



Router Security and Infrastructure Protection



Agenda

- Introduction to Core Security

 - Denial of Service (DoS) and Worm Review

 - Six-Phase Methodology

- Device Best Practices

 - Router Security

 - Routing Protocol Security

 - Planes, Paths, and Punts

 - Receive ACL

 - Control Plane Policing

 - Control Plane Protection

 - Management Plane Protection

Agenda (Cont.)

- Infrastructure Security

 - RFC 2827/BCP 38

 - Infrastructure ACLs

 - Flexible Packet Matching

- Network Telemetry

 - SNMP, RMON and Their ilk

 - NetFlow for Security Purposes

Agenda (Cont.)

- Traceback Techniques

 - NetFlow Traceback Techniques

 - Attract and Analyze: Sinkholes

- Reacting to Attacks

 - Reacting with ACL

 - Reacting with BGP

 - Packet Scrubbing

Simple Methodology

- Simple methodology—expanding the scope

Best practices to:

Protect the device

Protect the infrastructure

- With a solid foundation in place, we turn our attention to leveraging the network itself as a security toolkit

Denial of Service (DoS) and Worm



What Is Core Security?

- Often thought of as “SP Security”

What is an SP today?

- Internal networks are no longer truly internal

Tunneling

VPN

Worms, worms, worms

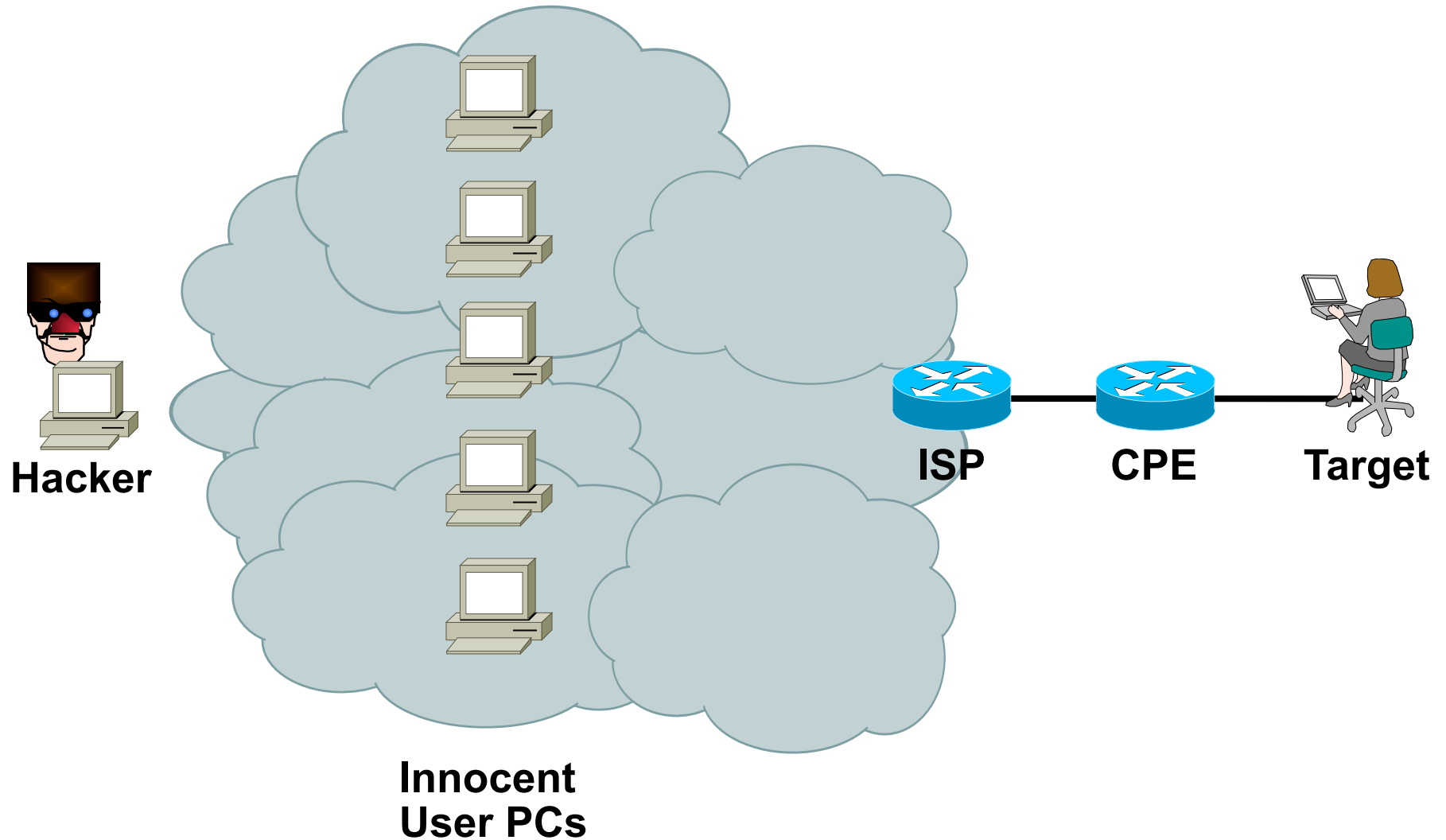
- The infrastructure is critical; if we can't protect it, nothing else matters

Edge security initiatives abound: NAC, 802.1X, HIPS (CSA), personal firewalls, etc.

Denial of Service Attacks

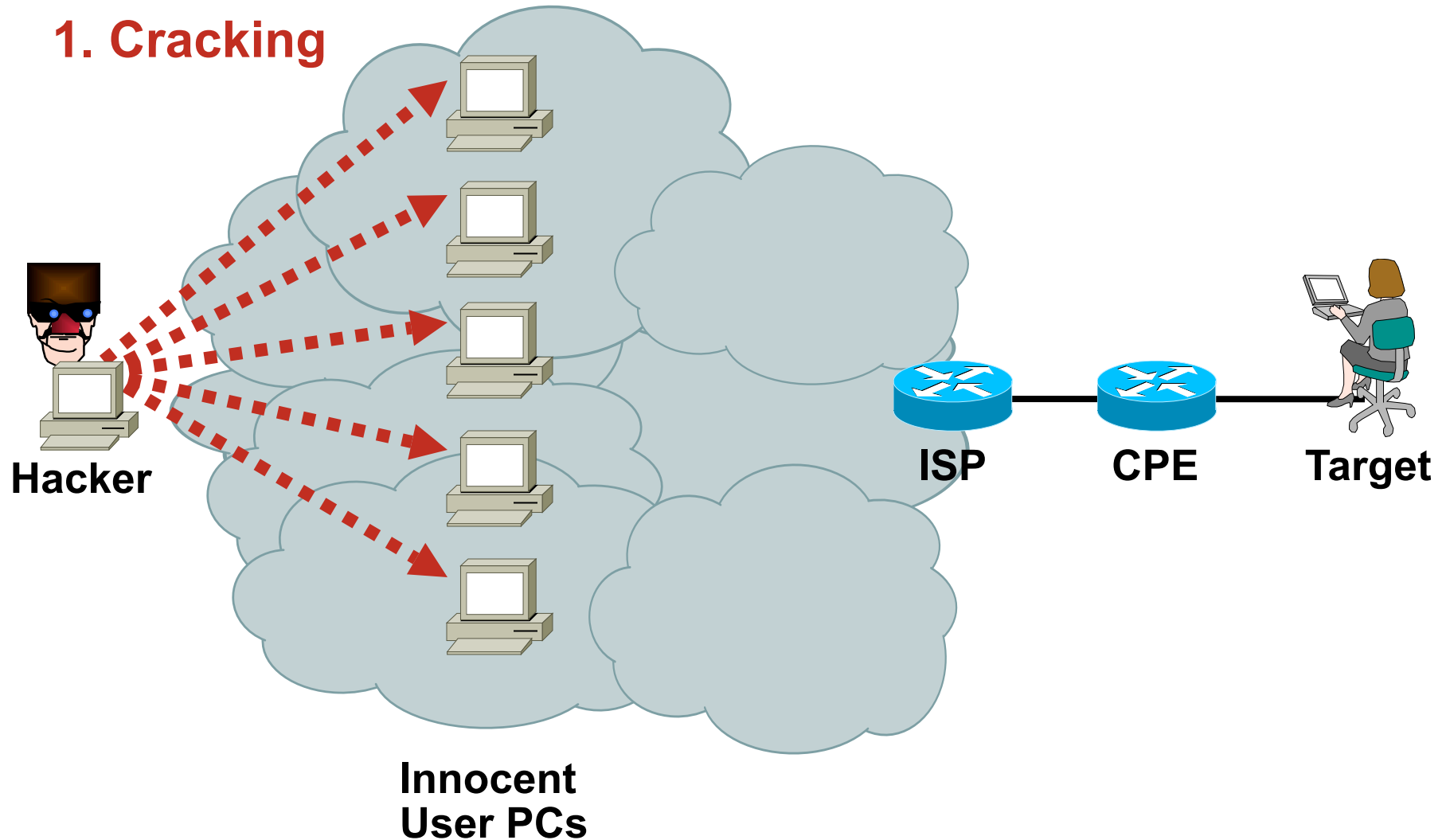
- We understand intrusions (patch, patch, patch ;-))
- What about DoS? Do “the right things” and still suffer
- The vast majority of modern DoS attacks are distributed
DDoS IS DoS
- DoS is often driven by financial motivation
DoS for hire :-(
Economically-driven miscreant community
- DoS cannot be ignored; your business depends on effective handling of attacks

DoS: The Procedure



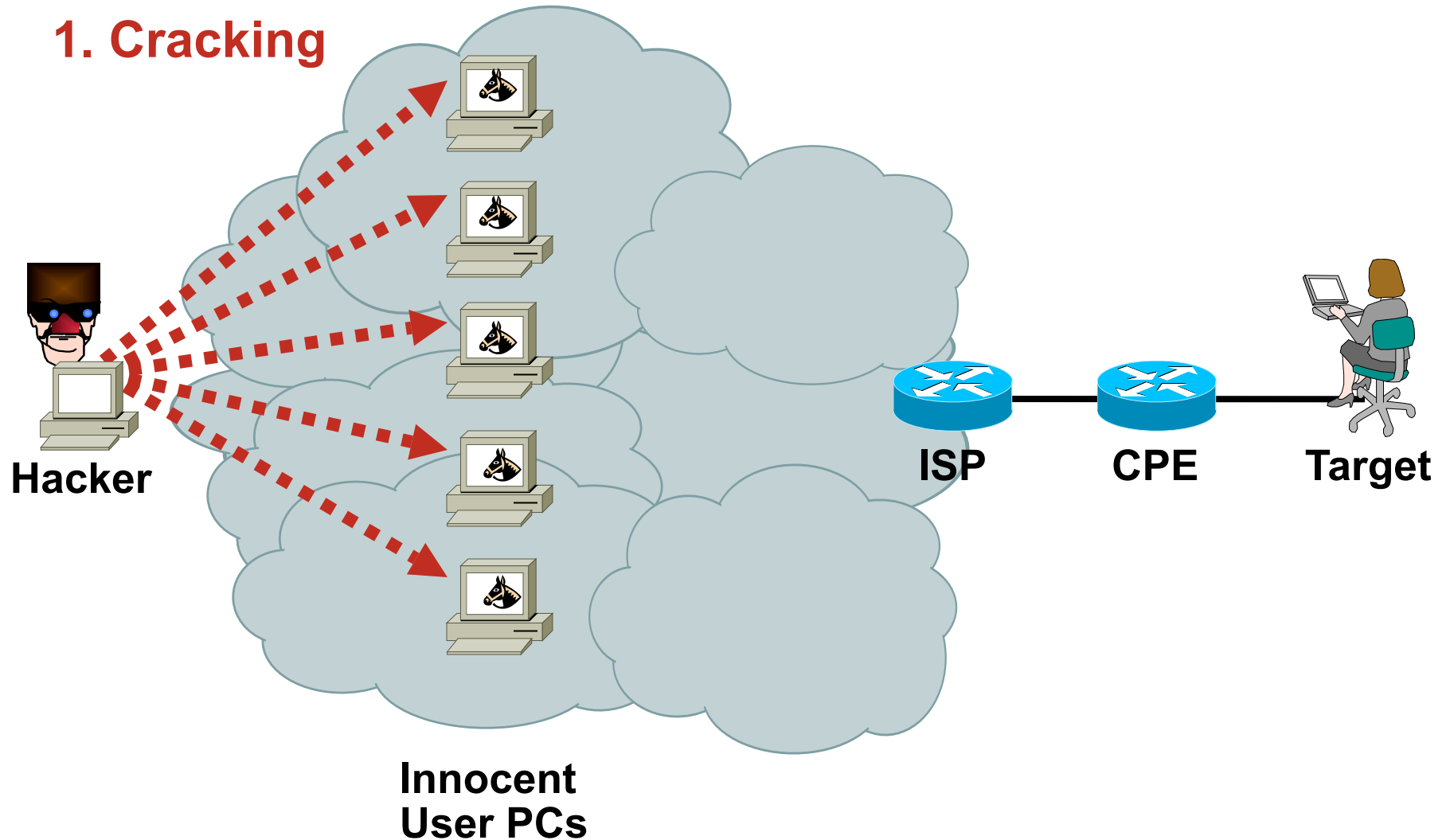
DoS: The Procedure

1. Cracking



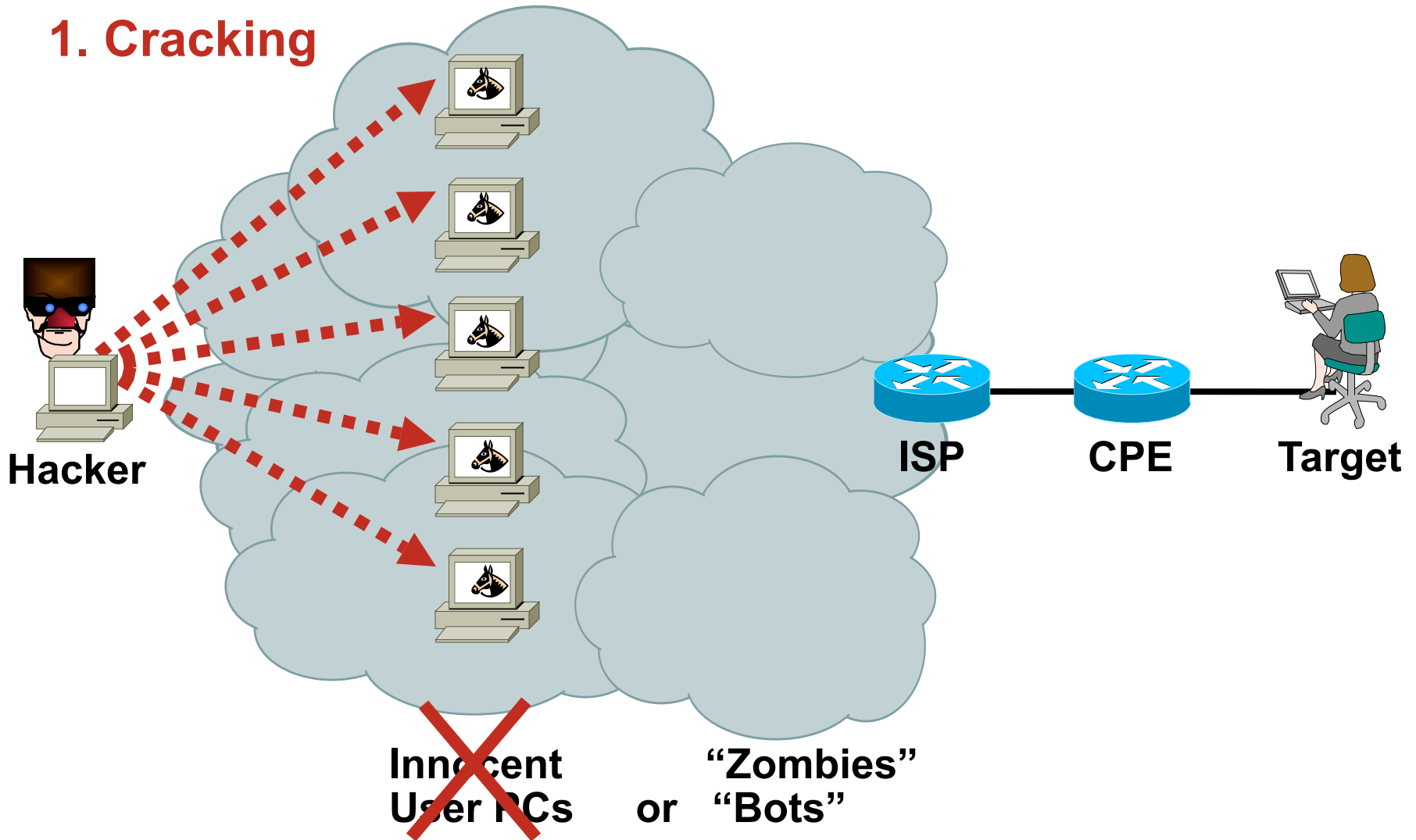
DoS: The Procedure

1. Cracking



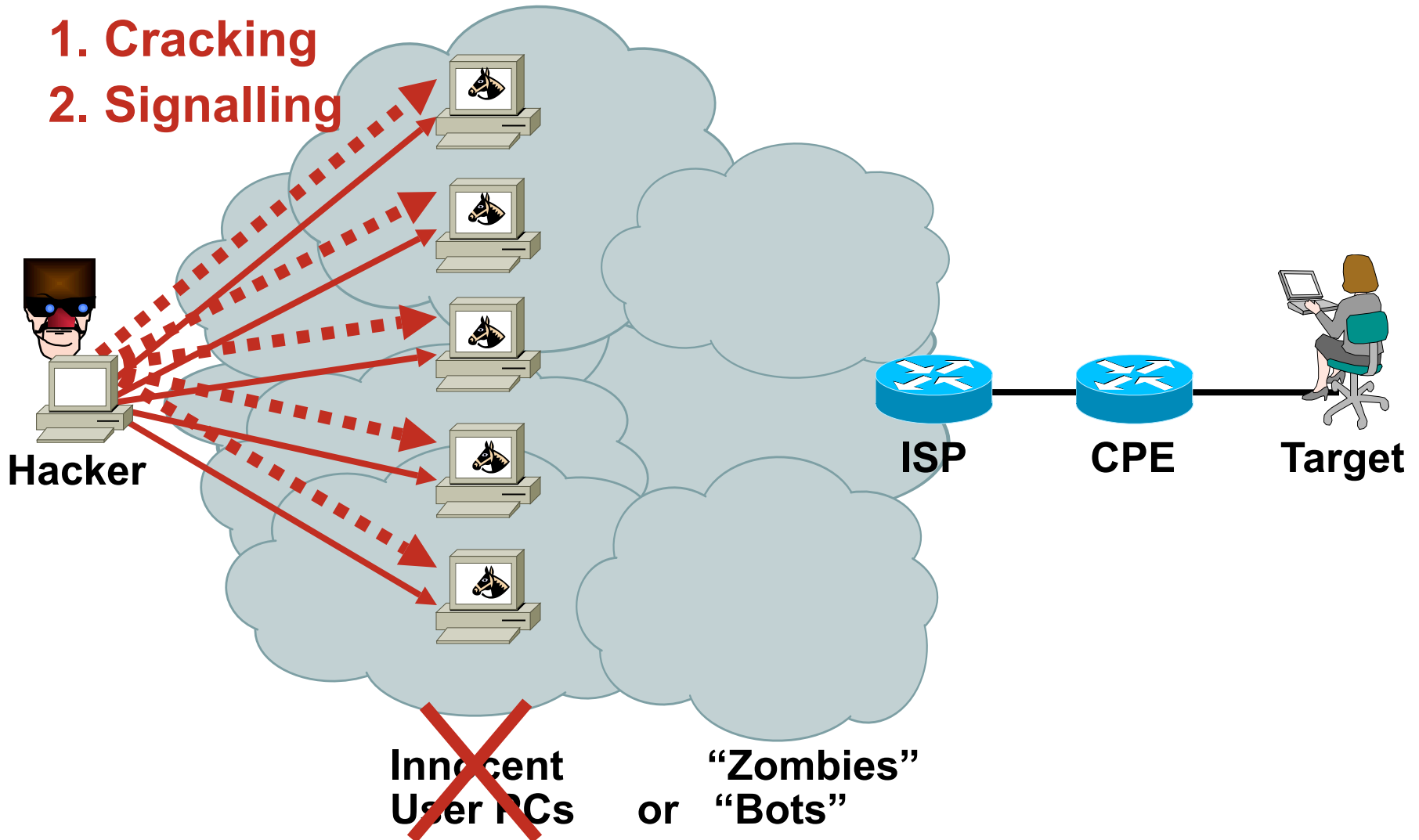
DoS: The Procedure

1. Cracking



DoS: The Procedure

1. Cracking
2. Signalling

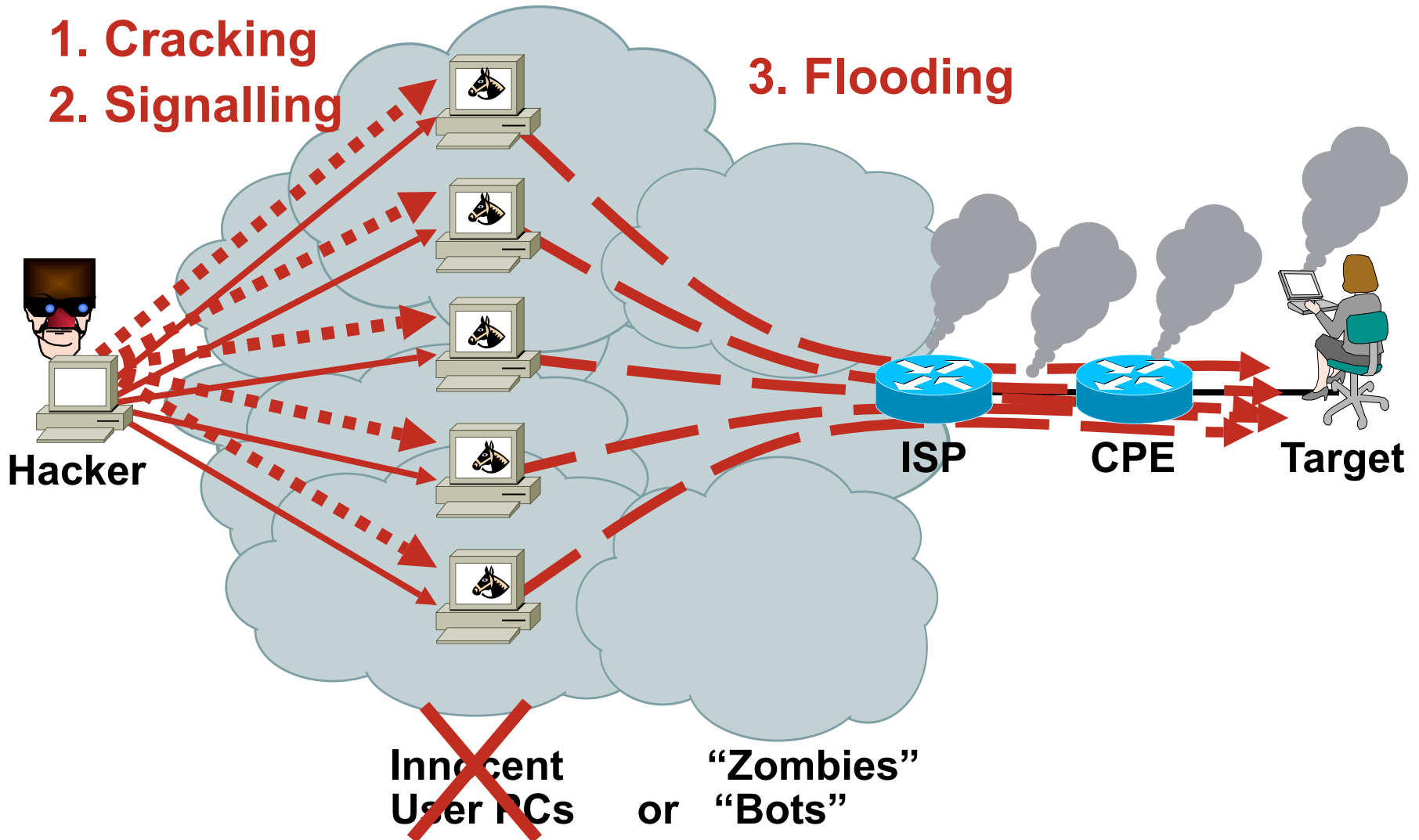


DoS: The Procedure

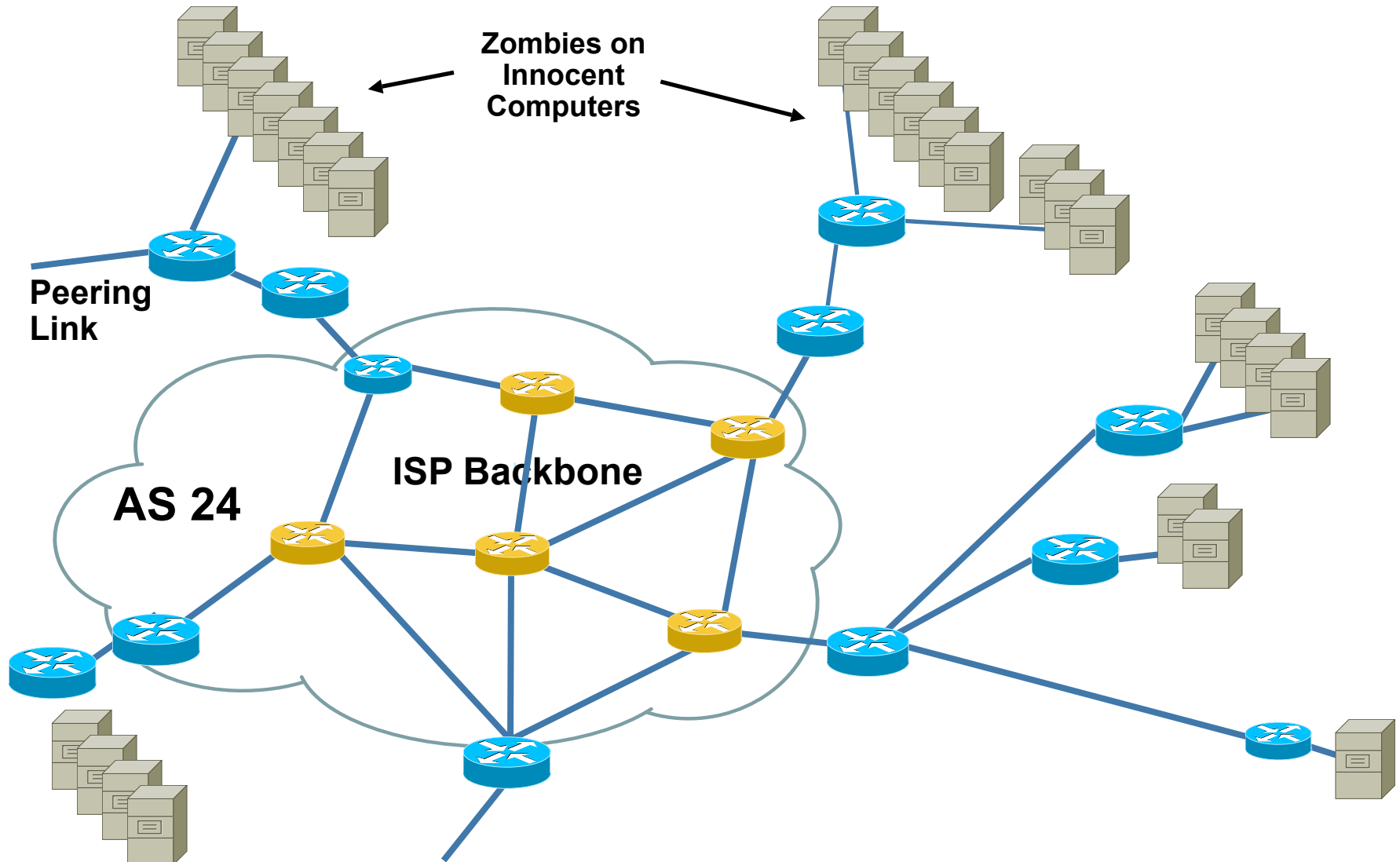
1. Cracking

2. Signalling

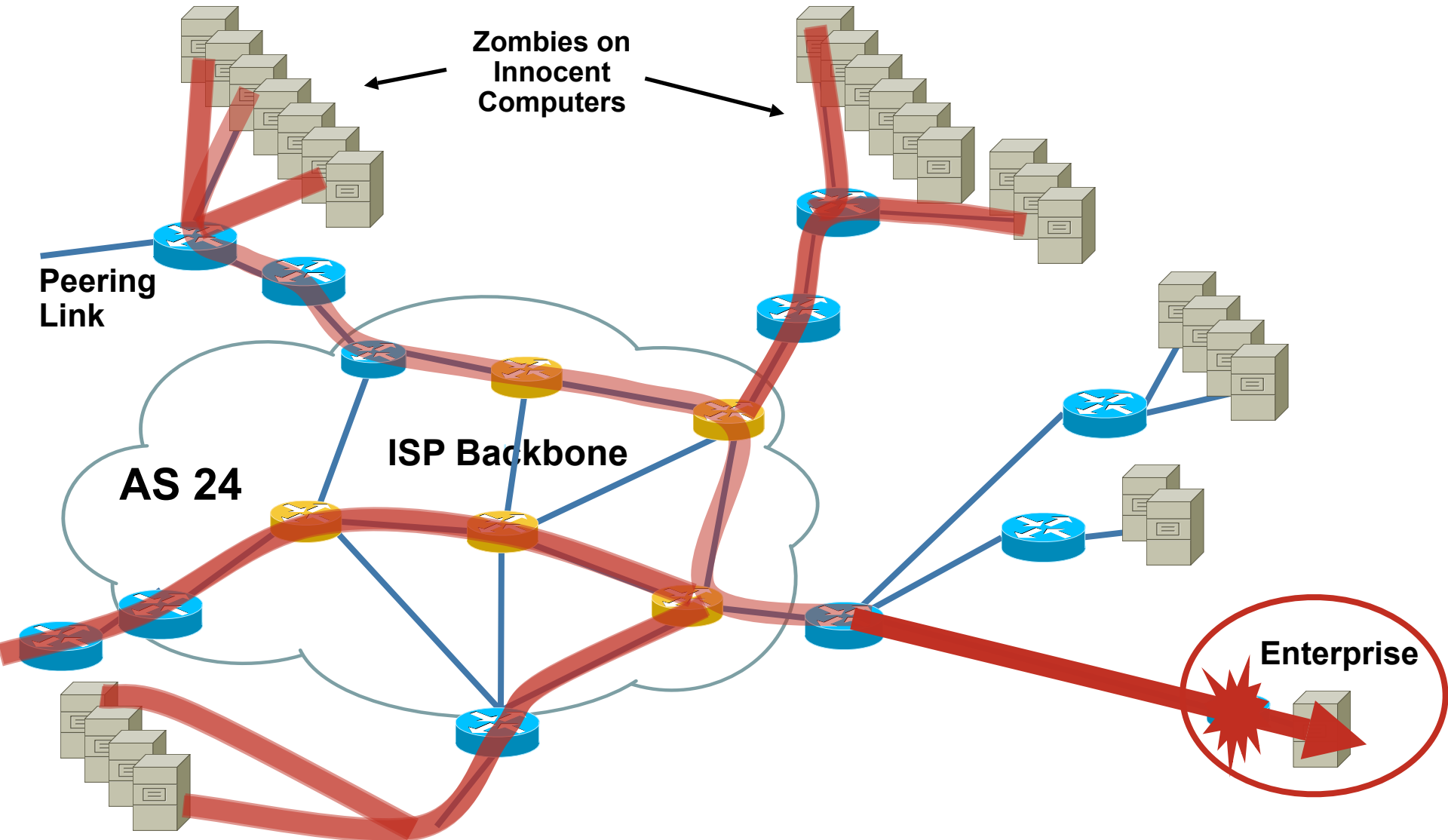
3. Flooding



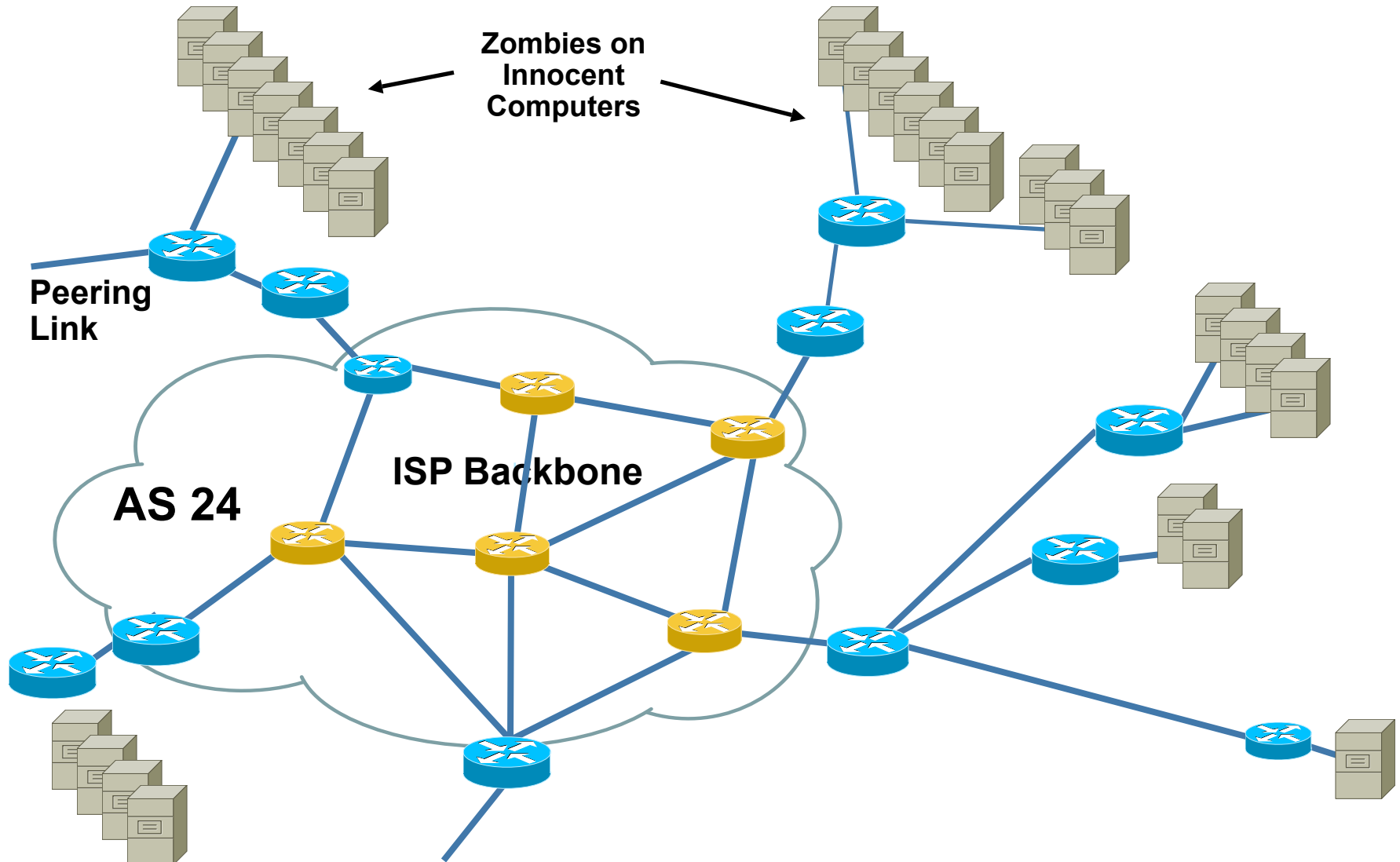
An SP View: Denial of Service



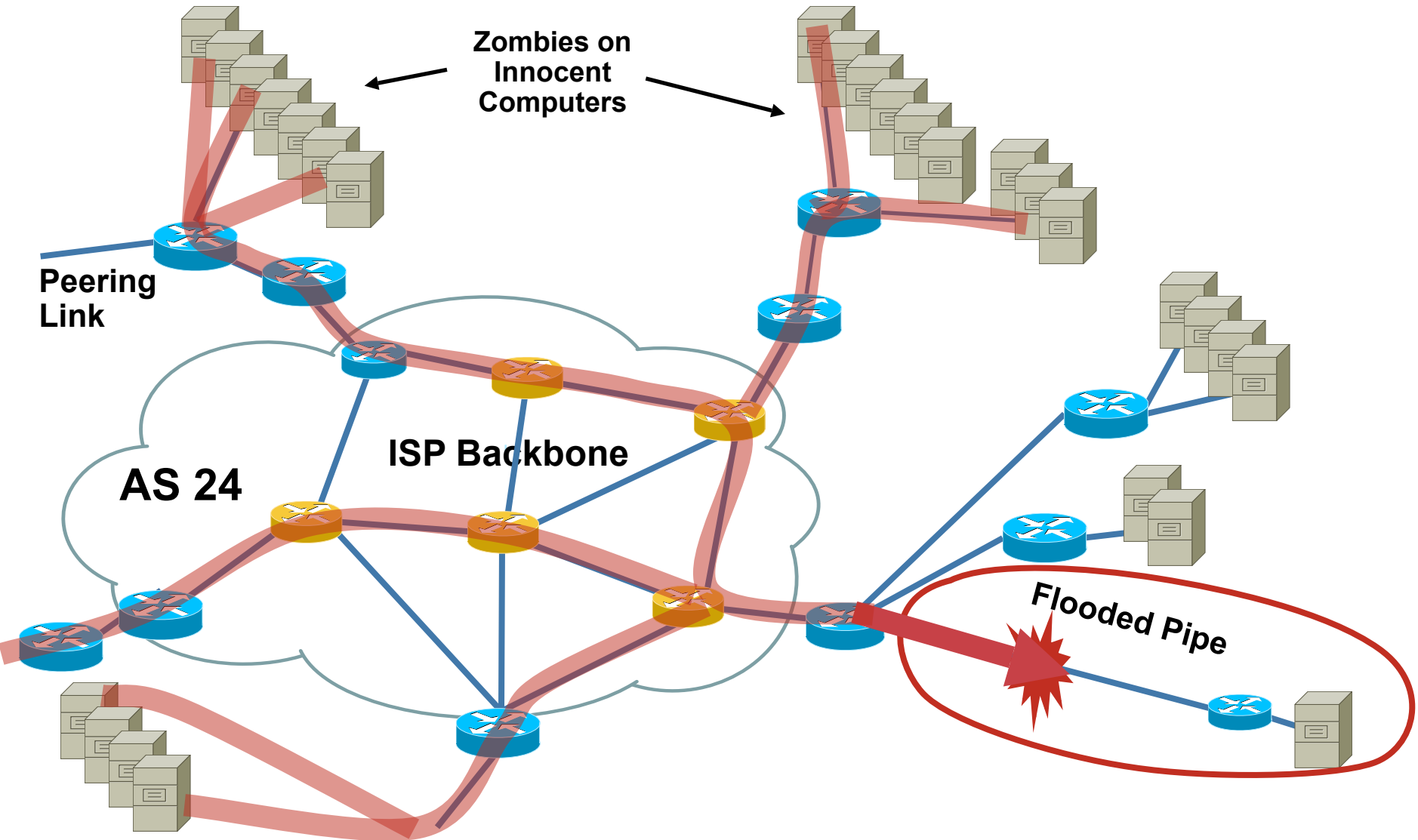
An SP View: Denial of Service



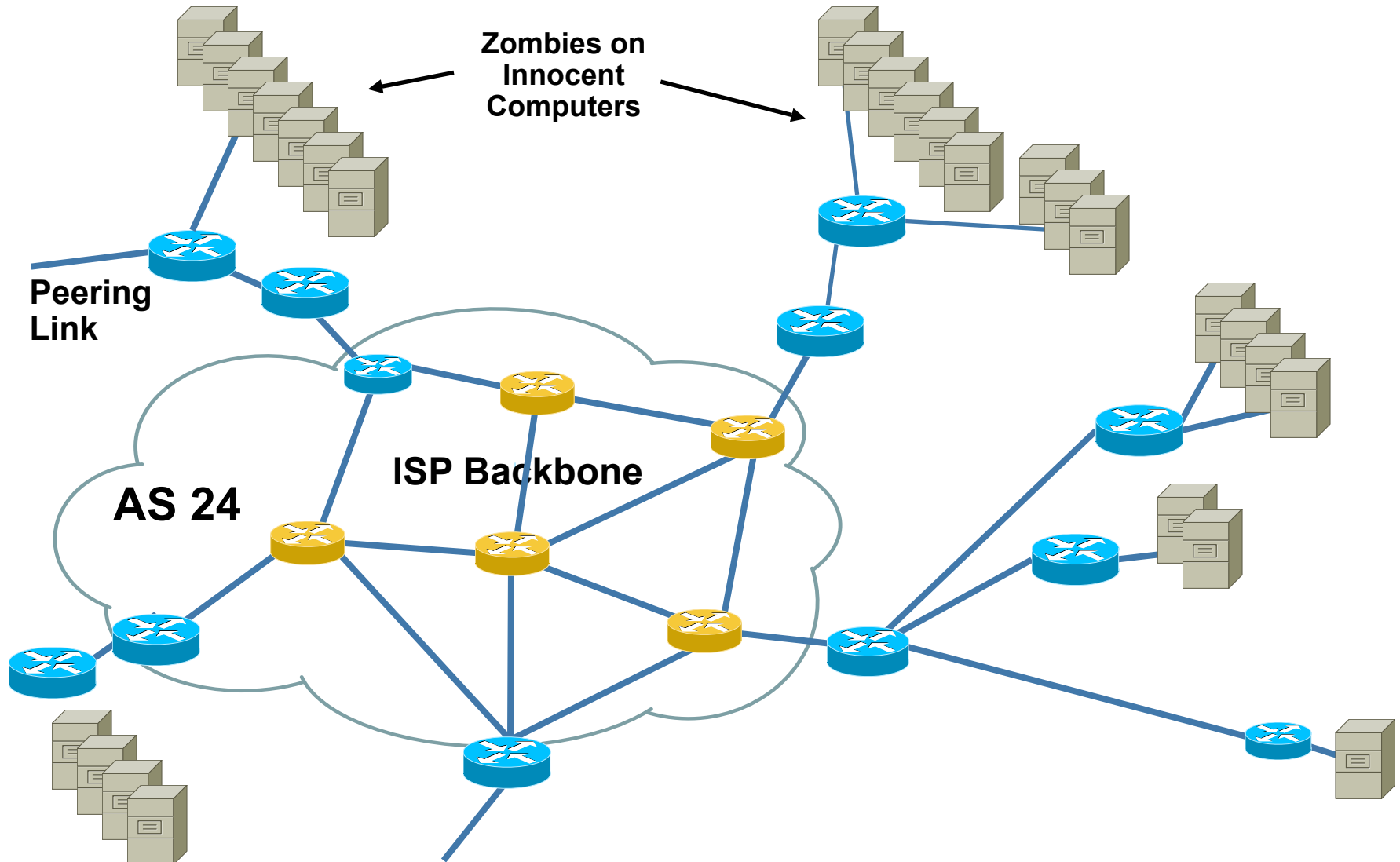
An SP View: Denial of Service



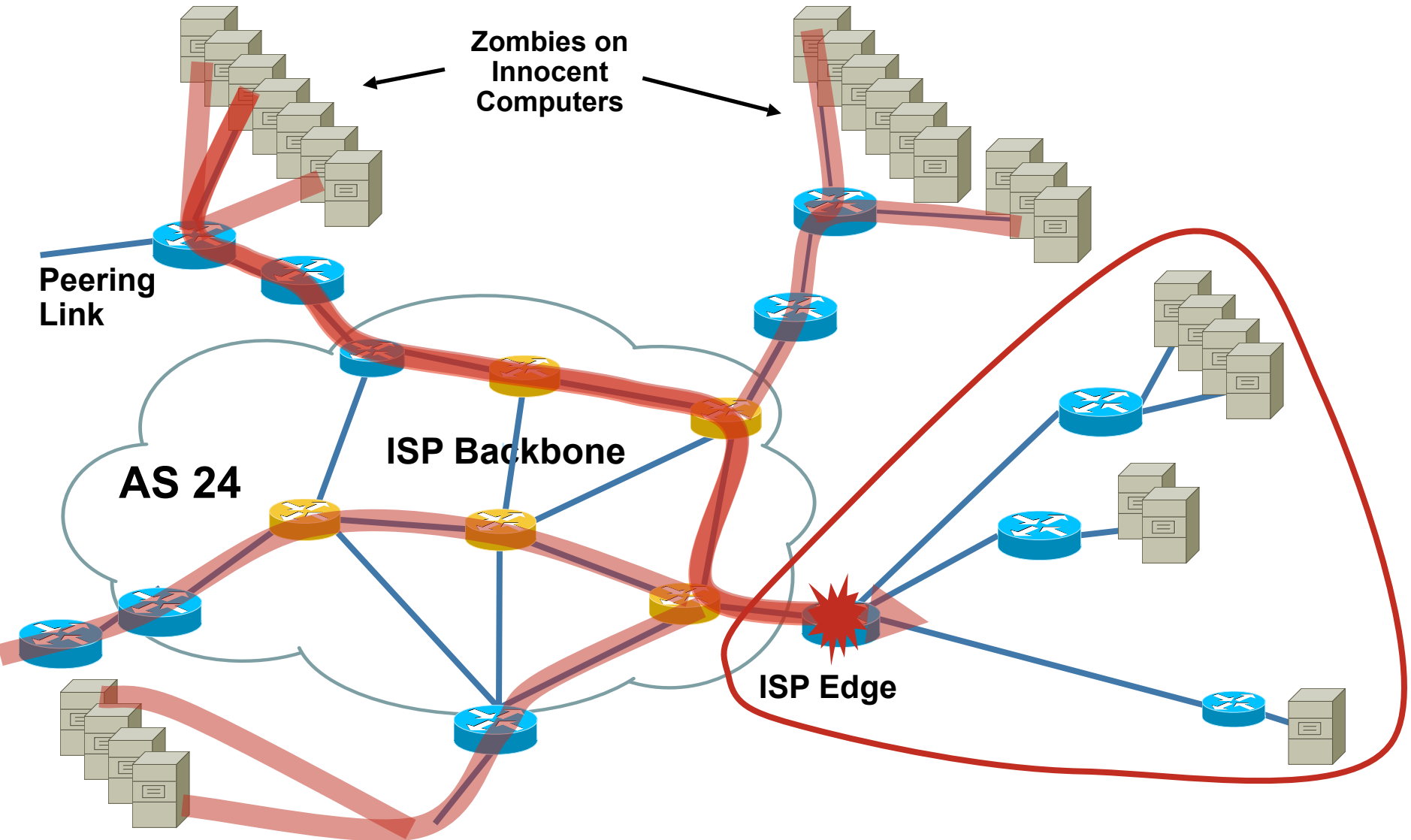
An SP View: Denial of Service



An SP View: Denial of Service



An SP View: Denial of Service



Denial of Service Trends

- Multipath
 - Truly distributed
 - DNS servers, large botnets
- Multivector
 - SYN AND UDP AND...
- Use of non-TCP/UDP/ICMP protocols
 - Get past ACLs
 - Increased awareness in community
- Financial incentive
 - SPAM, DoS-for-hire
 - Large, thriving business
 - Forces us to reassess the risk profile

Infrastructure Attacks

- Infrastructure attacks increasing in volume and sophistication

Sites with Cisco documents and presentations on routing protocols (and I don't mean Cisco.com)

Presentations about routers, routing and Cisco IOS® vulnerabilities at conferences like Blackhat, Defcon and Hivercon

Router attack tools and training are being published

- Why mount high-traffic DDoS attacks when you can take out your target's gateway routers?
- Hijacked routers valuable in spam world, which has a profit driver
- Router compromise (0wn3d) due to weak password

From Bad to Worms

- Worms have emerged as the new security reality
- Old worms never die
 - Millions of UPnP and Slammer packets still captured daily
- Most worms are intended to compromise hosts
- Worm propagation is dependant on network availability
- Worms and DoS are closely related
 - Secondary worm effects can lead to denial of service
 - Worms enable DoS by compromising hosts → BOTnets
- Perimeters are crumbling under the worm onslaught (VPN/mobile workers, partners, etc.)

Worms and the Infrastructure

- Worms typically infect end-stations
- To date, worms have not targeted infrastructure **but** secondary effects have wreaked havoc
 - Increased traffic
 - Random scanning for destination
 - Destination address is multicast
 - TTL and other header variances
- At the core SP level, the aggregate affects of a worm can be substantial
- Worm severity is escalating and evolving

Botnets

- **Botnet:** a collection of compromised machines running programs under a common command and control infrastructure

- Building the Botnet:

Viruses, worms; infected spam; drive-by downloads; etc.

- Controlling the Botnet:

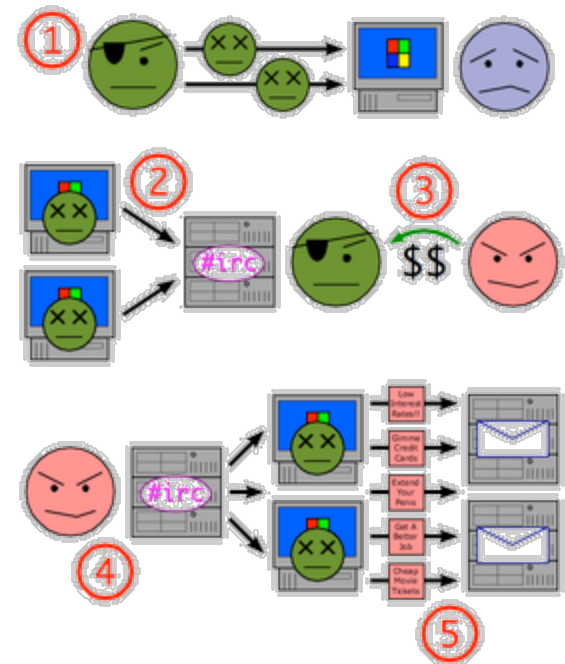
Covert-channel of some form; typically IRC

Historically have used free DNS hosting services to point bots to the IRC server

Recent attempts to sever the command infrastructure of botnets has resulted in more sophisticated control systems

Control services increasingly placed on compromised high-speed machines (e.g., in academic institutions)

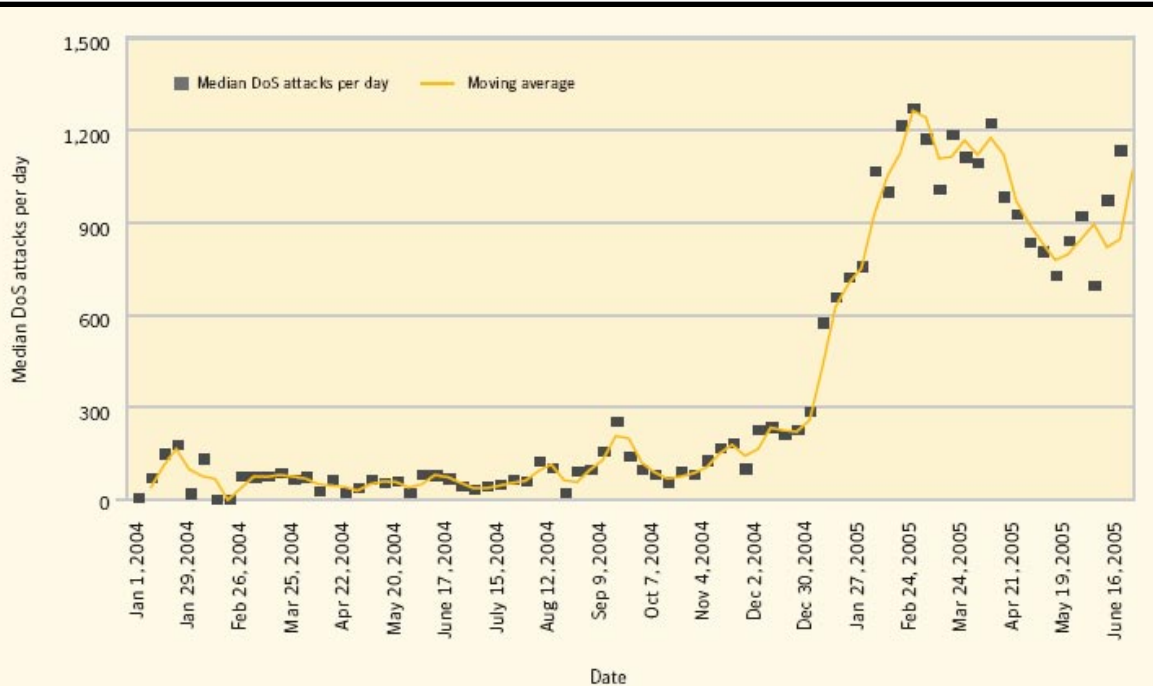
Redundant systems and blind connects are implemented for resiliency



Using a Botnet to Send Spam

1. A botnet operator sends out viruses or worms, infecting ordinary users' Windows PCs
2. The PCs log into an IRC server or other communications medium
3. A spammer purchases access to the botnet from the operator
4. The spammer sends instructions via the IRC server to the infected PCs
5. ...causing them to send out spam messages to mail servers

Botnets Growing Too Fast



Symantec Internet Security
Report – June '05

Figure 2. DoS attacks per day
Source: Symantec Corporation

- 10,000+ new bots added everyday
- DoS attacks grow from 119 to 927 per day—an increase of 679%
- Large % of DDoS attacks are motivated by extortion demands

CNN: Over 75 Million computers are infected with Botnet software

<http://www.cnn.com/2006/TECH/internet/01/31/furst/>

How Do You Respond?

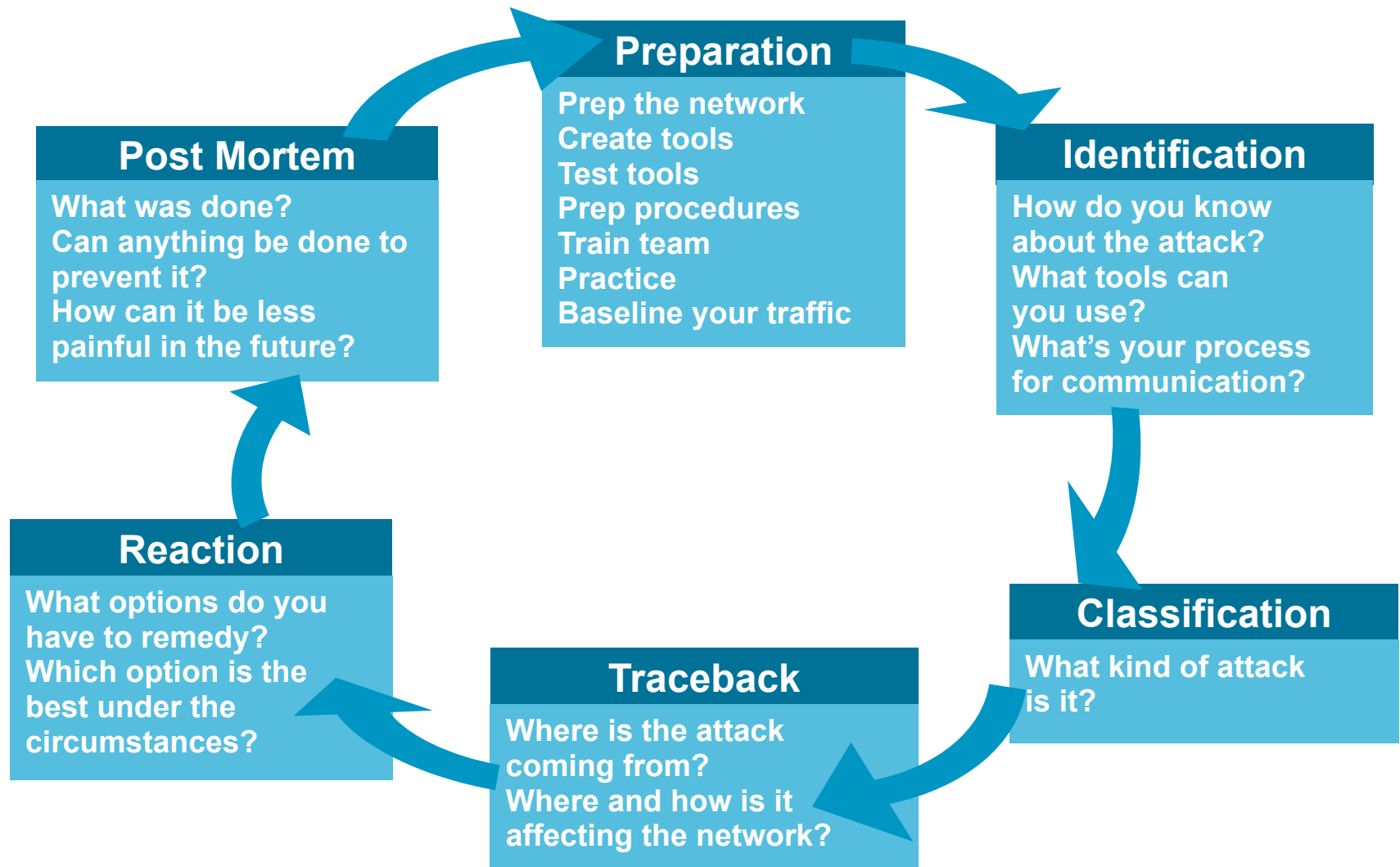
With Money Being the Key Driver of Miscreant Activity, Large Network Operators Need to Respond

- BCP deployment
- Execution of a broad and deep security toolkit
- Rethink some network/service architectures
- Create, staff, and train an operational security (OPSEC) team
- Practice, practice, practice

Six-Phase Methodology



Six Phases of Incident Response



Preparation

Preparation—Develop and Deploy a Solid Security Foundation

- Includes technical and non-technical components
- Encompasses best practices
- The hardest, yet most important phase
- Without adequate preparation, you are destined to fail
- The midst of a large attack is not the time to be implementing foundational best practices and processes

Preparation

- Know the enemy

 - Understand what drives the miscreants

 - Understand their techniques

- Create the security team and plan

 - Who handles security during an event?

 - Is it the security folks? The networking folks?

- Harden the devices

- Prepare the tools

 - Network telemetry

 - Reaction tools

 - Understand performance characteristics

Identification

Identification—How Do You Know You or Your Customer Is Under Attack?

- It is more than just waiting for your customers to scream or your network to crash
- What tools are available?
- What can you do today on a tight budget?

Identification—Ways to Detect

- Customer call
 - “The Internet is down”
- Unexplained changes in network baseline
 - SNMP: line/CPU overload, drops
 - Bandwidth
 - NetFlow
- ACLs with logging
- Backscatter
- Packet capture
- Network IPS
- Anomaly detection

Identification—Network Baselines

- NMS baselines
- Unexplained changes in link utilization
 - Worms can generate a lot of traffic, sudden changes in link utilization can indicate a worm
- Unexplained changes in CPU utilization
 - Worm scans can affect routers/switches resulting in increased CPU - process and interrupt switched traffic
- Unexplained syslog entries
- These are examples
 - Changes don't always indicate a security event
 - Must know what's normal in order to identify abnormal behavior

Classification

- Classification—understand the details and scope of the attack

Identification is not sufficient; once an attack is identified, details matter

Guides subsequent actions

- Identification and classification are often simultaneous

Classification

- Qualify and quantify the attack without jeopardizing services availability (e.g., crashing a router)

What type of attack has been identified?

What's the effect of the attack on the victim(s)?

What next steps are required (if any)?

- At the very least:

Source and destination address

Protocol information

Port information

Traceback

- Traceback—what are the sources of the attack?

 - How to trace to network ingress points

 - Your Internet connection is not the only vector

 - Understand your topology

- Traceback to network perimeter

 - NetFlow

 - Backscatter

 - Packet accounting

Traceback

- Retain attack data

- Use to correlate interdomain traceback

- Required for prosecution

- Deters future attacks

- Clarify billing and other disputes

- Post mortem analysis

Reaction

Reaction—Do Something to Counter the Attack

- Should you mitigate the attack?

Where? How?

- No reaction is a valid form of reaction in certain circumstances
- Reaction often entails more than just throwing an ACL onto a router

Post Mortem

Post Mortem—Analyze the Event

- The step everyone forgets
- What worked? What didn't? How can we improve?
- Protect against repeat occurrences?
- Was the DoS attack you handled the real threat?
Or was it a smoke screen for something else that just happened?
- What can you do to make it faster, easier, less painful in the future?
- Metrics are important
 - Resources, headcount, etc.

Device Best Practices



Router Security



Taking a Measured Approach

- The techniques discussing are extremely useful, but must be applied in an architecturally-sound, situationally-appropriate, and operationally-feasible manner
- Don't try to do this all at once—pick a technique with which you are comfortable and which you think will benefit you the most and start there
- Pilot chosen technique in a controlled manner and in a designated portion of your network
- Take lessons learned from pilot and work them into your general deployment plan and operational guidelines
- Rinse, repeat

Router Security

- Routers as shipped from the factory have:
 - Default configuration
 - Many services switched on to make getting started easier
- Once a router has an IP address, it is accessible to the outside world
 - Campus LAN
 - Company LAN/WAN
 - Internet
- Before you connect a new router to the network, you should “harden” the configuration

Disable Unneeded Services

Command	Impact	12.4M Default
No service config	Reconnaissance or Gain Access	Disabled
No boot network	Reconnaissance	Disabled
No cdp run	Reconnaissance	Enabled
No service pad	Gain Access	Enabled
No service tcp-small-servers	Reconnaissance or DoS	Disabled
No service udp-small-servers	Reconnaissance or DoS	Disabled

Disable Unneeded Services

Command	Impact	12.4M Default
No service finger	Reconnaissance	Disabled
No ip bootp server	Reconnaissance or DoS	Enabled
No service dhcp	Reconnaissance or DoS	Enabled
No ip http server	Reconnaissance or Gain Access	Disabled (Enabled on SDM Routers)
No ip domain lookup	Reconnaissance	Enabled

Disable Unneeded Services

Command	Impact	12.4M Default
No ip source-route	DoS	Enabled
No ip proxy-arp	DoS	Enabled
No ip directed-broadcast	DoS	Disabled
No ip unreachable	Reconnaissance	Enabled
No ip mask-reply	Reconnaissance	Disabled
No ip information-reply	Reconnaissance	Disabled
No ip redirects	Reconnaissance	Enabled
No ip identd	Reconnaissance	Disabled

No ip domain lookup

- Prevents router from performing DNS name resolution
- Disable service if name servers are not configured
- Without name servers, router will broadcast DNS queries
- If name servers are not reachable, router will not fallback to broadcast
- Access-lists and outbound connections can't use DNS names if disabled
- Locally configured host names can be used

No ip http server

- Applications that use the HTTP Server:

Cisco IOS Homepage Server, HTTP-based EXEC Server,
and HTTP Cisco IOS File System (IFS) Server

Security Device Manager (SDM)

VPN Device Manager (VDM)

QoS Device Manager (QDM)

IP Phone and Cisco IOS Telephony Service applications

Intrusion Prevention System (IPS)

- Selective Enabling of Applications Using an HTTP or HTTPS Server—available in 12.3(14)T

http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a008035b9b0.html

No ip http server

- Secure the HTTP Server:

 - ip http access-class

 - ip http authentication

 - ip http max-connections

 - ip http timeout-policy

- Use HTTPS—HTTP Server with SSL 3.0

 - <http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t15/ftsslsh.htm>

 - Always disable the standard HTTP server

 - Available in 12.2(15)T

Cisco Discovery Protocol

- CDP can be used to learn information about neighboring devices that are running CDP
IP address, software version, etc.
- CDP is configured per-interface basis
- Disable CDP when it is not needed
Public-facing interfaces, for example

ICMP Unreachable Overload

- Packets that cannot be delivered due to:
 - Null0 next-hops (in some cases)
 - No route in table
 - Administratively filtered packets
- Risk → high number of unreachables overloading CPU
 - no ip unreachables
- In certain situations we may want ICMP unreachables enabled, but need to limit generation to protect the router:

ICMP unreachable rate-limiting command:

```
ip icmp rate-limit unreachable [df] <1-4294967295 milliseconds>  
no ip icmp rate-limit unreachable [df]
```

Configuring Syslog on a Router

- Syslog data is invaluable

- Attack forensics

- Day-to-day events and debugging

- To log messages to a syslog server host, use the logging global configuration command

- `logging host`

- `logging trap <level>`

- To log to internal buffer use:

- `logging buffered size`

- Ensure timestamps

- `service timestamps log`

- Avoid Debug logging to the console

SNMP

- Version 1 sends cleartext community strings and has no policy reference
- Version 2 still uses cleartext community strings and adds bulk retrieval and detailed error reports
- Version 3 provides authentication and encryption

AES and 3-DES support

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124newft/124t/124t2/snmpv3ae.htm>

Confirm NMS application support

SNMP Authentication and Authorization

- Line ACL can filter SNMP access
- Don't use **public** and **private** community strings
- SNMP filtering
 - RO → read only
 - RW → read write
 - View → MIB restriction

```
access-list 4 permit 172.16.2.100
snmp-server community <string> RO 4
snmp-server community <string> view <MIB view>
```

Banners

- Login Banner

This is a legal requirement in some jurisdictions;
check with your legal group

- MOTD Banner

Displayed to all terminals

Appears before login prompt

- Exec Banner

Used to remind staff of specific conditions

Banners

banner login ^

Authorised access only

This system is the property of Galactic Internet

Disconnect IMMEDIATELY if you are not an authorised user!

Contact noc@isp.net 555-1212 for help.

^

banner motd ^

Notice: all routers in \$(domain) will be upgraded beginning July 1

^

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 config for this router is NON-STANDARD

Contact Network Engineering 555-1212 for more info.

^

Network Time Protocol

- Synchronize time across all devices
- When security event occurs, data must have consistent timestamps

From external time source (Upstream ISP, Internet, GPS, atomic clock)

From internal time source

Router can act as **stratum 1** time source

```
ntp source loopback0  
ntp server 10.1.1.1 source loopback0
```

Network Time Protocol Security

- Authenticate NTP messages
- NTP access controls

<http://www.cisco.com/warp/public/707/NTP-pub.shtml>

- Disable NTP on interfaces that don't need it

```
ntp authenticate
ntp authentication-key 1 md5 <value>
ntp trusted-key 1
ntp access-group {query-only | serve-only | serve | peer}
<ACL number>

Interface fa0/0
    ntp disable
```

Access to the Router

- Console, VTY
- SSH—encrypted access
- Telnet (prefer SSH)
- Local passwords
 - Usernames configured on the router with MD5 passwords
- External AAA
 - TACACS+, RADIUS, Kerberos
- One-time passwords (OTP)



Use Enable Secret

- Service password-encryption is reversible

```
service password-encryption
!  
hostname Router  
!  
enable password 7 14181C0E2A2B182A2824
```

- The “enable secret” password hashed via MD5

```
!  
hostname Router  
!  
enable secret 5 $1$hM3l$.s/DgJ4TeKdDkTVCJpIBw1
```

VTY Security

- Access to VTYs should be controlled
- ACL used to filter incoming data
- Logging can be used to provide more information
 - Adds little overhead: 90% vs 88% CPU @ 10Kpps
- Consider service tcp-keepalives-in

```
service tcp-keepalives-in
access-list 3 permit 215.17.1.0 0.0.0.255
access-list 3 deny any log
line vty 0 4
    access-class 3 in
    transport preferred none
    transport input ssh
    transport output ssh
```

SSH

- Replaces telnet for a protected command and control communication channel
- Privacy and integrity provided through the use of strong cryptographic algorithms
- Supports TACACS+, RADIUS and local authentication
- Secure Copy (SCP) available in new SSH enabled code
- Restrict access to ssh via “transport input ssh” command
- Newer Cisco IOS versions support SSHv2

Cisco IOS TACACS+ Login Authentication

Encrypts Passwords with Encryption (7) →

Define List “tech” to Use TACACS+ Then the Local User and Password →

Enable Secret Overrides the (7) Encryption →

Define Local Users →

```
!  
service password-encryption  
!  
hostname Router  
!  
aaa new-model  
aaa authentication login neteng group tacacs+ enable  
aaa authentication login tech group tacacs+ local  
aaa authentication enable default group tacacs+ enable  
enable secret 5 $1$hM3I$.s/DgJ4TeKdDk...  
!  
username cisco password c!sc0
```


Cisco IOS TACACS+ Login Authentication

Encrypts Passwords with Encryption (7) →

Define List “neteng” to Use TACACS+ →

Define List “tech” to Use TACACS+ Then the Local User and Password →

Enable Secret Overrides the (7) Encryption →

Define Local Users →

```
!  
service password-encryption  
!  
hostname Router  
!  
aaa new-model  
aaa authentication login neteng group tacacs+ enable  
aaa authentication login tech group tacacs+ local  
aaa authentication enable default group tacacs+ enable  
enable secret 5 $1$hM3I$.s/DgJ4TeKdDk...  
!  
username cisco password c!sc0
```

Cisco IOS TACACS+ Login Authentication

```
tacacs-server host 172.16.1.4
tacacs-server key CKr3t#
!
line con 0
 login authentication neteng
line aux 0
 login authentication neteng
line vty 0 4
 login authentication tech
!
end
```

Cisco IOS TACACS+ Login Authentication

```
tacacs-server host 172.16.1.4  
tacacs-server key CKr3t#  
!  
line con 0  
  login authentication neteng  
line aux 0  
  login authentication neteng  
line vty 0 4  
  login authentication tech  
!  
end
```

**Defines the IP Address
of the TACACS+ Server**



Cisco IOS TACACS+ Login Authentication

```
tacacs-server host 172.16.1.4  
tacacs-server key CKr3t#  
!  
line con 0  
  login authentication neteng  
line aux 0  
  login authentication neteng  
line vty 0 4  
  login authentication tech  
!  
end
```

Defines the IP Address
of the TACACS+ Server

Defines the Shared
Key for Communicating
with the TACACS+ Server

Cisco IOS TACACS+ Login Authentication

```
tacacs-server host 172.16.1.4  
tacacs-server key CKr3t#  
!  
line con 0  
  login authentication neteng  
line aux 0  
  login authentication neteng  
line vty 0 4  
  login authentication tech  
!  
end
```

**Defines the IP Address
of the TACACS+ Server**

**Defines the Shared
Key for Communicating
with the TACACS+ Server**

**Uses the Authentication
Mechanisms Listed in
“neteng”—TACACS+
Then Enable Password**

Cisco IOS TACACS+ Login Authentication

```
tacacs-server host 172.16.1.4  
tacacs-server key CKr3t#  
!  
line con 0  
  login authentication neteng  
line aux 0  
  login authentication neteng  
line vty 0 4  
  login authentication tech  
!  
end
```

**Defines the IP Address
of the TACACS+ Server**

**Defines the Shared
Key for Communicating
with the TACACS+ Server**

**Uses the Authentication
Mechanisms Listed in
“neteng”—TACACS+
Then Enable Password**

**Uses the Authentication
Mechanisms Listed in
“tech”—TACACS+ Then
a Local User/Password**


AAA Configuration with Default Method

```
aaa new-model
aaa authentication login default tacacs+ local enable
aaa authentication enable default tacacs+ local enable
aaa authorization exec default tacacs+ local
aaa authorization commands 1 default tacacs+ local
aaa authorization commands 15 default tacacs+ local
aaa accounting exec start-stop tacacs+
ip tacacs source-interface Loopback0
tacacs-server host 10.1.1.1
tacacs-server host 10.2.1.1
tacacs-server key CKr3t#
line vty 0 4
  access-class 3 in
!
username cisco password cisco
```

**Try 10.1.1.1 First; If No
Reply, Use 10.2.1.1**

AAA Configuration with Default Method


```
aaa new-model
aaa authentication login default tacacs+ local enable
aaa authentication enable default tacacs+ local enable
aaa authorization exec default tacacs+ local
aaa authorization commands 1 default tacacs+ local
aaa authorization commands 15 default tacacs+ local
aaa accounting exec start-stop tacacs+
ip tacacs source-interface Loopback0
tacacs-server host 10.1.1.1
tacacs-server host 10.2.1.1
tacacs-server key CKr3t#
line vty 0 4
  access-class 3 in
!
username cisco password cisco
```



Try 10.1.1.1 First; If No Reply, Use 10.2.1.1

AAA Configuration with Default Method

```
aaa new-model
aaa authentication login default tacacs+ local enable
aaa authentication enable default tacacs+ local enable
aaa authorization exec default tacacs+ local
aaa authorization commands 1 default tacacs+ local
aaa authorization commands 15 default tacacs+ local
aaa accounting exec start-stop tacacs+
ip tacacs source-interface Loopback0
tacacs-server host 10.1.1.1
tacacs-server host 10.2.1.1
tacacs-server key CKr3t#
line vty 0 4
  access-class 3 in
!
username cisco password cisco
```



Try 10.1.1.1 First; If No Reply, Use 10.2.1.1

Set Privileges

- Set level of privilege for each user class

```
privilege configure level 5 interface  
privilege interface level 5 shutdown  
privilege exec level 5 show ip route  
privilege exec level 5 configure terminal  
privilege exec level 5 show version
```

- Initially difficult to deploy
- Long-term benefit outweighs short-term pain

Role-Based CLI Access

- New Feature: Role-based CLI, also known as CLI views
- Defines CLI access based on administrative roles
- Security

Enhances the security of the device by defining the set of CLI commands that are accessible to a particular user

- Availability

Avoids unintentional execution of CLI commands by unauthorized personnel

- Operational efficiency

Prohibits users from viewing CLI commands that are inaccessible to them, greatly improving usability

- Available in 12.3(7)T

http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_white_paper09186a00801ee18d.shtml

Role-Based CLI Access

- “Views” restrict user access to Cisco IOS CLI and configuration information

```
Router(config)# parser view first
Router(config-view)# secret firstpass
Router(config-view)# commands exec include show version
Router(config-view)# commands exec include configure terminal
Router(config-view)# commands interface include shutdown
Router(config-view)# commands interface include no shutdown
Router(config-view)# commands configure include interface
Router(config-view)# commands exec include show ip route
Router(config-view)# commands configure include interface GigabitEthernet0/0
Router(config-view)# commands configure include interface GigabitEthernet0/1
```

Device Security Best Practice Checklist

	Device Security Best Practices
	No ip domain lookup
	No ip http server
	Cisco Discovery Protocol
	ICMP Unreachable Overload
	Configuring Syslog on a Router
	SNMP
	Cisco IOS Warm Upgrade
	Banners
	Auto Secure
	Cisco IOS Login Enhancements
	Login Password Retry Lockout
	AAA Server

Routing Protocol Security



Routing Protocol Security

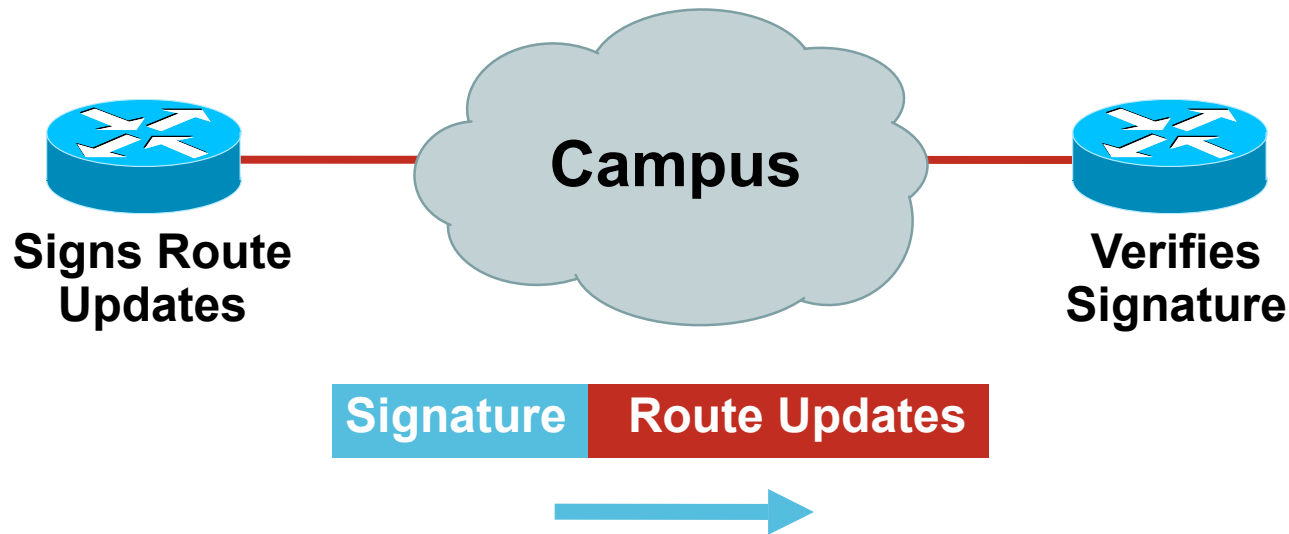
Routing Protocols Can Be Attacked

- Denial of service
- Smokescreens
- False information
- Reroute packets

May Be Accidental or Intentional

Secure Routing—Route Authentication

Configure Routing Authentication



Certifies **Authenticity** of Neighbor and **Integrity** of Route Updates

Route Authentication

- Shared key included in routing updates
 - Plain text—protects against accidental problems only
 - Message Digest 5 (MD5)—protects against accidental and intentional problems
- Multiple keys supported
- Supported for BGP, IS-IS, OSPF, RIPv2, and EIGRP
- Update keys before protocol timeout to avoid session bounce
- Often non-implemented
 - “Never seen an attack”
 - “My peer doesn’t use it”

OSPF and ISIS Authentication Example

OSPF

```
interface ethernet1
  ip address 10.1.1.1
  255.255.255.0

  ip ospf message-
  digest-key 100 md5
  qa*&gt;HH3
  !

  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 pe#$rt@s
  level-2
```

BGP Route Authentication

```
router bgp 200
  no synchronization
  neighbor 4.1.2.1 remote-as 300
  neighbor 4.1.2.1 description Link to Excalibur
  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 7 q23dc%$#ert
```

BGP Route Authentication

- Works per neighbor or for an entire peer-group
- Two routers with password mismatch:
`%TCP-6-BADAUTH: Invalid MD5 digest from [peer's IP address]:11004 to [local router's IP address]:179`
- One router has a password and the other does not:
`%TCP-6-BADAUTH: No MD5 digest from [peer's IP address]:11003 to [local router's IP address]:179`

BGP Support for TTL Security Check

- AKA BGP TTL Security Hack (BTSH)
- Protects eBGP sessions from CPU attacks using forged IP packets
- Prevents attempts to hijack eBGP session by attacker not part of either BGP network or that is not between the eBGP peers
- Configure minimum Time To Live (TTL) for incoming IP packets from a specific eBGP peer

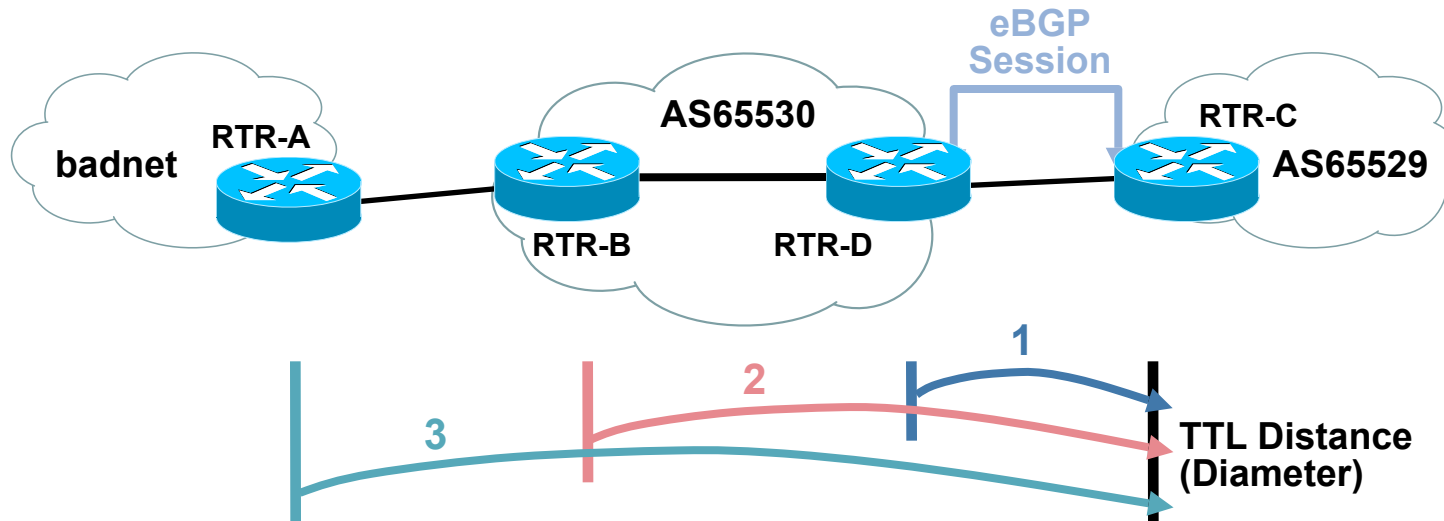
BGP session established and maintained only if TTL in IP packet header is equal to or greater than configured TTL value. Initial TTL set to 255

If value is less than configured value packet is silently discarded and no ICMP message generated

- Not supported for iBGP and occurs after MD5 check if enabled
- Available in 12.0(27)S, 12.3(7)T, 12.2(25)S, 12.2(18)SXE

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123newft/123t/123t_7/gt_btsh.htm

BGP TTL Security Check: How Does It Work?



Example on RTR-C:

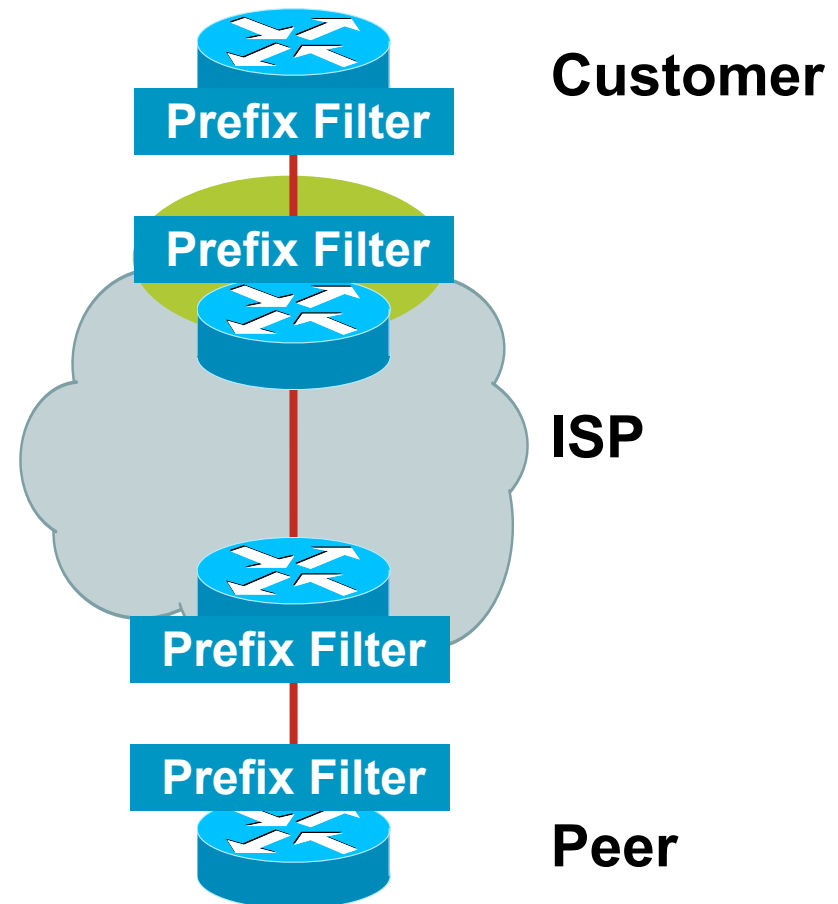
```
router bgp 65529
  neighbor 10.1.1.1 ttl-security hops 1
  ! expected TTL value in IP header is now 254 not 0
```

- Spoofed IP packets may have correct IP source and destination addresses (and TCP source and destination ports); however, unless these packets originate on a network segment that is between the eBGP peers, the TTL values will be less than the “minimum” configured in the BGP TTL security check

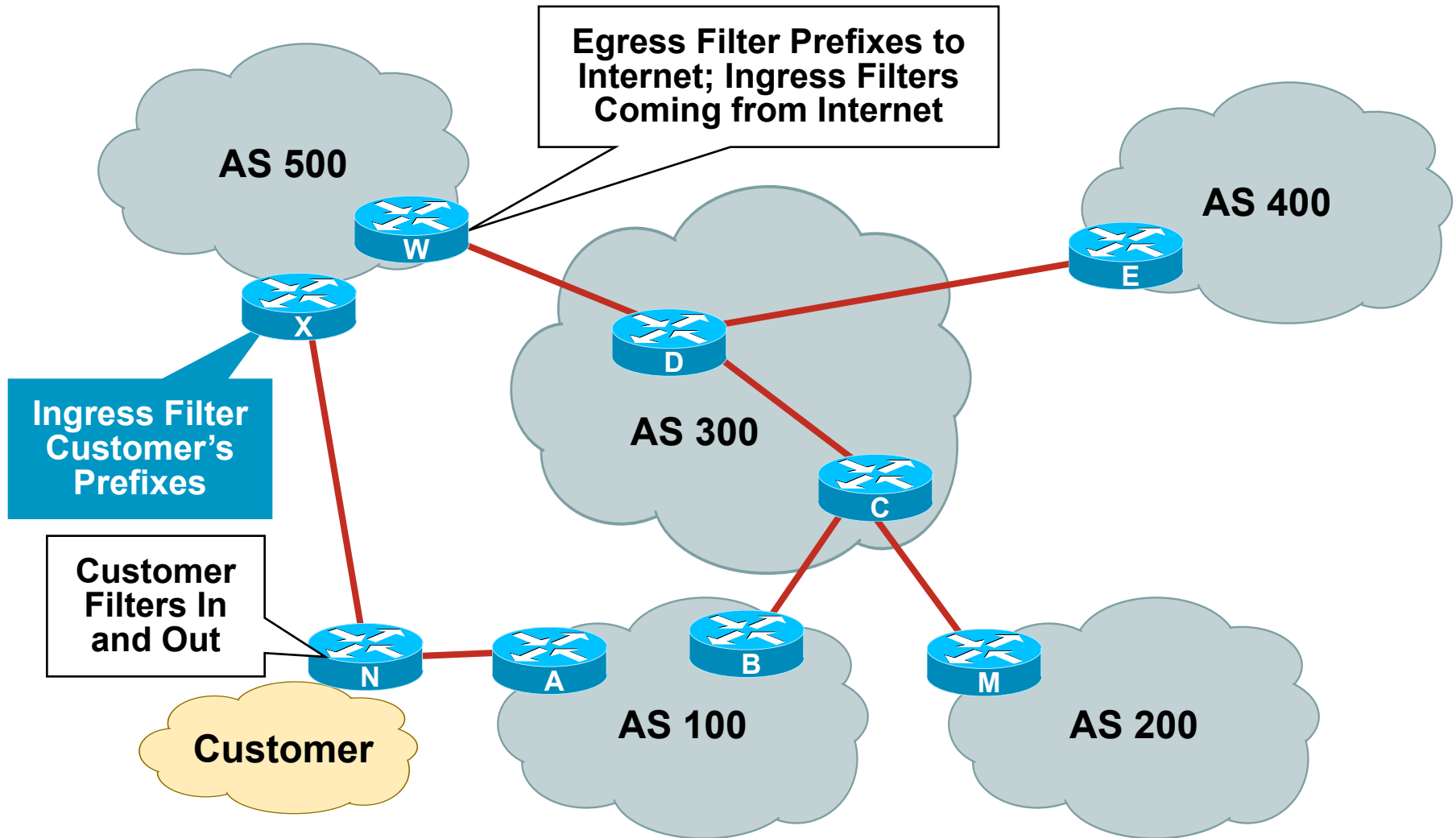
Prefix Filters

Apply Prefix Filters to All eBGP Neighbors to Prevent Injection of False Routing Information

- To/from customers
- To/from peers
- To/from upstreams



Where to Prefix Filter?



Bogons and Special Use Addresses

- IANA has reserved several blocks of IPv4 that have yet to be allocated to a RIR:

<http://www.iana.org/assignments/ipv4-address-space>

- These blocks of IPv4 addresses should never be advertised into the global internet route table
- Filters should be applied on the AS border for all inbound and outbound advertisements
- Special Use Addresses (SUA) are reserved for special use :-)

Defined in RFC3330

Examples: 127.0.0.0/8, 192.0.2.0/24

Ingress and Egress Route Filtering

- Two flavors of route filtering:
 - Distribute list—widely used
 - ACL entries generate hit count
 - Prefix list—increasingly used
- Both work—engineering preference
- Two filtering techniques:
 - Explicit permit (permit then deny any)
 - Explicit deny (deny then permit any)

Extended ACL for a BGP Distribute List

```
access-list 150 deny ip host 0.0.0.0 any
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 permit ip any any
```

BGP with Distribute List Route Filtering

```
router bgp 65535
  no synchronization
  bgp dampening
  neighbor 220.220.4.1 remote-as 210
  neighbor 220.220.4.1 version 4
  neighbor 220.220.4.1 distribute-list 150 in
  neighbor 220.220.4.1 distribute-list 150 out
  neighbor 222.222.8.1 remote-as 220
  neighbor 222.222.8.1 version 4
  neighbor 222.222.8.1 distribute-list 150 in
  neighbor 222.222.8.1 distribute-list 150 out
  no auto-summary
```

!

Prefix-List for a BGP Prefix List

```
ip prefix-list rfc1918-dsua seq 5 deny    0.0.0.0/8 le 32
ip prefix-list rfc1918-dsua seq 10 deny   10.0.0.0/8 le 32
ip prefix-list rfc1918-dsua seq 15 deny   127.0.0.0/8 le 32
ip prefix-list rfc1918-dsua seq 20 deny   169.254.0.0/16 le 32
ip prefix-list rfc1918-dsua seq 25 deny   172.16.0.0/12 le 32
ip prefix-list rfc1918-dsua seq 30 deny   192.0.2.0.0/24 le 32
ip prefix-list rfc1918-dsua seq 35 deny   192.168.0.0/16 le 32
ip prefix-list rfc1918-dsua seq 40 deny   224.0.0.0/3 le 32
ip prefix-list rfc1918-dsua seq 45 permit 0.0.0.0/0 le 32
```

BGP with Prefix-List Route Filtering

```
router bgp 65535
  no synchronization
  bgp dampening
  neighbor 220.220.4.1 remote-as 210
  neighbor 220.220.4.1 version 4
  neighbor 220.220.4.1 prefix-list rfc1918-dsua in
  neighbor 220.220.4.1 prefix-list rfc1918-dsua out
  neighbor 222.222.8.1 remote-as 220
  neighbor 222.222.8.1 version 4
  neighbor 222.222.8.1 prefix-list rfc1918-dsua in
  neighbor 222.222.8.1 prefix-list rfc1918-dsua out
  no auto-summary
```

!

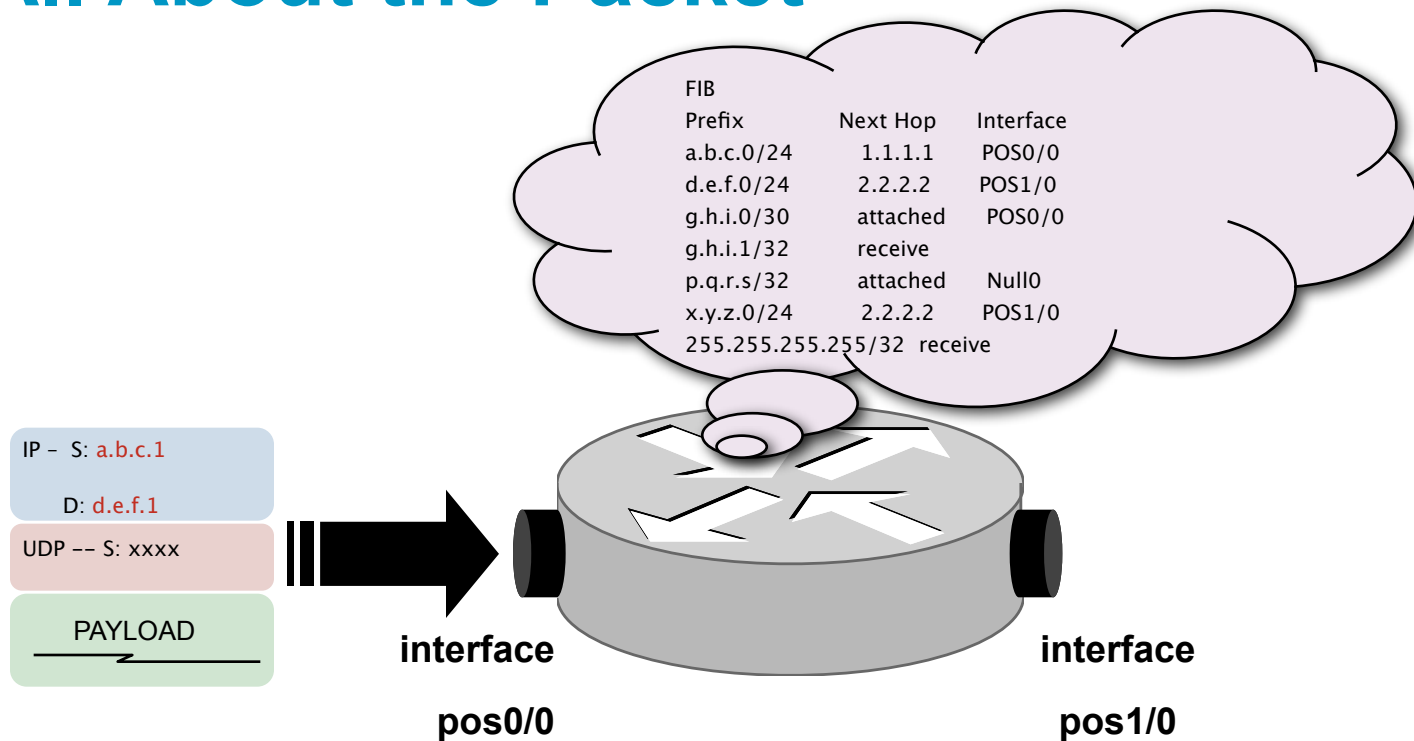
Planes, Paths and Punts



The Four Planes

- Data plane—packets going through the router
- Control plane—routing protocols gluing the network together
- Management plane—tools and protocols used to manage the device
- Services plane—customer traffic (similar to the data plane), traffic requiring specialized forwarding functions applied it

It's All About the Packet



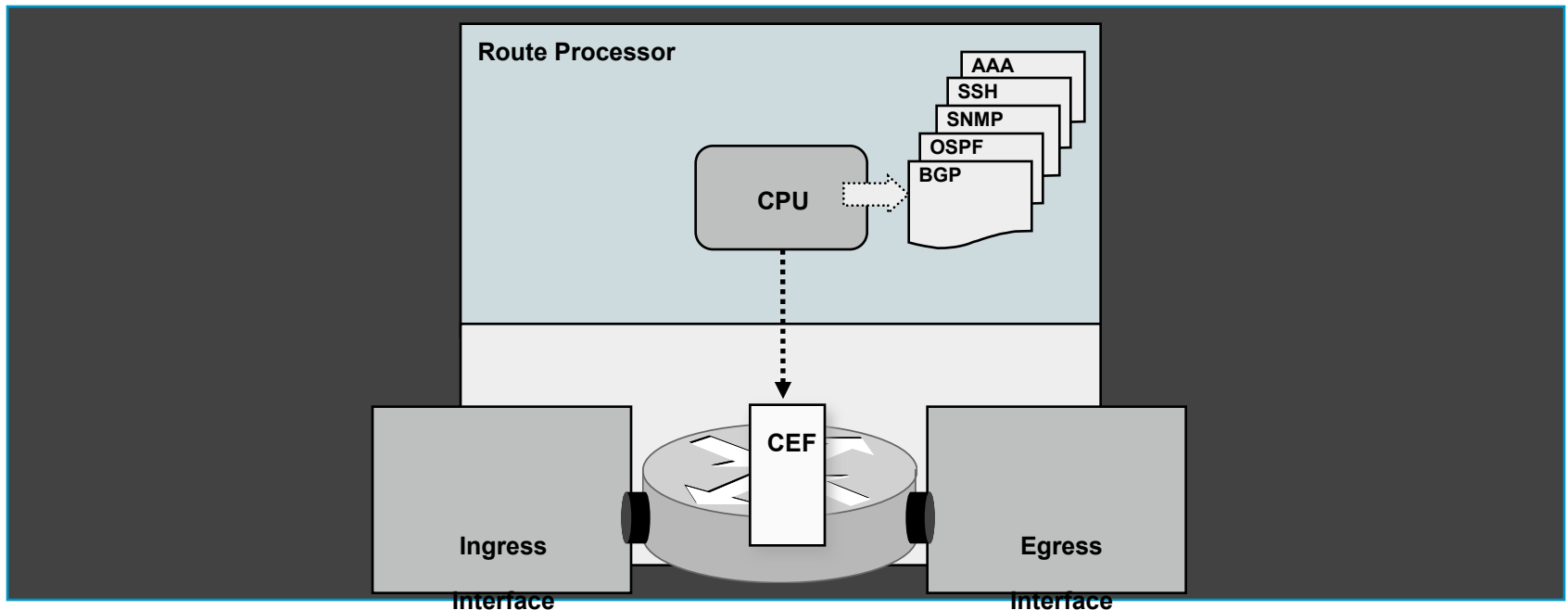
- Once a packet gets into the Internet, **some device, somewhere** has to do one of two things: [1] **Forward the Packet*** or [2] **Drop the Packet**
- In the context of security, the questions are more granular:
 - Who** forwarded the packet, and **what** resources were required to do so...
 - Who** dropped the packet, and **why** was it dropped...

* Forwarding Could Entail Adding a **Service** to the Packet as well...

Transit Packets

Transit Packets

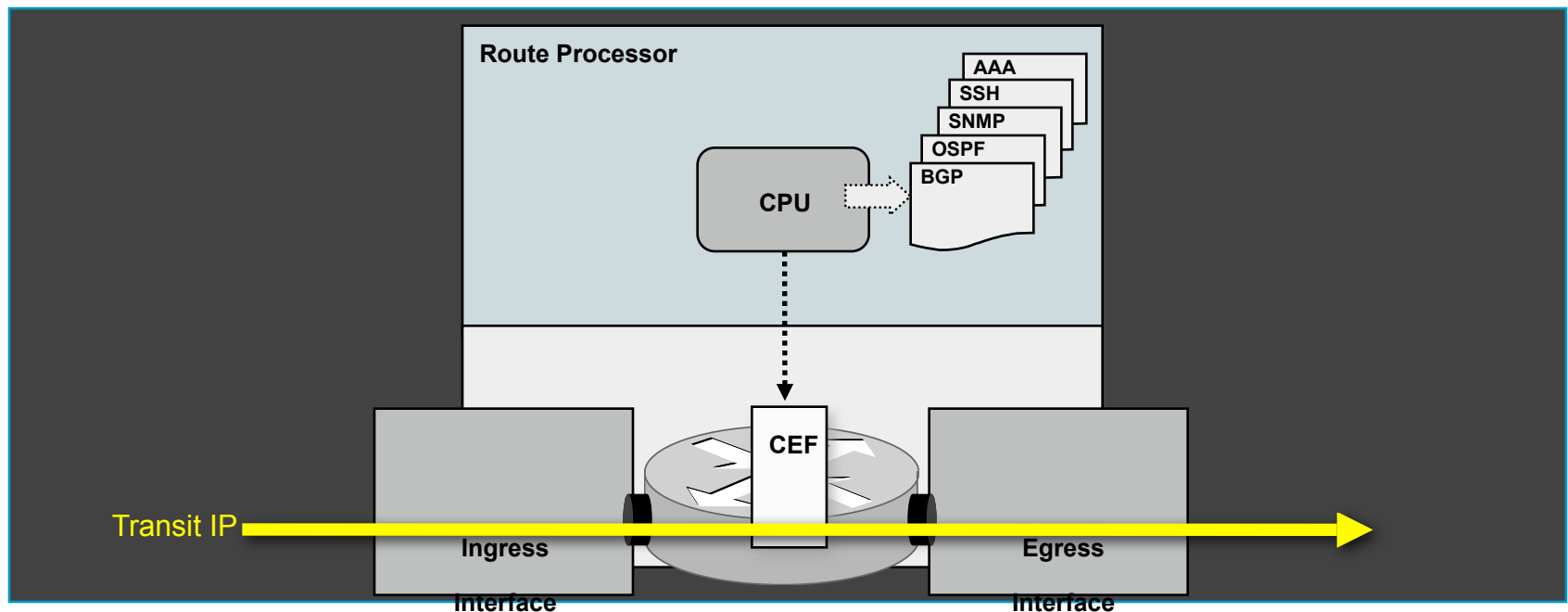
- Well-formed IP packets that follow standard, destination IP address-based forwarding processes. No extra processing by the route processor is required to forward these packets.
- The destination IP address of these packets is located downstream from the network device and thus, the packet is forwarded out an egress interface



Transit Packets

Transit Packets

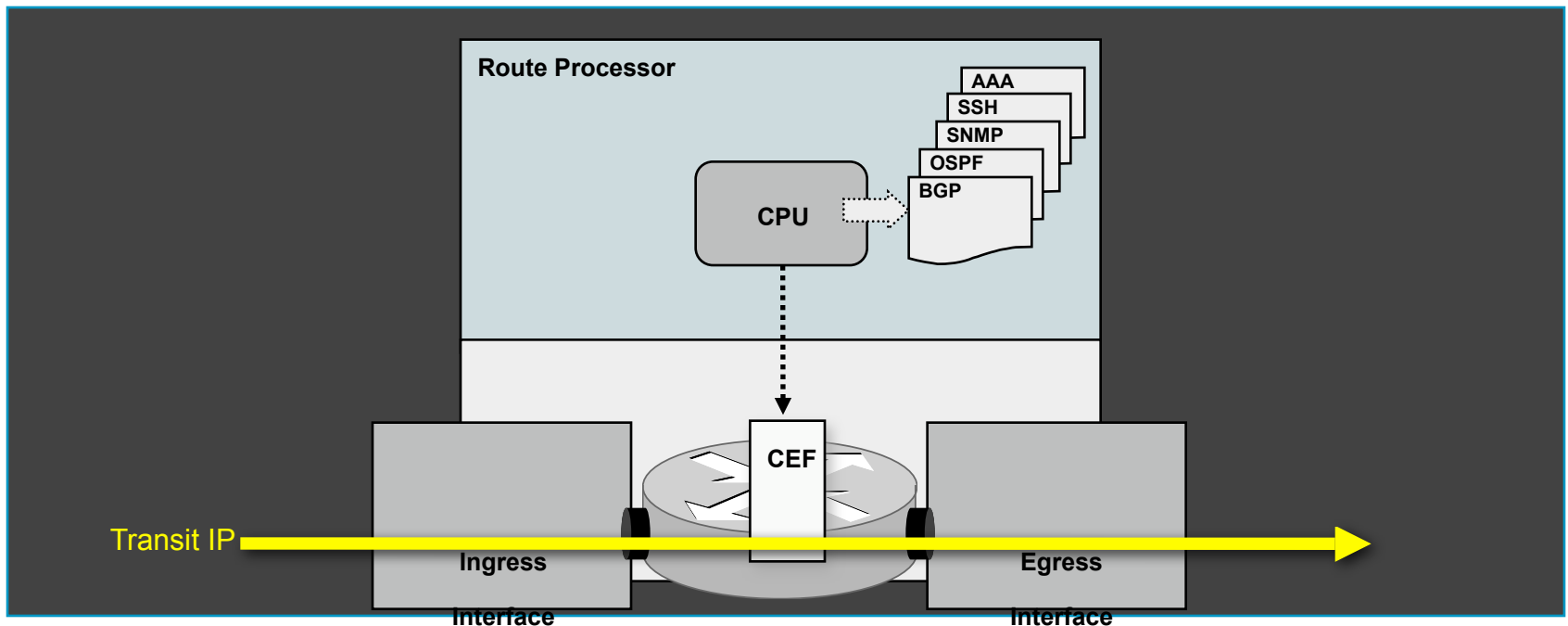
- Well-formed IP packets that follow standard, destination IP address-based forwarding processes. No extra processing by the route processor is required to forward these packets.
- The destination IP address of these packets is located downstream from the network device and thus, the packet is forwarded out an egress interface



Receive Packets

Receive Packets

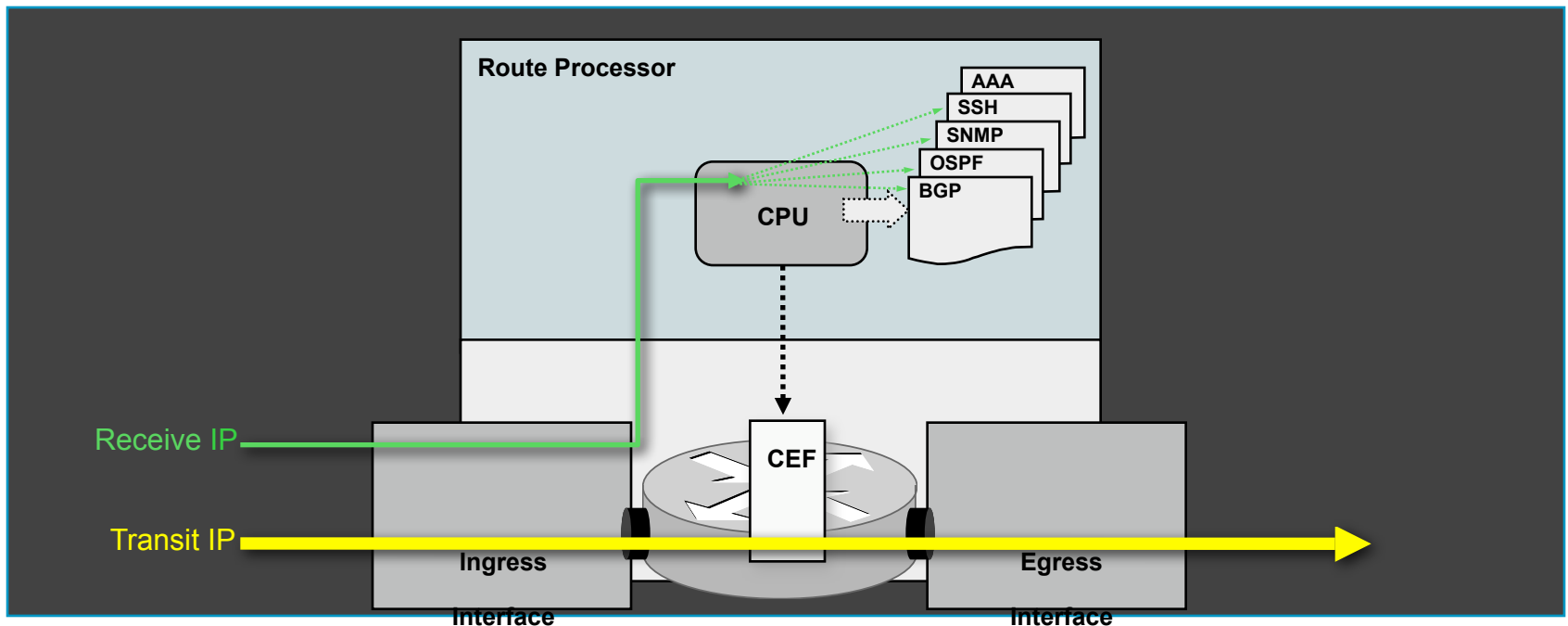
- Packets that are destined to the network device itself (e.g. control and management packets) must be handled by the route processor CPU since they ultimately are destined for and handled by applications running at the process level.
- All of the packets in this set must be handled by the route processor



Receive Packets

Receive Packets

- Packets that are destined to the network device itself (e.g. control and management packets) must be handled by the route processor CPU since they ultimately are destined for and handled by applications running at the process level.
- All of the packets in this set must be handled by the route processor



Receive Adjacencies

- CEF entries for traffic destined to router

Real interfaces and Loopbacks

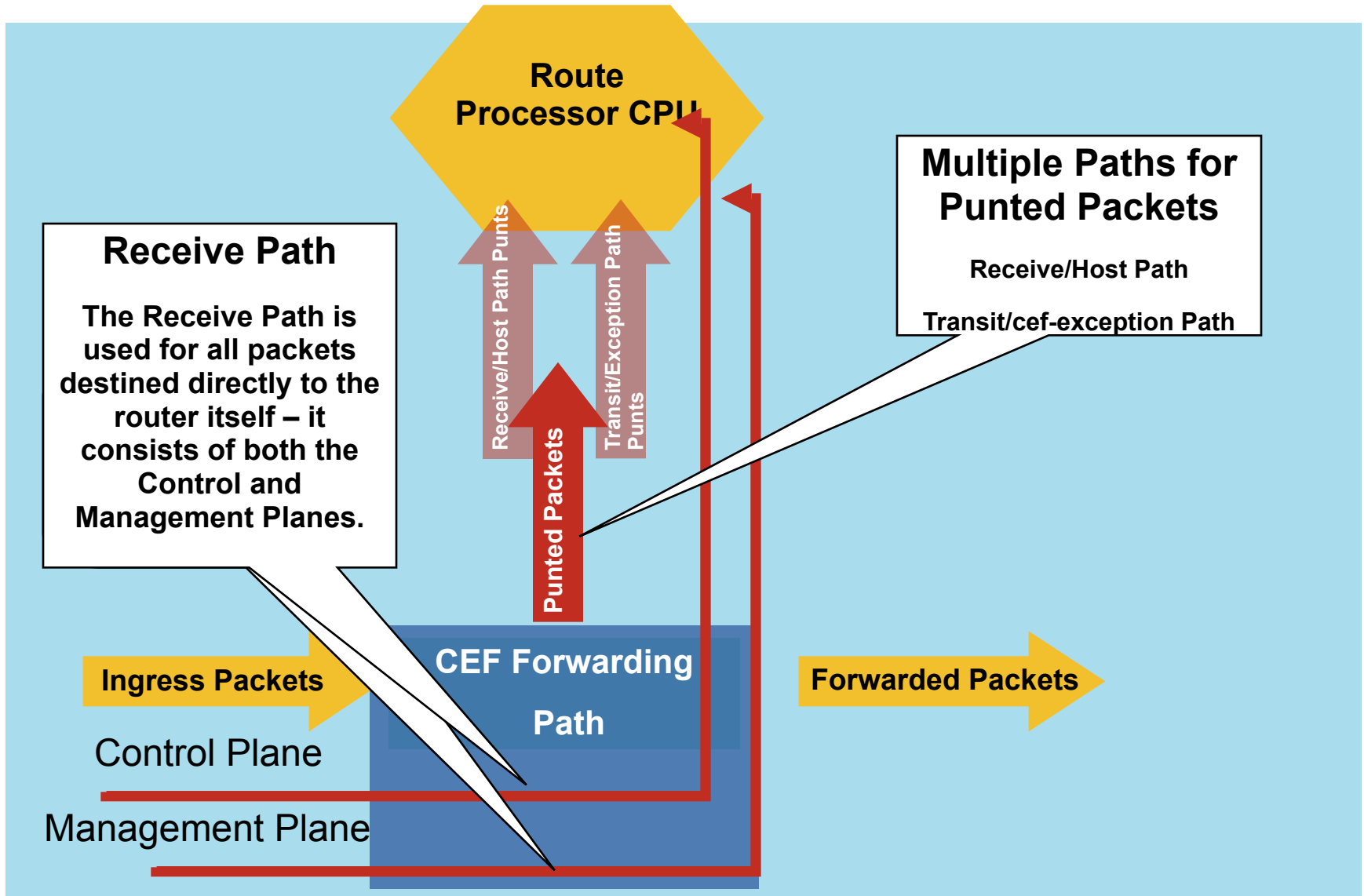
Broadcast and Multicast address space

```
router#sh ip cef
```

Prefix	Next Hop	Interface
255.255.255.255/32	receive	
10.1.2.0/24	172.16.1.216	GigabitEthernet3/0
10.1.3.0/24	172.16.1.216	GigabitEthernet3/0
224.0.0.0/24	receive	

- Packets with next hop **172.16.1.196/32** **receive** are sent to the router for processing
- Traffic usually routing protocols, management, and multicast control traffic

Receive Path



What Is a Punt?

- Receive adjacency
- Transit packets that need additional processing

Specific router configuration: ACL logging, Cisco IOS FW, etc.

IP Options set

Require fragmentation

ICMP Unreachables due to routing, MTU, or filtering

Expired TTL (ICMP Time Exceeded)

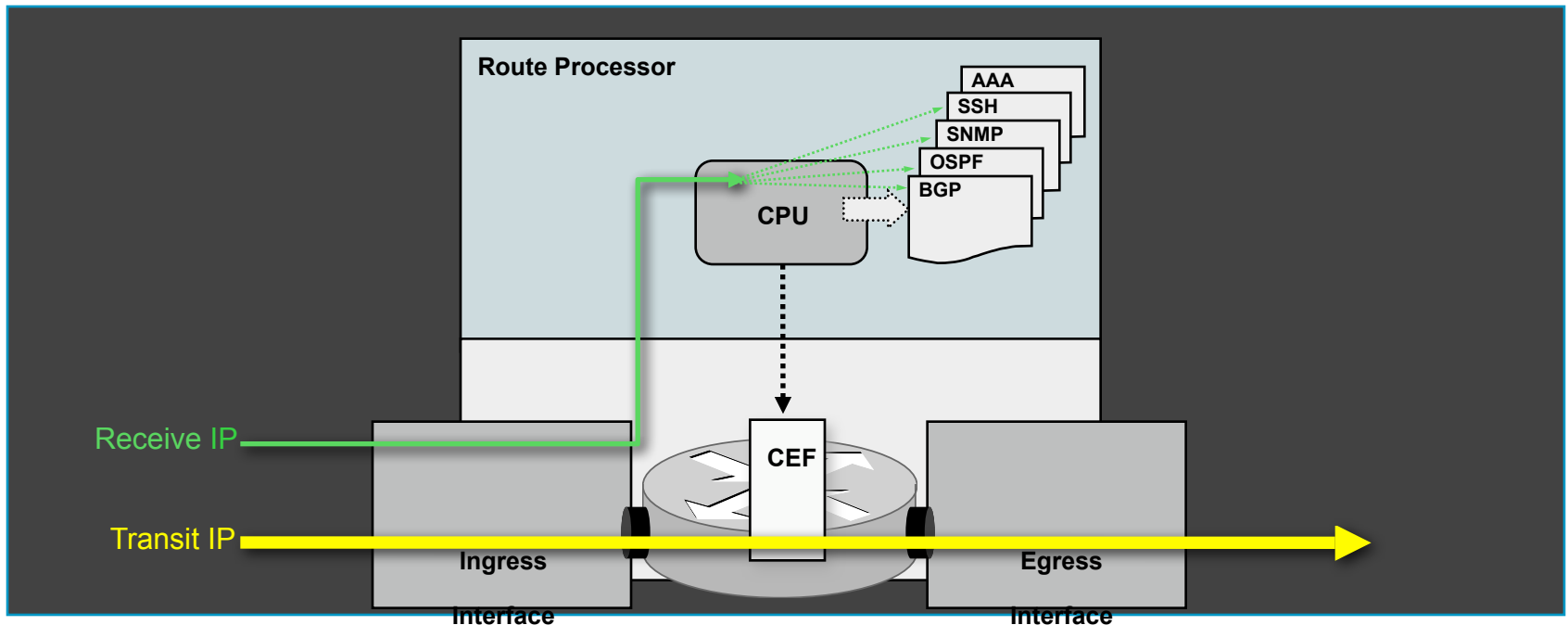
Destinations lacking a next-hop adjacency
(ARP—CEF Glean punt)

Malformed fields (ICMP Parameter error)

Exceptions IP Packets

Exceptions IP Packets

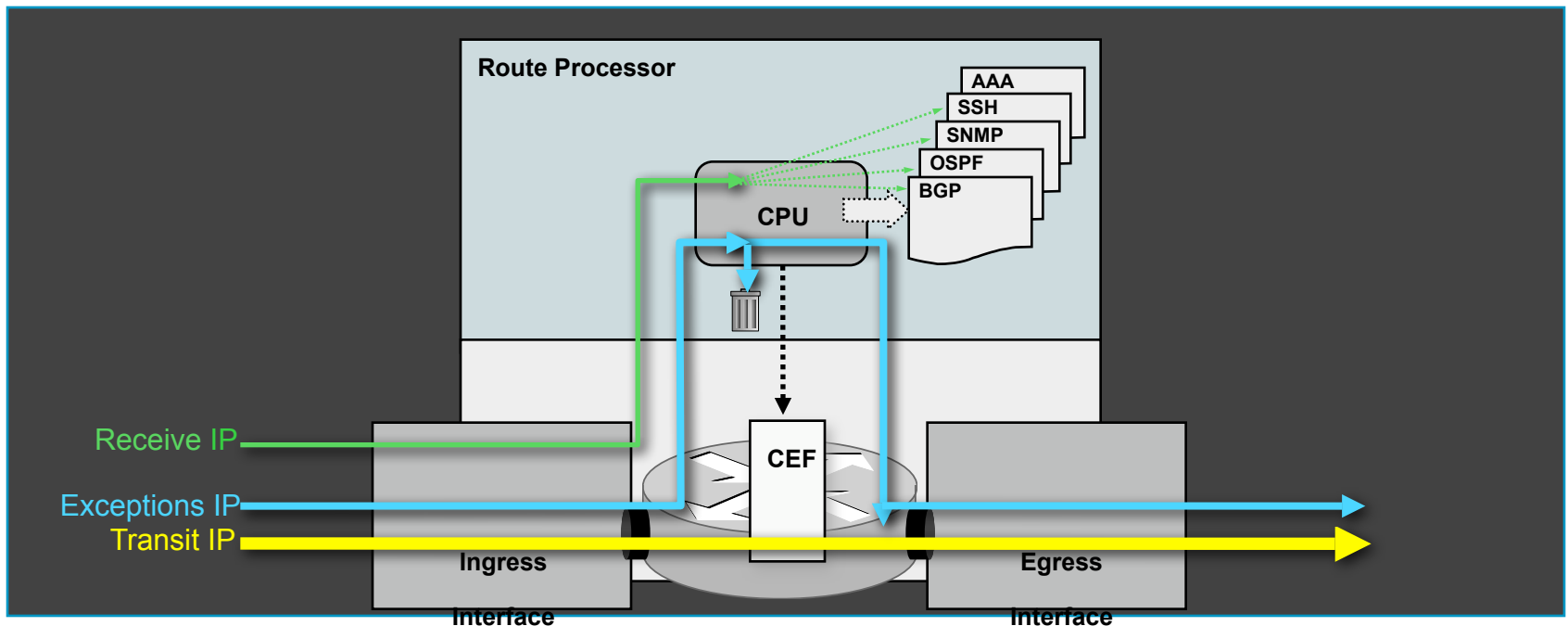
- Exception IP packets include, for example, IPv4 and IPv6 packets containing IP header options, IP packets with expiring TTLs, and certain transit IP packets under specific conditions, such as the first packet of a multicast flow or a new NAT session
- All of the packets in this set must be handled by the route processor



Exceptions IP Packets

Exceptions IP Packets

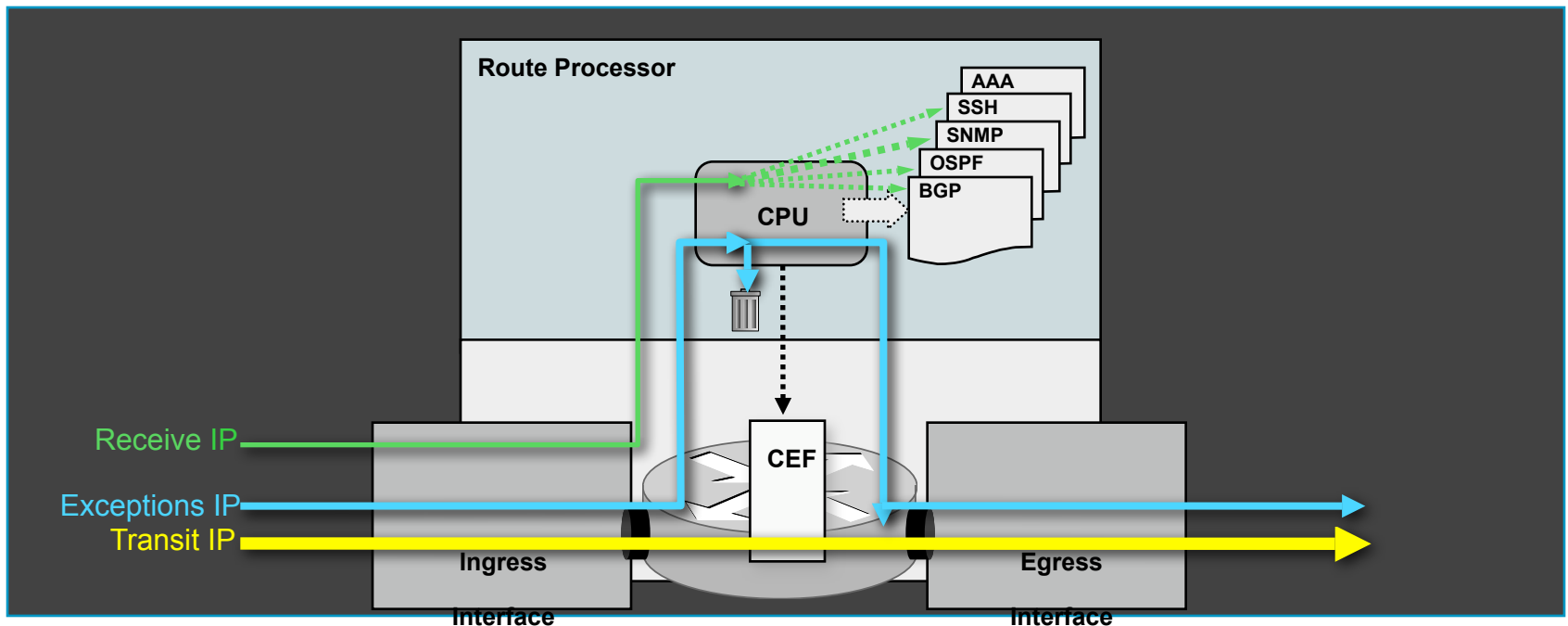
- Exception IP packets include, for example, IPv4 and IPv6 packets containing IP header options, IP packets with expiring TTLs, and certain transit IP packets under specific conditions, such as the first packet of a multicast flow or a new NAT session
- All of the packets in this set must be handled by the route processor



Non-IP Packets

Non-IP Packets

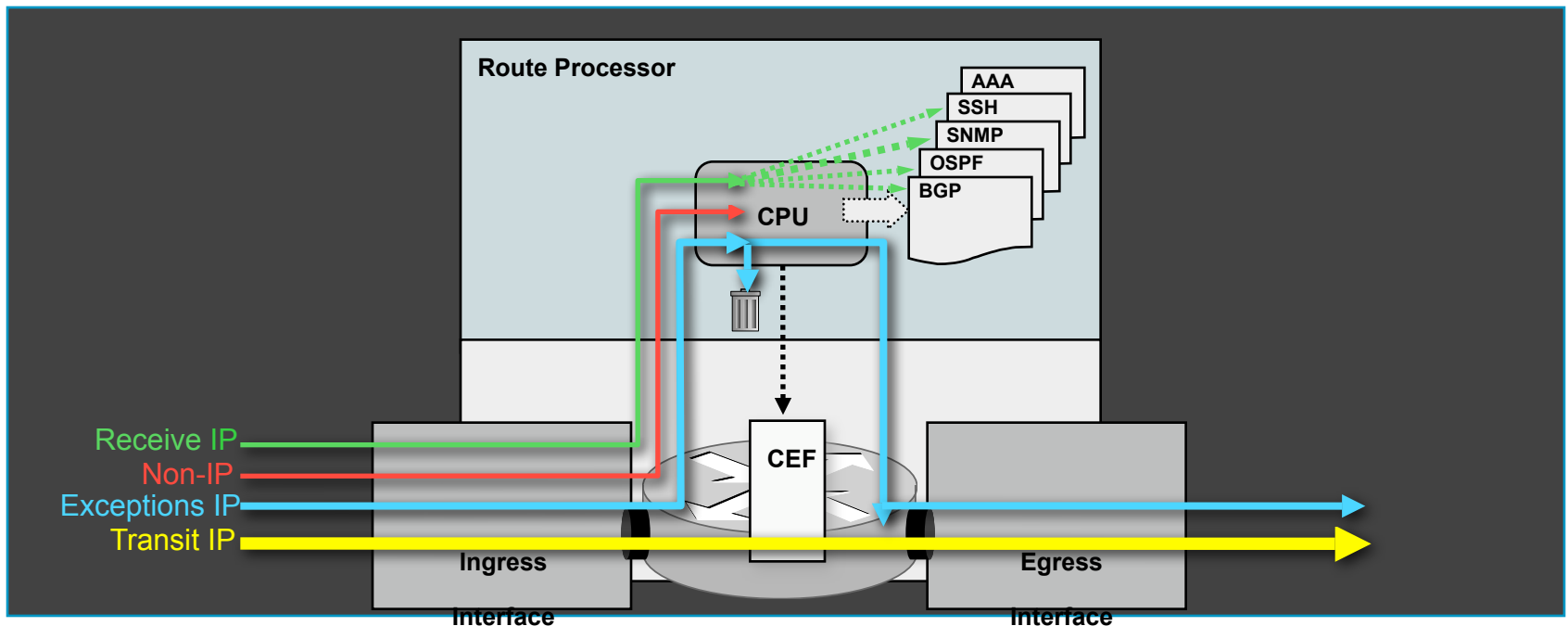
- Layer 2 keepalives, ISIS packets, Cisco Discovery Protocol (CDP) packets, and PPP Link Control Protocol (LCP) packets are examples of non-IP packets
- All of the packets in this set must be handled by the route processor



Non-IP Packets

Non-IP Packets

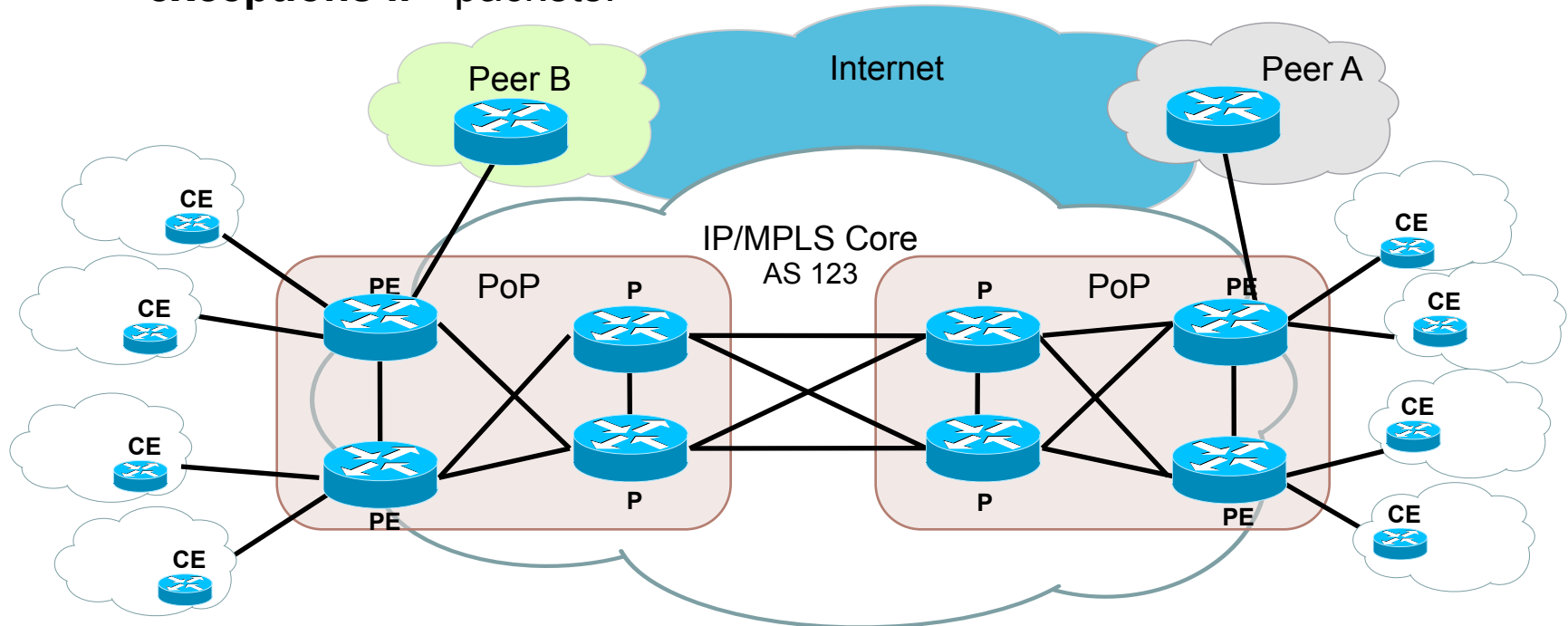
- Layer 2 keepalives, ISIS packets, Cisco Discovery Protocol (CDP) packets, and PPP Link Control Protocol (LCP) packets are examples of non-IP packets
- All of the packets in this set must be handled by the route processor



IP Data Plane

IP Data Plane

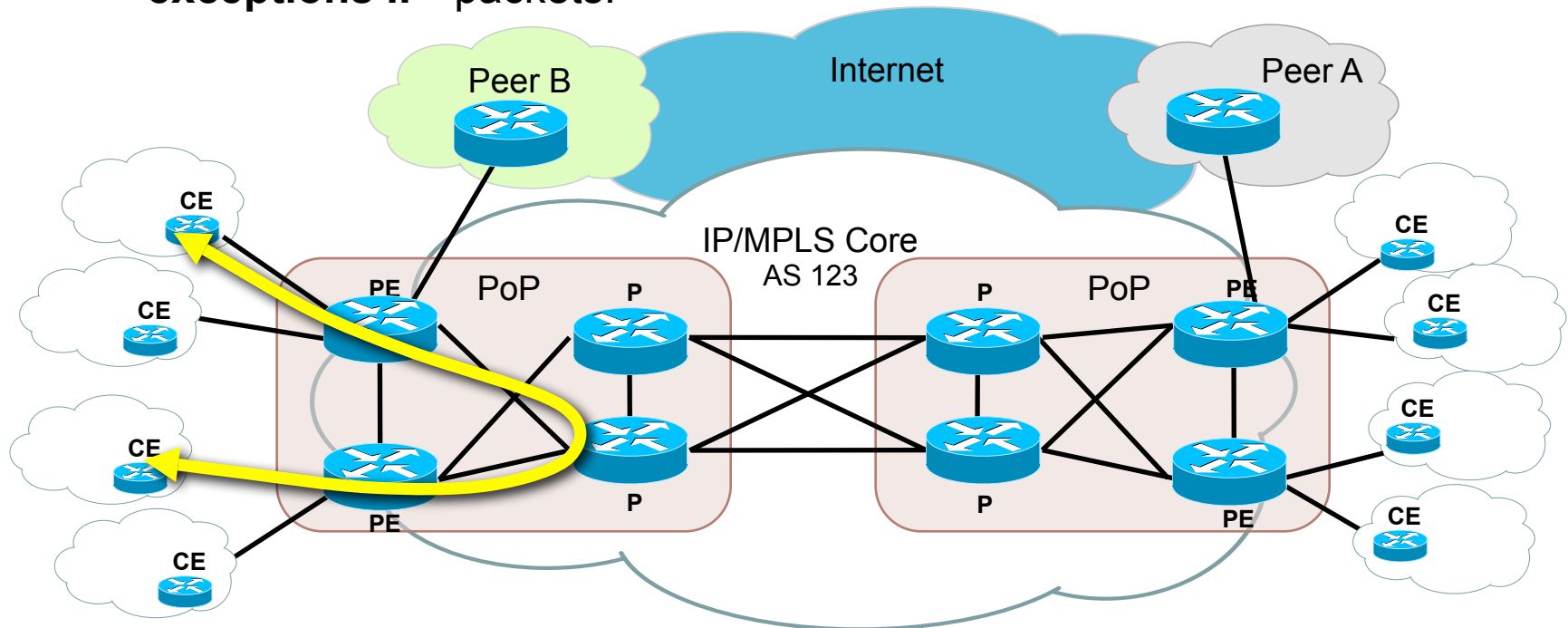
- The **logical** group containing all “**customer**” application traffic generated by hosts, clients, servers, and applications that are sourced from and destined to other devices
- Data plane traffic is always be seen as **transit** packets by network elements. Most will be forwarded in the fast path; some may be “**exceptions IP**” packets.



IP Data Plane

IP Data Plane

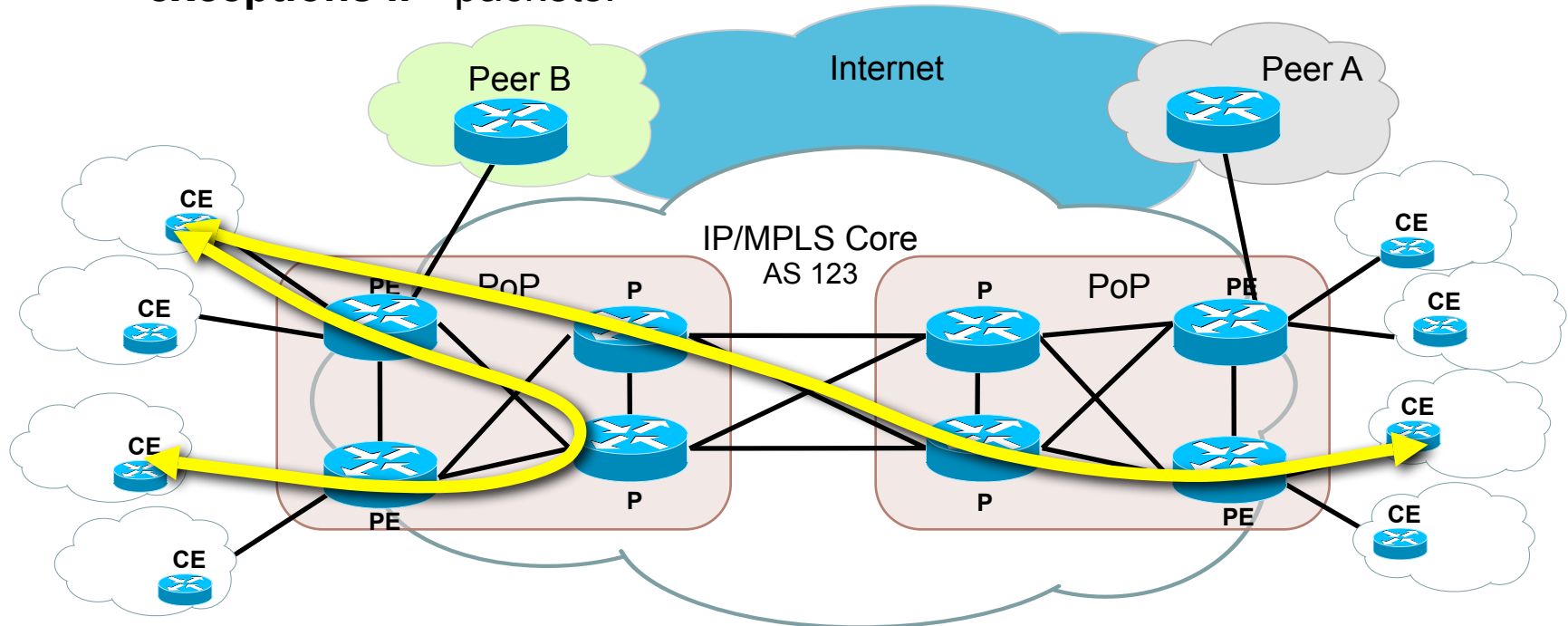
- The **logical** group containing all “**customer**” application traffic generated by hosts, clients, servers, and applications that are sourced from and destined to other devices
- Data plane traffic is always be seen as **transit** packets by network elements. Most will be forwarded in the fast path; some may be “**exceptions IP**” packets.



IP Data Plane

IP Data Plane

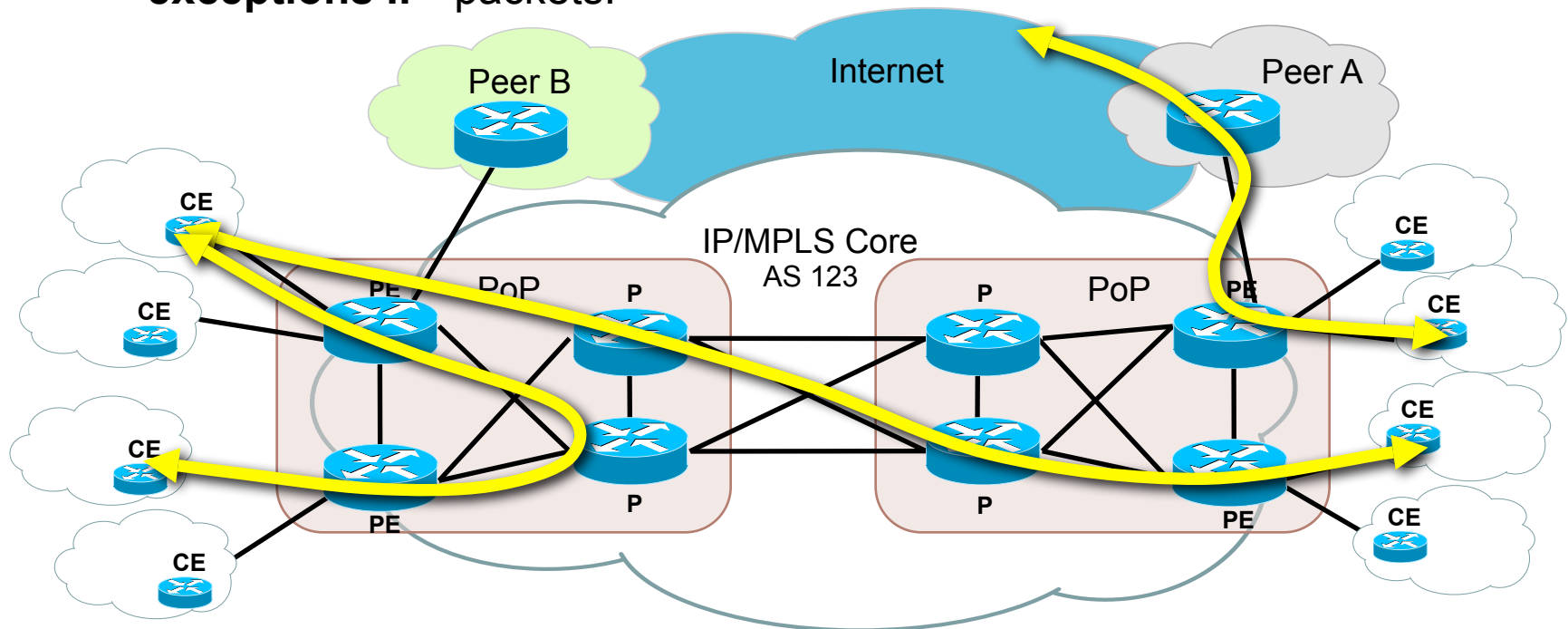
- The **logical** group containing all “**customer**” application traffic generated by hosts, clients, servers, and applications that are sourced from and destined to other devices
- Data plane traffic is always be seen as **transit** packets by network elements. Most will be forwarded in the fast path; some may be “**exceptions IP**” packets.



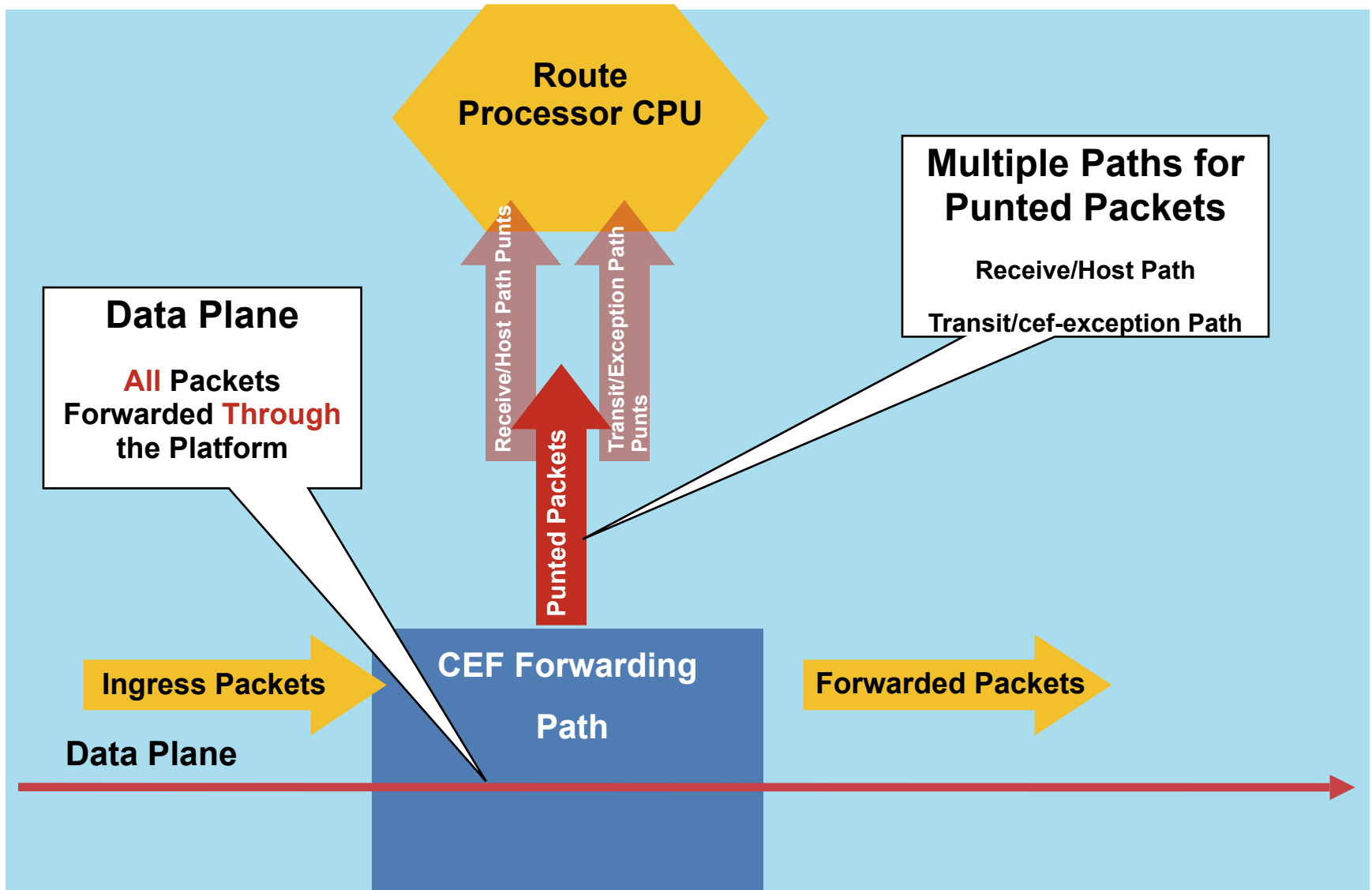
IP Data Plane

IP Data Plane

- The **logical** group containing all “**customer**” application traffic generated by hosts, clients, servers, and applications that are sourced from and destined to other devices
- Data plane traffic is always be seen as **transit** packets by network elements. Most will be forwarded in the fast path; some may be “**exceptions IP**” packets.



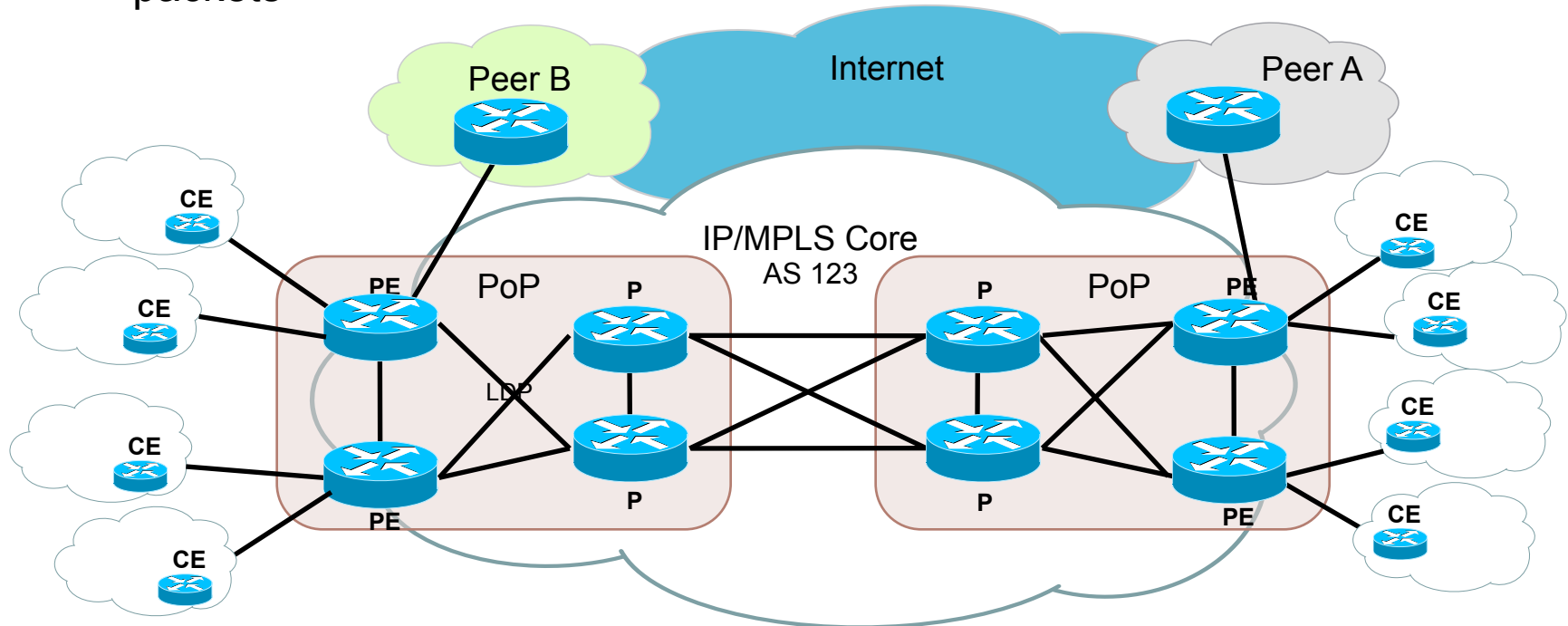
Data Plane



IP Control Plane

IP Control Plane

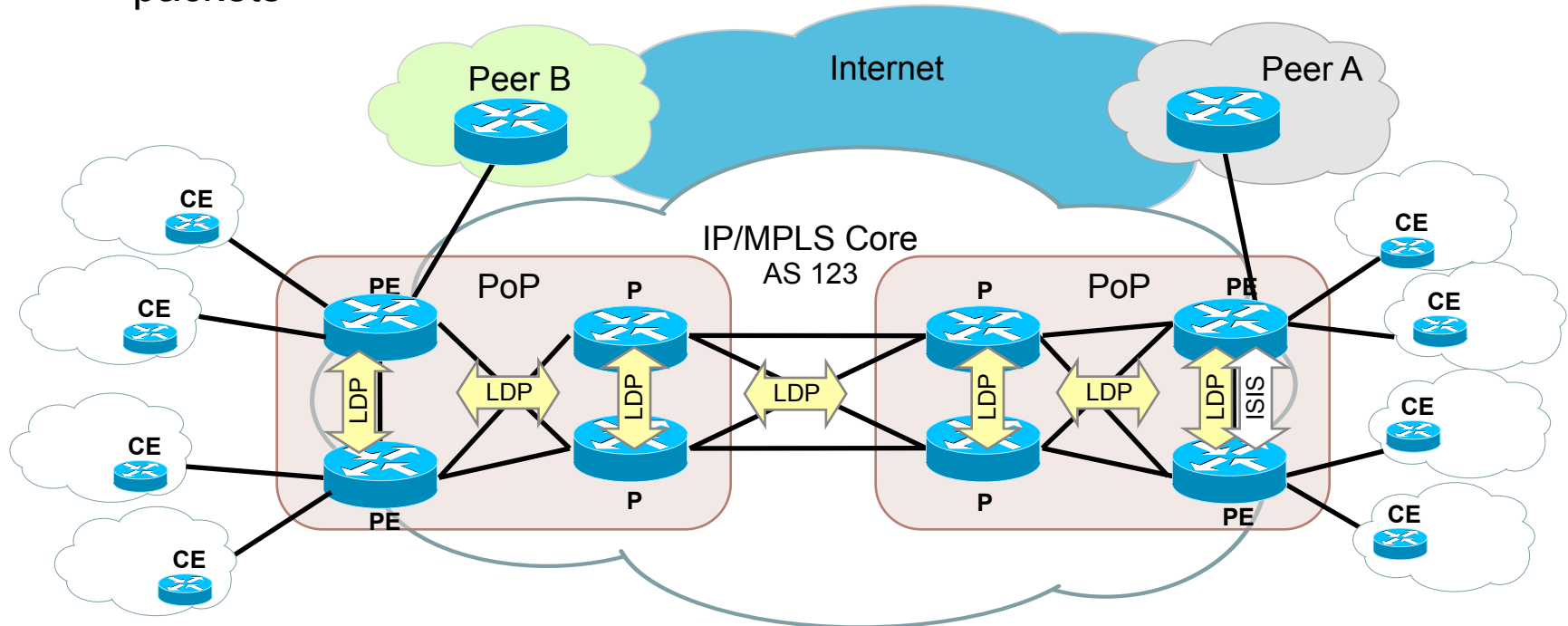
- The **logical** group containing all **routing**, **signaling**, **link-state**, and other control protocols used to create and maintain the state of the network and interfaces.
- Control plane traffic always includes **receive** packets from the perspective of the src/dst network element, but **logically** includes certain transit packets



IP Control Plane

IP Control Plane

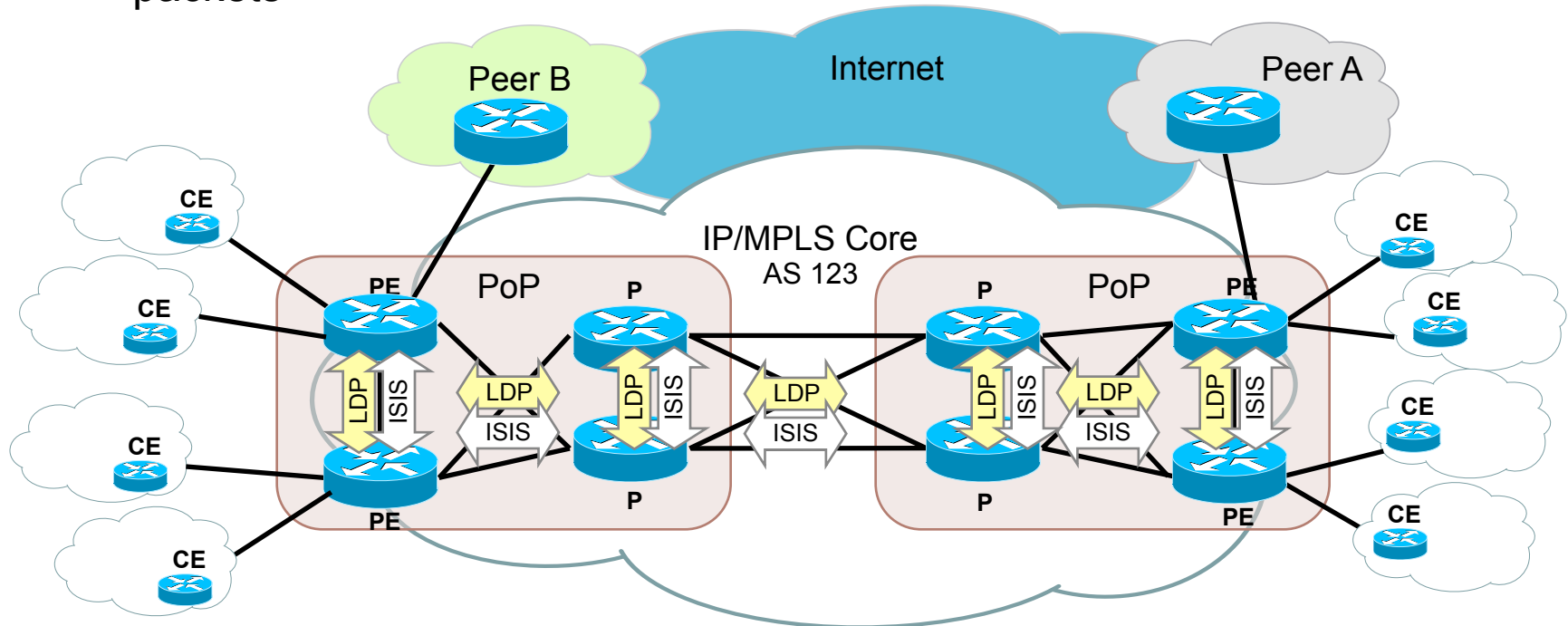
- The **logical** group containing all **routing, signaling, link-state**, and other control protocols used to create and maintain the state of the network and interfaces.
- Control plane traffic always includes **receive** packets from the perspective of the src/dst network element, but **logically** includes certain transit packets



IP Control Plane

IP Control Plane

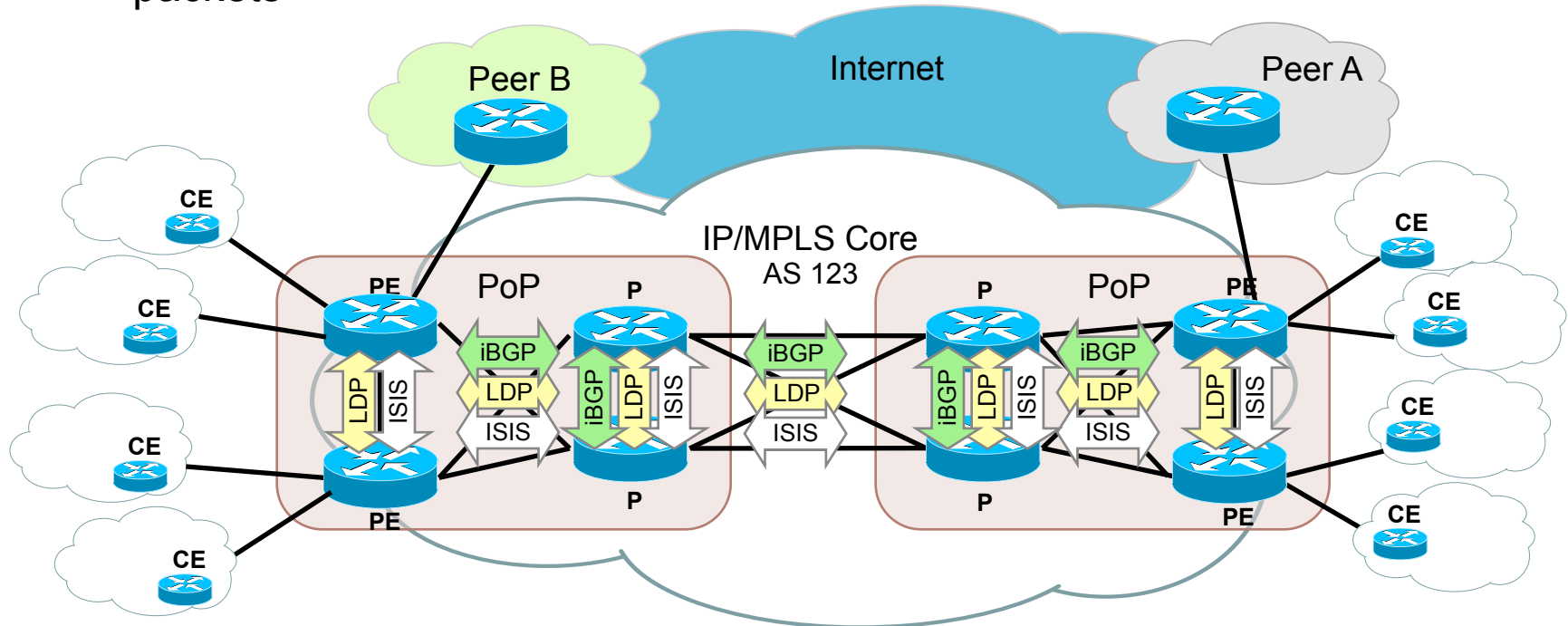
- The **logical** group containing all **routing**, **signaling**, **link-state**, and other control protocols used to create and maintain the state of the network and interfaces.
- Control plane traffic always includes **receive** packets from the perspective of the src/dst network element, but **logically** includes certain transit packets



IP Control Plane

IP Control Plane

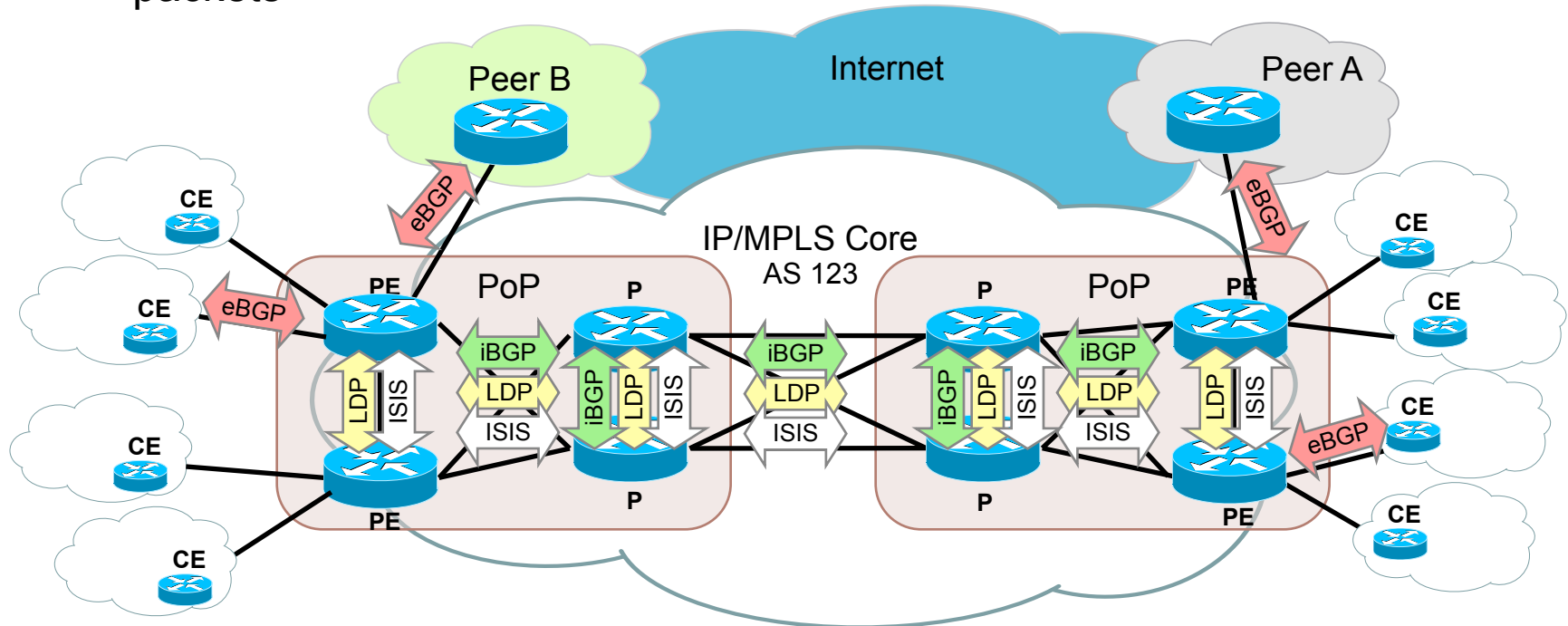
- The **logical** group containing all **routing, signaling, link-state**, and other control protocols used to create and maintain the state of the network and interfaces.
- Control plane traffic always includes **receive** packets from the perspective of the src/dst network element, but **logically** includes certain transit packets



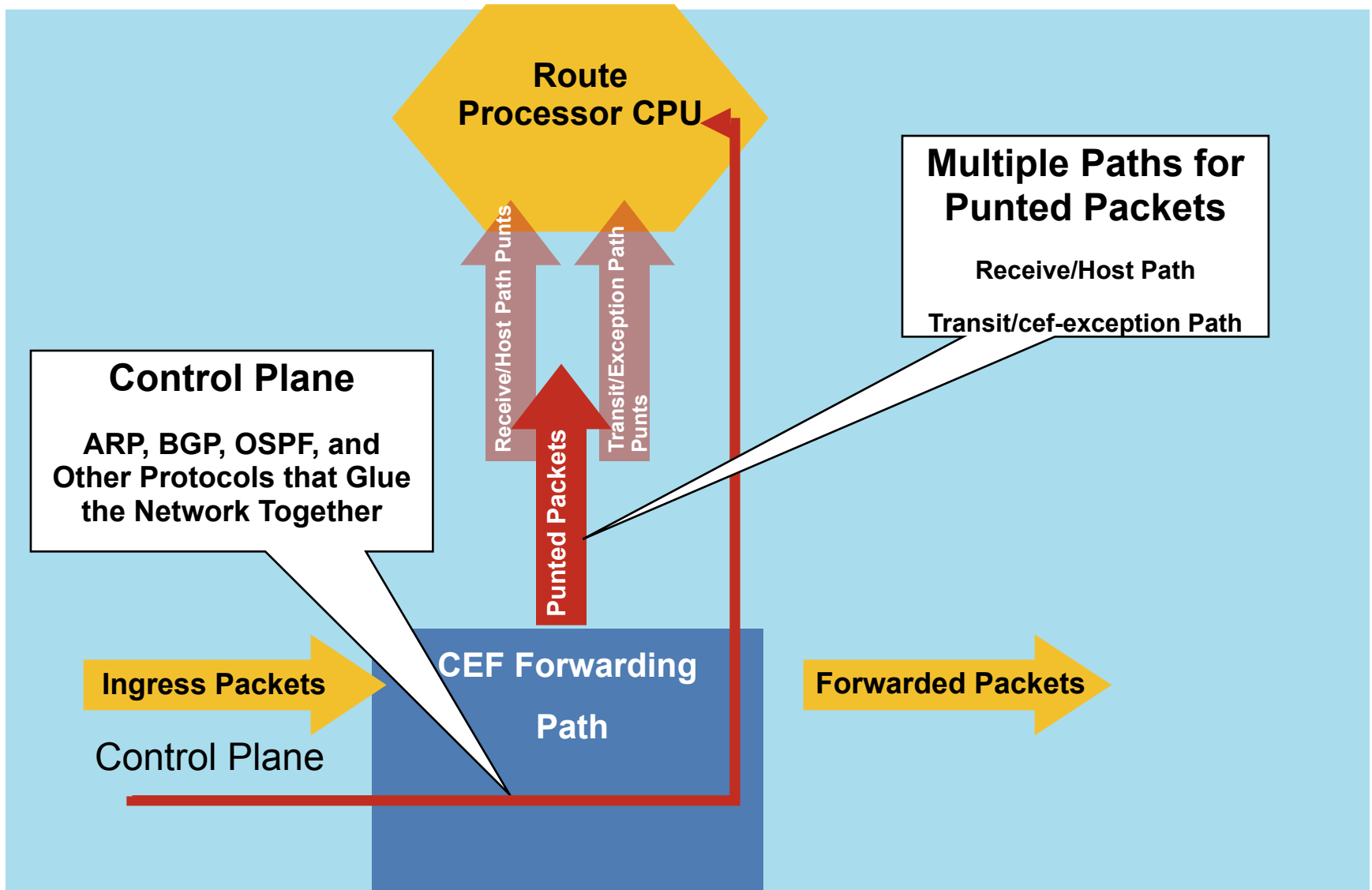
IP Control Plane

IP Control Plane

- The **logical** group containing all **routing, signaling, link-state**, and other control protocols used to create and maintain the state of the network and interfaces.
- Control plane traffic always includes **receive** packets from the perspective of the src/dst network element, but **logically** includes certain transit packets



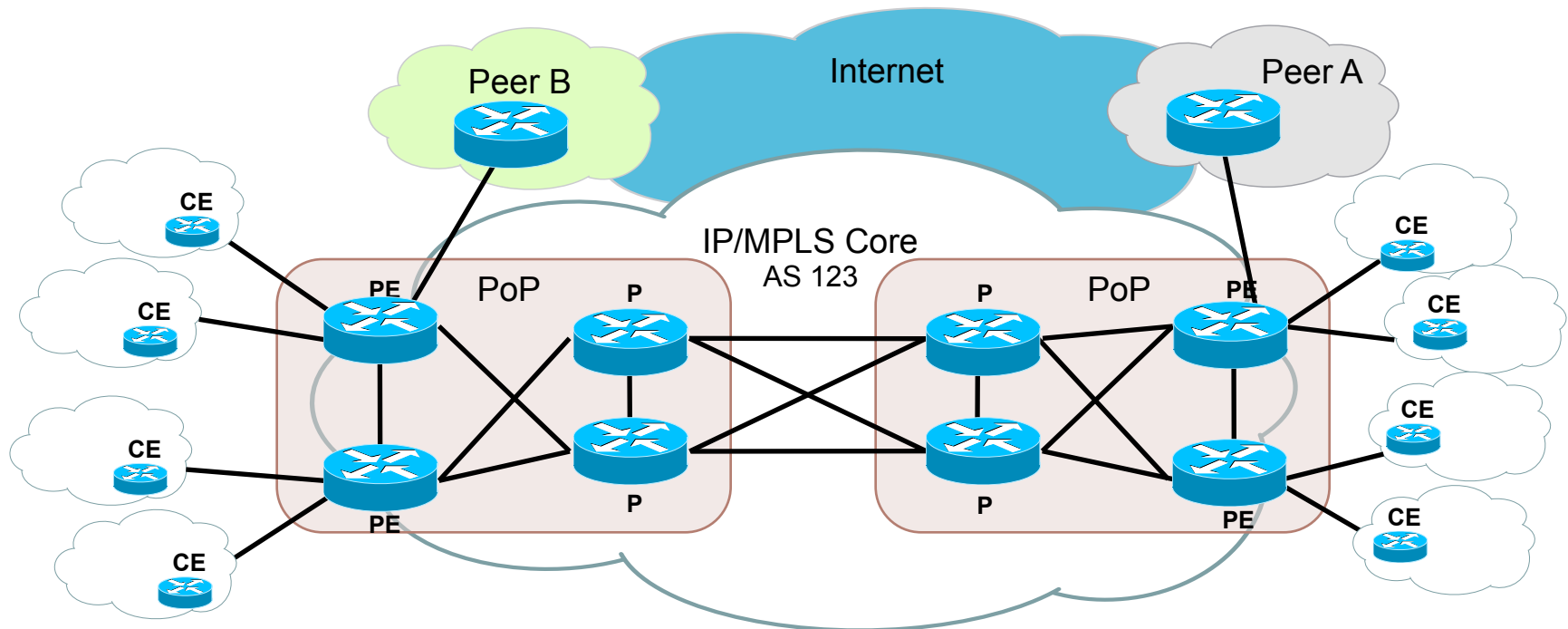
Control Plane



IP Management Plane

IP Management Plane

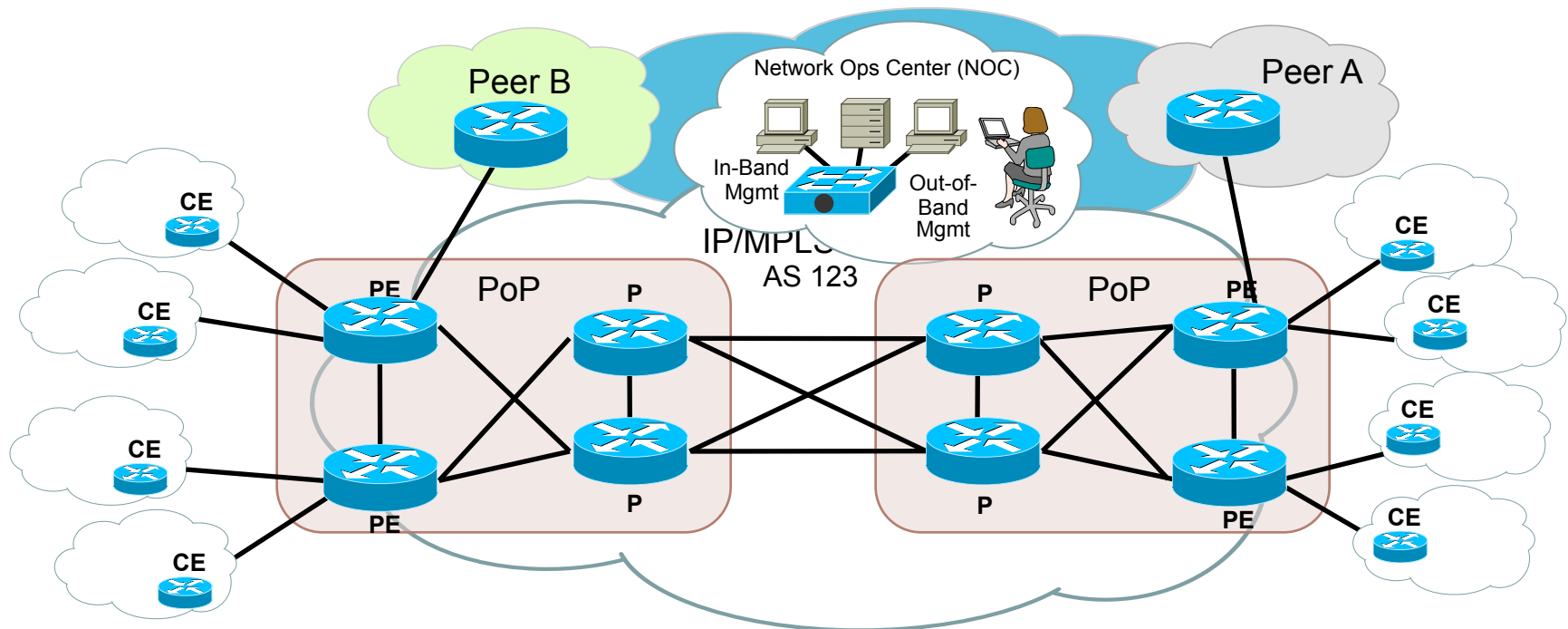
- The **logical** group containing all **management** traffic supporting provisioning, maintenance, and monitoring functions for the network..
- Management plane traffic always includes **receive** packets from the perspective of the src/dst network element, but **logically** includes certain **transit** packets



IP Management Plane

IP Management Plane

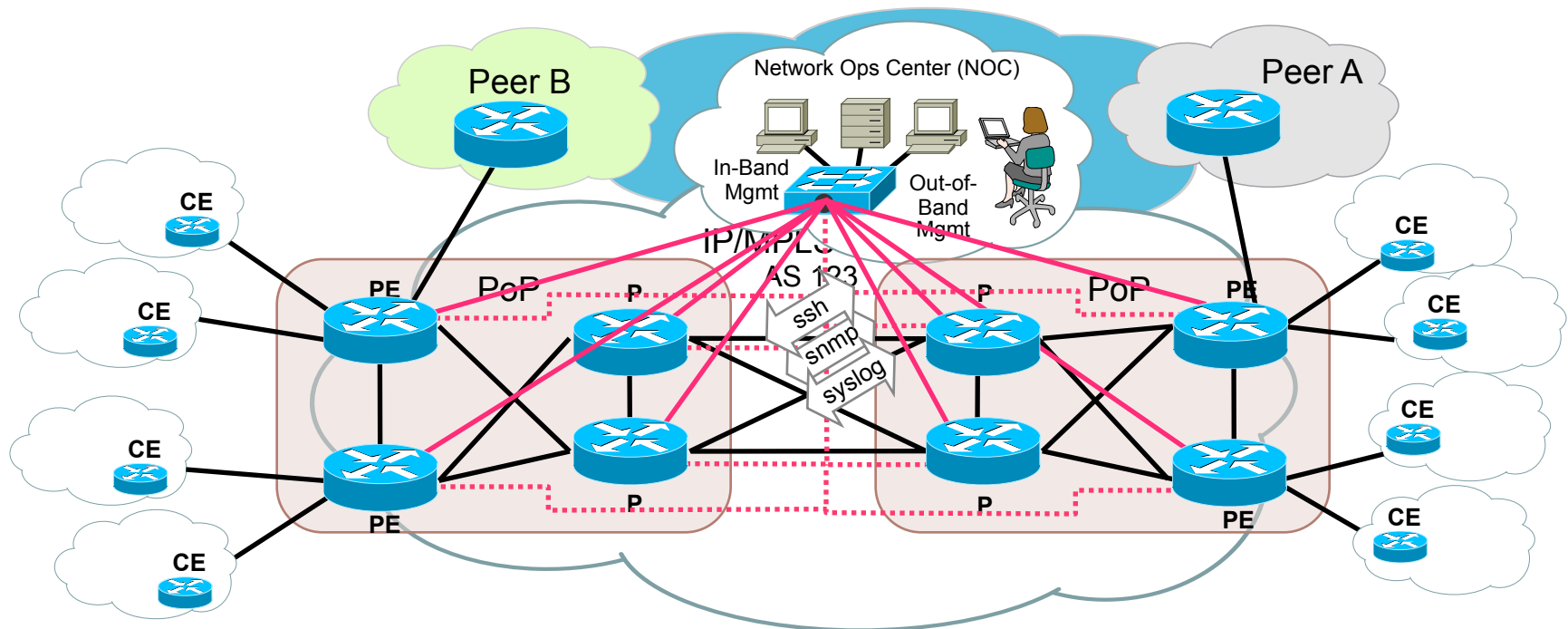
- The **logical** group containing all **management** traffic supporting provisioning, maintenance, and monitoring functions for the network..
- Management plane traffic always includes **receive** packets from the perspective of the src/dst network element, but **logically** includes certain **transit** packets



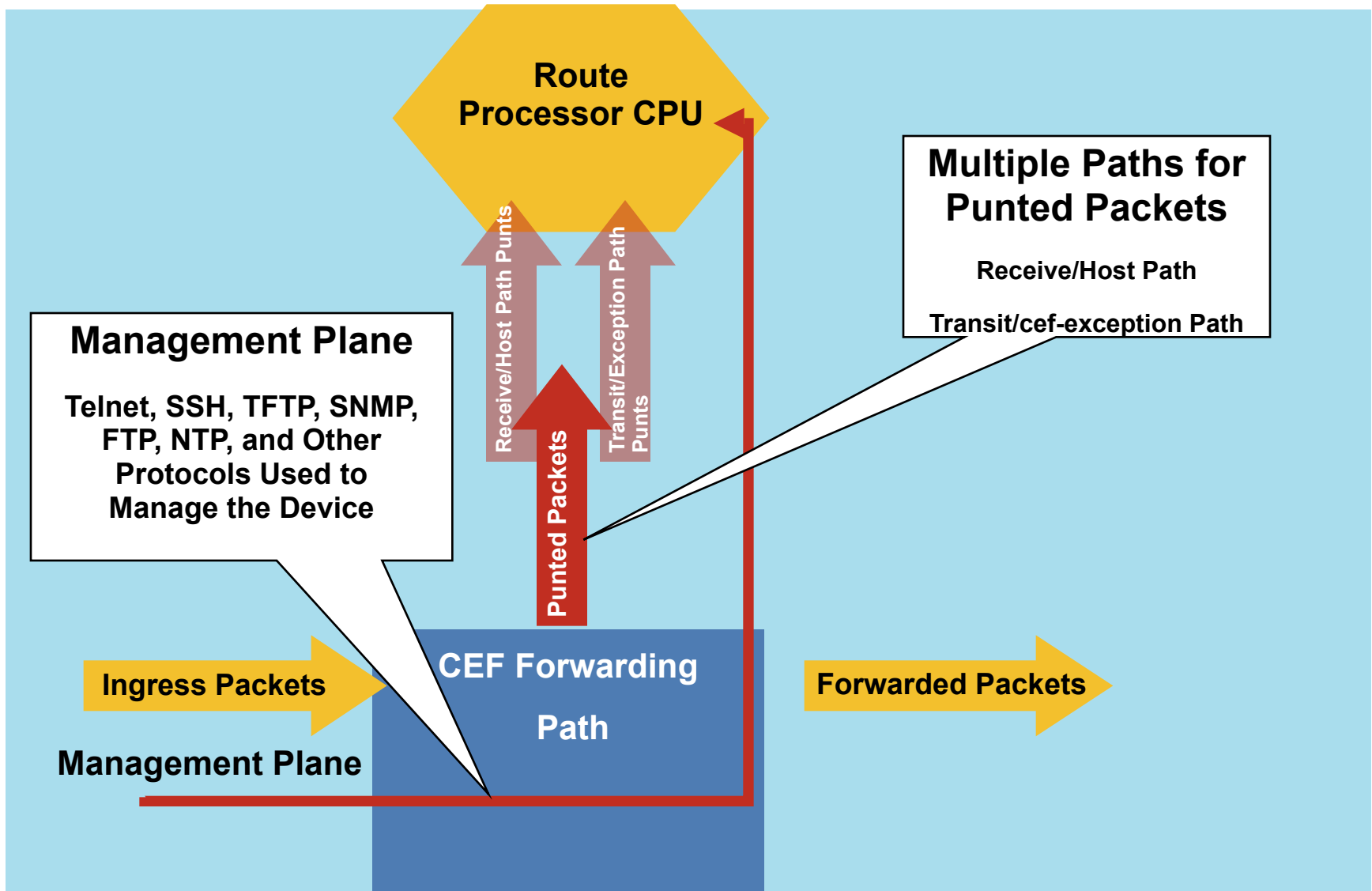
IP Management Plane

IP Management Plane

- The **logical** group containing all **management** traffic supporting provisioning, maintenance, and monitoring functions for the network..
- Management plane traffic always includes **receive** packets from the perspective of the src/dst network element, but **logically** includes certain **transit** packets



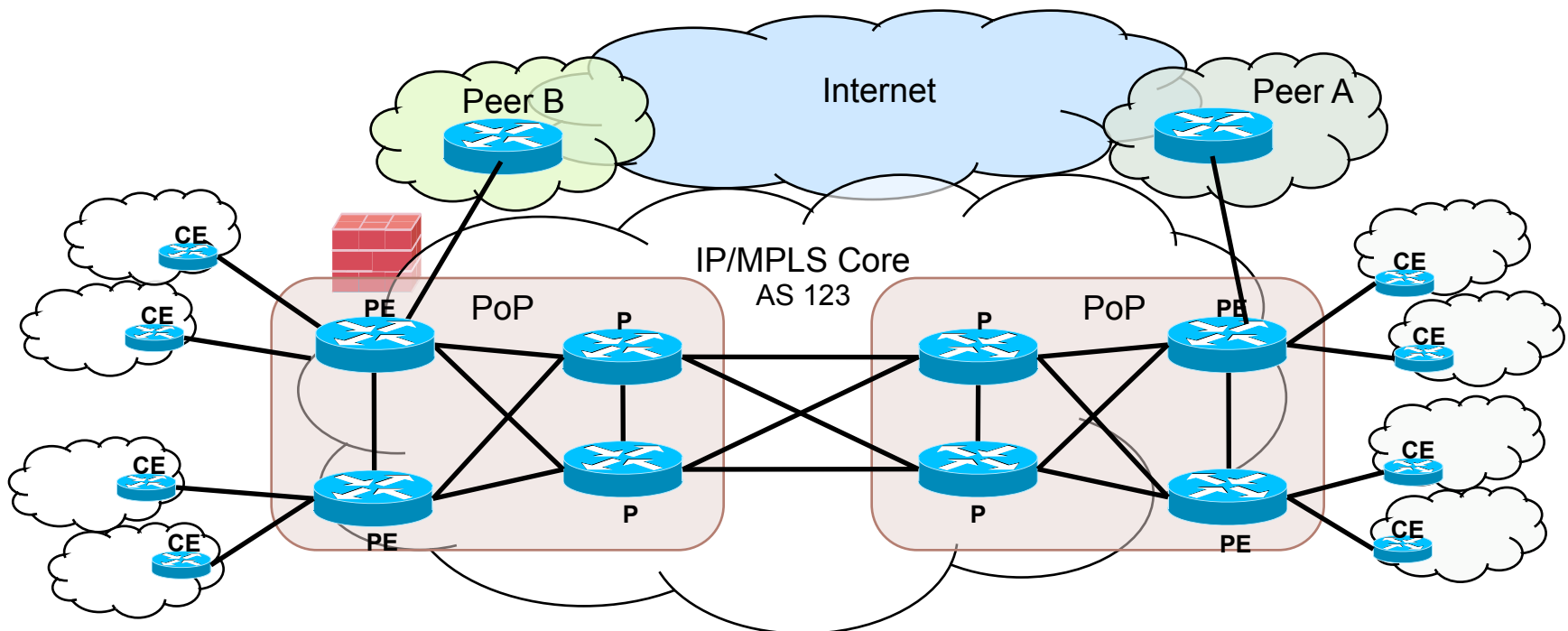
Management Plane



IP Services Plane

IP Services Plane

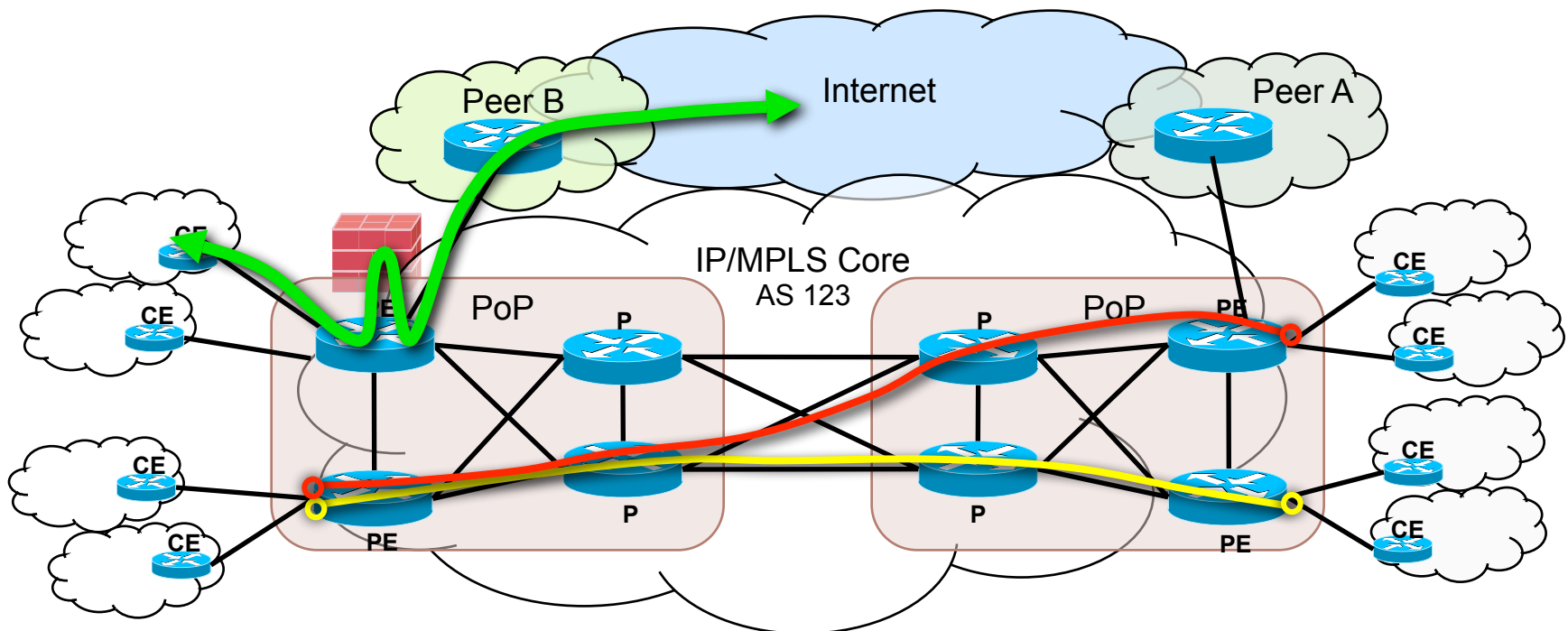
- The logical group containing “**customer**” traffic (like the data plane), but with the major difference that this traffic requires **specialized forwarding functions** applied it, and possibly consistent handling applied end to end.
- Services plane traffic is “**transit**” traffic, but network elements use **special handling** to apply or enforce the intended policies for various service types



IP Services Plane

IP Services Plane

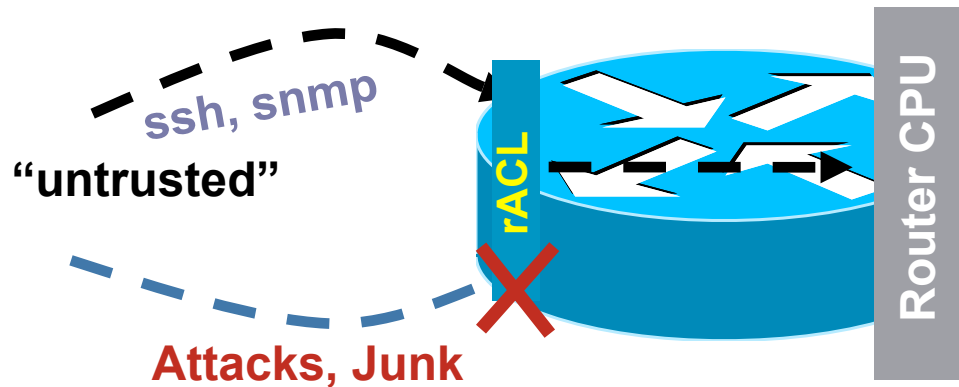
- The logical group containing “**customer**” traffic (like the data plane), but with the major difference that this traffic requires **specialized forwarding functions** applied it, and possibly consistent handling applied end to end.
- Services plane traffic is “**transit**” traffic, but network elements use **special handling** to apply or enforce the intended policies for various service types



RP Protection: Receive ACLs



Receive ACLs



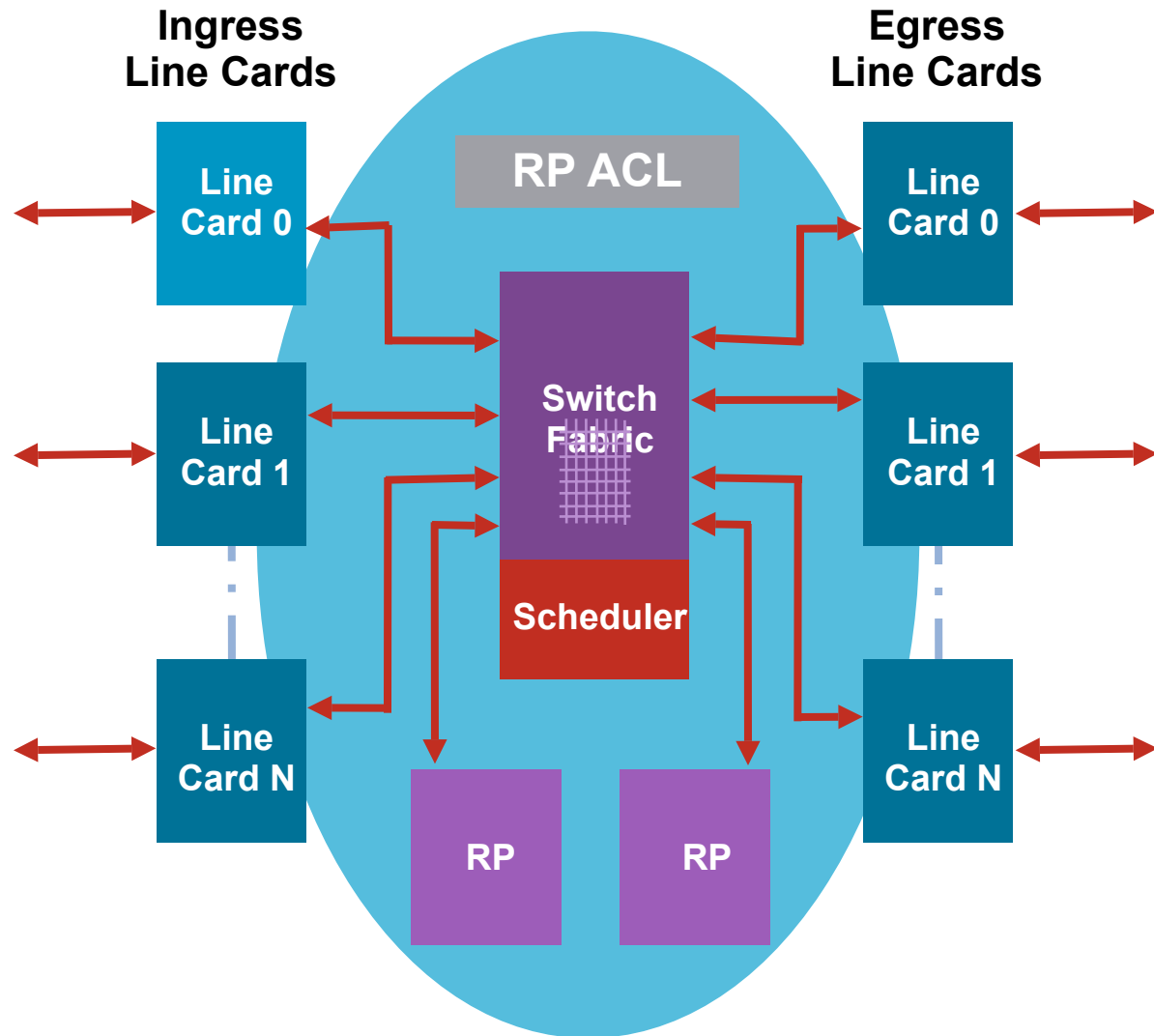
- rACL protects the CPU from undesired traffic from any line card
- rACL is executed on all receive adjacency packets at LC before queued to the RP
- rACL does **not** affect transit traffic
- rACLs enforce security policy by filtering who/what can access the router

Receive ACL Command

- Introduced in 12.0(21)S2/12.0(22)S for GSR and 12.0(24)S for 7500
- Global command
- `ip receive access-list [number]`
Standard, extended or compiled ACL
- As with other ACL types, show access-list provide ACE hit counts
- Log keyword can be used for more detail

Receive ACL

- Standard, extended, or turbo ACL is created on the RP; this ACL is then pushed down to all the line cards
- The rACL is executed on all receive adjacency packets before they are queued to be sent to the RP



rACL: Building Your ACL

- Develop list of required protocols
- Develop address requirements
- Determine interface on router
 - Does the protocol access one interface?
 - Many interfaces?
 - Loopback or real?
- Deployment is an iterative process
 - Start with relatively “open” lists → tighten as needed

rACL: Sample Entries

■ OSPF

```
access-list 110 permit ospf host ospf_neighbour host 224.0.0.5
! DR multicast address, if needed
access-list 110 permit ospf host ospf_neighbour host 224.0.0.6
access-list 110 permit ospf host ospf_neighbour host local_ip
```

■ BGP

```
access-list 110 permit tcp host bgp_peer host loopback eq bgp
access-list 110 permit tcp host bgp_peer eq bgp host loopback
```

■ EIGRP

```
access-list 110 permit eigrp host eigrp_neighbour host 224.0.0.10
access-list 110 permit eigrp host eigrp_neighbour host local_ip
```

rACL: Sample Entries

- SSH/Telnet

```
access-list 110 permit tcp management_addresses host loopback eq 22
```

```
access-list 110 permit tcp management_addresses host loopback eq telnet
```

- SNMP

```
access-list 110 permit udp host NMS_stations host loopback eq snmp
```

- Traceroute (router originated)

```
! Each hop returns a ttl exceeded (type 11, code 3) message and  
the ! final destination returns an ICMP port unreachable  
(type3,code0)
```

```
access-list 110 permit icmp any routers_interfaces ttl-exceeded
```

```
access-list 110 permit icmp any routers_interfaces port-unreachable
```

Control Plane Policing



Control Plane Policing (CoPP)

- rACLs are great but
 - Only available on GSR/7500
 - Limited granularity—permit/deny only
- Need to protect all platforms
 - To achieve protection today, need to apply ACL to all interfaces
 - Some platform implementation specifics
- Some packets need to be permitted but at limited rate
 - Think ping :-)

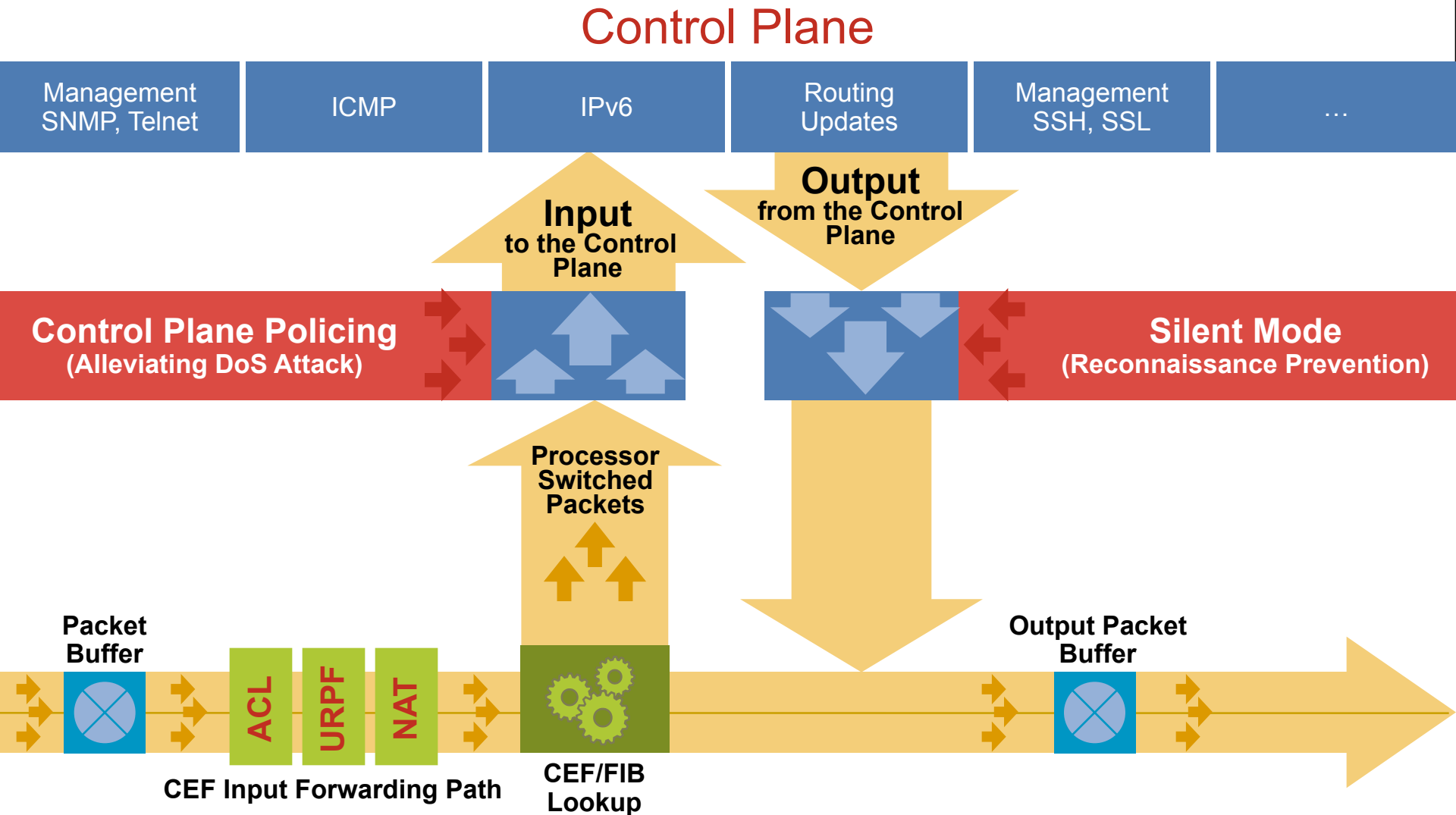
Control Plane Policing (CoPP)

- DoS attacks targeting CPU typically involve high rates of CPU destined traffic
- Symptoms include:
 - High CPU utilization (near 100%)
 - Loss of L2 keepalives and routing protocol updates
 - Interactive sessions slow or unresponsive
 - Exhaustion of memory and buffer resources
- CoPP protects control and management planes
 - Ensures routing stability
 - Reachability
 - Packet delivery

Control Plane Policing (CoPP)

- Protection against DoS attacks targeted toward the network infrastructure
- Single point of application via control-plane “interface”
- Increases the reliability, security, and availability of the network devices
- MQC based framework allows consistent implementation strategy across all Cisco hardware
- Class Based QoS (CBQoS) MIB allows for SNMP information polling

Protecting the Control Plane



Configuring CoPP

1. Classify traffic
router(config)# ip access-list extended <acl name>
router(config-ext-nacl)# permit / deny <classification criteria>
2. Define traffic classes
router(config)# class-map <traffic_class_name>
router(config-cmap)# match <access-group>
3. Define service policy
router(config)# policy-map <service_policy_name>
router(config-pmap)# class <traffic_class_name>
router(config-pmap-c)# police <rate> conform-action transmit
exceed-action drop
4. Apply CoPP policy
router(config)# control-plane
router(config-cp)# service-policy input <service_policy_name>

Sample CoPP Configuration

```
Router(config)# access-list 140 permit tcp host 10.1.1.1 any eq 22
Router(config)# access-list 140 permit udp host 10.1.1.2 any eq snmp
```

```
Router(config)# class-map mgmt-class
Router(config-cmap)# match access-group 140
Router(config-cmap)# exit
```

Traffic to Be Rate-Limited: SNMP and SSH from Mgmt Host

Define Class-Map for This Traffic

```
Router(config)# policy-map control-plane-policy
Router(config-pmap)# class mgmt-class
Router(config-pmap-c)# police 80000 conform transmit exceed drop
Router(config-pmap-c)# exit
Router(config-pmap)# exit
```

Define the Policy for This Class Map: Up to 80 Kbps: Transmit, Else Drop

```
Router(config)# control-plane
Router(config-cp)# service-policy input control-plane-policy
Router(config-cp)# exit
```

Apply Policy to Control-Plane

Deploying CoPP

- Do you know what rate of TCP/179 traffic is normal or acceptable?
- Deploying as “rACL” replacement relatively simple
 - I know that I need BGP/OSPF/etc., deny all else
- CoPP ‘Default’ class **catches** the traffic that would be dropped by the rACL
- To get the most value from CoPP, detailed planning is required
 - bps vs. PPS (Caveat: Cat6K only does bps)
 - Inbound vs. Outbound
 - Layer 2 traffic—ARP is ‘handled’ (Caveat: not on Cat6k)

Deploying CoPP: Challenges

- Every network is going to have different rates for traffic
 - Only time and experience will help
 - Show commands can help - ACL hits and rate information
- Real-world hardware vs. software performance implications
- Available in 12.3(4)T and 12.2(18)S
- Deployment CoPP:

http://www.cisco.com/en/US/products/ps6642/products_white_paper0900aecd804fa16a.shtml

Deploying CoPP: The Approach

1. Drop packets prior to CoPP

ACLs, uRPF, and additional features

2. Identify necessary protocols and maybe even initial rate-limiting values

Based on configurations, NetFlow data, classification ACLs, and show commands (show ip traffic, show ip socket, etc.)

3. Develop and pilot CoPP framework without enforcing rate-limits

4. Refine policy/adjust rates based on observation

5. Deploy policy and enforce rate-limits as required

Dropping Packets Prior to CoPP

- Interface ACLs, uRPF, etc.
- IP Options Handling
 - “ip options drop”
 - “ip options ignore” (GSR only)
 - More efficient than CoPP but does not allow rate-limiting
- ACL Logging
 - “ip access-list logging interval <msec>”
 - Rate-limits logging punts—effective when logging **denied** packets
- Disable unnecessary services to minimize punts

Identify Necessary Protocols

- Configuration review

- NetFlow Data

Punts show up with output ifIndex of 0/zero/null

(More on this later)

- “show ip socket” and “show tcp tcb”
- “show cef not-cef-switched”
- “show ip traffic”
- “show ip protocols”
- Classification ACLs

Sample Traffic Classification

1. **Known Undesirable** - malicious traffic we expect to see - fragments and the like - drop
2. **Critical Traffic** - routing protocols—control plane—likely no rate-limit
3. **Important Traffic** - SNMP, SSH, AAA, NTP - management plane - maybe rate-limit
4. **Normal Traffic** - other expected non-malicious traffic - ping and other ICMP - rate-limit
5. **Reactive Undesirable** - reactive handling of potentially malicious traffic, i.e. vulnerabilities - drop
6. **Catch-all** - remaining unclassified IP traffic - rate-limit
7. **Default** - non-IP traffic - maybe rate-limit

Known Undesirable Traffic

- ! **Known Undesirable** - Traffic that should never touch the RP

```
ip access-list extended known-undesirable-acl
```

```
    permit tcp any any fragments
```

```
    permit udp any any fragments
```

```
    permit icmp any any fragments
```

```
    permit ip any any fragments
```

```
    permit udp any any eq 1434
```

- Permit means “match” as opposed to “allow”
- Security vulnerabilities go here
- Do you need fragments?

Critical Traffic

- **! Critical - Defined as routing protocols**

```
ip access-list extended critical-acl
```

```
! iBGP peers
```

```
permit tcp <iBGP loopback space> gt 1024 <iBGP loopback space> eq bgp
```

```
permit tcp <iBGP loopback space> eq bgp <iBGP loopback space> gt 1024
```

```
! eBGP peers
```

```
permit tcp <eBGP peer> gt 1024 <eBGP peering address> eq bgp
```

```
permit tcp <eBGP peer> eq bgp <eBGP peering address> gt 1024
```

```
! IGP
```

```
permit ospf <core address space> host 224.0.0.5
```

```
permit ospf <core address space> host 224.0.0.6
```

```
permit ospf <core address space> <core address space>
```

- Use summarization to your advantage - loopbacks and multicast

Important Traffic

- **! Important** - Defined as traffic required to manage the router

```
ip access-list extended important-acl
```

```
  permit tcp <core address space> eq 22 any established
```

```
  permit tcp <core address space> any eq 22
```

```
  permit tcp <AAA hosts> eq tacacs <loopback space> established
```

```
  permit udp <NMS systems> <loopback space> eq snmp
```

- Specify source address space to limit zone of trust
- Specify destination address to tighten security—use loopbacks for management

Normal Traffic

- **! Normal** - Defined as other traffic destined to the router to track and limit

```
ip access-list extended normal-acl
  permit icmp any any ttl-exceeded
  permit icmp any any port-unreachable
  permit icmp any any echo-reply
  permit icmp any any echo
  permit icmp any any packet-too-big
```

Reactive Undesirable Traffic

- **! Reactive Undesirable** - Traffic that should never touch the RP

```
ip access-list extended reactive-undesirable-acl  
    permit udp any any eq snmp
```

- If using a reactive undesirable class, this acl must exist otherwise default permit and all matches

Catch-All Traffic

- ! Catch All - Defined as other IP traffic destined to the router

```
ip access-list extended catch-all-acl
  permit tcp any any
  permit udp any any
  permit icmp any any
  permit ip any any
```

- Using a more granular ACL helps identify what falls into this class in case policy corrections are necessary

CoPP: Sample Class-Map

**! Define a class for each “type” of traffic and associate the
! appropriate ACL**

```
class-map match-all CoPP-known-undesirable
  match access-group name known-undesirable-acl
class-map match-all CoPP-critical
  match access-group name critical-acl
class-map match-all CoPP-important
  match access-group name important-acl
class-map match-any CoPP-normal
  match access-group name normal-acl
class-map match-any CoPP-reactive-undesirable
  match access-group name reactive-undesirable-acl
class-map match-any CoPP-catch-all
  match access-group name catch-all-acl
```


CoPP: Sample Policy-Map

```
policy-map CoPP
```

```
  class CoPP-known-undesirable
```

```
    drop
```

```
  class CoPP-critical
```

```
    <no operation specified – no rate-limit>
```

```
  class CoPP-important
```

```
    police <rate> conform-action transmit exceed-action drop
```

```
  class CoPP-normal
```

```
    police <rate> conform-action transmit exceed-action drop
```

```
  class CoPP-reactive-undesirable
```

```
    drop
```

```
  class CoPP-catch-all
```

```
    police <rate> conform-action transmit exceed-action drop
```

CoPP: Piloting Rate-Limits

```
policy-map CoPP
```

```
  class <rate-limited CoPP class>
```

```
    police <rate> conform-action transmit exceed-action transmit
```

- Use a “transmit/transmit” police to help pilot rate-limits
- We can monitor CoPP and then iteratively adjust rate-limits to a suitable value

Monitoring CoPP

- “show policy-map control-plane” - invaluable for reviewing and tuning site-specific policies and troubleshooting

Displays dynamic information about number of packets (and bytes) conforming or exceeding each policy definition

- “show access-list” displays hit counts on a per ACL entry (ACE) basis

Hits indicate flows for that data type to control plane as expected

- Use SNMP queries to automate process of reviewing service-policy transmit and drop rates

Cisco QoS MIB (CISCO-CLASS-BASED-QOS-MIB) provides primary mechanisms for MQC-based policy monitoring

Show Policy-Map Command

```
Router#show policy-map control-plane input
Control Plane
```

`Service policy input. COPP` — Service Policy Map Name and “Direction”

`Class-map: Classify (match-all)` — Class-Map Name and “Criteria”
16 packets, 2138 bytes — Number of Packets/Bytes Matched

5 minute offered rate 0 bps, drop rate 0 bps — ACL Name/Number

Match: access-group 120

police:

cir 125000 bps, bc 1500 bytes
conformed 16 packets, 2138 bytes; actions:
transmit
exceeded 0 packets, 0 bytes; actions:
transmit
conformed 0 bps, exceed 0 bps

— Police “Action”

— Default Class

`Class-map: class-default (match-any)`

250 packets, 84250 bytes

5 minute offered rate 0 bps, drop rate 0 bps

Match: any

police:

cir 8000 bps, bc 1500 bytes
conformed 41 packets, 5232 bytes; actions:
transmit
exceeded 0 packets, 0 bytes; actions:
drop
conformed 0 bps, exceed 0 bps

— Police “Action”

```
Router#
```

Control Plane Protection (CPPr)



Control Plane Protection (CPPr)

- The Control Plane Protection extends CoPP providing finer policing granularity and additional control-plane protection mechanisms

Further classifies control-plane traffic into different categories

Mechanism to drop packets directed toward non-listening Cisco IOS TCP/UDP ports

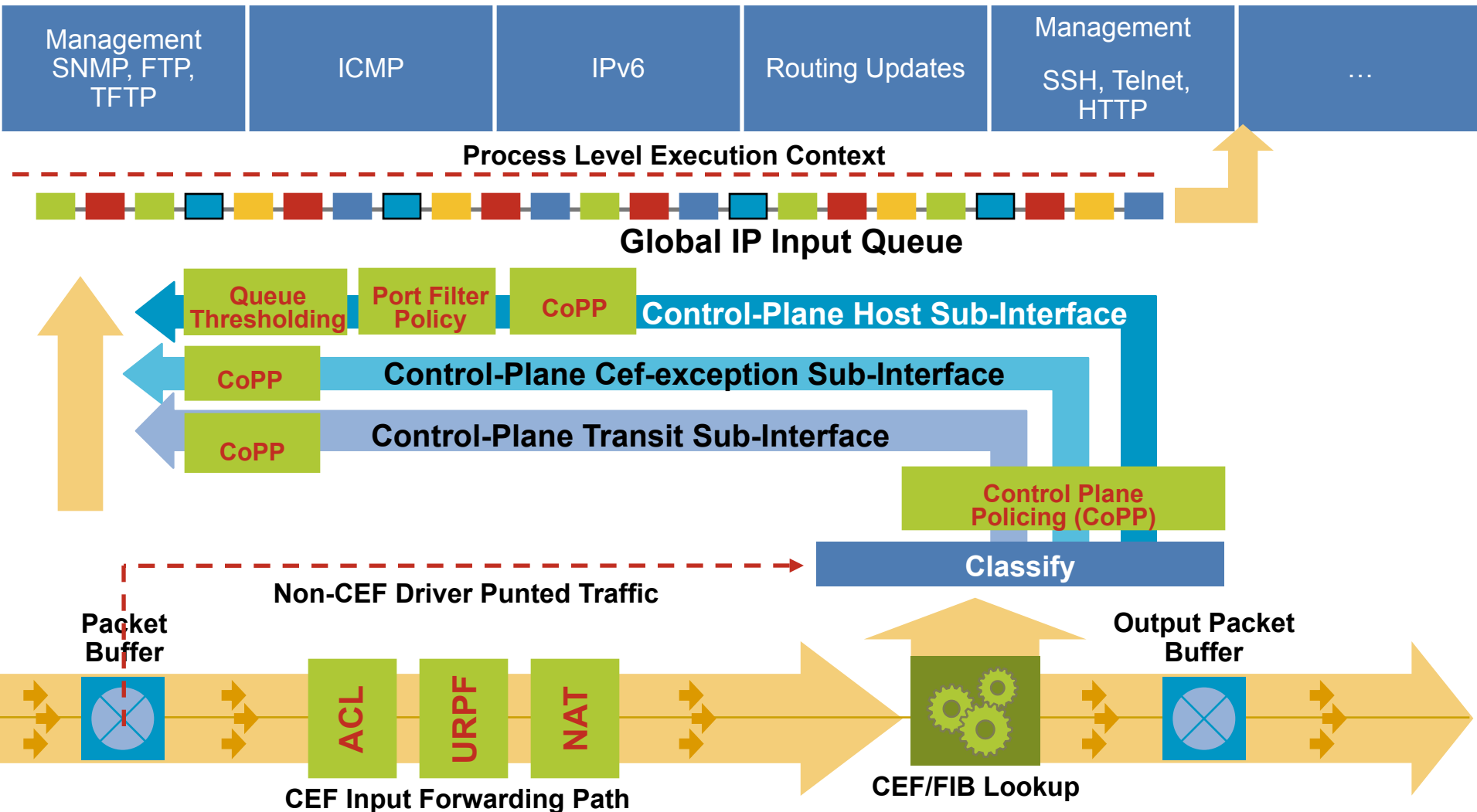
Ability to limit protocol queue usage

- Available in 12.4(4)T

http://www.cisco.com/en/US/products/ps6441/products_feature_guide09186a0080556710.html

Control Plane Interfaces

Control Plane



Control Plane Interfaces

- Each type of traffic is mapped to a corresponding control-plane sub-interface:
 - host sub-interface** - receive adjacency traffic. Includes management and routing traffic
 - cef-exception sub-interface** - packets redirected to the RP before route lookup:
 1. Configured input feature requires additional processing: decryption or other translation
 2. Packets having header information that requires further processing: IP Options or TTL = 0 or 1
 3. Packets redirected by the Interface driver such as ARP, Layer-2 keepalives
 - transit sub-interface** - pass-through traffic sent to control-plane for more processing after route lookup and before forwarding. Includes CBAC and egress ACL Logging
- Control-plane interfaces allow individual policing and protection policies on each
- Existing CoPP policies work with Control-plane interfaces

Control-Plane Interfaces

```
Router#show running-config | begin control-plane
control-plane host
!
control-plane transit
!
control-plane cef-exception
!
!
control-plane
!
!
```

Additional Control Plane Protection Features

New Control-Plane Protection Features Can Be Applied to the Host Sub-Interface

- Port Filtering - packet filtering on a TCP/UDP port basis with manual and auto-detection of configured services
- Queue-Thresholding - controls number of unprocessed protocol packets allowed into the control plane input queues

Port Filtering Feature Overview

- Pre-port filtering

 - Open port information maintained at process-level

 - Each process had to maintain its own port information

 - Packets destined to closed ports must be evaluated at process-level only to be dropped

 - Results in process overhead and potential attack vector

- Port filtering

 - Early drop of packets destined for closed TCP/UDP ports

 - Eliminates processing of traffic discarded anyway

 - Drop packets matching specified TCP/UDP ports

 - Uses MQC/CPL class and policy maps

 - Only attached to control-plane host sub-interface

Control-Plane Open Ports

Port Filter Feature Maintains Dynamic Global Database of Open TCP/UDP Ports Including Ephemeral Ports Created by Applications

```
Router#show control-plane host open-ports
```

```
Active internet connections (servers and established)
```

Prot	Local Address	Foreign Address	Service	State
tcp	*:22	*:0	SSH-Server	LISTEN
tcp	*:23	*:0	Telnet	LISTEN
tcp	*:44095	172.16.2.1:179	BGP	ESTABLIS
tcp	*:80	*:0	HTTP CORE	LISTEN
tcp	*:179	*:0	BGP	LISTEN
tcp	*:443	*:0	HTTP CORE	LISTEN
udp	*:67	*:0	DHCPD Receive	LISTEN
udp	*:123	*:0	NTP	LISTEN
udp	*:161	*:0	IP SNMP	LISTEN
udp	*:162	*:0	IP SNMP	LISTEN
udp	*:56837	*:0	IP SNMP	LISTEN
.
.

Management Plane Security



IP Management Plane Security

- The management plane is the **logical** group containing all management traffic supporting provisioning, maintenance, and monitoring functions for the network
- It is, therefore, critical that management plane resources and protocols are:

Secured to mitigate the threat of unauthorized access and malicious network reconnaissance, which inevitably leads to attacks within the IP data, control, and services planes

Protected to mitigate the risk of DoS attacks

Remain available during attacks such that attack sources can be identified and attacks themselves can be mitigated

Management Interface Types

- **In-band:** a physical (or logical) interface that carries both management and data plane traffic
- **Out-of-band:** a physical interface that connects to a physically separate, isolated network dedicated exclusively to the operation and management of all network elements
- Why OOB?
 - Availability
 - Scalability

Management Interfaces

- In-band

Virtual terminal lines (VTY): have no associated physical interface and are used exclusively for remote terminal access (e.g., Telnet, SSH)

- Out-of-band

Console port: the console port (CTY) is an asynchronous serial port that uses a DCE RJ-45 receptacle for connecting a data terminal (DTE)

Auxiliary port: the auxiliary port (AUX) is also an asynchronous serial port that uses a DTE RJ-45 receptacle for connecting a modem or other DCE device

Management Ethernet port: on certain routers, a separate Ethernet port is made available strictly for OOB management connectivity

Management Interfaces

By default, no passwords are defined!

- **In-band**

Virtual terminal lines (VTY): have no associated physical interface and are used exclusively for remote terminal access (e.g., Telnet, SSH)

- **Out-of-band**

Console port: the console port (CTY) is an asynchronous serial port that uses a DCE RJ-45 receptacle for connecting a data terminal (DTE)

Auxiliary port: the auxiliary port (AUX) is also an asynchronous serial port that uses a DTE RJ-45 receptacle for connecting a modem or other DCE device

Management Ethernet port: on certain routers, a separate Ethernet port is made available strictly for OOB management connectivity

Management Plane Protection (MPP)

- Problem

Devices allow management traffic through any configured interface

Filter incoming management traffic primarily by ACLs on every interface

Very time consuming and difficult to scale

If not done, router vulnerable to DoS attacks

- Solution

Choose which interfaces can accept management traffic

User defined set of protocols—SSH, SNMP, etc. destined to the device

- Benefit

Fewer ACLs

Easier configurations

Available in 12.4(6)T

http://www.cisco.com/en/US/products/ps6441/products_feature_guide09186a0080617022.html

MPP Feature Overview

- Designate interfaces as management interfaces via CLI command
- Device only accepts management traffic on a management interface
- Simplifies using interface ACLs
- Protocol support: SNMP (all versions), HTTP, HTTPS, FTP, TFTP, Telnet, SSH (v1 and v2)
- Interfaces supported: physical, sub-interfaces, tunnel (GRE/VTI)
- Does not impact traffic that is switched/routed through the device

MPP Configuration Example

- Configure MPP so that SSH and SNMP are allowed access **only** through GigabitEthernet 0/3 interface

```
router#config t
router(config)#control-plane host
router(config-cp)#management-interface GigabitEthernet 0/3
allow ssh snmp
router(config-cp)#
.Aug  2 15:25:32.846: %CP-5-FEATURE: Management-Interface
feature enabled on Control plane host path
```

- All other supported management protocols are dropped on interface GigabitEthernet 0/3
- All supported management protocols including SSH and SNMP dropped on all other interfaces

MPP Monitoring Command

Verify MPP Configuration and View Stats:

```
router#show management-interface
```

```
Management interface GigabitEthernet0/3
```

Protocol	Packets processed
ssh	0
snmp	0

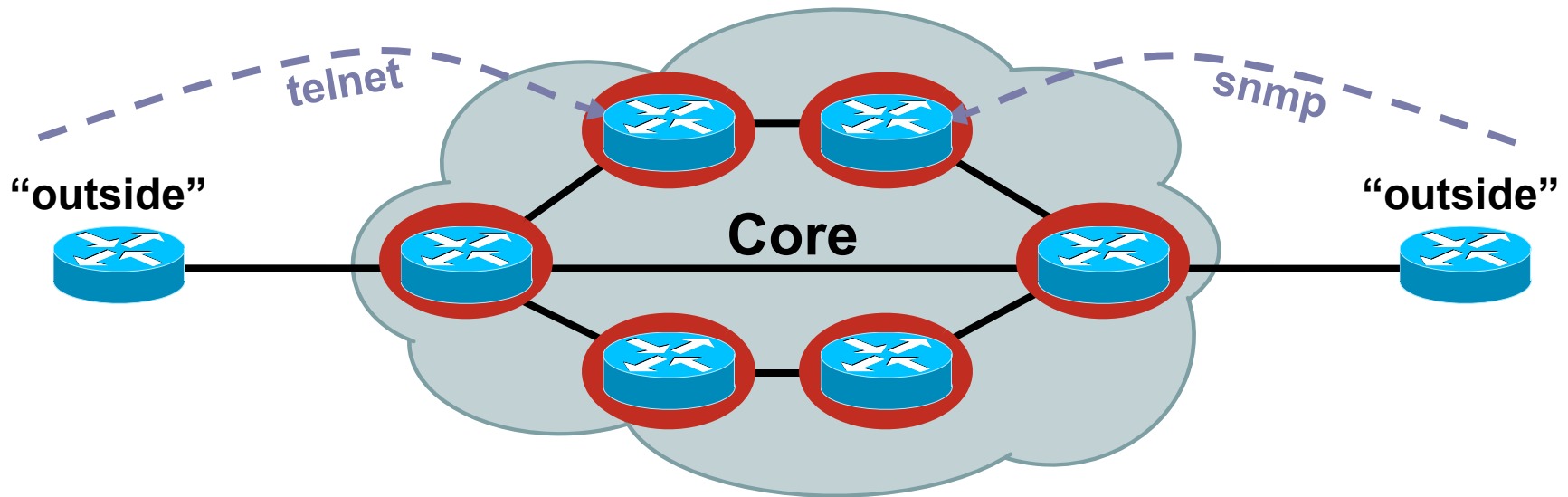
Disable Unused Management Services

- **BOOTP** services
- **HTTP**
- **Finger** service
- **EXEC** mode on unused lines
- **DHCP** server and relay functions
- **CDP**: best practice to disable on external (untrusted) interfaces
- **DNS**-based host name-to-address translation (i.e., no ip domain lookup); alternatively configure name servers explicitly
- **NTP**
- **PAD**
- **Small TCP servers**
- **Small UDP servers**

Infrastructure Security

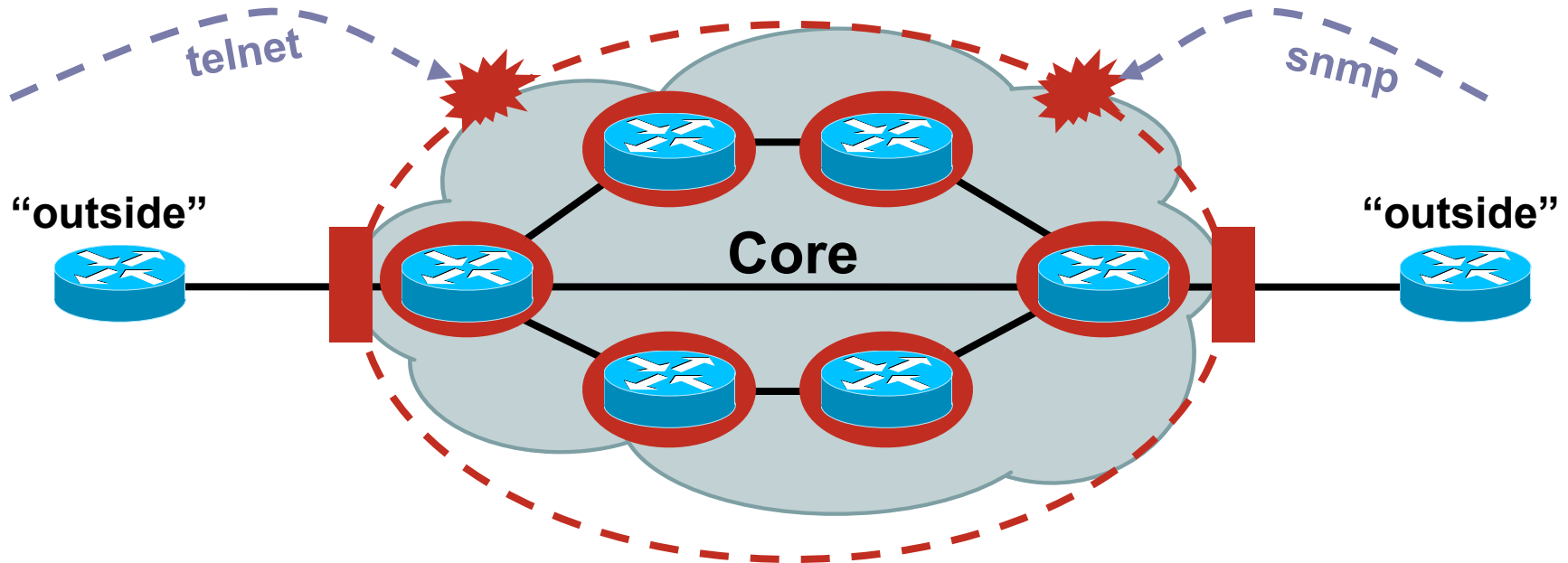


The Old World



- Core routers individually secured
- Every router accessible from outside

The New World



- Core routers individually secured **plus**
- Infrastructure protection
- Routers generally **not** accessible from outside

RFC 2827/BCP 38



RFC 2827/BCP 38 Ingress Packet Filtering

- Packets should be sourced from valid, allocated address space, consistent with the topology and space allocation

Internet Connectivity Guidelines for BCP38

- Networks connecting to the Internet

Must use inbound and outbound packet filters to protect the network

- Configuration example

Outbound—only allow my network source addresses out

Inbound—only allow specific ports to specific destinations in

BCP 38: Consequences of No Action

No BCP 38 Means That:

- Devices can (wittingly or unwittingly) send traffic with spoofed and/or randomly changing source addresses out to the network
- Complicates traceback immensely
- Sending bogus traffic is **not** free

BCP 38 Packet Filtering Principles

- Filter as close to the edge as possible
- Filter as precisely as possible
- Filter both source and destination where possible

Techniques for BCP 38 Filtering

- Static ACLs on the edge of the network
- Dynamic ACLs with AAA profiles
- Unicast RPF strict mode
- IP source guard
- Cable source verify (DHCP)

Using ACLs to Enforce BCP38

- Static ACLs are the traditional method of ensuring that source addresses are not spoofed:

Permit all traffic whose source address equals the allocation block

Deny any other packet

- Principles:

Filter as close to the edge as possible

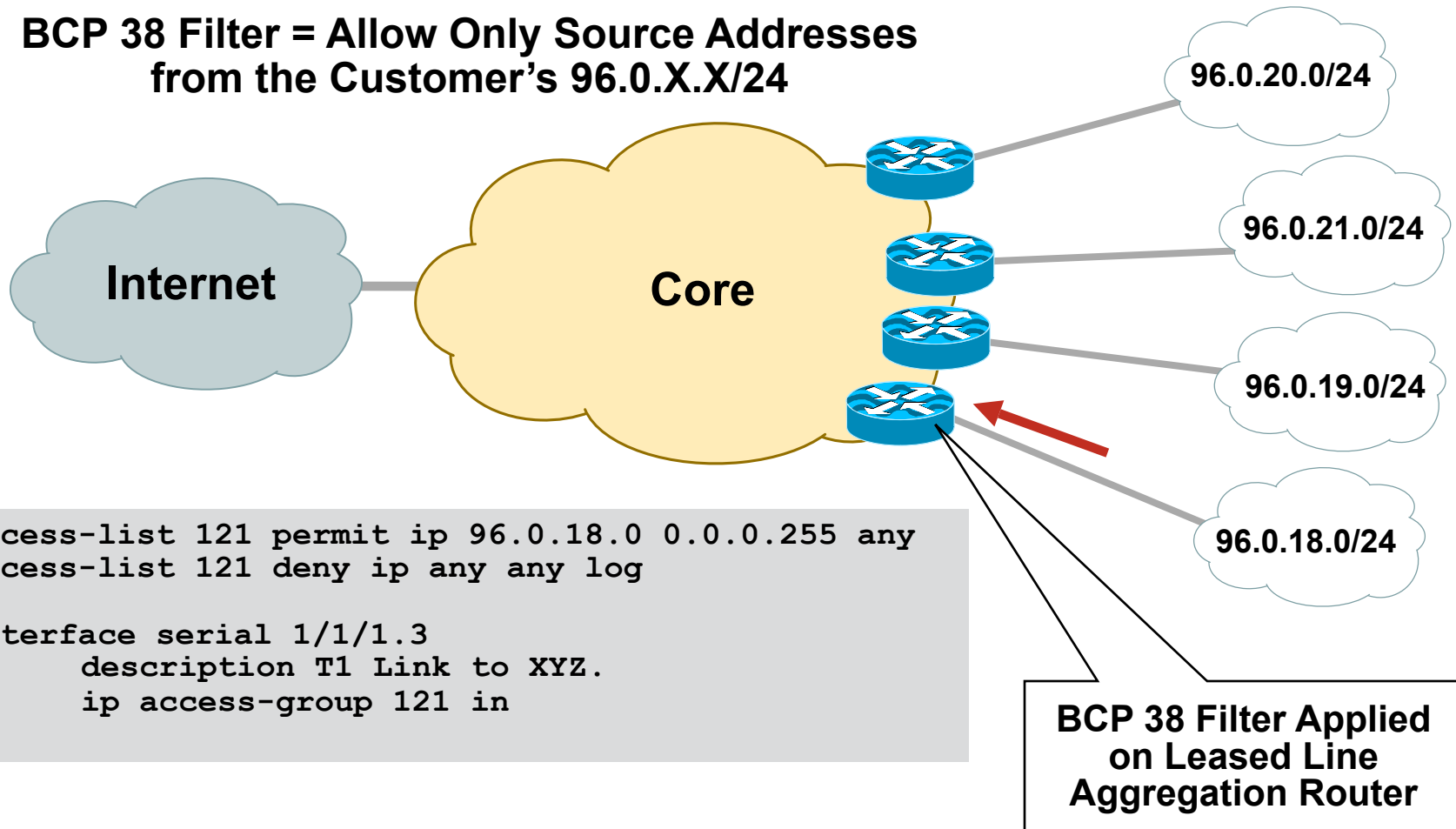
Filter as precisely as possible

Filter both source and destination where possible

Static ACLs for BCP 38 Ingress Packet Filtering

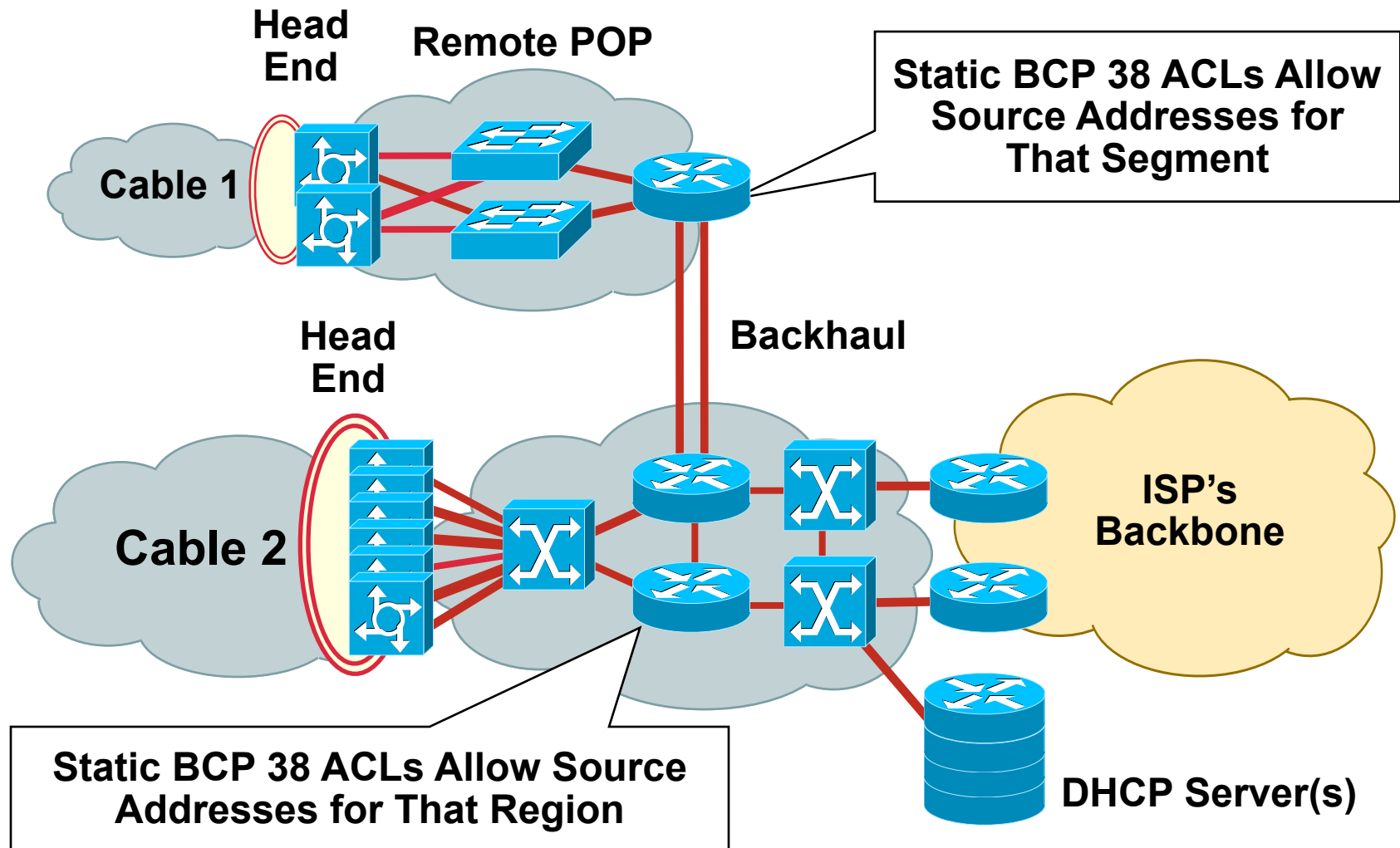
Allocation Block: 96.0.0.0/19

**BCP 38 Filter = Allow Only Source Addresses
from the Customer's 96.0.X.X/24**



ISP

Static BCP 38 ACLs: DHCP



BCP ACL Guidelines

- ISPs

Make sure your customers install filters on their routers - give them a template they can use

- Customer end-sites

Make sure you install strong filters on routers you use to connect to the Internet

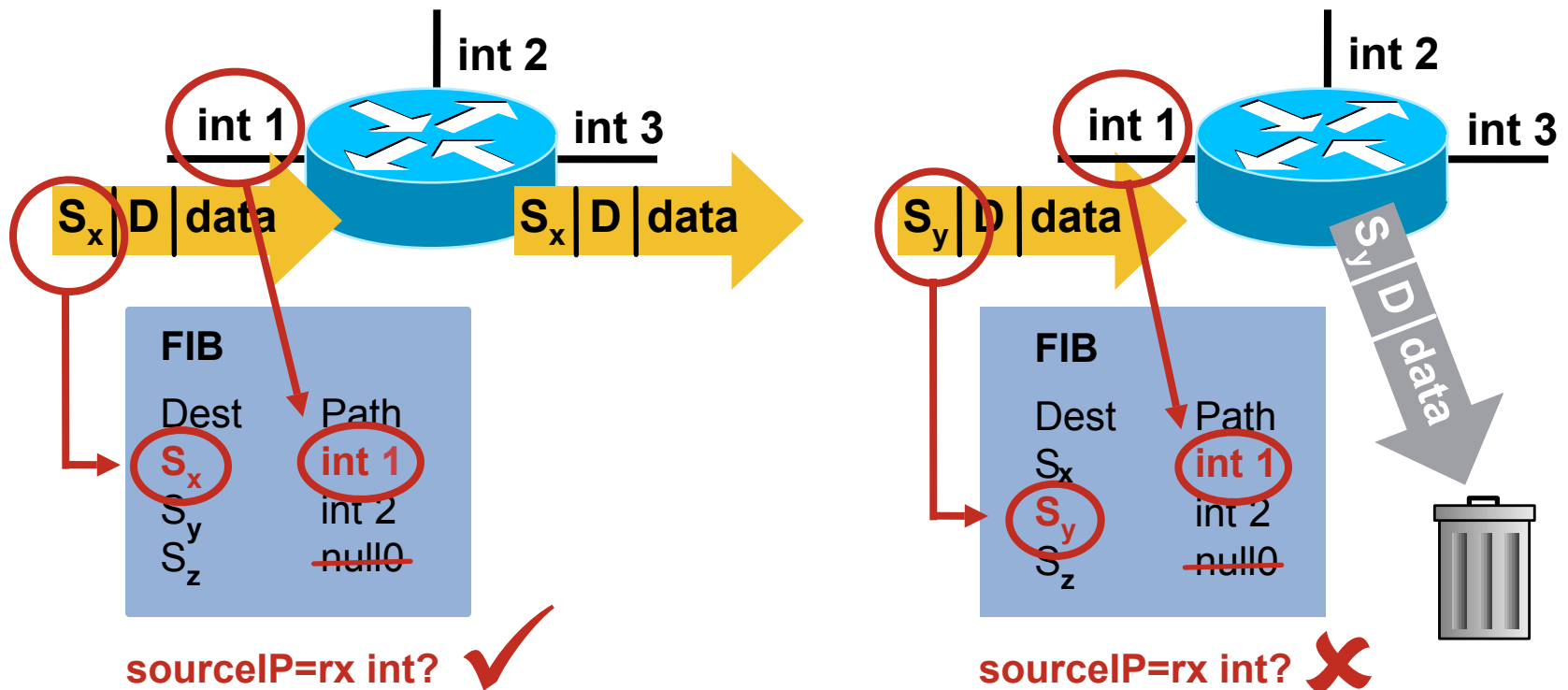
First line of defense - **never** assume your ISP will do it

Unicast Reverse Path Forwarding (uRPF)

- CEF is required
- The purported source of ingress IP packets is checked to ensure that the route back to the source is “valid”
- Two flavors of uRPF:
 - Strict mode uRPF
 - Loose mode uRPF

uRPF—Strict Mode

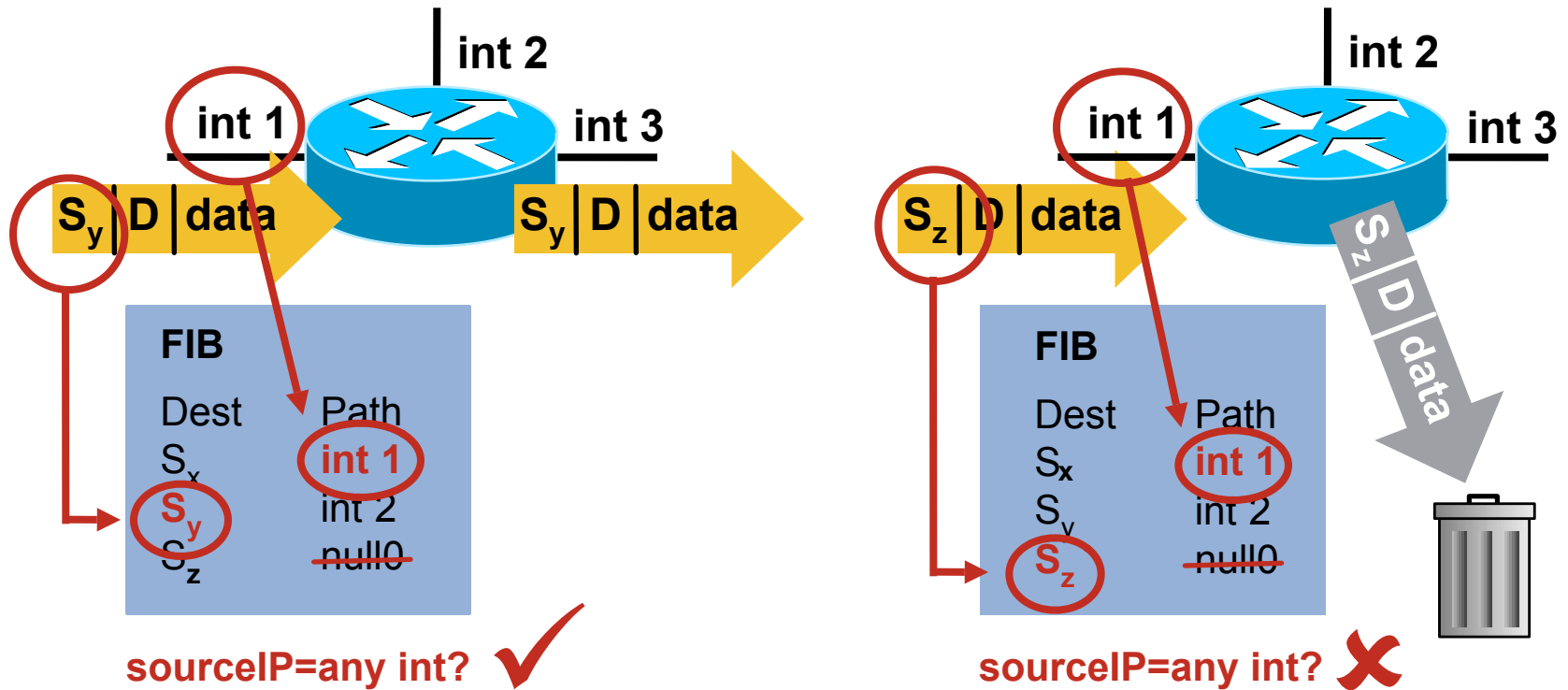
router(config-if)# ip verify unicast source reachable-via rx
(deprecated syntax: ip verify unicast reverse-path)



IP Verify Unicast Source Reachable—Via rx

uRPF—Loose Mode

router(config-if)# ip verify unicast source reachable-via any

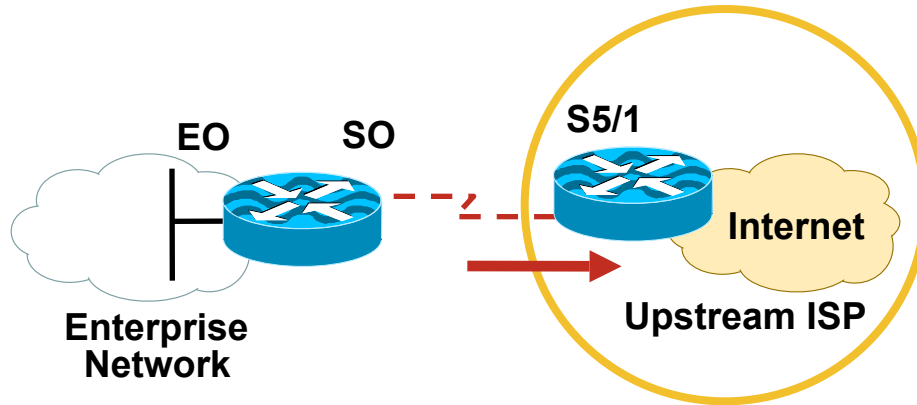


IP Verify Unicast Source Reachable—Via any

Unicast RPF (Strict Mode)

Simple Single Homed Customer Example

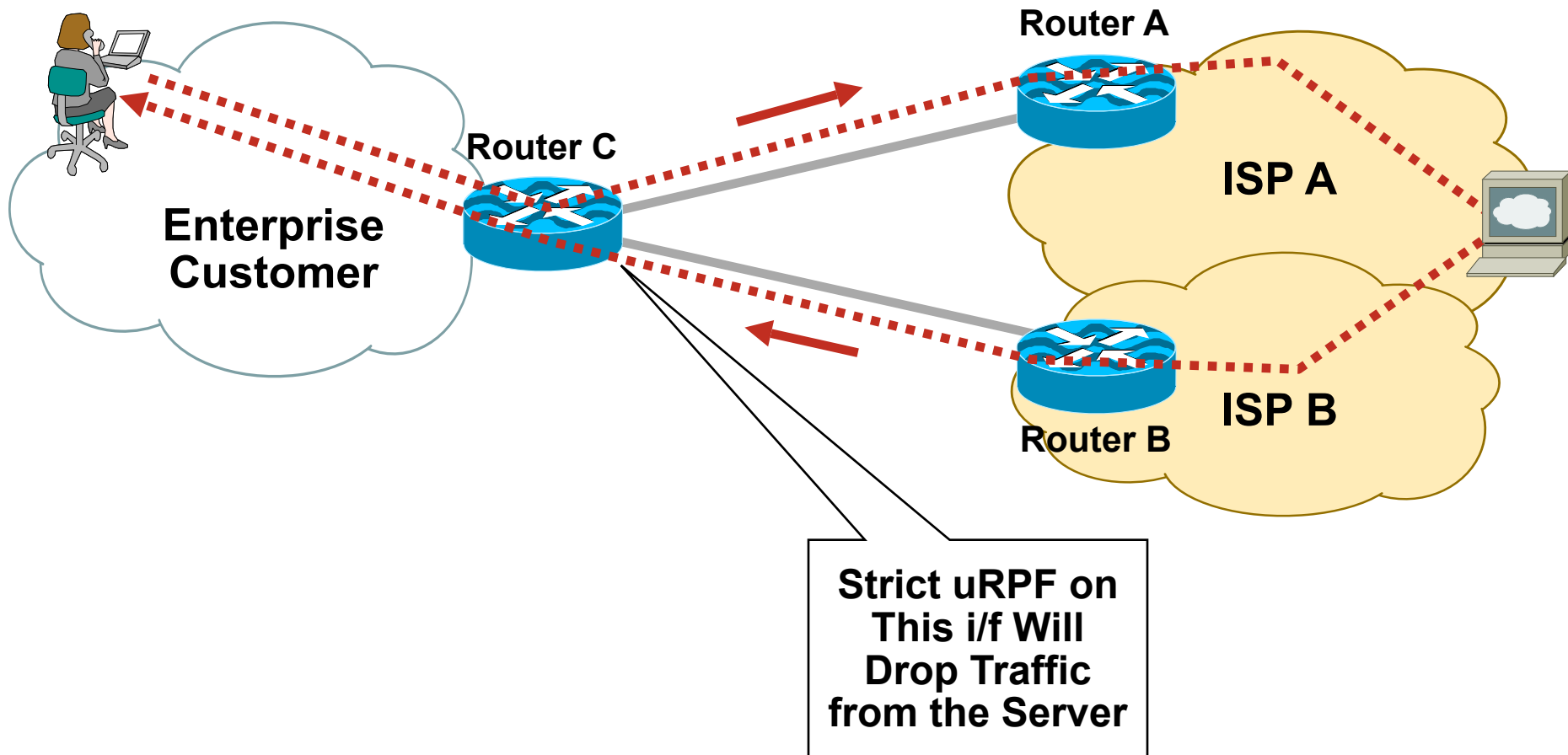
ISP Using uRPF for Ingress Filtering



```
interface Serial 5/1
description 128K HDLC link to Galaxy Publications Ltd [galpub1]
bandwidth 128
ip unnumbered loopback 0
!Unicast RPF activated
ip verify unicast source reachable-via rx
no ip redirects
no ip directed-broadcast
no ip proxy-arp
```

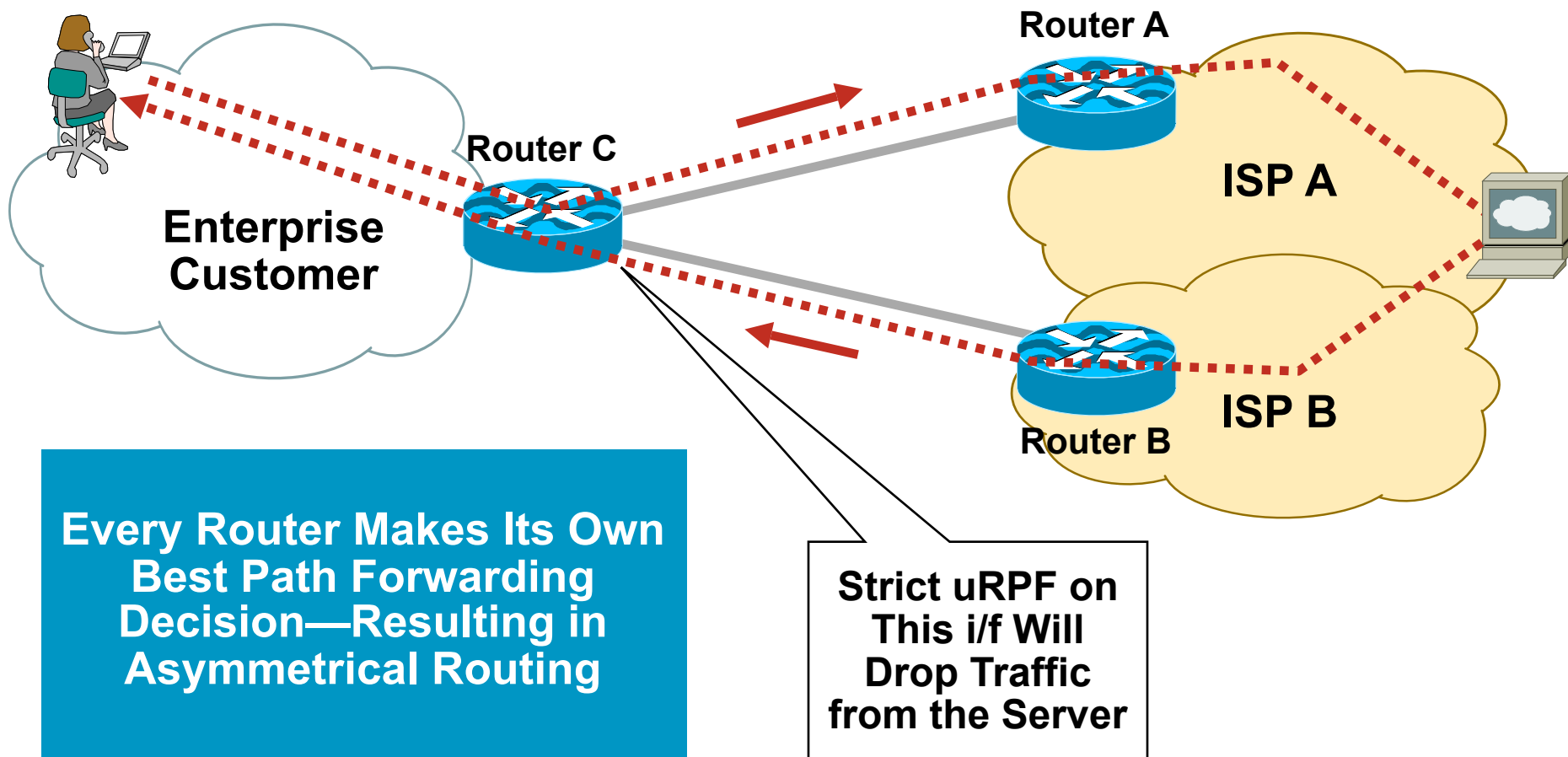
uRPF and Multihomed Customers

What Is Asymmetrical Routing?



uRPF and Multihomed Customers

What Is Asymmetrical Routing?



Strict uRPF and Asymmetric Routing

- Traffic originating from multihomed customers can be verified with uRPF
- Solution: make routing symmetric
- Details in ISP Essentials:

<ftp://ftp-eng.cisco.com/cons/isp/security>
(a must-read for all SP engineers)

- Loose vs. Strict uRPF reference:

Unicast Reverse Path Forwarding Loose Mode

http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_feature_guide09186a00803fa70b.html

Static BCP 38 Filtering in DHCP Networks

- Many broadband cable and DSL networks use DHCP for their CPE client provisioning
- DHCP works per shared segment, hence BCP 38 filters can be applied on the gateway router(s)

Limitation is that people on the same segment can spoof each other

- For Ethernet-based networks we have IP source guard on Cisco Catalysts

IP source verification that is also DHCP aware

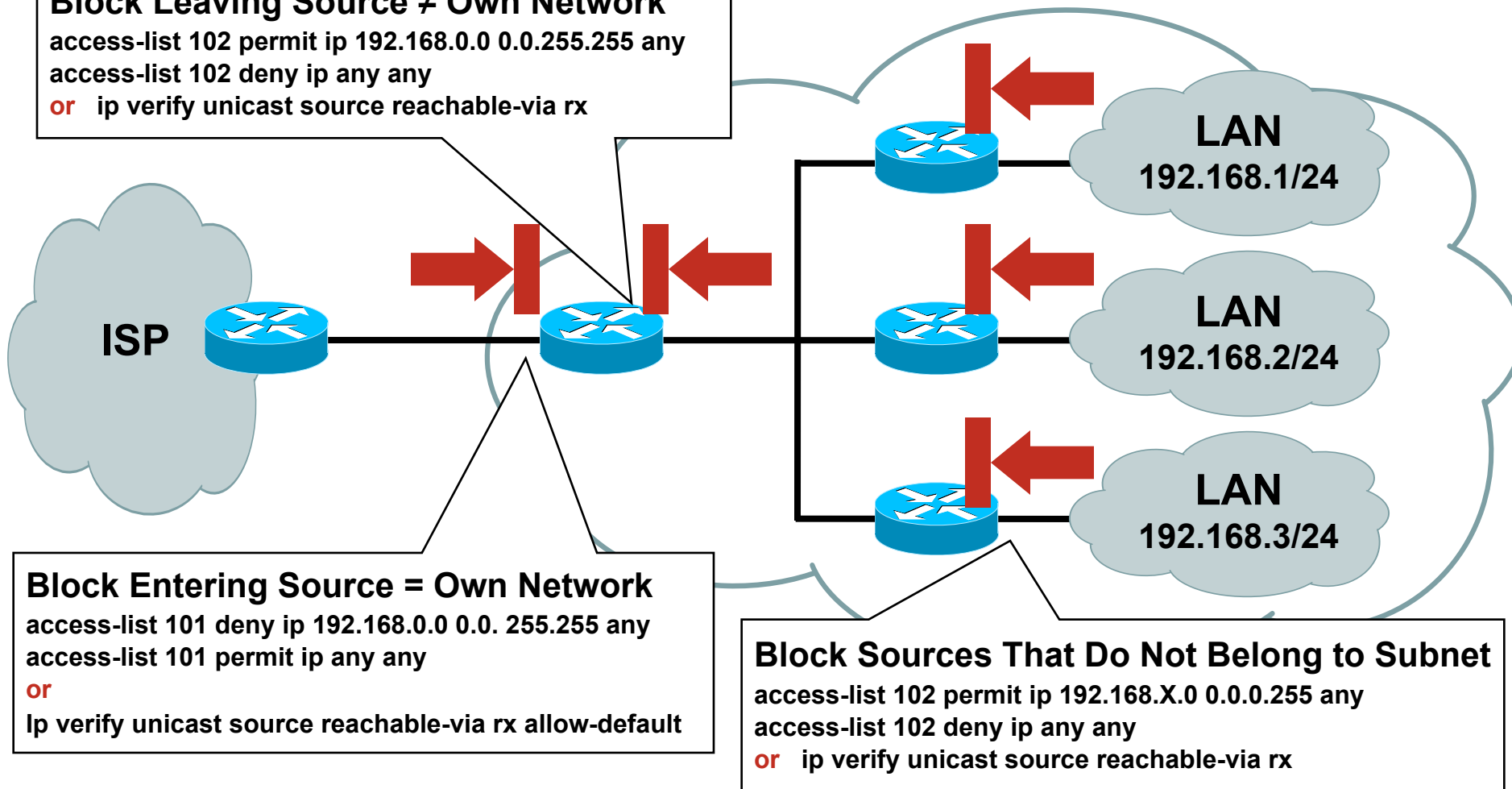
http://www.cisco.com/en/US/products/hw/switches/ps4324/products_configuration_guide_chapter09186a008019d0c8.html

Address Spoofing Prevention in the Enterprise

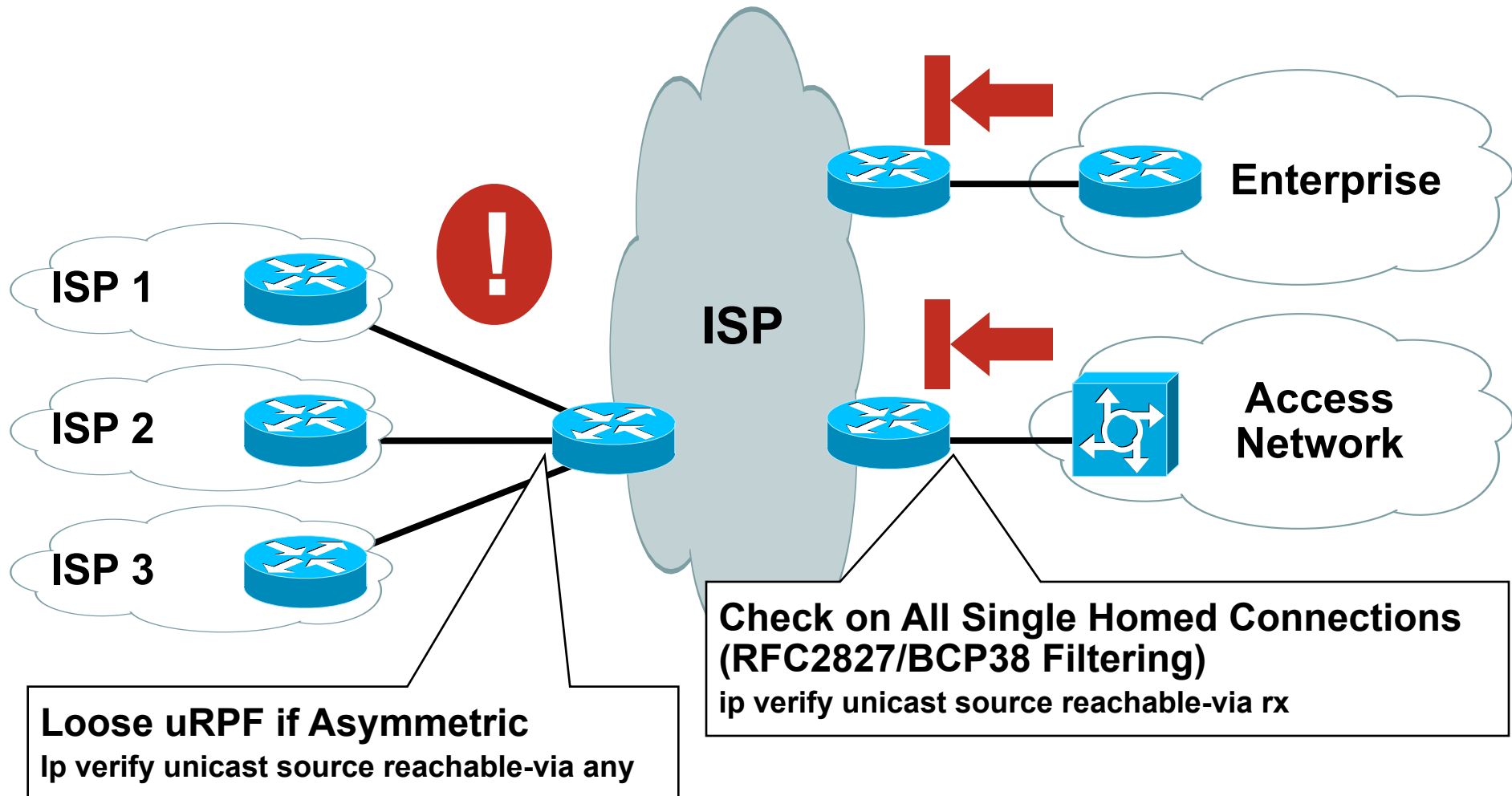
Block Leaving Source \neq Own Network

```
access-list 102 permit ip 192.168.0.0 0.0.255.255 any
access-list 102 deny ip any any
or ip verify unicast source reachable-via rx
```

Enterprise: 192.168.0.0/16



Address Spoofing Prevention on the SP Network



BCP 38 Filtering: Summary

- BCP 38 is an operational reality
 - It works, it is scalable
 - It is operationally deployable and maintainable
 - It works on a wide variety of equipment
 - Deployable in the vast majority of situations—no more excuses
- Take time to understand source address validation techniques, see which ones will work for you
- Find ways to gain operational confidence in the BCP 38 techniques
- BCP 84 lists specific filtering methods

Infrastructure ACLs



Infrastructure ACLs

- Basic premise: filter traffic destined **to** your core routers

Do your core routers really need to process all kinds of garbage?

- Develop list of required protocols that are sourced from outside your AS and access core routers

Example: eBGP peering, GRE, IPSec, etc.

Use classification ACL as required

- Identify core address block(s)

This is the protected address space

Summarization is critical → simpler and shorter ACLs

Infrastructure ACLs

- Infrastructure ACL will permit only required protocols and deny **all** others to infrastructure space
- ACL should also provide anti-spoof filtering
 - Deny your space from external sources
 - Deny RFC1918 space
 - Deny multicast sources addresses (224-239)
 - RFC3330 defines special use IPv4 addressing

Filtering Fragments

- Fragments can be explicitly denied
- Fragment handling is enabled via fragments keyword
- Default permit behavior → permit fragments that match ACE L3 entries
- Denies fragments and classifies fragment by protocol:

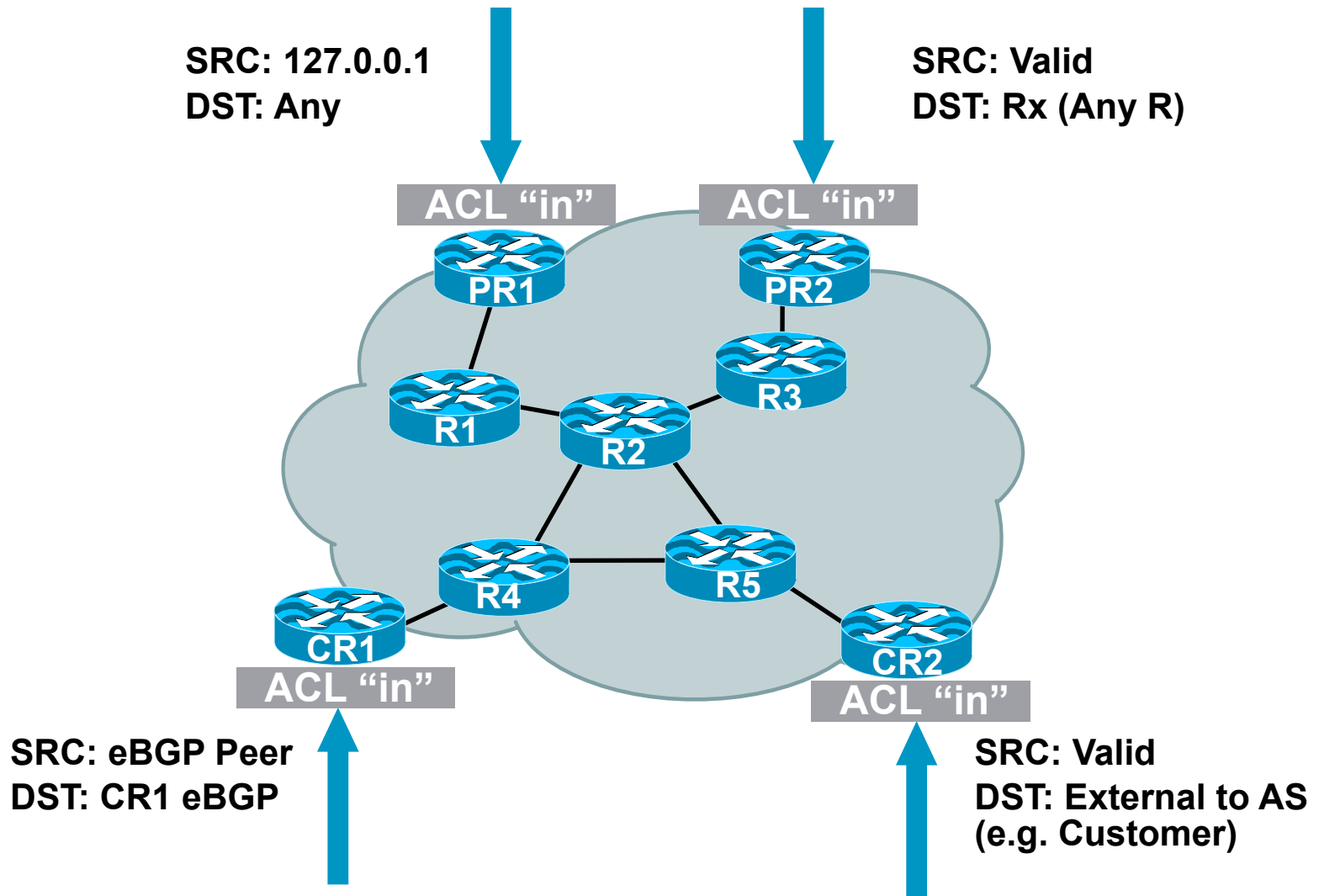
```
access-list 110 deny tcp any any fragments
access-list 110 deny udp any any fragments
access-list 110 deny icmp any any fragments
```

Infrastructure ACLs

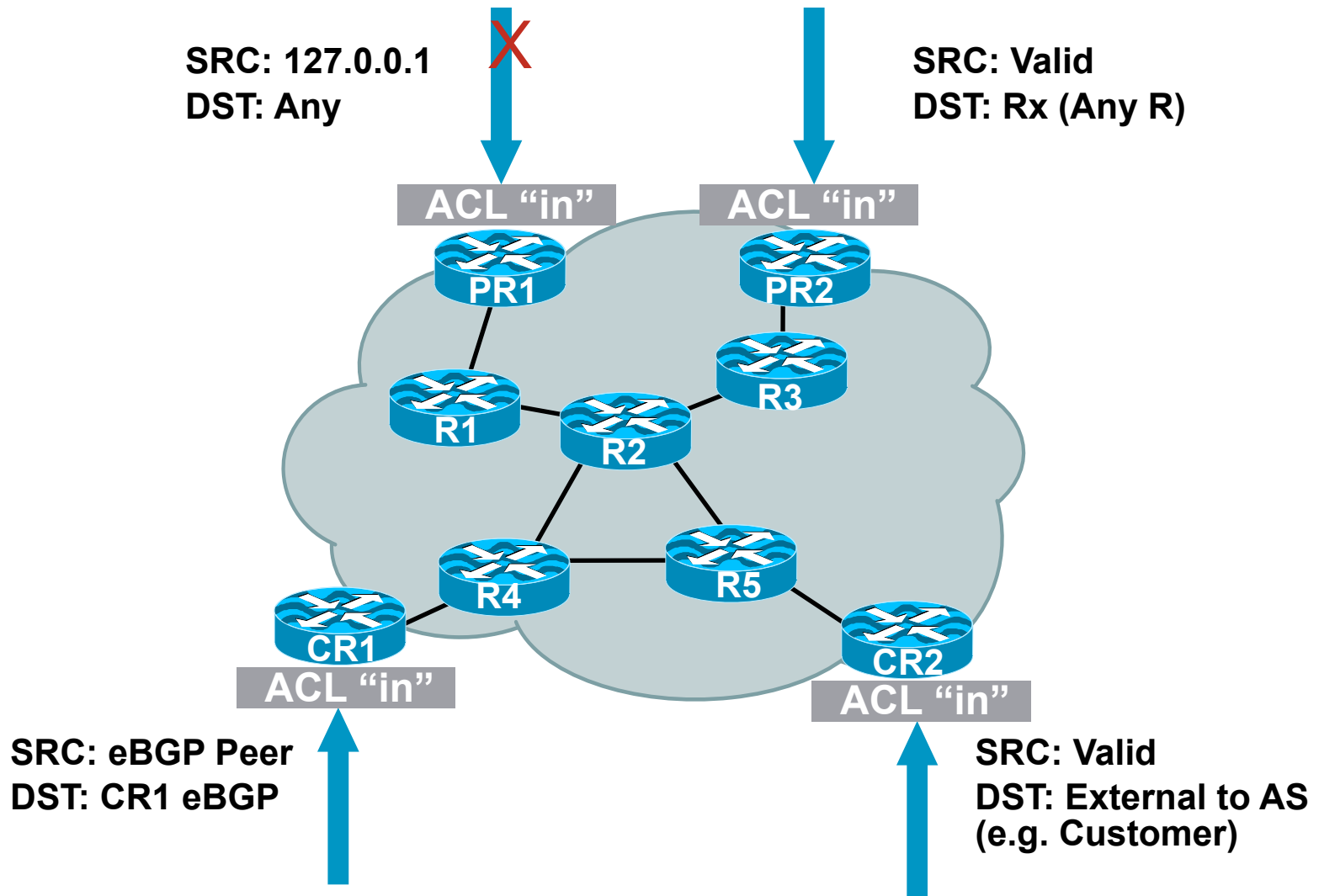
- Infrastructure ACL must permit transit traffic
Traffic passing through routers must be allowed via permit IP any any
- ACL is applied inbound on ingress interfaces
- Fragments destined to the core can be filtered via fragments keyword
- Note: **log** keyword can be used for additional detail; hits to ACL entry with **log** will increase CPU utilization; impact varies by platform; consider:

```
Router(config)#ip access-list logging interval <interval ms>
```

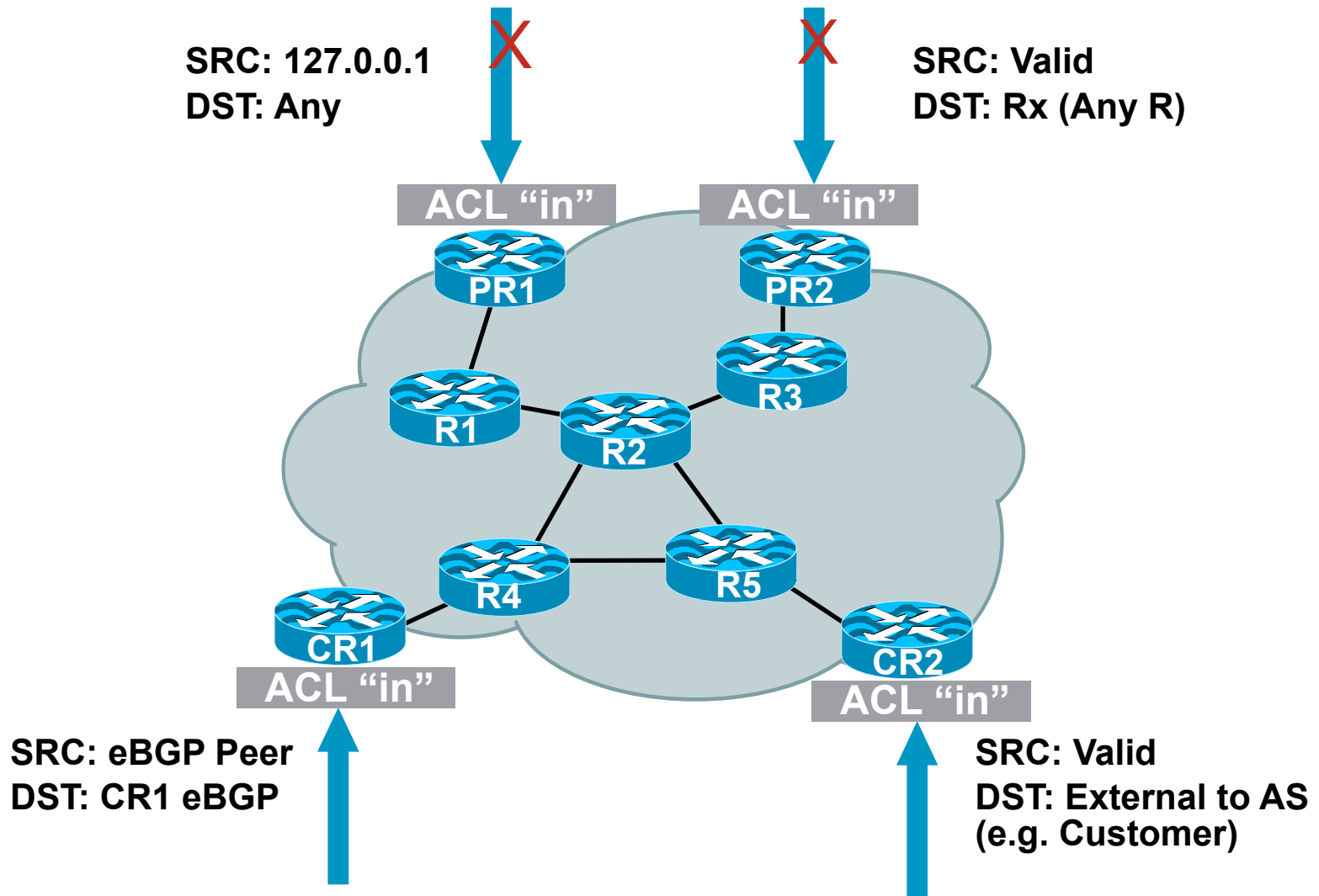
Infrastructure ACL in Action



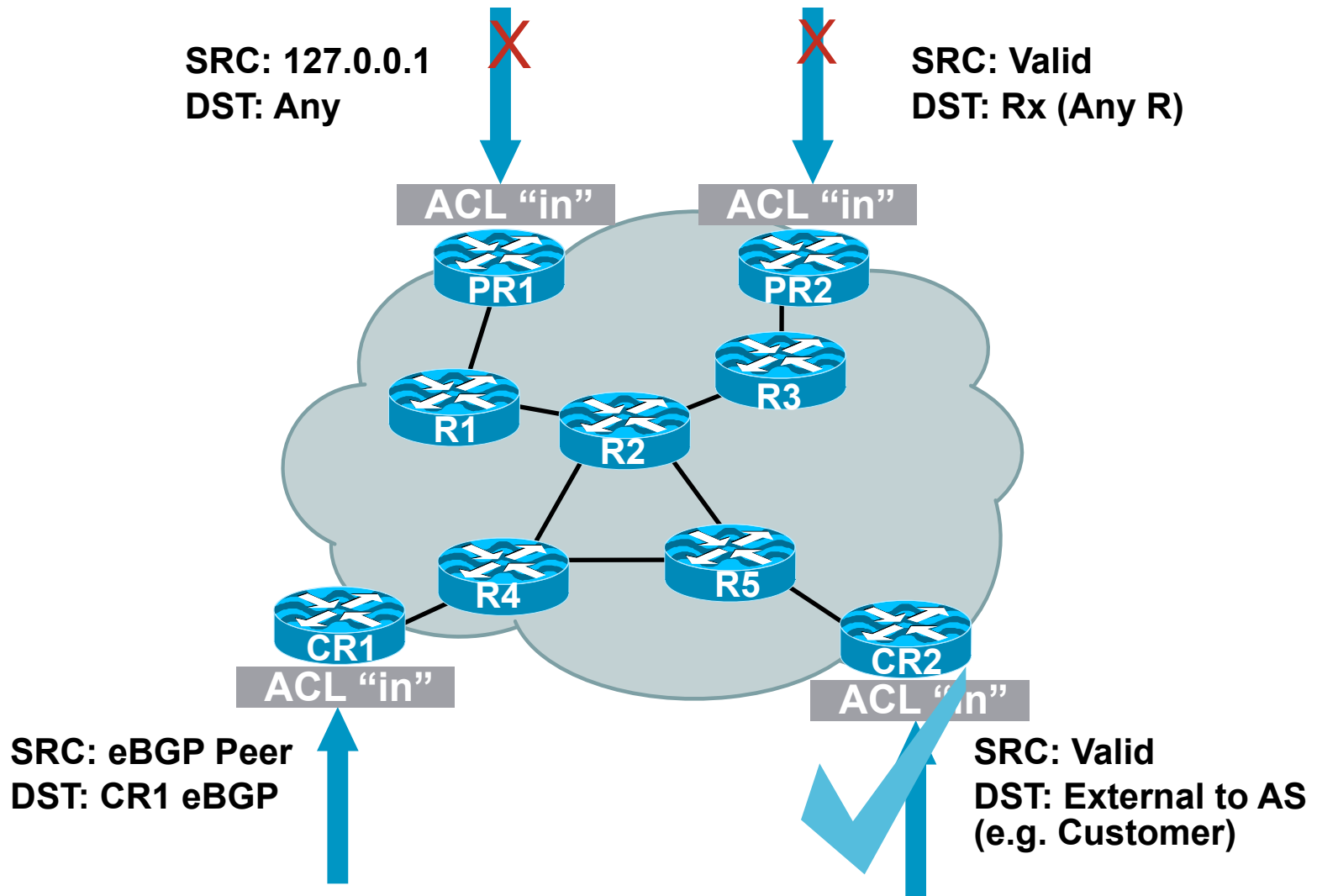
Infrastructure ACL in Action



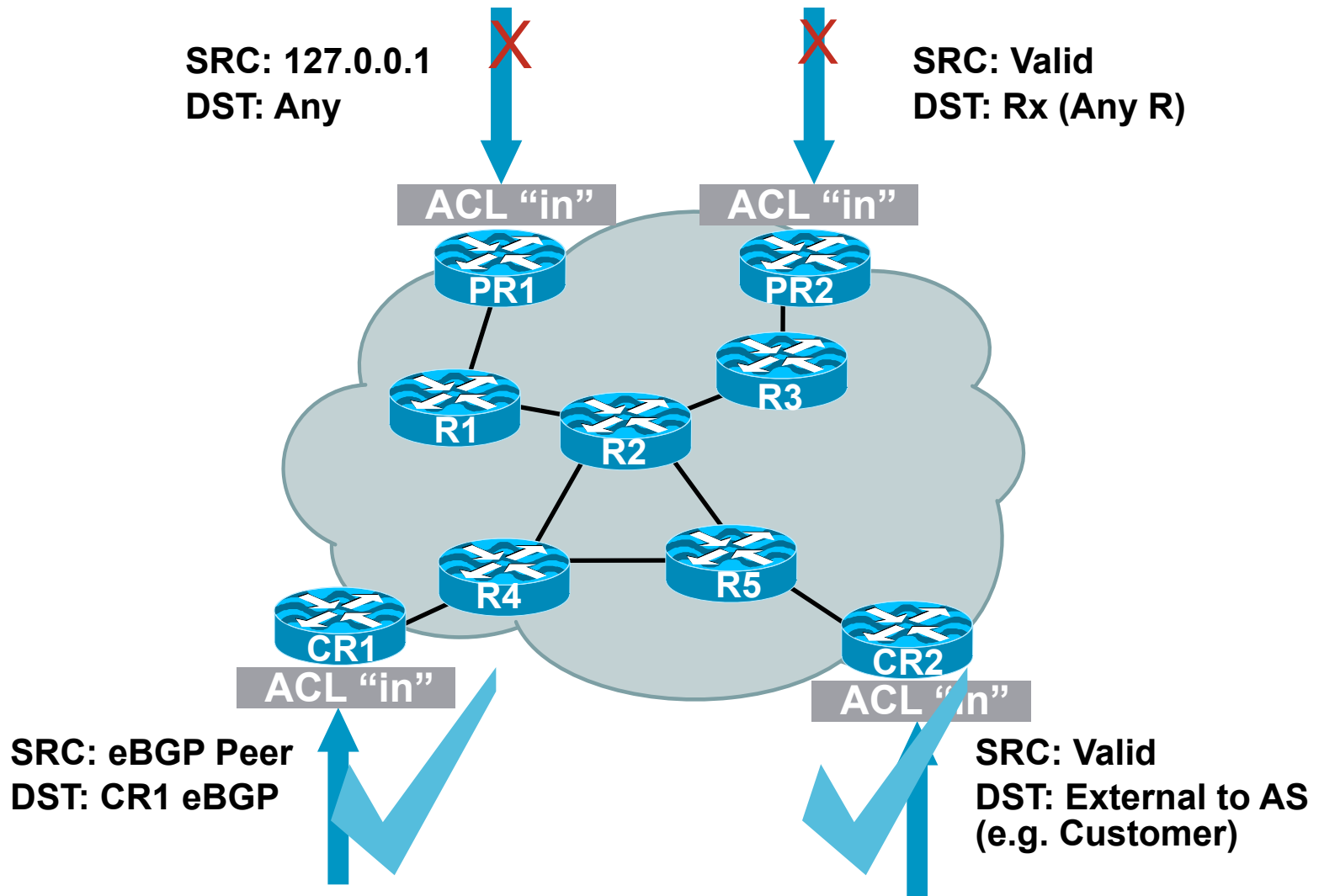
Infrastructure ACL in Action



Infrastructure ACL in Action



Infrastructure ACL in Action



Iterative Deployment: Step 1

- Typically a very limited subset of protocols needs access to infrastructure equipment
- Even fewer are sourced from outside your AS
- Identify required protocols via classification ACL

Step 1: IP Protocols

- TCP - BGP, SSH, SSL
- UDP - SNMP, NTP, DNS
- IGP - OSPF, EIGRP
- GRE and IPv6 Tunneling
- ICMP to/from core routers

ICMP unreachable/TTL expired for traceroute

Do you require other ICMP? (e.g. echo and echo-reply)

Caution: ICMP can be used for DoS

Step 1: IP Protocols

- IPSec (ESP and maybe AH) + IKE
- Others?
- How many of these come from outside and terminate on your infrastructure?

Step 1: Classification ACL

- Classification ACL is used to identify required protocols
- Series of permit statements that provide insight into required protocols
- Initially, many protocols can be permitted, only required ones permitted in next step

Unexpected results should be carefully analyzed → do not permit protocols that you can't explain

Step 1: Classification ACL

- Example:

```
permit tcp any core_CIDR_block  
permit udp any core_CIDR_block  
permit gre any core_CIDR_block  
permit esp any core_CIDR_block  
permit ip any any
```

- Classification ACLs affect data plane traffic

All ACLs have **implicit deny**

Classification ACL must have permit any any to allow normal traffic to flow

Step 1: Classification ACL

- Use show access-list command to view ACE hit counts

- Example:

```
permit tcp any 10.86.183.0 0.0.0.255 (8 matches)
```

- Protocols that display hits should be reviewed to ensure that they are indeed required

Unexpected results should be analyzed

Needed protocols must be explicitly permitted

- Log keyword can be used as well to help provide details

Step 2: Begin to Filter

- Permit protocols identified in Step 1 to infrastructure only address blocks
- Deny all other to addresses blocks
 - Watch ACE counters
 - Log keyword can help identify protocols that have been denied but are needed
- Last line: **permit ip any any** ← permit transit traffic
- The ACL now provides basic protection and can be used to ensure that the correct suite of protocols has been permitted

Steps 3 and 4: Restrict Source Addresses

- ACL is providing basic protection
- Required protocols permitted, all other denied
- Identify source addresses and permit only those sources for requires protocols
 - e.g. external BGP peers, tunnel end-points
- Increase security: deploy destination address filters if possible

Example: Infrastructure ACL

! Deny our internal space as a source of external packets

```
access-list 101 deny ip our_CIDR_block any
```

! Deny src addresses of 0.0.0.0 and 127/8

```
access-list 101 deny ip host 0.0.0.0 any
```

```
access-list 101 deny ip 127.0.0.0 0.255.255.255 any
```

! Deny RFC1918 space from entering AS

```
access-list 101 deny ip 10.0.0.0 0.255.255.255 any
```

```
access-list 101 deny ip 172.16.0.0 0.15.255.255 any
```

```
access-list 101 deny ip 192.168.0.0 0.0.255.255 any
```

Example: Infrastructure ACL

! The only protocol that require infrastructure access is eBGP

! We have defined both src and dst addresses

```
access-list 101 permit tcp host peerA host peerB eq 179
```

```
access-list 101 permit tcp host peerA eq 179 host peerB
```

! Deny all other access to infrastructure

```
access-list 101 deny ip any core_CIDR_block
```

! Permit all data plane traffic

```
access-list 101 permit ip any any
```

Infrastructure ACL: Blocking IP Options

- Provide control functions that may be required in some situations but unnecessary for most common IP communications
- Include provisions for time stamps, security, and special routing
- The option field is variable in length. There may be zero or more options
- Complete list and description of IP Options in RFC 791
- Options can be set when using extended ping

```
Router#ping
```

```
Protocol [ip]: ip
```

```
Target IP address: 10.1.1.1
```

```
...
```

```
Extended commands [n]: y
```

```
...
```

```
Loose, Strict, Record, Timestamp, Verbose[none]:
```

Infrastructure ACL: Blocking IP Options

- ip access-list extended drop-ip-option—slower than drop/ignore

deny ip any any option any-options

permit ip any any

or

- ip options drop—preferred over access-list

Consider deploying at enterprise edge

Available in 12.0(23)S, 12.3(19), 12.3(4)T and 12.2(25)S

or

- ip options ignore—router ignores options

ISP best practice when router doesn't need to process options

“ignore” not available on all routing platforms

Available in 12.0(23)S for GSR

- ip options drop and ignore reference:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1829/products_feature_guide09186a00801d4a94.html

Block All IP Options: Considerations

- IP options not switched in hardware
- Require control plane software processing
 - Process the options
 - Rewrite IP header
- Drop and ignore reduce load on RP - switched in hardware
- Malformed IP options
 - Processed before interface access-lists
 - IP option access-lists don't apply
 - Dropped by receiving router
 - Router generates ICMP Type 12 messages (Self DoS)

Block All IP Options: Considerations

- Can be configured as part of CoPP default class
 - CoPP protects against malformed IP options in 12.2(32.8)S, 12.4(7) and 12.4(6)T
- Some legitimate protocols use options
 - RSVP (NetMeeting)
 - MPLS TE
 - MPLS OAM
 - IGMPv2
 - IGMPv3
 - DVMRP
 - PGM

ACL Support for Filtering on TTL Value

- Filter packets based on IP header time-to-live
- Interface TTL ACLs
 - TTL 0 or 1 denies are process switched
 - Use CoPP to mitigate TTL 0 or 1 drops
 - Other TTLs dropped in fastest switching path
- Denies prevent ICMP time-exceeded messages
- Example:

Extended IP access list ttl-drops

10 deny ip any any ttl range 0 1

20 permit ip any any

- Available in 12.4(2)T

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124newft/124t/124t2/htaclttl.htm>

Infrastructure ACL: TCP Flags Filtering

- ACL to block Nmap scanning

```
ip access-list extended block-nmap
  remark block stealth fin scan
  deny tcp any any match-all -ack +fin -psh -rst -syn -urg log
  remark block xmas scan
  deny tcp any any match-all +fin +psh +urg log
  remark allow syn or ack which should block null scan
  permit tcp any any match-any +ack +syn
  deny tcp any any log
  permit ip any any
```

- Available in 12.3(4)T and 12.2(25)S

http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_chapter09186a0080431049.html

Flexible Packet Matching (FPM)



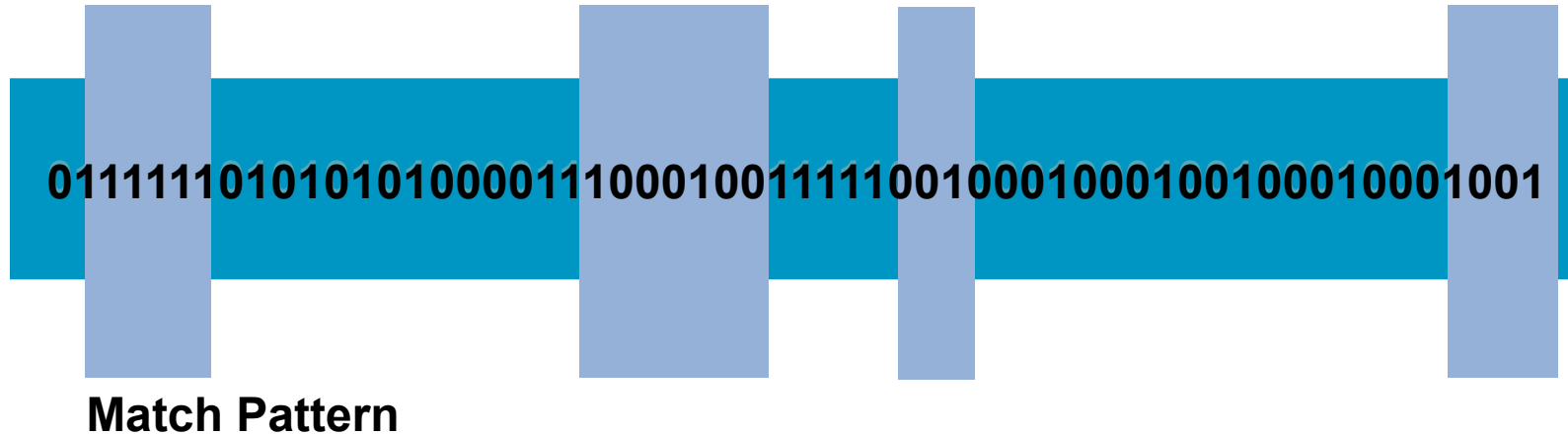
FPM

- Specify match criteria against any part of a packet header and payload
- Allows a choice of corresponding actions
- Create own traffic classifiers either on and off network device
- Deploy classifiers and associated actions
- Router doesn't require reload to apply or remove a classifier and action
- Available in 12.4(4)T

http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124newft/124t/124t4/ht_fpm.htm

FPM

Workers at Layers 2 to 7—Bit/Byte Matching Capability at Any Offset Within the Packet



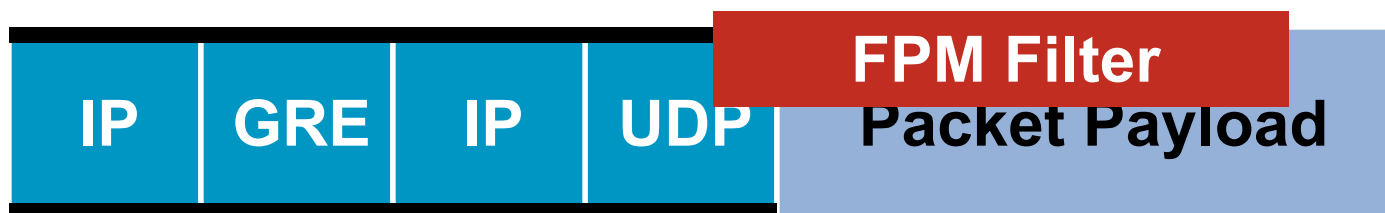
FPM—Concept

- Protocol unaware matching engine
- Protocol template/stencil support
 - Protocol Header Definition File (PHDF)—XML Protocol Header Definition File defines match criteria fields and implicit (constraints) match criteria
 - Breakaway from hard coded CLI match criteria in Cisco IOS—Dynamic CLIs
- Complex protocol structure support
 - Stack class-map types constructs define hierarchical protocol relationships
- Portability of configuration
 - TCDF—XML Traffic Classification Definition File defines FPM filters
- Applied as an access-control service-policy to individual interfaces

FPM—Stack Flexibility

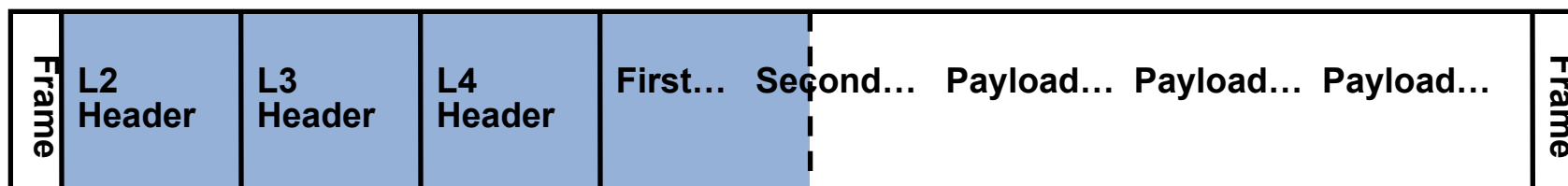


The Same FPM Filter Can Be Applied to Multiple Protocol Stacks



FPM—Phase 1: 12.4(4)T

Capabilities



Can Match on “First” But Not “Second”

- Static offset
 - L2 or L3 header start + numeric (< 256) value
- 32 filters with maximum of eight match criteria each
- Initially designed to require acceleration hardware for performance with greater than 32 filters

FPM—Phase 1

Classification

- Supports Operation Operators
 - eq or neq with mask
 - gt, lt
 - range
- Support logical **and** and **or** operations
- Supports searches using regular expressions

FPM—Phase 1

Actions

- Count
- Drop
- Log
- Send-Response—ICMP-unreachable
- Service-Policy—nested policy

FPM—Phase 1

Delivery Mechanism

- Supported on Access platforms via Advanced Security images
- PHDFs available via Cisco.com
 - Ethernet, IP, TCP, UDP and ICMP
- Supported under Control-Plane Aggregate path

Phase 1.5: 12.4(6)T

Features

- TCDF load and unload support
- Tunneled protocol layering support
- String search capability
- TCDF templates

FPM—Class-Based Policy Language (CPL)

- class-map type stack match-all ip_udp_class
- description "match UDP over IP packets"
- match field ip protocol eq 17 next udp

Stack Class
Defines ip-udp Stack

- class-map type access-control match-all slammer_class
- description "match on slammer packets"
- match field udp dest-port eq 1434
- match field ip length eq 404
- match start udp payload-start offset 196 size 4 eq 0x4011010

Access-Control Class
Defines Match Pattern

- policy-map type access-control fpm_udp_policy
- description "policy for UDP based attacks"
- class slammer_class
- drop
- log

Access-Control Policy-Map
Defines Action

- policy-map type access-control fpm_policy
- description "drop worms and malicious attacks"
- class ip_udp_class

Nested Policy
First Match ip-udp
Then Slammer

FPM—Monitoring

- Show all or designated FPM class maps
rtr# **show class-map type** [stack | access-control] [<name>]
- Show all or designated FPM policy maps
rtr# **show policy-map type access-control** [<name>]
- Show FPM policy maps on designated interface. Also show number of packets matched
rtr# **show policy-map type access-control interface** <interface>
or
rtr# **show policy-map type access-control control-plane** <>
- Show runtime classification information for loaded FPM classes and policies
rtr# **show protocols phdf** <loaded-protocol>
- Show listing of user-defined PHDFs stored locally on router
rtr# **dir disk0:*.phdf**
- Track all FPM events in both control plane and data plane
rtr# **debug fpm event**

Infrastructure Protection

- Understand the risk
- Understand the product benefits and limitations
- Deploying infrastructure protection can be difficult

Start working on designs now

Remember the Boy Scout motto: “Be prepared”

Network Telemetry



SNMP, RMON and Their ilk



Types of Network Telemetry

- SNMP
- NetFlow
- RMON
- BGP
- Syslog
- Packet capture
- Others

SNMP

- SNMP = Simple Network Management Protocol
- Canonical method of obtaining real-time information from network devices
- SNMPv3 provides authentication, encryption
- MIBs support polling of statistics ranging from interface bandwidth to CPU utilization to chassis temperature, etc.
- Both a “pull” model for statistical polling and a “push” model for trap generation based upon events such as link up/down
- Many open-source and commercial collection systems, visualization tools
- Easiest way to get into profiling of general network characteristics

SNMP: Net-Snmp Toolset

- Formerly known as UCD-SNMP toolset
- Open source SNMP command-line tools, library, trap-generator, agent, etc. available from <http://www.net-snmp.org/>
- Included with most Linux distros, FreeBSD, etc.
- Command-line access to SNMP data from enabled routers, switches, etc.
- Runs on Linux, FreeBSD, Mac OS/X, Solaris, other *NIX, Windows
- Perl modules available via CPAN

SNMP: MRTG

- MRTG—the Multi Router Traffic Grapher
- Open source SNMP visualization toolset developed by Tobi Oetiker, available from <http://oss.oetiker.ch/mrtg/>
- Long track-record—(in general use since 1995)
- Can be used to graph router/switch data, host performance information from systems running SNMP agents, etc. (generates HTML w/PNG images)
- Runs on Linux, FreeBSD, Mac OS/X, Solaris, other *NIX, Windows
- Written in Perl, has its own SNMP implementation

Example: MRTG Graphs

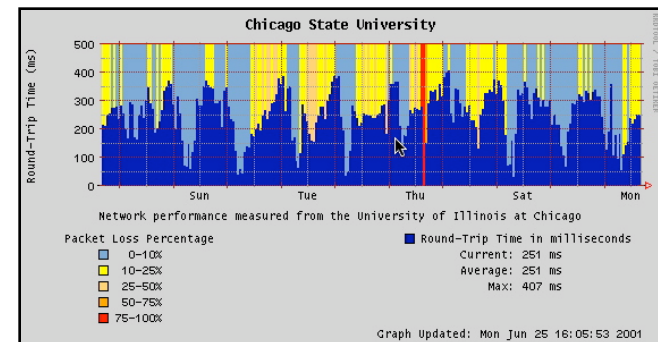
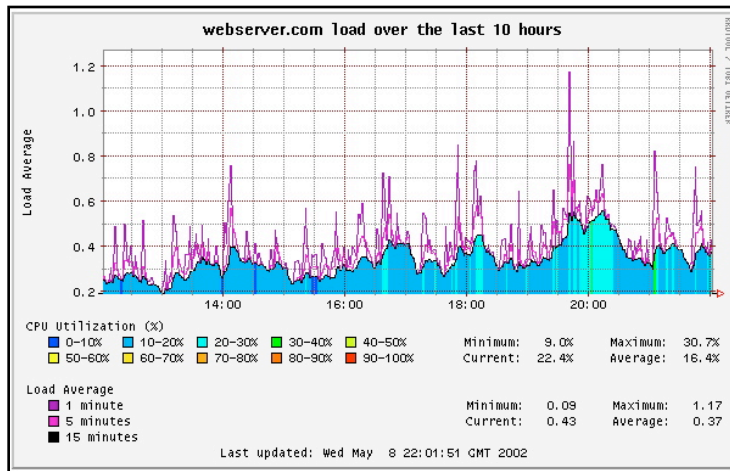
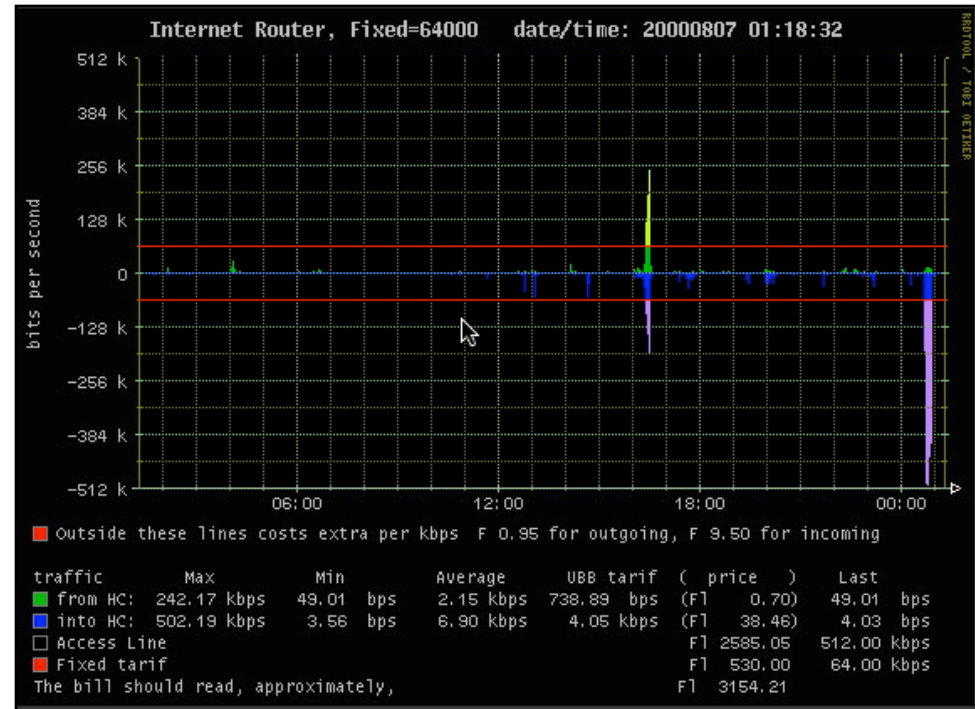
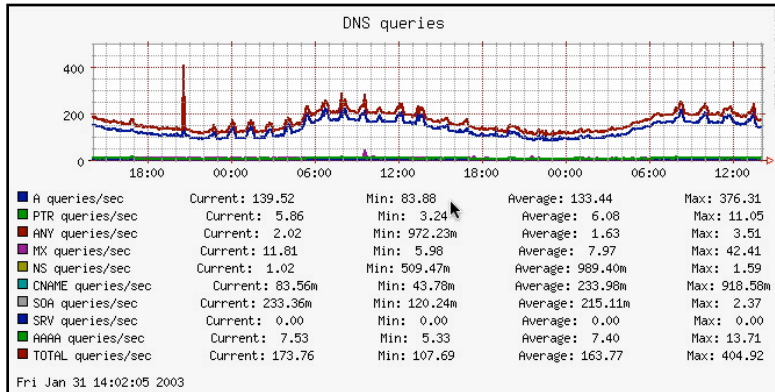


Source: mrtg.org

SNMP: RRDTool

- RRDTool—the Round Robin Database Tool
- Another open source SNMP visualization toolset developed by Tobi Oetiker, available from <http://oss.oetiker.ch/rrdtool/>
- Improved graphing performance, new types of graphs
- Can be used in conjunction with MRTG—does not do its own SNMP collection (can also be used w/NetFlow via OSU flow-tools and FlowScan)
- Runs on Linux, FreeBSD, Mac OS/X, Solaris, other *NIX, Windows
- Many nice HTML/PHP front-ends such as Cacti, Cricket, Big Sister, etc.

Example: RRDTool Graphs

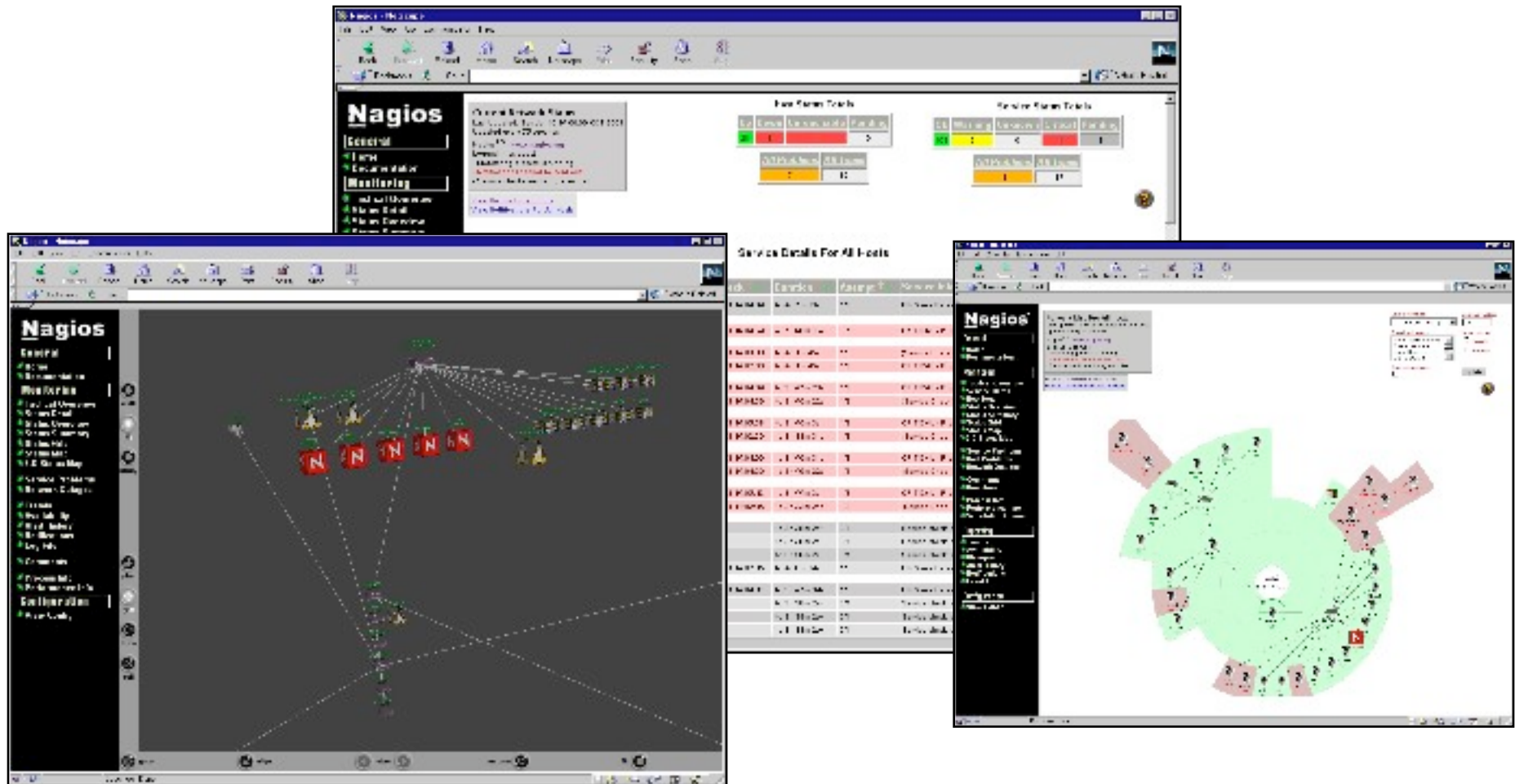


Source: <http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/>

SNMP: NMS

- Network Management Systems (NMS) can serve as SNMP consoles, among other things
- Many can use SNMP traps and/or other forms of telemetry as triggers for paging, scripted actions, etc.
- Pulling information together can be useful for NOCs, operations teams
- Commercial systems such as HP OpenView, Micromuse NetCool, IBM Tivoli, CA Unicenter
- Several open source systems—Big Brother (<http://bb4.com/>), Big Sister (<http://bigsisiter.graeff.com/>), Nagios (<http://www.nagios.org/>), and others

Nagios Examples

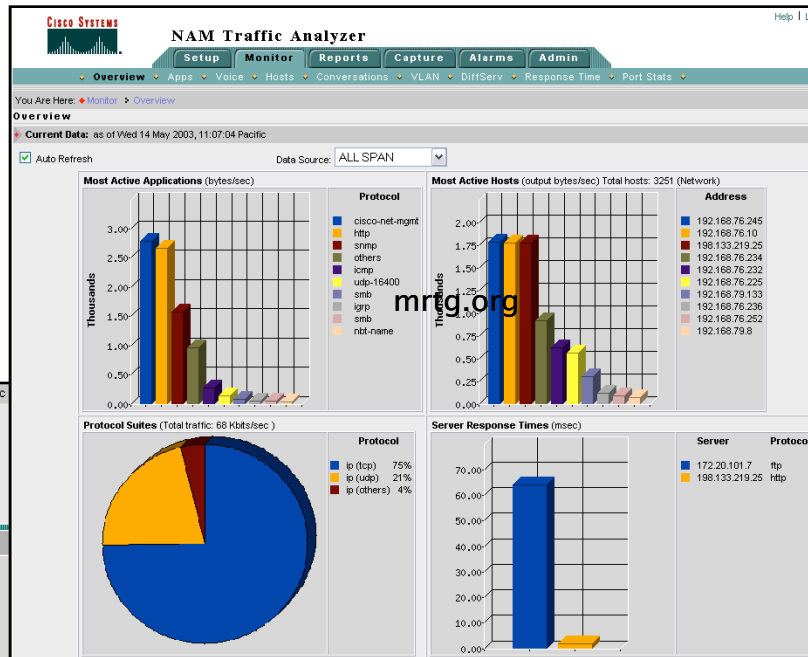


Source: <http://www.nagios.org>

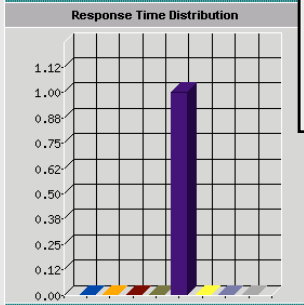
RMON: Remote MONitoring

- RMON is a standard defining how remote probes or agents relay network traffic information back to a central console
- Not as prevalent as SNMP or NetFlow—supported mainly by commercial network management systems
- Cisco Network Analysis Module-2 (NAM-2), ntop (<http://www.ntop.org>) are examples of RMON probes
- Most RMON probes look at raw packets via SPAN/RSPAN and generate statistics from observed traffic
- Mini-RMON statistics available on Cisco Catalyst 6500/NAM-2, provides detailed stats from Layer 2 access ports

NAM-2 Examples



Current Data: as of Fri 16 May 2003, 12:27:27 Pacific



Retries

Response Time Distribution (msec)	
Responses < 5	0
Responses between 5 and 15	0
Responses between 15 and 50	0
Responses between 50 and 100	0
Responses between 100 and 200	1
Responses between 200 and 500	0
Responses between 500 and 3000	0
Responses > 3000	0

CISCO SYSTEMS NAM Traffic Analyzer

Setup Monitor Reports Capture Alarms Admin

You Are Here: Monitor > Overview

Overview

Current Data: as of Fri 16 May 2003, 14:22:50 Pacific

Auto Refresh Data Source: ALL SPAN

Current Rates **TopN Chart** **Cumulative Data**

Port Name: Filter Clear

Showing 1-13 of 13 records

Station	Dropped Events/s	Bytes/s	Packets/s	Broadcast/s	Multicast/s	CRC Align Errors/s	Undersize/s	Oversize/s	Fragments/s	Jabbers/s	Collisions/s
2. 1/1	0.51	0.00	1118995.57	7459.23	6.52	6.00	0.00	0.00	0.00	0.00	0.00
3. 3/37	0.05	0.00	1091637.70	7460.63	0.00	1.07	0.00	0.00	0.00	0.00	0.00
4. 3/6	0.01	0.00	3111.78	13.48	0.00	1.02	0.00	0.00	0.00	0.00	0.00
5. 15/1	0.01	0.00	8260.25	60.42	0.00	30.52	0.00	0.00	0.00	0.00	0.00
6. 3/4	0.01	0.00	1367.98	7.87	0.00	1.02	0.00	0.00	0.00	0.00	0.00
7. 3/17	0.00	0.00	380.05	4.27	2.32	1.80	0.00	0.00	0.00	0.00	0.00
8. 3/18	0.00	0.00	380.05	4.27	2.32	1.80	0.00	0.00	0.00	0.00	0.00
9. 2/1	0.00	0.00	3469.28	39.47	3.30	32.97	0.00	0.00	0.00	0.00	0.00
10. 2/2	0.00	0.00	3032.98	38.10	3.30	32.97	0.00	0.00	0.00	0.00	0.00
11. 2/4	0.00	0.00	3025.37	37.53	3.30	33.03	0.00	0.00	0.00	0.00	0.00
12. 2/3	0.00	0.00	3020.48	37.68	3.30	32.97	0.00	0.00	0.00	0.00	0.00
13. 2/6	0.00	0.00	2942.18	37.35	3.30	32.90	0.00	0.00	0.00	0.00	0.00

Rows per page: 15 of 1

Select an item then take an action -->

Details Real-Time Report

Source: Cisco Systems, Inc.

Syslog

- De facto logging standard for hosts, network infrastructure devices, supported in all Cisco routers and switches
- Many levels of logging detail available—choose the level(s) which are appropriate for each device/situation
- ACL logging is generally contraindicated due to CPU overhead—NetFlow provides more information, doesn't max the box
- Can be used in conjunction with Anycast and databases such as MySQL (<http://www.mysql.com>) to provide a scalable, robust logging infrastructure
- Different facility numbers allows for segregation of log information based upon device type, function, other criteria
- Syslog-ng from http://www.balabit.com/products/syslog_ng/ adds a lot of useful functionality

Packet Capture

- Sometimes, there's just no substitute for looking at the packets on the wire
- SPAN/RSPAN/ERSPAN allow packet capture from Cisco Catalyst switches; ip packet export allows packet capture from routers
- Open source tools such as tcpdump, snoop, Wireshark (<http://www.wireshark.org>) on free *NIX or Windows allow inexpensive packet-capture solutions to be built and deployed
- Commercial tools such as Cisco NAM-2, NAI Sniffer/Distributed Sniffer, Wandel and Goltermann available
- Use macroanalytical telemetry such as SNMP, NetFlow, RMON to guide your use of microanalytical telemetry (i.e., packet capture)

NetFlow for Security Purposes



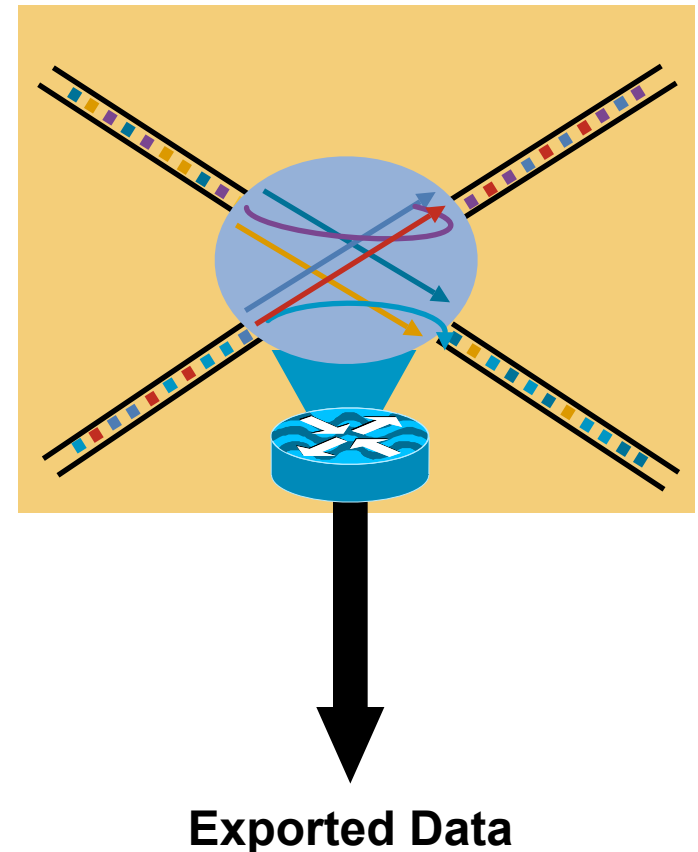
NetFlow Origination

- Developed by Darren Kerr and Barry Bruins at Cisco Systems in 1996
- Primary network accounting technology in the industry
- Emerging standard traffic engineering/capacity planning technology
- Primary network anomaly-detection technology
- Answers questions regarding IP traffic:
 - Who
 - What
 - Where
 - When
 - How
 - What cryptologists call “traffic analysis”

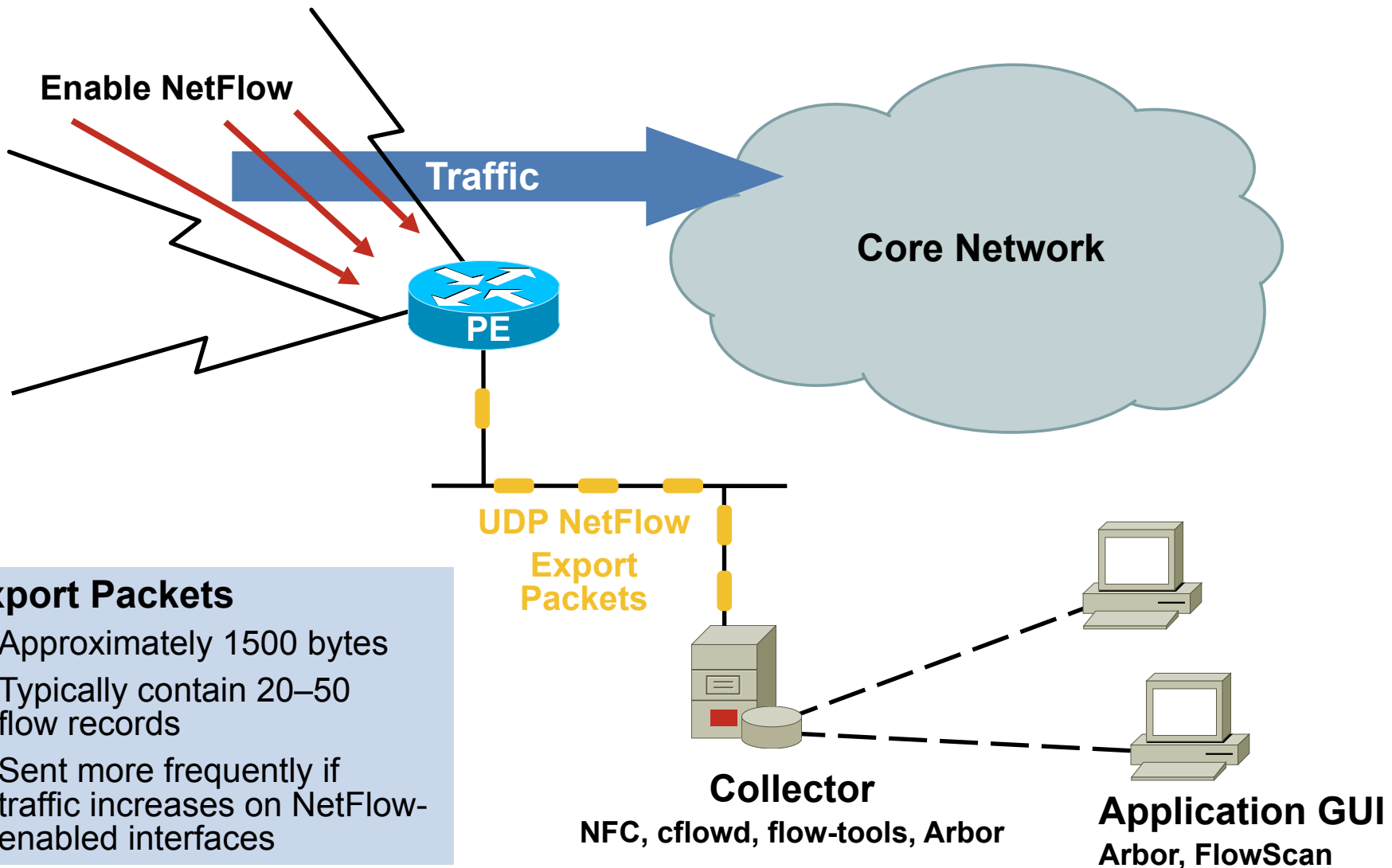
What Is a Flow?

Defined by Seven Unique Keys:

- Source IP address
- Destination IP address
- Source port
- Destination port
- Layer 3 protocol type
- TOS byte (DSCP)
- Input logical interface (ifIndex)



Creating Export Packets



Uses of NetFlow

Service Provider	Enterprise
<ul style="list-style-type: none">▪ Peering Arrangements▪ SLA VPN User Reporting▪ Usage-Based Billing▪ DoS/Worm Detection▪ Traffic Engineering▪ Troubleshooting	<ul style="list-style-type: none">▪ Internet Access Monitoring (Protocol Distribution, Traffic Origin/Destination)▪ Associate Cost of IT to Departments▪ More Scalable Than RMON▪ DoS/Worm Detection▪ Policy Compliance Monitoring▪ Troubleshooting

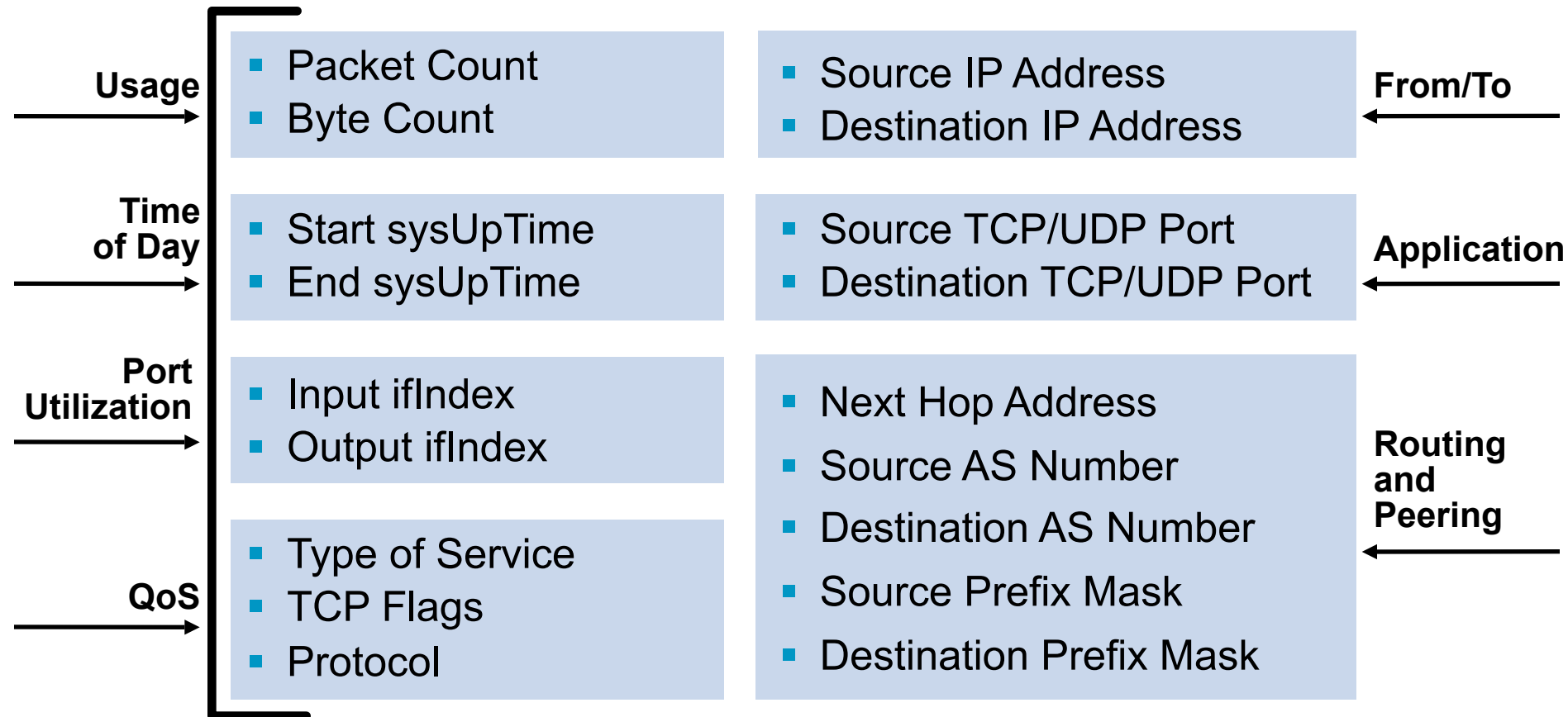
Key Concept: NetFlow Scalability

- Packet capture is like a **wiretap**
- NetFlow is like a **phone bill**
- This level of granularity allows NetFlow to scale for very large amounts of traffic
- We can learn a lot from studying the phone bill
- Who's talking to whom, over what protocols and ports, for how long, at what speed, for what duration, etc.
- NetFlow is a form of **telemetry** pushed from the routers/switches—each one can be a sensor

NetFlow Versions

NetFlow Version	Comments
1	Original
5	Standard and Most Common
7	Specific to Cisco Catalyst 6500 and 7600 Series Switches Similar to Version 5, but Does Not Include AS, Interface, TCP Flag and TOS Information
8	Choice of 11 Aggregation Schemes Reduces Resource Usage
9	Flexible, Extensible File Export Format to Enable Easier Support of Additional Fields and Technologies; Coming Out Now Are MPLS, Multicast, and BGP Next-Hop

Version 5: Flow Format



IOS NetFlow Configuration— Version 5 Export

ip cef

...

interface FastEthernet0

ip address 192.168.131.100 255.255.255.0

no ip redirects

no ip unreachable

no ip proxy-arp

ip flow ingress

...

ip flow-export destination 192.168.131.200 2055

...

ip flow-export version 5

CatOS NetFlow Configuration— Version 5 Export

```
#mls
! Set NetFlow Flow Mask to Full
set mls flow full
! Configure NetFlow Export Version 5
set mls nde version 5
! Identify NetFlow Collector
set mls nde 192.168.131.200 2055
! Enable NetFlow Data Export (NDE)
set mls nde enable
```

Why a New Version?

- Fixed formats (versions 1, 5, 7 and 8) are not flexible and adaptable

Cisco needed to build a new version each time a customer wanted to export new fields

- When new versions are created, partners need to reengineer to support the new export format

Solution: Build a **Flexible** and **Extensible** Export Format

NetFlow v9 Principles

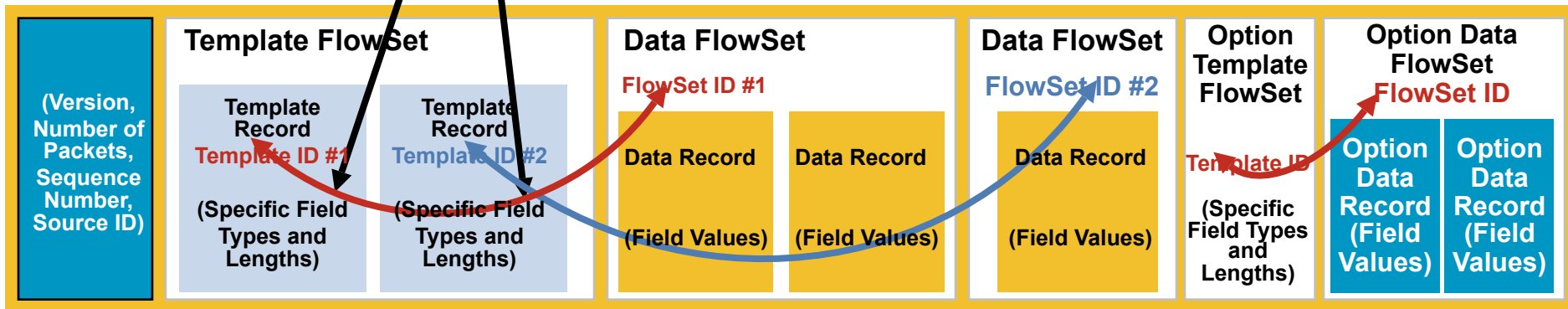
- Version 9 is an **export format**
- Still a push model
- Send the template regularly (configurable)
- Independent of the underlying protocol, it is ready for any reliable protocol (i.e., TCP, SCTP)

NetFlow v9 Export Packet

To Support Technologies such as MPLS or Multicast, this Export Format Can Be Leveraged to Easily **Insert New Fields**

Flows from Interface A

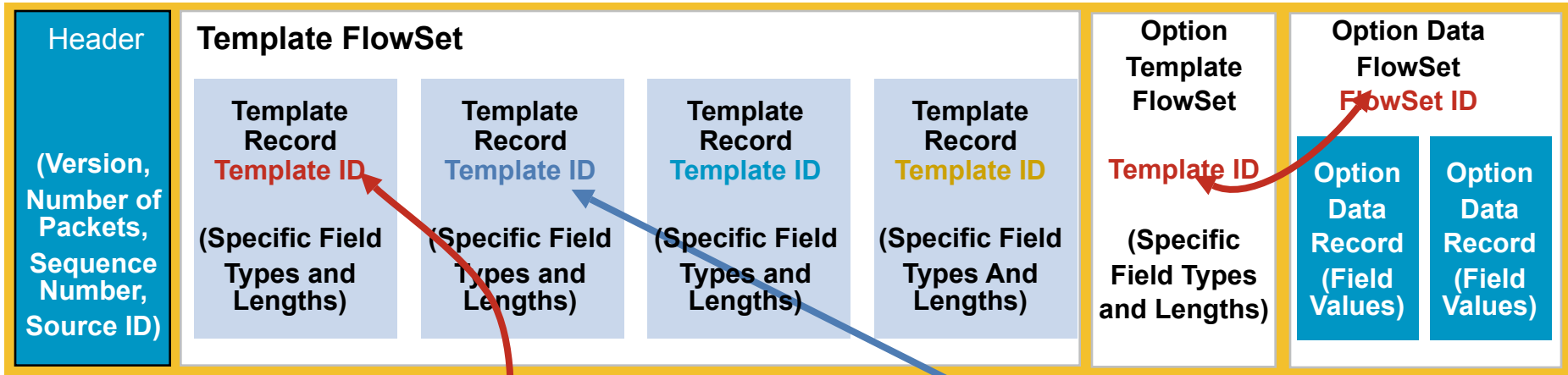
Flows from Interface B



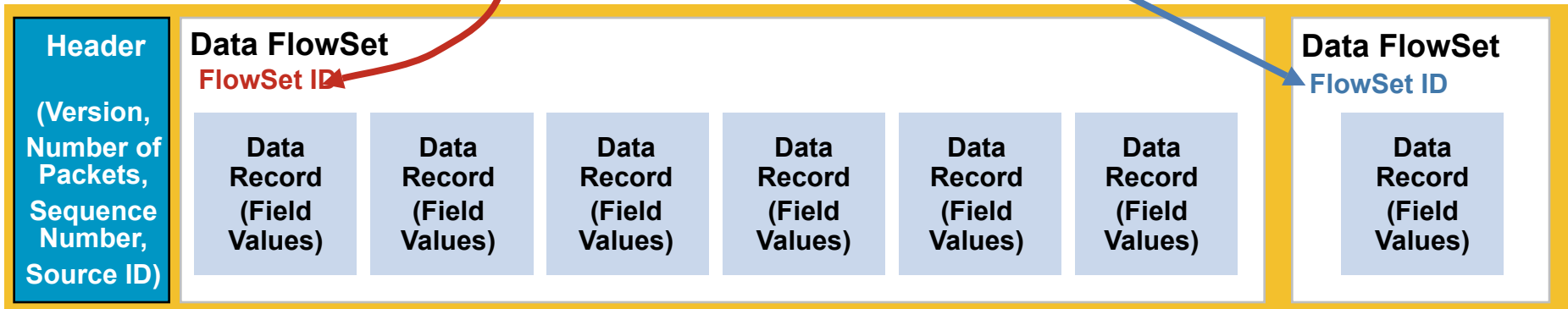
- Matching ID numbers is the way to associate template to the data records
- The header follows the same format as prior NetFlow versions so collectors will be backward compatible
- Each data record represents one flow
- If exported flows have the same fields then they can be contained in the same template record, e.g. unicast traffic can be combined with multicast records
- If exported flows have different fields then they can't be contained in the same template record, e.g. BGP next-hop can't be combined with MPLS aware NetFlow records

NetFlow v9 Flexible Format

Example of Export Packet Right After Router Boot or NetFlow Configuration



Example of Export Packets Containing Mostly Flow Information



NetFlow v9 Export

Configuring Version 9 Export

```
pamela(config)# ip flow-export version ?
```

1
5
9

**Export Versions Available for
Standard NetFlow Flows**

Configuring Version 9 Export for an Aggregation Scheme

```
pamela(config)# ip flow-aggregation cache as
```

```
pamela(config-flow-cache)# enabled
```

```
pamela(config-flow-cache)# export ?
```

destination Specify the Destination IP address

version configure aggregation cache export version

```
pamela(config-flow-cache)# export version ?
```

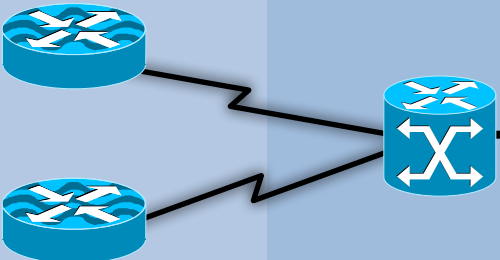

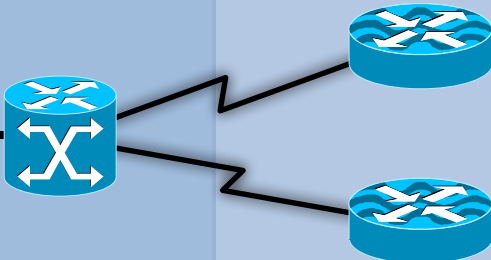
8 Version 8 export format

9 Version 9 export format

**Export Versions Available for
Aggregated NetFlow Flows**

```
pamela(config-flow-cache)# export version 9
```

NetFlow in the Topology

Network Layer	Access	Distribution	Core	Distribution	Access
					
Applications	<ul style="list-style-type: none"> Attack detection User (IP) monitoring Application monitoring 	<ul style="list-style-type: none"> Billing Chargeback AS peer monitoring Attack detection 	<ul style="list-style-type: none"> Traffic engineering Traffic analysis Attack detection 	<ul style="list-style-type: none"> Billing Chargeback AS peer monitoring Attack detection 	<ul style="list-style-type: none"> Attack detection User (IP) monitoring Application monitoring
NetFlow Features	<ul style="list-style-type: none"> Aggregation schemes (v8) "show ip cache flow" command Arbor Networks 	<ul style="list-style-type: none"> NetFlow MPLS egress accounting BGP next-hop (v9) Arbor Networks 	<ul style="list-style-type: none"> MPLS Aware NetFlow (v9) BGP Next-hop (v9) Sampled NetFlow Arbor Networks 	<ul style="list-style-type: none"> NetFlow MPLS Egress Accounting BGP Next-hop (v9) Arbor Networks 	<ul style="list-style-type: none"> Aggregation Schemes (v8) "show ip cache flow" command Arbor Networks

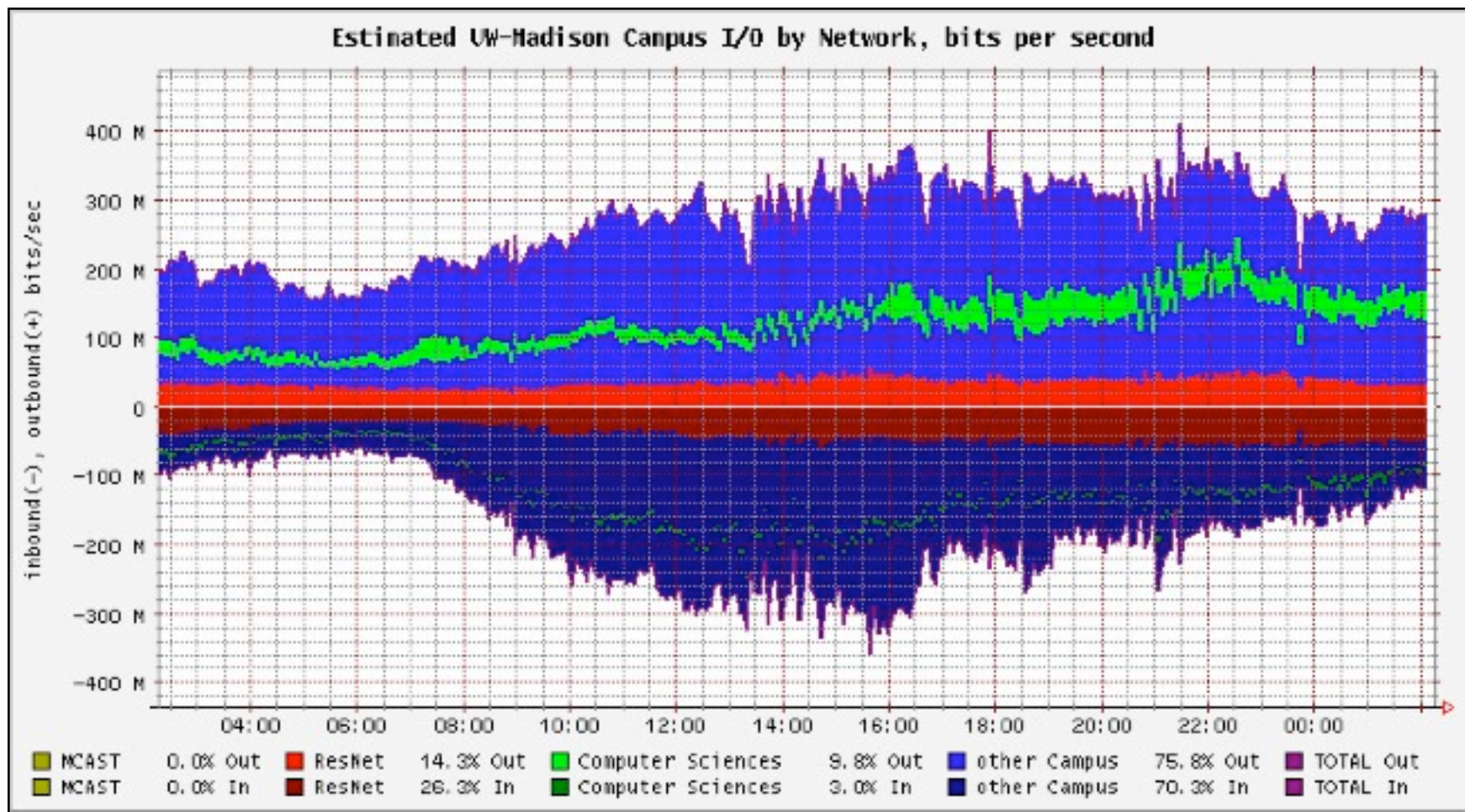
Getting Started with NetFlow Collection: The OSU Flow-Tools

- Open source NetFlow collection and retrieval tools
- Developed and maintained by Mark Fullmer, available from <http://www.splintered.net/sw/flow-tools/>
- Runs on common *NIX platforms (Linux, FreeBSD, Mac OS/X, Solaris, etc.)
- Command-line tools allow for very display/sorting of specific criteria (source/dest IP, source/dest ASN, protocol, port, etc.)
- Data can be batched and imported into database such as Oracle, MySQL, Postgres, etc.
- Can be combined with other tools to provide visualization of traffic patterns
- Many other useful features—check it out today

Getting Started with NetFlow Visualization: FlowScan

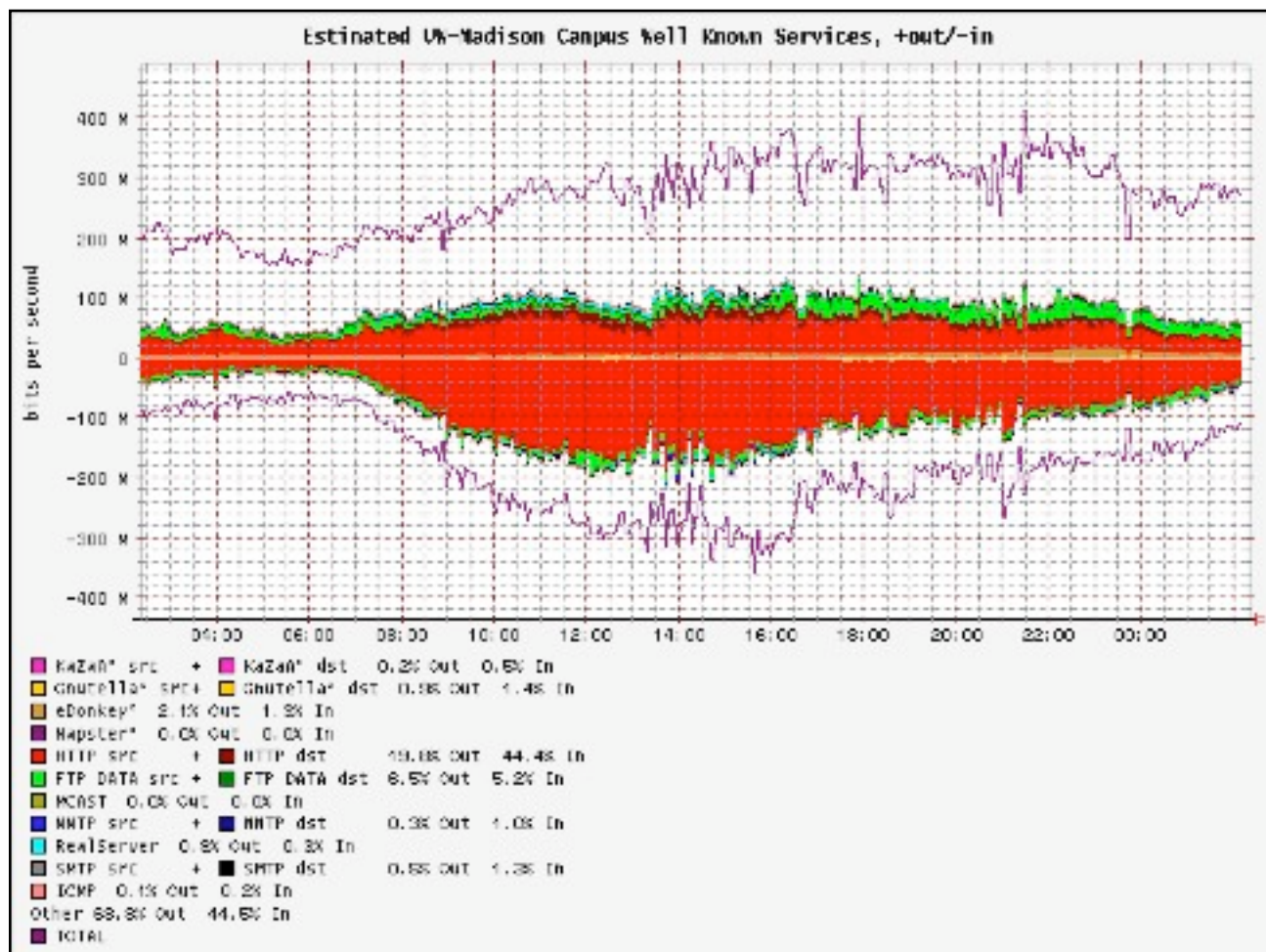
- Open source NetFlow graphing/visualization tools
- Developed and maintained by Dave Plonka, available from <http://net.doit.wisc.edu/~plonka/FlowScan/>
- Runs on common *NIX platforms (Linux, FreeBSD, Mac OS/X, Solaris, etc.)
- Makes use of NetFlow data collected via flow-tools to build traffic graphs
- Top-talkers by subnet, other types of reports supported
- Makes use of RRDTool for graphing
- Add-ons such as JKFlow module allow more detailed graphing

Example: FlowScan Graphs



Source: University of Wisconsin

Example: FlowScan Graphs



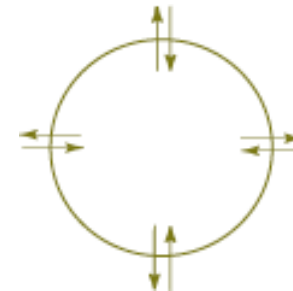
Source: University of Wisconsin

What Is an Anomaly?

- An event or condition in the network that is identified as a statistical abnormality when compared to typical traffic patterns gleaned from previously collected profiles and baselines

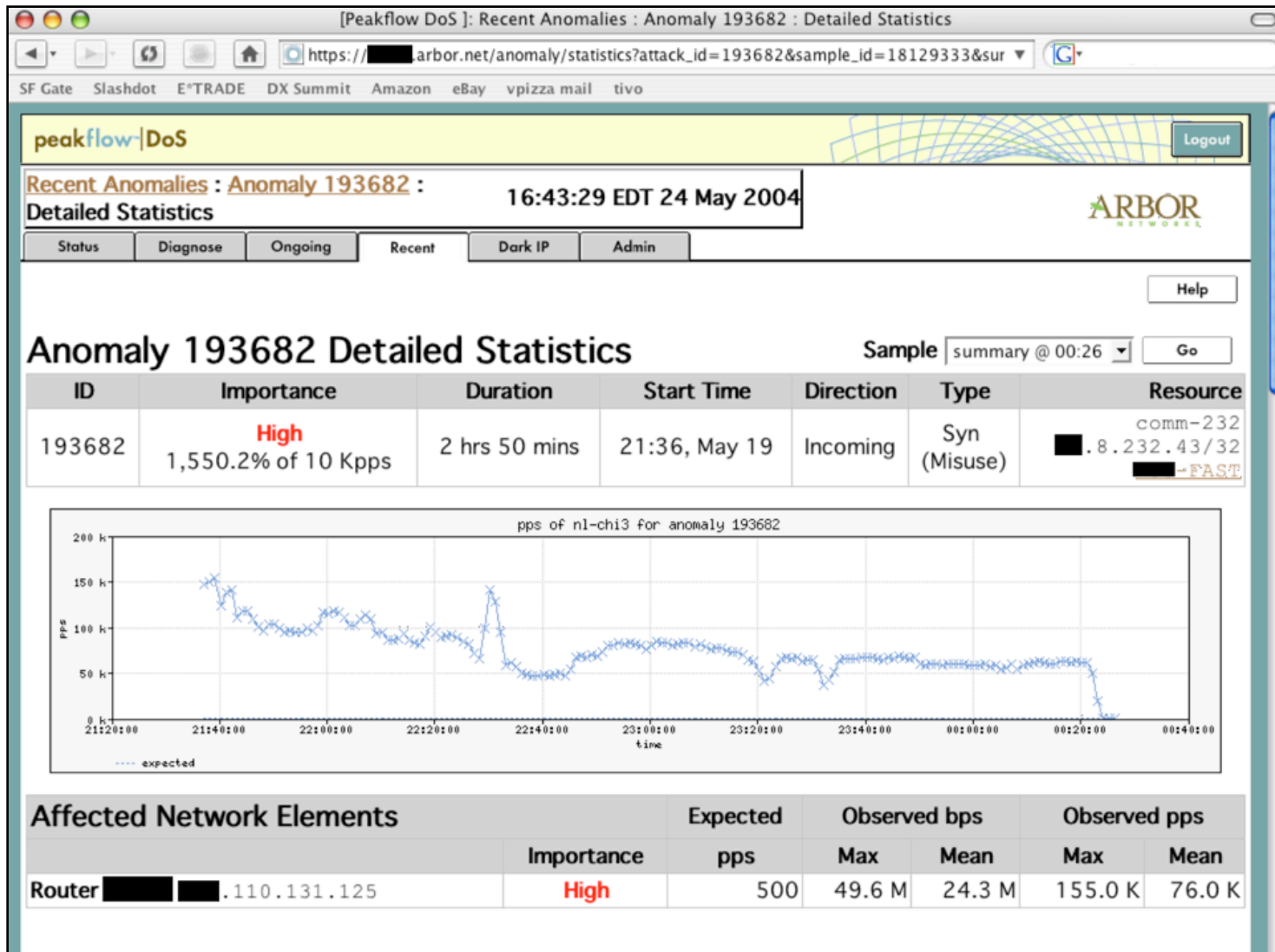
NetFlow-Based Traffic Characterization and Anomaly Detection with Arbor Networks

Network Anomaly Detection and Traffic Characterization/
Capacity Planning

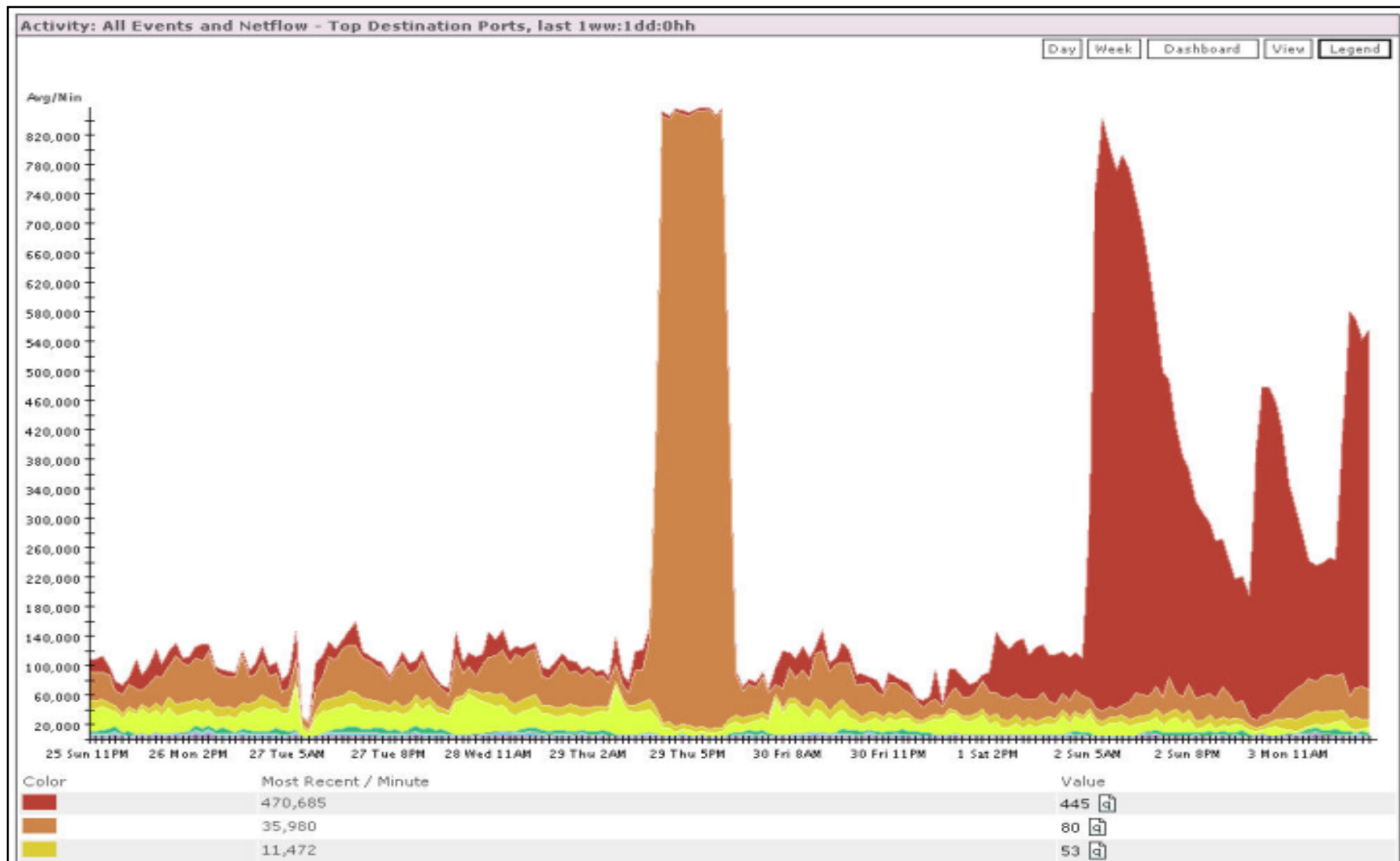


- Most widely deployed anomaly detection system for SPs
- Uses NetFlow to quickly identify, classify, and scope DoS, worms, etc.
- Traffic component combines NetFlow traffic characterization with BGP
- Allows comprehensive peering analysis in real-time
- A “force multiplier” which greatly reduces reaction-times by providing the relevant information up-front
- Can also generate its own flows from packet-capture if NetFlow isn’t available

Anomaly Example: Detail



Sasser Detection



Traceback Techniques



Traceback Essentials

- If source prefix is not spoofed:
 - Routing table
 - Internet Routing Registry (IRR)—whois
 - Direct site contact—ARIN, RIPE, APNIC
- If source prefix is spoofed:
 - Trace packet flow through the network
 - Find upstream connection
 - Upstream needs to continue tracing

Traceback Spoofed IPv4 Addresses

- Source: inside or outside?
- Once you have a fundamental understanding of the type of attack (source address and protocol type), you then need to trace to the ingress point
- Two main techniques:
 - Hop-by-hop
 - Jump to ingress

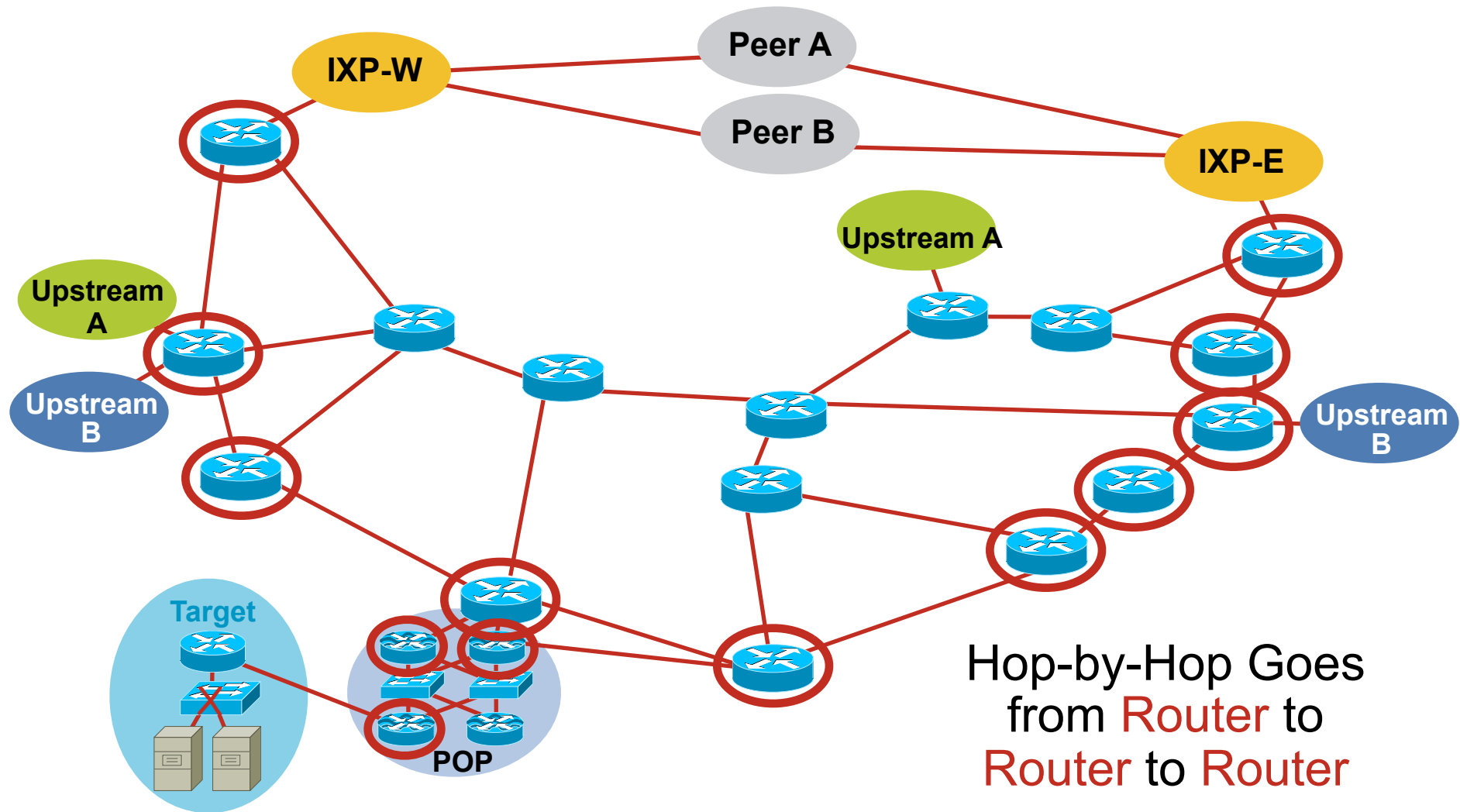
Traceback via Hop-by-Hop Technique

Hop-by-Hop Traceback Takes Time

- Starts from the beginning and traces to the source of the problem
- Needs to be done on each router
- Often requires splitting—tracing two separate paths
- Speed is the limitation of the technique



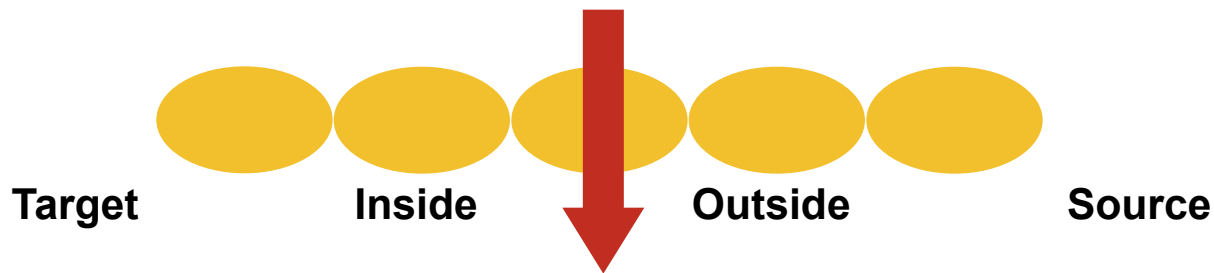
Traceback via Hop-by-Hop Technique



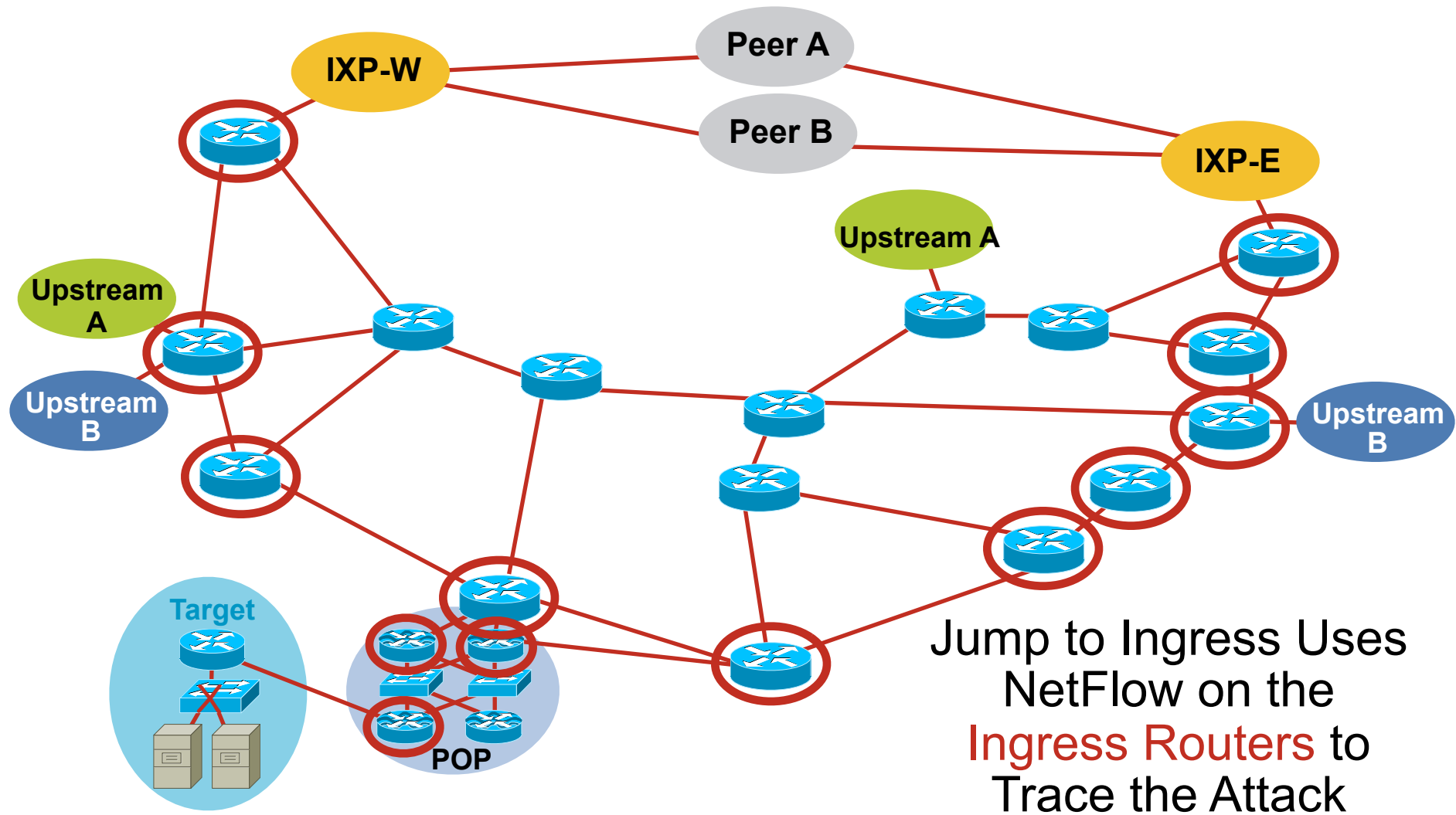
Traceback via the Jump to Ingress Technique

Jump to Ingress Tracebacks Divides the Problem in Half

- Is the attack originating from **inside** the network or **outside** the network?
- Jump to the ingress border routers to see if the attack is entering the network from the outside
- Advantage: speed—are we the source or is someone else the source?



Traceback via the Jump to Ingress Technique



Traceback Spoofed IPv4 Addresses

Traceback Techniques

- Apply temporary ACLs with log-input and examine the logs (like classification)
- Query NetFlow's flow table
 - Show ip cache-flow if NetFlow is enabled
- Backscatter traceback technique
- Traceback using NetFlow telemetry

Traceback with ACLs

- Original traceback technique
- Risk: inserting change into a network that is under attack
- Risk: **log-input** requires the forwarding ASIC to punt the packet to capture log information
- BCP is to apply the filter, capture just enough information, then remove the filter

Traceback with ACLs

```
access-list 170 permit icmp any any echo
access-list 170 permit icmp any any echo-reply log-input
access-list 170 permit udp any any eq echo
access-list 170 permit udp any eq echo any
access-list 170 permit tcp any any established
access-list 170 permit tcp any any
access-list 170 permit ip any any

interface serial 0
  ip access-group 170 out
! Wait a short time - (i.e 10 seconds)
  no ip access-group 170 out
```

Traceback with ACLs Output

- Validate the capture with **show access-list 170**; make sure it the packets we counted
- View the log with **show logging** for input interface:

```
%SEC-6-IPACCESSLOGDP: list 170 permit icmp 192.168.212.72  
(Serial0 *HDLC*) -> 172.19.61.10 (0/0), 1 packet
```

```
%SEC-6-IPACCESSLOGDP: list 170 permit icmp 172.16.132.154  
(Serial0 *HDLC*) -> 172.19.61.10 (0/0), 1 packet
```

```
%SEC-6-IPACCESSLOGDP: list 170 permit icmp 192.168.45.15  
(Serial0 *HDLC*) -> 172.19.61.10 (0/0), 1 packet
```

```
%SEC-6-IPACCESSLOGDP: list 170 permit icmp 192.168.45.142  
(Serial0 *HDLC*) -> 172.19.61.10 (0/0), 1 packet
```

```
%SEC-6-IPACCESSLOGDP: list 170 permit icmp 172.16.132.47  
(Serial0 *HDLC*) -> 172.19.61.10 (0/0), 1 packet
```

Netflow Traceback Techniques



Traceback with NetFlow

Victim

```
router1#sh ip cache flow | include <destination>  
Se1 <source> Et0 <destination> 11 0013 0007 159  
.... (lots more flows to the same destination)
```

The Flows Come from Serial 1

```
router1#sh ip cef se1
```

Prefix	Next Hop	Interface
0.0.0.0/0	10.10.10.2	Serial1
10.10.10.0/30	attached	

Find the Upstream Router on Serial 1

Continue on This Router

show ip cache flow

```
router_A#sh ip cache flow
```

```
IP packet size distribution (85435 total packets):
```

```
1-32    64    96   128   160   192   224   256   288   320   352   384   416   448   480
.000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

512   544   576  1024  1536  2048  2560  3072  3584  4096  4608
.000 .000 .000 .000 1.00 .000 .000 .000 .000 .000 .000
```

```
IP Flow Switching Cache, 278544 bytes
```

```
2728 active, 1368 inactive, 85310 added
```

```
463824 age polls, 0 flow alloc failures
```

```
Active flows timeout in 30 minutes
```

```
Inactive flows timeout in 15 seconds
```

```
last clearing of statistics never
```

Protocol	Total	Flows	Packets	Bytes	Packets	Active (Sec)	Idle (Sec)
-----	Flows	/Sec	/Flow	/Pkt	/Sec	/Flow	/Flow
TCP-X	2	0.0	1	1440	0.0	0.0	9.5
TCP-other	82580	11.2	1	1440	11.2	0.0	12.0
Total:	82582	11.2	1	1440	11.2	0.0	12.0

Protocol

Flow Information
Summary

Flow Details

SrcIf	SrcIPaddress	DestIf	DestIPaddress	Pr	SrcP	DestP	Pkts
Et0/0	132.122.25.60	Se0/0	192.168.1.1	06	9AEE	0007	1
Et0/0	139.57.220.28	Se0/0	192.168.1.1	06	708D	0007	1
Et0/0	165.172.153.65	Se0/0	192.168.1.1	06	CB46	0007	1

Useful NetFlow CLI Tricks

- Router>show ip cache flow | include <ip address>
Determine flows pertaining to a specific victim or attacker
- Router>show ip cache flow | include _1\$
Determine single packet flows (potential scanning flows)
- Router>show ip cache flow | include K|M\$
Determine really large flows (in 1,000s or 1,000,000s of packets)
- Router>show ip cache flow | include <protocol / port>
Determine flows with specific protocols/ports

Traceback with NetFlow Example

Tracing W32.Blaster Infected Hosts

W32.Blaster-Infected Hosts Attempt to Replicate to Random Systems Using Port 135, Which Is Hex 0087

```
Router>show ip cache flow | include 0087
```

:

SrcIf	SrcIPaddress	DstIf	DstIPaddress	Pr	SrcP	DstP	Pkts
Fa2/0	XX.XX.XX.242	Fa1/0	XX.XX.XX.119	06	0B88	0087	1
Fa2/0	XX.XX.XX.242	Fa1/0	XX.XX.XX.169	06	0BF8	0087	1
Fa2/0	XX.XX.XX.204	Fa1/0	XX.XX.XX.63	06	0E80	0087	1
Fa2/0	XX.XX.XX.204	Fa1/0	XX.XX.XX.111	06	0CB0	0087	1
Fa2/0	XX.XX.XX.204	Fa1/0	XX.XX.XX.95	06	0CA0	0087	1
Fa2/0	XX.XX.XX.204	Fa1/0	XX.XX.XX.79	06	0C90	0087	1

Traceback with NetFlow Telemetry

- Routers on the edge of the network can export NetFlow data reporting detailed traffic flow information
- This **telemetry** can be processed to detect anomalies and to traceback the attack to the source(s)
- Open source and commercial products available
- Arbor PeakFlow provides one example that has operationally proven its value

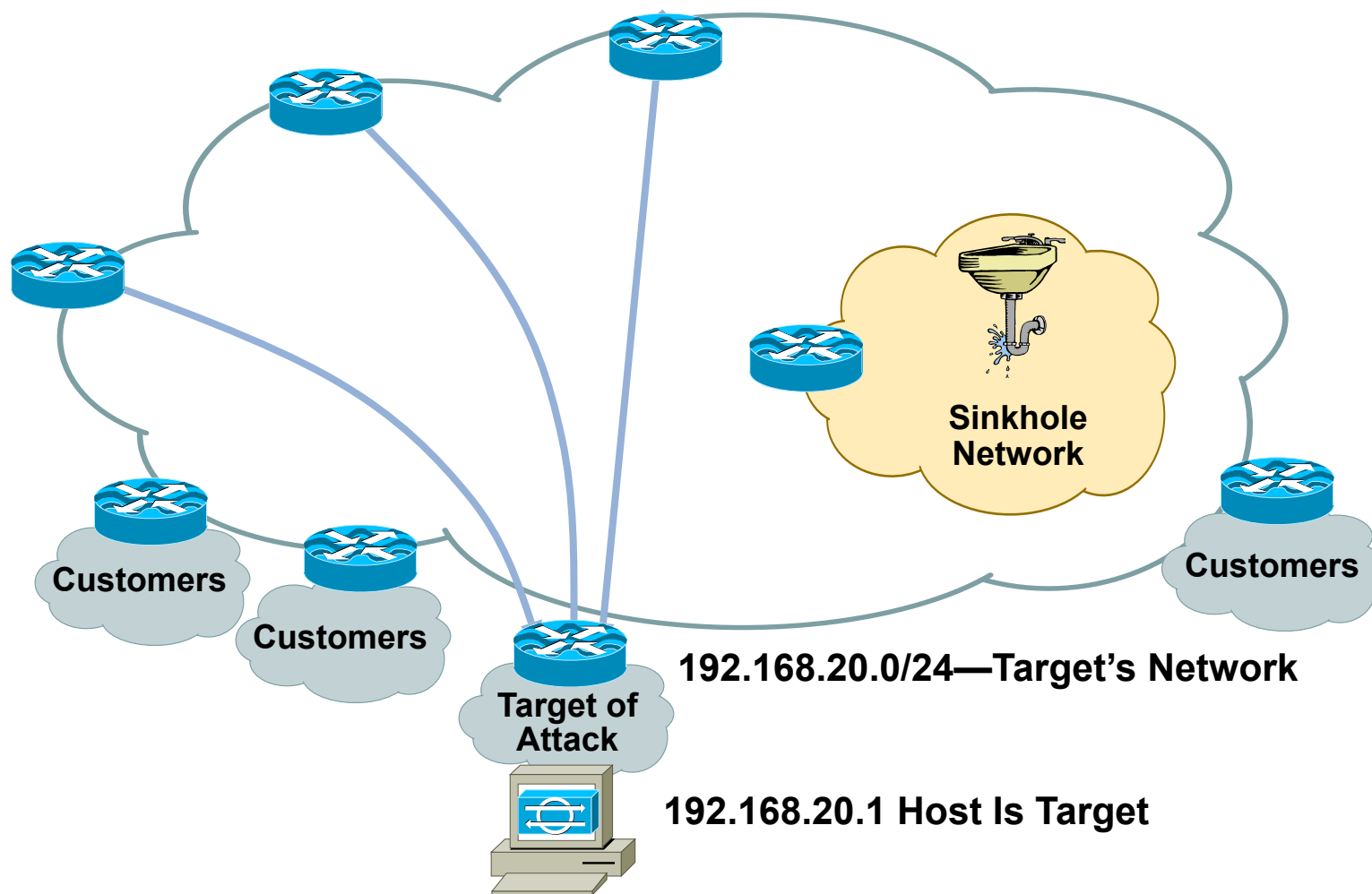
Attract and Analyze: Sinkholes



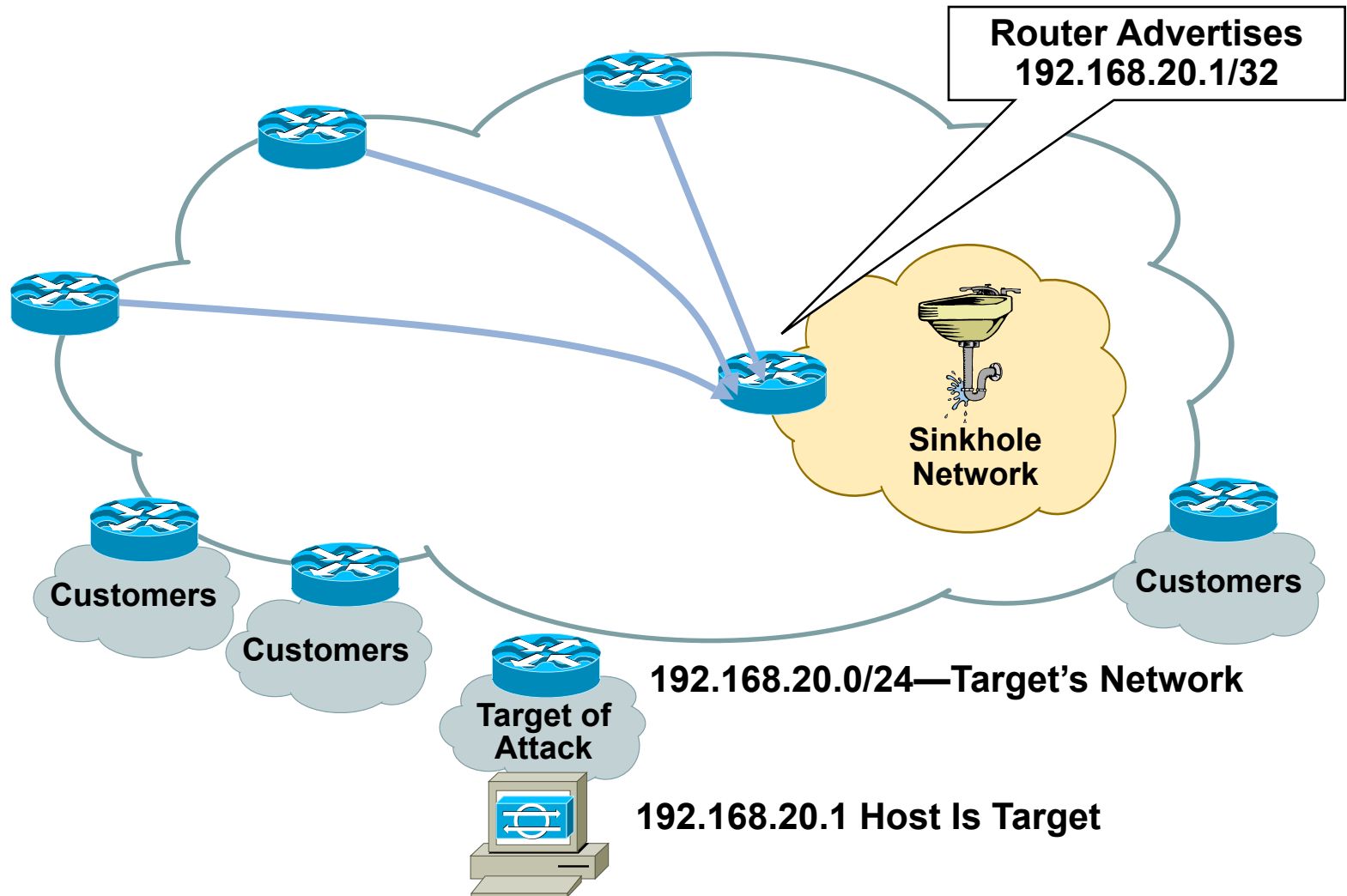
Sinkhole Routers/Networks

- Sinkholes are a topological security feature—think network honeypot
- Router or workstation built to suck in traffic and assist in analyzing attacks (original use)
- Redirect attacks away from the customer—working the attack on a router built to withstand the attack
- Used to monitor attack noise, scans, data from misconfiguration and other activity (via the advertisement of default or unused IP space)
- Traffic is typically diverted via BGP route advertisements and policies
- Leverage instrumentation in a controlled environment
 - Pull the traffic past analyzers/analysis tools

Sinkhole Routers/Networks

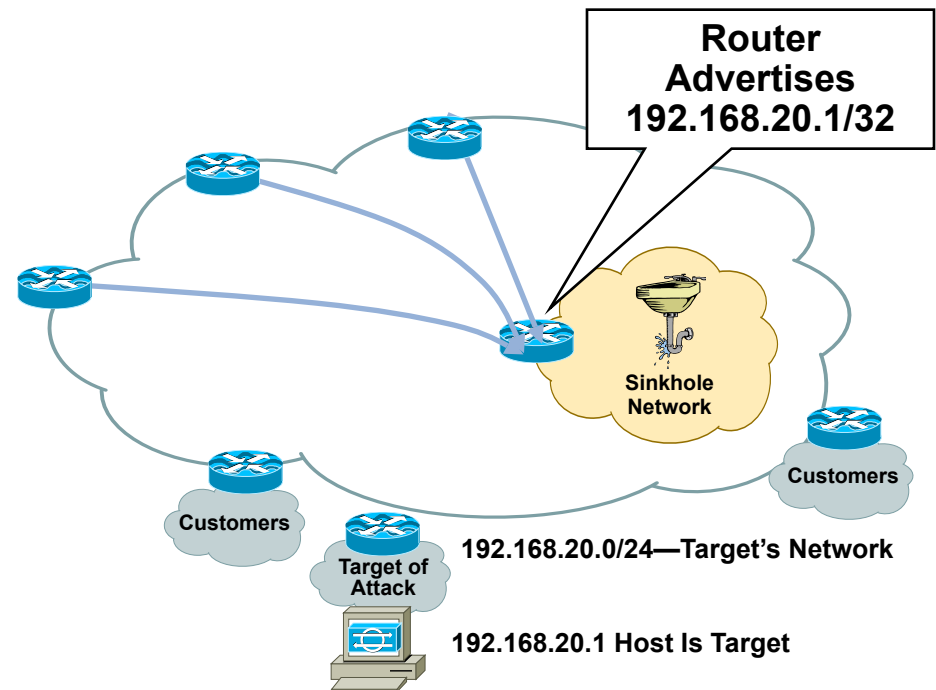


Sinkhole Routers/Networks



Sinkhole Routers/Networks

- Attack is pulled away from customer/aggregation router
- Can now apply classification ACLs, packet capture, etc.
- Objective is to minimize the risk to the network while investigating the attack incident



Sinkhole Routers/Networks

- Advertising “space” from the sinkhole will pull down all sorts of garbage (and potentially interesting) traffic:

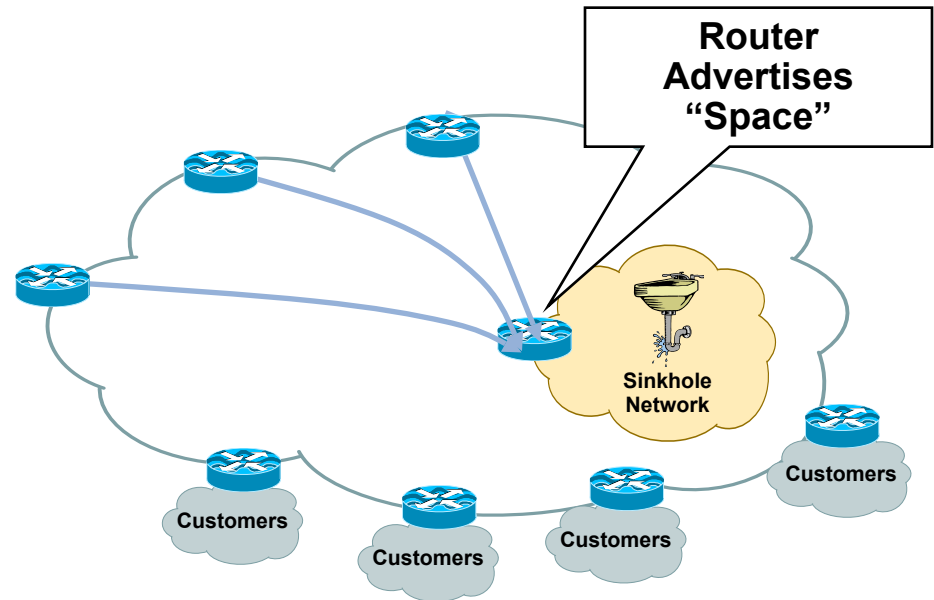
Customer traffic when circuits flap

Network scans to unallocated address space

Worm traffic

Backscatter

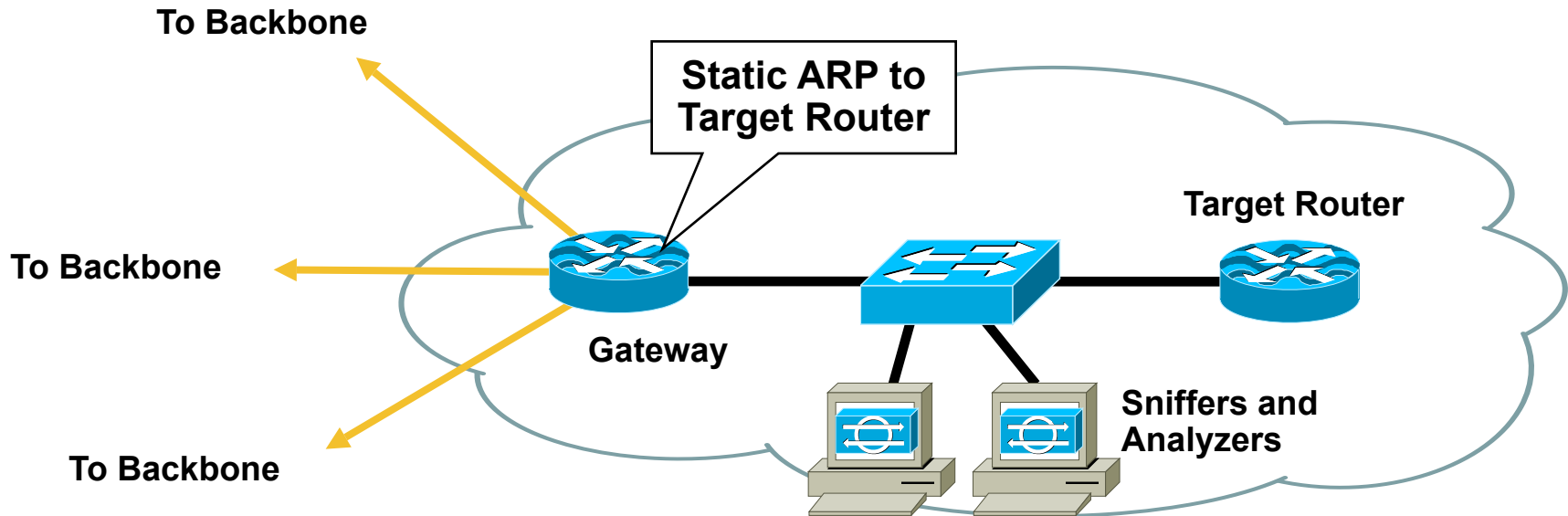
- Place tracking tools in the sinkhole network to monitor the noise



What to Monitor in a Sinkhole?

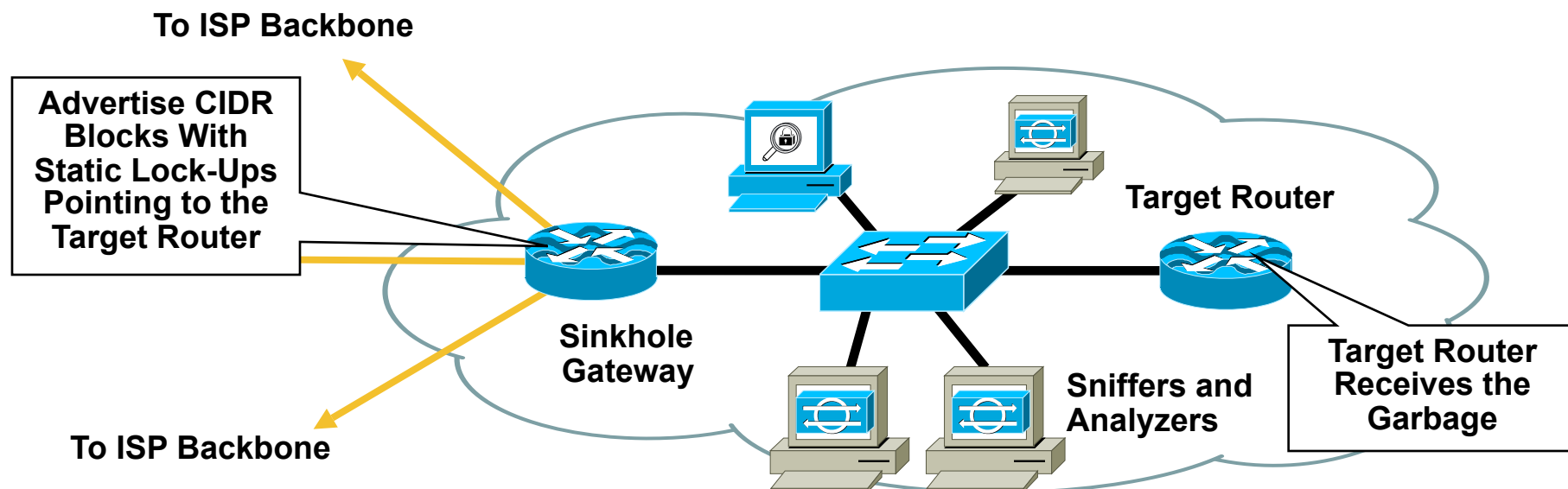
- Scans on dark IP (allocated and announced but unassigned address space)
 - Who is scoping out the network—pre-attack planning, worms
- Scans on bogons (unallocated)
 - Worms, infected machines, and Bot creation
- Backscatter from attacks
 - Who is getting attacked
- Backscatter from garbage traffic (RFC-1918 leaks)
 - Which customers have misconfiguration or “leaking” networks

Sinkhole Architecture



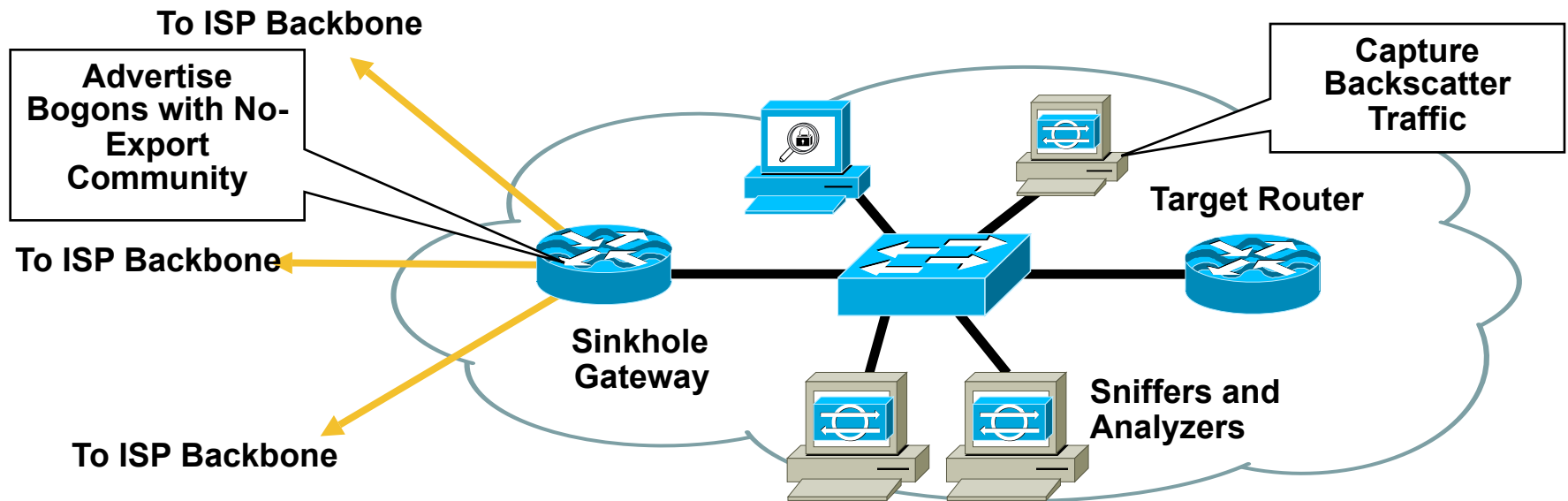
- Expand sinkhole with dedicated router into a variety of tools
- Pull DDoS attack to the sinkhole and forward data toward target router
- Static ARP to the target router keeps the sinkhole operational—target router can crash from attack and static ARP will keep gateway forwarding traffic to the Ethernet switch—rather than generating lots of ICMP error messages
- Observe trends and deviations, reserve packet detail for research and specific analysis

Sinkholes: Advertising Dark IP



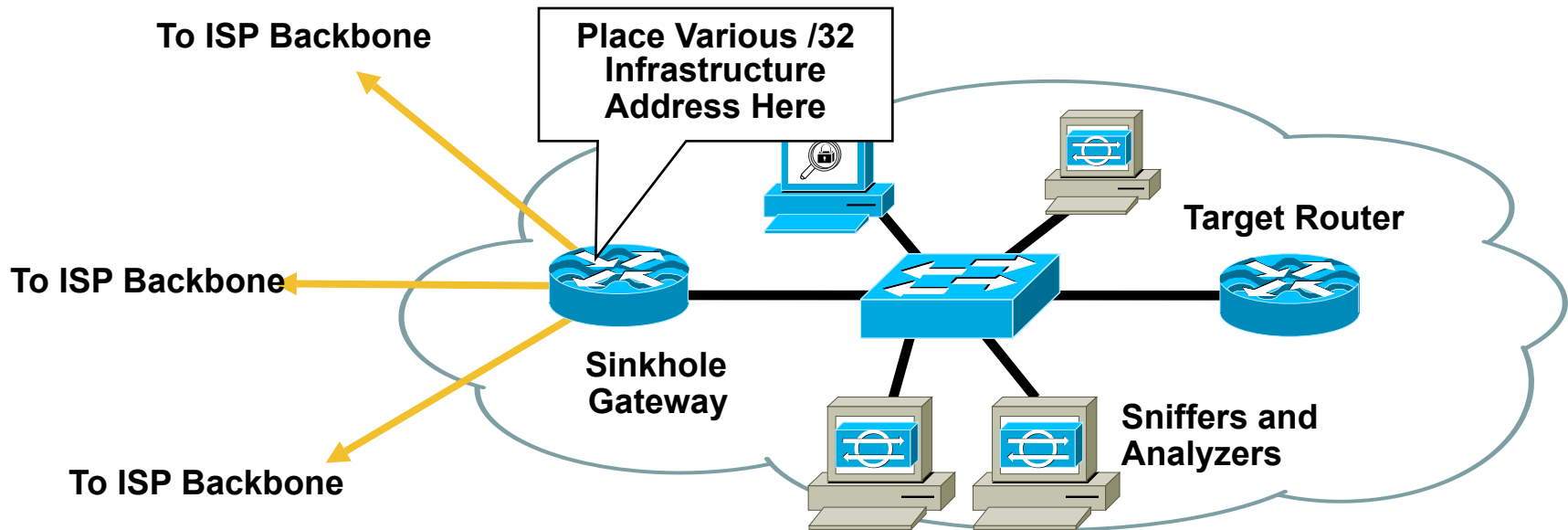
- Move the CIDR Block Advertisements (or at least more-specifics of those advertisements) to sinkholes
- Does not impact BGP routing—route origination can happen anywhere in the iBGP mesh (careful about MEDs and aggregates)
- Control where you drop the packet
- Turns networks inherent behaviors into a security tool

Monitoring Backscatter



- Advertise bogon blocks with NO_EXPORT community and an explicit safety community (plus prefix-based egress filtering on the edge)
- Static/set the BGP NEXT_HOP for the bogon to a backscatter collector workstation (as simple as TCPdump)
- Pulls in backscatter for that range—allows monitoring

Monitoring Scan Rates

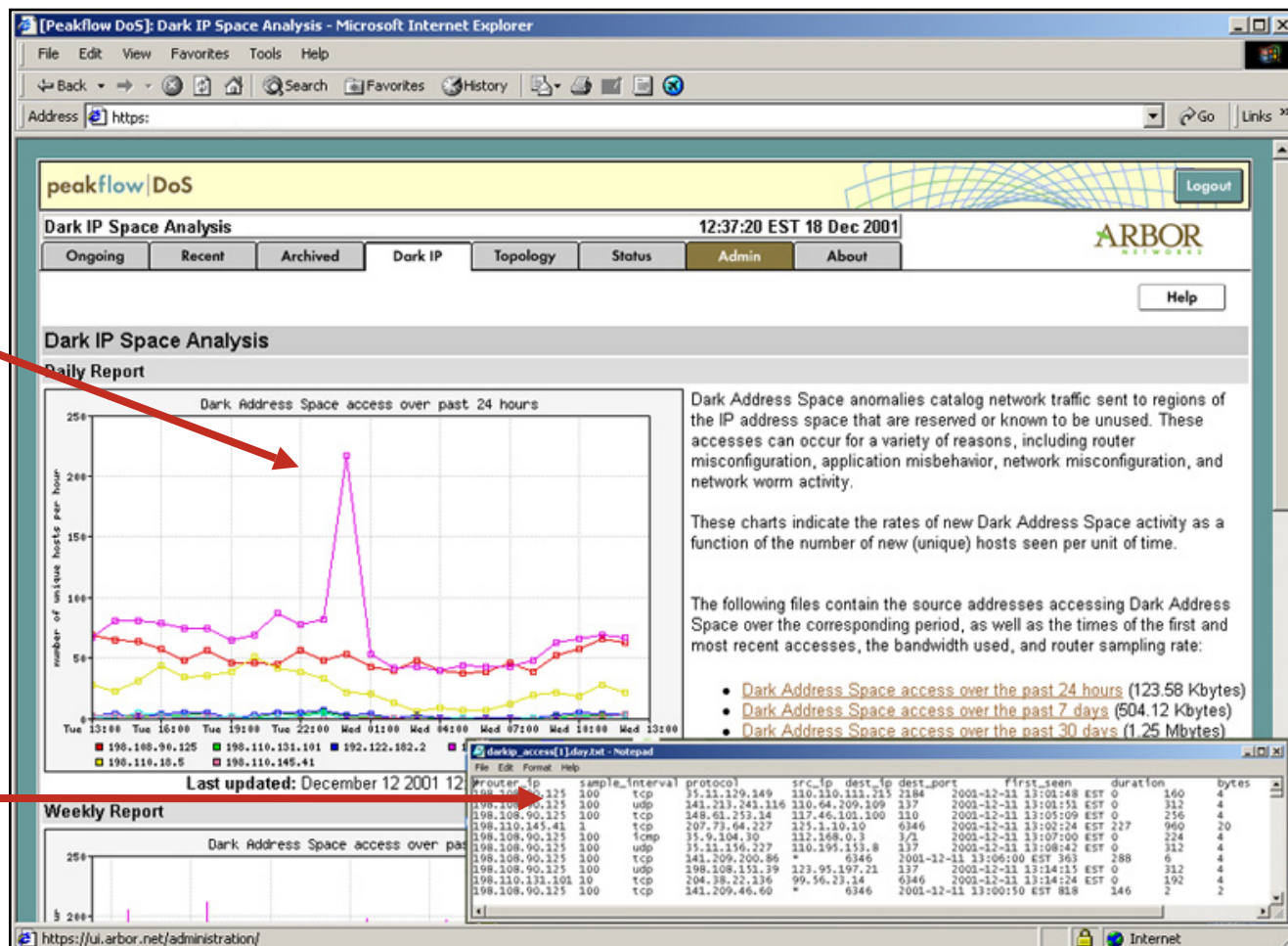


- Select /32 (or larger) address from different block of your address space; advertise them out the sinkhole
- Assign them to a workstation built to monitor and log scans (Arbor Network's Dark IP PeakFlow module is one turnkey commercial tool that can monitor scan rates via data collected from the network)

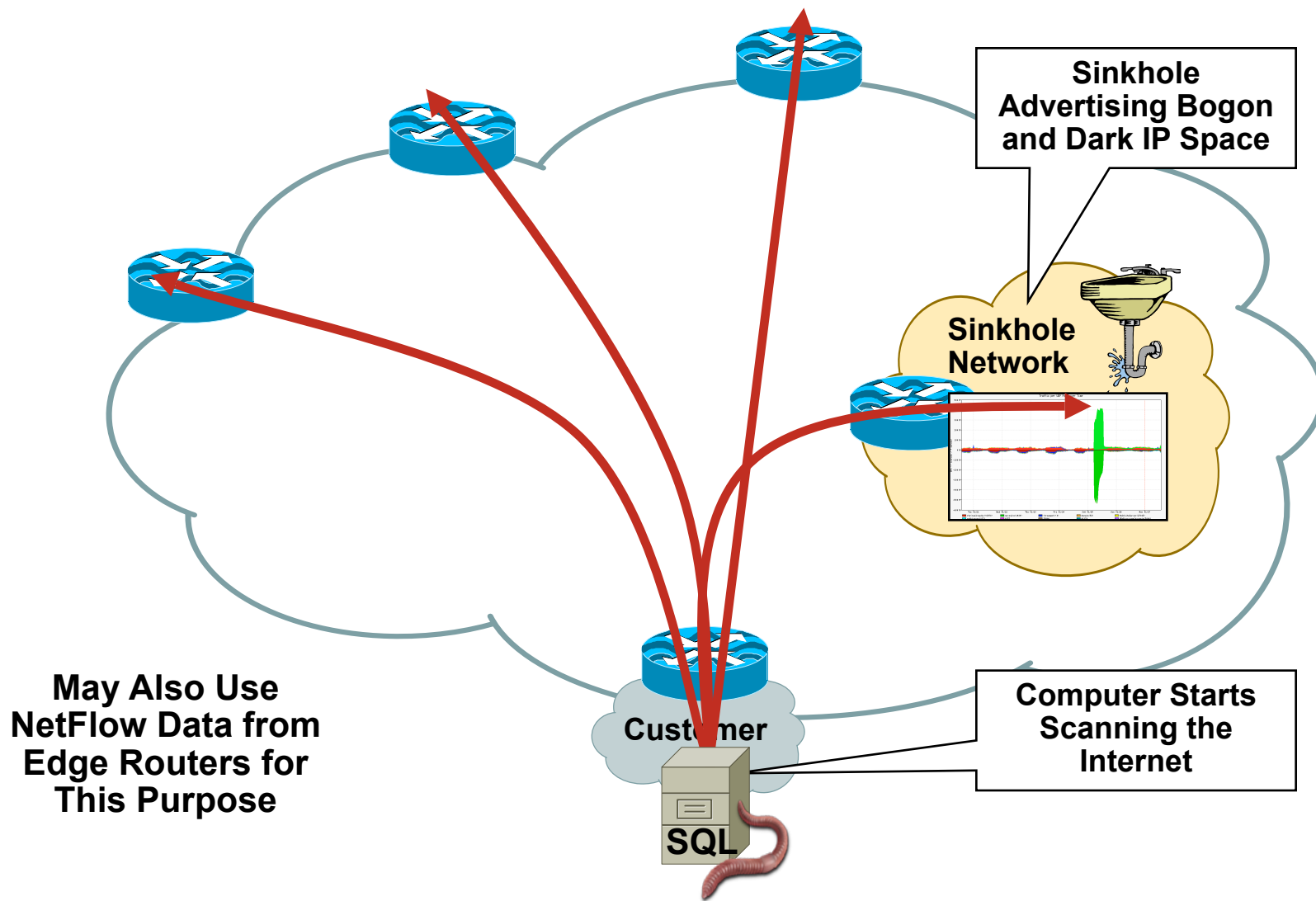
Worm Detection and Reporting UI

Operator
Instantly Notified
of Worm Infection

System
Automatically
Generates a List
of Infected Hosts
for Quarantine
and Cleanup



Sinkholes: Worm Detection



But I'm Not a Core Provider?

- All networks aggregate traffic somewhere

Control where and how, control your traffic, not vice versa

- Default route is a “strange attractor”

Do you use a default route? Congratulations, you have a sinkhole

Don't let those packets drop in vain

- Collect data about the traffic and realize the benefits of sinkholes

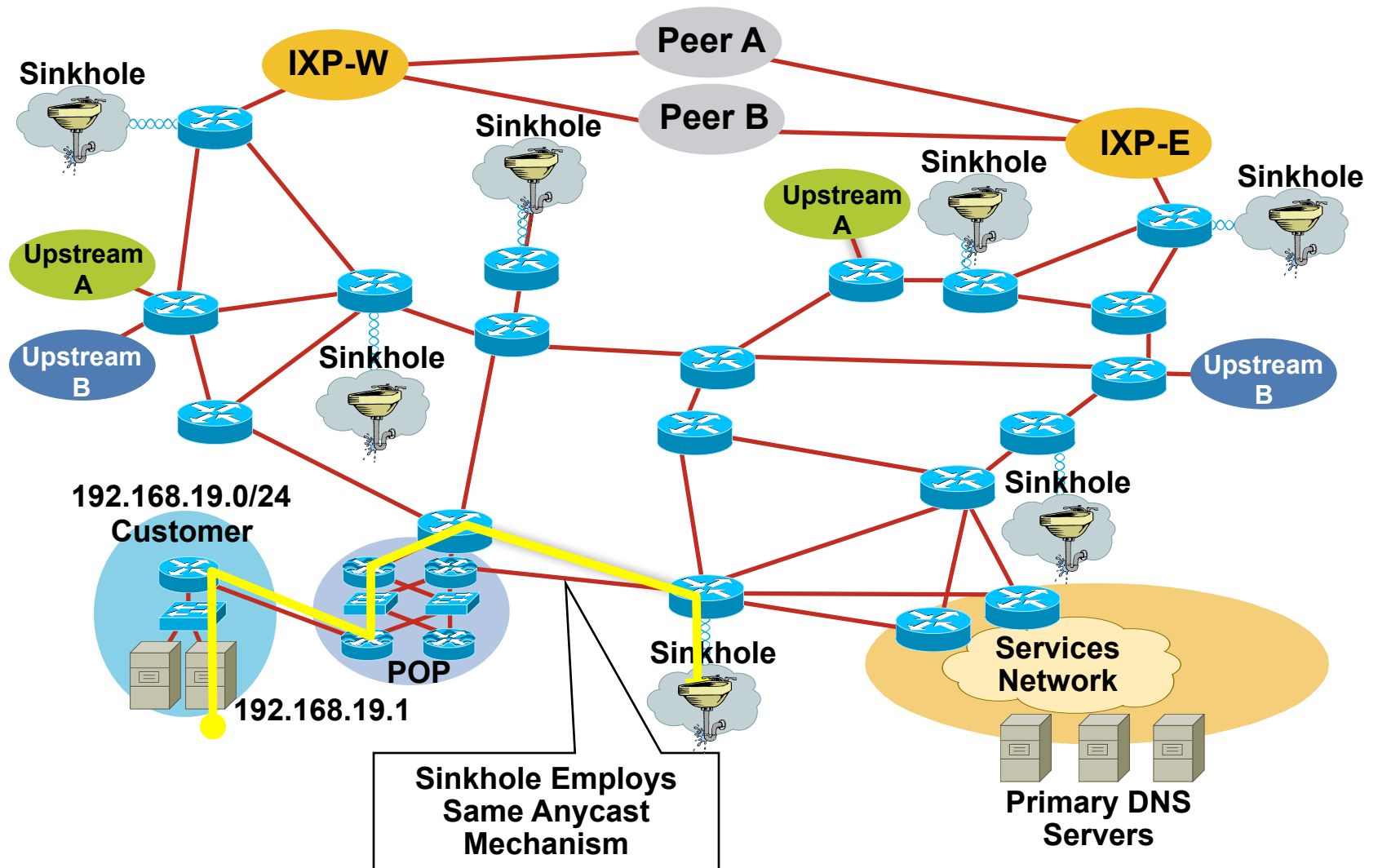
Why Sinkholes?

- They work; providers, enterprise operators and researchers use them in their network for data collection and analysis
- More uses are being found through experience and individual innovation
- Deploying sinkholes correctly takes preparation

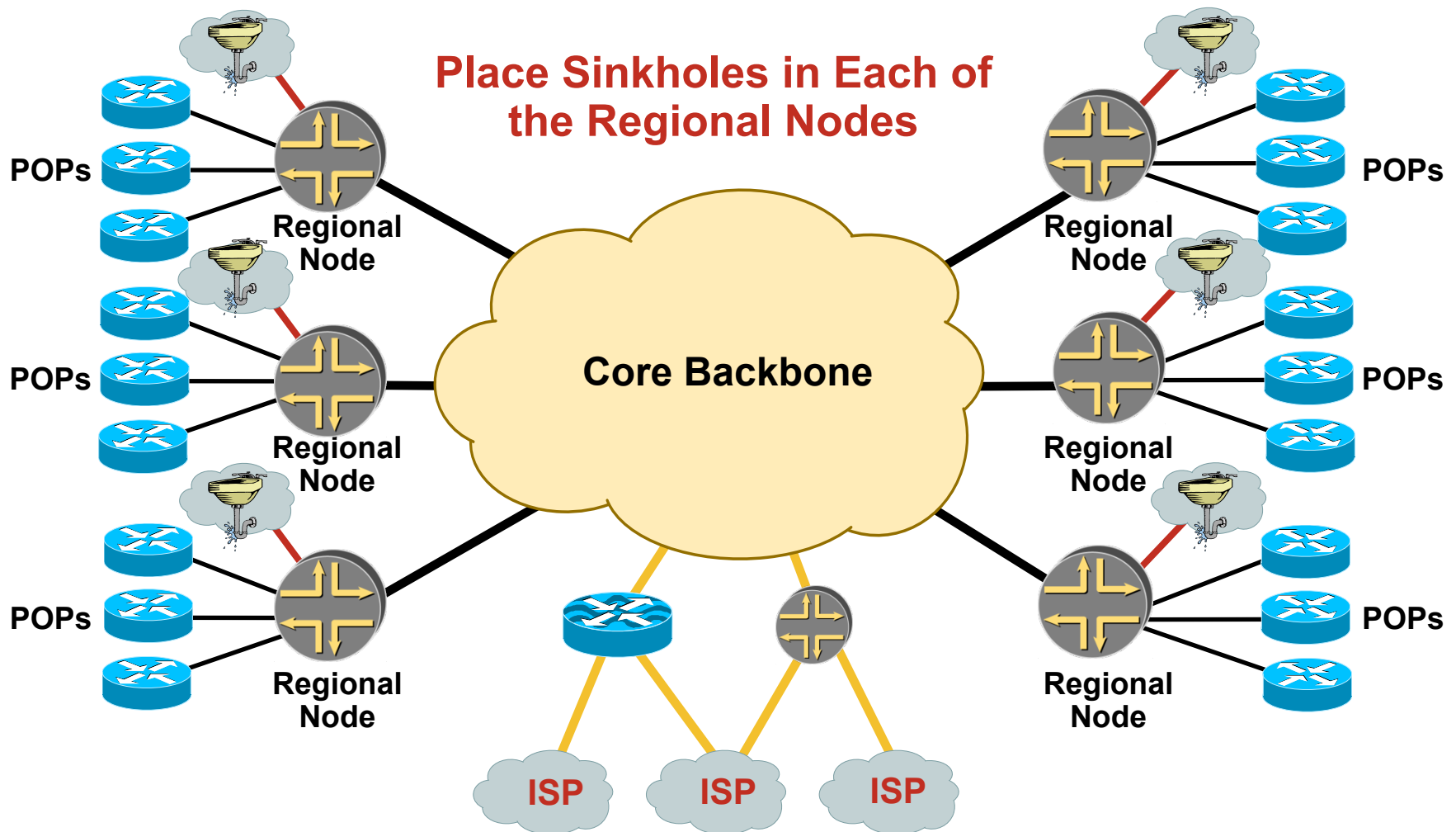
Anycast and Sinkholes

- Sinkholes are designed to pull in traffic, potentially large volumes
- Optimal placement in the network requires mindful integration and can have substantial impact on network performance and availability
- A single sinkhole might require major re-engineering of the network
- Anycast sinkholes provide a means to distribute the load throughout the network

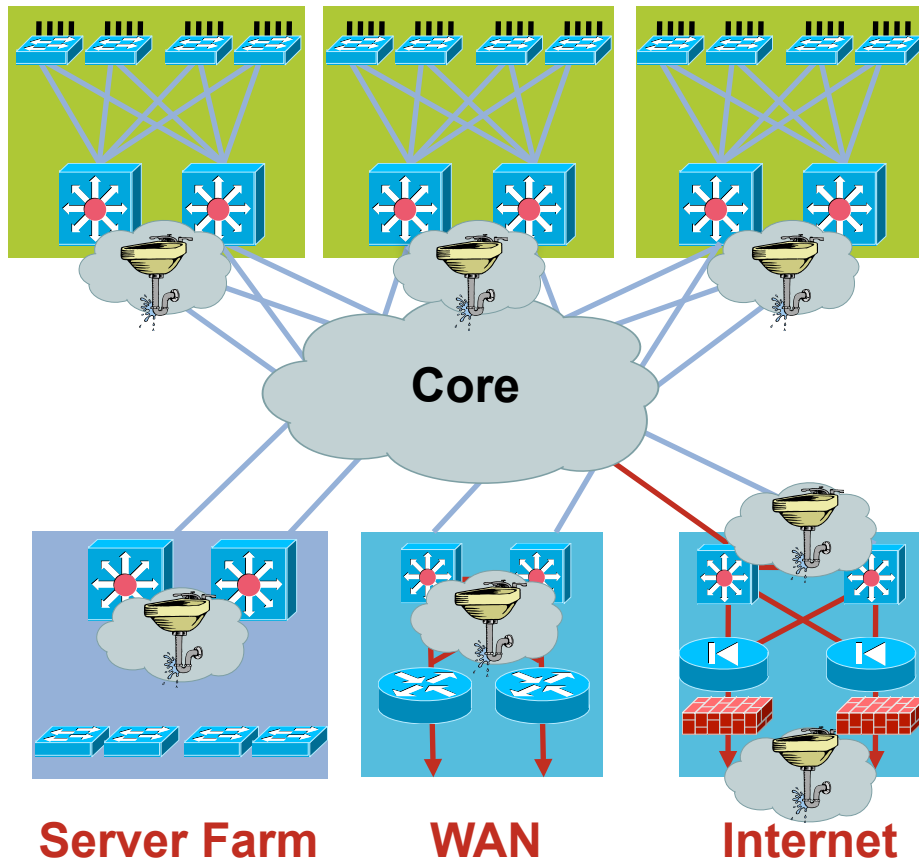
Anycast Sinkholes



Anycast Sinkhole Placement



Enterprise Sinkhole Placement



- Baselineing is the key
Measure derivations from “normal”
- Distribute sinkholes as appropriate for traffic engineering **and** routing architecture
- Some key locations:
 - Inside internet connection
 - In front of servers
 - Distribution layer

Safety Precautions

- Do not allow advertisements to leak:
 - BGP no-export, no-advertise, additive communities
 - Explicit egress prefix policies (community, prefix, etc.)
- Do not allow traffic to escape the sinkhole:
 - Backscatter from a sinkhole defeats the function of a **sinkhole** (egress ACL on the sinkhole router)
- Advanced sinkhole designs
 - True honeypot potential → protect resources in the sinkhole
 - Don't become part of the attack**
 - Filter/rate limit outgoing connections

Reacting to Attacks



Reaction Tools

- Wide range of response options exists

Access-control lists

QoS tools such as CAR, traffic policing and NBAR

Firewalls

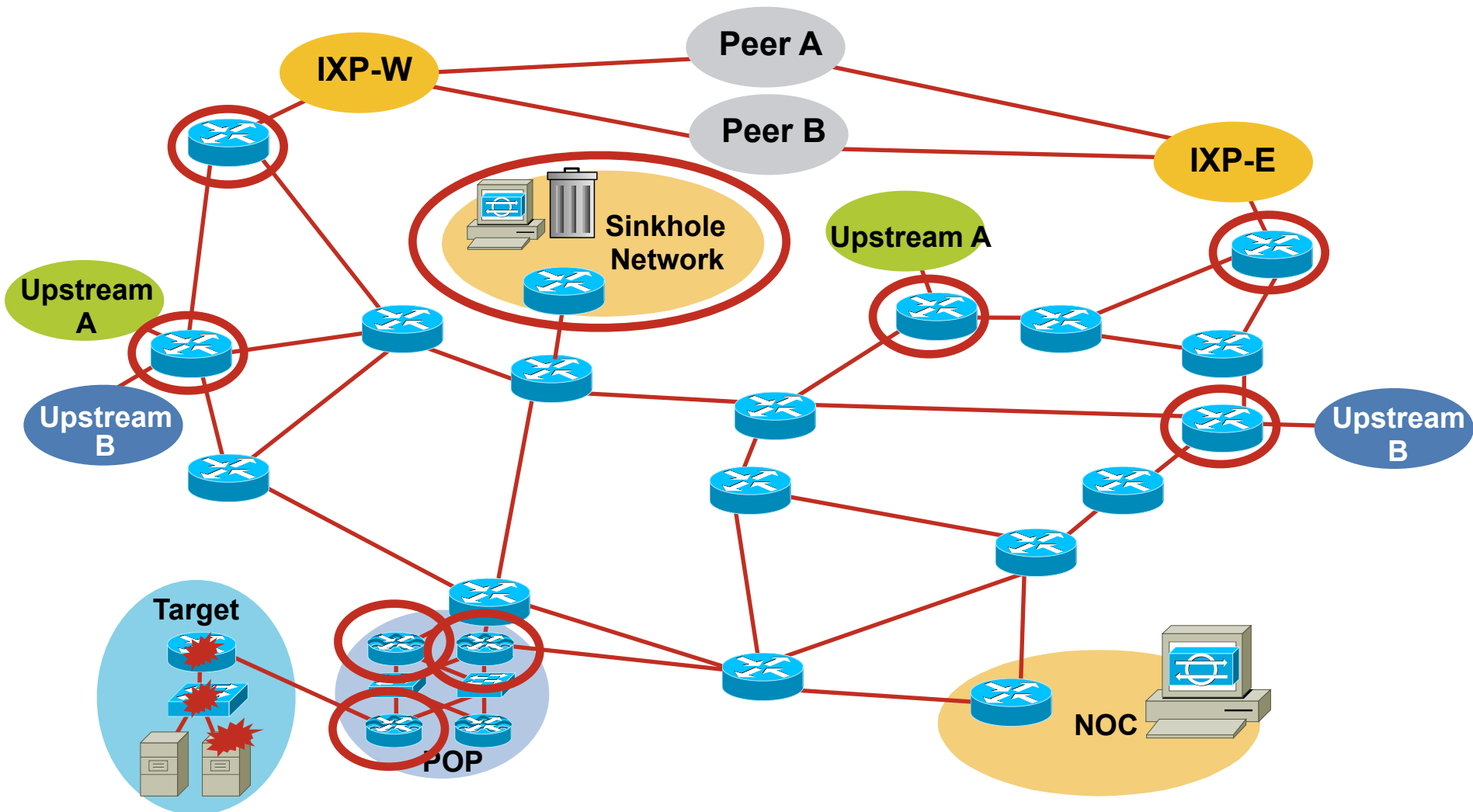
Various IPS technologies: NIDS, HIDS, anomaly detection

BGP triggers

Packet scrubbing

- Today, we will focus on core-centric tools

Where to React?



QoS at the Edge as Attack Mitigation

- Tag all ingress packets at the internet edge
- Doesn't require application or ip address awareness
- Provides proactive and reactive mitigation:

Proactively

Knocks down ToS 5-7

Can be added to CoPP ACL's:

```
access-list 152 permit tcp any any eq 22 dscp af13
```

Reactively

ACL's on the fly at internal chokepoints

Scavenger QoS, see:

Scavenger-Class QoS Strategy for DoS/Worm Attack Mitigation

http://www.cisco.com/application/pdf/en/us/guest/tech/tk759/c1482/cdccont_0900aecd80295ac7.pdf

QoS at the Edge as Attack Mitigation

- Configuration

```
class-map match-all edge-color
  match any
policy-map edge-color
  class edge-color
    set dscp af13

interface GigabitEthernet0/1
  service-policy input edge-color
```

- Considerations

- CPU impact - 3825 at 50,000 pps

- Without tagging 12% CPU

- With tagging 25% CPU

- Integration with existing QoS policy

- Treats all inbound traffic equally

- Differentiate responses to inside connections?

- Business critical inbound connections?

- Recolor ToS 6/7 instead?

Reacting with ACLs



Reacting to an Attack with ACLs

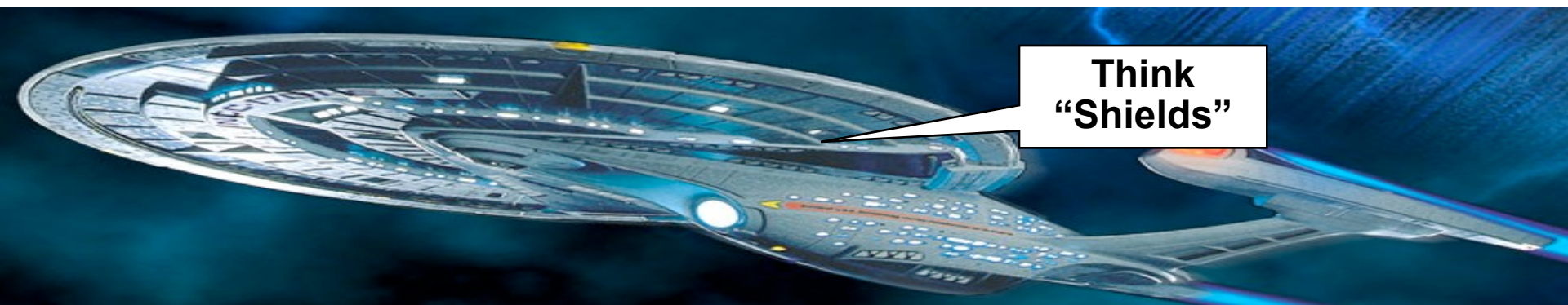
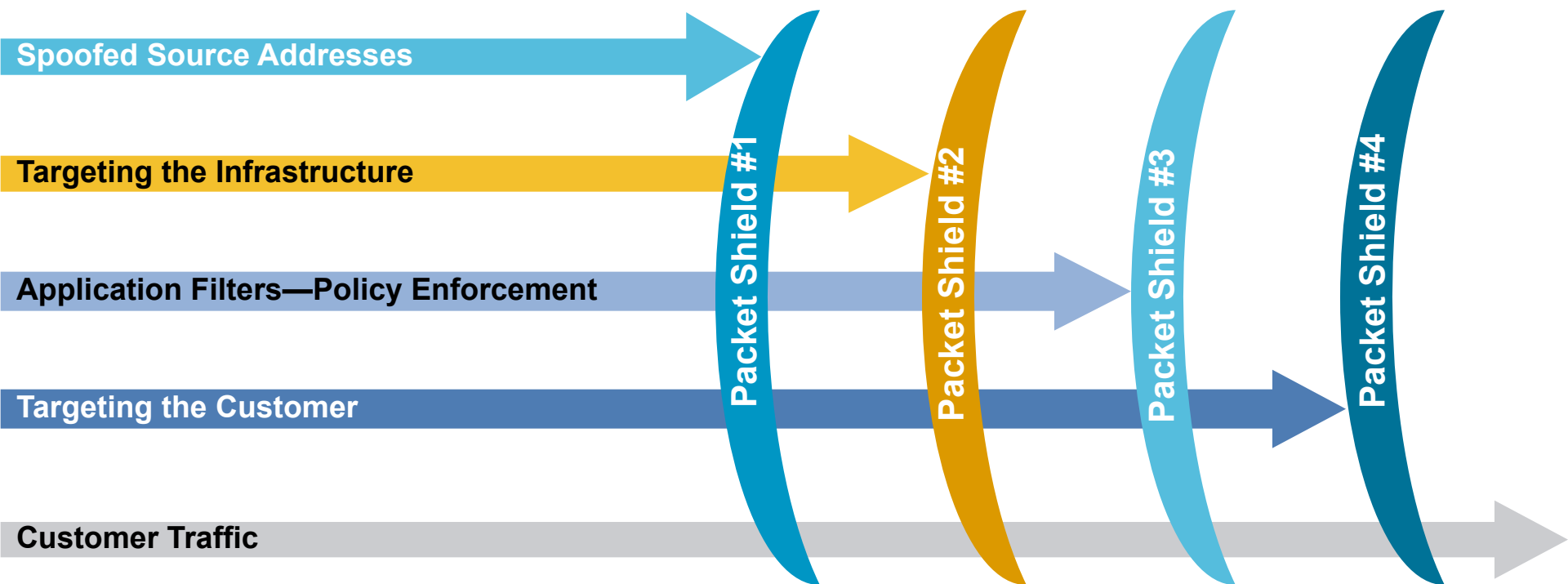
- Traditional method for stopping attacks
- Scaling issues encountered:
 - Operational difficulties
 - Changes on the fly
 - Multiple ACLs per interface
 - Performance concerns
- How does the ACL load into the router? Does it interrupt packet flow?
- How many ACEs can be supported in hardware?
In software?
- How does ACL depth impact performance?
- How do multiple concurrent features affect performance?

ACL Logging

- Access-list logging has significant impact on performance
- If you must “log” access-list entries, use "ip access-list logging interval"
- ACL drops line rate packets at 52% CPU utilization
- Add logging and CPU goes to 99%
- Add “ip access-list logging interval 1000”, CPU goes down to 60%

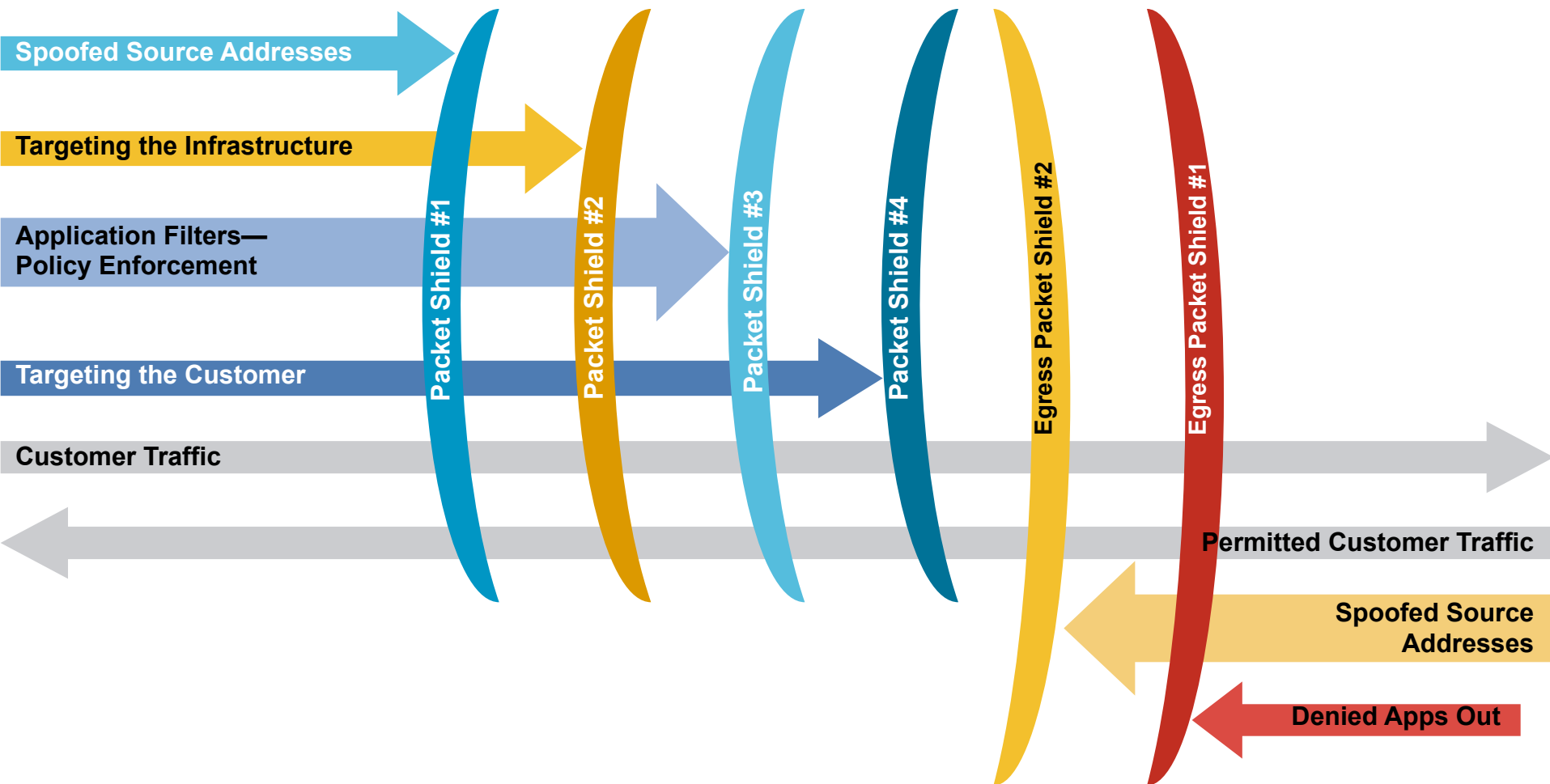
Packet Filtering

Viewed Horizontally



Packet Filtering

Remember to Filter the Return Path



ACL Construction

- Most common problem: poorly-constructed ACLs
- Scaling and maintainability issues with ACLs are commonplace
- Make your ACLs as modular and simple as possible: KISS
- Examples and best practices see:

Transit Access Control Lists: Filtering at Your Edge

http://www.cisco.com/en/US/tech/tk648/tk361/technologies_white_paper09186a00801afc76.shtml

Protecting Your Core: Infrastructure Protection ACLs

http://www.cisco.com/en/US/tech/tk648/tk361/technologies_white_paper09186a00801a1a55.shtml

ACL Categories

Explicit Permit	Explicit Deny
Anti-Spoofing	Anti-Spoofing
Anti-Bogon (Source)	Anti-Bogon (Source)
Infrastructure	Infrastructure
Incident Reaction	Incident Reaction
Explicit Permit L3	Explicit Deny L3
Explicit Permit L4	Explicit Deny L4
Explicit Deny Everything Else (Auditing)	Explicit Permit Everything Else (Auditing)

ACL Categories: Hybrid Philosophy

Hybrid Permit/Deny

- Anti-spoofing
- Anti-bogon (source)
- Infrastructure
- Explicit deny specific L3
- Explicit deny specific L4
- Incident reaction
- Explicit permit L3 (good traffic)
- Explicit permit L4 (good traffic)
- Explicit deny everything else (auditing)

ACL Maintenance: Frequency of Change

Hybrid Permit/Deny

Anti-spoofing	Rarely Changes
Anti-bogon (source)	Rarely Changes
Infrastructure	Rarely Changes
Explicit deny specific L3	Sometimes Changes
Explicit deny specific L4	Sometimes Changes
Incident reaction	Changes Everyday
Explicit permit L3 (good traffic)	Sometimes Changes
Explicit permit L4 (good traffic)	Sometimes Changes
Explicit deny everything else (auditing)	Rarely Changes

ACL Summary

- ACLs are widely deployed as a primary containment tool
- Prerequisites: identification and classification—need to know what to filter
- Apply as specific an ACL as possible
- ACLs are good for static attacks, not as effective for rapidly changing attack profiles
- Understand ACL performance limitations before an attack occurs

Reacting with BGP



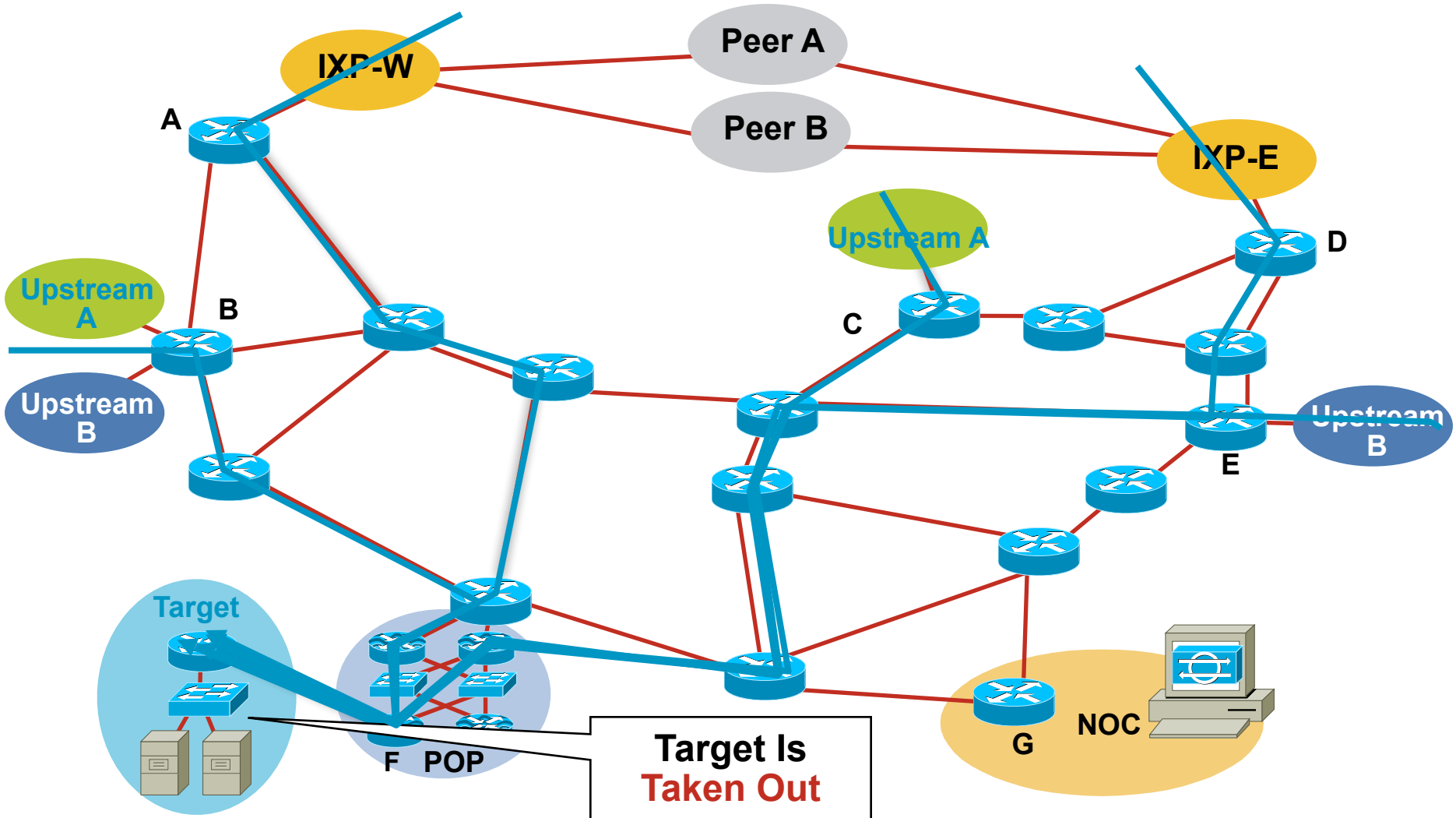
Blackhole Filtering

- **Blackhole Filtering** or **Blackhole Routing** forwards a packet to a router's **bit bucket**

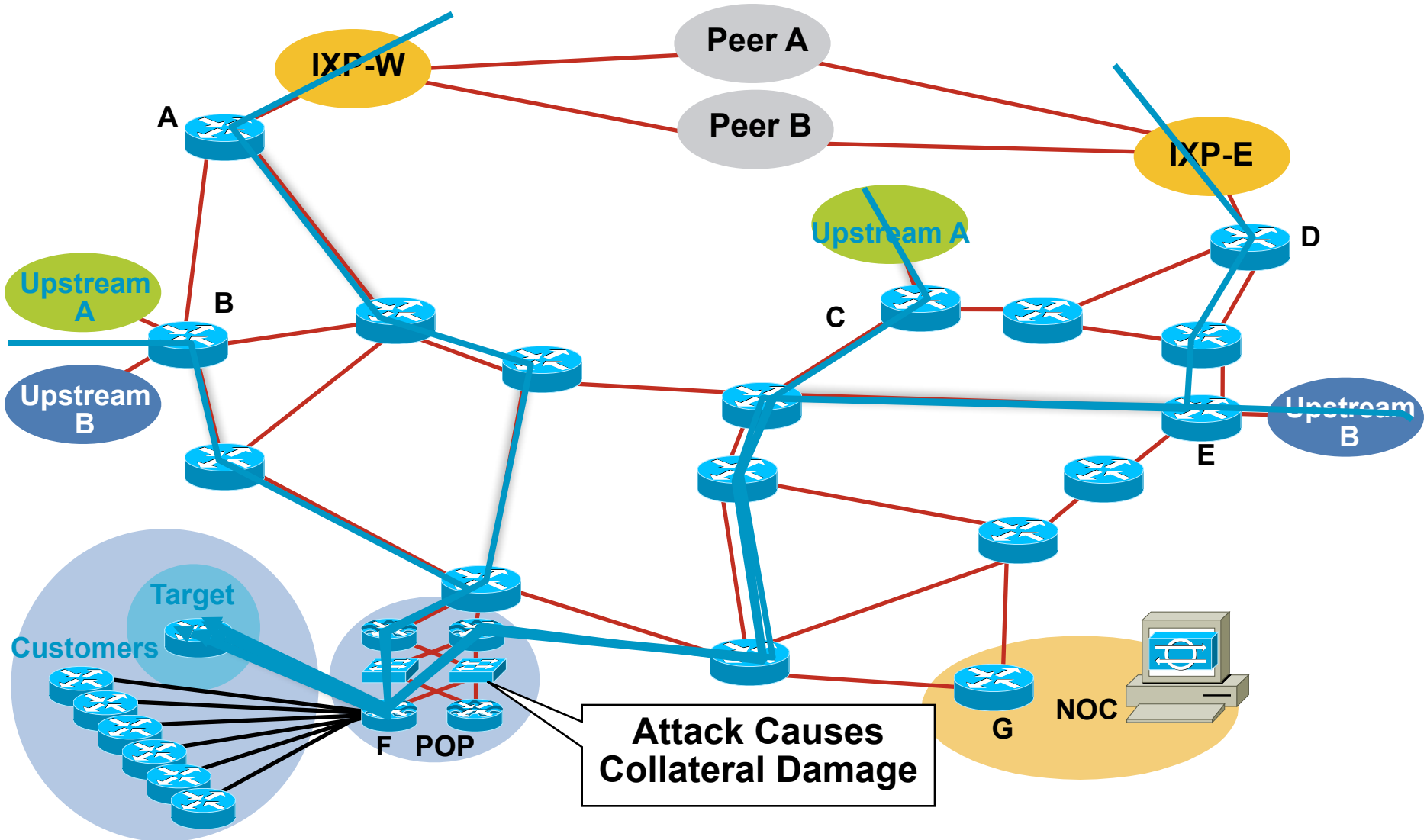
Also known as “route to Null0”

- Works only on destination addresses, since it is really part of the forwarding logic
- Forwarding ASICs are designed to work with routes to Null0—dropping the packet with minimal to no performance impact
- Used for years as a means to “blackhole” unwanted packets

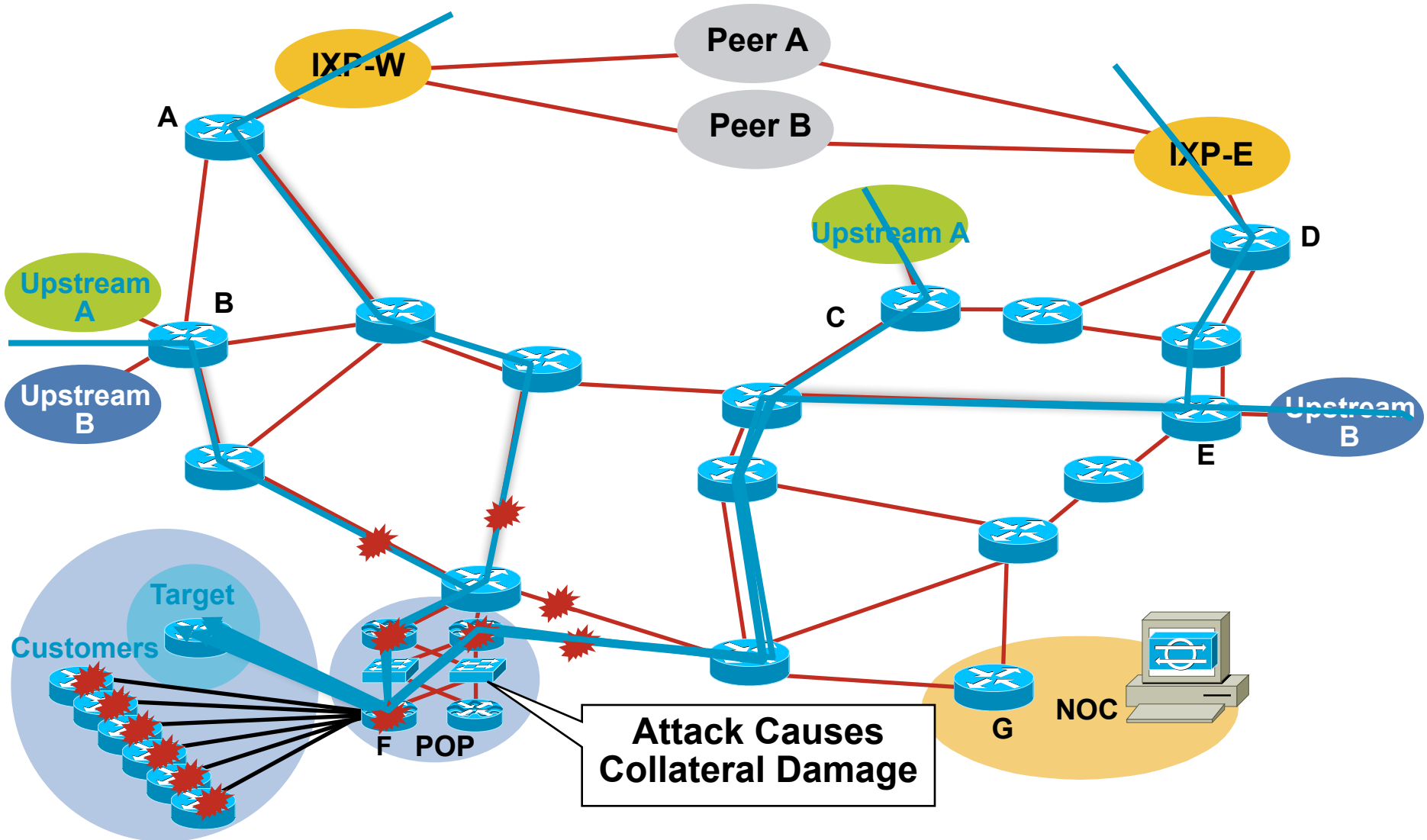
Customer Is DoSed: Before



Customer Is DoSed: Before— Collateral Damage



Customer Is DoSed: Before— Collateral Damage



Remotely Triggered Blackhole Filtering

- We will use BGP to trigger a networkwide response to an attack
- A simple static route and BGP will enable a networkwide destination address blackhole as fast as iBGP can update the network
- This provides a tool that can be used to respond to security related events and forms a foundation for other remote triggered uses
- Often referred to as RTBH

Remote Triggered Blackhole

- Configure all edge routers with static route to Null0 (must use “reserved” network)

```
ip route 192.0.2.1 255.255.255.255 Null0
```

- Configure trigger router

Part of iBGP mesh

Dedicated router recommended

- Activate blackhole

Redistribute host route for victim into BGP with next-hop set to 192.0.2.1

Route is propagated using BGP to all BGP speaker and installed on routers with 192.0.2.1 route

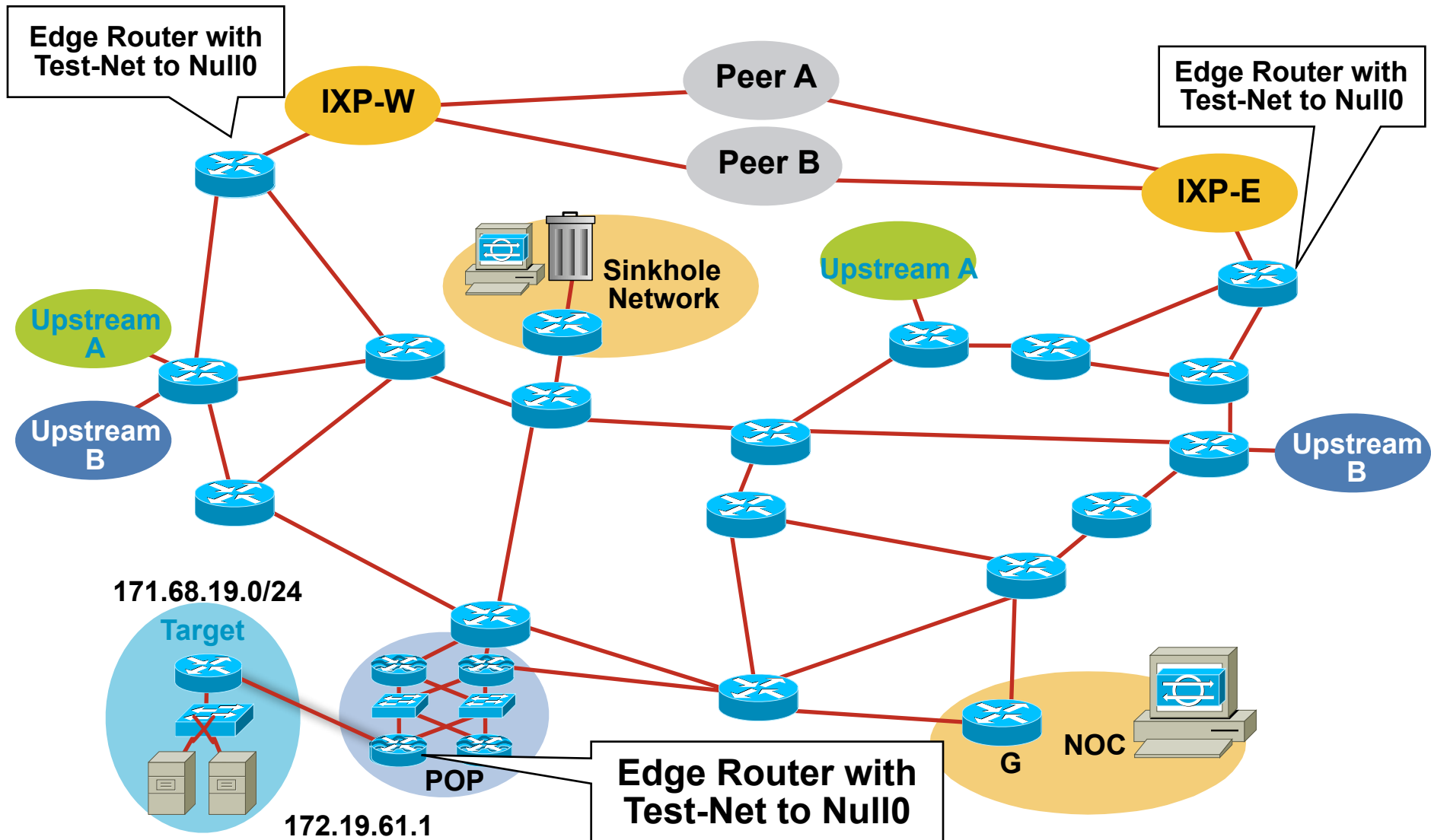
All traffic to victim now sent to Null0

Step 1: Prepare All the Routers With Trigger

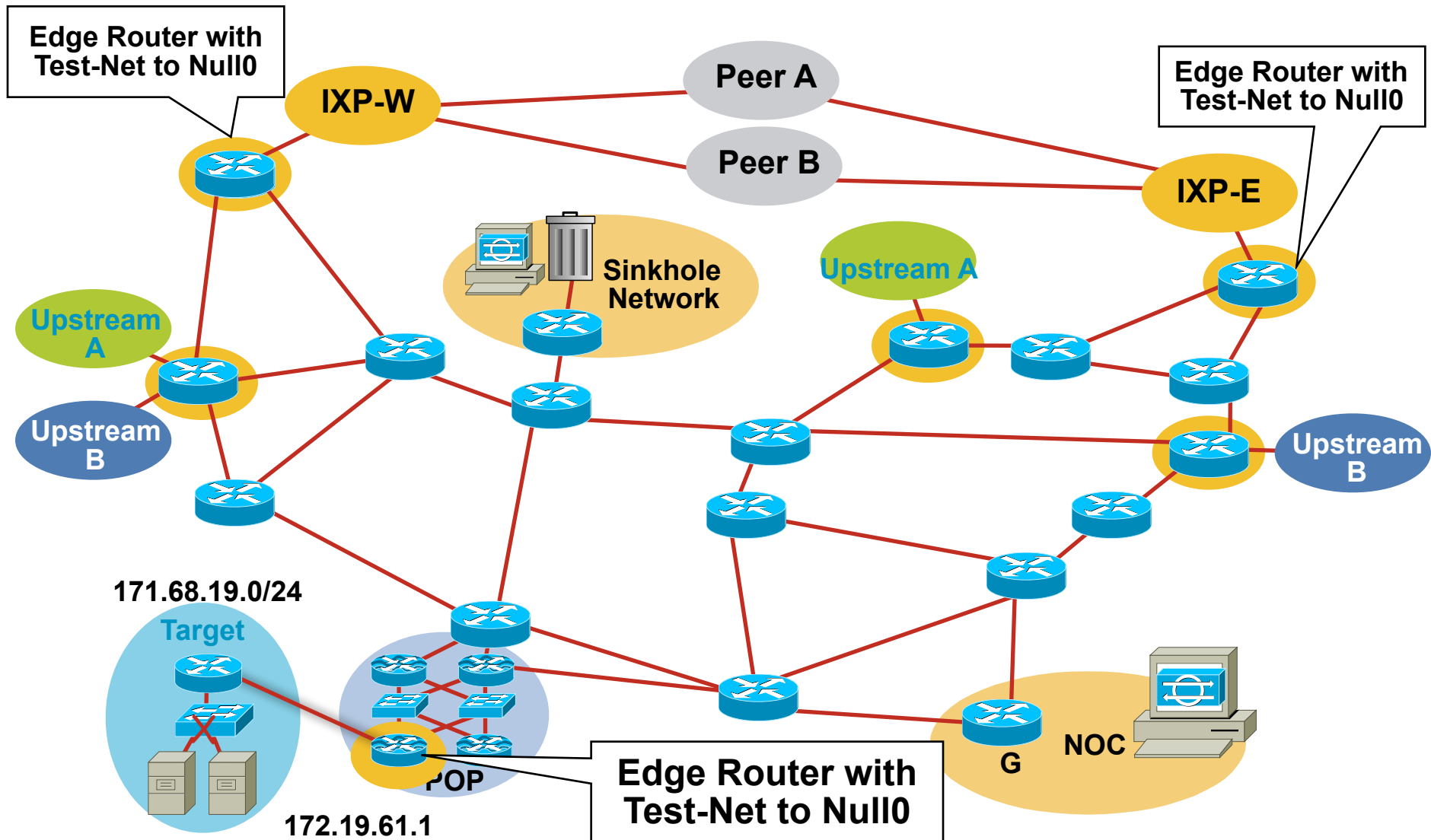
- Select a small block that will not be used for anything other than blackhole filtering; test Net (192.0.2.0/24) is optimal since it should not be in use
- Put a static route with a /32 from Test-Net—192.0.2.0/24 to Null 0 on every edge router on the network

```
ip route 192.0.2.1 255.255.255.255 Null0
```


Step 1: Prepare All the Routers With Trigger



Step 1: Prepare All the Routers With Trigger



Step 2: Prepare the Trigger Router

The Trigger Router Is the Device that Will Inject the iBGP Announcement into the ISP's Network

- Should be part of the iBGP mesh—but does not have to accept routes
- Can be a separate router (recommended)
- Can be a production router
- Can be a workstation with Zebra/Quagga (interface with Perl scripts and other tools)

Trigger Router's Configuration

Redistribute Static with a Route-Map

```
router bgp 65535
.
redistribute static route-map static-to-bgp
.
```

Match Static Route Tag

```
!
route-map static-to-bgp permit 10
```

```
match tag 66
```

```
set ip next-hop 192.0.2.1
```

```
set local-preference 200
```

```
set community no-export
```

```
set origin igp
```

```
!
```

Set Next-Hop to the Trigger

Set Local-Pref

```
Route-map static-to-bgp permit 20
```

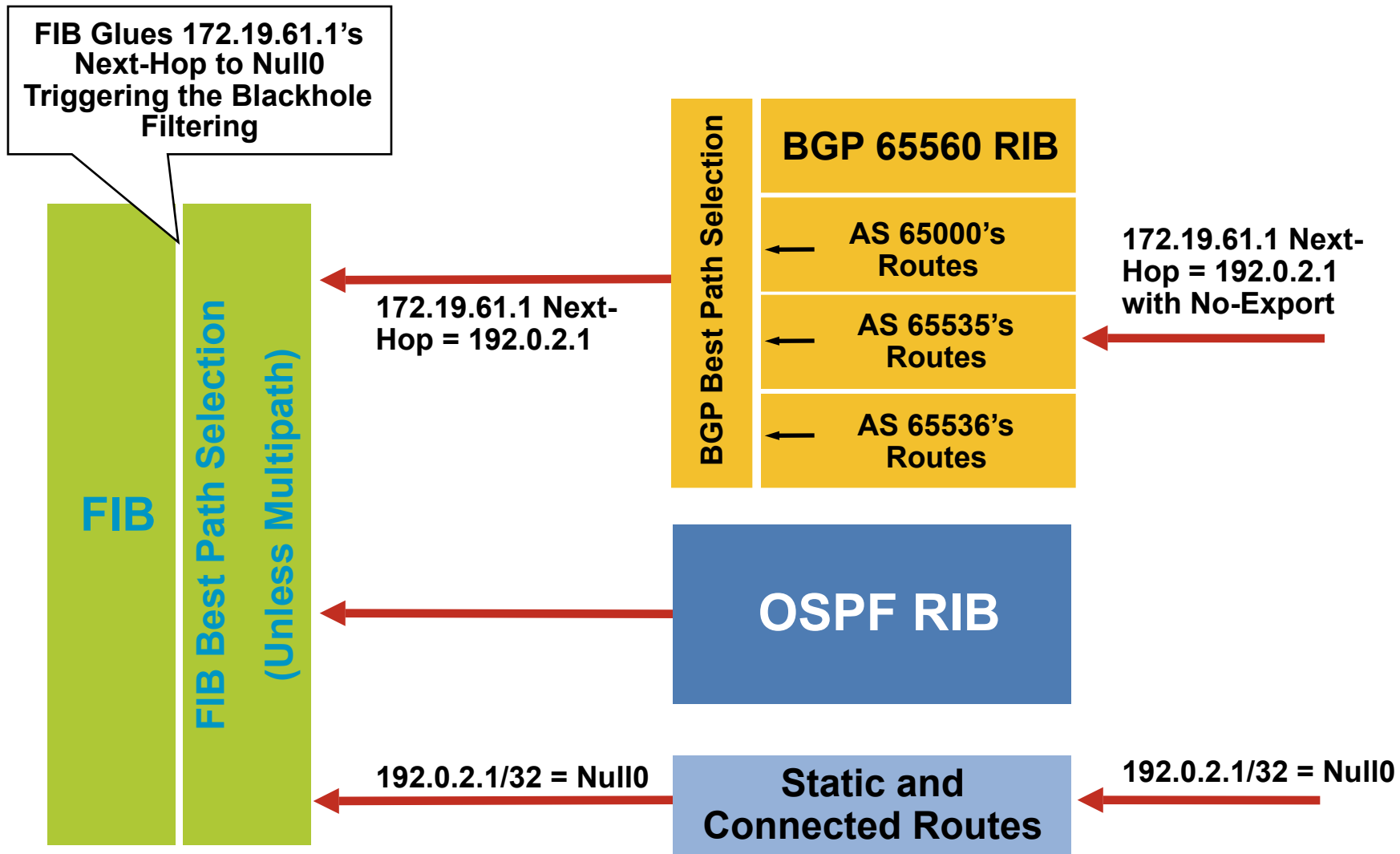
Step 3: Activate the Blackhole

- Add a static route to the destination to be blackholed; the static is added with the “tag 66” to keep it separate from other statics on the router

```
ip route 172.19.61.1 255.255.255.255 Null0 Tag 66
```

- BGP advertisement goes out to all BGP speaking routers
- Routers received BGP update, and “glue” it to the existing static route; due to recursion, the next-hop is now Null0

Step 3: Activate the Blackhole



Step 3: Activate the Blackhole

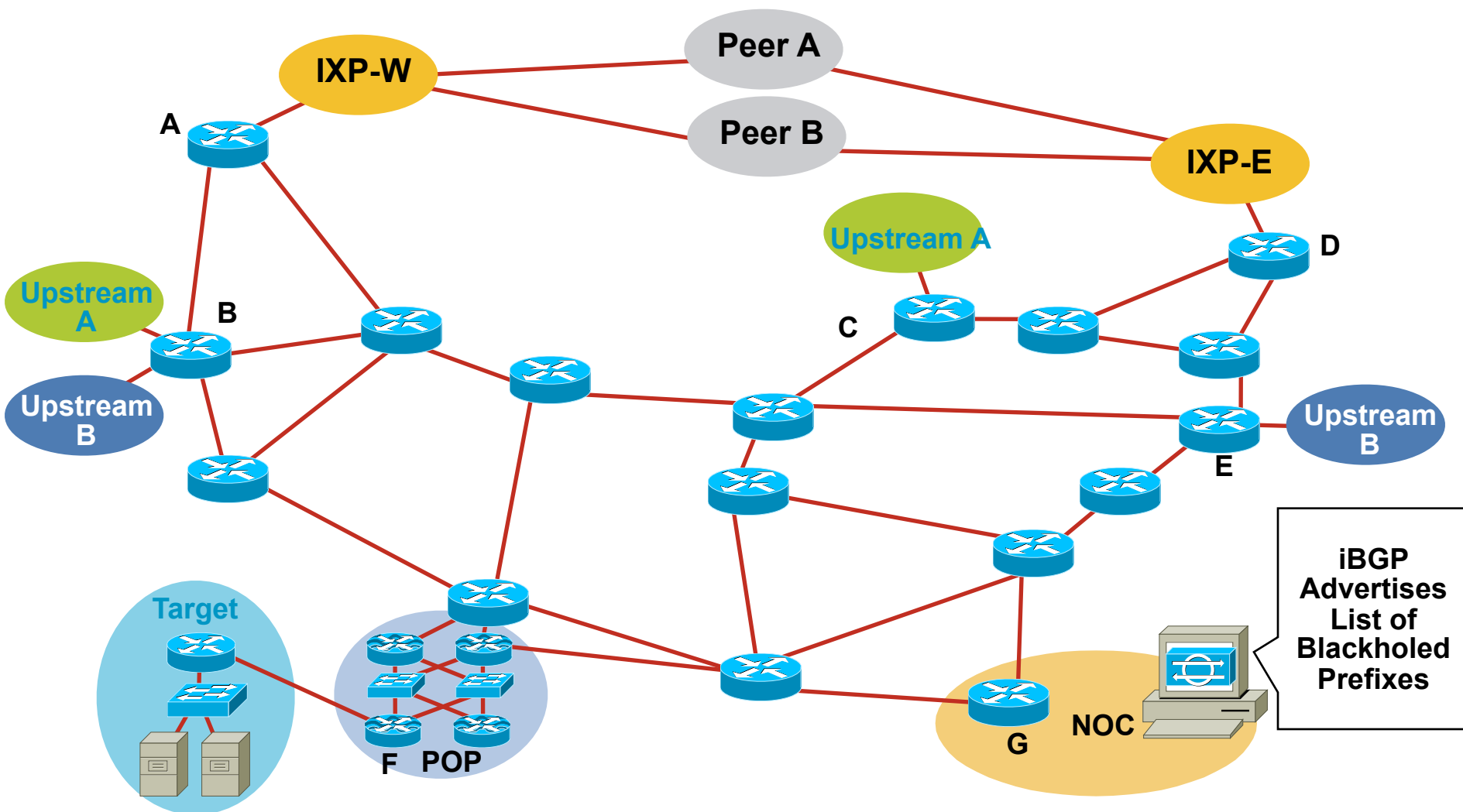
BGP Sent—172.19.61.1 Next-Hop = 192.0.2.1

Static Route in Edge Router—192.0.2.1 = Null0

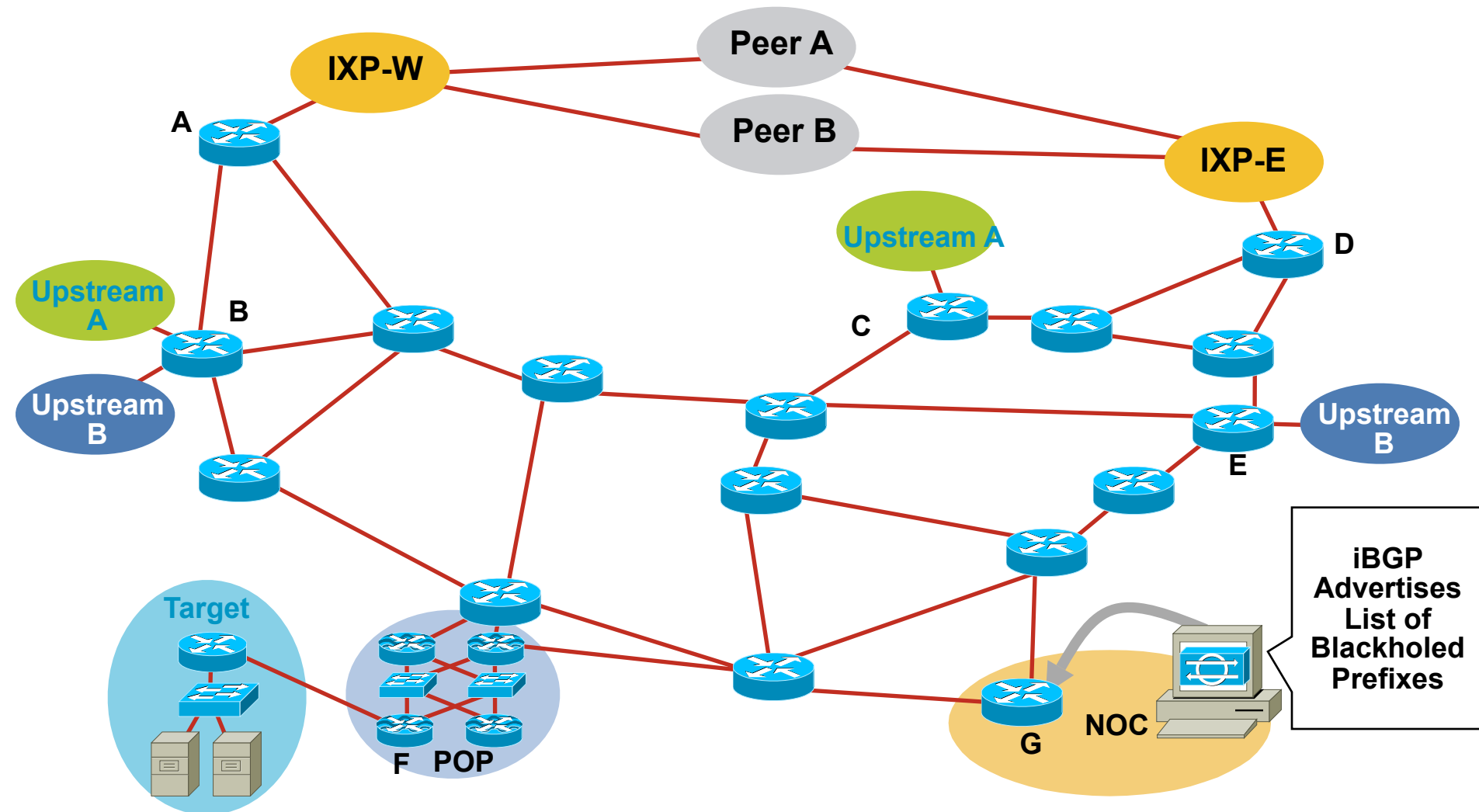
172.19.61.1 = 192.0.2.1 = Null0

**Next-Hop of 172.19.61.1
Is Now Equal to Null0**

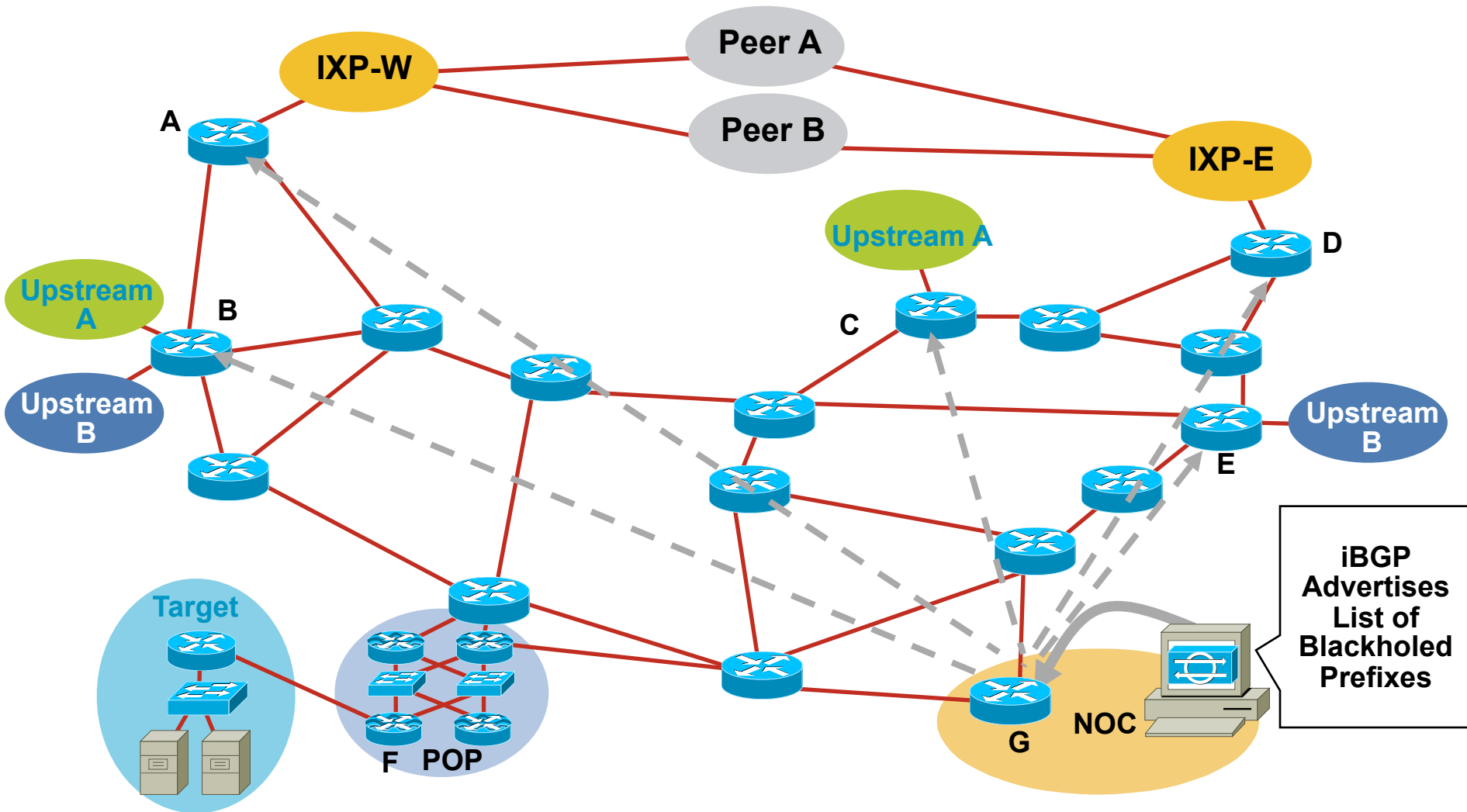
Step 3: Activate the Blackhole



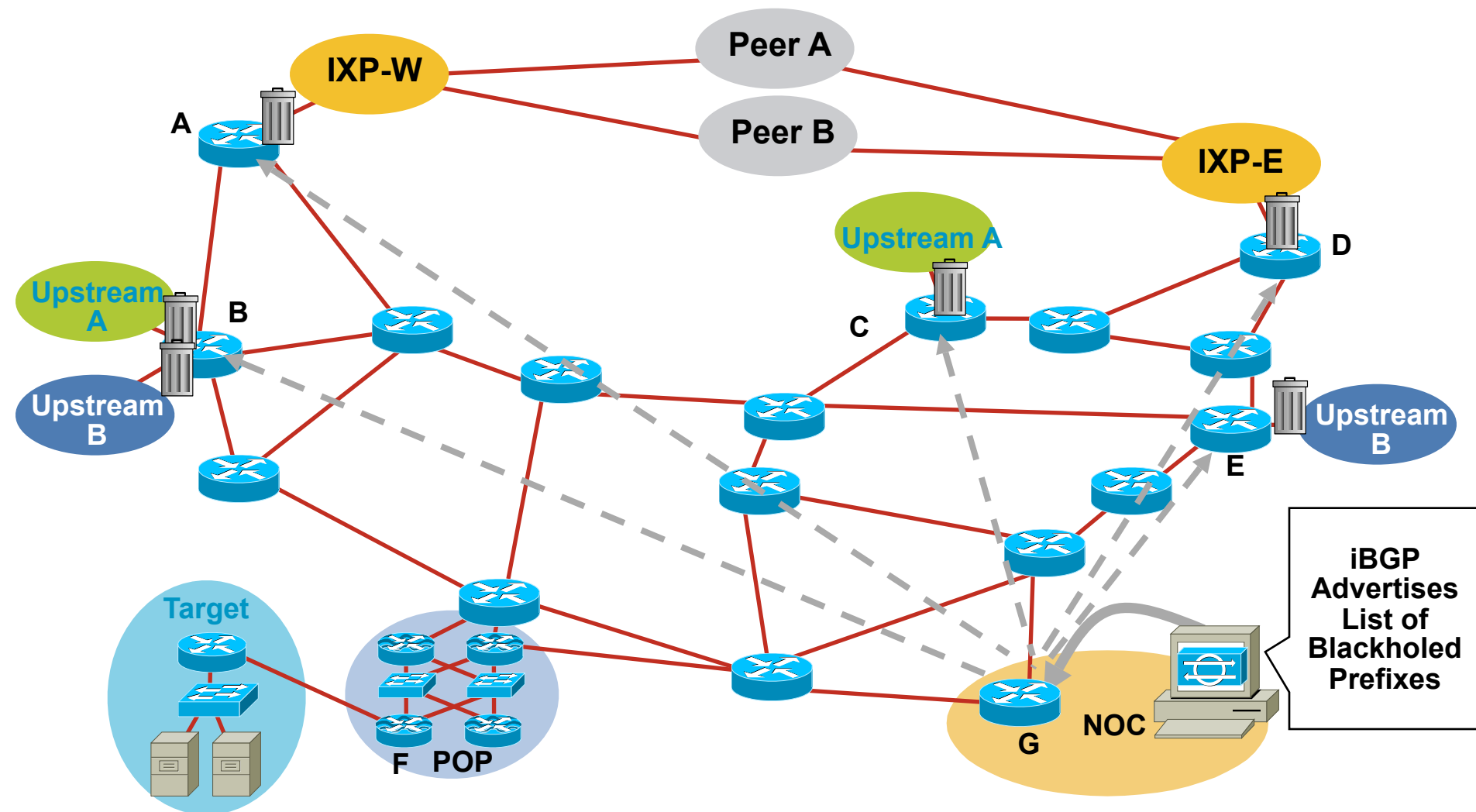
Step 3: Activate the Blackhole



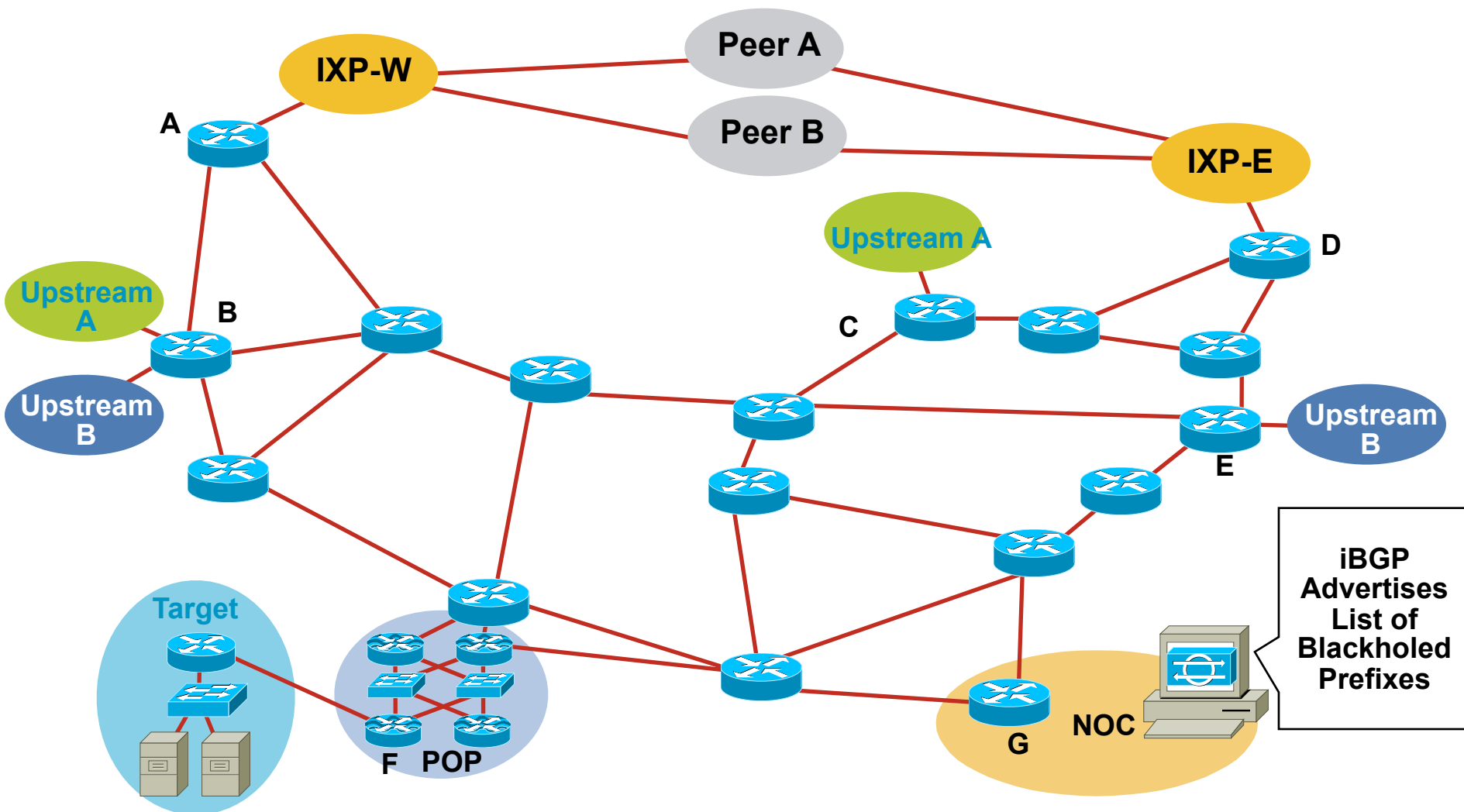
Step 3: Activate the Blackhole



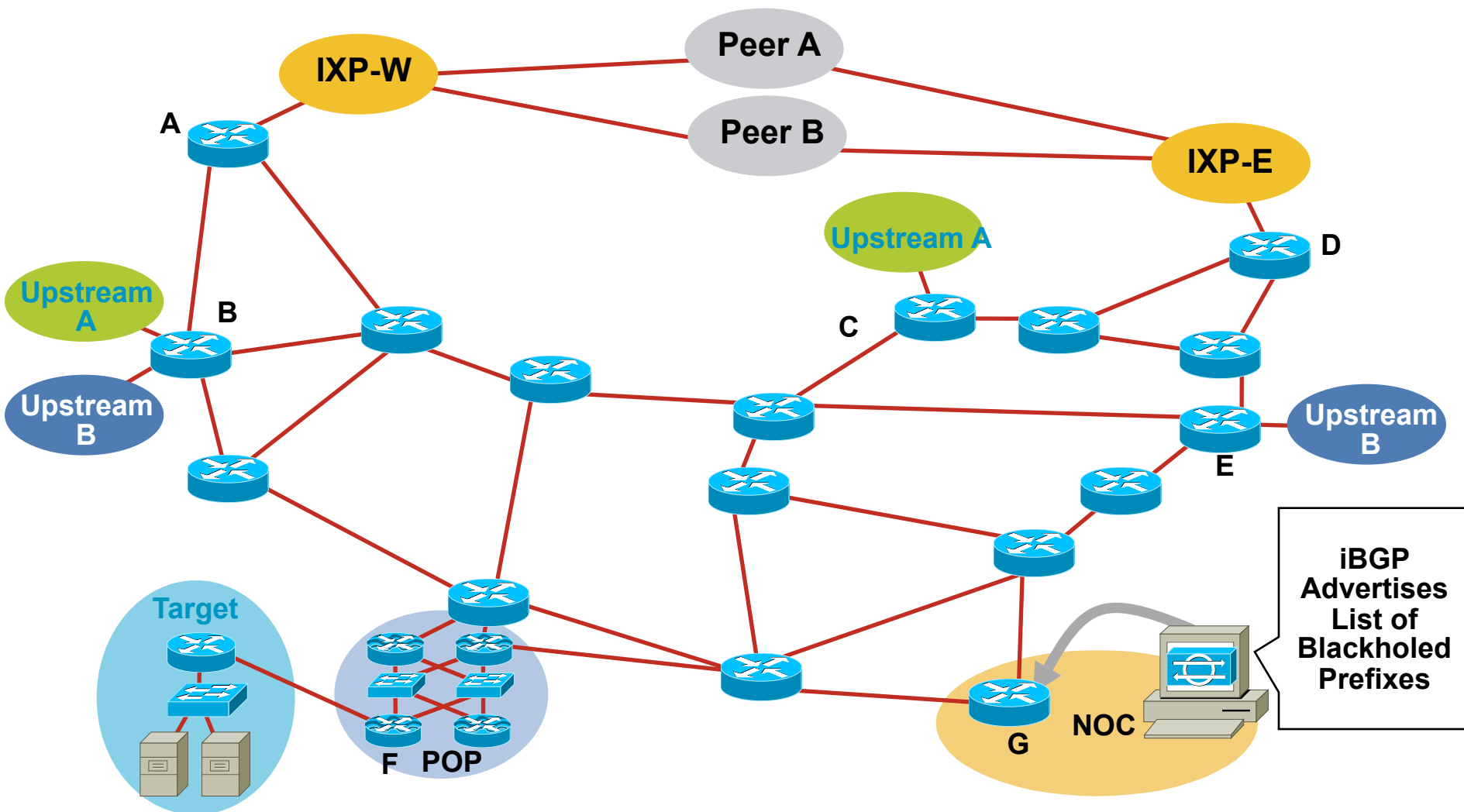
Step 3: Activate the Blackhole



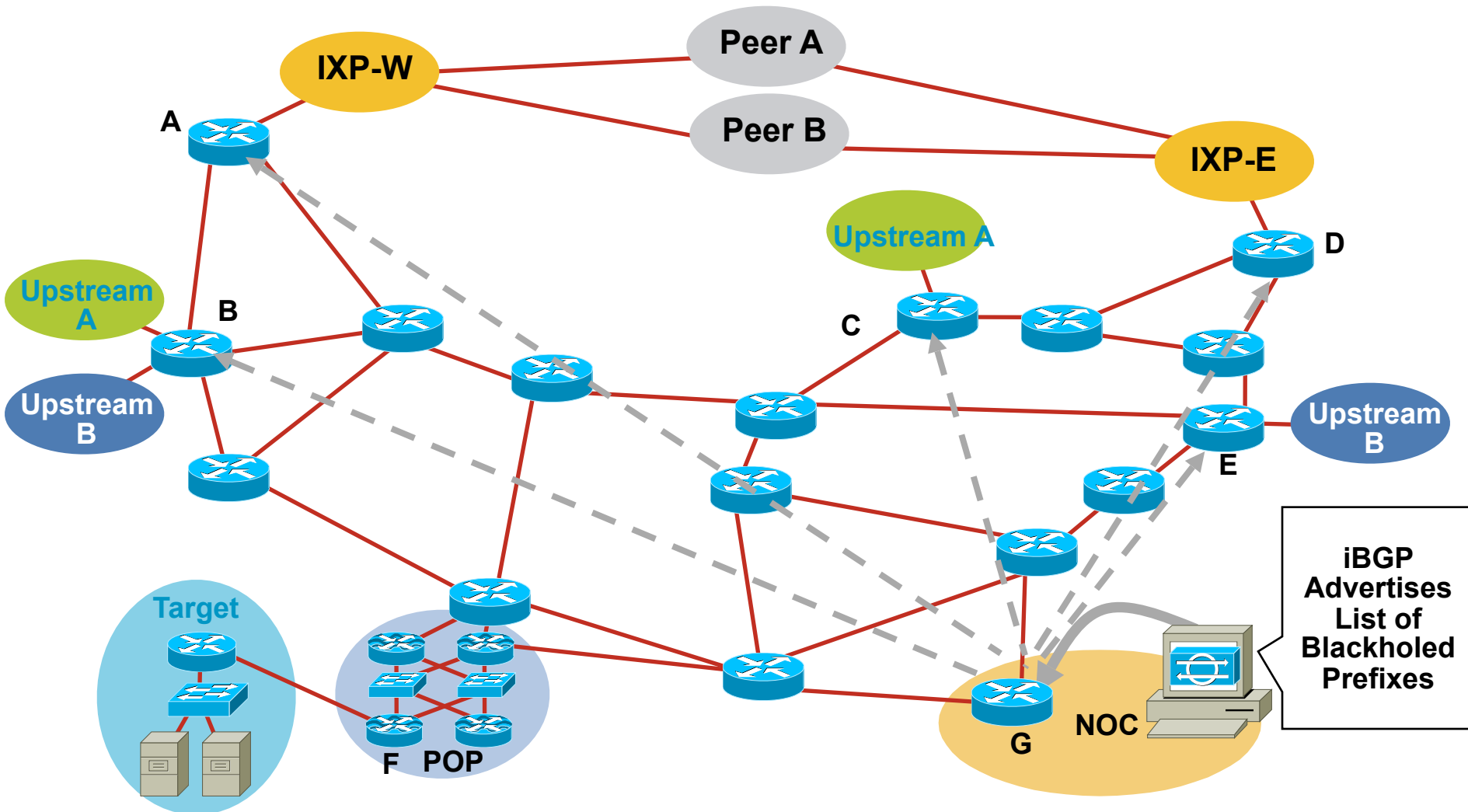
Customer Is DoSed: After— Packet Drops Pushed to the Edge



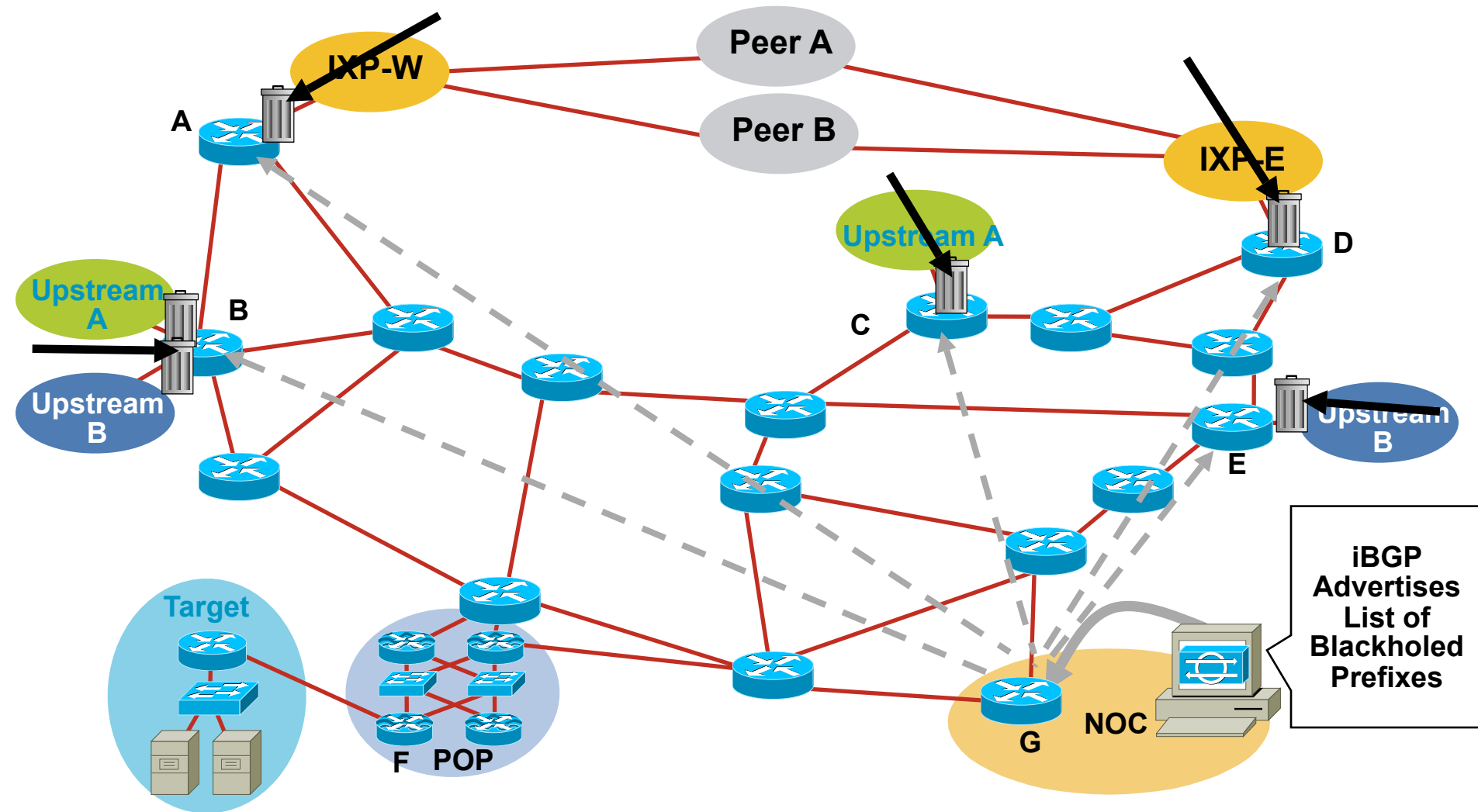
Customer Is DoSed: After— Packet Drops Pushed to the Edge



Customer Is DoSed: After— Packet Drops Pushed to the Edge



Customer Is DoSed: After— Packet Drops Pushed to the Edge



Using Remote Triggered Blackhole

- Is this done today?

Yes, service providers and enterprises use frequently

- Often only scaleable answer to large-scale DoS attack

Has proven very effective

- Interprovider triggers not implemented

Rely on informal channels

- **Service: customer triggered**

Edge customers trigger the update, SP doesn't get involved

Implication: you detect, you classify, etc.

- White list allowed traffic to prevent self-DoS

<http://www.cymru.com/gillsr/documents/golden-networks>

BGP Sinkhole Trigger

- Leverage the same BGP technique used for RTBH
- Dedicated trigger router redistributes more specific route for destination being re-rerouted

Next-hop set via route-map

- All BGP-speaking routers receive update
- Complex design can use multiple route-maps and next-hops to provide very flexible designs
- May require BGP on all routers

Example: BGP Sinkhole Triggers

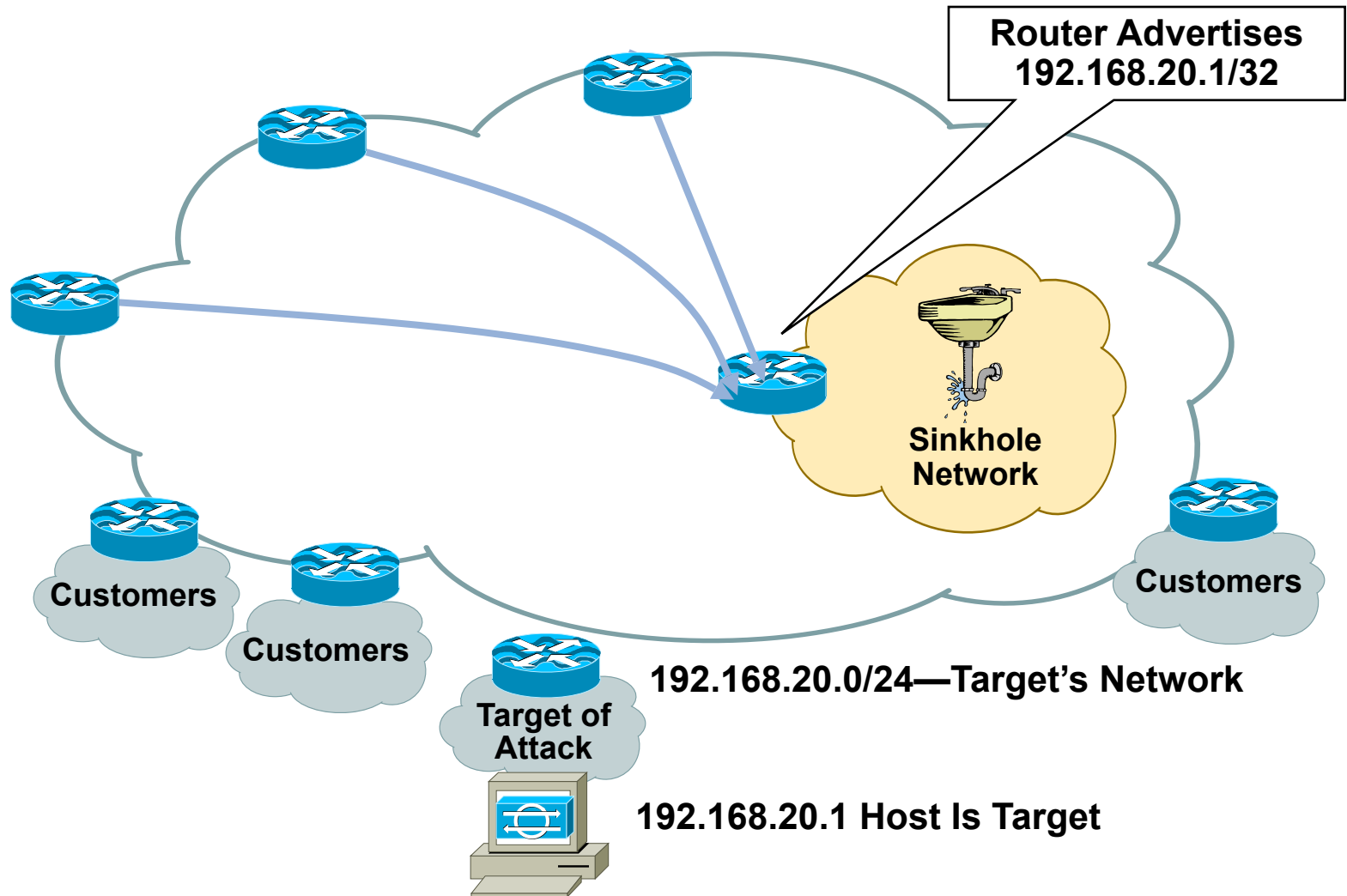
- Sinkhole IP: 192.0.2.8
- Victim IP: 192.168.20.1
- Trigger router configuration

```
router bgp 100
 redistribute static route-map static-to-bgp

route-map static-to-bgp permit 10
 match tag 66
 set origin igp
 set next-hop 192.0.2.8  <-- sinkhole address, not Null0
 set community NO-EXPORT

ip route 192.168.20.1 255.255.255.255 Null0 tag 66
```

Sinkhole Routers/Networks



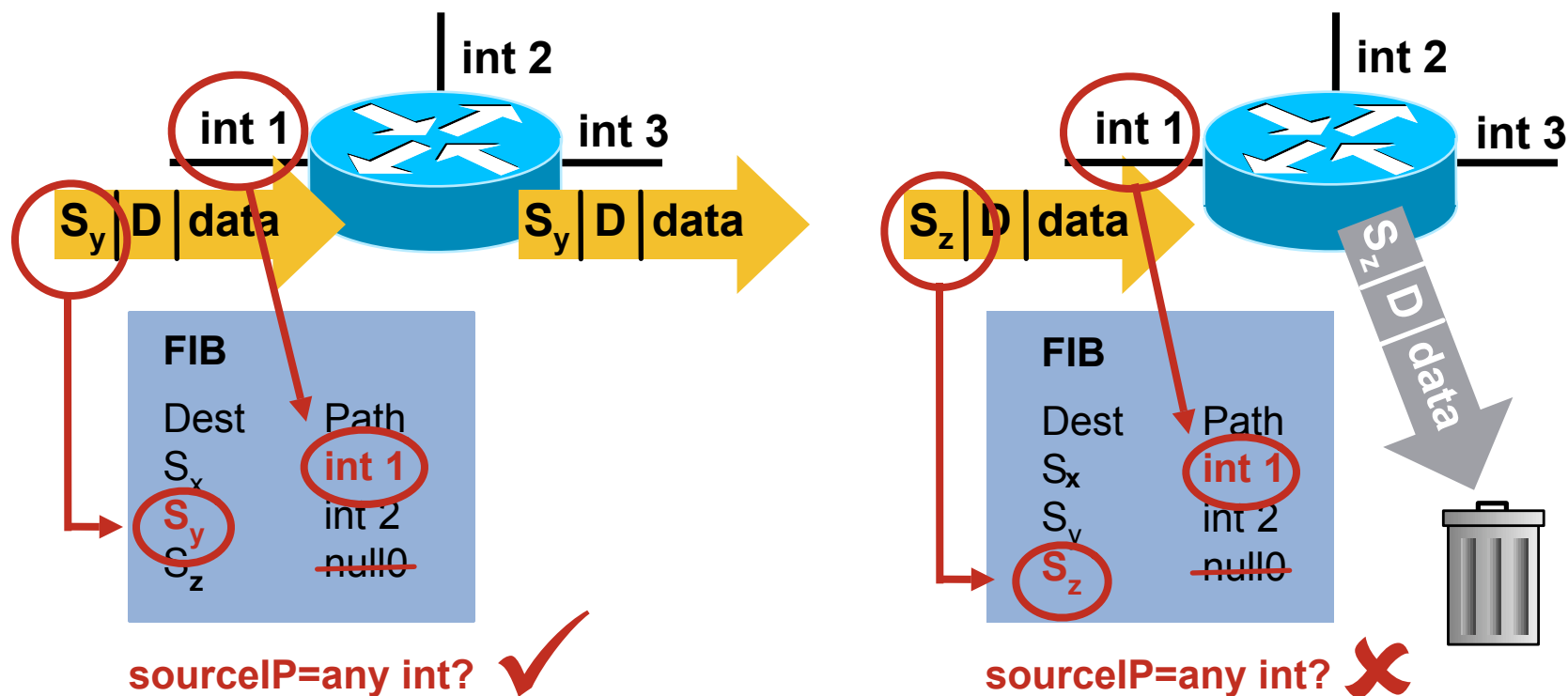
Flipping RTBH Around

Triggered Source Drops

- Dropping on destination is very important
 - Dropping on source is often what we really need
- Reacting using source address provides some interesting options:
 - Stop the attack without taking the destination offline
 - Filter command and control servers
 - Filter (contain) infected end stations
- Must be rapid and scalable
 - Leverage pervasive BGP again

Quick Review: uRPF—Loose Mode

`router(config-if)# ip verify unicast source reachable-via any`



IP Verify Unicast Source Reachable—Via any

Source-Based Remote Triggered Blackhole Filtering

Uses the Same Architecture as Destination-Based Filtering + Unicast RPF

- Edge routers must have static in place
- They also require Unicast RPF
- BGP trigger sets next hop—in this case the “victim” is the source we want to drop

Source-Based Remote Triggered Blackhole Filtering

- What do we have?

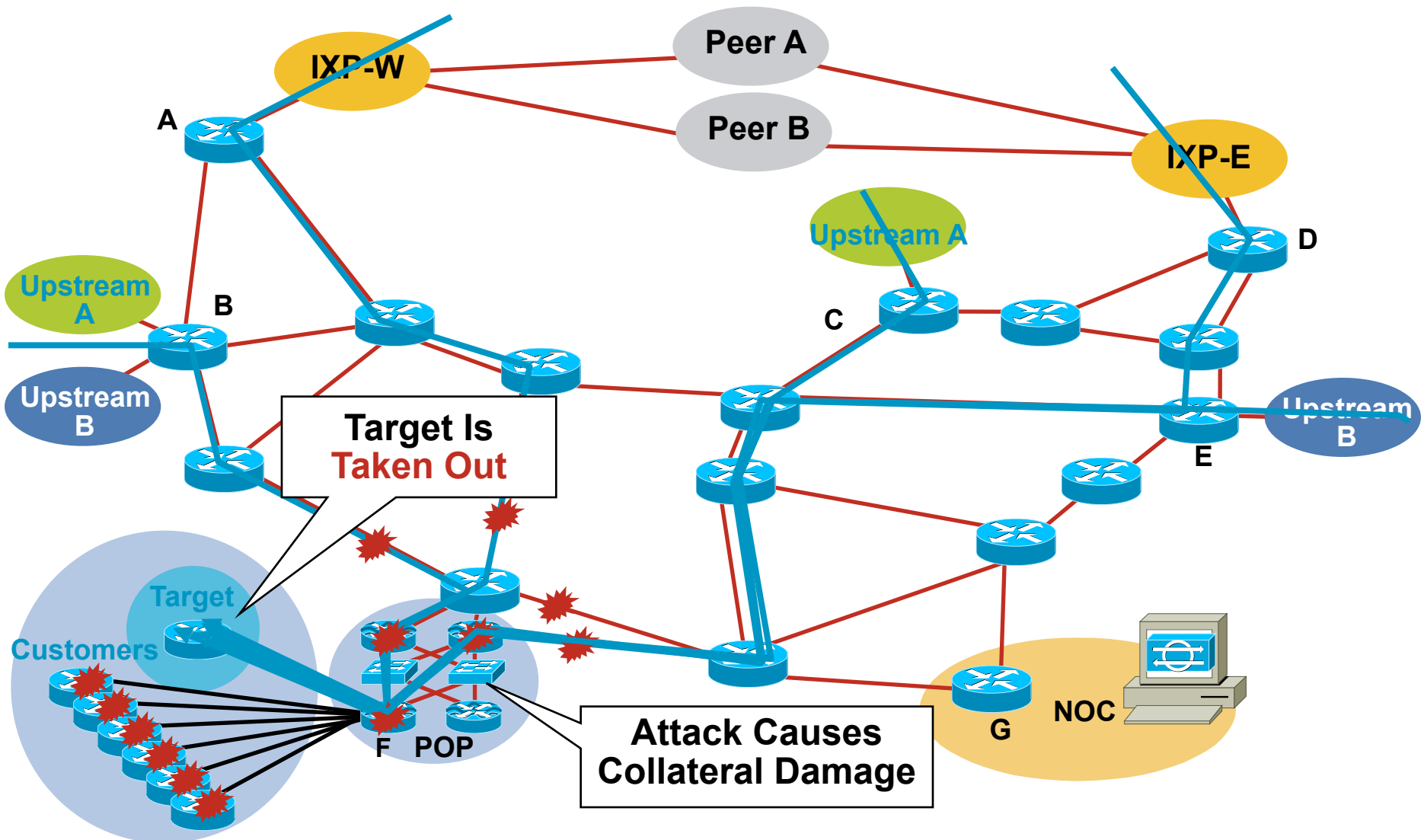
Blackhole Filtering—if the **destination** address equals Null0, we drop the packet

Remote Triggered—trigger a prefix to equal Null0 on routers across the Network at iBGP speeds

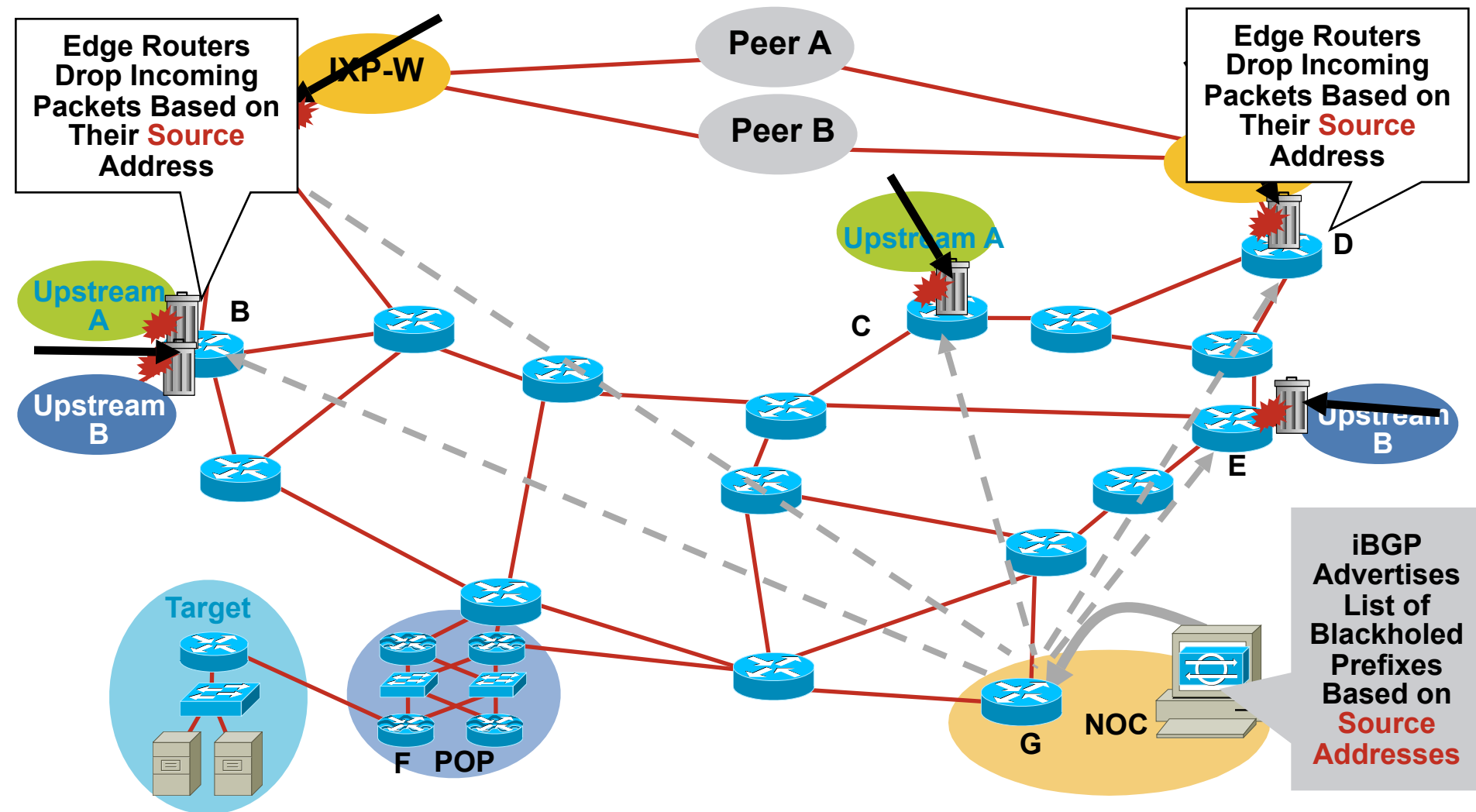
uRPF Loose Check—if the **source** address equals Null0, we drop the packet

- Put them together and we have a tool to trigger drop for any packet coming into the network whose source or destination equals Null0

Customer Is DoSed: Before

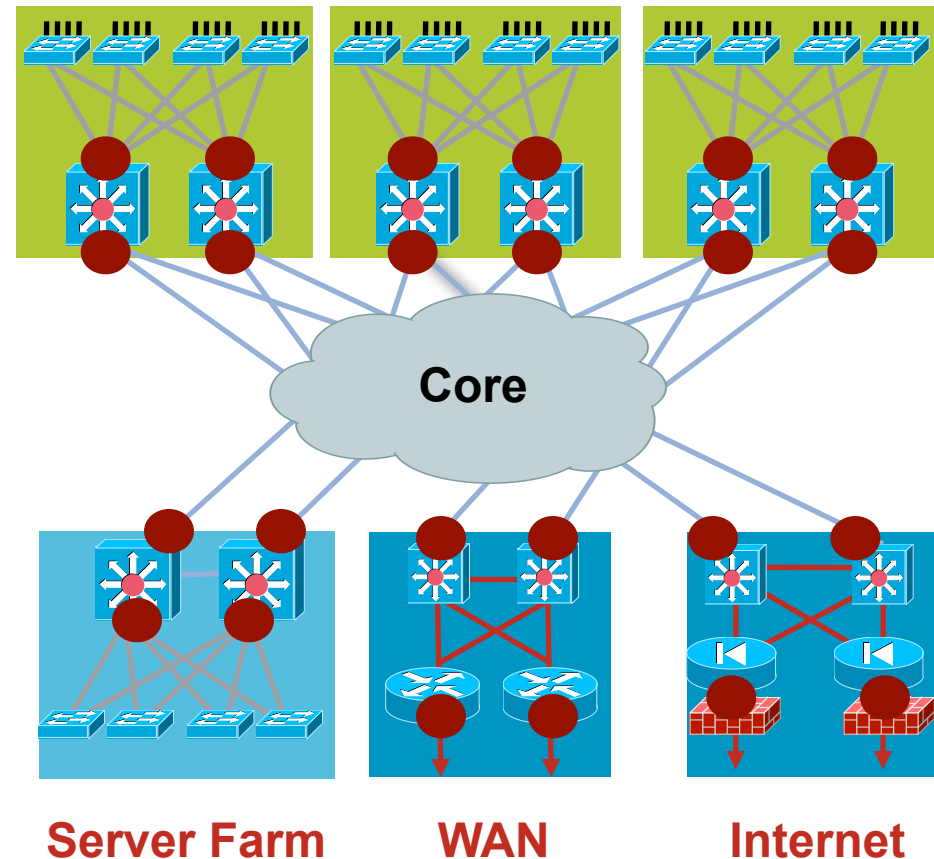


Customer Is DoSed: After Packet Drops Pushed to the Edge



Possible Remote Trigger Placement

- Dark red dots indicate possible remote drop location
- L3 boundaries between network components
 - Drop infections at distribution layer
 - Drop incoming Internet attack at Internet edge
 - React to incoming attacks from remote office across the WAN
 - etc.



Community-Based Trigger

- BGP community-based triggering allow for more fined tuned control over where you drop the packets
- Three parts to the trigger:
 - Static routes to Null0 on all the routers
 - Trigger router sets the community
 - Reaction routers (on the edge) matches community and sets the next-hop to the static route to Null0

Why Community-Based Triggering?

Allows for More Control on the Attack Reaction

- Trigger community #1 can be for all routers in the network
- Trigger community #2 can be for all peering routers; no customer routers - allows for customers to talk to the DoSed customer within your AS
- Trigger community #3 can be for all customers; used to push a inter-AS traceback to the edge of your network
- Trigger communities per ISP Peer can be used to only blackhole on one ISP Peer's connection; allows for the DoSed customer to have partial service

BGP: Not Just For Routing, Anymore

- “I don’t want to use BGP as a routing protocol”

Think of BGP as a signaling protocol

Routing protocols operate as “ships in the night”

- BGP has a unique property among routing protocols: arbitrary next hops can be administratively defined

- There is no need to actually carry routes in BGP

Deploy iBGP mesh internally and do not use it for routing

Under normal conditions, BGP holds zero routes

When used for drops, only the blackholed addresses are in the table

- If BGP is used for inter-region routing, drop boundaries can be both local within a campus and global

Use communities to “scope” the drops

Internal Source-Based Drops

- Both source and destination drops can be used internally

Source drops likely the most interesting case

Destination drops still result in target DoS

Don't forget the Internet and WAN edges

- Provides a very effective mechanism to handle internal attacks

Drop worm infected PCs off the network

Drop “owned” devices off the network

Protect the infrastructure

Whitelist to prevent self DoS

Source-Based RTBH

Key Advantages

- No ACL update
- No change to the router's configuration
- Drops happen in the forwarding path
- Frequent changes when attacks are dynamic (for multiple attacks on multiple customers)

What If I Can't Deploy RTBH?

- Start with uRPF and static routes to NULL0
- Results in traffic source drops

```
interface g0/0
  ip verify unicast source reachable-via rx allow-default
ip route 10.0.0.0 255.0.0.0 Null0
ip route 169.254.0.0 255.255.0.0 Null0
ip route 172.16.0.0 255.240.0.0 Null0
ip route 192.0.2.0 255.255.255.0 Null0
ip route 192.168.0.0 255.255.0.0 Null0
```

- For example, traffic **from** 10.1.1.1 will be discarded
- Can be deployed in reaction to attacks
- A start but... won't be fast and doesn't scale

ACLs or uRPF Remote-Triggered Drop?

- ACLs key strengths:
 - Detailed packet filtering (ports, protocols, ranges, fragments, etc.)
 - Relatively static filtering environment
 - Clear filtering policy
- ACLs can have issues when faced with:
 - Dynamic attack profiles (different sources, different entry points, etc.)
 - Frequent changes
 - Quick, simultaneous deployment on a multitude of devices
- Combining ACLs with uRPF remote-triggered drops allows for ACLs to handle the strict static policies while uRPF remote-triggered blackhole handles the dynamic source-based drops

Recap



Topics Covered

- Introduction to Core Security

 - Denial of Service (DoS) and Worm Review

 - Six-Phase Methodology

- Device Best Practices

 - Router Security

 - Routing Protocol Security

 - Planes, Paths, and Punts

 - Receive ACL

 - Control Plane Policing

 - Control Plane Protection

 - Management Plane Protection

Topics Covered (Cont.)

- Infrastructure Security

 - RFC 2827/BCP 38

 - Infrastructure ACLs

 - Flexible Packet Matching

- Network Telemetry

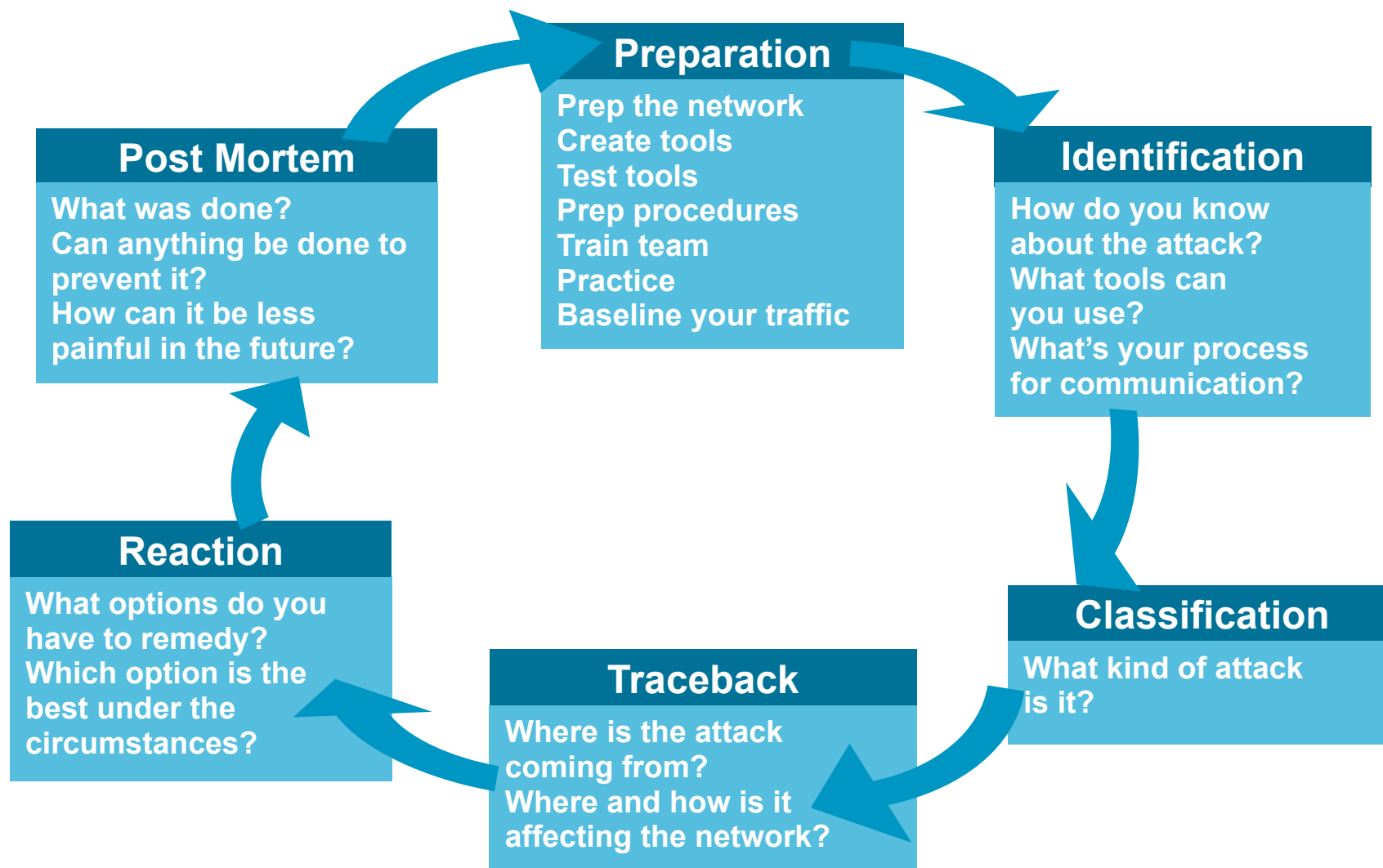
 - SNMP, RMON and Their ilk

 - NetFlow for Security Purposes

Topics Covered (Cont.)

- Traceback Techniques
 - NetFlow Traceback Techniques
 - Attract and Analyze: Sinkholes
- Reacting to Attacks
 - Reacting with ACL
 - Reacting with BGP
 - Packet Scrubbing

Six Phases of Incident Response



References



References

- DoS detection:

“Tackling Network DoS on Transit Networks”: David Harmelin, DANTE, March 2001

<http://www.dante.net/pubs/dip/42/42.html>

“Inferring Internet Denial-of-Service Activity”: David Moore et al, May 2001

<http://www.caida.org/outreach/papers/2001/BackScatter/usenixsecurity01.pdf>

“The Spread of the Code Red Worm”: David Moore, CAIDA, July 2001

<http://www.caida.org/analysis/security/code-red/>

- DoS tracing:

“Tracing Spoofed IP Addresses”: Rob Thomas, Feb 2001

(good technical description of using NetFlow to trace back a flow)

<http://www.cymru.com/Documents/tracking-spoofed.html>

- Other:

“DoS Attacks against GRC.com”: Steve Gibson, GRC, June 2001 (a real-life description of attacks from the victim side; somewhat disputed, but fun to read)

<http://grc.com/dos/grcdos.htm>

SECURITY@CISCO

<http://www.cisco.com/security/>

NetFlow—More Information

- Cisco NetFlow home

[http://www.cisco.com/en/US/tech/tk812/
tsd_technology_support_protocol_home.html](http://www.cisco.com/en/US/tech/tk812/tsd_technology_support_protocol_home.html)

- Linux NetFlow reports HOWTO

<http://www.dynamicnetworks.us/netflow/netflow-howto.html>

- Arbor Networks PeakFlow SP

http://www.arbornetworks.com/products_sp.php

SNMP—More Information

- Cisco SNMP object tracker

<http://www.cisco.com/pcgi-bin/Support/Mibbrowser/mibinfo.pl?tab=4>

- Cisco MIBs and trap definitions

<http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>

- SNMPLink

<http://www.snmplink.org/>

RMON—More Information

- IETF RMON WG

<http://www.ietf.org/html.charters/rmonmib-charter.html>

- Cisco RMON home

http://www.cisco.com/en/US/tech/tk648/tk362/tk560/tsd_technology_support_sub-protocol_home.html

- Cisco NAM product page

<http://www.cisco.com/en/US/products/hw/modules/ps2706/ps5025/index.html>

Packet Capture—More Information

- tcpdump/libpcap home

<http://www.tcpdump.org/>

- Vinayak Hegde's Linux Gazette article

<http://linuxgazette.net/issue86/vinayak.html>

Syslog—More Information

- Syslog.org

<http://www.syslog.org/>

- Syslog logging with PostGres HOWTO

http://kdough.net/projects/howto/syslog_postgresql/

- Agent Smith explains Syslog

<http://routergod.com/agentsmith/>

BGP—More Information

- Cisco BGP home

http://www.cisco.com/en/US/tech/tk365/tk80/tsd_technology_support_sub-protocol_home.html

- Slammer/BGP analysis

http://www.cs.colostate.edu/~massey/pubs/conf/massey_iwdc03.pdf

- Team CYMRU BGP tools

<http://www.cymru.com/BGP/index.html>

Traceback—Direct Contact Information

- APNIC—reporting network abuse: spamming and hacking

<http://www.apnic.net/info/faq/abuse/index.html>

- RIPE—reporting network abuse: spamming and hacking

<http://www.ripe.net/info/faq/abuse/index.html>

- ARIN—network abuse: FAQ

<http://www.arin.net/abuse.html>

References

- Product security:

Cisco's product vulnerabilities

http://www.cisco.com/en/US/products/products_security_advisories_listing.html

Cisco Security Center

<http://www.cisco.com/security>

- ISP essentials:

Technical tips for ISPs every ISP should know

<ftp://ftp-eng.cisco.com/cons/isp/>

- Technical tips:

Troubleshooting High CPU Utilization on Cisco Routers

<http://www.cisco.com/warp/public/63/highcpu.html>

The “show processes” command

http://www.cisco.com/warp/public/63/showproc_cpu.html

NetFlow performance white paper

http://www.cisco.com/en/US/partner/tech/tk812/technologies_white_paper0900aecd802a0eb9.shtml

- Mailing list:

cust-security-announce@cisco.com: all customers should be on this list

Q and A



