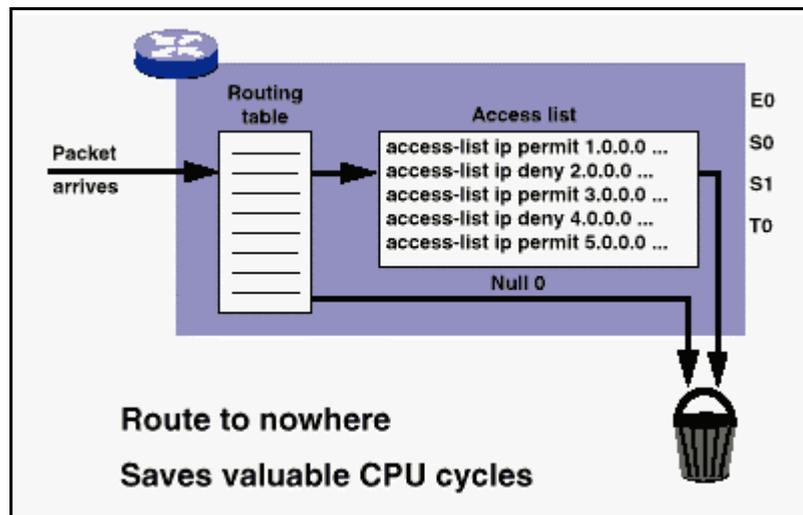


Figure 19 – Using static host routes to null 0 for black list filtering

## Unicast RPF – (Reverse Path Forwarding)<sup>19</sup>

Unicast Reverse Path Forwarding (Unicast RPF) is a feature used to prevent problems caused by packets with malformed or forged IP source addresses passing through a router. Unicast RPF will help prevent Denial of Service (DoS) attacks based on Source IP address spoofing – including the infamous SMURF and other denial of service attacks. Unicast RPF requires the Cisco Express Forwarding (CEF) mechanism to be enabled. The effect of Unicast RPF is that it stops forged packets at the ISP’s PoP (leased line and dial-up). This protects the ISP’s network, its customers’ networks, as well as the rest of the Internet (good Network Citizen operation).

When uRPF is enabled on an interface, the router will verify that all packets received from that interface have a source address which is reachable via that same interface. This “look backwards” ability is only available when CEF (Cisco Express Forwarding) is enabled, since the lookup relies on the presence of the Forwarding Information Base (FIB). Unicast RPF ensures that there is a *reverse path route* to the input interface of the packet. If there is a *reverse path route*, the packet is forwarded as normal. If there is not a reverse path route, the packet is dropped.

When a packet is received by a router’s interface with Unicast RPF and ACLs configured, the following occurs:

1. Check input ACLs configured on the inbound interface.
2. Unicast RPF validates the packet has a return path through the inbound interface by checking the CEF table (CEF or dCEF).
3. CEF table (FIB) lookup is carried out for packet forwarding.
4. Output ACLs are checked on outbound interface.
5. Packet is forwarded.

For example, if a customer sent a packet with the source address of 210.210.1.1 from interface FDDI 2/0/0, RPF will check the FIB to see if 210.210.1.1 has a path to FDDI 2/0/0. If there is a matching path, the packet will be forward. If there is no matching path, the packet will be dropped.

Unicast RPF’s advantage when used for IP address spoof prevention is that it dynamically adapts to changes in the dynamic routing tables, including static routes. Unicast RPF has minimal CPU overhead and operates a few percent less than CEF/optimum/fast switching rates. Unicast RPF has far lower performance impact as an anti-spoofing tool compared with the access-list approach.

<sup>19</sup> Original documentation on Unicast RPF was by Bruce R. Babcock [[bbabcock@cisco.com](mailto:bbabcock@cisco.com)]

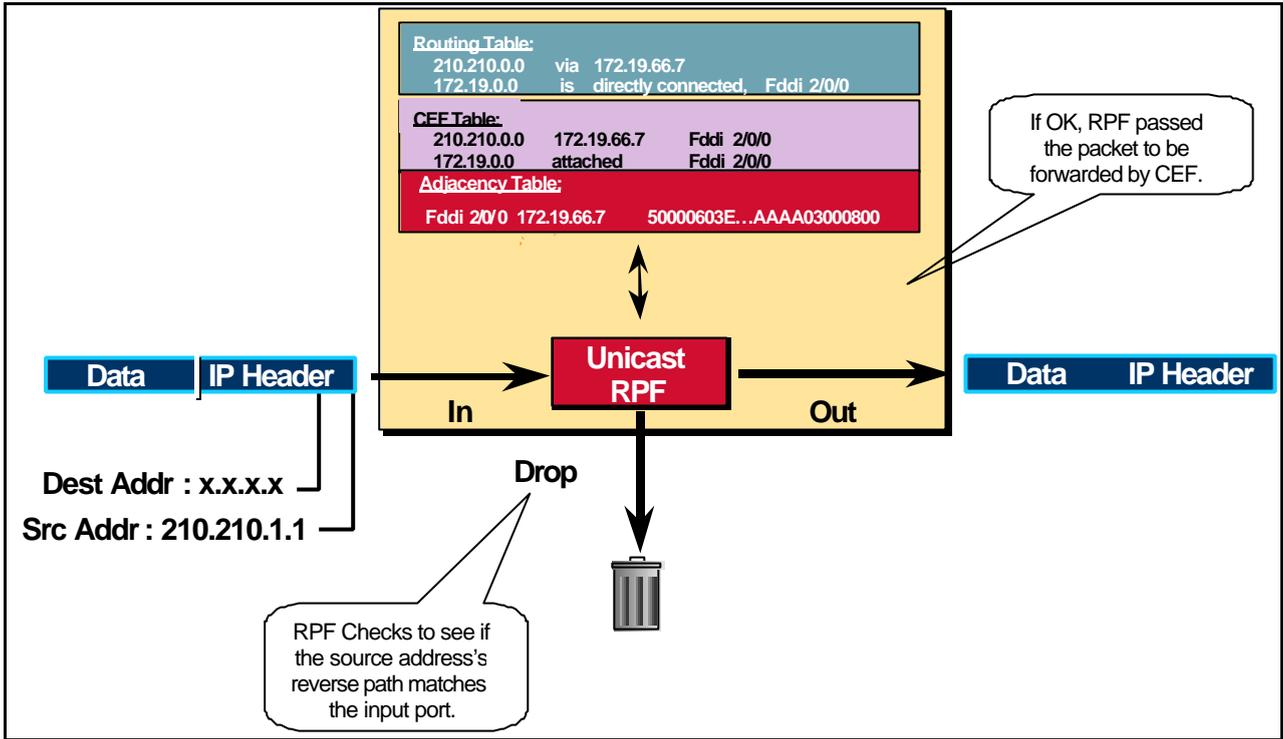


Figure 20 – Unicast RPF validating IP source addresses

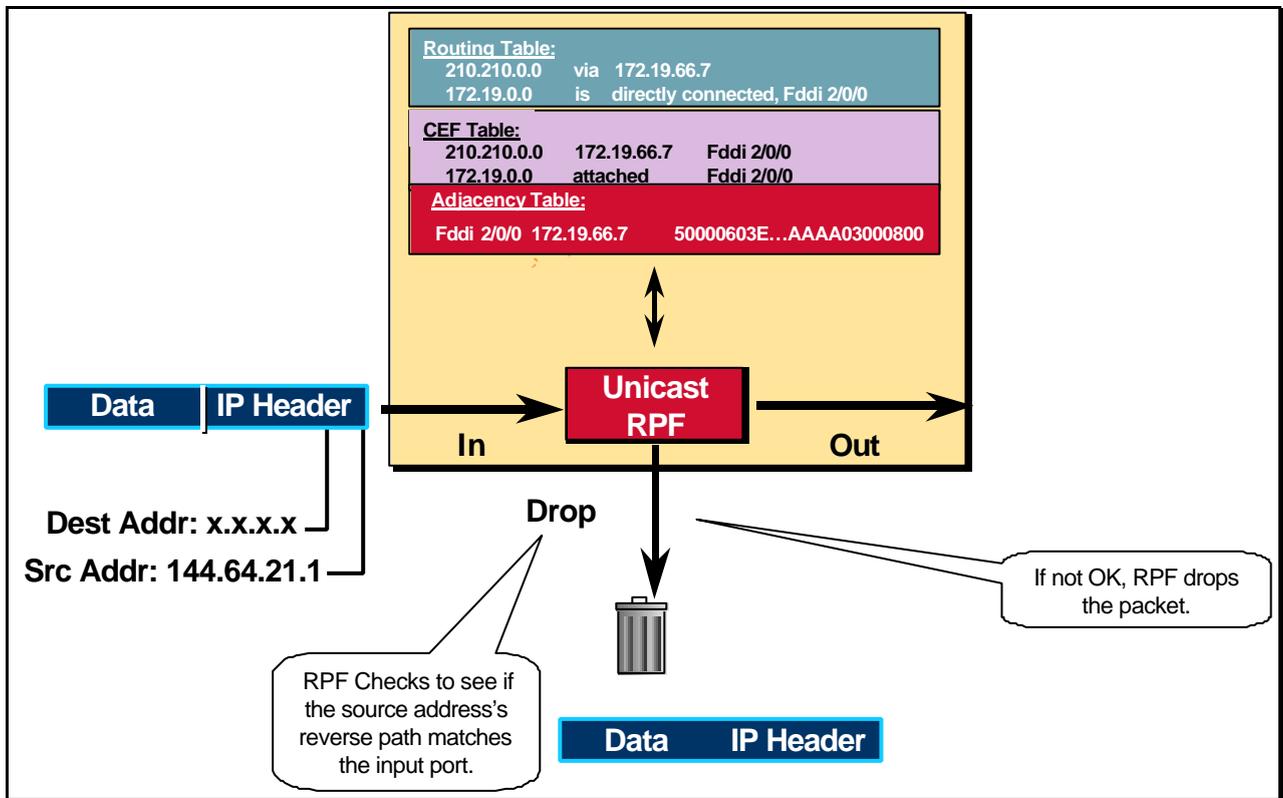


Figure 21 – Unicast RPF dropping packets which fail verification

Unicast RPF also requires less operational maintenance than traditional approaches which use IP access or extended-access lists. It can be added to the customer's default configuration on the ISP's router (remember this will only work if the router has CEF configured).

```
! Configuration template for customer interfaces
description [enter description of interface]
no ip redirect
no ip direct broadcast
no ip proxy-arp
ip verify unicast reverse-path
bandwidth [bandwidth in kbps]
```

Unicast RPF is compatible with CEF's per-packet and per-destination load sharing.

Unicast RPF was first supported in 11.1(17)CC CEF images on the RSP7000, 7200 and 7500 platforms. It is not supported in any 11.2 or 11.3 images. Unicast RPF is included in 12.0 on platforms that support CEF. This includes the AS5800, uBR, and 6400 – which means that Unicast RPF can be configured on the PSTN/ISDN dial-up interfaces on that platform.

### ***RPF Configuration Details (as of IOS Version 12.0(10)S1 )***

To use unicast RPF, enable 'CEF switching' or 'CEF distributed switching' in the router. There is no need to configure the input interface for CEF switching. This is because unicast RPF has been implemented as a search through the FIB using the source IP address. As long as CEF is running on the router, individual interfaces can be configured with other switching modes. RPF is an input side function that is enabled on an interface or sub-interface supporting any type of encapsulation and operates on IP packets received by the router. **It is very important for CEF to be turned on globally in the router – RPF will not work without CEF.**

Configure RPF on the interface using the following interface command syntax:

```
[no] ip verify unicast reverse-path [<ACL>]
```

For example on a leased line aggregation router:

```
ip cef
! or "ip cef distributed" for an RSP+VIP based box
!
interface serial 5/0/0
 ip verify unicast reverse-path
```

For another example, the AS5800 supports CEF in the IOS 12.0. The *interface Group-Async* command makes it even easier to apply Unicast RPF on all the dial-up ports:

```
ip cef
!
interface Group-Async1
 ip verify unicast reverse-path
```

Use the command *show cef interface [interface]* to verify RPF is operational:

```
Excalabur#sh cef inter serial 2/0/0
Serial2/0/0 is up (if_number 8)
Internet address is 169.223.10.2/30
ICMP redirects are never sent
Per packet loadbalancing is disabled
IP unicast RPF check is enabled
Inbound access list is not set
Outbound access list is not set
Interface is marked as point to point interface
Packets switched to this interface on linecard are dropped to next slow path
Hardware idb is Serial2/0/0
```

Unicast RPF  
Enabled

Wednesday, June 06, 2001

```
Fast switching type 4, interface type 6
IP Distributed CEF switching enabled
IP LES Feature Fast switching turbo vector
IP Feature CEF switching turbo vector
Input fast flags 0x40, Output fast flags 0x0, ifindex 7(7)
Slot 2 Slot unit 0 VC -1
Transmit limit accumulator 0x48001A02 (0x48001A02)
IP MTU 1500
```

Use the *show ip interface [interface]* to find specific drops on a interface ( as of IOS Version 12.0(10)S1 ):

```
Excalabur#sh ip inter fastEthernet 4/1/0
FastEthernet4/1/0 is up, line protocol is up
.
.
Unicast RPF ACL 100
55 unicast RPF drops
0 unicast RPF suppressed drops
```

A counter is maintained to count the number of discards due to RPF. The value of the counter will be displayed as part of the output from the command:

```
show ip traffic
```

The RPF drop counter is included in the *IP Statistics* section:

```
ISP-LAB-7505-3#sh ip traffic
IP statistics:
Rcvd: 1471590 total, 887368 local destination
      0 format errors, 0 checksum errors, 301274 bad hop count
      0 unknown protocol, 0 not a gateway
      0 security failures, 0 bad options, 0 with options
Opts: 0 end, 0 nop, 0 basic security, 0 loose source route
      0 timestamp, 0 extended security, 0 record route
      0 stream ID, 0 strict source route, 0 alert, 0 other
Frag: 0 reassembled, 0 timeouts, 0 couldn't reassemble
      0 fragmented, 0 couldn't fragment
Bcast: 205233 received, 0 sent
Mcast: 463292 received, 462118 sent
Sent: 990158 generated, 282938 forwarded
Drop: 3 encapsulation failed, 0 unresolved, 0 no adjacency
      0 no route, 0 unicast RPF, 0 forced drop
      ^^^^^^^^^^^^^^^^^^^
```

Figure 22 – Unicast RPF Drop Counter

## *ACL Option (added in IOS Version 12.0(10)S1)<sup>20</sup>*

The optional ACL parameter to the command can be used to control the exact behavior when the received frame fails the source IP address check.

The ACL can be either a standard or an extended IP access list<sup>21</sup>:

```
<1-99>      IP standard access list
<100-199>   IP extended access list
<1300-1999> IP standard access list (expanded range)
<2000-2699> IP extended access list (expanded range)
```

<sup>20</sup> Section taken from the release notes of CSCdp76668 by the development engineer who coded this function - Neil Jarvis [njarvis@cisco.com].

<sup>21</sup> There was a bug in 12.0(10)S – the help option in IOS will only display the standard and extended standard access lists as options. Standard/Expanded and Extended/Expanded ACLs both work.

If an ACL is specified, then when (and only when) a packet fails a unicast RPF check, the ACL is checked to see if the packet should be dropped (using a deny ACL) or forwarded (using a permit ACL). In both cases the packet is counted as before. ACL logging (log & log-input), and match counts operate as normal.

Example:

```
ip cef distributed
!
interface ethernet 0/1/1
 ip address 192.168.200.1 255.255.255.0
 ip verify unicast reverse-path 197
!
access-list 197 permit ip 192.168.200.0 0.0.0.255 any log-input
```

Frames sourced from 192.168.200.10 arriving at ethernet 0/1/1 are forwarded (because of permit), ACL logged, and counted per-interface.

Frames sourced from 192.168.201.100 arriving at ethernet 0/1/1 are dropped (because of deny), ACL logged, counted per-interface, and counted globally.

Counting is seen per interface:

```
Router> show ip interface ethernet 0/1/1 | include RPF
Unicast RPF ACL 197
 1 unicast RPF drop
 1 unicast RPF suppressed drop
```

globally,

```
Router> show ip traffic | include RPF
 0 no route, 1 unicast RPF, 0 forced drop
```

and per ACL

```
Router> show access-lists
Extended IP access list 197
 permit ip 192.168.200.0 0.0.0.255 any log-input (100 match)
```

Unicast RPF ACL function has two primary functions. The first and obvious function is to allow for exceptions. Some networks may need to get through the Unicast RPF check – hence the ACL will allow bypass technique. The second function is to identify spoof packets. Unicast RPF will not send any notifications of which packets it is dropping. Counters will increment, hence the operator will be able to notice excessive Unicast RPF drops. Yet, the operator would question what is being dropped. An ACL can be used to determine if the drops are valid (spoofed source addresses) or in error (valid packets being dropped). In the following example, Unicast RPF will take each packet that fails the reverse patch forwarding check and apply it to ACL 171. ACL will still drop the packet, but will also log the packet in the ACL counters and the log file on the processor or VIP/Linecard.

```
interface ethernet 0/1/1
 ip address 192.168.200.1 255.255.255.0
 ip verify unicast reverse-path 171
!
access-list 171 deny icmp any any echo log-input
access-list 171 deny icmp any any echo-reply log-input
access-list 171 deny udp any any eq echo log-input
access-list 171 deny udp any any eq echo any log-input
access-list 171 deny tcp any any established log-input
access-list 171 deny tcp any any log-input
access-list 171 deny ip any any log-input
```

```
Excalabur#sh controllers vip 4 logging
show logging from Slot 4:
.
.
```

Wednesday, June 06, 2001

```
4d00h: %SEC-6-IPACCESSLOGNP: list 171 denied 0 20.1.1.1 -> 255.255.255.255, 1 packet
.
```

## *debug ip cef drops rpf*

RPF packets that are dropped can be captured with a *debug ip cef drops rpf <ACL>* statement (as of 12.0(4)S but not in 11.1CA or in 11.1CC).

**Warning!** Care must be taken with any use of the debug feature on a production router. The amount of debug information easily overwhelms the ability of the console and logging functions of a router. This is especially true when the router handling several tens of megabytes of DoS traffic.

The way this Unicast RPF debug tool is used depends on the whether the router is a VIP or no VIP platform. On Cisco 7500s with VIP card debug results do not leave the VIP. To see the results of a debug on a VIP, use an ACL on the *debug ip cef drops rpf* with the *log-input* function applied to the ACL. You can then use the *show controllers vip <number> logging* to see the results of the debug. Example 4 provides an example of *show controllers vip 1 logging* together with the *debug ip cef drops rpf 88* command.

```
Thundershild#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Thundershild(config)#
Thundershild(config)#access-list 88 permit 1.19.1.4 0.0.0.0 log
Thundershild(config)#exit
Thundershild#debug ip cef drops rpf 88
Thundershild#sh controller vip 1 logging

Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
  Console logging: level debugging, 59 messages logged
  Monitor logging: level debugging, 0 messages logged
  Buffer logging: level debugging, 65 messages logged

Log Buffer (8192 bytes):
smallest_local_pool_entries = 192, global particles = 618
highest_local_visible_bandwidth = 100000
.
.
2d16h: CEF-Drop: Packet from 1.19.1.4 via FastEthernet1/0/0 -- unicast rpf check
2d16h: CEF-Drop: Packet from 1.19.1.4 via FastEthernet1/0/0 -- unicast rpf check
2d16h: CEF-Drop: Packet from 1.19.1.4 via FastEthernet1/0/0 -- unicast rpf check
2d16h: CEF-Drop: Packet from 1.19.1.4 via FastEthernet1/0/0 -- unicast rpf check
2d16h: CEF-Drop: Packet from 1.19.1.4 via FastEthernet1/0/0 -- unicast rpf check
.
.
Thundershild#no debug ip cef drops rpf 88
```

**Example 4 – Using the *show controller vip1 logging* on a Cisco 7500 VIP card**

## ***Routing Tables Requirements***

Unicast RPF needs accurate information in the CEF tables to work properly. The fundamental requirement for Unicast RPF to work is...

**...a valid and preferred path must exist in the forwarding table (FIB) that matches the source address to the input interface.**

This does not mean the router must have the entire Internet routing table. The amount of routing information needed in the CEF tables depend on where Unicast RPF is configured and what functions the router plays in the ISP's network. For example, a router that is a leased line aggregation router for customers only needs the information based on the static routes

redistributed into the IGP or iBGP (depending on which technique is used in the network). Unicast RPF would be configured on the customers' interfaces – hence the requirement for minimal routing information. In another scenario, a single homed ISP can place Unicast RPF on the gateway link to the Internet. The full Internet routing table would be required. This would help protect the ISP from external DoS attacks that use addresses that are not in the Internet routing table.

### ***Unicast RPF Exceptions***

There are some source IP addresses which should be allowed through the Unicast RPF filtering (see Equation 1). Unicast RPF will now allow packets with 0.0.0.0 source and 255.255.255.255 destination to pass so that BOOTP and DHCP function. This feature (CSCdk80591) was added from IOS release 12.0(3.05) (but is not in 11.1CC). Also, if the destination address is multicast, Unicast RPF will exempt those packets.

```
lookup source address in forwarding database
  if the source address is reachable via the source interface
    pass the packet
  else
    if the source is 0.0.0.0 and destination is a 255.255.255.255
      /* BOOTP and DHCP */
      pass the packet
    else if destination is multicast
      pass the packet
    else
      drop the packet
```

**Equation 1 – Unicast RPF Algorithm as of IOS 12.0(9)S**

## **Implementing Unicast RPF**

Unicast RPF's key implementation principles are:

- There must be a route in the FIB matching the prefix to the interface. This can be done via connected interface, static route, network statement (BGP, OSPF, RIPv2, etc), or from dynamic routing updates.
- Traffic from the interface must match the prefix for the interface.
- If there are multiple entries for the prefix in the route tables, the prefix local to the router implementing Unicast RPF must be preferred.

Given these three implementation principles, Unicast RPF becomes a tool that ISPs can use not only for their customers, but also for their downstream ISPs – even if the downstream ISP has other connections to the Internet.

## **Unicast RPF for Service Providers and ISPs**

An ISP or Service provider uses Unicast RPF to protect their network from their customers and (perhaps) the rest of the Internet. By policing their customer's connections with Unicast RPF, the ISP/Service Provider in turn helps to protect the rest of the Internet. The following are some common implementation techniques for Unicast RPF.

### ***Single Homed Lease Line Customers***

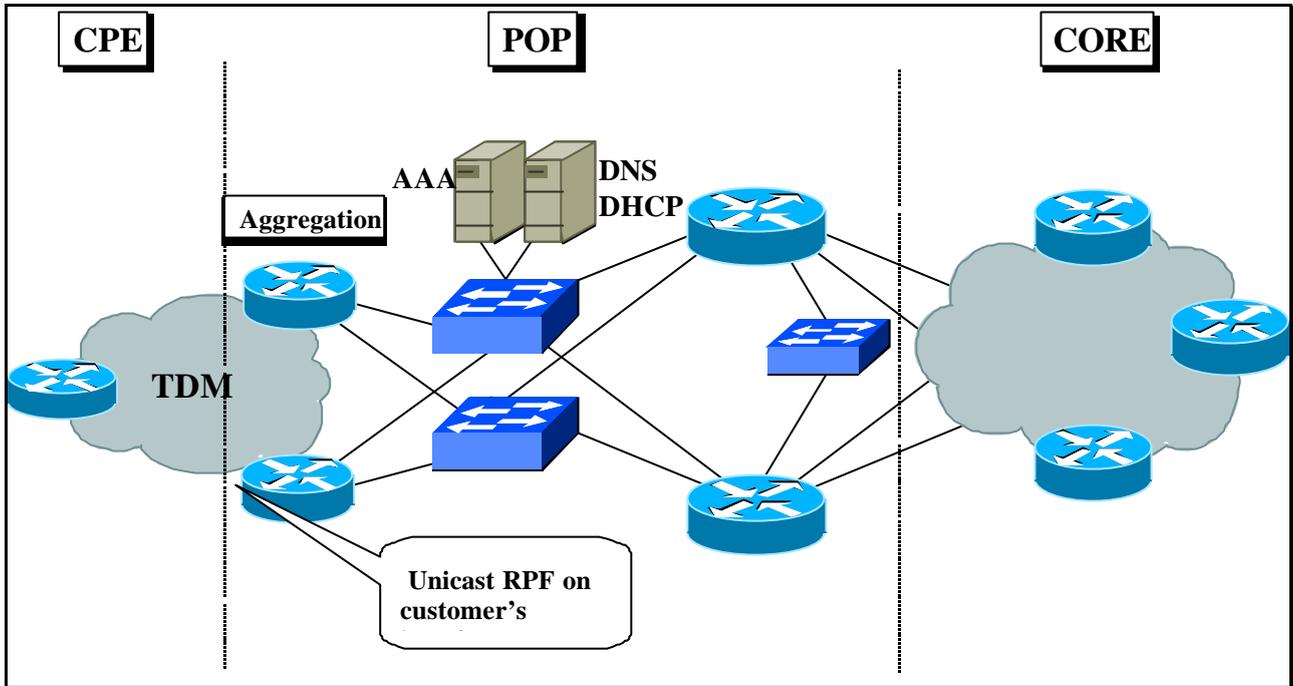
Unicast RPF is best used at the edge of the network for customer network terminations (see Figure 23) where leased line customer aggregation routers are connecting single homed customers<sup>22</sup>. In this topology, the customer aggregation routers don't require the full Internet Routing Table but only the routing prefixes assigned to the customer<sup>23</sup>. Hence, information configured or redistributed in the IGP or iBGP (depending on the way you add customer routes into your network) would be enough for Unicast RPF to do its job.

<sup>22</sup> For multihomed customers – see the section on Unicast RPF Limitations.

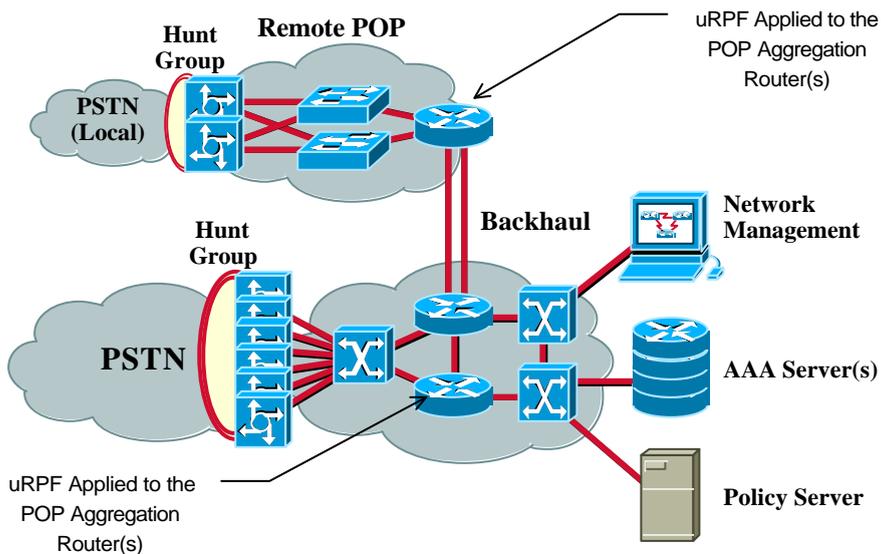
<sup>23</sup> The customer's assigned IP Block (i.e. routing prefixes) usually is inserted into the ISP's network in one of several ways. Any of these ways will work as the information is passed to the CEF tables.

**NAS Application – Applying Unicast RPF in PSTN/ISDN PoPs**

Unicast RPF is not limited to leased line connections. It works equally well on PSTN/ISDN/xDSL customer connections into the Internet. In fact, dial-up connections are reputed to be the greatest source of DoS attacks using forged IP addresses. As long as the Network Access Server (NAS) supports CEF, Unicast RPF will work.



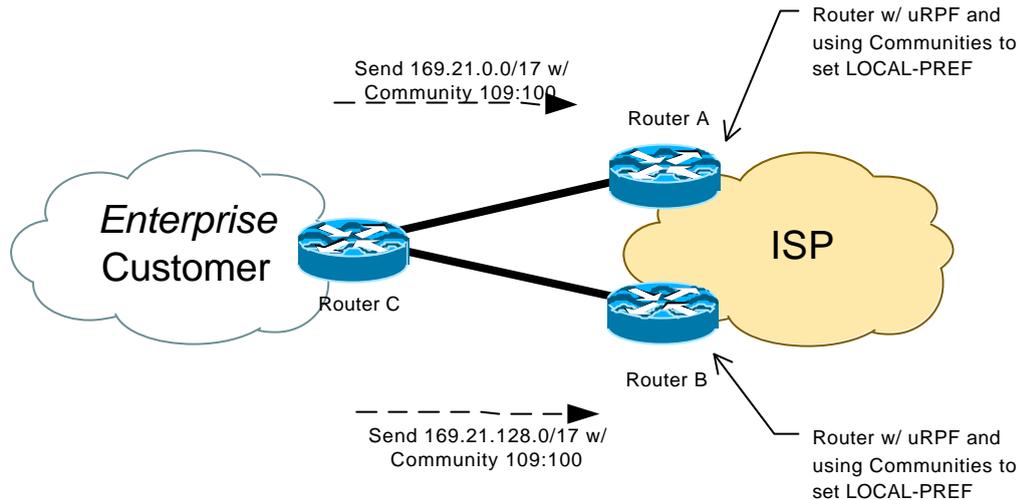
**Figure 23 – Unicast RPF applied to Leased Line Customer Connections**



**Figure 24 – Unicast RPF applied to PSTN/ISDN Customer Connections**

### *Multihomed Lease Line Customers (one ISP)*

With some careful thoughtful planning, Unicast RPF can be applied to customers of an ISP who are multihomed to them – even when there are multiple paths to the customer. For Unicast RPF to work only one path should be in the router’s forwarding table (the FIB in CEF switching) and that entry must point back out the interface unicast RPF is applied. Enterprise customers with multiple connections to the Internet will typically use BGP with a private AS number to connect to their upstream ISP. The ISP uses BGP *weights* to ensure that the route local to the router is the preferred path. BGP weights are local to the router and are easily configured on the BGP neighbour configuration.



**Figure 25 – Multihomed Lease Line Customer & Unicast RPF**

In this example, the *Enterprise* customer of the ISP is multihomed into two different routers. BGP is used with a private AS number assigned by the ISP. The *Enterprise’s* IP address block would be allocated from the ISP or from an IP registry. As the route is advertised into the ISP’s routers, an internal BGP weight is applied. This ensures that if there is a tie between two identical prefixes, the one heard directly from router C would be preferred on the local router and entered into the Forward Table (FIB).

In Example 5, the customer divided their allocated /16 address block into two /17s – allowing for some level of traffic engineering. The customer advertises the /16 and two /17s out both links (required to have unicast RPF work on the ISP’s ingress). Preference is communicated to the ISP via BGP Communities. The ISP uses the communities advertised from the customer to set the BGP Local-Preference on the /17 – making one preferred over the other.<sup>24</sup>

There are some basic restrictions to applying Unicast RPF to these multihomed customers:

- Customers should not be multihomed to the same router. This is common sense – it breaks the purpose of building a redundant service for the customer. If the same router is used, then the circuits should be configured for parallel paths (i.e. using something like CEF load balancing). Unicast RPF works with Parallel path circuits between two routers.
- Customers need to ensure that the packets flow up the link (out to the Internet) need to match the prefix advertise out the link. Otherwise, Unicast RPF will filter those packets. Advertising the same routes out both links is the best way to assure this will happen.

<sup>24</sup> RFC 1998 – *An Application of the BGP Community Attribute in Multi-home Routing* is now well used in the ISP Community.

Wednesday, June 06, 2001

- The traffic-engineering trick of splitting the IP address space in half with each half advertised on one link cannot be used. For example, the Enterprise Customer cannot advertise 169.23.0.0/17 and 169.23.0.0/16 out one upstream link and 169.23.128.0/17 and 169.23.0.0/16 out the other link. This would break unicast RPF – unless an ACL is applied to Unicast RPF to allow for this case. The recommended technique is highlighted in Example 5 using BGP Communities to set Local-Pref.<sup>24</sup>

Router A

```
interface serial 1/0/1
description Link to Acme Computer's Router C
ip address 192.168.3.2 255.255.255.252
ip verify unicast reverse-path
no ip redirects
no ip directed-broadcast
no ip proxy-arp
ip route-cache distributed

router bgp 109
.
neighbor 192.168.10.3 remote-as 65000
neighbor 192.168.10.3 description Multihomed Customer - Acme Computers
neighbor 192.168.10.3 update-source Loopback0
neighbor 192.168.10.3 send-community
neighbor 192.168.10.3 soft-reconfiguration inbound
neighbor 192.168.10.3 route-map set-customer-local-pref in
neighbor 192.168.10.3 weight 255
.

ip route 192.168.10.3 255.255.255.255 serial 1/0/1
ip bgp-community new-format
```

Router B

```
interface serial 6/1/1
description Link to Acme Computer's Router C
ip address 192.168.3.6 255.255.255.252
ip verify unicast reverse-path
no ip redirects
no ip directed-broadcast
no ip proxy-arp
ip route-cache distributed

router bgp 109
.
neighbor 192.168.10.3 remote-as 65000
neighbor 192.168.10.3 description Multihomed Customer - Acme Computers
neighbor 192.168.10.3 update-source Loopback0
neighbor 192.168.10.3 send-community
neighbor 192.168.10.3 soft-reconfiguration inbound
neighbor 192.168.10.3 route-map set-customer-local-pref in
neighbor 192.168.10.3 weight 255
.

ip route 192.168.10.3 255.255.255.255 serial 6/1/1
ip bgp-community new-format
```

Router C

```
!
interface serial 1/0/
description Link to Upstream Router A
ip address 192.168.3.1 255.255.255.252
ip verify unicast reverse-path
no ip redirects
no ip directed-broadcast
no ip proxy-arp
ip load-sharing per-destination
ip route-cache distributed
```

```

interface serial 1/0
description Link to Upstream ISP Router B
ip address 192.168.3.5 255.255.255.252
ip verify unicast reverse-path
no ip redirects
no ip directed-broadcast
no ip proxy-arp
ip load-sharing per-destination
ip route-cache distributed

router bgp 65000
no synchronization
network 169.21.0.0
network 169.21.0.0 mask 255.255.128.0
network 169.21.128.0 mask 255.255.128.0
neighbor 171.70.18.100 remote-as 109
neighbor 171.70.18.100 description Upstream Connection #1
neighbor 171.70.18.100 update-source Loopback0
neighbor 171.70.10.100 send-community
neighbor 171.70.18.100 soft-reconfiguration inbound
neighbor 171.70.18.100 route-map Router-A-Community out
neighbor 171.70.18.200 remote-as 109
neighbor 171.70.18.200 description Upstream Connection #2
neighbor 171.70.18.200 update-source Loopback0
neighbor 171.70.18.200 send-community
neighbor 171.70.18.200 soft-reconfiguration inbound
neighbor 171.70.18.200 route-map Router-B-Community out
maximum-paths 2
no auto-summary
!
ip route 169.21.0.0 0.0.255.255 Null 0
ip route 169.21.0.0 0.0.127.255 Null 0
ip route 169.21.128.0 0.0.127.255 Null 0
ip route 171.70.18.100 255.255.255.255 s 1/0
ip route 171.70.18.200 255.255.255.255 s 1/1
ip bgp-community new-format
!
access-list 50 permit 169.21.0.0 0.0.127.255
access-list 51 permit 169.21.128.0 0.0.127.255
!
route-map Router-A-Community permit 10
match ip address 51
set community 109:70
!
route-map Router-A-Community permit 20
match ip address 50
set community 109:100
!
route-map Router-B-Community permit 10
match ip address 50
set community 109:70
!
route-map Router-B-Community permit 20
match ip address 51
set community 109:100
!

```

### Example 5 – Multihomed Unicast RPF

#### *Multihomed Leased Line Customers (two ISPs)*

Unicast RPF would also work with a multihomed customer who has a connection to two different ISPs. Figure 26 shows how the downstream customer (*Enterprise or ISP*) connects to two upstream ISPs. These two ISPs – *Alpha & Beta* – interconnect with each other at various places in the world (combining private peering, IXP peering, and transit).

Wednesday, June 06, 2001

Therefore each ISP will have two BGP entries for the downstream customer's prefix. Yet each ISP would select the "shortest path" entry from the BGP Table as the *best path* – enabling Unicast RPF to work.

BGP Weight should be used in Routers A & B for all the prefixes being advertised from Router C. This is necessary to provide a safeguard against AS path prepending. It is normal practice for multihomed customers to use the AS path prepending technique to affect the balance of the incoming traffic flows. There could be cases where the prepending of AS numbers would break the Unicast RPF. For example, the downstream customer prepends enough AS numbers to their advertisements to Router A such that Router A's best path to Router C would be via Router B. This means that the Router A → C forwarding path would actually select a Router A → B → C forwarding path. Unicast RPF would not have a valid path for source addresses coming up the Router C → A link, effectively blocking the downstream customer's outbound traffic on the Router C → A link. A BGP weight (see Example 5) applied on Router A & B would override the local effects of AS path prepends.

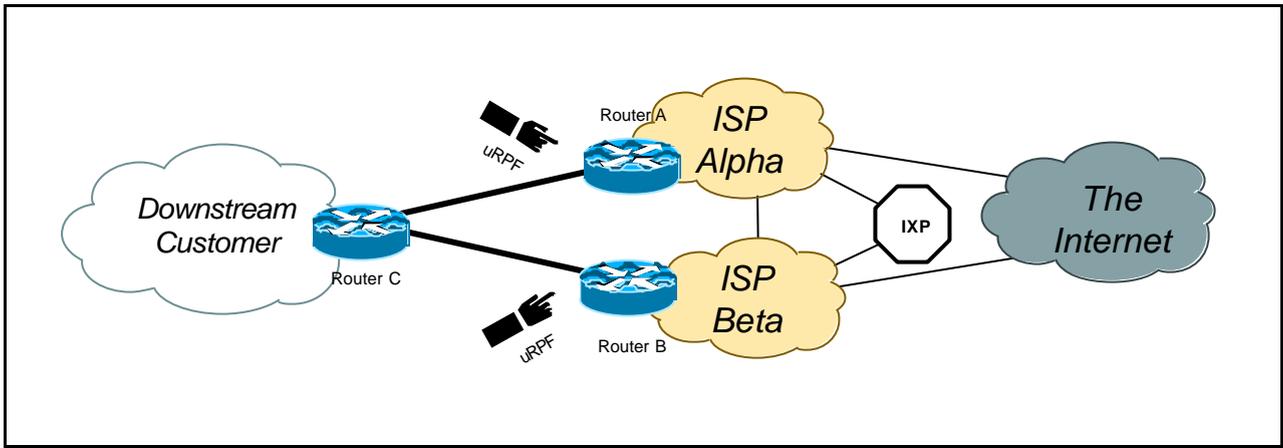


Figure 26 – Enterprise Customer Multihomed to two ISPs

Note that this is a case where the customer's traffic would be *asymmetric* through the two upstream connections. Router C would forward traffic based on the shortest path information provided by router A & B. There will be traffic flows that would exit the router C → A link yet return via the router C → B link (and vice versa). In both cases of asymmetrical flows, Unicast RPF will block unauthorised traffic from the *Enterprise Customer's* network.

## Unicast RPF for Enterprise Networks

An enterprise network would use Unicast RPF as an ingress filter to protect themselves from the Internet. This includes their upstream ISP. While ingress ACLs work fine, Unicast RPF provides some advantages over traditional ACLs. The following sections will provide some examples of how Unicast RPF is a valuable option for networks connected to the Internet.

### Single Homed Enterprise Networks – Filtering Incoming Traffic

Unicast RPF will work as an ingress filter for an Enterprise Network's one connection to the Internet. Traditionally, networks with one connection to the Internet would use ingress packet filtering to prevent spoofed packets from the Internet from entering their local network. This works well for the majority of single homed customers. Yet, there are trade offs when ACLs are used at ingress filters. Packet Per Second (PPS) Performance at very high packet rates and maintenance of the ACL (whenever there are new addresses added to the network) are two of the most referenced limitations. Unicast RPF is one tool that answers both of these limitations. With Unicast RPF, ingress filtering is done at CEF PPS rates which makes a difference when the link is more than 1 Mbps. Additionally, since Unicast RPF uses the FIB, no ACL maintenance is necessary. The follow figure and example demonstrates how this is configured.

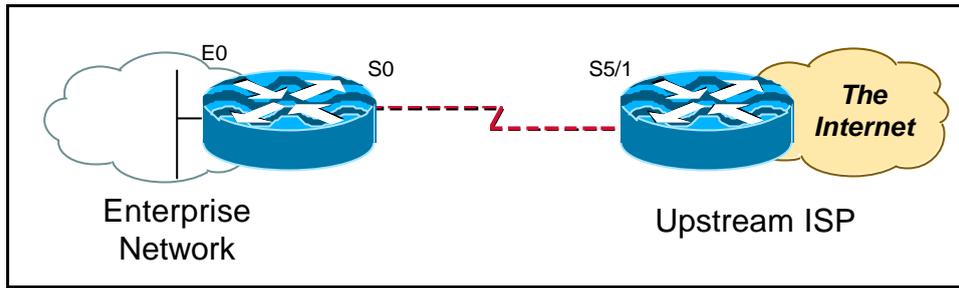


Figure 27 – Enterprise Network using Unicast RPF for Ingress Filtering

Using Figure 27, a typical configuration on the ISP's router would be as follows (assumes that CEF is turned on):

```
interface loopback 0
  description Loopback interface on Gateway Router 2
  ip address 215.17.3.1 255.255.255.255
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
!
interface Serial 5/0
  description 128K HDLC link to Galaxy Publications Ltd [galpub1] WT50314E R5-0
  bandwidth 128
  ip unnumbered loopback 0
  ip verify unicast reverse-path ! Unicast RPF activated here
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
!
ip route 215.34.10.0 255.255.252.0 Serial 5/0
```

The Enterprise Networks gateway router configuration would look something like (assumes that CEF is turned on):

```
interface Ethernet 0
  description Galaxy Publications LAN
  ip address 215.34.10.1 255.255.252.0
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
!
interface Serial 0
  description 128K HDLC link to Galaxy Internet Inc WT50314E C0
  bandwidth 128
  ip unnumbered ethernet 0
  ip verify unicast reverse-path ! Unicast RPF activated here
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
!
ip route 0.0.0.0 0.0.0.0 Serial 0
```

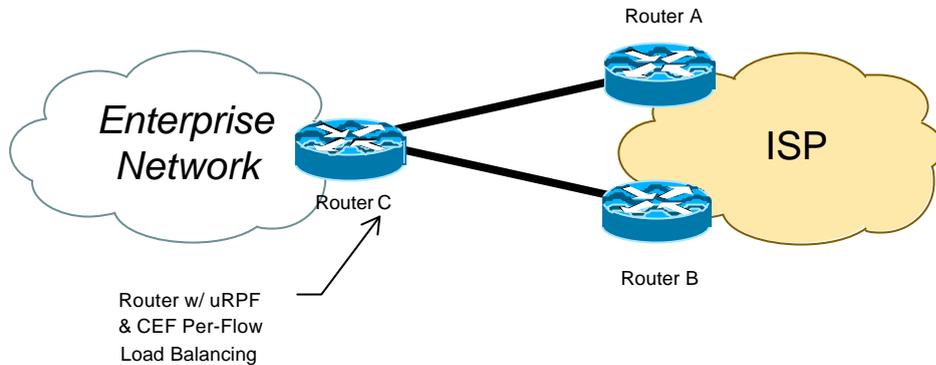
Notice that Unicast RPF works with a single default route. There are no additional routes or routing protocols. Network 215.34.10.0/22 is a connected network. Hence Unicast RPF will drop packets coming from the Internet with a source address in 215.34.10.0/22's range.

### ***Multi-Homed Enterprise Networks to One Upstream ISP – Filtering Incoming Traffic***

Unicast RPF **will** work as an ingress filter for multi-homed connections to the Internet. There is a perception that this cannot be done. The reality is that it will work – with some thought and planning on the Enterprise's Network Engineers.

Wednesday, June 06, 2001

Using the configuration in Example 5 and Figure 28 as a reference, the Enterprise network has configured their network to use both upstream links to the ISP. Since the BGP information provided to Router C from Router A & B will result in equal AS paths to the Internet, BGP will select by default the path with the lowest *router-id*. The BGP *maximum-paths* statement is used to ensure that both paths to the upstream ISP are added to the forward table. Since the two paths are equal, Unicast RPF will permit any traffic from the Internet that is in the BGP tables from the upstream ISP. Upstream load balancing between the two equal paths are best done with CEF's Per-Packet or Per-Flow load balancing.



**Figure 28 – Multi-Homed Enterprise Networks**

Unicast RPF as an ingress filter in the enterprise has two key advantages as an ingress filter over the traditional ingress ACLs filters – performance and filtering unallocated spoofed addresses. As mentioned before, Unicast RPF operates in the CEF switching path. This reduces impact of filtering on the PPS forwarding performance of the router. In addition to the performance benefits, Unicast RPF applied in this way for ingress filters will drop any packet with a source address that is not in the forward table. Hence packets using source address that have not been allocated by one of the IP registries are dropped. It has been demonstrated that some denial of service attacks use these un-allocated IP addresses as a way to bypass an RFC 1918 filter. If RFC 1918 filters are applied to the incoming BGP traffic, then no ingress filters are needed. Unicast RPF covers all known source address spoofing techniques.

### ***Multi-Homed Enterprise Networks to Multiple Upstream ISP – Filtering Incoming Traffic***

While the BGP *maximum-paths* enables the Unicast RPF technique to be used with the same upstream ISP, it does not help with there are multiple ISPs. BGP feeds from multiple ISPs means that best paths will be selected. This means that routing asymmetry would have packets going out the best path to one of the ISPs, but returning back the path of another ISP that is not the best path – resulting in Unicast RPF dropping the packets. In this case, the Enterprise ISP has several options:

1. Use traditional ACL based Ingress Packet Filters.
2. Use Unicast RPF with a BGP filter on the local router so that BGP derived routes are not used in the local forwarding path. Equal static defaults and CEF Per-Flow load balancing is used.

Traditional ACLs are tools that work. Unicast RPF replaces ACLs for many situations of ingress filtering, but there are situations where they don't. When an Enterprise customer is connected to many upstream ISPs and using BGP to select the closest path to their site, Unicast RPF will tend to break with the return flows become asymmetric. Hence, an ACL based ingress filter that denies any packet with a source address matching the allocated IP address block of the Enterprise Network is the best workable option.

The only known alternative<sup>25</sup> is for the Enterprise network to use BGP just for the advertising their prefixes to the Internet. Inbound BGP from the multiple ISPs are filtered at the Enterprise so that *no* BGP information is used in the Enterprise's

<sup>25</sup> Engineers are always finding new ways of doing things. Although this is the "only known alternative," it is expected that peers in the community will develop other interesting techniques.

forward table. Instead, the Enterprise uses static default with CEF per flow load balancing for their upstream traffic. This permits Unicast RPF to work.

There are two disadvantages to this technique. First, the upstream traffic to the Internet may not take the shortest path to the destination site. Second, since Unicast RPF is filtering based on default, RFC 1918 and non-allocated IP addresses in the source address will get past the Unicast RPF filter. Both of these factors are limitations that might be unacceptable to the Enterprise Network.

### Where not to use Unicast RPF

Unicast RPF should not be used on interfaces that are *internal* to the ISP. Internal interfaces are likely to have routing asymmetry could happen (see Figure 29). Unicast RPF should only be applied where this is natural or configured symmetry (see sections on using Unicast RPF with multihomed customers). As long as ISP Engineers mindfully plan which interfaces they activate Unicast RPF on, routing asymmetry is actually not a serious problem.

For example, routers at the edge of an ISP's network are more likely to have symmetrical reverse paths. Alternatively, routers that are in the core of the ISP's network have no guarantee that the best forwarding path out of the router will be the path selected for packets returning to the router (see Figure 29). Hence, it not recommended apply *ip verify unicast reverse-path* where there is a chance of asymmetric routing. It is simplest to place Unicast RPF on the customer edge of an ISP's network (as in Figure 23).

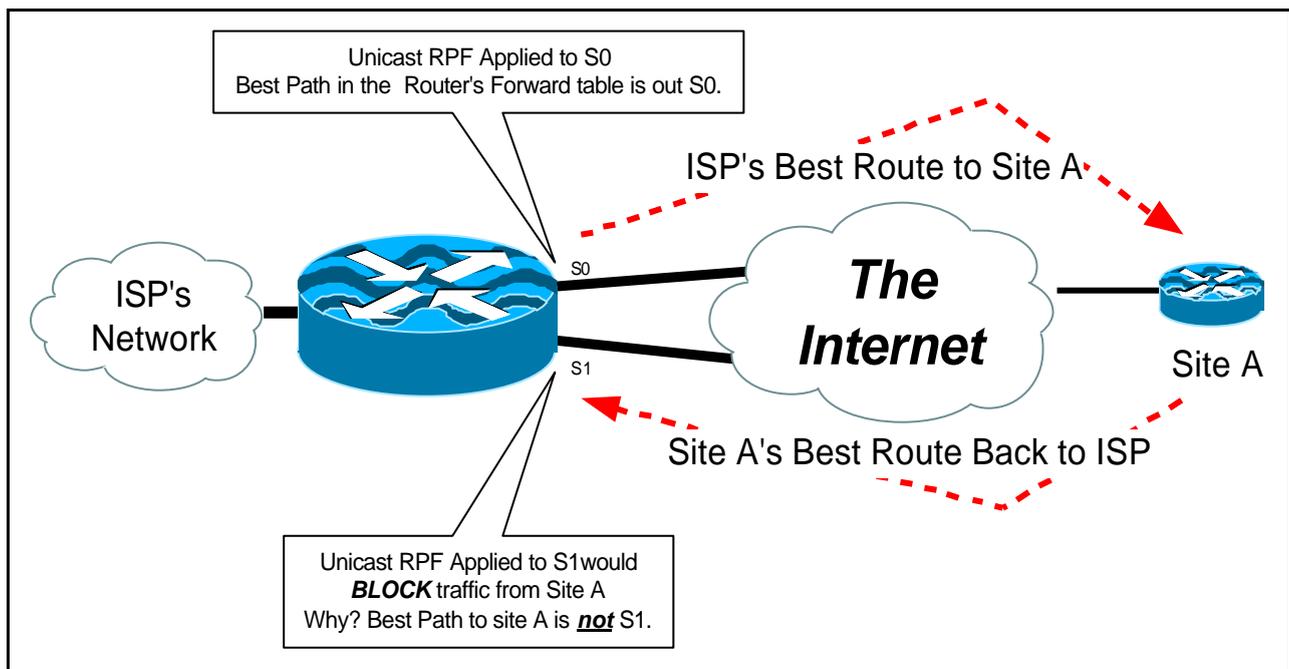


Figure 29 – How asymmetrical routing would not work with Unicast RPF

## Unicast RPF Examples – Putting it all together

The following is an ISP allocated CIDR block 165.21.0.0/16 with both inbound and outbound filters on the upstream interface:

```
ip cef distributed
!
interface Serial 5/0/0
description Connection to Upstream ISP
ip address XXX.XXX.XXX.XXX 255.255.255.252
```

Wednesday, June 06, 2001

```
no ip redirects
no ip directed-broadcast
no ip proxy-arp
ip verify unicast reverse-path
ip access-group 111 in
ip access-group 110 out
!
access-list 110 permit ip 165.21.0.0 0.0.255.255 any
access-list 110 deny ip any any log
access-list 111 deny ip host 0.0.0.0 any log
access-list 111 deny ip 127.0.0.0 0.255.255.255 any log
access-list 111 deny ip 10.0.0.0 0.255.255.255 any log
access-list 111 deny ip 172.16.0.0 0.15.255.255 any log
access-list 111 deny ip 192.168.0.0 0.0.255.255 any log
access-list 111 deny ip 165.21.0.0 0.0.255.255 any log
access-list 111 permit ip any any
```

## Other Considerations

This example used a very simple single homed ISP to demonstrate the concepts of ingress/egress filters. Be mindful that ISPs are usually not single homed (or if they are, they soon become multihomed). Hence, provisions for asymmetrical flows<sup>26</sup> need to be designed into the filters on the ISP's borders.

## *Committed Access Rate (CAR) to Rate Limit or Drop Packets<sup>27</sup>*

### Why use a QoS Tool for Security?

The use of a QoS tool like Committed Access Rate (CAR) is an unintended result of the types of attacks on the Internet. In 1997 a new generation of attacks were launched on the Internet – SMURFs. The “smurf” attack is a specific Denial of Service (DoS) attack, named after its exploit program. It is a recent category of network-level attacks against hosts. A perpetrator sends a large amount of ICMP echo (ping) traffic to specific IP broadcast addresses. All the ICMP echo packets will have the spoofed source address of a victim. If the routing device delivering traffic to those broadcast addresses performs the IP broadcast to layer 2, then the ICMP broadcast function will be forward to all host on the layer 2 medium (see Figure 30). Most hosts on that IP network will take the ICMP echo request and reply to it with an echo reply. This multiplies the traffic by the number of hosts responding. On a multi-access broadcast network, there could potentially be hundreds of machines to reply to each packet.

The “smurf” attack’s cousin is “fraggle”, which uses UDP echo packets in the same fashion as the ICMP echo packets; it was a simple re-write of the “smurf” programme. Currently, the systems most commonly hit are Internet Relay Chat (IRC) servers and their providers.

Two parties are hurt by this attack:

- The intermediary (broadcast) devices – called the “amplifiers”
- The spoofed address target – the “victim”

The victim is the target of the large amount of traffic that the amplifiers generate.

Consider a scenario, which paints a picture of the dangerous nature of this attack. Assume a co-location switched network with 100 hosts, and that the attacker has a T1 circuit. The attacker sends, say, a 768kb/s stream of ICMP echo (ping) packets, with a spoofed source address of the victim, to the broadcast address of the “bounce” or amplifier site. These ping packets hit the bounce site’s broadcast network of 100 hosts; each of them takes the packet and responds to it, creating 100

<sup>26</sup> *Asymmetrical Flows* are when the outbound traffic goes out one link and returns via a different link.

<sup>27</sup> This section is a edited version of Craig A. Huegen’s work on SMURF and FRAG protection. For the latest information, please refer to Craig’s page at <http://www.pentics.net/>. Craig can be contacted at [chuegen@cisco.com](mailto:chuegen@cisco.com).

ping replies out-bound. If you multiply the bandwidth, you will see that 76.8 Mbps is generated outbound from the “bounce site”. Because of the spoofed source address of the originating packets, these reply packets are then directed towards the victim.

## Passive SMURF Defences

All the various tools listed previously are effective in minimising SMURF on the Internet. The top three are listed below. Please review each section for details.

- Interface Services (*no ip directed broadcast*)
- Egress and Ingress Filtering
- Committed Access Rate (CAR)

## Active SMURF Defences

Active defences are tools/techniques/procedures executed during attacks. They are used to limit and/or block the attack in progress. In many instances these tools will have an effect on other Internet applications and services – yet the trade off is between no Internet services and limited disruption. It is highly recommended that the ISP document and train staff on the use of these tools. In that way, the ISP’s NOC can quickly respond to an attack in progress.

Cisco’s IOS has security tools to cover the broad range of networking. New security tools are being added all the time. Yet, many of these tools are situation specific. Some tools were written to help enterprise networks – other tools/techniques are more suited for ISP environment. The following sections list the known tools.

### *Rate Limiting with CAR*

It is an inevitable fact on the Internet that every ISP is bound to experience a Denial of Service (DoS) attack. Hence, ISPs should have tools and procedures in place to respond to these DoS attacks. Committed Access Rate (CAR) is one such tool that an ISP can use react to DoS attacks.

CAR is a functionality that works with Cisco Express Forwarding, found in 11.1CC and releases from 12.0. It allows network operators to rate limit certain types of traffic to specific sources and/or destinations. The main advantage of CAR is that it can work on packets as they arrive on the router’s interface – dropping/rate limiting the DoS flow before any other packet processing.

The following URLs provide details on CAR:

Committed Access Rate (CAR)

<http://www.cisco.com/warp/public/732/Tech/car/index.html>

Configuring Committed Access Rate

[http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos\\_c/qcpart1/qccar.htm](http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcpart1/qccar.htm)

**Example 1** – A provider has filtered its IRC server from receiving ICMP echo-reply packets in order to protect it. Now many attackers are going after the customer’s devices in order to fill some network segments.

The provider above chose to use CAR in order to limit all ICMP echo and echo-reply traffic received at the borders to 256 Kbps. An example follows:

```
! traffic we want to limit
access-list 102 permit icmp any any echo
access-list 102 permit icmp any any echo-reply
! interface configurations for borders
interface Serial3/0/0
rate-limit input access-group 102 256000 8000 8000 conform-action transmit exceed-action drop
```

Wednesday, June 06, 2001

This limits ICMP echo and echo-reply traffic to 256 Kbps with a small amount of burst. Multiple “rate-limit” commands can be added to an interface in order to control other kinds of traffic as well.

The command “*show interface [interface-name] rate-limit*” will show the statistics for rate-limiting; “*clear counters [interface-name]*” will clear the statistics for a fresh look.

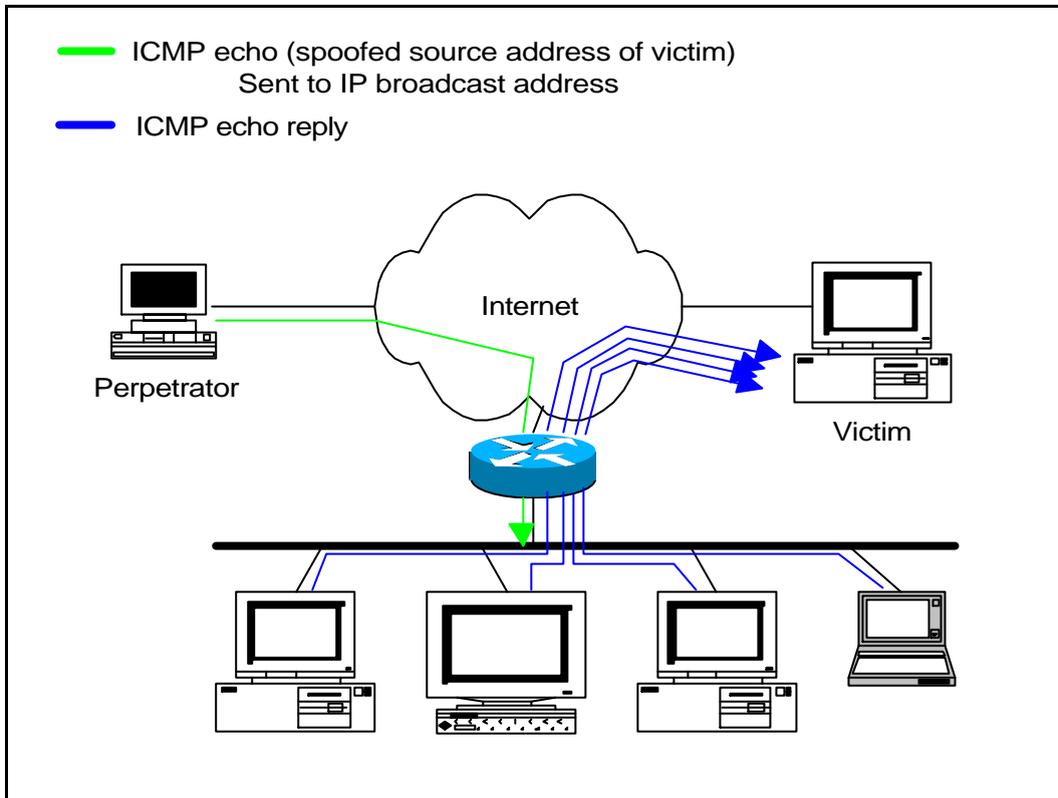


Figure 30 – How SMURF uses amplifiers

**Example 2** – Use CAR to limit TCP SYN floods to particular hosts -- without impeding existing connections. Some attackers have started using very high streams of TCP SYN packets in order to harm systems.

This example limits TCP SYN packets directed at host 10.0.0.1 to 8 kbps or so:

```
! We don't want to limit established TCP sessions -- non-SYN packets
!
access-list 103 deny tcp any host 10.0.0.1 established
!
! We do want to limit the rest of TCP (this really only includes SYNs)
!
access-list 103 permit tcp any host 10.0.0.1
!
! interface configurations for network borders
!
interface Serial3/0/0
rate-limit input access-group 103 8000 8000 8000 conform-action transmit exceed-action drop
!
```

## Reacting to Security Incidents

### Based on the Evolving Threats – What should the ISP's Objective be?

#### First Policy Decision – Drop It, Rate Limit It, or Do Nothing.

The first policy decision an ISP will need to make is to decide what to do with the packets when an attack is identified. When a DDOS attacks are identified and classified, the ISP will need to make a decision. Classification will provide the DDOS flow's source IP addresses, hence a target for filtering. However, do you filter all the packets from that source address, some of the packets from that source address, or just rate limit the attack from that address?

The source IP addresses from DDOS attacks can have one of four characteristics. The characteristic of the source will be a factor in the decision on course packet drops, detailed packet drops, rate limiting, or no action at all.

1. **Spoofed RFC 1918 and Special use addresses.** These are the easiest addresses to spot and drop. They are well known, not in the global Internet Route Table, and can easily be dropped with passive packet filtering tools. Many networks have ingress/egress packet filters, which block these packets from propagating through the Internet. Unfortunately, these passive packet filters are not ubiquitous. So, there will be occasions when an ISP will see DDOS attacks with these addresses. The reaction decision is easy – drop these.
2. **Spoofed addresses that are not in the Global Internet Route Table.** Cyberpunks have found way to get around the well known packet filters that drop well known RFC 1918 and special use addresses. The most common technique is to use addresses that have not been allocated to any ISP nor added to the Global Internet Route table. Regional Internet Registries (RIRs) allocation blocks of IP addresses to ISPs and sub-registries. These RIR allocations come from delegations of authority from the Internet Assigned Numbers Authority (IANA). The list of IPv4 address blocks that have not been activated is easily accessible at <http://www.isi.edu/in-notes/iana/assignments/ipv4-address-space>. This makes it easy for someone to create an attack with spoofed addresses based on this list. Yet, since these addresses are not in the Global Internet Route Table, checking the forward table can easily aid their identification. Some tools – like Unicast RPF – make this check, making it harder for these attacks to be successful. The reaction when faced with these sorts of attacks is to drop the packets.
3. **Valid address from a DDOS Agent** Some attackers will sacrifice the DDOS agent they have penetrated in order to achieve their objectives. In this case, the source address will be the actual address of the penetrated system. For some attacks (i.e. TCP syn and ack attacks) these would be easy to identify and classify. Other attacks, which are designed to overwhelm the target with valid traffic, are not as easy to identify and classify – especially when the attacks are based on simple HTTP GET request. Once a particular source address has been classified as a DDOS agent, the ISP needs to decide whether to drop the all the packets from that penetrated machine, drop only the attacking traffic flow, or just rate limit the attack. This is a valid computer with a person(s) behind. Hence, dropping all packets from this source address may seem extreme. Yet, it is a penetrated machine – making it a threat that needs to be resolved. The reaction decision for these sorts of attacks is mixed. Most can be dropped. A few may just be rate limited.
4. **Spoofed address from a valid address somewhere else on the Internet.** The most difficult source addresses to react to are those that are valid addresses but are spoofing another site. This is a case where you will get a reverse DDOS attack. The attacker(s) actual target is not the target the DDOS agents send packets to, but the real target is the one whose addresses are being spoofed. For example, the real target is an enterprise with several connections to the Internet. The attacker knows that several connections, redundancy, IDS tools, firewalls, and other tricks make the enterprise a tough target to take down. So instead of trying to take down the enterprise's network, the attacker works to get others to isolate it from the Net. DDOS attacks spoofing the enterprise's source IP range are frequently attempted.

Wednesday, June 06, 2001

## ***ISP Security References and Deepening Materials***

Please refer to the Technical References section at the end of this document for references and pointers to further technical documents discussing ISP security issues.

---

## ROUTING

---

*The Routing chapter of this document assumes that the ISP Engineer has a working understanding of the core routing protocols used in the Internet. If not, please refer to the on-line whitepapers and/or publication listed in the technical reference section.*

### *Hot Standby Routing Protocol*

A new feature in recent versions of IOS is HSRP (see RFC2281 – <http://www.ietf.org/rfc/rfc2281.txt>). This feature is especially useful on LANs within the ISP's backbone, for example for network servers, non-Cisco access servers, and hosted servers.

The motivation for this protocol is to support the need for a default gateway on LAN networks when there are two gateway routers providing connectivity to the wider network and Internet. Routers tend to support the full set of routing protocols. Computer workstations in majority run variants of Unix or Windows95/NT, for which there are either no or minimally functional routing software. Configuring something like the public domain software *gated* or the vendor's own software to perform a dynamic routing function is usually a poor and unreliable compromise. The best solution is to configure a static default route and use HSRP.

Figure 31 shows a typical LAN with two routers used to connect to the ISP backbone. To implement HSRP, the configuration required for these two routers looks something like this:

```
Router 1:
interface ethernet 0/0
description Server LAN
ip address 169.223.10.1 255.255.255.0
standby 10 ip 169.223.10.254
```

```
Router 2:
interface ethernet 0/0
description Service LAN
ip address 169.223.10.2 255.255.255.0
standby 10 priority 150
standby 10 preempt
standby 10 ip 169.223.10.254
```

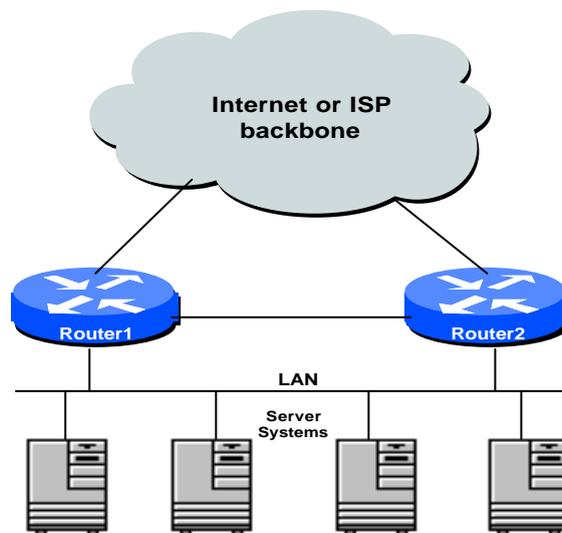


Figure 31 – Dual gateway LAN

Wednesday, June 06, 2001

The two routers have their LAN IP addresses conventionally defined in the configuration above. However, another IP address has been defined, in the command line `standby 10 ip 169.223.10.254`. This address is the address of the virtual default gateway defined on the LAN. All the systems on the LAN, apart from **router1** and **router2** use this address as the “default route”. Router2 has a standby priority of 150, higher than the default of 100. Therefore **router2** will be the default gateway at all times unless it is unavailable (i.e. down). The `preempt` directive tells **router1** and **router2** that **router2** should be used as default gateway whenever possible. For example, if **router2** were temporarily out of service, it would take over from **router1** when it is returned to normal operation.

A problem here is that all the outbound traffic goes through **router2**, whereas inbound traffic may be shared between the two. Some ISPs like to share outbound traffic between the two routers, and this is achieved by setting up two standby groups as in the following example:

```
Router 1:
interface ethernet 0/0
description Server LAN
ip address 169.223.10.1 255.255.255.0
standby 10 ip 169.223.10.254
standby 11 priority 150
standby 11 preempt
standby 11 ip 169.223.10.253
```

```
Router 2:
interface ethernet 0/0
description Service LAN
ip address 169.223.10.2 255.255.255.0
standby 10 priority 150
standby 10 preempt
standby 10 ip 169.223.10.254
standby 11 ip 169.223.10.253
```

Provided the ISP shares the default route configuration on the servers between the two virtual default gateways he should see a reasonably balanced sharing of traffic on the two outbound links.

## ***CIDR Features***

All network devices connected to the Internet should be CIDR compliant. Cisco routers are made CIDR compliant if the two commands are entered:

```
ip subnet-zero
ip classless
```

All Cisco routers connected to the Internet must have these commands turned on. IOS releases from 12.0 have these commands turned on by default. It makes good sense in the ISP operations world never to rely on any equipment vendor’s defaults. Defaults are for enterprise and other networks, not the specialised and public ISP Backbones. (See BGP section for BGP requirements.)

## ***Selective Packet Discard***

When a link goes to a saturated state, the router will drop packets. The problem is that the router will drop any type of packets – including routing protocol packets. Selective Packet Discard (SPD) will attempt to drop non-routing packets instead of routing packets when the link is overloaded.

The basic idea behind Selective Packet Discard (SPD) is this: if we mark all BGP and IGP packets as being “important” and prefer these packets over others, we should process a larger percentage of the packets that will allow routing and, consequently, keep the BGP and IGP sessions stable.

SPD can be configured as shown below – some of these are hidden from the IOS configuration helper, but have been made known to the ISP community. (Note the difference between 11.1CC and 12.0 release versions of the commands.)

```
[no] ip spd enable    ! for 11.1CC
[no] spd enable      ! for 12.0+
```

turns SPD on/off, default is on

```
ip spd headroom      ! for 11.1CC
spd headroom         ! for 12.0+
```

Default value is 100. Specifies how many high-precedence packets we will enqueue over the normal input hold queue limit. This is to reserve room for incoming high precedence packets.

```
[no] ip spd queue {min-threshold | max-threshold} <n>
```

Sets lower and upper ip process-level queue thresholds for SPD. With SSE based SPD, lower precedence packets are randomly dropped when the queue size hits min-threshold. The drop probability increases linearly with the queue size until max-threshold is reached, at which point all lower precedence packets are dropped. For regular SPD, lower precedence packets are dropped when the queue size reaches min-threshold. Defaults are 50 and 75, respectively. These values were not based on real life experience and may need some tuning.

New “show ip spd” exec command:

```
Current mode: normal.
Queue min/max thresholds: 73/74
IP normal queue: 0, priority queue: 0.
External SPD enabled.
```

(Shows SPD mode, current and max size of IP process level input queue, and status of external (SSE) SPD. SPD mode will be one of disabled, normal, random drop, or full drop. The priority queue is where high-precedence packets go.)

“show interface switching” has some extra information too:

```
gmajor#sh int eth 3/0 switching
Ethernet3/0
  Throttle count:          0
SPD Flushes Fast          0      SSE      542019
  (separates RP/SP flushes)
SPD Priority Inputs:      123     Drops:      0
  (priority packets received and dropped due to exceeding
  headroom threshold.)
```

Note that “switching” is a hidden command, and needs to be entered in full to be recognised by the command parser.

## IP Source Routing

The Cisco IOS software examines IP header options on every packet. It supports the IP header options Strict Source Route, Loose Source Route, Record Route, and Time Stamp, which are defined in RFC 791. If the software finds a packet with one of these options enabled, it performs the appropriate action. If it finds a packet with an invalid option, it sends an ICMP Parameter Problem message to the source of the packet and discards the packet.

Wednesday, June 06, 2001

IP has a feature that allows the source IP host to specify a route through the IP network. This feature is known as source routing. Source routing is specified as an option in the IP header. If source routing is specified, the software forwards the packet according to the specified source route. The default is to perform source routing.

Some ISPs do not want their customers to have access to source routing, hence it is turned off at the customer edge. The way to do this is the generic command:

```
no ip source-route
```

Some ISPs like to have source routing available to troubleshoot their and other networks – especially when routing has broken inside one of those networks. Other ISPs require source routing to be turned on when peering with ISPs. An example of using traceroute with source-routing enabled on the intermediate networks is given below. The ISP is checking the path from 192.121.154.170 to [www.sprint.net](http://www.sprint.net) – the **-g** option specifies that traceroute should do the probe as though packets are coming from the 192.121.154.170 source address. Notice the **!S** in step 18 – this states that that host (router) is blocking ip source routing.

```
Unix% /usr/local/bin/traceroute -g 192.121.154.170 www.sprint.net
traceroute to www.sprint.net (208.27.196.10), 30 hops max, 40 byte packets
 1 fe0-1-0.hrl.cbgl.gbb.uk.uu.net (158.43.128.192)  1.294 ms  0.703 ms  0.539 ms
 2 pos0-0.cr1.cbgl.gbb.uk.uu.net (158.43.129.129)  0.748 ms  0.993 ms  0.747 ms
 3 pos0-0.cr2.cbgl.gbb.uk.uu.net (158.43.129.133)  1.586 ms  1.146 ms  2.145 ms
 4 pos0-2.cr2.lnd6.gbb.uk.uu.net (158.43.254.2)    4.43 ms   4.143 ms  3.731 ms
 5 ge2-0.cr1.lnd6.gbb.uk.uu.net (158.43.254.65)   4.395 ms  4.044 ms  4.148 ms
 6 POS11-0-0.GW2.LND1.Alter.Net (146.188.5.41)   4.898 ms  10.705 ms 5.082 ms
 7 122.at-2-0-0.XR2.LND2.Alter.Net (146.188.15.170) 5.995 ms  6.179 ms  6.039 ms
 8 194.ATM1-0-0.HR2.LND1.Alter.Net (146.188.15.129) 12.422 ms 7.229 ms  6.018 ms
 9 146.188.13.165 (146.188.13.165) 7.4 ms   8.331 ms  8.376 ms
10 gblon802-ta-d5-1-0.ebone.net (192.121.154.113) 16.846 ms 15.647 ms 21.246 ms
11 gblon801-ta-p6-0-0.ebone.net (192.121.154.170) 21.513 ms 21.535 ms 21.871 ms
12 gblon304-tb-p0-1.ebone.net (195.158.226.233) 366.643 ms 358.076 ms 269.65 ms
13 usnyk106-tc-p0-3.ebone.net (195.158.229.17) 359.798 ms 204.034 ms 241.634 ms
14 s1-bb11-nyc-5-3.sprintlink.net (144.232.9.229) 380.835 ms 84.641 ms 96.338 ms
15 s1-bb10-nyc-9-0.sprintlink.net (144.232.7.1) 94.433 ms 90.512 ms 87.394 ms
16 s1-bb12-rly-7-0.sprintlink.net (144.232.9.226) 102.331 ms 88.533 ms 93.294 ms
17 s1-bb11-rly-8-0.sprintlink.net (144.232.7.213) 89.341 ms 103.678 ms 174.892 ms
18 s1-bb5-dc-4-0-0.sprintlink.net (144.232.7.166) 86.662 ms !S 88.132 ms !S *
```

As a general rule of thumb, *if you are not using ip source routing, turn it off*. IP source routing is a well-known security vulnerability used in attacks against a system.

## Configuring Routing Protocols

This section examines the most efficient and effective ways of configuring IGP and BGP to give greatest scalability in IOS.

There are essentially three IGPs which are used by ISPs. These are ISIS (Intermediate System to Intermediate System), OSPF (Open Shortest Path First) and EIGRP (Enhanced Internal Gateway Routing Protocol). The former are industry standards (ISIS was developed by OSI now being enhanced by the IETF, OSPF was developed by the IETF), while EIGRP is an enhancement of IGRP developed by Cisco since the late '80s.

BGP version 4 has been used since 1994 as the routing protocol for autonomous systems to exchange prefixes with each other. BGP4 is the classless version of the Border Gateway Protocol and has seen many enhancements over the last few years as the Internet grows larger and larger.

## Router ID

The router identifier in an IGP is chosen from the loopback interface of the router, if configured. If more than one loopback interface is configured with an IP address, the router identifier is the highest of those IP addresses. If the loopback is not

configured, the highest IP address configured on the router at the time the IGP was started is used. ISPs prefer stability, so the loopback interface is usually configured and active on most ISP routers.

The router ID is also used as the last step of the BGP path selection process. Another reason to ensure that the loopback interface is configured and has an IP address – if there is no loopback, the router ID is the highest IP address configured on the box at the time the BGP process was started.

## Choosing an IGP

A general discussion on the choice of IGPs is currently beyond the scope of this document. However, the choice of IGP generally seems to be made on the basis of experience, as technically there is little to choose between the three for most practical purposes. Those engineers with a strong ISIS background will always choose ISIS. Those with a strong OSPF background will always choose OSPF. The rule of thumb seems to be that beginners to interior routing choose EIGRP as it is easy to get started. However, OSPF is a better choice as it forces good IGP design to ensure that the network will scale. And those who are very experienced tend to choose ISIS on Cisco routers as it allows for better scaling with more configuration options than the other two IGPs.

## Configuring an IGP

How to configure an IGP in detail is also currently beyond the scope of this document too. There are many good documents available on CCO which can help with IGP design. However, listed below are some good IOS tips available which will help ISPs whichever IGP is chosen. There is no harm in repeating the ISP engineer's well known design tips though:

- There are three types of prefixes:
  1. Access network prefixes
  2. Infrastructure prefixes
  3. External prefixes
- IGPs carry **infrastructure** prefixes only
- Access network prefixes are not part of the infrastructure – use **ip unnumbered** if possible
- Customer prefixes are **never** carried in an IGP – BGP is designed for this
- IGPs are kept small for best convergence speed – use a good address plan and summarisation
- iBGP and eBGP prefixes are **never ever** distributed into an IGP
- IGP prefixes are **never ever** distributed into BGP

## Putting Prefixes into the IGP

### *The Network Statement*

There are at least three possible ways of inserting prefixes into the IGP. However, ISPs aim to find the most efficient method, and one which will give greatest scalability of their network. The most efficient, scalable, and safe method is to use the **network** statement to cover the infrastructure addresses which will have an active IGP running over them and/or require to be carried in the IGP. A sample configuration might be:

```
!
interface loopback 0
 ip address 220.220.16.1 255.255.255.255
!
interface serial 0/0
 ip address 220.220.17.1 255.255.255.252
!
interface serial 1/0
 ip unnumbered loopback 0
!
router ospf 100
 network 220.220.16.1 0.0.0.0 area 0
 network 220.220.17.0 0.0.0.3 area 0
 passive-interface loopback 0
```

Wednesday, June 06, 2001

```
    passive-interface serial 1/0
!
```

Notice the use of the passive-interface command to disable OSPF neighbour discovery on any interfaces which do not have connected neighbours or don't need to run a routing protocol.

It is not good practice, and indeed strongly discouraged to redistribute anything into an interior routing protocol. It is rarely required.

### ***Redistribute Connected into an IGP***

Using the **redistribute connected** command takes the prefixes assigned to every connected interface and injects them into the IGP. Apart from creating an external network type which, for example, in OSPF is carried across all areas, the router periodically has to examine the Routing Information Base (RIB) to see if there are any changes to the connected state. This takes extra CPU cycles. Additionally, any link state changes are passed into the IGP, something which may not be desirable, especially in the case of link addresses going to external connections. In the OSPF example, the external link state change is heard over the whole network.

Another problem is that redistributing connected interfaces into an IGP covers **all** connected interfaces on a routing device. Some addresses maybe aren't desirable in an IGP, so ISPs would then have to add a route-map to apply an access-list filter to the redistribution process. Which could add something else to go wrong – access-lists can be deleted by accident, resulting in prefixes being leaked into the IGP. Again, undesirable.

**Conclusion: don't use redistribute connected.**

### ***Redistribute Static into an IGP***

Using the **redistribute static** command takes all static routes and injects them into the IGP. Apart from creating an external network type which, for example, in OSPF is carried across all areas, the router periodically has to examine the Routing Information Base (RIB) to see if there are any additions to or deletions from the static route configuration. This takes extra CPU cycles. And if the next hop to which the static route points disappears, the static route is withdrawn, with the resulting withdrawal from the IGP. (Ofcourse, the recently introduced permanent static route is designed to workaroud this problem, but the problem is best avoided in the first place!)

The other problem with redistributing static routes into an IGP is that it covers **all** static routes configured on a routing device. Some prefixes maybe aren't desirable in an IGP, resulting in a situation similar to that noted in the "redistributed connected" case.

**Conclusion: don't use redistribute static.**

### ***Redistribute <anything> into an IGP***

**Conclusion: don't.** ☺

## **IGP Summarisation**

It is good practice, and aids scalability, to ensure that summarisation is implemented wherever possible. IGP's work most efficiently with as few prefixes as possible in the routing protocol. Small IGP's converge more quickly, ensuring rapid network healing in case of link failures. How to configure summarisation is different for each protocol – consult the documentation for those. For example, OSPF uses the "**area <n> range**" OSPF subcommand.

## **IGP Adjacency Change Logging**

Enable neighbour state logging in each IGP. This means that it becomes easier to find out about neighbour states, reasons for state changes, etc. In OSPF's case, logging is enabled by the OSPF subcommand "**ospf log-adjacency-changes**".

If logging is enabled, log messages are sent to the router console, and wherever else the logging output has been configured. For most ISPs, this would be a Unix syslog server – see the section discussing router logs for how to configure this. The typical output would be similar to what is given below – this example shows the activation of a neighbour adjacency with a router connected to serial3/2.

```
Nov 10 03:56:23.084 AEST: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.2.2 on Serial3/2 from DOWN to
INIT, Received Hello
Nov 10 03:56:32.620 AEST: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.2.2 on Serial3/2 from INIT to
2WAY, 2-Way Received
Nov 10 03:56:32.620 AEST: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.2.2 on Serial3/2 from 2WAY to
EXSTART, AdjOK?
Nov 10 03:56:32.640 AEST: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.2.2 on Serial3/2 from EXSTART
to EXCHANGE, Negotiation Done
Nov 10 03:56:32.676 AEST: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.2.2 on Serial3/2 from EXCHANGE
to LOADING, Exchange Done
Nov 10 03:56:32.676 AEST: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.2.2 on Serial3/2 from LOADING
to FULL, Loading Done
```

## Putting Prefixes into BGP

### *The Network Statement*

As with IGP, there are at least three possible ways of inserting prefixes into BGP. The safest method is to use is one **network** statement per prefix which is to be injected into BGP. A sample configuration might be:

```
interface loopback 0
 ip address 220.220.16.1 255.255.255.255
!
interface serial 1/0
 ip unnumbered loopback 0
!
router bgp 100
 no synchronization
 network 220.220.18.0 mask 255.255.252.0
!
 ip route 220.220.18.0 255.255.252.0 serial 1/0
```

It is not good practice, and indeed strongly discouraged, to redistribute any interior routing protocol into BGP. It is never required and in the past has led to many serious accidents on the global Internet.

### *Redistribute Connected into BGP*

Using the **redistribute connected** command takes the prefixes assigned to any connected interface and injects them into BGP. One disadvantage, as with IGP, is that the router periodically has to examine the RIB to see if there are any changes to the connected state, taking extra CPU cycles. Additionally, any link state changes are passed into BGP generating a route flap on the Internet routing table, wasting considerable CPU cycles on all routers in the Internet which have a view of this prefix.

Another problem is that redistributing connected interfaces into BGP covers **all** connected interfaces on a routing device. If some addresses aren't desirable in BGP, ISPs would need to add a route-map to apply an access-list filter to the redistribution process. Which could add something else to go wrong – access-lists can be deleted by accident, resulting in prefixes being leaked into BGP. Again, very undesirable.

**Conclusion: don't use redistribute connected.**

### *Redistribute Static into BGP*

Using the **redistribute static** command takes all static routes and injects them into BGP. Again the router periodically has to examine the RIB to see if there are any additions to or deletions from the static route configuration. This takes extra

Wednesday, June 06, 2001

CPU cycles. And if the next hop to where the static route points disappears, the static route is withdrawn, with the resulting withdrawal from BGP. A withdrawal from BGP results in a route flap, as above. (As in the IGP case, the permanent static route is designed to work around this issue, but the problem is best avoided in the first place!)

Again, redistributing static routes into BGP covers **all** static routes configured on a routing device. Some prefixes maybe aren't desirable in BGP (eg the RFC1918 prefixes), so ISPs would add a route-map to apply an access-list filter to the redistribution process. This can potentially lead to the situation discussed in the case of "**redistribute connected**" as per above.

Some situations demand **redistribute static**, especially where communities are attached to the prefixes inserted into the BGP process. This appears to be the preferred process amongst ISPs as they find it easier to manage than applying a prefix matching route-map to individual network statements. The following example is very typical, on a customer aggregation or edge device:

```
router bgp 65534
 redistribute static route-map static-to-bgp
 !
 route-map static-to-bgp permit 5
  match ip address prefix-list netblock-to-bgp
  set community 65534:1234
  set origin igp
 !
 ip prefix-list netblock-to-bgp permit 220.220.0.0/16 le 30
 !
 ip route 220.220.1.16 255.255.255.224 serial 0/0
```

which basically sets community of 65534:1234 on all the ISP's customer prefixes. There are dangers with this – the prefix list could be deleted, but the ISP has to weigh up this danger with the administrative inconvenience/overhead of setting communities, etc, on a per network statement basis.

**Conclusion: don't use redistribute static unless you fully understand why you are doing it.**

### ***Redistribute <anything> into BGP***

**Conclusion: don't.** ☺

## ***The BGP Path Selection Process***<sup>28</sup>

BGP routers typically receive multiple paths to the same destination. The BGP bestpath algorithm decides which is the best path to actually install in the IP routing table, and therefore the route to use for forwarding traffic. The following text summarises this algorithm.

For the purposes of understanding this algorithm, begin by assuming that all received paths for particular prefix are arranged in a list, similar to the output of *show ip bgp <prefix>* command.

Some paths received by the router will not be considered as candidates for the best path. Such paths typically do not have the *valid* flag in the output of *show ip bgp x.x.x.x*. It is important to understand these cases, before we look at the algorithm itself.

1. Ignore paths marked as "not synchronized" in the output of *show ip bgp x.x.x.x*. If bgp synchronization is enabled – which is the current default in IOS – there must be a match for the prefix in the ip routing table in order for an internal (i.e. iBGP) path to be considered a valid path. Most ISPs will want to disable synchronization using the *no synchronization* bgp subcommand.

---

<sup>28</sup> This section was written by Mark Turner, [markt@cisco.com](mailto:markt@cisco.com), and can also be found on CCO at <http://www.cisco.com/warp/public/459/25.shtml>

2. Ignore paths for which the NEXT\_HOP is inaccessible. This is why it is important to have an IGP route to the NEXT\_HOP associated with the path.
3. Ignore paths from an EBGp neighbor if the local AS appears in the AS path. Such paths are denied upon ingress into the router, and not even installed in the BGP RIB. The same applies to any path denied by routing policy implemented via access, prefix, as-path, or community lists, **unless** “inbound soft reconfiguration” is configured for the neighbor.
4. If **bgp bestpath enforce-first-as** is enabled and the UPDATE does not contain the AS of the neighbor as the first AS number in the AS\_SEQUENCE, send a NOTIFICATION and close the session.
5. Ignore paths marked as (*received-only*) in the output of **show ip bgp x.x.x.x**. This path has been rejected by policy, but has been stored by the router because **soft-reconfiguration inbound** has been configured for the neighbor sending the path.
6. Ignore paths with a next-hop metric marked as *inaccessible*.

## The BGP Best Path Algorithm for IOS

Assign the first valid path as the current best path. Now compare the bestpath with the next path in list, until the end of the list of valid paths is reached.

1. Prefer the path with the largest *weight*. Note that *weight* is a Cisco specific parameter, local to the router on which it is configured.
2. Prefer the path with the largest LOCAL\_PREF.
3. Prefer the path which was locally originated via a *network* or *aggregate* bgp subcommand, or through redistribution from an IGP.
4. Prefer locally sourced *network/redistributed* paths over locally generated aggregates.
5. Prefer the path with the shortest AS path.
  - a) This step is skipped if **bgp bestpath as-path ignore** is configured.
  - b) An AS\_SET counts as one AS, no matter how many ASs are in the set. The AS\_CONFED\_SEQUENCE is not include in the AS path length.
6. Prefer the path with the lowest origin type: IGP is lower than EGP, and EGP is lower than INCOMPLETE.
7. Prefer the path with the lowest MED.
  - a) This comparison is only done if the first (i.e. neighbouring) AS is the same in the two paths; any confederation sub-ASes are ignored. In other words, MEDs are compared only if the first AS in the AS\_SEQUENCE is the same – any preceding AS\_CONFED\_SEQUENCE is ignored.
  - b) If **bgp always-compare-med** is enabled, MEDs are compared for all paths. This knob needs to be enabled over the entire AS, otherwise routing loops could occur.
  - c) If **bgp bestpath med confed** is enabled MEDs are compared for all paths that consist **only** of AS\_CONFED\_SEQUENCE (i.e. paths originated within the local confederation).
  - d) Paths received with no MED are assigned a MED of 0, unless **bgp bestpath missing-is-worst** is enabled, in which case they are effectively considered to have (although not actually assigned) a MED of 4,294,967,295. Any route

Wednesday, June 06, 2001

received from a neighbor with a MED of 4,294,967,295 will have the MED changed to 4,294,967,294 before insertion into the bgp table.

- e) BGP Deterministic MED (see later) can also influence this step.
8. Prefer the External (eBGP) over Internal (iBGP) paths. Note that paths containing AS\_CONFED\_SEQUENCE are local to the confederation, and therefore treated as internal paths. There is no distinction between Confederation External and Confederation Internal.
9. Prefer the path with the lowest IGP metric to the BGP nexthop.
10. If *maximum-paths* *N* is enabled, and there are multiple external/confederation-external paths from the same neighboring AS/sub-AS, then insert up to *N* most recently received paths in the IP routing table. This allows eBGP multipath load sharing. The maximum value of *N* is currently 6; the default value, with the knob disabled, is 1. The oldest received path is marked as the **best** path in the output of *show ip bgp x.x.x.x*, and the equivalent of *next-hop-self* is performed before forwarding this best path on to internal peers.
11. Prefer the path which was received first (i.e. the oldest one).
  - a) This step minimizes route-flap, since a newer path will not displace an older one, even if it would otherwise be selected on account of the additional decision criteria below. It makes more sense to only apply the additional decision steps below to iBGP paths, in order to ensure a consistent bestpath decision within the network, and thereby avoid loops.
  - b) This step is skipped if *bgp bestpath compare-routerid* is enabled.
  - c) This step is skipped if the ROUTER\_ID is the same, since the routers were received from the same router.
  - d) This step is skipped if there is no current best path. An example of losing the current bestpath occurs when the neighbor offering the path goes down.
12. Prefer the route coming from the BGP router with the lowest router ID. The router ID is the highest IP address on the router, with preference given to loopback interfaces, if one or more is configured. It can also be set manually via *bgp router-id x.x.x.x*. Note that if a path contains Route Reflector attributes, the originator ID is substituted for the router ID in the path selection process.
13. If the originator/router ID is the same, prefer the path with the minimum cluster-id length – this will only be present in BGP route-reflector environments, and allows clients to peer with RRs/clients in other clusters. In this scenario the client **must** be aware of the Route Reflector specific BGP attributes.
14. Prefer the path coming from the lowest neighbor address. This is the ip address used in the *bgp neighbor* configuration, and corresponds to the address the remote peer uses in the TCP connection with the local router.

## ***BGP Features and Commands***

BGP is the heart of the Internet. It is the essential protocol that keeps it all glued together. Yet, because the only thing you can guarantee on the Internet is change, BGP needs consistent updates with new features and functionality to help ISPs manage and scale their networks. This section provide early documentation and configuration notes for ISP Network Engineers. It is not a replacement for the CCO documentation (<http://www.cisco.com/univercd/home/home.htm>), yet most of these will be documented here long before they are available on CCO.

We will cover these new features/commands briefly in this document. ISP engineers need to understand when to use these commands on their backbones. Hence, detailed examples on how these commands are used can be found in *Using the Border Gateway Protocol for Interdomain Routing* (<http://www.cisco.com/univercd/cc/td/doc/cisintwk/ics/icsbgp4.htm>)

## Stable iBGP Configuration

An ISP's backbone should be built using an IGP to carry internal infrastructure addressing, and iBGP to carry customer networks and Regional Registry assigned aggregates. There is the obvious distinction in services and function but more importantly, IGP's converge faster than iBGP, and respond to changes in conditions (physical link status) more quickly. This design is deployed by many ISPs.

A common error is to use IP addresses of ethernet or FDDI interfaces as the remote peer addresses. There may be no issue when the remote address is active, but as soon as the interface goes down, the peering is torn down as the address is no longer reachable. Most ISP backbone designs have routers connected by at least two interfaces to the rest of their network. It would be a unfortunate to have the iBGP peering go down if one WAN link had gone down when in fact the router is perfectly accessible via another connection.

The design using two exit paths per router should be taken advantage of. This is done by using the loopback interface, a "real" interface but without any physical connectivity. The interface always exists, and can only be shut down by the IOS configuration command. Choosing the loopback interface and assigning it a host IP address (/32) guarantees a more stable iBGP, no matter the underlying physical network issues which are taken care of by the IGP.

```
hostname gateway1
!
interface loopback 0
 ip address 215.17.1.34 255.255.255.255
!
router bgp 200
 neighbor 215.17.1.35 remote-as 200
 neighbor 215.17.1.35 description iBGP with CR2
 neighbor 215.17.1.35 update-source loopback 0
 neighbor 215.17.1.36 remote-as 200
 neighbor 215.17.1.36 description iBGP with CR3
 neighbor 215.17.1.36 update-source loopback 0
!
```

Routers 215.17.1.35 and 215.17.1.36 see the BGP updates coming from the address 215.17.1.34 of gateway1. Likewise, gateway1 hears the BGP updates coming from 215.17.1.35 and 215.17.1.36 (loopback interfaces configured on those routers). This configuration is independent of the backbone infrastructure, and is not dependent on the physical connectivity of the routers.

The loopback interfaces on an ISP network are usually addressed out of one block of address space, with each loopback address carried around in the ISP's IGP as a /32 address. The reasons for assigning loopback interface addresses out of one block will become apparent later in this appendix.

Notice the use of the "description" command. This is new in 11.1CC and 12.0 software, allowing a description of the BGP peering to be entered into the router configuration.

Note that for eBGP configuration, IP addresses of **real** interfaces are generally used. The reason for this is that there is no IGP running between ISP backbones or ASes. eBGP peering routers have no way of finding out about external networks apart from the other end of a point-to-point WAN link which will be linking the two together.

## BGP Auto Summary

In IOS auto-summarisation will be turned on by default for all prefixes which are redistributed into BGP from other routing protocols. This feature will automatically summarise subprefixes to the classful network boundaries when crossing classful network boundaries. The Internet Registries are now allocating from the former Class A space – an ISP today would more

Wednesday, June 06, 2001

likely be allocated /18 IPv4 address from what used to be the class A space. BGP's default behaviour would be to take that /18 and advertise a /8. Without the BGP command `no auto-summary`, BGP will auto-summarise the /18 into a /8. This will cause at least confusion on the Internet, but worse potentially "attracting" other service providers' unroutable traffic to the local backbone, with due consequences on circuit and systems loading.

Example: An ISP was allocated 24.10.0.0/18. The ISP would sub-allocate this /18 for their customers. The ISP would want to advertise the /18 to the Internet. BGP's default behaviour would be to `auto summarize` the /18 into the *classful* boundary – 24.0.0.0/8 – the old class A. The problem is that other ISPs are also getting /18 allocations from the IPv4 Registry.

In today's classless Internet world where the former class A space is being efficiently sub-allocated, ISP and Enterprise backbones which use BGP and redistribute prefixes from other routing protocols need to use `no auto-summary`.

It should be noted that redistribution into BGP for any service provider backbone is **strongly discouraged!** Therefore providers who are not doing redistribution can safely omit the `no auto-summary` command. However, as with everything else, it is good practice to include the command – it prevents any future accidents from happening.

## BGP Synchronisation

The (historical) default in IOS is for BGP not to advertise a route until all routers within the AS have learned about the route via an IGP. In today's Internet, and certainly since the mid 1990's, ISPs have designed their networks so that the iBGP carries customer and Internet prefixes, and the IGP carries the infrastructure addresses. It is very uncommon for a prefix to appear both because of an IGP and because of iBGP. So synchronisation **MUST** be TURNED OFF for all ISP configurations. To do this, use the BGP configuration command:

```
no synchronization
```

## BGP Community Format

Many ISPs make extensive use of BGP communities for routing policy decision making. A bgp community tutorial is beyond the scope of this document, but ISP engineers should be aware of the two formats supported in IOS. The original format for a community number took the form of a 32-bit integer. More recently, this format was redefined as two 16-bit integers separated by a colon as per the BGP standard. The first 16-bit number is accepted as being the ISP's AS (because community numbers are exportable between ASes with the `send-community` bgp directive). The second 16-bit number is used to represent different policies the ISP wishes to implement. Note that there are some standard communities defined as common or current practice – these are documented in RFC1998.

The configuration command is:

```
ip bgp-community new-format
```

and it converts communities from looking something like 13107210 to a more human friendly 200:10. It should be noted that BGP does not care which format is used internally – the field is still 32 bits in total. This new format is purely for appearance and practicality purposes only.

## BGP Neighbour Shutdown

A new feature introduced into 11.1CC and 12.0 software is the ability to shutdown a BGP peering without actually removing the configuration. Previously the only way to disable a BGP peering was to delete the configuration from the router. This is very disruptive to the router's functioning, and it increases the likelihood of making mistakes when reinstating the configuration at a later stage.

A neighbouring peering is shutdown with the command example:

```
router bgp 200
neighbor 169.223.10.1 shutdown
```

To reinstate the peering once the problem or reason for shutdown has been removed, simply enter the opposite command:

```
router bgp 200
no neighbor 169.223.10.1 shutdown
```

Notice that the command is `no neighbor 169.223.10.1 shutdown` with the **no** coming first rather than the more intuitive `neighbor 169.223.10.1 no shutdown` – be very aware of this as it is possible to be caught out by forgetting or mistyping the **shutdown** portion of the command which can delete the whole BGP configuration for that neighbour.

All users of BGP are encouraged to use neighbour shutdown rather than deleting the configuration – it greatly enhances the ease and reliable operation of the network.

## BGP Dynamic Reconfiguration

There are two methods available now to dynamically reset a BGP peering session without tearing down the entire peering. Normally when an ISP requires to change the policy in a BGP peering, the peering itself has to be torn down so that the new policy can be implemented. For peerings exchanging a number of routes in the Internet, this can be extremely disruptive, putting load on the CPU of both routers involved, and resulting in a routing “flap” through the backbone as the ISP’s network announcements are withdrawn, and then reinstated.

The first and older method is soft-reconfiguration; the second, and strongly recommended method, is route-refresh. It is considered best current practice for ISPs who have external BGP peerings to use dynamic reconfiguration. The impact on their networks, their customers’ perceived service quality, and that of the Internet is too great without this feature. Consider the impact and effects of the following section discussing BGP Route Flap Damping. Also, it is recommended to discourage all operations staff from doing a hard clear of BGP unless it is absolute last resort.

### Soft-Reconfiguration

BGP soft reconfiguration capability was introduced in 11.1CC, 11.2P and 12.0 software. With this feature enabled, once the policy changes have been made, the ISP can simply do a soft reset of the peering without having to tear it down. To support soft-reconfiguration on a peering, the router requires one extra configuration command, for example:

```
router bgp 200
neighbor 215.17.3.1 remote-as 210
neighbor 215.17.3.1 soft-reconfiguration in
neighbor 215.17.3.1 route-map in-filter in
neighbor 215.17.3.1 route-map out-filter out
```

If the policy on the peering requires to be changed, the ISP makes the changes to the route-map configuration in the example above, and then simply issues the command “`clear ip bgp neighbor 215.17.3.1 soft`”. If only the inbound or outbound policy needs to be changed, the clear command can be supplemented with *in* or *out* directives.

Notice the above configuration. Soft-reconfiguration is only required inbound – outbound does not require to be explicitly configured because to make outbound policy configuration changes the BGP process simply has to send an incremental update to its neighbour.

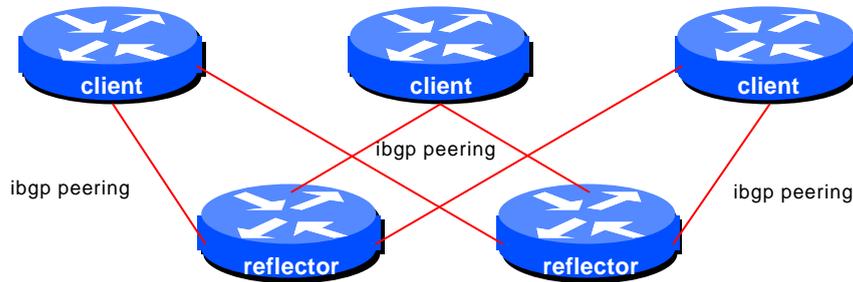
One caveat is that soft-reconfiguration can require a lot more router memory as the router has to store prefixes it has received prior to the BGP inbound policy being implemented. If the inbound policy is complex and/or there are multiple peerings it is possible that almost **twice** the amount of memory will be required by the BGP process than if soft reconfiguration was not configured.

### Route Refresh

A new feature available from IOS 12.0(5)S is route refresh (documented in RFC2918). The concept is similar to soft reconfiguration but is a capability shared between two BGP speakers (as opposed to soft reconfiguration which is configured on the local router only) and is negotiated automatically at the time the BGP session is brought up. To find out if Route Refresh is supported, check the BGP neighbour using the command below:



However, most ISPs choose to implement clusters with two route reflectors as in Figure 33. This gives them redundancy in the cluster in the event of one route reflector failing.



**Figure 33 – BGP Route Reflector Cluster with two RRs**

There are caveats which network designers should be aware of when configuring route reflectors. As soon as a router is configured as a route reflector, it is assigned a cluster identifier automatically by the BGP process. This cluster-id is the BGP router-id, usually the loopback interface address of the router.

The goal of cluster-id is **ONLY** to reduce the propagation of BGP updates between route reflectors when multiple route reflectors are used by the same client. A RR will not accept an update coming from another RR having the same ID (the ID is stored in cluster-list attribute). This is because this update has been received already from the client (since the client peers with all RRs of the cluster). Now, what if the cluster-id in both RRs isn't the same? In that case a client connected to two RRs will send its update to both, and each RR will send this update to the other. The result is that they will receive the same update twice. Now RRs act as BGP routers and when they discover that the update is exactly the same they will just select one (the one with shortest cluster-list) and propagate that one to other neighbours. No routing loops (since we are talking about IDENTICAL UPDATES: same net/mask, same next-hop).

Some ISPs choose to change the default cluster-id to values which let them more easily identify regions/clusters in the backbone. (For example, a cluster-id may match the OSPF process id.) They should only do so with the purpose of the cluster-id kept in mind. Especially:

- Each client **MUST** peer with both route reflectors. Failure to do so may result in routing loops.
- Is the overhead of one extra routing update worth sacrificing for an increased risk in routing blackholes due to network topology?

Note the last point – it seems to be accepted wisdom that the cluster-id should be the same on both route reflectors. This is potentially dangerous and could cause more problems than leaving the status quo. (One example might be the situation where one reflector is only physically reachable from the client via the other reflector. The bestpath could be valid via the cluster-id, but be unreachable.)

## Next-Hop-Self

The BGP command `next-hop-self` simply does as it suggests. The next-hop on BGP announcements from this router are set to the local router-id rather than the IP address of the origin of the prefix. Two common implementation examples are given in the following sections. To configure `next-hop-self`, use the following BGP configuration:

```
router bgp 200
  neighbor 215.17.3.1 remote-as 210
  neighbor 215.17.3.1 next-hop-self
```

## External connections

If a prefix is heard from an external network, the next hop is preserved throughout the IGP. However, setting `next-hop-self` on the border router which is distributing the eBGP prefix into iBGP means that the external prefix will receive the router-id (loopback interface IP address) of the border router rather than the external next hop. This feature is especially used by ISPs at exchange points to ensure consistency and reliability of connections across the exchange.

Wednesday, June 06, 2001

## ***Aggregation routers***

If a prefix is injected into the iBGP at a gateway router (the standard way of injecting a customer prefix into the iBGP), the next hop address is the IP address of the point-to-point link between the gateway aggregation router and the customer. This will mean that the iBGP has a large number of next-hop addresses to resolve from the IGP (not bad in itself) and that the IGP will be larger, resulting in slower convergence and greater potential instability in case of instability or failures in the network.

The obvious solution to this is to use ip unnumbered – the next-hop address in that case will be the router-id. However, many ISPs who started off using IP addresses on point to point links have a lot of work to do to migrate to IP unnumbered on all their customer point-to-point connections. A quicker fix, which will let them scale the IGP but not require undue haste (i.e. panic!) at renumbering large numbers of customer interfaces, is to set next-hop-self on the iBGP peers of the gateway router. The customer prefixes will now appear in the iBGP with the loopback address of the aggregation router, simplifying the route lookups, and more importantly meaning that the IGP no longer has to carry all the point-to-point link addresses used for customer connections. (This also assumes that the point to point link addresses have been assigned in a block per router, and are transported by the iBGP. If the point to point addresses are not in the iBGP, then the customer point-to-point link will not be pingable, which may be an unacceptable situation for providers who monitor the state of their customer links.)

## **BGP Route Flap Damping**

### ***Background***

Route flap damping (introduced in Cisco Internetwork Operating System [Cisco IOS] Release 11.0) is a mechanism for minimising the instability caused by route flapping. A route flap is when a BGP network prefix is withdrawn and reannounced, specifically a BGP speaker hears a WITHDRAW followed by an UPDATE for a prefix. (A peering with an eBGP neighbour being reset does NOT count as a flap.) Whenever a network goes down, the rest of the Internet is told about it. Hence BGP propagates this state change throughout the Internet (unless of course ISPs in the path do something to block this propagation through route filters or similar).

If the state change is happening from faulty circuits (frequently going up and down) or from mis-configured routing (redistributing the IGP into the EGP), the Internet would experience several hundred BGP state changes a second. For every state change, BGP must allocate time to process the work and pass on the changes to all other BGP neighbours. This places unnecessary extra strain on the backbone routers recomputing best paths and propagating these changes. Hence, the tool to control and minimise the effect of route flaps – BGP Damping.

### ***Command Syntax***

The following are the commands used to control route damping:

```
bgp dampening [[route-map map-name] / [half-life-time reuse-value suppress-value maximum-suppress-time]]
```

*half-life-time* – range is 1 – 45 minutes; current default is 15 minutes.

*reuse-value* – range is 1 – 20000; default is 750.

*suppress-value* – range is 1 – 20000; default is 2000.

*max-suppress-time* – maximum duration a route can be suppressed. Range is 1 – 255; default is four times half-life time (60 minutes).

**show ip bgp dampened-routes** – Display all the damped routes with the time remaining to unsuppress. Very useful for find out which sites are having instability problems.

`clear ip bgp dampening [<address> <mask>]` – Clear the damping related information. This will also unsuppress the suppressed routes. Very useful when one of your customers call you about a “unreachable” network that has been suppressed.

A route map can be associated with bgp damping to selectively apply the damping parameters if certain criteria are found. Example selective damping criteria include matching on:

- A specific IP route
- AS Path
- BGP community

Adjusting the damping timers becomes essential when administrators cannot afford to have a long outage for a specific route. BGP damping with route maps is a powerful tool to selectively penalise ill-behaved routes in a user-configurable and controlled manner.

Detailed examples of BGP Damping Techniques with route maps can be found in the book *Internet Routing Architectures 2<sup>nd</sup> Edition*, by Sam Halabi with Danny McPherson.

Recommended route flap damping parameters for use by ISPs were composed into a document by the RIPE Routing Working Group ([routing-wg@ripe.net](mailto:routing-wg@ripe.net)) and are available at <http://www.ripe.net/docs/ripe-210.html>. These values are used by many European and US ISPs, and are based on the operational experienced gained in the industry. (Note that at the time of writing that document is being updated to reflect current features and operational experience.)

## Implementation

For each flap a penalty (1000) is applied to the route. For an attribute change, a penalty of 500 is applied to the route. As soon as the penalty exceeds the ‘suppress-limit’, the advertisement of route will be suppressed. The penalty will be exponentially decayed based on a pre-configured half-life time. Once the penalty decreases below the *reuse-limit*, it will be unsuppressed.

The routes external to an AS learned via iBGP will not be damped. This is to avoid the iBGP peers having higher penalty for routes external to the AS. It also ensures that only directly connected eBGP peers will have route flap damping applied to them, even though the configuration on the router is applied on a global basis.

Any route that flaps at a rate more than the half-life time will be eventually suppressed. The penalty is decayed at a granularity of 5 seconds and the entries are unsuppressed with the granularity of 10 second. Further, when the penalty is less than half of the reuse-limit, the damping information is purged.

The *maximum-suppress-time* is the final parameter which is important for configuring route flap damping. It is the maximum time that a prefix can be suppressed, and the other three parameters must be chosen accordingly. It is also a key parameter in the software implementation for flap damping (see RFC2439 for a detailed discussion). If the half life time is 15 minutes, and maximum suppress time is 60 minutes, the reuse limit is used to compute what the maximum possible penalty can be. All route flap penalties cannot exceed this maximum value. It is most important that the suppress-limit is LESS than the maximum possible value that the route flap penalty can be, otherwise no prefixes will be suppressed. Parameters chosen should be checked to ensure that they achieve the desired effect. (A maximum is set to the value of the penalty so that if a prefix is oscillating frequently the resultant penalty attached to it doesn’t generate a huge number and therefore become unlikely ever to be reused.) The mathematical equation might be:

$$\text{max-penalty} = \text{reuse-limit} \times 2^{\left(\frac{\text{max-suppress-time}}{\text{half-life}}\right)}$$

For the IOS default configuration of `bgp dampening 15 750 2000 60`, this would make the maximum possible penalty 12000. So even if the prefix carries on flapping after the penalty has reached 12000, the penalty will not be increased any more.

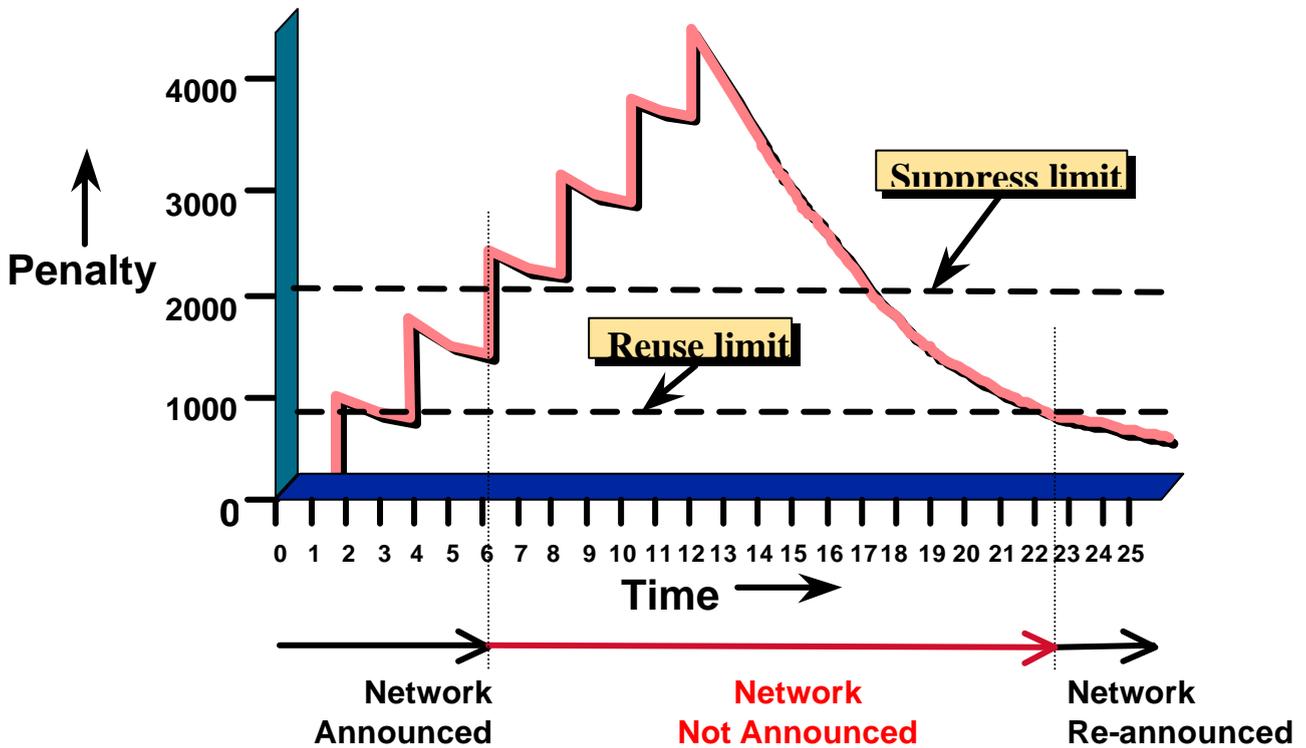


Figure 34 – BGP Route Flap Damping

There now follow some examples of good and bad flap damping parameters:

- **bgp dampening 30 2000 3000 60** – these values are valid. The maximum-suppress-time is 60 minutes, and with a reuse-limit of 2000, the maximum possible penalty is 8000. A suppress-limit of 3000 means that penalties can easily exceed the suppress-limit. (A maximum possible penalty of 8000 decays to 4000 after one half life (30 minutes) and to 2000 after two half lives (60 minutes).)
- **bgp dampening 15 750 3000 45** – these values are valid. The maximum-suppress-time is 45 minutes, and with a reuse-limit of 750, the maximum possible penalty is 6000. The suppress-limit of 3000 is easily reachable.
- **bgp dampening 15 500 2500 30** – these values are illegal and won't give rise to any prefixes being suppressed due to flaps. The maximum-suppress-time is 30 minutes, so to reach the reuse-limit of 500 with a halflife time of 15 minutes will required the penalty to reach 2000. However, the suppress limit has been set to 2500, so no prefixes will be suppressed.
- **bgp dampening 30 750 3000 60** – these values are valid but won't give rise to any prefixes being suppressed due to flaps. The maximum-suppress-time is 60 minutes, so to reach the reuse-limit of 750 with a halflife time of 30 minutes requires a suppress limit of 3000. However, prefixes are only suppressed once the penalty reaches 3000. As the penalty is adjusted every 5 seconds, the prefix will only be suppressed if an update arrives within 5 seconds of the withdrawal. This is extremely unlikely, so no prefixes will be suppressed with this example.

### Designing Flap Damping Parameters

One to note is that prefixes are only suppressed when they are heard from a neighbour and have a flap history and a penalty value which will cause them to be suppressed. A prefix will attract a penalty of 1000 when the WITHDRAW is heard from

the neighbouring eBGP peer. However, because the prefix has been withdrawn, it is not present in the BGP table apart from having a history entry, so will not be damped. The penalty will decay however. When the prefix is reannounced by the neighbour, if the penalty is still above the suppress limit, the prefix will be suppressed. If the penalty is below the suppress limit, the prefix will not be suppressed. It is important to remember this when designing flap damping parameters. The maximum possible penalty should be set considerably larger than the suppress limit otherwise there is the possibility of a slowly oscillating prefix not being suppressed as intended.

For example, **bgp dampening 30 751 3000 60** will generate a maximum possible penalty of 3004, which is only 4 higher than the suppress limit. The fastest the prefix would reappear in the BGP table is one minute later the next time the BGP process runs, and that intervening time could have attracted around 100 points of decay, thereby bringing the penalty below 3000 and not causing any damping to happen. If **bgp dampening 30 800 3000 60** was used, the penalty would still be above 3000, and the prefix would be suppressed as intended.

## BGP Flap Statistics

It is possible to monitor the flaps of all the paths that are flapping. The statistics will be lost once the route is not suppressed and stable for at least one half-life time. The display will look like the following:

```
cerdiwen#sh ip bgp neighbors 171.69.232.56 flap-statistics
BGP table version is 18, local router ID is 172.19.82.53
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          From           Flaps Duration Sup-time Path
*> 5.0.0.0          171.69.232.56   1    0:02:21   300
*> 6.0.0.0          171.69.232.56   2    0:03:21   300
```

The following are the new commands that will display flap statistics:

**show ip bgp flap-statistics** – Displays flap statistics for all the paths.

**show ip bgp flap-statistics regexp <regexp>** – Display flap statistics for all paths that match the regular expression

**show ip bgp flap-statistics filter-list <list>** – Display flap statistics for all paths that pass the filter

**show ip bgp flap-statistics x.x.x.x <m.m.m.m>** – Display flap statistics for a single entry

**show ip bgp flap-statistics x.x.x.x m.m.m.m longer-prefix** – Display flap statistics for more specific entries

**show ip bgp neighbor x.x.x.x flap-statistics** – Display flap statistics for all paths from a neighbor

NOTE: As we maintain information about only one path for a neighbour, the *show ip bgp flap-statistics neighbor* could show a different path for the same NLRI<sup>29</sup>.

The following commands could be used to clear the flap statistics.

**clear ip bgp flap-statistics** – Clear flap statistics for all routes

**clear ip bgp flap-statistics regexp <reg>** – Clear flap statistics for all the paths that match the regular expression

**clear ip bgp flap-statistics filter-list <list>** – Clear flap statistics for all the paths that pass the filter

<sup>29</sup> NLRI stands for Network Layer Reachability Information.

Wednesday, June 06, 2001

```
clear ip bgp flap-statistics x.x.x.x <m.m.m.m> – Clear flap statistics for a single entry
```

```
clear ip bgp x.x.x.x flap-statistics – Clear flap statistic for all paths from a neighbour
```

## BGP Neighbour Authentication

You can invoke MD5 authentication between two BGP peers. This feature must be configured with the same password on both BGP peers otherwise the connection between them will not be made. Invoking authentication causes the Cisco IOS software to generate and check the MD5 digest of every segment sent on the TCP connection. If authentication is invoked and a segment fails authentication, then a message appears on the console.

Configuring a password for a neighbour will cause an existing session to be torn down and a new one established. If you specify a BGP peer group by using the *peer-group-name* argument, all the members of the peer group will inherit the characteristic configured with this command. If a router has a password configured for a neighbour, but the neighbour router does not, a message such as the following will appear on the console while the routers attempt to establish a BGP session between them:

```
%TCP-6-BADAUTH: No MD5 digest from [peer's IP address]:11003 to [local router's IP address]:179
```

Similarly, if the two routers have different passwords configured, a message such as the following will appear on the console:

```
%TCP-6-BADAUTH: Invalid MD5 digest from [peer's IP address]:11004 to [local router's IP address]:179
```

The following example specifies that the router and its BGP peer at 145.2.2.2 invoke MD5 authentication on the TCP connection between them:

```
router bgp 109
 neighbor 145.2.2.2 password v61ne0qkel133&
```

## BGP MED not set

When an MED is not set on a route, Cisco IOS has always assumed that the MED is zero. Some other vendors have assumed that the MED is 4294967295 ( $2^{32}-1$ ). This divergence can result in eBGP routing loops between Cisco and other vendor routers. This confusion was due to the lack of any definition in the BGP standard on what to do if MED was not set.

The most recent IETF decision regarding BGP MED assigns a value of infinity to the missing MED, making the route lacking the MED variable the least preferred. The default behaviour of BGP routers running Cisco IOS software is to treat routes without the MED attribute as having a MED of 0, making the route lacking the MED variable the most preferred. To configure the router to conform to the belated IETF standard, use the command (new in 12.0 and 12.1, but not in 12.0S):

```
router bgp 109
 bgp bestpath missing-as-worst
```

Service Providers should seriously consider whether they need to implement this command given the large deployment of routers in the Internet which are using the Cisco default. Implementing this command without considering the consequences will have just as a serious impact as not being aware of the situation in the first place.

## BGP Deterministic MED

If *bgp deterministic-med* is not enabled, the order in which routes are received may impact MED-based best path decisions. This may occur when the same route is received from multiple AS or confederation sub-AS, with exactly the same path length and different MEDs.

For example, consider the following routes, received in the order shown:

- 1) AS\_PATH 1, MED 100, internal, igp metric to NEXT\_HOP 10
- 2) AS\_PATH 2, MED 150, internal, igp metric to NEXT\_HOP 5
- 3) AS\_PATH 1, MED 200, external

Received in this order, a bgp router **without** deterministic MED configured would choose path 2) over path 1), due to a lower IGP metric to reach the NEXT\_HOP (see step 10 below). It would then prefer path 3) to path 2) because it is external. However, path 3) has a higher MED than path 1).

Enabling *bgp deterministic-med* removes any temporal dependency of MED-based best path decisions. It ensures that correct MED comparison is made across all routes received from the same AS, and that path 1) is chosen as the best path in the above example.

Cisco strongly recommends enabling *bgp deterministic-med* in all new network deployments.

Note that if *bgp always-compare-med* is enabled, BGP MED decisions will always be deterministic.

## Comparing Router-IDs

As part of the standard path selection process in IOS, the router does not switch between two eBGP paths based solely upon the router-id. Consider a situation where a router hears two announcements of a particular prefix from two different neighbours. If the path selection process determines that all attributes are identical apart from the router-id, it currently will take the path with the oldest entry in the routing table, not the entry with the lowest router-id. This choice was made due to reasons of maintaining stability in operational backbones over the last decade. However, it is actually counter to RFC1771 which states that the router-id should always be compared.

A new BGP option has been added (fully documented in CSCdr47086) from IOS 12.0(11)S and 12.1(3) which makes the path selection process compliant with the RFC. It is:

```
bgp bestpath compare-routerid
```

and this will tell the bgp path selection process to always compare router-id's at the appropriate stage. The IOS default remains unchanged.

## BGP Network Statement

There was a historical limitation to the number of network statements which could be applied to the BGP configuration. In the early days of IOS, where the average router had at most 30 interfaces, the limit of 200 network statements seemed a large number compared with the number of networks such a router could originate. However, in the last 5 years or so, routers have grown through having hundreds of interfaces, to the thousands of interfaces we see on some of the platforms today. Obviously 200 network statements is a severe limitation (and unfortunately has resulted in many ISPs heading down the path of redistributing prefixes from other routing protocols into BGP).

CSCdj57631 removed the 200 network statement limit for BGP (as well as OSPF, RIP and EIGR) in late 1997, in 11.1CC and 11.3 software.

## Removing Private ASes

Some ISPs use private ASes within their network (typically but not exclusively for customers who multihomed onto their backbone). There is a BGP option (CSCdi64489) which prevents any private ASes from being leaked to the Internet:

```
router bgp 109
 neighbor 145.2.2.2 remove-private-AS
```

Wednesday, June 06, 2001

If the BGP update has private ASes<sup>30</sup> in the AS path, this option will remove the private AS numbers. Note that it will not work if there are private and public ASes in the AS path – so this option cannot be used in the case where a network is using a private AS for its BGP, yet providing transit to the public Internet (this is considered a configuration error). Note that in the case of BGP confederations, the private AS will be removed so long as the private AS appears after (and outside) the confederation portion of the AS path. It should be noted that this command can only be used for eBGP peers – it has no effect for iBGP peers.

Prior to the existence of this command, most ISPs used a combination of as-path access-lists and proxy announcements to remove private ASes. That process was clumsier, and often prone to error. For this reason, remove-private-AS is strongly recommended to remove private ASes from the public Internet. For reference and comparison purposes, the old method is:

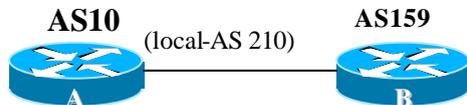
```
router bgp 109
  network 220.10.0.0 mask 255.255.224.0
  neighbor 145.2.2.2 filter-list 5 out
!
ip as-path access-list 5 deny ^(65534_)+$
!
ip route 220.10.0.0 255.255.224.0 null0 250
```

This configuration has the same effect as using one line remove-private-AS for the AS65534 peer, but is more prone to error through requiring a filter list and a static pull up route to ensure that the /19 prefix appears in the routing table.

## BGP local-as

### Configuration

A recent addition to BGP is the **local-as** neighbour option, which allows the AS number of the network to be changed for eBGP peerings. It isn't possible in IOS to configure BGP to run in more than one AS – the local AS feature provides a solution to this requirement.



Router A is in AS 109, and Router B is in AS 159. However, when A peers with B it uses AS 210 as it's AS number. As far as B is concerned it is peering with a router in AS 210. Operationally this is equivalent to Router B peering with a router in AS 210, and that router peering with Router A in AS 109. The AS path for all prefixes learned from Router B as seen on Router A would be 210\_159 – AS 210 is inserted into the AS path sequence. The AS path for all prefixes learned from Router A by Router B would be 210\_109.

#### Router A configuration:

```
router bgp 109
  neighbor 145.2.2.2 remote-as 159
  neighbor 145.2.2.2 local-as 210
```

#### Router B configuration:

```
router bgp 159
  neighbor 144.2.2.1 remote-as 210
```

**Local-as** can be configured per eBGP peer, or per peer-group. It cannot be configured for individual members of a peer-group. Also, local-as cannot be configured with the AS number of the BGP process of the local router, nor can it be configured with the AS number of the BGP process on the remote router. **Local-as** cannot be used between two confederation eBGP peers – they must be true eBGP peers.

<sup>30</sup> Private ASes are those AS numbers between 64512 and 65534. AS65535 is reserved by IANA.

## Motivation

The local-as feature is most often used and was requested by ISPs who are active in acquiring other SPs. If an ISP purchases another ISP, there is a significant amount of work which goes into renumbering the acquired ISP's backbone into the same AS number as that of the purchaser. However, there are situations where the acquired ISP may have agreements with peers, or a specific eBGP configuration which is hard to migrate to the new AS (could be political or technical reasons, or both). The local-as option can make the peering router in AS109 look as though it really is in AS210, the ISP network which was purchased, until the political issues with the inter-provider peering can be sorted out.

## BGP Neighbour Changes

It is possible to log neighbour state changes to a Unix syslog server. This is extremely useful for most syslog based monitoring systems as it gives early warning of problems with iBGP peers, and more especially external BGP neighbours. The logging is enabled by:

```
router bgp 109
  bgp log-neighbor-changes
```

Note that as from IOS 12.1(4) and 12.0(12)ST, the default action is to log all BGP neighbour changes (see DDTS CSCdm59903). At time of writing, it was not yet included in 12.0S.

Also, note that as from CSCdr54231, BGP notification messages are now included when **bgp log-neighbor-changes** is configured. Prior to CSCdr54231, the only way of seeing these messages was to enable **debug ip bgp** which was a bit limiting for most service providers! This alteration has been included in IOS Releases from 12.0(10)S3, 12.1(3), and 11.1(34)CC.

## Limiting the Number of Prefixes from a Neighbour

There have been times, either via configuration error or a blatant attack on the Internet, that the Global *Default Free* Routing Table jumped to 2 to 3 times its size. This has caused severe problems on sections of the Internet. The BGP neighbour command `maximum-prefix` was added to help networks safe guard against these sorts of problems.

This command allows you to configure a maximum number of prefixes a BGP router is allowed to receive from a peer. It adds another mechanism (in addition to distribute lists, filter lists, and route maps) to control prefixes received from a peer. When the number of received prefixes exceeds the *maximum* number configured, the router terminates the peering (by default). However, if the keyword **warning-only** is configured, the router instead only sends a log message, but continues peering with the sender. If the peer is terminated, the peer stays down until the **clear ip bgp** command is issued.

In the following example, the maximum number of prefixes allowed from the neighbour at 129.140.6.6 is set to 150000 (the Global Internet Route Table was around 106000 at the time of writing):

```
router bgp 109
  network 131.108.0.0
  neighbor 129.140.6.6 maximum-prefix 150000
```

The `maximum-prefix` command sends log messages also, so any overrun can be trapped by a management system which monitors the router's syslog output. One message is sent when the number of prefixes received reaches the configured threshold value:

```
%BGP-4-MAXPFX: No. of unicast prefix received from 129.140.6.6 reaches 113021, max 150000
```

The default threshold is 75%. This can be changed by specifying the threshold percentage in the `maximum-prefix` line – the following example sets the threshold to 95%:

```
router bgp 109
  bgp log-neighbor-changes
  neighbor 129.140.6.6 maximum-prefix 120000 95
```

Wednesday, June 06, 2001

Another message is sent when the number of prefixes received exceed the maximum number of prefixes configured. Logging of neighbour changes is included for completeness in this example:

```
%BGP-3-MAXPFXEXCEED: No. of unicast prefix received from 129.140.6.6: 123411 exceed limit 120000
%BGP-5-ADJCHANGE: neighbor 129.140.6.6 Down - BGP Notification Sent
```

## BGP Fast External Fallover

By default if a BGP peer doesn't respond within a few seconds, the peering relationship will be reset. By adding the *no bgp fast-external-fallover* configuration, the peering will be held open for considerably longer. This configuration is desirable, if not essential, in the case of long distance peering links, or unreliable or long latency connections to other AS's, and in the case where ISPs prefer stability over convergence speed in large networks.

```
router bgp 109
no bgp fast-external-fallover
```

Important note: this configuration option should be used with care. It is recommended that fast-external-fallover is only used for links to an ISP's upstream provider, or over unreliable links. Because the fallover is slower, it is possible to blackhole routes for up to 3 minutes. This may prove problematic on, for example, a link to a multihomed customer, where their peering may have suffered an unintentional reset due to human activity.

## BGP Peer-group<sup>31</sup>

### *Background*

The major benefits of BGP peer-groups are the reduction of resource (CPU load and memory) required in update generation. It also simplifies BGP configuration.

With BGP peer-groups, the routing table is walked only once per peer-group and updates are replicated to all other peer-group members that are in sync. Depending on the number of members, the number of prefixes in the table and the number of prefixes advertised, this could significantly reduce the load. It is thus highly recommended that peers with identical outbound announcement policies be grouped into peer-groups.

### *Requirements*

All members of a peer-group must share identical outbound announcement policies (e.g., distribute-list, filter-list, and route-map), except for the originating default which is handled on a per-peer basis even for peer-group members.

The inbound update policy can be customised for each individual member of a peer-group.

A peer-group must be either internal (with iBGP members) or external (with eBGP members). Members of an external peer-group have different AS numbers.

### *Historical Limitations*

There used to be several limitations with BGP peer-groups:

- If used for clients of a route reflector, all the clients should be fully meshed.
- If used as eBGP peer-group, transit can not be provided among the peer-group members.
- All the eBGP peer-group members should be from the same subnet to avoid non-connected nexthop announcements.

---

<sup>31</sup> Thanks to Enke Chen for providing most of the text describing BGP Peer Groups.

Inconsistent routing would occur if these limitations were not followed. They have been removed starting with the following IOS versions: 11.1(18)CC, 11.3(4), and 12.0. Only the router on which the peer-groups are defined needs to be upgraded to the new code.

### *Typical Peer-group Usage*

ISP Network Engineers group BGP peers on a router into peer-groups based on their outbound update policies. A list of peer-groups commonly by ISPs are listed as follows:

- Normal iBGP peer-group: for normal iBGP peers.
- iBGP Client peer-group: for reflection peers on a route reflector.
- eBGP Full-routes: for peers to receive full Internet routes.
- eBGP customer-routes: for peers to receive routes from direct customers of the ISP only. Some members can be configured with “default-origination” to receive the default route as well as the customer routes.
- eBGP default-routes: for peers to receive the default route, and possibly along with a few other routes.

### *BGP Peer-Group Examples*

This example shows an iBGP Peer-Group for a router inside an ISP’s backbone:

```
router bgp 109
neighbor internal peer-group
neighbor internal remote-as 109
neighbor internal update-source loopback 0
neighbor internal send-community
neighbor internal route-map send-domestic out
neighbor internal filter-list 1 out
neighbor 131.108.10.1 peer-group internal
neighbor 131.108.20.1 peer-group internal
neighbor 131.108.30.1 peer-group internal
neighbor 131.108.30.1 filter-list 3 in
```

This example shows an eBGP Peer-Group for a router peering with several ISPs all with the same advertisement policies:

```
router bgp 109
neighbor external-peer peer-group
neighbor external send-community
neighbor external-peer route-map set-metric out
neighbor external-peer route-map filter-peer in
neighbor 160.89.1.2 remote-as 200
neighbor 160.89.1.2 peer-group external-peer
neighbor 160.89.1.4 remote-as 300
neighbor 160.89.1.4 peer-group external-peer
```

## Using Prefix-list in Route Filtering<sup>32</sup>

### *Introduction*

The prefix-list feature offers significant performance improvement (in terms of CPU consumed) over the access-list in route filtering of routing protocols. It also provides for faster loading of large lists, and support for incremental configuration. In addition, the command line interface is much more intuitive. This feature is available in IOS versions from 11.1 (17)CC, 11.3(3) and 12.0.

According to one ISP who has done some in depth performance testing of prefix-lists and access-lists, a 7507/RSP4 running 11.1(20)CC took over 15 minutes to boot using extended ACLs for filtering BGP routes. The same test with prefix-lists had the router booting and fully operational in under 5 minutes. The configuration was around 95000 lines of

<sup>32</sup> The core of this section is by Bruce R. Babcock [[bbabcock@cisco.com](mailto:bbabcock@cisco.com)] and Enke Chen

Wednesday, June 06, 2001

ACL (or prefix-list) total for all neighbours, a configuration around 6Mbytes in total size. Given this experience it is easy to see why most ISPs are now using prefix-lists in favour of ACLs for prefix filtering on their BGP peering sessions.

The prefix-list preserves several key features of access-list:

- Configuration of either “permit” or “deny”.
- Order dependency – first match wins.
- Filtering on prefix length – both exact match and range match.

However, prefix-lists, or prefix-lists in route-maps, do not support packet filtering. This document presents the detailed configuration commands and several applications of the prefix-list in route filtering.

## ***Configuration Commands***

There are three configuration commands related to the prefix-list.

```
no ip prefix-list <list-name>
```

where <list-name> is the string identifier of a prefix-list. This command can be used to delete a prefix-list.

```
[no] ip prefix-list <list-name> description <text>
```

This command can be used to add/delete a text description for a prefix-list.

```
[no] ip prefix-list <list-name> [seq <seq-value>] deny|permit \ <network>/<len> [ge <ge-value>]  
[le <le-value>]
```

This command can be used to configure or delete an entry of a prefix-list.

## ***Command Attributes***

<list-name>: Mandatory. A string identifier of a prefix-list.

seq <seq-value>: Optional. It can be used to specify the sequence number of an entry of a prefix list. By default, the entries of a prefix list would have sequence values of 5, 10, 15 and so on. In the absence of a specified sequence value, the entry would be assigned with a sequence number of (Current\_Max+ 5).

**Like an access-list, a prefix-list is an ordered list. The number is significant when a given prefix matches multiple entries of a prefix list, the one with the smallest sequence number is considered as the real match.**

deny|permit: Mandatory. An action taken once a match is found.

<network>/<len>: Mandatory. The prefix (i.e., network and prefix length). Multiple policies (exact match or range match) with different sequence numbers can be configured for the same <network>/<len>.

ge <ge-value>: Optional.

le <le-value>: Optional.

Both “ge” and “le” are optional. They can be used to specify the range of the prefix length to be matched for prefixes that are more specific than <network>/<len>. Exact match is assumed when neither “ge” nor “le” is specified. The range is assumed to be from “ge-value” to 32 if only the “ge” attribute is specified. And the range is assumed to be from “len” to “le-value” if only the “le” attribute is specified.

A specified <ge-value> and/or <le-value> must satisfy the following condition:

```
len < ge-value < le-value <= 32
```

## Configuration Examples

### Specification of Exact Prefixes

Deny the default route 0.0.0.0/0	<code>ip prefix-list abc deny 0.0.0.0/0</code>
Permit the prefix 35.0.0.0/8	<code>ip prefix-list abc permit 35.0.0.0/8</code>

### Specification of Group of Prefixes

In 192/8, accept up to /24	<code>ip prefix-list abc permit 192.0.0.0/8 le 24</code>
In 192/8, deny /25+	<code>ip prefix-list abc deny 192.0.0.0/8 ge 25</code>
In all address space, permit /8 – /24	<code>ip prefix-list abc permit 0.0.0.0/0 ge 8 le 24</code>
In all address space, deny /25+	<code>ip prefix-list abc deny 0.0.0.0/0 ge 25</code>
In 10/8, deny all	<code>ip prefix-list abc deny 10.0.0.0/8 le 32</code>
In 204.70.1/24, deny /25+	<code>ip prefix-list abc deny 204.70.1.0/24 ge 25</code>
Permit all	<code>ip prefix-list abc permit 0.0.0.0/0 le 32</code>

## Incremental Configuration

A prefix-list can be re-configured incrementally; that is, an entry can be deleted or added individually. For example, to change a prefix-list from the initial configuration to a new configuration, only the difference between the two needs to be deployed as follows:

The initial configuration:

```
ip prefix-list abc deny 0.0.0.0/0 le 7
ip prefix-list abc deny 0.0.0.0/0 ge 25
ip prefix-list abc permit 35.0.0.0/8
ip prefix-list abc permit 204.70.0.0/15
```

The new configuration:

```
ip prefix-list abc deny 0.0.0.0/0 le 7
ip prefix-list abc deny 0.0.0.0/0 ge 25
ip prefix-list abc permit 35.0.0.0/8
ip prefix-list abc permit 198.0.0.0/8
```

The difference between the two configurations:

```
no ip prefix-list abc permit 204.70.0.0/15
ip prefix-list abc permit 198.0.0.0/8
```

## Special Note on the Sequence number

The sequence number is used internally to identify the “real” match (the one with the lowest sequence number) when multiple prefix-list entries match a given prefix. It can also be used to insert an entry to a specific relative position (e.g., sequence number of 7). However, in most cases a prefix-list can be structured such that there is no need to specify

Wednesday, June 06, 2001

sequence numbers, and such an approach would make it easier to automate prefix-list generation, “diff” generation, and deployment. The sequence numbers can be “switched off” from appearing in the configuration by the command:

```
no ip prefix-list sequence-number
```

## *How Does Match Work*

The matching is similar to that of the access-list. More specifically:

- An empty prefix-list would permit all prefixes.
- An implicit deny is assumed if a given prefix does not match any entries of a prefix-list.
- When multiple entries of a prefix-list match a given prefix, the one with the smallest sequence is considered as the “real” match. In short, the first match wins!

Here is an example to illustrate the “first match rule”. Supposed a prefix-list is configured as follows:

```
ip prefix-list abc deny 10.0.0.0/8 le 32
ip prefix-list abc permit 0.0.0.0/0 le 32
```

Then, a given prefix 10.1.0.0/16 would match both entries. However, the prefix will be “denied”, as the first entry is the real match.

## *Show and Clear Commands*

show ip prefix-list [detail|summary] – Displays information of all prefix-lists.

show ip prefix-list [detail|summary] [<name>] – Displays information of a prefix-list.

show ip prefix-list <name> [seq <seq-num>] – Display the prefix-list entry with the given sequence number

show ip prefix-list <name> <network>/<len> – Displays the policy associated with the node <network>/<len>

show ip prefix-list <name> <network>/<len> longer – Displays all entries of a prefix list that are more specific than the given <network>/<len>

show ip prefix-list <name> <network>/<len> first-match – Displays the entry of a prefix list that matches the given <network>/<len>

clear ip prefix-list [<name>] [<network>/<len>] – Resets the “hit count” of prefix-list entries

## *Using Prefix-list with BGP*

The prefix-list can be used as an alternative to the BGP neighbor x.x.x.x distribute-list command. The configuration of prefix-list and distribute-list for a BGP peer are mutually exclusive.

```
router bgp xxx
neighbor x.x.x.x prefix-list <name> in|out
```

## *Using Prefix-list in Route-map*

The prefix-list can be used as an alternative to access-lists used in the command match ip address|next-hop|route-source <access-list> of a route-map. The configuration of prefix-lists and access-lists are mutually exclusive within the same sequence of a route-map.

```
route-map <name> permit|deny <seq-num>
match ip address|next-hop|route-source prefix-list <name> [<name> ...]
```

Besides its application in BGP, route-maps using prefix-lists can be used for route filtering, default-origination, and redistribution in other routing protocols as well. For example, the following configuration can be used to conditionally originate a default route (0.0.0.0/0) when there exists a prefix 10.1.1.0/24 in the routing table:

```
ip prefix-list cond permit 10.1.1.0/24
!
route-map default-condition permit 10
  match ip address prefix-list cond
!
router rip
  default-information originate route-map default-condition
```

### *Using Prefix-list in Other Routing Protocols*

The prefix-list can be used to filter inbound and outbound routing updates, as well as to control route redistribution between different routing protocols. Compared with using the access-list, prefix-list based filtering offers the ability of prefix length filtering.

As usual, access-list and prefix-list are mutually exclusive in one “distribute-list” command.

### **Filtering on inbound updates**

Inbound updates can be filtering on the prefix, or the gateway or both prefix and gateway:

```
router rip | igmp | eigrp
distribute-list {prefix <name1>} | {gateway <name2>} | {prefix <name1> gateway <name2>} in
 [<interface>]
```

Where <names> is the name of a prefix-list to be applied to the prefix being updated, and <name2> the name of a prefix-list to be applied to the gateway (i.e., next-hop) of a prefix being updated. The filtering can also be specified with a specific interface.

### **Filtering on outbound updates**

```
router rip | igmp | eigrp ...
distribute-list prefix <name1> out [<routing_process> | <interface>]
```

### **Example**

In the following configuration, the RIP process will only accept prefixes with prefix length of /8 to /24:

```
router rip
  version 2
  network x.x.x.x
  distribute-list prefix max24 in
!
ip prefix-list max24 seq 5 permit 0.0.0.0/0 ge 8 le 24
!
```

Also, the following configuration will make RIP accept routing update only from 192.1.1.1, besides the filtering on prefix length:

```
router rip
  distribute-list prefix max24 gateway allowlist in
!
ip prefix-list allowlist seq 5 permit 192.1.1.1/32
!
```

Wednesday, June 06, 2001

## BGP filter processing order

When constructing BGP filters, most service providers use prefix-lists or distribute-lists, as-path filters and route-maps to implement their filtering policies. It is important to be aware that these lists are processed in a particular order – newer engineers are sometimes caught out by the standard IOS sequence.

In the example:

```
router bgp 10
 neighbor 220.0.0.1 remote-as 11
 neighbor 220.0.0.1 prefix-list rfc1918-in in
 neighbor 220.0.0.1 prefix-list rfc1918-out out
 neighbor 220.0.0.1 route-map in-peer in
 neighbor 220.0.0.1 route-map out-peer out
 neighbor 220.0.0.1 filter-list 5 in
 neighbor 220.0.0.1 filter-list 6 out
```

the IOS configuration generator (NVGEN) has pretty printed the configuration into the above order. Implementation order is different though and is given here:

```
router bgp 10
 neighbor 220.0.0.1 remote-as 11
 !
 ! actual implementation order
 !
 neighbor 220.0.0.1 route-map in-peer in
 neighbor 220.0.0.1 filter-list 5 in
 neighbor 220.0.0.1 prefix-list rfc1918-in in
 !
 neighbor 220.0.0.1 prefix-list rfc1918-out out
 neighbor 220.0.0.1 filter-list 6 out
 neighbor 220.0.0.1 route-map out-peer out
 !
```

For inbound prefix announcements, the route-map is applied first, followed by the filter-list, finally by the prefix-list (or distribute-list – remember that prefix-list and distribute-list are mutually exclusive). For outbound prefix announcements, the order is reversed, with prefix-lists implemented first, followed by filter-lists, finally by the route-map.

## BGP Conditional Advertisement

Conditional advertisement of prefixes has been introduced into 11.1CC, 11.2, 12.0 and more recent versions of IOS in an effort to contribute to the stability of large BGP based networks (specifically the Internet). Conditional advertisement is usually configured when an AS has at least two connections to another AS. The inter-AS peering routers “watch” the links between the ASes – if one link fails, prefixes are advertised out of the other link. This allows ISPs to set up efficient and effective multihoming without leaking many subprefixes to the Internet, yet retain a good degree of backup in the case of uplink failure.

The configuration command is this:

```
router bgp 109
 neighbor 129.140.6.6 remote-as 159
 neighbor 129.140.6.6 advertise-map announce non-exist-map monitor
```

The **non-exist-map** describes the prefix which will be monitored by the BGP router. The **advertise-map** statement describes the prefix which will be advertised when the prefix in non-exist-map has disappeared from the BGP table. The route-maps **announce** and **monitor** are standard IOS route-maps and are used to configure the prefixes which will be part of the conditional advertisement process.



Wednesday, June 06, 2001

```
!  
route-map ISP2-backbone permit 10  
  match ip address 2                               ! ISP2-backbone-prefix  
!  
access-list 1 permit 202.9.64.0 0.0.1.255         ! ISP2-subblock-prefix  
access-list 2 permit 140.222.0.0 0.0.255.255     ! ISP2-backbone-prefix  
!
```

Access-list 1 is the prefix which will be announced during failure mode. Access-list 2 describes the prefix which is monitored to detect whether a failure has happened.

Router4 can, ofcourse, be configured with a similar conditional advertisement statement, monitoring a prefix announced by ISP1 towards Router2.

**Summary:** conditional advertisements ensure that a multihomed AS will have good backup/failure modes without having to leak subprefixes into the Internet routing table unnecessarily. Only in the case of link failure will a sub prefix be leaked – consider the effects of route flap damping using conditional advertisements, as opposed to using standard BGP backup configuration.

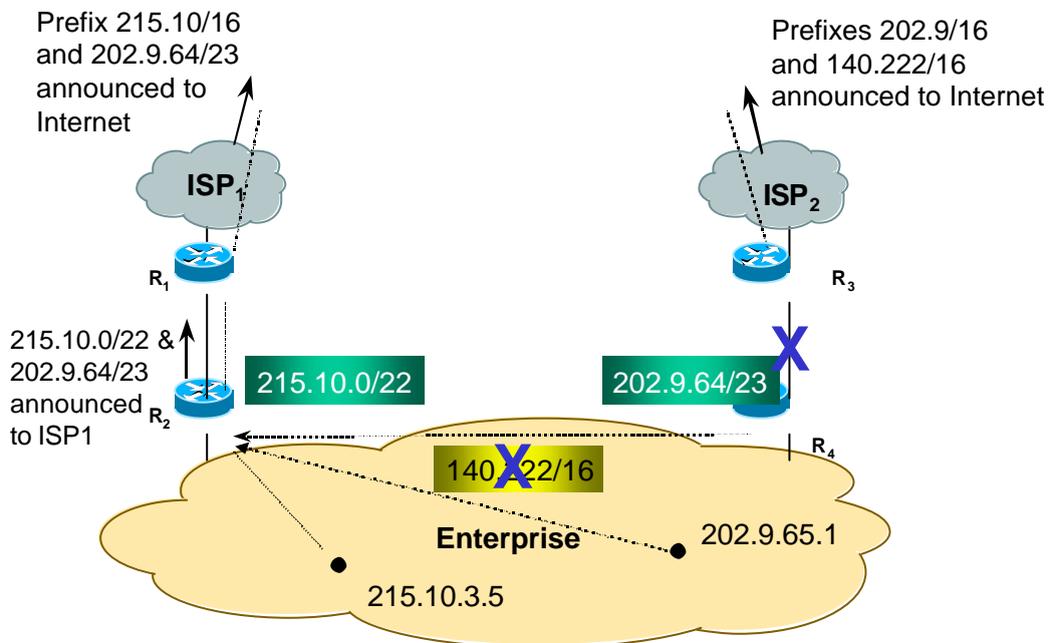


Figure 36 – BGP Conditional Advertisement – Failure Mode

## BGP Outbound Route Filter Capability

This new feature, supported from IOS 12.0(5)S onwards, allows one BGP speaker to install its inbound locally configured prefix-list filter on to the remote BGP speaking router. This is especially used for reducing the amount of unwanted routing update from the remote peer.

The remote BGP speaker would apply the received prefix-list filter, in addition to its locally configured outbound filters (if any), to constrain/filter its outbound routing updates to the neighbor. This mechanism can be used to avoid unwanted routing updates and thus help reduce resources required for routing update generation and processing.

For example, Prefix-List ORF can be used to address the issue of receiving (“unwanted”) full routes from multihomed BGP customers. The customer can simply enable this feature on their router and thus allow their providers to manage the filtering of their route announcements. This avoids unwanted routing updates coming from the customer to their upstream ISP.

Currently the Prefix-List ORF is implemented for IPv4 unicast only. Some points to note about the implementation:

- By default, the Prefix-List ORF Capability is not advertised to any neighbours
- The capability can not be advertised to a neighbour that is a peergroup member.
- The Prefix-List ORF is pushed over to the peer router immediately after the session is established if the local router has received the ORF capability, and has configured inbound prefix-list filter for the neighbour.

## Configuration

The router configuration command is included in the following example:

```
router bgp Y
 neighbor x.x.x.x remote-as Z
 neighbor x.x.x.x description Peer router R2
 neighbor x.x.x.x capability prefix-filter
 neighbor x.x.x.x prefix-list FilterZ-in in
```

This command can be used to enable the advertisement of the Prefix-List ORF Capability to a neighbour. Using the “no neighbor x.x.x.x capability prefix-filter” command disables the Prefix-List ORF Capability.

When the BGP peering is established in this example, the above router (R1) will push its prefix-list “FilterZ-in” over to its peer router x.x.x.x (R2). R2 will receive the prefix-list filter and apply it to its outbound update to R1 (in addition to its local policy, if any is configured).

## Pushing out a Prefix-list ORF

The command to push out a Prefix-list ORF and receive route refresh from a neighbour is:

```
clear ip bgp x.x.x.x in prefix-filter
```

When the inbound prefix-list changes (or is removed), this command can be used to push out the new prefix-list, and consequently receive route refresh from the neighbour based on the new prefix-list. The keyword “prefix-filter” will be ignored if the Prefix-list ORF Capability has not been received from the neighbour.

Without the keyword “prefix-filter”, the command:

```
clear ip bgp x.x.x.x in
```

would simply perform the normal route refresh from the neighbour. It does not push out the current inbound prefix-list filter to the neighbour. The command is useful when inbound routing policies other than the prefix-list filter such as route-map changes.

## Displaying Prefix-list ORF

The command to display the prefix-list ORF received from a neighbour is:

```
show ip bgp neighbor x.x.x.x received prefix-filter
```

This will display the received prefix-list. Changes to the output of “show ip bgp neighbor x.x.x.x” are:

```
Prefixlist ORF
 Capability advertised; received
 Filter sent; received (25 entries)
```

Wednesday, June 06, 2001

## BGP Policy Accounting

### Overview

BGP Policy Accounting allows you to account for IP traffic differentially by assigning counters based on community-list, AS number, and/or AS-path on a per input interfaces basis.

Using BGP Policy Accounting, you can account for traffic (and apply billing) according to the route specific traffic traverses. This way, eg. domestic, international, terrestrial, satellite, etc. traffic can all be identified and accounted for on a per customer basis.

This feature takes advantage of BGP's table-map ability to classify the prefixes that it puts into the routing table according to community-lists, AS-path, AS number, etc. Based on those match criteria this feature will set a bucket number (currently 1 to 8) of an accounting table that is associated with each interface. Each bucket thus represents a traffic classification.

This allows IP traffic to be accounted differentially by community-list, AS number, AS-path per input interface.

### Configuration

Specify communities into community-lists (or define AS-path lists, etc) which will classify traffic for accounting.

```
ip community-list 30 permit 100:190
ip community-list 40 permit 100:198
ip community-list 50 permit 100:197
ip community-list 60 permit 100:296
ip community-list 70 permit 100:201
!
```

Define a route-map to match community-lists and set appropriate bucket number:

```
route-map set_bucket permit 10
match community 30
set traffic-index 2           ! <-- look here
!
route-map set_bucket permit 20
match community 40
set traffic-index 3
!
route-map set_bucket permit 30
match community 50
set traffic-index 4
!
route-map set_bucket permit 40
match community 60
set traffic-index 5
!
route-map set_bucket permit 50
match community 70
set traffic-index 6
```

Modify the bucket number when the IP routing table is updated with BGP learned routes:

```
router bgp 110
table-map set_bucket           ! <-- look here
network 15.1.1.0 mask 255.255.255.0
neighbor 14.1.1.1 remote-as 100
!

ip classless
ip bgp-community new-format
```

Enable the policy accounting feature on the input-interface connected to customer:

```

interface POS7/0
ip address 15.1.1.2 255.255.255.252
no ip directed-broadcast
bgp-policy accounting          ! <-- Look here
no keepalive
crc 32
clock source internal

```

Each customer is off of an input interface (this works on software interface descriptor blocks, so can be subinterfaces) which will have the above displayed table of counters associated with it.

### *Displaying BGP Policy Accounting status*

To inspect which prefix is assigned which bucket and which communit[y/ies]

```

Router#sh ip cef 196.240.5.0 detail
196.240.5.0/24, version 21, cached adjacency to POS7/2
0 packets, 0 bytes, traffic_index 4          ! <-- Look Here
  via 14.1.1.1, 0 dependencies, recursive
    next hop 14.1.1.1, POS7/2 via 14.1.1.0/30
    valid cached adjacency

Router#sh ip bgp 196.240.5.0
BGP routing table entry for 196.240.5.0/24, version 2
Paths: (1 available, best #1)
  Not advertised to any peer
  100
    14.1.1.1 from 14.1.1.1 (32.32.32.32)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Community: 100:197          ! <-- Look Here

```

To look at traffic statistics per-interface:

```

LC-Slot7#sh cef interface traffic-statistics
:
POS7/0 is up (if_number 8)
Bucket      Packets      Bytes
-----
1           0             0
2           0             0
3           50            5000
4          100           10000
5          100           10000
6           10            1000
7           0             0
8           0             0

```

### *Displaying BGP Policy Accounting statistics*

The statistics are stored in a table of packet/byte counters per input software interface (with the assumption that each customer is connected to an input software interface). You can display them by **show cef interface <int> policy-stat**. SNMP support will be added soon.

The statistics are actually displayed per configured table-map match category. Using a route-map you can match against configured community-lists, AS-paths, etc and set to correspond to a specific bucket/index in the above mentioned table.

---

---

## FURTHER STUDY AND TECHNICAL REFERENCES

---

---

This section lists the technical references and further reading which are pertinent to this document. The reader is encouraged to review the references quoted. The references from each section of Cisco ISP Essentials have been listed separately.

### *Software and Router Management*

#### **Cisco Product Bulletins**

<http://www.cisco.com/warp/public/cc/general/bulletin/index.shtml>

#### **Cisco IOS Software Release 12.0S New Features**

[http://www.cisco.com/warp/public/cc/pd/iosw/iore/iomjre12/prodlit/934\\_pb.htm](http://www.cisco.com/warp/public/cc/pd/iosw/iore/iomjre12/prodlit/934_pb.htm)

#### **Cisco IOS Software Release 12.0S Ordering Procedures and Platform Hardware Support**

[http://www.cisco.com/warp/public/cc/pd/iosw/iore/iomjre12/prodlit/935\\_pb.htm](http://www.cisco.com/warp/public/cc/pd/iosw/iore/iomjre12/prodlit/935_pb.htm)

#### **Cisco IOS Software Release Notes for Release 12.0S**

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/relnote/7000fam/rn120s.htm>

#### **Cisco IOS 12.0S Migration Guide**

[http://www.cisco.com/warp/public/cc/pd/iosw/iore/iomjre12/prodlit/940\\_pb.htm](http://www.cisco.com/warp/public/cc/pd/iosw/iore/iomjre12/prodlit/940_pb.htm)

#### **Cisco IOS Releases**

<http://www.cisco.com/warp/public/732/Releases/>

#### **Types of Cisco IOS Software Releases**

[http://www.cisco.com/warp/customer/cc/pd/iosw/iore/prodlit/537\\_pp.htm](http://www.cisco.com/warp/customer/cc/pd/iosw/iore/prodlit/537_pp.htm)

#### **Cisco IOS Release Designations Defined - *Software Lifecycle Definitions***

<http://www.cisco.com/warp/customer/417/109.html>

#### **Software Naming Conventions for Cisco IOS**

<http://www.cisco.com/warp/customer/432/7.html>

#### **Cisco IOS Reference Guide**

<http://www.cisco.com/warp/public/620/1.html>

#### **Cisco IOS Roadmap**

<http://www.cisco.com/warp/public/620/roadmap.shtml>

#### **Cisco Resource Manager**

<http://www.cisco.com/warp/public/cc/pd/wr2k/rsmn/index.shtml>

#### **Private I**

<http://www.opensystems.com/index.asp>

#### **Crystal Reports**

<http://www.seagatesoftware.com/crystalreports/>

## Netforensics

<http://www.netforensics.com/>

## NTP RFCs

RFC-1128, RFC-1129, RFC-1165 and RFC-1305, all available at <http://www.ietf.org/rfc>.

## Cisco IOS NTP Architecture

<http://www.cisco.com/univercd/cc/td/doc/product/iaabu/cddm/cddm111/adguide/ntp.htm>

## Network Time Protocol (NTP) Master Clock for the US

<http://tycho.usno.navy.mil/>

## Datum Inc, Bancomm Timing Division

<http://www.datum.com/>

## True Time Inc.

<http://www.true-time.com>

## The Time Web Server (Time Sync) by Dave Mills

<http://www.eecis.udel.edu/~ntp/>

## Coetanian Systems Time Synchronisation Server 100

<http://www.coetanian.com/tss/tss100.htm>

## UCD-SNMP project home page

<http://net-snmp.sourceforge.net/>

## Creating Core Dumps in Cisco IOS

[http://www.cisco.com/univercd/cc/td/doc/cisintwk/itg\\_v1/tr19aa.htm](http://www.cisco.com/univercd/cc/td/doc/cisintwk/itg_v1/tr19aa.htm)

## *General Features*

### Cisco IOS Command Line Interface

[http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgr/fun\\_r/index.htm](http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgr/fun_r/index.htm)

### CEF Whitepaper

[http://www.cisco.com/warp/public/cc/pd/iosw/iore/tech/cef\\_wp.htm](http://www.cisco.com/warp/public/cc/pd/iosw/iore/tech/cef_wp.htm)

### Cisco Express Forwarding Overview (12.0 Docs)

[http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgr/switch\\_c/xcprt2/xcccf.htm](http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgr/switch_c/xcprt2/xcccf.htm)

### Cisco Express Forwarding Configuration Task List

[http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgr/switch\\_c/xcprt2/xccfcf.htm](http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgr/switch_c/xcprt2/xccfcf.htm)

### Phil Harris's Packet Magazine Article on CEF (more of a CEF FAQ)

<http://www.cisco.com/warp/public/784/packet/oct99/router.html>

Wednesday, June 06, 2001

## ***Security***

**Increasing Security on IP Networks.** An old, but very important document on some of the security essentials in IP based networks.

<http://www.cisco.com/univercd/cc/td/doc/cisintwk/ics/cs003.htm>

**Cisco's INTERNET SECURITY ADVISORIES.** An online list of all of Cisco's Security advisories. It includes tutorials and details on how to protect yourself from some of the ugliness on the Internet today.

<http://www.cisco.com/warp/customer/707/advisory.html>

**Cisco's DS Documentation – 11.2 Security Configuration Guide.** The documentation with the 11.2 software release reorganised many of the security features of IOS into their own chapter. This documentation is the foundation for 12.0. Available on the *Cisco Documentation CD* or publicly on-line via Cisco Connection On-Line (CCO):

[http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/secur\\_c/index.htm](http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/secur_c/index.htm)

**Cisco IOS Password Encryption Tips**

<http://www.cisco.com/warp/public/701/64.html>

**Defining Strategies to Protect Against UDP Diagnostic Port Denial of Service Attacks**

<http://www.cisco.com/warp/public/707/3.html>

**Cisco public domain TACACs+ Server for Unix**

[ftp://ftp-eng.cisco.com/pub/tacaacs/tac\\_plus.F4.0.4.alpha.tar.Z](ftp://ftp-eng.cisco.com/pub/tacaacs/tac_plus.F4.0.4.alpha.tar.Z)

**Neighbouring Router Authentication: Overview and Guidelines**

[http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed\\_cr/secur\\_c/scprt5/scrouter.htm](http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed_cr/secur_c/scprt5/scrouter.htm)

**Cisco IOS TurboACL documentation**

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/120s6/turboacl.htm>

**Committed Access Rate (CAR)**

<http://www.cisco.com/warp/public/732/Tech/car/index.html>

**Configuring Committed Access Rate**

[http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos\\_c/qcpart1/qccar.htm](http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcpart1/qccar.htm)

**RFC 1812 Requirements for IP version 4 routers.** F Baker (ed). June 1995. (Status: PROPOSED STANDARD). Also see the update, RFC2644.

<http://www.ietf.org/rfc/rfc1812.txt>

**RFC 2196/FYI8 Site Security Handbook.** B. Fraser. September 1997. (Obsoletes RFC1244) (Status: INFORMATIONAL) One of the most useful starting places for Internet security.

<http://www.ietf.org/rfc/rfc2196.txt>

**RFC 2827/BCP38 Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing.** P Ferguson and D Senie. May 2000. (Status: BEST CURRENT PRACTICE).

<http://www.ietf.org/rfc/rfc2827.txt>

**RFC 2350/BCP21 Expectations for Computer Security Incident Response.** N Brownlee and E Guttman, June 1998. (Status: BEST CURRENT PRACTICE).

<http://www.ietf.org/rfc/rfc2350.txt>

**RFC 2644/BCP34 Changing the Default for Directed Broadcasts on Routers.** D Senie. August 1999. (Supplement to RFC1812) (Status: BEST CURRENT PRACTICE).

<http://www.ietf.org/rfc/rfc2644.txt>

**RFC 3013/BCP46 Security Expectations for Internet Service Providers** T Killalea, November 2000. (Status: BEST CURRENT PRACTICE).

<http://www.ietf.org/rfc/rfc3013.txt>

**Security Checklist for Internet Service Provider (ISP) Consumers draft-ietf-grip-user-02.txt (now expired)** T Hansen, June 1999.

<http://www.ietf.org/internet-drafts/draft-ietf-grip-user-02.txt>

**Site Security Handbook Addendum for ISP's draft-ietf-grip-ssh-add-00.txt (now expired)** T Debeaupuis, August 1999.

<http://www.ietf.org/internet-drafts/draft-ietf-grip-ssh-add-00.txt>

**Craig Huegen's SMURF page.** A very useful resource for ISPs to learn how to protect themselves from the many flavours of denial of service attacks.

<http://www.pentics.net/>

**Denial of Service Attacks Information Pages** by Paul Ferguson and Daniel Senie. Another very useful resource for ISPs to learn how to protect themselves from the many flavours of denial of service attacks.

<http://www.denialinfo.com/>

**Jared Mauch's [jared@puck.nether.net] Smurf Sweep Results Page.** Jared has scanned large sections of the Internet looking for networks that could be used as smurf amplifiers. This page lists his results and provides a way to check IP prefixes and AS numbers.

<http://puck.nether.net/smurf-check/>

## Routing

**Internet Routing Architectures: 2nd Edition**, Cisco Press <http://www.ciscopress.com/book.cfm?series=1&book=155> ISBN 1-578-702-33X Authors: Sam Halabi with Danny McPherson.

**RFC2281 Cisco Hot Standby Routing Protocol.** March 1998 (Status: INFORMATIONAL)

<http://www.ietf.org/rfc/rfc2281.txt>

**Using the Border Gateway Protocol for Interdomain Routing.** Available on the *Cisco Documentation CD* or publicly on-line via Cisco Connection On-Line (CCO):

<http://www.cisco.com/univercd/cc/td/doc/cisintwk/ics/icsbgp4.htm>

**Internetworking Technology Overview.** On-line whitepapers and tutorials on the essentials of routing and switching. Available on the *Cisco Documentation CD* or publicly on-line via Cisco Connection On-Line (CCO):

[http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito\\_doc/index.htm](http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/index.htm)

**Technology Information and Whitepapers.** Key references and practical internetworking examples. Available on the *Cisco Documentation CD* or publicly on-line via Cisco Connection On-Line (CCO):

<http://www.cisco.com/univercd/cc/td/doc/cisintwk/index.htm>

## Other References and Recommended Reading

Cisco ISP Essentials On-Line version

<http://www.cisco.com/public/cons/isp/documents/IOSEssentialsPDF.zip>

Wednesday, June 06, 2001

**Computer Networks, Third Edition** by Andrew Tannenbaum (ISBN: 0-13349-945-6)

**Interconnections : Bridges and Routers (second Ed)** by Radia Perlman (ISBN: 0-20163-448-1)

**Internetworking with TCP/IP, Volume 1: Principles, Protocols, and Architecture** by Douglas Comer (ISBN: 0-13216-987-8)

**IP Routing Fundamentals** by Mark Sportack (ISBN: 1-57870-071-x)

**IP Routing Primer** by Robert Wright (ISBN: 1-57870-108-2)

**Routing in the Internet** by Christian Huitema (ISBN: 0-13132-192-7)

**OSPF Network Design Solutions** by Thomas, Thomas M. (ISBN: 1-57870-046-9)

**ISP Survival Guide : Strategies for Running a Competitive ISP** by Geoff Huston (ISBN: 0-47131-499-4)

**Documenting Special Use IPv4 Address Blocks that have been registered with IANA** – Internet Draft: Bill Manning  
<http://www.ietf.org/internet-drafts/draft-manning-dsua-06.txt>

---

## GLOSSARY

---

ACL	Access Control List
AFRINIC	African Internet Network Information Centre
APNIC	Asia Pacific Network Information Centre
APOPS	Asia Pacific Operators
APRICOT	Asia Pacific Region Internet Conference on Operation Technologies
ARIN	American Registry for Internet Numbers
ARPANET	Advanced Research Projects Agency Network
AS	Autonomous System
ASIC	Application Specific Integrated Circuit
ATM	Asynchronous Transfer Mode
BCP	Best Current Practice
BGP	Border Gateway Protocol
CAR	Committed Access Rate
CCO	Cisco Connection Online
CEF	Cisco Express Forwarding
CIDR	Class InterDomain Routing
CLI	Command Line Interface
CPU	Central Processor Unit
DNS	Domain Name System
EGP	Exterior Gateway Protocol
EOF	European Operators Forum
FIB	Fowarding Information Base
FTP	File Transfer Protocol
GPS	Global Positioning System
HSRP	Hot Standby Routing Protocol
IANA	Internet Assigned Numbers Authority
IEPG	Internet Engineering & Planning Group
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
INET	Internet Society's annual conference
INTERNIC	Inter-Network Information Center
IOS	Internet Operating System (Cisco IOS)
IRR	Internet Routing Registry
ISIS	Intermediate System to Intermediate System
ISP	Internet Service Provider
LACNIC	Latin American & Caribbean Network Information Center
LIR	Local Internet Registry
MED	Multi-Exit Discriminator
MIT	Maintenance Induced Trouble
NANOG	North American Network Operators Group
NAS	Network Access Server
NOC	Network Operations Centre
NSP	Network Services Provider
NTP	Network Time Protocol
ORF	Outbound Route Filter
OSPF	Open Shortest Path First
PGP	Pretty Good Privacy
PoP	Point of Presence
PPP	Point to Point Protocol
PPS	Packet Per Second

Wednesday, June 06, 2001

QoS	Quality of Service
RADB	Routing Arbiter DataBase
RAM	Random Access Memory
RFC	Request for Comments – IETF document
RIP	Routing Information Protocol
RIPE	Réseaux IP Européens
RIPE NCC	RIPE Network Coordination Centre
RIR	Regional Internet Registry
RR	Route Reflector
RPF	Reverse Path Forwarding
SPD	Selective Packet Discard
STD	Internet Standard – status of IETF Document
TLD	Top Level Domain
TFTP	Trivial File Transfer Protocol

---

## APPENDIX 1 – ACCESS LIST AND REGULAR EXPRESSIONS

---

### *Access List Types*

<1-99>	IP standard access list
<100-199>	IP extended access list
<200-299>	Protocol type-code access list
<700-799>	48-bit MAC address access list
<1100-1199>	Extended 48-bit MAC address access list
<1300-1999>	IP standard access list (expanded range)
<2000-2699>	IP extended access list (expanded range)
compiled	Enable IP access-list compilation ( <b>new from 12.0(6)S</b> )
rate-limit	Simple rate-limit specific access-list
permit	Specify packets to forward
deny	Specify packets to reject
dynamic	Specify a DYNAMIC list of PERMITs or DENYs
<0-255>	An IP protocol number
ahp	Authentication Header Protocol
eigrp	Cisco's EIGRP routing protocol
esp	Encapsulation Security Payload
gre	Cisco's GRE tunneling
icmp	Internet Control Message Protocol
igmp	Internet Gateway Message Protocol
igrp	Cisco's IGRP routing protocol
ip	Any Internet Protocol
ipinip	IP in IP tunneling
nos	KA9Q NOS compatible IP over IP tunnelling
ospf	OSPF routing protocol
pcp	Payload Compression Protocol
pim	Protocol Independent Multicast
tcp	Transmission Control Protocol
udp	User Datagram Protocol
a.b.c.d	source or destination address
any	any source host
host	a single source host (equivalent to a.b.c.d 255.255.255.255)
log	log matches against this entry
log-input	log matches against this entry, including input interface
precedence	match packets with given precedence value
tos	match packets with given TOS value

There are other options, depending on which IP protocol has been chosen. For example, TCP has these further options for configuring an access-list:

Wednesday, June 06, 2001

ack, eq, established, fin, gt, log, log-input, lt, neq, precedence, psh, range, rst, syn, tos, urg

## *IOS Regular Expressions*

Here are some examples of regular expressions in use on ISP routers today. Hopefully the examples are useful. Please refer to the documentation for more in depth discussion and detailed examples.

<code>^200\$</code>	match AS200 only
<code>.*</code>	match anything
<code>.+</code>	match at least one character
<code>^*</code>	match all ASes including local AS
<code>^\$</code>	match this AS only
<code>^200_</code>	match all ASes received from AS200
<code>_200_</code>	match all ASes which have AS200 in the path
<code>_200\$</code>	match all ASes with AS200 origin only, whatever the path
<code>^200_210\$</code>	match AS210 origin and received from AS200 only
<code>_200_210_</code>	match all ASes which have been through AS200 $\leftrightarrow$ AS210 link
<code>^(200_)+\$</code>	match at least one of AS200 (or multiple occurrences of one AS) (usually from AS_PATH stuffing <sup>33</sup> )
<code>^(_[0-9]+)\$</code>	match at least one AS (or multiple occurrences of one AS)
<code>_\(65350\)_</code>	match all ASes which have confederation sub-AS 65350 in the path
<code>^[0-9]+\$</code>	match AS_PATH length of one
<code>^[0-9]+_[0-9]+\$</code>	match AS_PATH length of two
<code>^[0-9]*_[0-9]+\$</code>	match AS_PATH length of one or two
<code>^[0-9]*_[0-9]*\$</code>	match AS_PATH length of one or two
<code>^[0-9]+_[0-9]+_[0-9]+\$</code>	match AS_PATH length of three
<code>_(701 1800)_</code>	match anything which has gone through AS701 or AS1800
<code>_1849(_.+_)12163\$</code>	match anything of origin AS12163 and passed through AS1849

---

<sup>33</sup> AS path stuffing means seeing an AS path such as 200\_200\_200 for a network announcement. This is commonly used when defining particular routing policy, for example loadsharing.

---

## APPENDIX 2 – CUT AND PASTE TEMPLATES

---

The following are some cut and paste template you can modify to configure your routers. *Make sure you change the IP addresses and AS numbers* in the templates!

### *General System Template*

```
no service finger
no service pad
no service udp-small-servers
no service tcp-small-servers
no ip bootp server
service nagle
service timestamps debug datetime localtime show-timezone msec
service timestamps log datetime localtime show-timezone msec
service tcp-keepalives-in
no ip source-route
no logging console
logging buffered 16384
logging trap debugging
logging x.x.x.x
ip subnet-zero
ip classless
```

### *General Interface Template*

```
no ip redirect
no ip direct broadcast
no ip proxy-arp
no cdp enable
```

### *General Security Template*

```
service password-encryption
enable secret <removed>
no enable password
aaa new-model
aaa authentication login default tacacs+ enable
aaa authentication enable default tacacs+ enable
tacacs-server host x.x.x.x
tacacs-server key <removed>
```

### *General BGP template*

This generic template should be used for configuring BGP on a router. It also includes sample configuration for a peer-group.

```
router bgp 65280
  bgp dampening
  no synchronisation
  no auto-summary
  bgp log-neighbor-changes
  neighbor ibgp-peer peer-group
```

Wednesday, June 06, 2001

```
neighbor ibgp-peer remote-as 65280
neighbor ibgp-peer password <removed>
neighbor ibgp-peer send-community
neighbor ibgp-peer update-source loopback 0
```

## *Martian and RFC1918 Networks Template*

This list represents the common filtering practice of several ISPs. It includes default, multicast, and RFC1918 networks, as well as the so-called Martian networks as can be found in <http://www.ietf.org/internet-drafts/draft-manning-dsua-06.txt>. The use of these filters on inbound and outbound interfaces of border routers is recommended. It is also recommended to be used on BGP peerings where the service provider is receiving or sending full prefixes to a neighbouring Autonomous System.

### **IP Access-List Example**

```
! access-list 150 to deny RFC1918 and Martian networks
access-list 150 deny ip 0.0.0.0 0.255.255.255 255.0.0.0 0.255.255.255
access-list 150 deny ip 10.0.0.0 0.255.255.255 255.0.0.0 0.255.255.255
access-list 150 deny ip 127.0.0.0 0.255.255.255 255.0.0.0 0.255.255.255
access-list 150 deny ip 169.254.0.0 0.0.255.255 255.255.0.0 0.0.255.255
access-list 150 deny ip 172.16.0.0 0.15.255.255 255.240.0.0 0.15.255.255
access-list 150 deny ip 192.0.2.0 0.0.0.255 255.255.255.0 0.0.0.255
access-list 150 deny ip 192.168.0.0 0.0.255.255 255.255.0.0 0.0.255.255
access-list 150 deny ip 224.0.0.0 31.255.255.255 224.0.0.0 31.255.255.255
access-list 150 deny ip any 255.255.255.128 0.0.0.127
access-list 150 permit ip any any
!
```

### **IP Prefix-List Example**

```
ip prefix-list rfc1918-sua description Networks which shouldn't be announced
ip prefix-list rfc1918-sua deny 0.0.0.0/8 le 32
ip prefix-list rfc1918-sua deny 10.0.0.0/8 le 32
ip prefix-list rfc1918-sua deny 127.0.0.0/8 le 32
ip prefix-list rfc1918-sua deny 169.254.0.0/16 le 32
ip prefix-list rfc1918-sua deny 172.16.0.0/12 le 32
ip prefix-list rfc1918-sua deny 192.0.2.0/24 le 32
ip prefix-list rfc1918-sua deny 192.168.0.0/16 le 32
ip prefix-list rfc1918-sua deny 224.0.0.0/3 le 32
ip prefix-list rfc1918-sua deny 0.0.0.0/0 ge 25
ip prefix-list rfc1918-sua permit 0.0.0.0/0 le 32
!
```

## APPENDIX 3 – EXAMPLE CONFIGURATIONS

This appendix aims to give ISPs who are learning about the art of constructing a Service Provider backbone a guidance on some of the configuration steps and design hints which have been covered throughout IOS Essentials. The common elements of a service provider's network have all been included, including border, aggregation, and DIAL access routers. Clearly it is almost impossible to give a sample configurations which will work for all ISPs. But the following aims to give guidance to the small or medium ISP who is not long started in operations so that they gain the general principles of what is required for the routers on their network.

### *Simple Network Plan*

Figure 37 shows a simple network diagram of a basic ISP PoP which will be used in these examples. It has the key elements which an ISP PoP has: a border router, two core routers, aggregation routers (for leased line, or permanent connected customers), two service routers (for web hosting and the ISP's own services), a dial aggregation router, and a router which connects to the Network Operations Centre. Obviously as Service Providers grow their network will be more sophisticated than this, and their configuration needs will likewise grow to match. However, this example should serve as a good grounding for future growth.

An important element to pay attention to throughout these examples is the very rigorous filtering which has been applied throughout the network. Filtering at the right places ensures security for the service provider, as well as a reliably performing network.

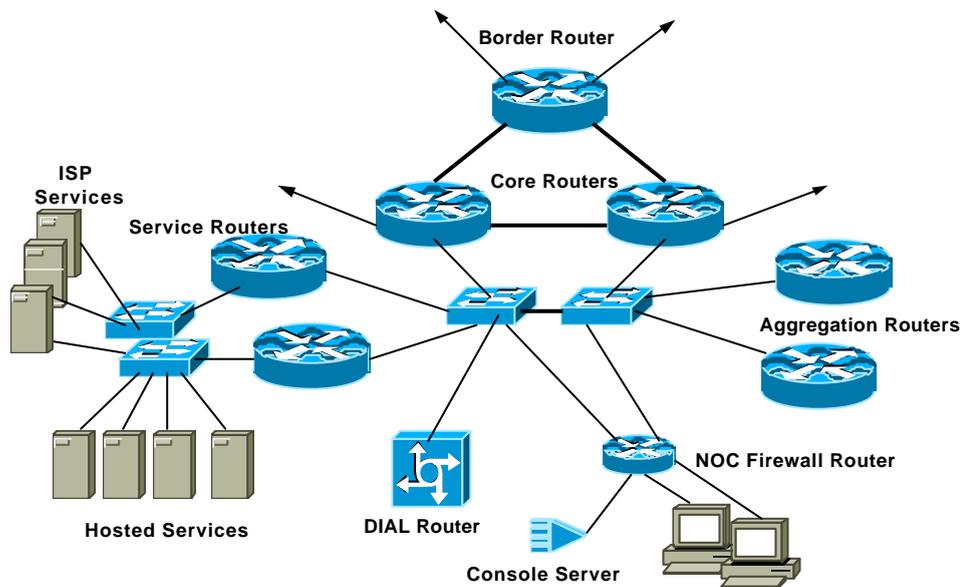


Figure 37 – Network Used for Configuration and Filtering Sample

### *Configurations*

The IOS configurations given below do work, and have been tested in a lab environment. They are heavily based on known current configurations used in the field today, with additions/modifications to include as many recommendations from the main IOS Essentials document as is feasible.

Wednesday, June 06, 2001

The configurations have been annotated using IOS comments “!” where more explanation is required. This should make it easier to “cut and paste” configuration into your own test environment. But the golden rule, never cut and paste into a live network, don’t implement any configuration until you have extensively tested it in your own environment.

The IP addresses and AS Numbers used are completely fictitious, and at the time of writing still belong to IANA Reserved Address space. At some time in the future these addresses or ASes will become allocated to an ISP and assigned to an end customer. Please do not use these addresses in your own backbone, or announce them to the Internet.

## ISP Addressing Plan

For reference purposes, the following addressing plan has been used for this ISP backbone. Address block is 220.144.128.0/19.

220.144.159.192/26	Router Loopbacks
220.144.159.128/26	ISP Network Operations Centre
220.144.159.64/26	point to point links and vLANs
220.144.159.0/26	ISP Services LAN
220.144.158.0/24	Customer Hosting LAN
220.144.157.0/24	DIAL pool
:	
:	
220.144.128.0 upwards	to be used for customer assignments

### NOC Hosts

220.144.159.129	loghost, tacacs+, ftp, tftp
220.144.159.130	Network management workstation
220.144.159.131	Secure Shell gateway
220.144.159.132	Radius Server for DIAL
220.144.159.160	Thru 175 – host range for terminal server console port access
220.144.159.180	Primary DNS
220.144.159.189	Out of band management terminal server
220.144.159.190	NOC firewall router

### ISP Services LAN

220.144.159.1	Secondary DNS, NTP server
220.144.159.2	Secondary MX, Caching DNS server
220.144.159.3	Primary MX, ftp, www
220.144.159.4	pop3, smtp relay for DIAL only
220.144.159.5	News reading system for DIAL only
220.144.159.59	Service-Router1
220.144.159.60	Service-Router2
220.144.159.61	HSRP gateway 1
220.144.159.62	HSRP gateway 2

## Border Router

The border router is perhaps the most important router in an ISP’s backbone network operation. It provides connectivity to the rest of the Internet for the service provider, and also protects the service provider’s network and customers from “the ravages of the Internet”. In ISP Workshops we present, we use the analogy of the front door of your house, or the garden gate – protecting you from the world out there. The configuration of the border router is critical to the correct and reliable operation of the rest of the ISP’s business. The following configuration example is similar to some of those which have been deployed on live networks today.

```
version 12.0
service nagle
```

```

no service pad
service tcp-keepalives-in
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
service password-encryption
!
hostname border-router
!
boot system flash c7200-k4p-mz.120-10.S2
boot system flash
!
logging buffered 16384 debugging
aaa new-model
aaa authentication login default tacacs+ enable
aaa authentication enable default tacacs+ enable
aaa accounting exec default start-stop tacacs+
aaa accounting commands 15 default start-stop tacacs+
!
enable secret shhhhhthisisasecret
!
clock timezone GMT 0
ip subnet-zero
ip cef
no ip source-route
no ip finger
ip telnet source-interface Loopback0
ip tftp source-interface Loopback0
ip ftp source-interface Loopback0
ip ftp username cisco
ip ftp password shhhhsecret
no ip bootp server
!
! Set up DNS - note that one secondary NS is hosted by my upstream ISP
ip domain-name net.galaxy
ip name-server 220.144.159.1
ip name-server 220.144.159.2
ip name-server 219.10.2.1
!
! SSH support
ip ssh time-out 120
ip ssh authentication-retries 3
!
interface Loopback0
description Loopback interface on border-router
ip address 220.144.159.192 255.255.255.255
no ip directed-broadcast
!
interface FastEthernet0/0
description Ethernet to Core1 (x-over ethernet)
ip address 220.144.159.65 255.255.255.252
no ip redirects
no ip directed-broadcast
no ip proxy-arp
ip route-cache flow
!
interface Serial1/0
description 256Kb HDLC link to Buzz Internet
bandwidth 256
ip address 219.10.1.2 255.255.255.252
ip access-group 100 in
ip access-group 101 out
no ip redirects
no ip directed-broadcast
ip route-cache flow
no fair-queue
!
interface Serial1/1
description 512Kb HDLC link to Whoosh Internet
bandwidth 512

```

Wednesday, June 06, 2001

```
ip address 219.50.10.2 255.255.255.252
ip access-group 100 in
ip access-group 101 out
no ip redirects
no ip directed-broadcast
ip route-cache flow
no fair-queue
!
interface FastEthernet2/0
description Ethernet to Core2 (x-over ethernet)
ip address 220.144.159.69 255.255.255.252
no ip redirects
no ip directed-broadcast
no ip proxy-arp
ip route-cache flow
!
router ospf 100
network 219.50.10.0 0.0.0.3 area 0
network 219.10.1.0 0.0.0.3 area 0
network 220.144.159.64 0.0.0.7 area 0
network 220.144.159.192 0.0.0.0 area 0
passive-interface Serial1/0
passive-interface Serial1/1
passive-interface Loopback0
log-adjacency-changes
!
router bgp 64511
no synchronization
bgp log-neighbor-changes
! Use peer-groups - more efficient
neighbor core-ibgp peer-group
neighbor core-ibgp remote-as 64511
neighbor core-ibgp update-source Loopback0
neighbor core-ibgp password BGPsecretPW
neighbor core-ibgp send-community
! Get full routing table from both upstreams - block RFC1918+ address space inbound
! only announce my address block outbound
neighbor 219.10.1.2 remote-as 64400
neighbor 219.10.1.2 description Connection to Buzz Internet
neighbor 219.10.1.2 prefix-list infilter in
neighbor 219.10.1.2 prefix-list outfilter out
neighbor 219.10.1.2 password BuzzSecretPW
neighbor 219.50.10.1 remote-as 64500
neighbor 219.50.10.1 description Connection to Whoosh Internet
neighbor 219.50.10.1 prefix-list infilter in
neighbor 219.50.10.1 prefix-list outfilter out
neighbor 219.50.10.1 password WhooshSecretPW
! iBGP with core routers - using route reflectors
neighbor 220.144.159.193 peer-group core-ibgp
neighbor 220.144.159.193 description Core1
neighbor 220.144.159.194 peer-group core-ibgp
neighbor 220.144.159.194 description Core2
no auto-summary
!
ip classless
ip route 10.0.0.0 255.0.0.0 Null0
ip route 172.16.0.0 255.240.0.0 Null0
ip route 192.168.0.0 255.255.0.0 Null0
ip route 220.144.128.0 255.255.224.0 Null0
ip tacacs source-interface Loopback0
ip bgp-community new-format
!
logging trap debugging
logging source-interface Loopback0
logging 220.144.159.129
! SNMP access-list
access-list 1 permit 220.144.159.129
access-list 1 permit 220.144.159.130
access-list 1 deny any log
! INBOUND access-list on external interfaces
```

```

access-list 100 deny ip 10.0.0.0 0.255.255.255 any ! BLOCK THE MARTIANS
access-list 100 deny ip 127.0.0.0 0.255.255.255 any
access-list 100 deny ip 172.16.0.0 0.15.255.255 any
access-list 100 deny ip 192.168.0.0 0.0.255.255 any
access-list 100 deny ip 220.144.128.0 0.0.31.255 any
access-list 100 deny ip any 0.0.0.255 255.255.255.0
access-list 100 deny ip any 0.0.0.0 255.255.255.0
access-list 100 permit icmp any host 219.10.1.2 ! ACCESS TO SERIAL PORTS
access-list 100 permit icmp any host 219.50.10.2
access-list 100 permit icmp any 220.144.128.0 0.0.31.255
access-list 100 permit udp any host 219.10.1.2 eq ntp
access-list 100 permit udp any host 219.50.10.2 eq ntp
access-list 100 deny ip any host 219.10.1.2 log
access-list 100 deny ip any host 219.50.10.2 log
access-list 100 permit tcp any 220.144.128.0 0.0.31.255 established
access-list 100 permit tcp any 220.144.128.0 0.0.31.255 eq 22 ! SSH
access-list 100 permit tcp any 220.144.128.0 0.0.31.255 eq ftp
access-list 100 permit tcp any 220.144.128.0 0.0.31.255 eq ftp-data
access-list 100 permit tcp any 220.144.128.0 0.0.31.255 eq ident ! Needed for web
access-list 100 permit udp any 220.144.128.0 0.0.31.255 eq ntp
access-list 100 permit tcp any 220.144.128.0 0.0.31.255 eq smtp
access-list 100 permit tcp any 220.144.128.0 0.0.31.255 eq www
access-list 100 permit tcp any 220.144.128.0 0.0.31.255 eq pop3
access-list 100 permit tcp any 220.144.128.0 0.0.31.255 eq 143 ! IMAP
access-list 100 permit tcp any 220.144.128.0 0.0.31.255 eq 389 ! LDAP
access-list 100 permit tcp any 220.144.128.0 0.0.31.255 eq 443 ! HTTPS
access-list 100 deny tcp any 220.144.128.0 0.0.31.255 eq 2049 log
access-list 100 deny udp any 220.144.128.0 0.0.31.255 eq 2049 log ! NFS
access-list 100 deny tcp any 220.144.128.0 0.0.31.255 eq 6000 log ! X
access-list 100 permit tcp any 220.144.128.0 0.0.31.255 gt 1023
access-list 100 permit udp any 220.144.128.0 0.0.31.255 gt 1023
access-list 100 deny ip any any log
! OUTBOUND access-list on external interfaces
access-list 101 permit ip 220.144.128.0 0.0.31.255 any ! My address block
access-list 101 permit ip host 219.10.1.2 any ! and external facing interfaces
access-list 101 permit ip host 219.50.10.2 any
access-list 101 deny ip any any log
! VTY access-list
access-list 198 permit ip 220.144.159.128 0.0.0.63 any ! NOC systems
access-list 198 permit ip 220.144.159.192 0.0.0.63 any ! Router loopbacks
access-list 198 deny ip any any log
! Industry convention is that acl 199 blocks everything
access-list 199 deny ip any any log
!
! BGP INBOUND filters
ip prefix-list infilter description Networks which shouldn't be accepted
ip prefix-list infilter deny 0.0.0.0/8 le 32
ip prefix-list infilter deny 10.0.0.0/8 le 32
ip prefix-list infilter deny 127.0.0.0/8 le 32
ip prefix-list infilter deny 169.254.0.0/16 le 32
ip prefix-list infilter deny 172.16.0.0/12 le 32
ip prefix-list infilter deny 192.0.2.0/24 le 32
ip prefix-list infilter deny 192.168.0.0/16 le 32
ip prefix-list infilter deny 220.144.128.0/19 le 32
ip prefix-list infilter deny 224.0.0.0/3 le 32
ip prefix-list infilter deny 0.0.0.0/0 ge 25
ip prefix-list infilter permit 0.0.0.0/0 le 32
!
! BGP OUTBOUND filters
ip prefix-list outfilter description Networks which should be announced to upstreams
ip prefix-list outfilter permit 220.144.128.0/19
!
tacacs-server host 220.144.159.129
tacacs-server key SecretToo
snmp-server community NotTelling RO 1
snmp-server location Somewhere
snmp-server contact Network Operations Centre <noc@net.galaxy>
snmp-server enable traps snmp
snmp-server host 220.144.159.130 SecretToo

```

Wednesday, June 06, 2001

```
banner login ^
Authorised Access Only

This system is the property of Galaxy Internet

Disconnect IMMEDIATELY if you are not an authorised user!

Contact noc@net.galaxy +98 765 4321 for help.
^
!
line con 0
  exec-timeout 3 0
  transport preferred none
  transport input none
  transport output telnet
line aux 0
  transport preferred none
  transport input none
  transport output telnet
line vty 0 4
  access-class 198 in
  exec-timeout 0 0
  transport preferred none
  transport input telnet ssh
  transport output telnet
!
! Where Router core-dumps go
exception protocol ftp
exception dump 220.144.159.129
! NTP configuration
ntp authentication-key 1 md5 secretAlso
ntp authenticate
ntp trusted-key 1
ntp source Loopback0
ntp server 219.10.1.1
ntp peer 220.144.159.1 key 1
ntp server 219.50.10.1
end
```

## Core Router

Two core routers have been given in the example in Figure 37 above. However, only one configuration is given here – the configuration for Core-router2 is almost identical. Notice that the core router has a simpler configuration than the routers at the edge of the ISP's network. Core routers tend not to do packet filtering or routing policy, the design goal being more for reliability and a configuration which requires no change in the short to medium term. The example here assumes that the Core Router is a Cisco 7206.

```
version 12.0
service nagle
no service pad
service tcp-keepalives-in
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
service password-encryption
!
hostname Core-router1
!
boot system flash slot0:c7200-k4p-mz.120-10.S2
boot system flash
!
logging buffered 16384 debugging
aaa new-model
aaa authentication login default tacacs+ enable
aaa authentication enable default tacacs+ enable
aaa accounting exec default start-stop tacacs+
aaa accounting commands 15 default start-stop tacacs+
enable secret shhhhhthisisasecret
```

```

!
ip cef
clock timezone GMT 0
ip subnet-zero
no ip source-route
no ip finger
ip telnet source-interface Loopback0
ip tftp source-interface Loopback0
ip ftp source-interface Loopback0
ip ftp username cisco
ip ftp password shhhhsecret
no ip bootp server
!
ip domain-name net.galaxy
ip name-server 220.144.159.1
ip name-server 220.144.159.2
ip name-server 219.10.2.1
!
! SSH support
ip ssh time-out 120
ip ssh authentication-retries 3
!
interface Loopback0
description Loopback interface on Core-router1
ip address 220.144.159.193 255.255.255.255
no ip directed-broadcast
!
interface FastEthernet0/0
description Ethernet to Border (x-over ethernet)
ip address 220.144.159.66 255.255.255.252
no ip redirects
no ip directed-broadcast
no ip proxy-arp
!
interface Serial1/0
description E1 HDLC link to Smalltown PoP
bandwidth 2048
ip address 220.144.159.77 255.255.255.252
no ip redirects
no ip directed-broadcast
no fair-queue
!
interface FastEthernet2/0
description Ethernet to Core2 (x-over ethernet)
ip address 220.144.159.73 255.255.255.252
delay 9
no ip redirects
no ip directed-broadcast
no ip proxy-arp
!
interface FastEthernet3/0
description Core Ethernet (SW1 port 1)
ip address 220.144.159.97 255.255.255.240
no ip redirects
no ip directed-broadcast
no ip proxy-arp
!
interface FastEthernet4/0
description Core Ethernet (SW2 port 1)
ip address 220.144.159.113 255.255.255.240
no ip redirects
no ip directed-broadcast
no ip proxy-arp
!
router ospf 100
network 220.144.159.64 0.0.0.3 area 0
network 220.144.159.72 0.0.0.3 area 0
network 220.144.159.76 0.0.0.3 area 0
network 220.144.159.96 0.0.0.15 area 0

```

Wednesday, June 06, 2001

```
network 220.144.159.112 0.0.0.15 area 0
network 220.144.159.193 0.0.0.0 area 0
passive-interface Loopback0
log-adjacency-changes
!
router bgp 64511
network 220.144.128.0 mask 255.255.224.0
no synchronization
bgp log-neighbor-changes
! Two peergroups, one for the CORE iBGP, the other for Route Reflector Cluster
neighbor core-ibgp peer-group
neighbor core-ibgp remote-as 64511
neighbor core-ibgp update-source Loopback0
neighbor core-ibgp password BGPsecretPW
neighbor core-ibgp send-community
neighbor rr-client peer-group
neighbor rr-client remote-as 64511
neighbor rr-client update-source Loopback0
neighbor rr-client password BGPsecretPW
neighbor rr-client send-community
neighbor rr-client route-reflector-client
neighbor 220.144.159.192 peer-group core-ibgp
neighbor 220.144.159.192 description Border
neighbor 220.144.159.194 peer-group core-ibgp
neighbor 220.144.159.194 description Core2
neighbor 220.144.159.195 peer-group rr-client
neighbor 220.144.159.195 description Gateway1
neighbor 220.144.159.196 peer-group rr-client
neighbor 220.144.159.196 description Gateway2
neighbor 220.144.159.197 peer-group rr-client
neighbor 220.144.159.197 description Service1
neighbor 220.144.159.198 peer-group rr-client
neighbor 220.144.159.198 description Service2
neighbor 220.144.159.200 peer-group rr-client
neighbor 220.144.159.200 description DIAL Access Router
no auto-summary
!
ip classless
ip route 10.0.0.0 255.0.0.0 Null0
ip route 172.16.0.0 255.240.0.0 Null0
ip route 192.168.0.0 255.255.0.0 Null0
ip route 220.144.128.0 255.255.224.0 Null0
ip tacacs source-interface Loopback0
ip bgp-community new-format
!
logging trap debugging
logging source-interface Loopback0
logging 220.144.159.129
access-list 1 permit 220.144.159.129
access-list 1 permit 220.144.159.130
access-list 1 deny any log
access-list 198 permit ip 220.144.159.128 0.0.0.63 any ! VTY access-list - NOC
access-list 198 permit ip 220.144.159.192 0.0.0.63 any ! Router loopbacks
access-list 198 deny ip any any log
access-list 199 deny ip any any log
!
tacacs-server host 220.144.159.129
tacacs-server key SecretToo
snmp-server community NotTelling RO 1
snmp-server location Somewhere
snmp-server contact Network Operations Centre <noc@net.galaxy>
snmp-server enable traps snmp
snmp-server host 220.144.159.130 SecretToo
banner login ^
Authorized Access Only
```

This system is the property of Galaxy Internet

Disconnect IMMEDIATELY if you are not an authorised user!

```

Contact noc@net.galaxy +98 765 4321 for help.
^
!
line con 0
  exec-timeout 3 0
  transport preferred none
  transport input none
  transport output telnet
line aux 0
  transport preferred none
  transport input none
  transport output telnet
line vty 0 4
  access-class 198 in
  exec-timeout 0 0
  transport preferred none
  transport input telnet ssh
  transport output telnet
!
exception protocol ftp
exception dump 220.144.159.129
ntp authentication-key 1 md5 secretAlso
ntp authenticate
ntp trusted-key 1
ntp source Loopback0
ntp server 220.144.159.192 key 1
ntp server 220.144.159.1 key 1
ntp peer 220.144.159.194 key 1
end

```

## Aggregation Router

The aggregation or gateway routers are used for connecting fixed link customers to the service provider's backbone. As previously only one example will be given below. The second aggregation router will use a very similar configuration. The aggregation router in this example is assumed to be a 3640.

```

version 12.0
service nagle
no service pad
service tcp-keepalives-in
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
service password-encryption
!
hostname Gateway-router1
!
boot system flash
logging buffered 16384 debugging
aaa new-model
aaa authentication login default tacacs+ enable
aaa authentication enable default tacacs+ enable
aaa accounting exec default start-stop tacacs+
aaa accounting commands 15 default start-stop tacacs+
enable secret shhhhhthisisasecret
!
ip cef
clock timezone GMT 0
ip subnet-zero
no ip source-route
no ip finger
ip telnet source-interface Loopback0
ip tftp source-interface Loopback0
ip ftp source-interface Loopback0
ip ftp username cisco
ip ftp password shhhhsecret
no ip bootp server
ip domain-name net.galaxy

```

Wednesday, June 06, 2001

```
ip name-server 220.144.159.1
ip name-server 220.144.159.2
ip name-server 219.10.2.1
!
! SSH support
ip ssh time-out 120
ip ssh authentication-retries 3
!
partition flash 2 8 8
!
interface Loopback0
description Loopback interface on GW1
ip address 220.144.159.195 255.255.255.255
no ip directed-broadcast
!
interface FastEthernet0/0
description Core Ethernet (SW1 port 3)
ip address 220.144.159.99 255.255.255.240
no ip redirects
no ip directed-broadcast
no ip proxy-arp
!
interface FastEthernet0/1
description Core Ethernet (SW2 port 3)
ip address 220.144.159.115 255.255.255.240
no ip redirects
no ip directed-broadcast
no ip proxy-arp
!
interface Serial1/0
description HDLC 64K link to Cust1 CT1
bandwidth 64
ip verify unicast reverse-path
ip unnumbered Loopback0
no ip redirects
no ip directed-broadcast
no ip proxy-arp
!
interface Serial1/1
description 64K HDLC link to Cust2 CT2
bandwidth 64
ip unnumbered Loopback0
no ip redirects
no ip directed-broadcast
no ip proxy-arp
!
interface Serial1/2
description 64K HDLC link to Cust 3 CT3
bandwidth 64
ip verify unicast reverse-path
ip unnumbered Loopback0
no ip redirects
no ip directed-broadcast
no ip proxy-arp
!
interface Serial1/3
description 64k HDLC link to Cust 4 CT4
bandwidth 64
ip verify unicast reverse-path
ip unnumbered Loopback0
no ip redirects
no ip directed-broadcast
no ip proxy-arp
!
router ospf 100
network 220.144.159.64 0.0.0.3 area 0
network 220.144.159.72 0.0.0.3 area 0
network 220.144.159.76 0.0.0.3 area 0
network 220.144.159.96 0.0.0.15 area 0
network 220.144.159.112 0.0.0.15 area 0
```

```

network 220.144.159.193 0.0.0.0 area 0
passive-interface Loopback0
log-adjacency-changes
!
router bgp 64511
network 220.144.128.0
network 220.144.129.0 mask 255.255.255.224
network 220.144.129.32 mask 255.255.255.224
network 220.144.129.64 mask 255.255.255.192
no synchronization
bgp log-neighbor-changes
neighbor rr peer-group
neighbor rr remote-as 64511
neighbor rr update-source Loopback0
neighbor rr password BGPsecretPW
neighbor rr send-community
neighbor 220.144.159.193 peer-group rr
neighbor 220.144.159.193 description Core1
neighbor 220.144.159.194 peer-group rr
neighbor 220.144.159.194 description Core2
no auto-summary
!
ip classless
ip route 10.0.0.0 255.0.0.0 Null0
ip route 172.16.0.0 255.240.0.0 Null0
ip route 192.168.0.0 255.255.0.0 Null0
ip route 220.144.128.0 255.255.224.0 Null0
ip route 220.144.128.0 255.255.255.0 Serial1/0 permanent
ip route 220.144.129.0 255.255.255.224 Serial1/1 permanent
ip route 220.144.129.32 255.255.255.224 Serial1/2 permanent
ip route 220.144.129.64 255.255.255.192 Serial1/3 permanent
ip tacacs source-interface Loopback0
ip bgp-community new-format
!
logging trap debugging
logging source-interface Loopback0
logging 220.144.159.129
access-list 1 permit 220.144.159.129
access-list 1 permit 220.144.159.130
access-list 1 deny any log
access-list 198 permit ip 220.144.159.128 0.0.0.63 any ! VTY access-list - NOC
access-list 198 permit ip 220.144.159.192 0.0.0.63 any ! Router loopbacks
access-list 198 deny ip any any log
access-list 199 deny ip any any log
!
tacacs-server host 220.144.159.129
tacacs-server key SecretToo
snmp-server community NotTelling RO 1
snmp-server location Somewhere
snmp-server contact Network Operations Centre <noc@net.galaxy>
snmp-server enable traps snmp
snmp-server host 220.144.159.130 SecretToo
banner login ^
Authorised Access Only

This system is the property of Galaxy Internet

Disconnect IMMEDIATELY if you are not an authorised user!

Contact noc@net.galaxy +98 765 4321 for help.
^
!
line con 0
exec-timeout 3 0
transport preferred none
transport input none
transport output telnet
line aux 0
transport preferred none

```

Wednesday, June 06, 2001

```
transport input telnet
transport output telnet
line vty 0 4
access-class 198 in
exec-timeout 0 0
transport preferred none
transport input telnet ssh
transport output telnet
!
exception protocol ftp
exception dump 220.144.159.129
ntp authentication-key 1 md5 secretAlso
ntp authenticate
ntp trusted-key 1
ntp source Loopback0
ntp server 220.144.159.193 key 1
ntp server 220.144.159.194 key 1
ntp peer 220.144.159.196 key 1
ntp peer 220.144.159.197 key 1
ntp peer 220.144.159.198 key 1
end
```

## Service Router

The service routers are used for connecting the hosted services (e.g. content provider systems) and the ISPs own services (Mail, News, DNS) to the service provider's backbone. As previously only one example will be given below. The second aggregation router will use a very similar configuration. The router in this example is assumed to be a 2620 with four fast ethernet ports.

```
version 12.0
service nagle
no service pad
service tcp-keepalives-in
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
service password-encryption
!
hostname Service-router1
!
boot system flash
logging buffered 16384 debugging
aaa new-model
aaa authentication login default tacacs+ enable
aaa authentication enable default tacacs+ enable
aaa accounting exec default start-stop tacacs+
aaa accounting commands 15 default start-stop tacacs+
enable secret shhhhthisisasecret
!
clock timezone GMT 0
ip subnet-zero
no ip source-route
no ip finger
ip telnet source-interface Loopback0
ip tftp source-interface Loopback0
ip ftp source-interface Loopback0
ip ftp username cisco
ip ftp password shhhhsecret
no ip bootp server
ip domain-name net.galaxy
ip name-server 220.144.159.1
ip name-server 220.144.159.2
ip name-server 219.10.2.1
!
! SSH support
ip ssh time-out 120
ip ssh authentication-retries 3
!
```

```

partition flash 2 8 8
!
interface Loopback0
  description Loopback interface on SR1
  ip address 220.144.159.197 255.255.255.255
  no ip directed-broadcast
!
interface FastEthernet0/0
  description Core Ethernet (SW1 vlan1 port 4)
  ip address 220.144.159.101 255.255.255.240
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
!
interface FastEthernet0/1
  description Core Ethernet (SW2 vlan2 port 13)
  ip address 220.144.159.117 255.255.255.240
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
!
interface FastEthernet1/0
  description Hosted Server Ethernet (SW3 vlan1 port 1)
  ip address 220.144.158.1 255.255.255.0
  ip access-group 140 in
  ip access-group 141 out
  standby 10 priority 150
  standby 10 preempt
  standby 10 ip 220.144.158.253
  standby 20 ip 220.144.158.254
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
!
interface FastEthernet1/1
  description Galaxy ISP Server Ethernet (SW3 vlan2 port 8)
  ip address 220.144.159.59 255.255.255.192
  ip access-group 150 in
  ip access-group 151 out
  standby 30 priority 150
  standby 30 preempt
  standby 30 ip 220.144.159.61
  standby 40 ip 220.144.159.62
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
!
router ospf 100
  network 220.144.159.96 0.0.0.15 area 0
  network 220.144.159.112 0.0.0.15 area 0
  network 220.144.159.197 0.0.0.0 area 0
  network 220.144.159.0 0.0.0.63 area 0
  passive-interface FastEthernet1/1
  passive-interface Loopback0
  log-adjacency-changes
!
router bgp 64511
  network 220.144.158.0
  no synchronization
  bgp log-neighbor-changes
  neighbor rr peer-group
  neighbor rr remote-as 64511
  neighbor rr update-source Loopback0
  neighbor rr password BGPsecretPW
  neighbor rr send-community
  neighbor 220.144.159.193 peer-group rr
  neighbor 220.144.159.193 description Core1
  neighbor 220.144.159.194 peer-group rr
  neighbor 220.144.159.194 description Core2

```

Wednesday, June 06, 2001

```
no auto-summary
!
ip classless
ip route 10.0.0.0 255.0.0.0 Null0
ip route 172.16.0.0 255.240.0.0 Null0
ip route 192.168.0.0 255.255.0.0 Null0
ip route 220.144.128.0 255.255.224.0 Null0
ip tacacs source-interface Loopback0
ip bgp-community new-format
!
logging trap debugging
logging source-interface Loopback0
logging 220.144.159.129
access-list 1 permit 220.144.159.129
access-list 1 permit 220.144.159.130
access-list 1 deny any log
access-list 140 permit ip 220.144.158.0 0.0.0.255 any ! WHAT GETS OUT
access-list 140 deny ip any any log
access-list 141 permit ip host 220.144.158.1 host 220.144.158.2! WHAT GETS IN
access-list 141 permit ip host 220.144.158.2 host 220.144.158.1
access-list 141 permit ip 220.144.159.128 0.0.0.63 220.144.158.0 0.0.0.255
access-list 141 permit icmp any 220.144.158.0 0.0.0.255
access-list 141 permit tcp any 220.144.158.0 0.0.0.255 established
access-list 141 permit tcp any 220.144.158.0 0.0.0.255 eq ident
access-list 141 permit tcp any 220.144.158.0 0.0.0.255 eq 22 ! SSH
access-list 141 permit tcp any 220.144.158.0 0.0.0.255 eq www
access-list 141 permit udp any 220.144.158.0 0.0.0.255 gt 1023
access-list 141 deny ip any any log
access-list 150 permit ip 220.144.159.0 0.0.0.63 any ! WHAT GETS OUT
access-list 150 deny ip any any log
access-list 151 permit ip host 220.144.159.1 host 220.144.159.2! WHAT GETS IN
access-list 151 permit ip host 220.144.159.2 host 220.144.159.1
access-list 151 permit ip 220.144.159.128 0.0.0.63 220.144.159.0 0.0.0.63
access-list 151 permit icmp any 220.144.159.0 0.0.0.63
access-list 151 permit tcp any 220.144.159.0 0.0.0.63 established
access-list 151 permit tcp any 220.144.159.0 0.0.0.63 eq ident
access-list 151 permit udp any 220.144.159.0 0.0.0.63 gt 1023
! 220.144.159.1 is 2ary DNS, NTP
access-list 151 permit tcp any host 220.144.159.1 eq domain
access-list 151 permit udp any host 220.144.159.1 eq domain
access-list 151 permit udp any host 220.144.159.1 eq ntp
! 220.144.159.2 is 2ary MX, DNS cache
access-list 151 permit tcp any host 220.144.159.2 eq smtp
access-list 151 permit tcp any host 220.144.159.2 eq domain
access-list 151 permit udp any host 220.144.159.2 eq domain
! 220.144.159.3 is 1ary MX, FTP & WWW
access-list 151 permit tcp any host 220.144.159.3 eq smtp
access-list 151 permit tcp any host 220.144.159.3 eq ftp
access-list 151 permit tcp any host 220.144.159.3 eq ftp-data
access-list 151 permit tcp any host 220.144.159.3 eq www
! 220.144.159.4 is POP3 server and SMTP relay for DIAL only
access-list 151 permit tcp any host 220.144.159.4 eq pop3
access-list 151 permit tcp 220.144.157.0 0.0.0.255 host 220.144.159.4 eq smtp
! 220.144.159.5 is News reader for DIAL only
access-list 151 permit tcp 220.144.157.0 0.0.0.255 host 220.144.159.5 eq nntp
access-list 151 deny ip any any log
access-list 198 permit ip 220.144.159.128 0.0.0.63 any ! VTY access-list - NOC
access-list 198 permit ip 220.144.159.192 0.0.0.63 any ! Router loopbacks
access-list 198 deny ip any any log
access-list 199 deny ip any any log
!
tacacs-server host 220.144.159.129
tacacs-server key SecretToo
snmp-server community NotTelling RO 1
snmp-server location Somewhere
snmp-server contact Network Operations Centre <noc@net.galaxy>
snmp-server enable traps snmp
snmp-server host 220.144.159.130 SecretToo
banner login ^
  Authorised Access Only
```

```

This system is the property of Galaxy Internet

Disconnect IMMEDIATELY if you are not an authorised user!

Contact noc@net.galaxy +98 765 4321 for help.
^
!
line con 0
  exec-timeout 3 0
  transport preferred none
  transport input none
  transport output telnet
line aux 0
  transport preferred none
  transport input telnet
  transport output telnet
line vty 0 4
  access-class 198 in
  exec-timeout 0 0
  transport preferred none
  transport input telnet ssh
  transport output telnet
!
exception protocol ftp
exception dump 220.144.159.129
ntp authentication-key 1 md5 secretAlso
ntp authenticate
ntp trusted-key 1
ntp source Loopback0
ntp server 220.144.159.193 key 1
ntp server 220.144.159.194 key 1
ntp peer 220.144.159.196 key 1
ntp peer 220.144.159.197 key 1
ntp peer 220.144.159.198 key 1
end

```

## NOC Router

The NOC router is used for to connect the ISP essentials services (such as SYSLOG, TACACS+ and primary DNS) and the operations engineers workstations to the service provider's backbone. The router in this example is assumed to be a 2620 with three fast ethernet ports.

```

version 12.0
service nagle
no service pad
service tcp-keepalives-in
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
service password-encryption
!
hostname NOC-router
!
boot system flash
logging buffered 16384 debugging
aaa new-model
aaa authentication login default tacacs+ enable
aaa authentication enable default tacacs+ enable
aaa accounting exec default start-stop tacacs+
aaa accounting commands 15 default start-stop tacacs+
enable secret shhhhhthisisasecret
!
clock timezone GMT 0
ip subnet-zero
no ip source-route
no ip finger
ip telnet source-interface Loopback0
ip tftp source-interface Loopback0

```

Wednesday, June 06, 2001

```
ip ftp source-interface Loopback0
ip ftp username cisco
ip ftp password shhhhsecret
no ip bootp server
ip domain-name net.galaxy
ip name-server 220.144.159.1
ip name-server 220.144.159.2
ip name-server 219.10.2.1
!
! SSH support
ip ssh time-out 120
ip ssh authentication-retries 3
!
partition flash 2 8 8
!
interface Loopback0
description Loopback interface on NOC
ip address 220.144.159.199 255.255.255.255
no ip directed-broadcast
!
interface FastEthernet0/0
description Core Ethernet (SW1 vlan1 port 6)
ip address 220.144.159.103 255.255.255.240
no ip redirects
no ip directed-broadcast
no ip proxy-arp
!
interface FastEthernet0/1
description Core Ethernet (SW2 vlan2 port 15)
ip address 220.144.159.119 255.255.255.240
no ip redirects
no ip directed-broadcast
no ip proxy-arp
!
interface FastEthernet1/0
description NOC Ethernet
ip address 220.144.159.190 255.255.255.192
ip access-group 130 in
ip access-group 131 out
no ip redirects
no ip directed-broadcast
no ip proxy-arp
!
router ospf 100
network 220.144.159.96 0.0.0.15 area 0
network 220.144.159.112 0.0.0.15 area 0
network 220.144.159.199 0.0.0.0 area 0
network 220.144.159.128 0.0.0.63 area 0
passive-interface FastEthernet1/0
passive-interface Loopback0
log-adjacency-changes
!
ip classless
ip route 10.0.0.0 255.0.0.0 Null0
ip route 172.16.0.0 255.240.0.0 Null0
ip route 192.168.0.0 255.255.0.0 Null0
ip route 220.144.128.0 255.255.224.0 Null0
ip tacacs source-interface Loopback0
ip bgp-community new-format
!
logging trap debugging
logging source-interface Loopback0
logging 220.144.159.129
access-list 1 permit 220.144.159.129
access-list 1 permit 220.144.159.130
access-list 1 deny any log
access-list 130 permit ip 220.144.159.128 0.0.0.63 any ! WHAT GETS OUT
access-list 130 deny ip any any log
access-list 131 permit icmp any 220.144.159.128 0.0.0.63 ! WHAT GETS IN
access-list 131 permit tcp any 220.144.159.128 0.0.0.63 eq ident
```

```

access-list 131 permit tcp any 220.144.159.128 0.0.0.63 established
! 220.144.159.129 is LOGHOST, TACACS+, FTP-dumps, TFTP
access-list 131 permit udp 220.144.159.192 0.0.0.63 host 220.144.159.129 eq syslog
access-list 131 permit udp 220.144.159.192 0.0.0.63 host 220.144.159.129 eq tftp
access-list 131 permit tcp 220.144.159.192 0.0.0.63 host 220.144.159.129 eq ftp
access-list 131 permit tcp 220.144.159.192 0.0.0.63 host 220.144.159.129 eq ftp-data
access-list 131 permit tcp 220.144.159.192 0.0.0.63 host 220.144.159.129 eq tacacs
! 220.144.159.130 is NOC MGMT Workstation
access-list 131 permit udp 220.144.159.192 0.0.0.63 host 220.144.159.130 eq snmp
access-list 131 permit udp 220.144.159.192 0.0.0.63 host 220.144.159.130 eq snmptrap
! 220.144.159.131 is SSH gateway
access-list 131 permit tcp any host 220.144.159.131 eq 22
! 220.144.159.132 is DIAL Radius server (NT running Cisco Secure)
access-list 131 permit udp host 220.144.159.200 host 220.144.159.132 eq 1645
access-list 131 permit udp host 220.144.159.200 host 220.144.159.132 eq 1646
! 220.144.159.180 is Primary DNS
access-list 131 permit udp any host 220.144.159.180 eq domain
access-list 131 permit tcp any host 220.144.159.180 eq domain
access-list 131 deny ip any any log
access-list 198 permit ip 220.144.159.128 0.0.0.63 any ! VTY access-list - NOC
access-list 198 permit ip 220.144.159.192 0.0.0.63 any ! Router loopbacks
access-list 198 deny ip any any log
access-list 199 deny ip any any log
!
tacacs-server host 220.144.159.129
tacacs-server key SecretToo
snmp-server community NotTelling RO 1
snmp-server location Somewhere
snmp-server contact Network Operations Centre <noc@net.galaxy>
snmp-server enable traps snmp
snmp-server host 220.144.159.130 SecretToo
banner login ^
Authorized Access Only

This system is the property of Galaxy Internet

Disconnect IMMEDIATELY if you are not an authorised user!

Contact noc@net.galaxy +98 765 4321 for help.
^
!
line con 0
exec-timeout 3 0
transport preferred none
transport input none
transport output telnet
line aux 0
transport preferred none
transport input telnet
transport output telnet
line vty 0 4
access-class 198 in
exec-timeout 0 0
transport preferred none
transport input telnet ssh
transport output telnet
!
exception protocol ftp
exception dump 220.144.159.129
ntp authentication-key 1 md5 secretAlso
ntp authenticate
ntp trusted-key 1
ntp source Loopback0
ntp server 220.144.159.193 key 1
ntp server 220.144.159.194 key 1
end

```

Wednesday, June 06, 2001

## Access Server

This is the typical configuration of an access server with a connected modem bank. The router in this example is assumed to be a 3640 with the 64 asynchronous ports.

```
version 12.0
service nagle
no service pad
service tcp-keepalives-in
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
service password-encryption
!
hostname Access-router
!
boot system flash
logging buffered 16384 debugging
aaa new-model
aaa new-model
aaa authentication login default group tacacs+ enable
aaa authentication login radius-login group radius
aaa authentication enable default group tacacs+ enable
aaa authentication ppp default none
aaa authentication ppp ppp-auth if-needed group radius
aaa authorization network radius-auth group radius
aaa accounting network default stop-only group radius
enable secret shhhhhthisisasecret
!
clock timezone GMT 0
ip subnet-zero
no ip source-route
no ip finger
ip telnet source-interface Loopback0
ip tftp source-interface Loopback0
ip ftp source-interface Loopback0
ip ftp username cisco
ip ftp password shhhhsecret
no ip bootp server
ip domain-name net.galaxy
ip name-server 220.144.159.1
ip name-server 220.144.159.2
ip name-server 219.10.2.1
!
! SSH support
ip ssh time-out 120
ip ssh authentication-retries 3
!
partition flash 2 8 8
!
interface Loopback0
description Loopback interface on Access-Router
ip address 220.144.159.200 255.255.255.255
no ip directed-broadcast
!
interface FastEthernet0/0
description Core Ethernet (SW1 port 6)
ip address 220.144.159.104 255.255.255.240
no ip redirects
no ip directed-broadcast
no ip proxy-arp
!
interface FastEthernet0/1
description Core Ethernet (SW2 port 6)
ip address 220.144.159.120 255.255.255.240
no ip redirects
no ip directed-broadcast
no ip proxy-arp
!
```

```

interface Group-Async1
 ip unnumbered Loopback0
 no ip directed-broadcast
 encapsulation ppp
 ip tcp header-compression passive
 async mode interactive
 peer default ip address pool ppp-dialin
 no cdp enable
 ppp max-bad-auth 3
 ppp reliable-link
 ppp authentication chap ms-chap pap ppp-auth
 ppp authorization radius-auth
 group-range 65 128
!
router ospf 100
 log-adjacency-changes
 network 220.144.159.200 0.0.0.0 area 0
 network 220.144.159.96 0.0.0.15 area 0
 network 220.144.159.112 0.0.0.15 area 0
 passive-interface Loopback0
!
router bgp 64511
 network 220.144.157.0 mask 255.255.255.128
 no synchronization
 bgp log-neighbor-changes
 neighbor rr peer-group
 neighbor rr remote-as 64511
 neighbor rr update-source Loopback0
 neighbor rr password BGPsecretPW
 neighbor rr send-community
 neighbor 220.144.159.193 peer-group rr
 neighbor 220.144.159.193 description Core1
 neighbor 220.144.159.194 peer-group rr
 neighbor 220.144.159.194 description Core2
 no auto-summary
!
ip local pool ppp-dialin 220.144.157.1 220.144.157.64
!
ip classless
ip route 10.0.0.0 255.0.0.0 Null0
ip route 172.16.0.0 255.240.0.0 Null0
ip route 192.168.0.0 255.255.0.0 Null0
ip route 220.144.128.0 255.255.224.0 Null0
ip route 220.144.157.0 255.255.255.128 Null0
ip tacacs source-interface Loopback0
no ip http server
ip bgp-community new-format
!
ip radius source-interface Loopback0
logging trap debugging
logging facility local5
logging source-interface Loopback0
logging 220.144.159.129
access-list 1 permit 220.144.159.129
access-list 1 permit 220.144.159.130
access-list 1 deny any log
!
! Access-list 155 is a dynamic access used to restrict e-mail only clients to the
! POP3 and MAIL relay servers
access-list 155 permit tcp any host 220.144.159.4 eq pop3
access-list 155 permit tcp any host 220.144.159.4 eq smtp
access-list 155 deny ip any any
!
! Access-list 156 is a dynamic access list used to restrict clients with expired
! subscriptions to the billing server so they can renew their subscription - it is
! the default unless clients have paid up ☺
access-list 156 permit tcp any host 220.144.129.36 eq www
access-list 156 deny ip any any
access-list 198 permit ip 220.144.159.128 0.0.0.63 any ! VTY access-list - NOC

```

Wednesday, June 06, 2001

```
access-list 198 permit ip 220.144.159.192 0.0.0.63 any ! Router loopbacks
access-list 198 deny ip any any log
access-list 199 deny ip any any log
!
dialer-list 1 protocol ip permit
!
tacacs-server host 220.144.159.129
tacacs-server key SecretToo
snmp-server community NotTelling RO 1
snmp-server location Somewhere
snmp-server contact Network Operations Centre <noc@net.galaxy>
snmp-server enable traps snmp
snmp-server host 220.144.159.130 SecretToo
radius-server host 220.144.159.132 auth-port 1645 acct-port 1646
radius-server key AlsoSecret
radius-server vsa send accounting
radius-server vsa send authentication
banner login ^
Authorised Access Only

This system is the property of Galaxy Internet

Disconnect IMMEDIATELY if you are not an authorised user!

Contact noc@net.galaxy +98 765 4321 for help.
^
!
line con 0
exec-timeout 3 0
transport preferred none
transport input none
transport output telnet
line 65 128
session-timeout 5 output
exec-timeout 1 0
autoselect during-login
autoselect ppp
login authentication radius-login
modem Dialin
autocommand ppp
length 25
transport input all
escape-character NONE
autohangup
stopbits 1
flowcontrol hardware
line aux 0
transport preferred none
transport input telnet
transport output telnet
line vty 0 4
access-class 198 in
exec-timeout 0 0
transport preferred none
transport input telnet ssh
transport output telnet
!
exception protocol ftp
exception dump 220.144.159.129
ntp authentication-key 1 md5 secretAlso
ntp authenticate
ntp trusted-key 1
ntp source Loopback0
ntp server 220.144.159.193 key 1
ntp server 220.144.159.194 key 1
ntp peer 220.144.159.196 key 1
ntp peer 220.144.159.197 key 1
ntp peer 220.144.159.198 key 1
end
```

## Out-of-band Console Server

This is the typical configuration of an access server which has been configured as an out of band management device. The router in this case is a 2611 router with 32 asynchronous ports which have been wired to the consoles of the equipment used in Figure 37 and described in the above configurations.

```

version 12.1
service nagle
no service pad
service tcp-keepalives-in
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
service password-encryption
!
hostname cs2611
!
logging buffered 16384 debugging
aaa new-model
aaa authentication login default tacacs+ enable
aaa authentication exec default tacacs+ enable
aaa accounting exec default start-stop tacacs+
aaa accounting commands 15 default start-stop tacacs+
enable secret shhhhhthisisasecret
!
clock timezone GMT 0
ip subnet-zero
no ip source-route
no ip finger
ip rcmd source-interface Loopback0
ip telnet source-interface Loopback0
ip tftp source-interface Loopback0
ip ftp source-interface Loopback0
ip ftp username cisco
ip ftp password shhhhsecret
no ip bootp server
ip domain-name net.galaxy
ip name-server 220.144.159.1
ip name-server 220.144.159.2
ip name-server 219.10.2.1
!
!
interface Ethernet0/0
description Galaxy Net Core LAN
ip address 220.144.159.189 255.255.255.192
no ip route-cache
no ip mroute-cache
!
ip classless
ip route 0.0.0.0 0.0.0.0 220.144.159.190
ip tacacs source-interface Ethernet0/0
!
! map IP addresses to reverse telnet ports - easier for humans to deal with
! than trying to remember port numbers
ip alias 220.144.159.160 2065
ip alias 220.144.159.161 2066
ip alias 220.144.159.162 2067
ip alias 220.144.159.163 2068
ip alias 220.144.159.164 2069
ip alias 220.144.159.165 2070
ip alias 220.144.159.166 2071
ip alias 220.144.159.167 2072
ip alias 220.144.159.168 2073
ip alias 220.144.159.169 2074
ip alias 220.144.159.170 2075
ip alias 220.144.159.171 2076
ip alias 220.144.159.172 2077

```

Wednesday, June 06, 2001

```
ip alias 220.144.159.173 2078
ip alias 220.144.159.174 2079
ip alias 220.144.159.175 2080
ip alias 220.144.159.176 2081
ip alias 220.144.159.177 2082
ip alias 220.144.159.178 2083
ip alias 220.144.159.179 2084
!
no ip http server
!
logging trap debugging
logging source-interface Loopback0
logging 220.144.159.129
access-list 1 permit 220.144.159.129
access-list 1 permit 220.144.159.130
access-list 1 deny any log
access-list 198 permit ip 220.144.159.128 0.0.0.63 any ! VTY access-list - NOC
access-list 198 permit ip 220.144.159.192 0.0.0.63 any ! Router loopbacks
access-list 198 deny ip any any log
access-list 199 deny ip any any log
!
tacacs-server host 220.144.159.129
tacacs-server key SecretToo
snmp-server community NotTelling RO 1
snmp-server location Somewhere
snmp-server contact Network Operations Centre <noc@net.galaxy>
snmp-server enable traps snmp
snmp-server host 220.144.159.130 SecretToo
banner login ^
Authorised Access Only

This system is the property of Galaxy Internet

Disconnect IMMEDIATELY if you are not an authorised user!

Contact noc@net.galaxy +98 765 4321 for help.
^
!
line con 0
  exec-timeout 3 0
  transport preferred none
  transport input none
  transport output telnet
line aux 0
  transport preferred none
  transport input telnet
  transport output telnet
line 65
  location Console of SW1
  no exec
  transport preferred none
  transport input telnet
line 66
  location Console of SW2
  no exec
  transport preferred none
  transport input telnet
line 67
  location Console of Border-router
  no exec
  transport preferred none
  transport input telnet
line 68
  location Console of Core1
  no exec
  transport preferred none
  transport input telnet
line 69
  location Console of Core2
  no exec
```

```
transport preferred none
transport input telnet
line 70
location Console of Gateway1
no exec
transport preferred none
transport input telnet
line 71
location Console of Gateway2
no exec
transport preferred none
transport input telnet
line 72
location Console of Service1
no exec
transport preferred none
transport input telnet
line 73
location Console of Service 2
no exec
transport preferred none
transport input telnet
line 74
location Console of NOC router
no exec
transport preferred none
transport input telnet
line 75
location Console of SSH Gateway
no exec
transport preferred none
transport input telnet
line 76
location Console of LOGHOST/TACACS+/NOC Workstation
no exec
transport preferred none
transport input telnet
line 77
location Console of Network Management Workstation
no exec
transport preferred none
transport input telnet
line 78
location Console of RADIUS server
no exec
transport preferred none
transport input telnet
line 79
location Console of Primary DNS
no exec
transport preferred none
transport input telnet
line 80
location Console of Secondary DNS
no exec
transport preferred none
transport input telnet
line 81
location Console of Secondary MX
no exec
transport preferred none
transport input telnet
line 82
location Console of Primary MX, ftp, www
no exec
transport preferred none
transport input telnet
line 83
location Console of POP3 server
```

Wednesday, June 06, 2001

```
no exec
transport preferred none
transport input telnet
line 84
location Console of DIAL News
no exec
transport preferred none
transport input telnet
line 85 128
location Spare
no exec
transport preferred none
transport input telnet
line vty 0 4
access-class 198 in
exec-timeout 0 0
transport preferred none
transport input telnet ssh
transport output telnet
!
exception protocol ftp
exception dump 220.144.159.129
ntp authentication-key 1 md5 secretAlso
ntp authenticate
ntp trusted-key 1
ntp source Loopback0
ntp server 220.144.159.193 key 1
ntp server 220.144.159.194 key 1
end
```

---

## APPENDIX 4 – ROUTE FLAP DAMPING

---

### *BGP Flap Damping Configuration*

Recommended route flap damping parameters for use by ISPs were composed into a document by the RIPE Routing Working Group and are available at <http://www.ripe.net/docs/ripe-210.html> and <http://www.ripe.net/docs/ripe-210.ps> in Postscript form. These values are used by many European and US ISPs, and are based on the operational experience gained in the industry.

**NB.** The ripe-210 document is currently being rewritten and is due for release shortly. The rewrite is aimed at bringing the document up to date, and correcting the parameters for /24 damping. The corrected values are included below.

The configuration examples are reproduced here for convenience – the values have been updated to include recent changes in the locations of the root nameservers.

### IP Access-List Example

This is a configuration example using the IP access-list, similar to what is quoted in the RIPE-210 document. We don't recommend this method any longer – IP prefix-lists have long superseded access-list for prefix filtering. See the next section for the prefix-list configuration.

Note that the access-list 180 covering the root-nameserver and access-list 184 covering the global toplevel domain nameserver networks has been updated with the most recent values. It is strongly recommended to check these addresses and network prefixes announced to the Internet before implementing the filters. (You could use the Unix command “`dig . ns`” to find the nameserver addresses, and “`sh ip bgp x.x.x.x`” on a router to find the size of the prefix being advertised in the Internet Routing table.)

```
router bgp 65280
  bgp dampening route-map RIPE210-flap-damp

  ! no flap damping for special user defined networks defined in access-list 183
  route-map RIPE210-flap-damp deny 10
  match ip address 183
  ! no flap damping for root nameserver networks in access-list 180
  route-map RIPE210-flap-damp deny 20
  match ip address 180
  ! no flap damping for gtld nameserver networks in access-list 184
  route-map RIPE210-flap-damp deny 30
  match ip address 184
  ! flap damping for all the /24 and longer prefixes
  route-map RIPE210-flap-damp permit 30
  match ip address 181
  set dampening 30 820 3000 60
  ! flap damping for all /22 and /23 prefixes
  route-map RIPE210-flap-damp permit 40
  match ip address 182
  set dampening 15 750 3000 45
  ! flap damping for all remaining prefixes
  route-map RIPE210-flap-damp permit 50
  set dampening 10 1500 3000 30
  !

  ! Access Lists for route flap damping as per RIPE-210 definition
  ! with updated root server networks
  access-list 180 permit ip host 198.41.0.0 host 255.255.255.0 ! A J root servers
```

Wednesday, June 06, 2001

```
access-list 180 permit ip host 128.9.0.0 host 255.255.0.0      ! B root server
access-list 180 permit ip host 192.33.4.0 host 255.255.255.0 ! C root server
access-list 180 permit ip host 128.8.0.0 host 255.255.0.0    ! D root server
access-list 180 permit ip host 192.203.230.0 host 255.255.255.0! E root server
access-list 180 permit ip host 192.5.4.0 host 255.255.254.0   ! F root server
access-list 180 permit ip host 192.112.36.0 host 255.255.255.0! G root server
access-list 180 permit ip host 128.63.0.0 host 255.255.0.0    ! H root server
access-list 180 permit ip host 192.36.148.0 host 255.255.255.0! I root server
access-list 180 permit ip host 193.0.14.0 host 255.255.255.0 ! K root server
access-list 180 permit ip host 198.32.64.0 host 255.255.255.0 ! L root server
access-list 180 permit ip host 202.12.27.0 host 255.255.255.0 ! M root server
access-list 180 deny   ip any any
! match /24 prefixes
access-list 181 permit ip any 255.255.255.0 0.0.0.255
access-list 181 deny   ip any any
! match /22 and /23 prefixes
access-list 182 permit ip any 255.255.252.0 0.0.3.255
access-list 182 deny   ip any any
! match special prefixes
access-list 183 permit ip host 169.223.0.0 host 255.255.0.0
access-list 183 deny   ip any any
! updated gtld name server networks
access-list 184 permit ip host 192.5.6.0 host 255.255.255.0    ! A
access-list 184 permit ip host 192.33.14.0 host 255.255.255.0 ! B
access-list 184 permit ip host 192.26.92.0 host 255.255.255.0 ! C
access-list 184 permit ip host 192.31.8.0 host 255.255.255.0  ! D
access-list 184 permit ip host 192.12.94.0 host 255.255.255.0 ! E
access-list 184 permit ip host 192.35.51.0 host 255.255.255.0 ! F
access-list 184 permit ip host 192.42.93.0 host 255.255.255.0 ! G
! access-list 184 permit ip host xxxx                          ! no H
access-list 184 permit ip host 192.36.144.0 host 255.255.255.0! I
access-list 184 permit ip host 210.132.96.0 host 255.255.224.0! J
access-list 184 permit ip host 213.177.192.0 host 255.255.248.0! K
access-list 184 permit ip host 192.41.162.0 host 255.255.255.0! L
access-list 184 permit ip host 202.153.112.0 host 255.255.240.0! M
access-list 184 deny ip any any
!
```

## IP Prefix-List Example

Prefix-lists can also be used; and indeed should be used. The above example has been rewritten using the ip prefix-list commands available in 11.1CC and 12.0 software releases. This makes the configuration more readable, if not more intuitive. Again, remember to check the root nameserver networks for any changes before implementing these. It is also worth checking the BGP routing table to ensure that these networks are still announced with the prefix lengths listed below.

```
router bgp 65280
  bgp dampening route-map RIPE210-flap-damp
  !
  ip prefix-list my-nets description Networks we don't suppress
  ip prefix-list my-nets seq 5 permit 169.223.0.0/16
  !
  ip prefix-list suppress22 description Damping of /22 and /23 prefixes
  ip prefix-list suppress22 seq 5 permit 0.0.0.0/0 ge 22 le 23
  !
  ip prefix-list suppress24 description Damping of /24 and longer prefixes
  ip prefix-list suppress24 seq 5 permit 0.0.0.0/0 ge 24
  !
  ip prefix-list rootns description Root-nameserver networks
  ip prefix-list rootns permit 198.41.0.0/24      ! A & J
  ip prefix-list rootns permit 128.9.0.0/16       ! B
  ip prefix-list rootns permit 192.33.4.0/24      ! C
  ip prefix-list rootns permit 128.8.0.0/16       ! D
  ip prefix-list rootns permit 192.203.230.0/24   ! E
  ip prefix-list rootns permit 192.5.4.0/23      ! F
  ip prefix-list rootns permit 192.112.36.0/24    ! G
```

```

ip prefix-list rootns permit 128.63.0.0/16      ! H
ip prefix-list rootns permit 192.36.148.0/24    ! I
ip prefix-list rootns permit 193.0.14.0/24     ! K
ip prefix-list rootns permit 198.32.64.0/24    ! L
ip prefix-list rootns permit 202.12.27.0/24    ! M
!
ip prefix-list gtldns description Global Top Level Domain Servers
ip prefix-list gtldns permit 192.5.6.0/24      ! A
ip prefix-list gtldns permit 192.33.14.0/24    ! B
ip prefix-list gtldns permit 192.26.92.0/24    ! C
ip prefix-list gtldns permit 192.31.80.0/24    ! D
ip prefix-list gtldns permit 192.12.94.0/24    ! E
ip prefix-list gtldns permit 192.35.51.0/24    ! F
ip prefix-list gtldns permit 192.42.93.0/24    ! G
! ip prefix-list gtldns permit xxxx           ! No H
ip prefix-list gtldns permit 192.36.144.0/24   ! I
ip prefix-list gtldns permit 210.132.96.0/19   ! J
ip prefix-list gtldns permit 213.177.192.0/21  ! K
ip prefix-list gtldns permit 192.41.162.0/24   ! L
ip prefix-list gtldns permit 202.153.112.0/20  ! M
!
route-map RIPE210-flap-damp deny 10
 match ip address prefix-list my-nets
!
route-map RIPE210-flap-damp deny 20
 match ip address prefix-list rootns
!
route-map RIPE210-flap-damp deny 30
 match ip address prefix-list gtldns
!
route-map RIPE210-flap-damp permit 40
 match ip address prefix-list suppress24
 set dampening 30 820 3000 60
!
route-map RIPE210-flap-damp permit 50
 match ip address prefix-list suppress22
 set dampening 15 750 3000 45
!
route-map RIPE210-flap-damp permit 60
 set dampening 10 1500 3000 30
!

```

---

---

## APPENDIX 5 – TRAFFIC ENGINEERING TOOLS

---

---

### *Internet Traffic and Network Engineering Tools*

As a follow-up on how to track where your customer are going on the Net, here is a list of tools on the Internet that can be used to pull in statistics from your network. Most ISP do not use things like HP OpenView, SunNet Manager, Cisco Works, or Spectrum to manage their networks. These network management packages are great for the enterprise LANs, but do not have the simple scaleable tools needed for ISP networks. Instead, ISPs pull together different, mostly public domain, tools and use UNIX scripts to generate charts and reports (via GNU Plot) for traffic engineering and Quality of Service management.

Note that one Cisco specific solution is to use NetFlow. NetFlow is available on 75XX, 72XX and 7000/RSP running the 11.1CC, 12.0 and later software releases. It is available on the 3600 series and higher platforms from 12.0(1) and is available on the smaller platforms from 12.0(2)T. There are a few white papers on the Cisco Web pages. Check them out and/or let the authors know if you would like more information.

Stan Barber gave a talk at the February NANOG entitled *Monitoring your Network with Freely Available Statistics Reporting Tools*. The slides are at <http://www.academ.com/nanog/feb1998/nettools.html>.

### CAIDA

CAIDA (Co-operative Associations for Internet Data Analysis) has a very comprehensive page listing a lot of tools and pointers. This CAIDA effort is supported through assistance provided by US National Science Foundation and Cisco Systems.

CAIDA Measurement Tool Taxonomy <http://www.caida.org/Tools/taxonomy.html>

### NetScarf/Sicon

New project by MERIT to get a picture on what is happening on the Internet. They now have an NT version:

<http://nic.merit.edu/~netscarf/>

### NeTraMet/NetFlowMet

The old one and one of the best for TCP/IP flow analysis. SingNet used NeTraMet on an Intel PC with BSD Unix and a Digital FDDI card. The results were dumped on to a system that did all the flow analysis and the results were posted on to an internal Web server. Recently a capability has been added to analyse Cisco NetFlow records – NetFlowMet is part of this package now.

<http://www.auckland.ac.nz/net/NeTraMet/>

### Cflowd

One of the better tools out there for NetFlow analysis is cflowd. Our customers developed this tool for their own use. It is free and located at:

<http://www.caida.org/Tools/Cflowd/>

Other scripts based on cflowd are located at:

<http://buckaroo.xo.com/CFLOWD/>

The key Cisco documents on NetFlow are constantly being updated (because we are adding new features and functionality all the time). Do a keyword search on CCO to find all the documentation on NetFlow.

NetFlow tools (flowdata.h, fdrecorder.c, fdplayback.c, fdg.c) that were used to build cflowd are located on Cisco's FTP site:

<ftp://ftp-eng.cisco.com/ftp/NetFlow/fde/README>

## MRTG

Multi Router Traffic Grapher – a Perl script based package to create graphics of interface loading on the routers. Saves you having to create UNIX scripts to do the same thing! It works on UNIX (requires the more recent versions of Perl 5) and there is a version for **Windows NT** too! The package also contains contributed tools allowing you to monitor CPU loads, disk space, temperature, and many other functions which an ISP can use to watch its network.

<http://www.mrtg.org>

<http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/mrtg.html>

<http://mailer.gu.se/traffic/mrtg.html>

## RRDTool

RRDTool is from the author of MRTG, and is designed to be a more powerful and flexible system for graphing collected statistics. It is not meant to be a full replacement for MRTG, and future versions of MRTG are planned to sit on top of RRDTool.

<http://ee-staff.ethz.ch/~oetiker/webtools/rrdtool/>

## Vulture

SNMP Vulture is a tool to do long term SNMP data collection and analysis of routers and other similar devices. Vulture has a number of features that make it suitable for a number of different tasks:

- Per-interface configuration. Different data may be collected on each defined interface.
- Template based configuration. Different sorts of interfaces may require recording different information – whoever heard of collisions on a serial interface?
- Configurable per-router community strings.
- Web based graphical browsing of stored data.
- Built-in data archival mechanism for stale data.

Vulture is written in Perl version 5, and uses the CMU version 2 libraries to do the low level snmp access. The Vulture distribution includes both the CMU libraries and a small module to connect the libraries to Perl. The browser interface also requires the generally available Gnuplot and PBMPlus utilities to generate graphical output.

<http://www.vix.com/vulture/>

## CMU SNMP

Free command line SNMP software. Create your own scripts to pull down the MIB variables you wish to look at (like information on the links).

<http://www.boutell.com/lsm/lsmbyid.cgi/000033>

[http://hpux.petech.ac.za/hppd/hpux/Networking/Admin/cmu\\_snmp-1.2u/](http://hpux.petech.ac.za/hppd/hpux/Networking/Admin/cmu_snmp-1.2u/)

Wednesday, June 06, 2001

CMU SNMP Archives – <ftp://lancaster.andrew.cmu.edu/pub/snmp-dist/>

## UCD SNMP (the successor to CMU SNMP)

CMU SNMP has not been updated for some time now. UCD has taken over the project and their release contains a port and modified code of the CMU 2.1.2.1 snmp agent. It has been modified to allow extensibility quickly and easily. It is far from the best and most configurable system; but hey, it is free.

<ftp://ftp.ece.ucdavis.edu/pub/snmp/ucd-snmp.README>

<ftp://ftp.ece.ucdavis.edu/pub/snmp/ucd-snmp.tar.gz>

## Gnuplot

Gnuplot is a useful public domain graphing tool. If configuring MRTG is too much, or you need to graph something else quickly, this is probably the way to do it.

[http://www.cs.dartmouth.edu/gnuplot\\_info.html](http://www.cs.dartmouth.edu/gnuplot_info.html)

An example of what can be done is on the NOAA web site:

<http://www.erl.noaa.gov/network.html>

<http://www.erl.noaa.gov/ahsia/webshop/overview.html>

A Comprehensive list of other Public Domain SNMP Software can be found at:

<http://wwwsnmp.cs.utwente.nl/ietf/impl.html>

## NETSYS

NetSys is a suite of tools developed to map, track, and **predict** problems on your network. People who use it really like its capabilities of mapping your existing network and then working projections of what will happen to the links under specific conditions. Here are more details on the NETSYS suite of tools:

**NETSYS Connectivity Tools.** The first in a series of simulation-based planning and problem-solving products for network managers, analysts, and designers. The Connectivity Tools assist network planners with problem solving, design, and planning activities focusing on network connectivity, route, and flow analysis.

[http://www.cisco.com/warp/public/734/nslms4/cnmgr\\_ds.htm](http://www.cisco.com/warp/public/734/nslms4/cnmgr_ds.htm)

**NETSYS Performance Tools.** These tools allow users to create a network baseline from configuration and performance data, and then analyze the interactions between traffic flow, topology, routing parameters, and Cisco IOS features. Users can also diagnose and solve operational problems, test "what if" scenarios, tune the network configurations for improved performance, and plan for incremental network changes.

[http://www.cisco.com/warp/public/734/nslms4/pfmgr\\_ds.htm](http://www.cisco.com/warp/public/734/nslms4/pfmgr_ds.htm)

**NETSYS Advisor.** NETSYS Advisor works with both the Connectivity and Performance Tools to quickly isolate network problems and identify solutions using an exclusive model-based reasoning technology.

<http://www.cisco.com/warp/public/734/nslms4/>

## SysMon

Sysmon is a network monitoring tool designed by Jared Mauch to provide high performance and accurate network monitoring. Currently supported protocols include SMTP, IMAP, HTTP, TCP, UDP, NNTP, and PING tests. The latest version can be found at:

<http://puck.nether.net/sysmon/>

## Treno

Treno (a tools develop end to end performance) information and you can try it out from PSC via this WWW forms interface:

<http://www.psc.edu/~pscnoc/treno.html>

## Scotty – Tcl Extensions for Network Management Applications

Scotty is the name of a software package which allows to implement site specific network management software using high-level, string-based APIs. The software is based on the [Tool Command Language](#), which simplifies the development of portable network management scripts. The scotty source distribution includes two major components. The first one is the Tnm Tcl Extension, which provides access to network management information sources. The second component is the Tkined network editor which provides a framework for an extensible network management system.

This tool used to be called “tkined”. Scotty is now the ‘correct’ name for the project.

<http://wwwsnmp.cs.utwente.nl/~schoenw/scotty/>

## NetSaint

NetSaint is a web based network monitoring tool used by several ISPs. It can monitor servers and the network, and send e-mail or pages if there are problems. It’s written in C and designed to run on Linux.

<http://netsaint.sourceforge.net/>

## Big Brother

Another useful tool, similar to NetSaint, is Big Brother. Source code is available for Unix and Linux – clients include NT, Novell, and Mac.

<http://www.bb4.com/>

**With all of this software available, it is *not* expensive or difficult to collect and analyse data on your network. You can create your own tools and run them on the many Intel based UNIX systems (i.e. Linux, BSDI, etc.).**

Wednesday, June 06, 2001

## ***Other Useful Tools to Manage your Network***

### **Traceroute**

The standard Unix traceroute command is probably the most basic tool available for fault detection and diagnosis in the Internet today. And the utility has spawned several enhancements, for example pptraceroute from the PRIDE project and the more advanced traceroute which comes with Linux. Cisco IOS, of course, has its own traceroute command, using the routing table in the router to provide detailed information. Source for traceroute can be found at <ftp://ftp.ripe.net/tools/traceroute.tar.Z>. For those without access to any of these, or if you require a traceroute from another part of the Internet, an excellent public traceroute facility is at <http://www.traceroute.org>, a catalogue of traceroute “Looking Glasses” around the world.

### **Whois**

Another standard Unix command invaluable for use on the Internet is whois. Again there have been many enhancements to whois itself, by the PRIDE project, and the RIPE whois software which has been optimised to support lookups on Internet databases using the RIPE database software. The latter is available at <ftp://ftp.ripe.net/tools/ripe-whois-tools-2.4.tar.gz>.

### **RTRMon – A Tool for Router Monitoring and Manipulation**

The RTR system currently comes with three programs, rtrmon, rtrpass, and rtrlogin. rtrmon – for “router monitor” – is the core of the system, using predefined actions to log in to routers, issues a command, process the output, archive the result, and possibly mail reports. It is designed to provide the framework for a variety of potential monitoring tasks and be readily extensible with new reporting code if the built-in methods are insufficient for complex analysis. It can even update router configurations, despite its “monitor” moniker.

The rtrpass program is meant to provide an easy interface to a more secure method of storing passwords. Since rtrmon needs to be able to provide passwords to routers to log in to them and to gain “enable” privileges, they have to be accessible on the hosts on which rtrmon is running. To reduce the risks associated with this, rtrpass manages a PGP-encrypted file for each rtrmon user that contains his own password and his enable password, if he has enable access. Passwords can be controlled on a per router basis if desired.

rtrlogin is used to login to routers, automatically logging in using your username and password, getting “enable” privileges, setting the terminal length and width to the size of your window, and running any commands you have saved in your personal login file. The session can then be used interactively.

<http://www.vix.com/rtrmon/>

### **Cisco’s MIBs**

All the Cisco SNMP MIBs are publicly available. If you have commercial SNMP management packets and/or shareware-freeware packets, you may need to go and grab the MIB. Here is the FTP site:

<ftp://ftp.cisco.com/pub/mibs/>

### **Replacement SYSLOG Daemons**

#### ***SYSLOG-NG***

Syslog-NG is a replacement for the standard SYSLOG found on most Unix systems. It is supported on Linux, BSD, AIX, HP-UX and Solaris. It gives a much enhanced configuration scheme, which allows the filtering of messages based not only on priority/facility pairs, but also on message content. Regular expressions can also be used to direct log streams to different destinations. It also supports forwarding logs on TCP, with hash protected log files planned for a future release.

<http://www.balabit.hu/products/syslog-ng/>

## **SECURE SYSLOG (ssyslog)**

SECURE SYSLOG (ssyslog) is available for UNIX systems. Designed to replace the syslog daemon, ssyslog implements a cryptographic protocol called PEO-1 that allows the remote auditing of system logs. Auditing remains possible even if an intruder gains superuser privileges in the system, the protocol guarantees that the information logged before and during the intrusion process cannot be modified without the auditor (on a remote, trusted host) noticing.

<http://www.core-sdi.com/ssyslog>

## **Overall Internet Status and Performance Tools**

Many people are asking just how well the Internet is performing. Ever since the NSFnet was decommissioned, there has been no one place to understand the performance and traffic profiles on the Internet. Yet, there are people trying to figure out how to do this. The following lists are sites, projects, and software that is attempting to get a true “big picture” of what is happening on the Internet. ISPs can elect to join one or more of these programs to add more data to these projects.

### **NetStat**

A tool that pings various parts of the Internet from various locations on the Internet, collects the data, and provides an average response time on the major US backbone. The tool is based on ping.

<http://netstat.net/>

## **What other ISPs are doing...**

Here are examples of what ISPs from all over Internet are using to manage their network. Randy Bush [randy@psg.com](mailto:randy@psg.com) asked major ISPs in the US on the NANOG mailing list what they used for traffic analysis. Here is Randy’s summary. Notice the number of UNIX script based tools...

*We do SNMP polling ever 15 minutes at SESQUINET on every line over which we have administrative control and over every peering point. We produce a daily report on errors and usage. We are getting ready to switch to Vulture or NetScarf (or some combo) to give us more interactive information.*

*We perform measurement of certain basic network parameters, such as usage (bandwidth used / total bandwidth) and line error rates on all of our non-customer links. We perform CPU usage, memory usage, and environmental monitoring of all our routers. We also perform the line usage and error rate on all customer lines. We monitor all of our customers' routers unless they say otherwise, and notify them of any problems. Finally, we monitor select points throughout the Internet (root name servers, etc.) on a 4 times an hour basis using pings. We accomplish this monitoring using the following items: an in-house built package that uses SNMP, traceroute, and ping to provide graphs and tabular statistical information. We use Cabletron’s Spectrum for a quick network overview.*

*We do SNMP MIB-II stuff, plus the cflowd stuff and something we call ‘mxd’ (measures round trip times, packet loss and potential reason, etc. from a whole bunch of different points in our network to a bunch of other points in our network... We use it to create delay matrices, packet loss reports and other reports). There are some other things, but these are the biggies.*

Wednesday, June 06, 2001

*The mxd thing was originally just sort of a toy for neat reports, but in the last year it's become a critical tool for measuring delay variance for one of our VPDN customers that does real-time video stuff (and is to some extent helping us figure out where we've got delay jitter and why; on the other hand it's also raising more questions ☺).*

*Since most of my professional career has been in the enterprise world, I can offer you what we used to measure availability to our mail servers, web servers, DNS servers, etc., at one of my previous employers.*

*We employed several application tests, along with network performance tests. Our primary link was via UUnet, a burstable T1. We purchased an ISDN account from another local provider, who wasn't directly connected to UUnet. Probably a good example of a joe-average-user out there.*

*Every 5 minutes, we measured round-trip response times to each of the servers and gateway router (via ping) and recorded it. We also had application tests, such as DNS lookups on our servers, timing sendmail test mails to a /dev/null account, and time to retrieve the whole home page.*

*This wasn't meant to be a really great performance monitoring system; it was actually meant to 1) check how our availability looked from a "joe user" perspective on the net (granted, reachability/availability wasn't perfect because it was only one point in the net) and 2) look at response time trends / application trends to see if our hardware/software was cutting it.*

*We use a traffic flow monitoring system from Kaspia Systems. ([www.kaspia.com](http://www.kaspia.com)) The Kaspia product collects all sorts of data from router ports and RMON probes, stores the data and performs various trend analysis. We collect traffic flow, router CPU usage and router memory information plus various errors. There is a data reduction process which runs once a day, and a very nifty web interface. The product is not cheap, but the system definitely fills a void here.*

*Maybe I should organize a talk on what we are doing with it for an upcoming NANOG? As an old instrumentation engineer, I think the basis of our use of the tool is pretty solid. Plus, I actually developed a means for calibration of the accuracy of the flow data. Have not had time yet to work out a validation for the trends, but I'll get to it one of these decades.*

*Also, the Kaspia people will give you a thirty day trial on their product at no charge.*

*For non-intrusive stuff, we keep a log of all interface status changes on our routers, and we pull five-minute byte-counts inbound and outbound on each interface, which we graph against port speed. Watching the graphs for any sort of clipping of peaks gives a pretty good indication of problems, and watching for shifts of traffic between ports on parallel paths likewise.*

*As for intrusive testing, we do a three-packet min-length ping to the LAN-side port of each of our customers' routers once each five minutes, and follow that up with additional attempts if those three are lost. We log latency, and if we have to follow up with a burst, we log loss rate from the burst. Pinging through to the LAN port obviously lets us know when CPE routers konk out, as occasionally we see hung routers that still have operational WAN ports talking to us, likewise, simply watching VC-state isn't a reliable enough indicator of the status of the remote router. Plus it tells you if the customer has kicked the Ethernet transceiver off their equipment, for instance. Wouldn't matter to you, probably, but our demarc is all the way out at their WAN port, since we own and operate our customers' CPE.*

*I think a bit about what more we could be doing; flows-analysis and whatnot... It's nice to think about, and eventually we'll get around to it, but programmer-time is relatively precious, and other things have higher priority, since the current system works and tends to tell us most of what we seem to need to know to provide decent service.*

*We place quite a bit of emphasis on network stats. currently we have about 3 years of stats online, and are working on converting our inhouse engine to an rdbms so we can more easily perform trend analysis. besides kaspia, other commercial packages include trendsnmpp ([www.desktalk.com](http://www.desktalk.com)) and concord's packages ([www.concord.com](http://www.concord.com)). Our in house stuff is located at <http://netop.cc.buffalo.edu/> if you are curious about what we do.*

*We are Neanderthals right now – we use a hacked rcisco to feed data to nocol. We watch bandwidth (separately as well) on key links - and also watch input errors and interface transitions (for nocol) – all done with perl and expect-like routines, parsing 'sho int's every few minutes.*

*Emergency stuff goes through nocol; bandwidth summaries are mailed to interested parties overnight.*

*We have running here now the MRTG package that generate some fancy graphics, but in my opinion these graphics are useless and looking in detail to some of the reports they are not accurate, several of our clients request the raw data but this package only maintain few raw data just to generate the graphs, mean also useless.*

*In the past we use to have also a kind of ASCII reports (Vikas wrote some of the scripts and programs) generated from information obtained using the old snmp tool set developed by nysernet but I guess that nobody maintained the config files and I believe that the snmp library routines used aren't working fine.*

*So, I need to invest some time to provide a fast solution to this, I'll appreciate your help to identify some useful package or directions about how to generate some good looking and consistent reports.*

*We have been using the MRTG package which is basically a special SNMP agent that queries the routers for stats and then does some nice graphing of the data on the web.*

*SNMP queries with a heavily-modified version of MRTG from the nice guy in Germany. Works very nicely. We have recently installed NetScarf 2.0, and are contemplating merging NetScarf 3.0 with the MRTG front end.*

*I'm researching whether I can rewrite Steve Corbato's fastpoll program using the fastsnmp library from the NetScarf people. I think this will allow fastpoll to scale better. I've successfully written a quick C program that uses the library to collect the required data for a router – now I've just got to make it so we can manage it easily (i.e. auto-generated config files from our databases).*

*My goal is to be able to collect 1-2 minute period data on all links that are greater than 10 Mbps – 15 minutes data for everything else. The 2 minute collection period will allow to scale up to 280Mbps before experiencing two counter roll-overs within a polling interval. Hopefully that will hold us over the interface counters are available as Counter64 objects via SNMPv2 (if that ever happens).*

*BTW – what fastpoll collects now is ifInOctets, ifOutOctets, ifInUcastPkts, ifOutUcastPkts, ifInErrors and ifOutDiscards. Rather than storing the raw counters, it calculates the rate by taking the delta and dividing by the period. Getting the accurate period is actually the hard part – I am having SNMP send me the uptime of the router in each query and using that to calculate the interval between polls and to detect counter resets due to reboots. The other trick to handle is the fact that, while IOS updates the SNMP counters for process-switched packets as they are routed, it looks like the counter for SSE switched packets on C70X0 routers only get updated once every 10 seconds.*

---

---

## APPENDIX 6 – EXAMPLE ISP ACCESS SECURITY MIGRATION PLAN

---

---

What follows is one example of how and ISP can migrate their network equipment (routers, switches, and NAS) from a state where telnet access is open to the outside world to the point where only specific authorised workstations are allowed access to the telnet prompt.

Unfortunately, at the time this version was written, most ISPs still do not take these simple precautions to help secure their network. This section is specifically addressed to those ISPs who have not take the extra time undertake the basics. A simple procedure that draws a security circle around an ISP's network and then slowly narrows the circle tighter and tighter until just the authorised IP addresses are included in the VTY's ACL.

### *Phase One – Close off access to everyone outside your CIDR block.*

The first step for any ISPs securing their network is to close telnet access off to everyone but authorised workstations. This is done in phases. The first phase is to create a standard ACL that would permit telnet from IP addresses in the ISP's CIDR block. This ACL is used with the VTY's `access-class` command to ensure the source IP address of any telnet packet coming to the VTY port matches the ACL.

Why just the ISPs CIDR block? First, it's easy for a ISP who is not doing this technique now, to implement. The ISP does not have to worry about locking out staff members who need access to the router from different parts of the network. Hence, it can be done with the least worry that it will effect the ISP's operations. Second, it limits the threat for the entire Internet to just IP addresses inside the ISP's CIDR block. Minimising risk is one of the fundamental tenants of security. Finally, since the ISP is beginning with no protection, this incremental step insures everything is working before deepening the security configurations on the network. Phase One's theme is to get the ball rolling. Limiting access to just those IP addresses in the ISP's CIDR block gets the ball rolling.

Figure 38 is an example of an ISP's network. The allocated CIDR blocks are 169.223.0.0/16 and 211.255.0.0/19<sup>34</sup>. Phase One is to create an ACL that can be used with the VTY's `access-class` command to restrict telnet to IP addresses with the CIDR block.

```
aaa new-model
aaa authentication login ISP local
!
username Cisco1 password 7 11041811051B13
!
access-list 3 permit 211.255.0.0 0.0.31.255
access-list 3 permit 169.223.0.0 0.0.255.255
access-list 3 deny any
!
line vty 0 4
 access-class 3 in
 exec-timeout 5 0
 transport preferred none
 login authentication ISP
 history size 256
```

#### **Example 6 – Simple VTY Config to limit access to just the ISP's CIDR Block**

The configuration in Example 6 is used on all routers in the network. Similar configurations should be used on the switches in the network. Any staff workstations/servers should also use appropriate tools to limit telnet access to the workstations/server's resources<sup>35</sup>.

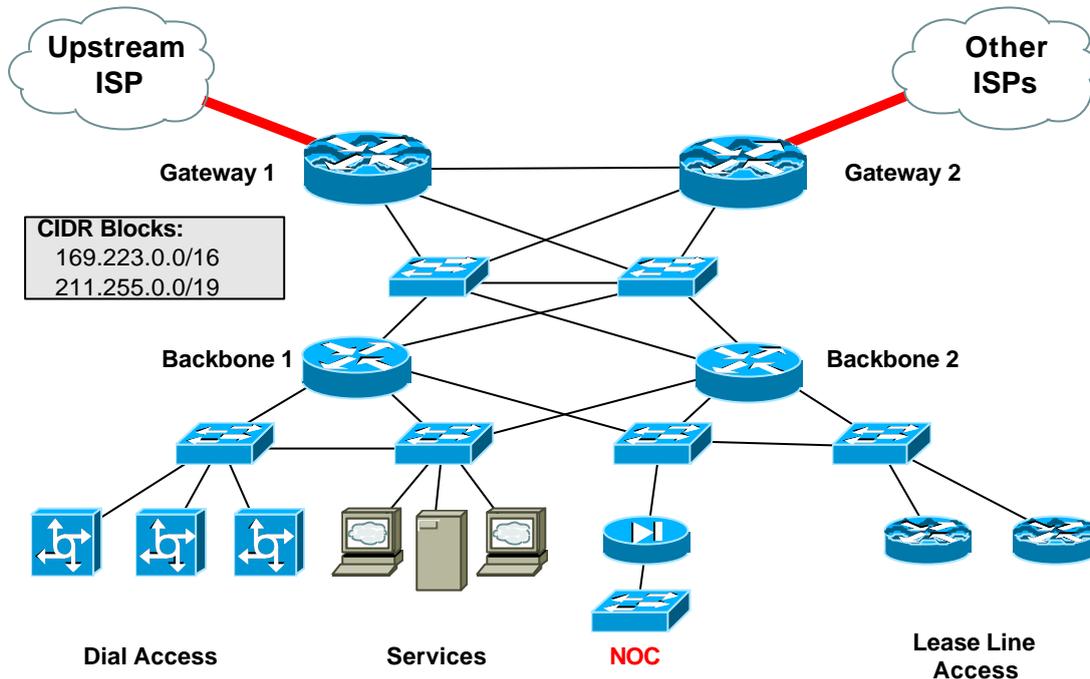


Figure 38 – ISP Network Example

## Phase Two – Add Anti-Spoofing Filters to your peers

Limiting the security risk through restricted telnet access is only the first step. The next step would be to insure parties outside the ISP's network would not be able to *spoof* source addresses from the ISP's CIDR block. There are several forms of source address spoofing and telnet sequence number hijacking attacks that can penetrate the VTY's access-class protections. To minimise the risk of these sort of attacks, an ISP can place anti-spoofing filters at the edges of their network.

As highlighted in the section on Egress and Ingress Filtering, anti-spoofing filters are used to insure that any address coming from the Internet into the ISP's network does not contain a source address from the ISP's network. For example, if a packet from the Internet with a source address of 211.255.1.1 comes into the ISP in Figure 39, the anti-spoofing filter would drop the packet.

**Remember! No one from the general Internet should be sending you packets with a source address from your own network!**

Few ISPs implement anti-spoofing packet filters. The key reason given is possible performance impact. Yes, applying packet filters to any router may cause a performance impact. Yet, another essential tenant of security is balancing the trade offs. Sacrificing some performance to minimise the security risk to valuable network resources<sup>36</sup> is a logical trade off. Especially with the new improvements in packet per second (PPS) that the latest Cisco IOS code offers ISPs.<sup>37</sup>

<sup>34</sup> Apologies if these network represent a real allocation. They were pulled from the APNIC blocks as an example, not to portray any real life network.

<sup>35</sup> Tools like TCP Wrapper are well known and have a role in an ISP's overall security architecture.

<sup>36</sup> Spoofing attacks are more likely to target workstation and server resources. These resources would likely depend on tools like TCP Wrapper – these wrappers can also be bypassed by spoofing attacks. So, placement of anti-spoofing filters protects the entire network, not just the routers.

<sup>37</sup> Distributed Switching (FIB), Cisco Express Forwarding (CEF), Distributed NetFlow, and other improvements in the 11.1CC and 12.0S code trains.

## Where to place the anti-spoofing packet filters?

Place anti-spoofing filters at the edge of an ISP's network. This usually means the router(s) that interconnect with other ISPs. Routers *Gateway1* and *Gateway2* are examples in Figure 39. Any attempt at spoofing from the core of the Internet (i.e. the upstream ISP) and/or an ISP peer connection would be dropped on the inbound interface.

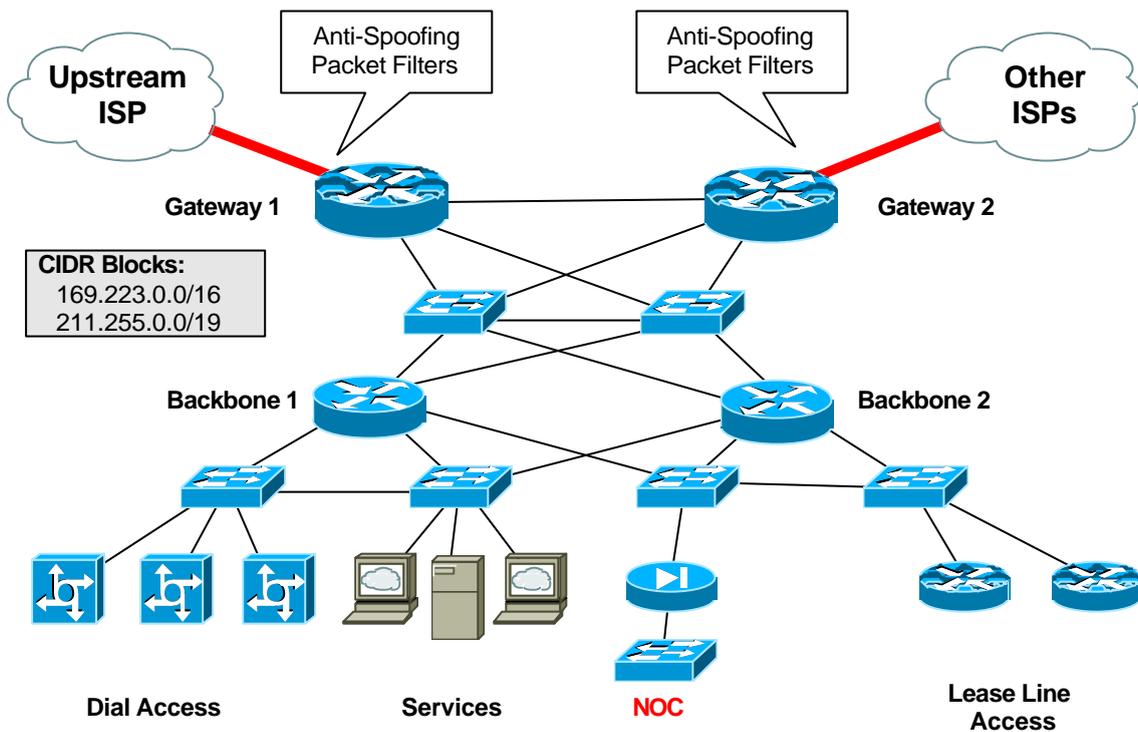


Figure 39 – Applying Anti-Spoofing Filters

Example 7 highlights a typical anti-spoofing filter. Notice that there are two CIDR blocks used in the example – 169.223.0.0/16 and 211.255.0.0/19 two lines of seven in the ACL 111. Here is what the other lines do:

- deny ip 127.0.0.0 0.255.255.255 – This is the loopback address for TCP workstations, PCs, and servers. It should not be transmitted over the Internet. If it is, then it is either a broken TCP stack or someone trying to break into a resource.
- deny ip 10.0.0.0 0.255.255.255 – RFC 1918 Private Address Space. Cisco advocates RFC 1918 Private Address Space use in enterprise networks in conjunction with Network Address Translation (NAT). Any packets with RFC1918 addresses in their source are either from a broken NAT implementation or part of a spoofing attack.
- deny ip 172.16.0.0 0.15.255.255 – RFC 1918 Private Address Space. Ibid.
- deny ip 192.168.0.0 0.0.255.255 – RFC 1918 Private Address Space. Ibid.
- deny ip 169.223.0.0 0.0.255.255 – One of the example ISP's CIDR blocks.
- deny ip 211.255.0.0 0.0.31.255 – The other CIDR Block.
- permit ip any any – Permit normal packets.

Each line with a *deny* has a *log* option turned on. This will take any matches and include them in the internal and output to syslog (if the router has it configured). The *log* option is not used on the last *permit*. Good packets do not need to be logged.

```

Router Gateway1
!
interface hssi 0/1
  description 16Mbps link to our upstream provider
  bandwidth 16384
  ip access-group 111 in
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
!
access-list 111 deny ip 127.0.0.0 0.255.255.255 any log
access-list 111 deny ip 10.0.0.0 0.255.255.255 any log
access-list 111 deny ip 172.16.0.0 0.15.255.255 any log
access-list 111 deny ip 192.168.0.0 0.0.255.255 any log
access-list 111 deny ip 169.223.0.0 0.0.255.255 any log
access-list 111 deny ip 211.255.0.0 0.0.31.255 any log
access-list 111 permit ip any any

Router Gateway2
!
interface serial 0/1
  description Compressed 2Mbps bi-lateral peer to our neighboring country
  bandwidth 2048
  ip access-group 111 in
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
!
access-list 111 deny ip 127.0.0.0 0.255.255.255 any log
access-list 111 deny ip 10.0.0.0 0.255.255.255 any log
access-list 111 deny ip 172.16.0.0 0.15.255.255 any log
access-list 111 deny ip 192.168.0.0 0.0.255.255 any log
access-list 111 deny ip 169.223.0.0 0.0.255.255 any log
access-list 111 deny ip 211.255.0.0 0.0.31.255 any log
access-list 111 permit ip any any

```

### Example 7 – Anti-Spoofing Configuration Example

## *Phase Three – Close off network equipment to unauthorised access*

Once Phase One and Phase Two is completed, work can begin on Phase Three – closing off access to everyone except the NOC Staff. The speed from which an ISP moves from the first two phases to the third phase is dependent on the confidence of the ISP's engineers. Some may wish to monitor the operational impact of the first two phases before incrementing another layer of security. Others may bypass Phase One and immediately restrict access only to the NOC Staff. Either way works. Each ISP needs to develop plans that suite their environment.

There are two basic steps to necessary to close telnet access to just the NOC Staff. First, identify the IP address block for the NOC's network. Figure 40 shows the NOC's network behind a firewall using IP block 211.255.1.0/24. Second, modify the ACLs for the VTY's access-class with the addresses assigned to the NOC. Example 8 highlights this modification.

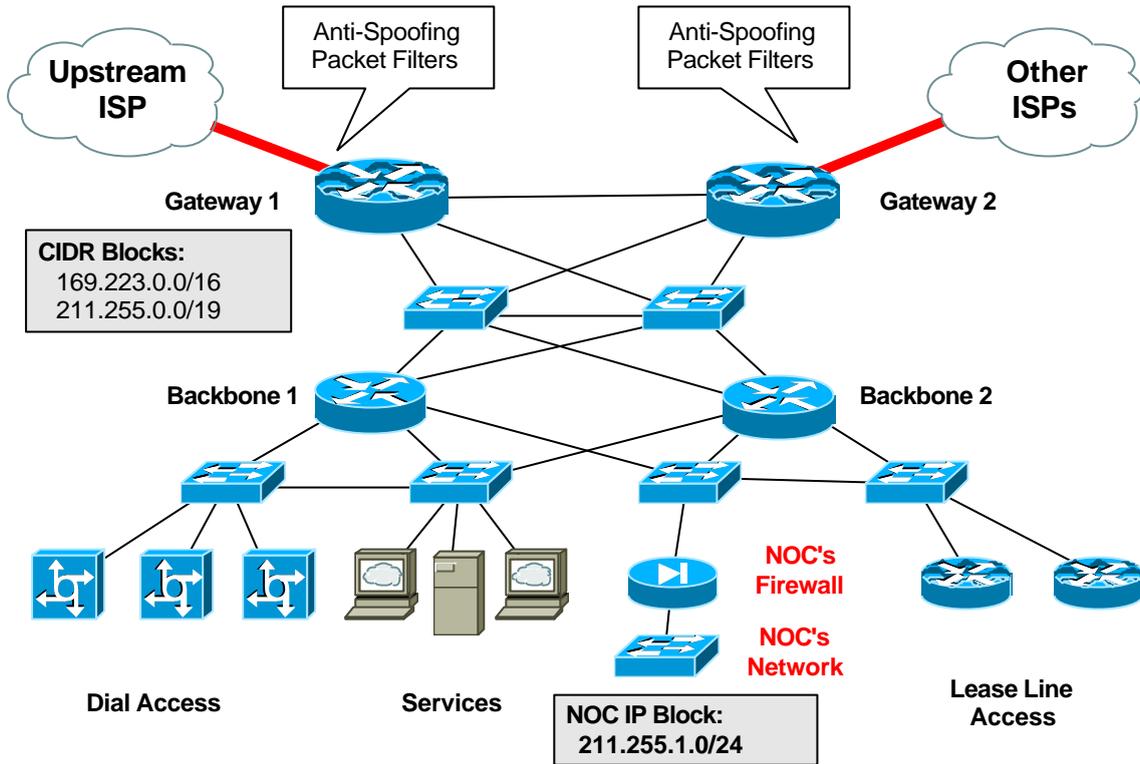


Figure 40 – Closing off access to everyone except the NOC Staff

```
aaa new-model
aaa authentication login ISP local
!
username Cisco1 password 7 11041811051B13
!
access-list 3 permit 211.255.1.0 0.0.0.255
access-list 3 deny any
!
line vty 0 4
access-class 3 in
exec-timeout 5 0
transport preferred none
login authentication ISP
history size 256
```

Example 8 – ACLs with Telnet Access closed to all but the NOC's Network