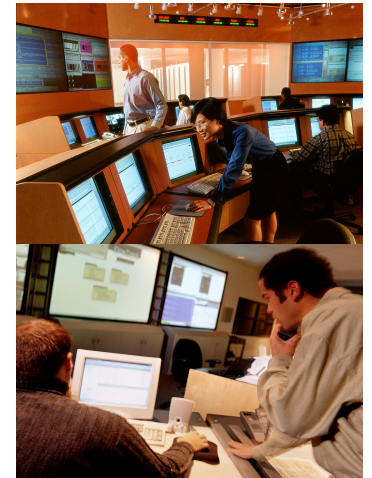
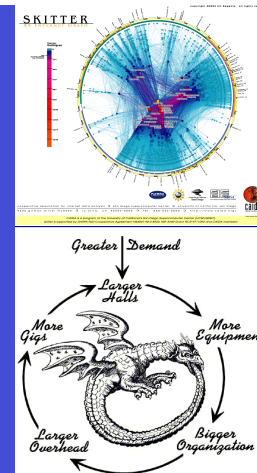


# Gain Visibility





# Total Visibility

---

- Network Telemetry: Why, What and Where
  - Why does one need to listen to the network?
  - What is one listening to?
  - Where does one gather data or information from?
- Network Telemetry: Tools, Techniques and Protocols
  - How to gather data or information?

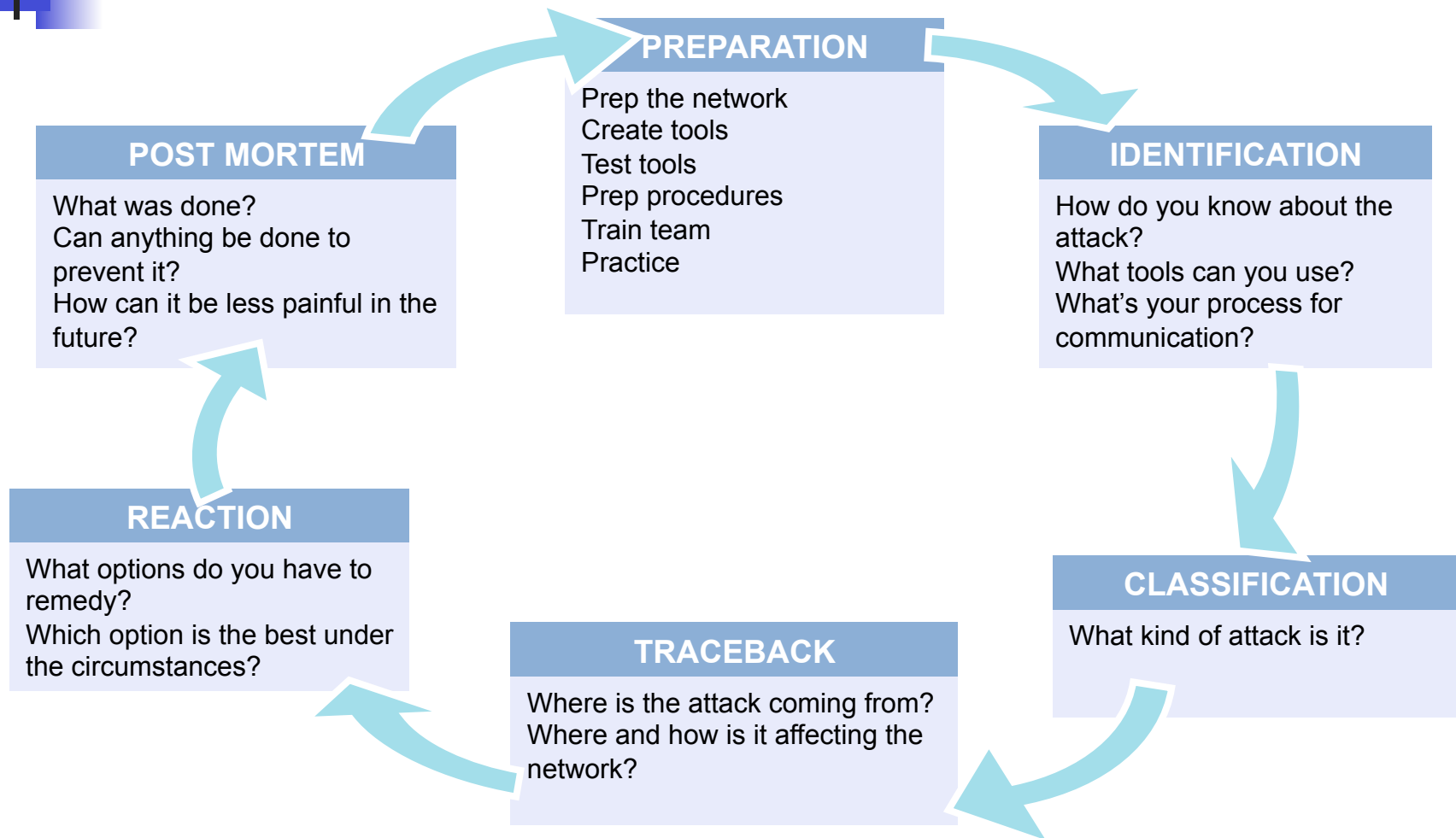


# Check List

---

- Check SNMP. Is there more you can do with it to pull down security information?
- Check RMON. Can you use it?
- Check Netflow. Are you using it, can you pull down more?
- See addendum for lots of links.

# Review: Six Phases of Incident Response





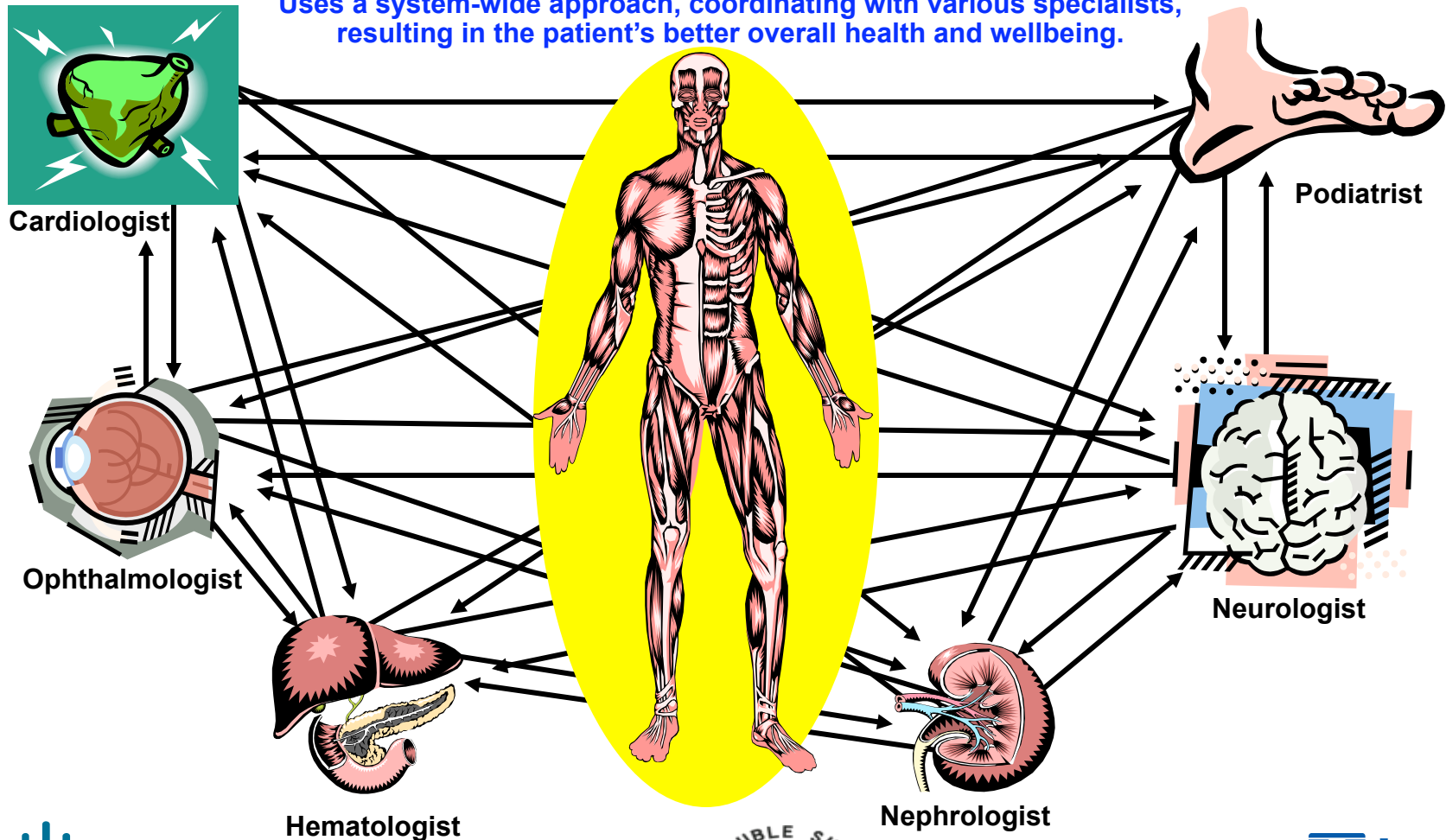
# Why Does One Need to Listen to the Network?

- First and foremost
  - Helps with the other 5 steps [P-I-C-T-R-P]
- Helps to understand the network baseline and behavior
- To understand telemetry elements for information gathering
  - sources (data collection points),
  - protocols to use for data collection
  - telemetry tools
- In the event of security incident
  - to identify and know beforehand that what information is available for forensic work – audit trail
  - faster response time to restore availability when using telemetry during a security incident

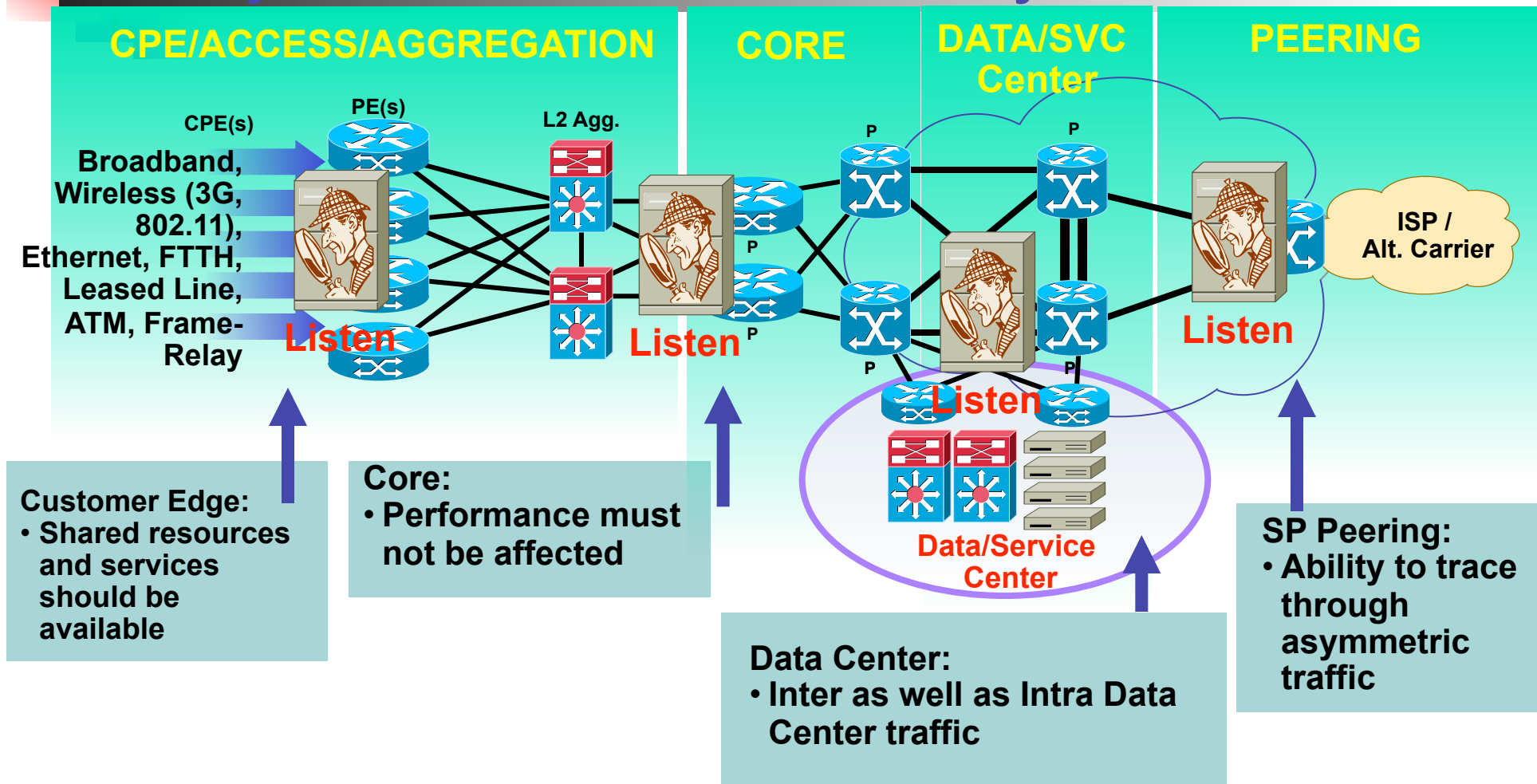
# Holistic Approach to System-Wide Telemetry

## Holistic Approach to Patient Care

Uses a system-wide approach, coordinating with various specialists, resulting in the patient's better overall health and wellbeing.



# Holistic Approach to System-Wide Telemetry



# Understand the Concept of Data Gathering

Risks and threats are **NOT** prevalent in one place **ONLY**...



Need to watch everywhere to avoid being eaten by thousand turkeys...



- Listening to a network element
  - Per device listening
  - Local data provide information about local threats
- Listening to Many
  - Correlation is a MUST
  - Intelligent analysis is a MUST





# High CPU

---

- Spikes in CPU load on routers, switches, servers, and other devices is often an indication that an event is taking place. Such occurrences should always be investigated.
- However, high CPU is not always an indicator of malicious activity. It is important to have both a baseline of historical CPU utilization statistics as well as an understanding of the various processes running on a given system, in order to determine the cause of CPU spikes.
- Correlating CPU utilization with other information such as network traffic statistics, routing-table changes, etc., is often required in order to gain an accurate understanding of the cause(s) and impact of an event.



# Link-Flaps

---

- Link-flaps are also an indication that something is amiss.
- They're often a sign of misconfiguration, backhole incidents and the like - but they can also result from malicious activity, such as a DoS attack against a router causing a reload due to CPU spike, and hence a link-flap.
- Routers and switches can be configured to notify monitoring systems when link-flaps occur.
- Correlating link-flaps with other forms of information is often necessary in order to gain a complete understanding of an event.



# Instrumentation

---

■ Network instrumentation offers the most extensive and useful detection capabilities.

- This instrumentation is often coupled with dedicated analysis systems which collect, analyze, and correlate information from disparate sources in order to present a more complete view of events taking place within the network.
- There are several forms of instrumentation built into routers, switches, and other network devices. Instrumentation is also present in most modern general-purpose operating systems.
- There are a number of open source and commercial tools available which greatly enhance the utility of instrumentation.

■ Getting started with network instrumentation is both inexpensive and relatively easy.

# Example - sh proc c

7600>show proc c | e 0.00%\_0.00%\_0.00%

CPU utilization for five seconds: 38%/26%; one minute: 40%; five minutes: 43%

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
5	192962596	13452649	14343	0.00%	0.52%	0.44%	0	Check heaps
15	4227662201540855414		274	0.65%	0.50%	0.49%	0	ARP Input
26	2629012683680473726		71	0.24%	0.29%	0.36%	0	Net Background
50	9564564	11374799	840	0.08%	0.07%	0.08%	0	Compute load avg
51	15291660	947844	16133	0.00%	0.03%	0.00%	0	Per-minute Jobs
58	15336356	92241638	166	0.08%	0.02%	0.00%	0	esw_vlan_stat_pr
67	10760516	506893631	21	0.00%	0.01%	0.00%	0	Spanning Tree
68	31804659682556402094		1244	7.02%	7.04%	7.75%	0	IP Input
69	25488912	65260648	390	0.00%	0.03%	0.00%	0	CDP Protocol
73	16425564	11367610	1444	0.08%	0.02%	0.00%	0	QOS Stats Export
81	12460616	1020497	12210	0.00%	0.02%	0.00%	0	Adj Manager
82	442430400	87286325	5068	0.65%	0.73%	0.74%	0	CEF process
83	68812944	11509863	5978	0.00%	0.09%	0.11%	0	IPC LC Message H
95	54354632	98373054	552	0.16%	0.12%	0.13%	0	DHCPD Receive
96	61891604	58317134	1061	1.47%	0.00%	4.43%	0	Feature Manager
111	9420	12010	784	0.00%	0.23%	0.46%	0	Exec
165	1817346481141817381		159	0.32%	0.57%	0.40%	0	IP SNMP
166	117953648	573360040	205	0.00%	0.32%	0.26%	0	PDU DISPATCHER
167	545931776	634808059	859	0.40%	1.37%	1.19%	0	SNMP ENGINE
171	22376852	154770330	144	0.00%	0.02%	0.04%	0	IGMP Input
175	680	263	2585	0.24%	0.21%	0.14%	1	SSH Process
177	748193523509072414		21	0.08%	0.02%	0.03%	0	Standby (HSRP)
182	14224288	2051379	6934	0.00%	0.02%	0.00%	0	BGP Scanner

CLI  
Pipes



# Example - sh proc c

7600>sh proc c | e 0.00

CPU utilization for five seconds: 41%/26%; one minute: 46%; five minutes: 44%

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
15	4227657321540854233		274	0.40%	0.39%	0.47%	0	ARP Input
26	2629008963680468704		71	0.08%	0.36%	0.39%	0	Net Background
50	9564512	11374786	840	0.08%	0.07%	0.08%	0	Compute load avg
68	31804578042556183430		1244	9.65%	8.49%	7.75%	0	IP Input
69	25488888	65260576	390	0.32%	0.05%	0.01%	0	CDP Protocol
82	442429604	87286223	5068	0.73%	0.73%	0.74%	0	CEF process
:								
175	624	92	6782	0.57%	0.49%	0.16%	1	SSH Process

CLI Pipes allow clean and crisp output

## IOS CLI - sh proc c (cont.)

- There are processes which are platform-specific - i.e., Feature Manager is found on the 6500/7600 only, while IPC CBus is 7500-specific.
- Aliasing the more complex sh proc c commands to a single-letter alias as part of the standard config is extremely useful when the box is under high load and it's hard to type on the console:

```
Router(config)#alias exec p show proc c |  
e 0.00%__0.00%__0.00%
```

- Understanding your platform(s), and what's normal - including periodically-run processes (BGP Scanner, for example) - is key
- On the 12000, one must either attach to a linecard or perform an execute command specifying a linecard in order to see its CPU load; on the 7500, one uses the if-con command to session to a VIP.



## IOS CLI - sh int

---

- Sh int displays interface-level statistics, including throughput (pps) and bandwidth (bps)
- Typically, routers are set to use a 5-minute decaying average for interface statistics by default - changing this to 1 minute gives more granular statistics
- Looking for high input/output rates over a period of a minute or so can be very helpful
- Clear the counters first, otherwise it's much harder to determine which interfaces are receiving high rates of traffic



# Example - sh int

```
GigabitEthernet3/13 is up, line protocol is up (connected)
  Hardware is C6k 1000Mb 802.3, address is 00d0.0136.000a (bia 00d0.0136.000a)
  Description: IP TELEPHONY
  Internet address is 10.89.254.130/26
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex mode, link type is autonegotiation, media type is SX
  output flow-control is unsupported, input flow-control is unsupported, 1000Mb/s
  Clock mode is auto
  input flow-control is off, output flow-control is off
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters 1y39w
  Input queue: 0/75/15005/235 (size/max/drops/flushes); Total output drops: 1
  Queueing strategy: fifo
  Output queue :0/40 (size/max)
  5 minute input rate 4751000 bits/sec, 3006 packets/sec
  5 minute output rate 4499000 bits/sec, 2755 packets/sec
  L2 Switched: ucast: 19841909032 pkt, 3347755205145 bytes - mcast: 96885779 pkt, 5131184435 bytes
  L3 in Switched: ucast: 27282638229 pkt, 5095662463006 bytes - mcast: 94 pkt, 5191 bytes mcast
  L3 out Switched: ucast: 43107617667 pkt, 7275264441541 bytes
    47118207406 packets input, 9306459456266 bytes, 0 no buffer
    Received 83653389 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 649 overrun, 0 ignored
    0 input packets with dribble condition detected
    43210876182 packets output, 8089398934796 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 babbles, 0 late collision, 0 deferred
    0 lost carrier, 0 no carrier
    0 output buffer failures, 0 output buffers swapped out
```

## Example - sh int

```
12000>sh int po1/1/0 | i 1 minute
```

```
1 minute input rate 56616000 bits/sec, 18097  
packets/sec
```

```
1 minute output rate 120609000 bits/sec, 24120  
packets/sec
```

Rate  
Interval

```
12000>sh int po1/1/0 | i 1 minute
```

```
1 minute input rate 59030000 bits/sec, 19171  
packets/sec
```

```
1 minute output rate 111233000 bits/sec, 22365  
packets/sec
```

```
12000>sh int po1/1/0 | i 1 minute
```

```
1 minute input rate 54307000 bits/sec, 17637  
packets/sec
```

```
1 minute output rate 119223000 bits/sec, 23936  
packets/sec
```



## IOS CLI - sh ip int

---

- `sh ip int` gives information about features configured on an interface
- It's useful to get the number or name of an ACL in order to check ACL counter hits (6500/7600 only shows ACL counters on Sup720 w/PFC3BXL)
- uRPF drop information is also available via `sh ip int`, shows information about spoofed and/or RTBH-dropped packets



## Example - sh ip int

---

```
12000>sh ip int po1/1/0 | i veri
```

```
IP verify source reachable-via ANY
```

```
794407 verification drops
```

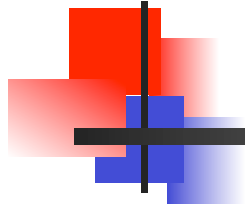
```
1874428129 suppressed verification  
drops
```

```
12000>sh ip int po1/1/0 | i veri
```

```
IP verify source reachable-via ANY
```

```
794408 verification drops
```

```
1874444463 suppressed verification  
drops
```



## IOS CLI - sh ip traffic

---

- Sh ip traffic provides a lot of useful global statistics, including per-protocol counts for ICMP, TCP, UDP, and multicast traffic
- Very useful for troubleshooting in general, as well as for spotting oddities
- Also shows global uRPF drop statistics





## Example - sh ip traffic

---

```
12000>sh ip traff | i RPF
```

```
0 no route, 124780722 unicast RPF, 0 forced drop
```

```
12000>sh ip traff | i RPF
```

```
0 no route, 124816525 unicast RPF, 0 forced drop
```

```
12000>sh ip traff | i RPF
```

```
0 no route, 127777619 unicast RPF, 0 forced drop
```

```
12000>sh ip traff | i RPF
```

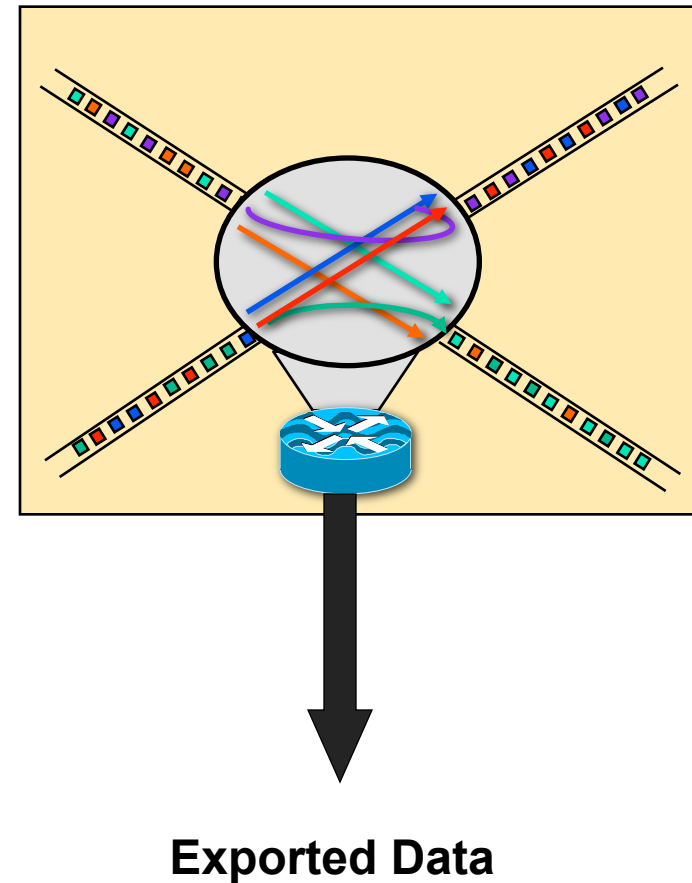
```
0 no route, 135875095 unicast RPF, 0 forced drop
```

```
12000>sh ip traff | i RPF
```

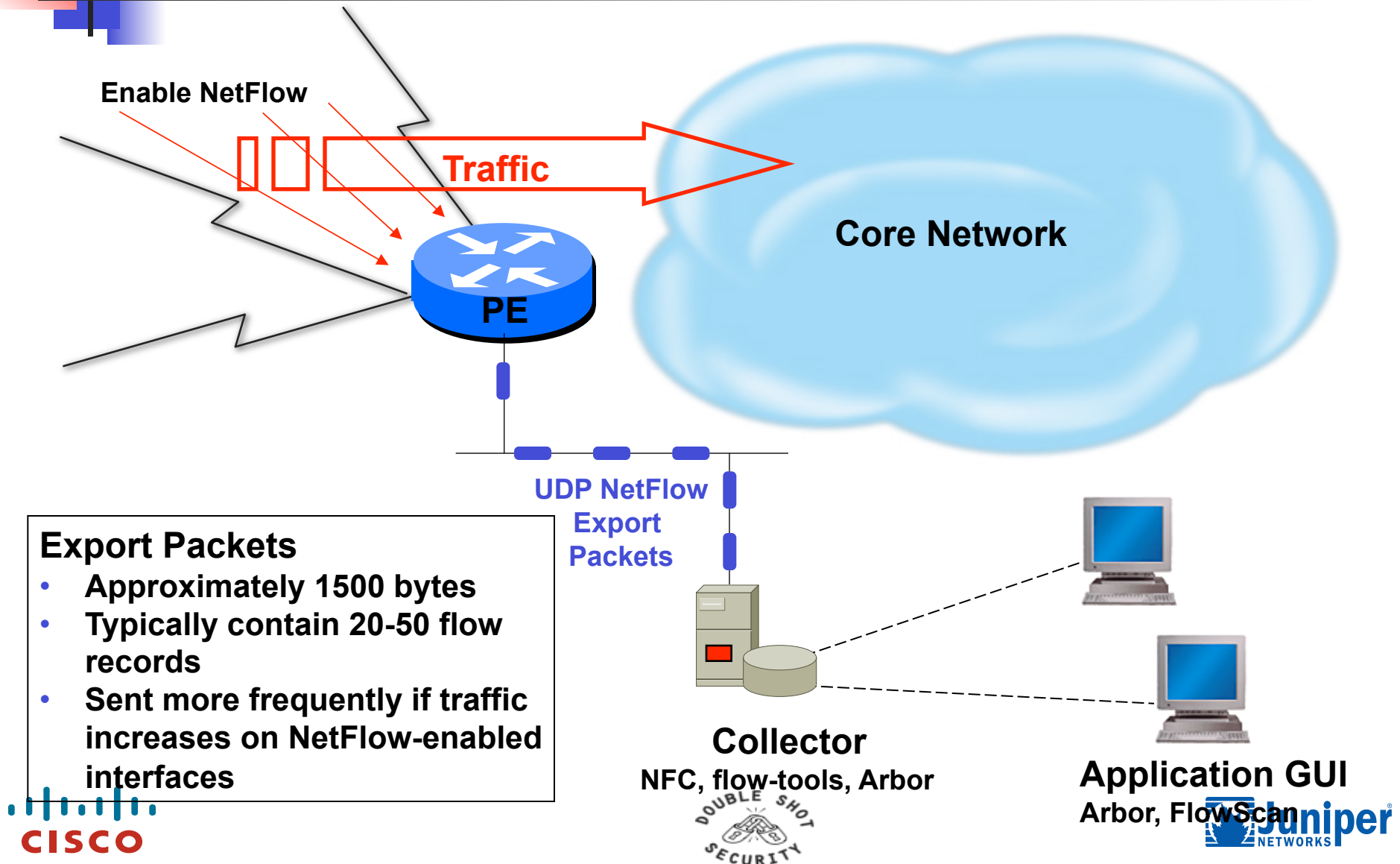
```
0 no route, 150883277 unicast RPF, 0 forced drop
```

# 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





## 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 & 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: Clarifying the Version Myth

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 & TOS information
8	Choice of eleven aggregation schemes Reduces resource usage
9	Flexible, extensible file export format to enable easier support of additional fields & technologies; coming out now are MPLS, Multicast, & BGP Next-Hop

**Cisco Catalyst 6500 Series Router supports  
versions 5 and 8 in Cisco IOS Software Release 12.1(13)E**



## 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!**



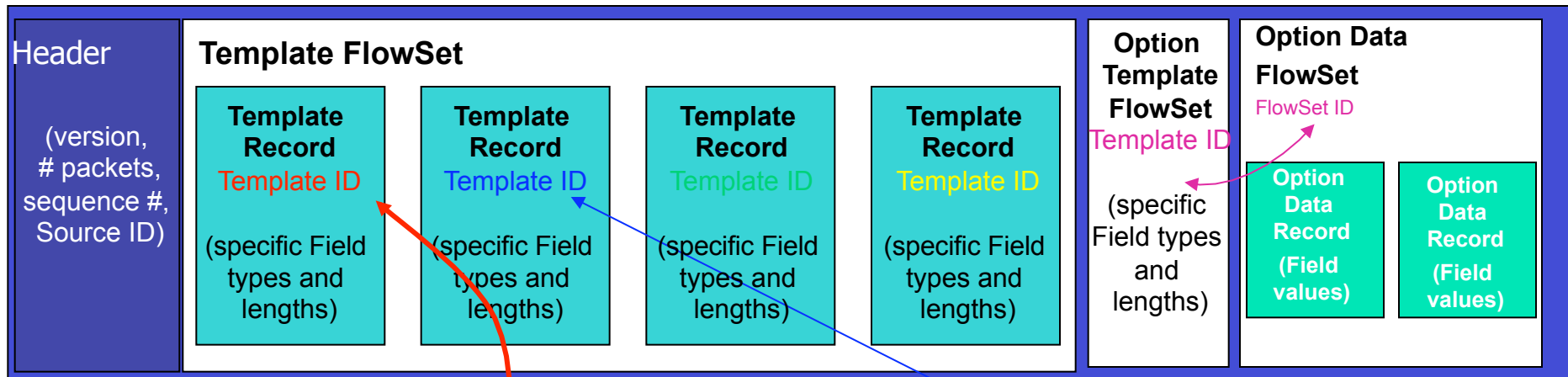
# Netflow v9 Principles

---

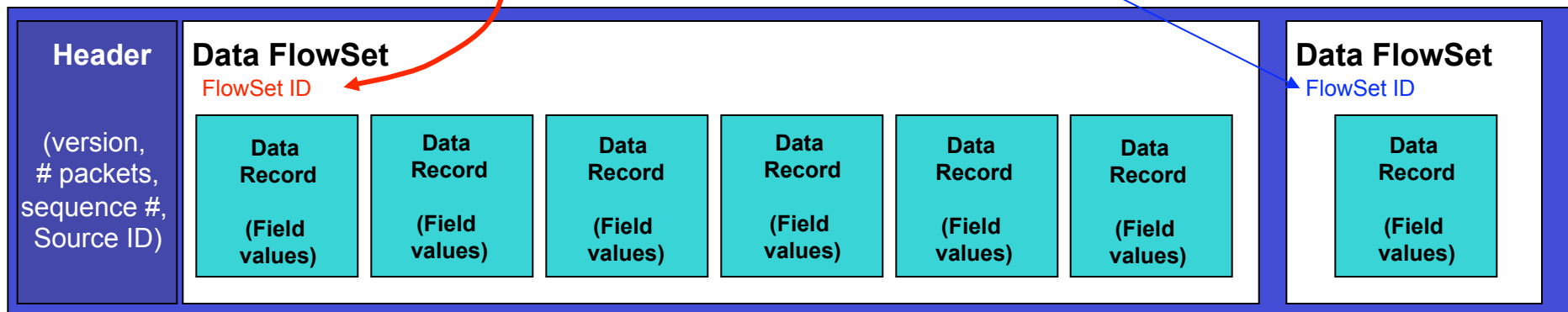
- Version 9 is an **export format**
- Works as a push model
- Send the template regularly (configurable)
- Independent of the underlying protocol, it is ready for any reliable protocol (e.g.,: TCP, SCTP)

# 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
```

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

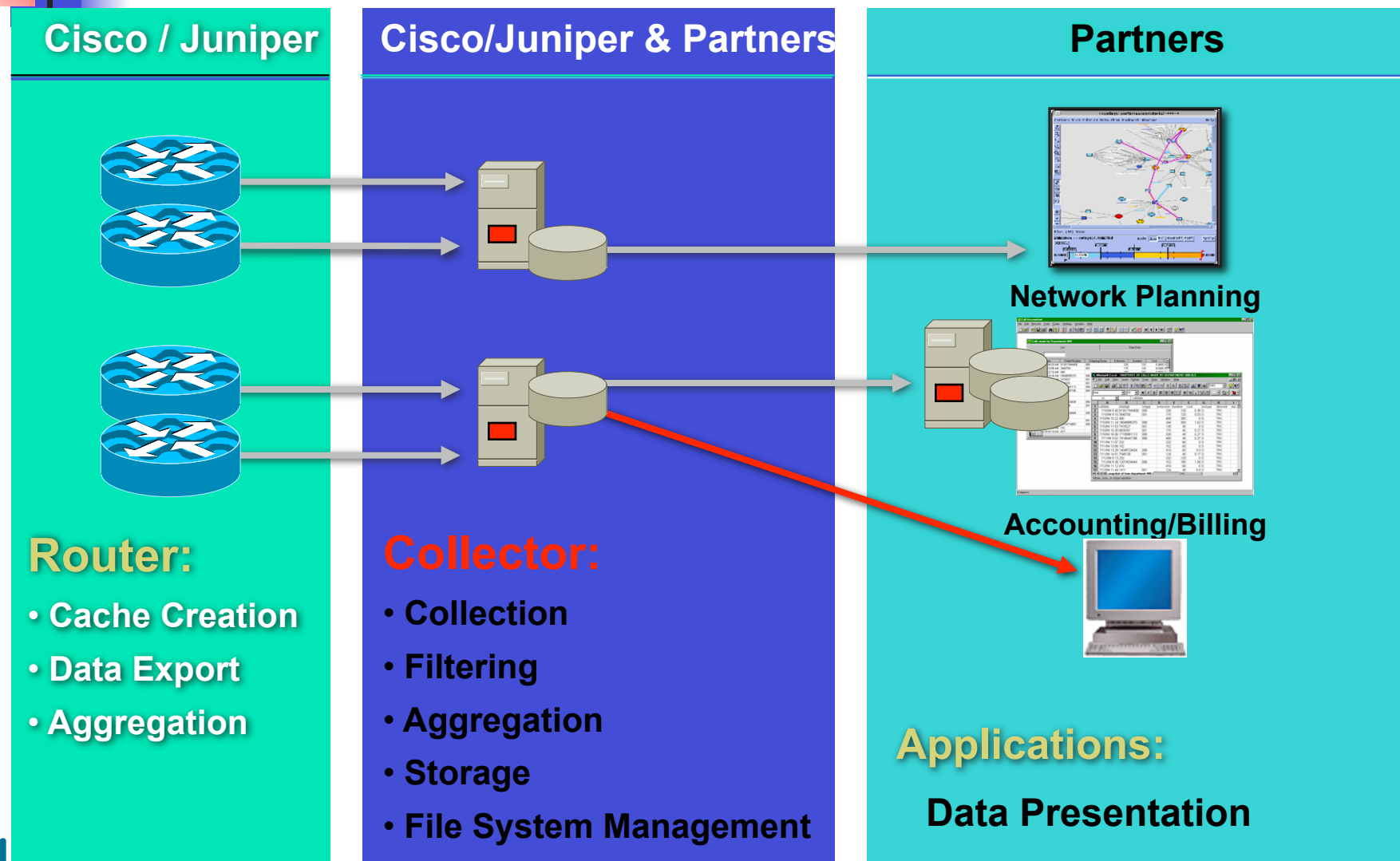
```
8 Version 8 export format
```

```
9 Version 9 export format
```

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

Export versions available for aggregated NetFlow flows

# NetFlow / jflow Infrastructure



# Cisco 7200 NetFlow Example

```
7200>sh ip cache flow
```

```
IP packet size distribution (14952M total packets):
```

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

.001	.325	.096	.198	.029	.014	.010	.010	.012	.003	.003	.
005	.003	.003	.002								

512	544	576	1024	1536	2048	2560	3072	3584	4096	4608
.004	.005	.009	.043	.217	.000	.000	.000	.000	.000	.000

**Active flows**

```
IP Flow Switching Cache, 4456704 bytes
```

```
65527 active, 9 inactive, 2364260060 added
```

```
4143679566 age polls, 0 flow alloc failures
```

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

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

```
last clearing of statistics never
```

**NetFlow Timeouts  
– tune to avoid the  
churn**

# Cisco 7200 NetFlow Example (Cont.)

## Traffic type

Protocol Idle (Sec)	Total Flows	Flows /Sec	Packets /Flow	Bytes /Pkt	Packets /Sec	Active (Sec) /Flow
----- Flow	Flows	/Sec	/Flow	/Pkt	/Sec	/Flow
TCP-Telnet 17.2	1398292	0.3	14	156	4.6	6.0
TCP-FTP 4.8	99569986	23.1	1	41	24.2	0.0
TCP-FTPD 17.4	185530	0.0	1	66	0.0	1.5
TCP-WWW 10.1	440235639	102.5	8	483	919.5	2.9
TCP-SMTP 20.0	18951357	4.4	21	629	94.1	6.4
TCP-X 40.8	11340	0.0	1	48	0.0	0.2
TCP-BGP 12.5	4018	0.0	2	51	0.0	7.5
TCP-NNTP 16.9	2701390	0.6	104	846	65.5	10.6
TCP-Frag 17.2	38932	0.0	11	407	0.1	1.9
TCP-other 18.6	403434143	93.9	7	444	688.2	6.9
UDP-DNS 17.7	65590214	15.2	1	114	24.0	1.6

**Hint:**  
How many  
TCP based  
applications  
you know  
have 1 pkt /  
flow?

## Cisco 7200 NetFlow Example (Cont.)

**Hint: What's going on here?**

SrcIf	SrcIPaddress	DstIf	DstIPaddress	Pr
SrcP DstP Pkts				
Fa0/1	10.66.74.46	Fa0/0	219.103.129.162	01
0000 0800	1			
Fa0/1	10.66.115.182	Fa0/0	194.22.114.198	01
0000 0800	1			
Fa2/1	10.66.74.46	Fa0/0	61.79.227.123	01
0000 0800	1			
Fa0/1	10.66.74.46	Fa0/0	211.167.105.242	01
0000 0800	1			
Fa0/0	129.42.184.35	Null	64.104.193.198	06
2891 0019	3			
Fa2/1	10.66.115.182	Fa0/0	202.20.138.184	01
0000 0800	1			
Fa2/1	10.66.115.182	Fa0/0	63.76.237.255	01
0000 0800	1			

# Cisco Catalyst 6500 and 7600 Series Switches

```
6500>sh mls netflow ip detail
```

```
Displaying Netflow entries in Supervisor Ear1
```

```
DstIP          SrcIP          Prot:SrcPort:DstPort  Src i/f:AdjPtr
```

```
-----  
Pkts          Bytes          Age      LastSeen  Attributes
```

**Review the output**

```
-----  
QoS          Police Count Threshold      Leak      Drop Bucket  Use-Tbl Use-Enable  
-----+-----+-----+-----+-----+-----+  
172.87.19.217 171.70.154.90 tcp :10112 :www      1023: 0  
3          144          10      00:07:11  L3 - Dynamic  
0x0          0          0          0          0          NO      48          NO          NO  
171.101.24.123 171.69.89.39 tcp :1303 :139      400 : 0  
0          0          39      00:06:42  L3 - Dynamic  
0x0          0          0          0          0          NO      48          NO          NO  
202.56.200.22 198.133.219.25 icmp:0 :0      1028: 0  
26          2028          383      00:07:05  L3 - Dynamic  
0x0          0          0          0          0          NO      78          NO          NO
```

# Cisco Catalyst 6500 and 7600 Series Switches (Cont.)

```
6500>sh mls netflow ip dest www.cisco.com det
Displaying Netflow entries in Supervisor Earl
DstIP          SrcIP          Prot:SrcPort:DstPort  Src 1/f:AdjPtr
```

```
-----
Pkts           Bytes          Age    LastSeen  Attributes
```

**Review the output.**

```
-----
QoS      Police Count Threshold    Leak    Drop Bucket  Use-Tbl Use-Enable
-----+-----+-----+-----+-----+-----+-----+
198.133.219.25  66.189.188.230  icmp:0    :0          1017: 0
1          60          28      00:16:36    L3 - Dynamic
0x0        0           0        0           0        NO      60          NO      NO
198.133.219.25  142.32.208.231  tcp :9415  :www        1016: 0
34         1501        32      00:16:32    L3 - Dynamic
0x0        0           0        0           0        NO      40          NO      NO
198.133.219.25  65.114.202.35   tcp :4936  :www        1017: 0
24         1099        24      00:16:40    L3 - Dynamic
0x0        0           0        0           0        NO      40          NO      NO
198.133.219.25  80.202.170.129  icmp:0    :0          1017: 0
1          60          32      00:16:32    L3 - Dynamic
0x0        0           0        0           0        NO      60          NO      NO
```



# Versions

---

- Some releases are vendor/product specific
- What you need to know
  - Version 5
    - Widely supported
  - Version 8
    - Adds security to reporting stream (DES)
  - Version 9
    - Adds generalized formatting
    - Reduces need to upgrade tools between versions





# cflowd Configuration Example

You must configure sampling for cflowd to work

```
forwarding-options {  
    sampling {  
        input {  
            family inet {  
                rate 1000;  
                run-length 9;  
            }  
        }  
        output {  
            file filename sample.cfld files 20 size 1m;  
            cflowd 10.1.86.2 {  
                port 2055;  
                version 5;  
            }  
        }  
    }  
}
```



# cflowd Output Option

- cflowd is an output option under the sampling configuration
  - Each option discussed in detail

```
forwarding-options {  
  sampling {  
    input {  
      family inet {  
        rate 1000;  
        run-length 9;  
      }  
    }  
    output {  
      file filename sample.cfld files 20 size 1m;  
      cflowd 10.1.86.2 {  
        port 2055;  
        version 5;  
      }  
    }  
  }  
}
```



# cflowd Aggregate Format

## Viewing the local log file on the router

```
lab@R1> show log sampled
```

```
Jan 7 18:30:44      Start time of flow: 3812598
Jan 7 18:30:44      End time of flow: 3812598
Jan 7 18:30:44      Src port: 1088
Jan 7 18:30:44      Dst port: 1241
Jan 7 18:30:44      TCP flags: 0x0
Jan 7 18:30:44      IP proto num: 6
Jan 7 18:30:44      TOS: 0x0
Jan 7 18:30:44      Src AS: 64514
Jan 7 18:30:44      Dst AS: 64513
Jan 7 18:30:44      Src netmask len: 16
Jan 7 18:30:44      Dst netmask len: 24
Jan 7 18:30:44 v5 flow entry
Jan 7 18:30:44      Src addr: 192.168.46.101
Jan 7 18:30:44      Dst addr: 172.16.3.18
Jan 7 18:30:44      Nhop addr: 10.1.84.0
Jan 7 18:30:44      Input interface: 30
Jan 7 18:30:44      Output interface: 40
Jan 7 18:30:44      Pkts in flow: 1
Jan 7 18:30:44      Bytes in flow: 46
Jan 7 18:30:44      Start time of flow: 3812603
```

```
Jan 7 18:30:44      End time of flow: 3812603
Jan 7 18:30:44      Src port: 1029
Jan 7 18:30:44      Dst port: 20
Jan 7 18:30:44      TCP flags: 0x0
Jan 7 18:30:44      IP proto num: 6
Jan 7 18:30:44      TOS: 0x0
Jan 7 18:30:44      Src AS: 64514
Jan 7 18:30:44      Dst AS: 64513
Jan 7 18:30:44      Src netmask len: 16
Jan 7 18:30:44      Dst netmask len: 24
```



# Remote cflowd Server

## Files created on the remote cflowd server

```
ping# ls /usr/local/arts/data/cflowd/flows
```

10.1.83.1.flows.0	10.1.83.1.flows.4	10.1.83.1.flows.8
10.1.83.1.flows.1	10.1.83.1.flows.5	10.1.83.1.flows.9
10.1.83.1.flows.2	10.1.83.1.flows.6	
10.1.83.1.flows.3	10.1.83.1.flows.7	

**FreeBSD server running CAIDA cflowd package**



# Raw Flows on the cflowd Server

## Viewing the raw flows on the remote cflowd server

```
ping# flowdump 10.1.83.1.flows.0
```

```
FLOW
```

```
index:          0xc7ffff
router:         10.1.86.1
src IP:         192.168.46.101
dst IP:         172.16.3.18
input ifIndex:  30
output ifIndex: 40
src port:       1029
dst port:       20
pkts:           1
bytes:          46
IP nexthop:     10.1.84.0
start time:     Mon Jan 7 21:30:12 2002
end time:       Mon Jan 7 21:30:12 2002
protocol:       6
tos:            0
src AS:         64514
dst AS:         64513
src masklen:    16
dst masklen:    24
TCP flags:      0x0
engine type:    0
engine id:      0
```

**FreeBSD server running CAIDA cflowd package**



# Principal NetFlow Benefits

## SERVICE PROVIDER

- Peering arrangements
- SLA VPN user reporting
- Usage-based billing
- DoS/worm detection
- Traffic engineering
- Troubleshooting

## ENTERPRISE

- 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



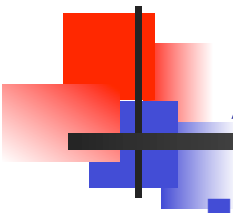
# Open Source Tools for NetFlow Analysis —The OSU Flow-Tools

- 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!





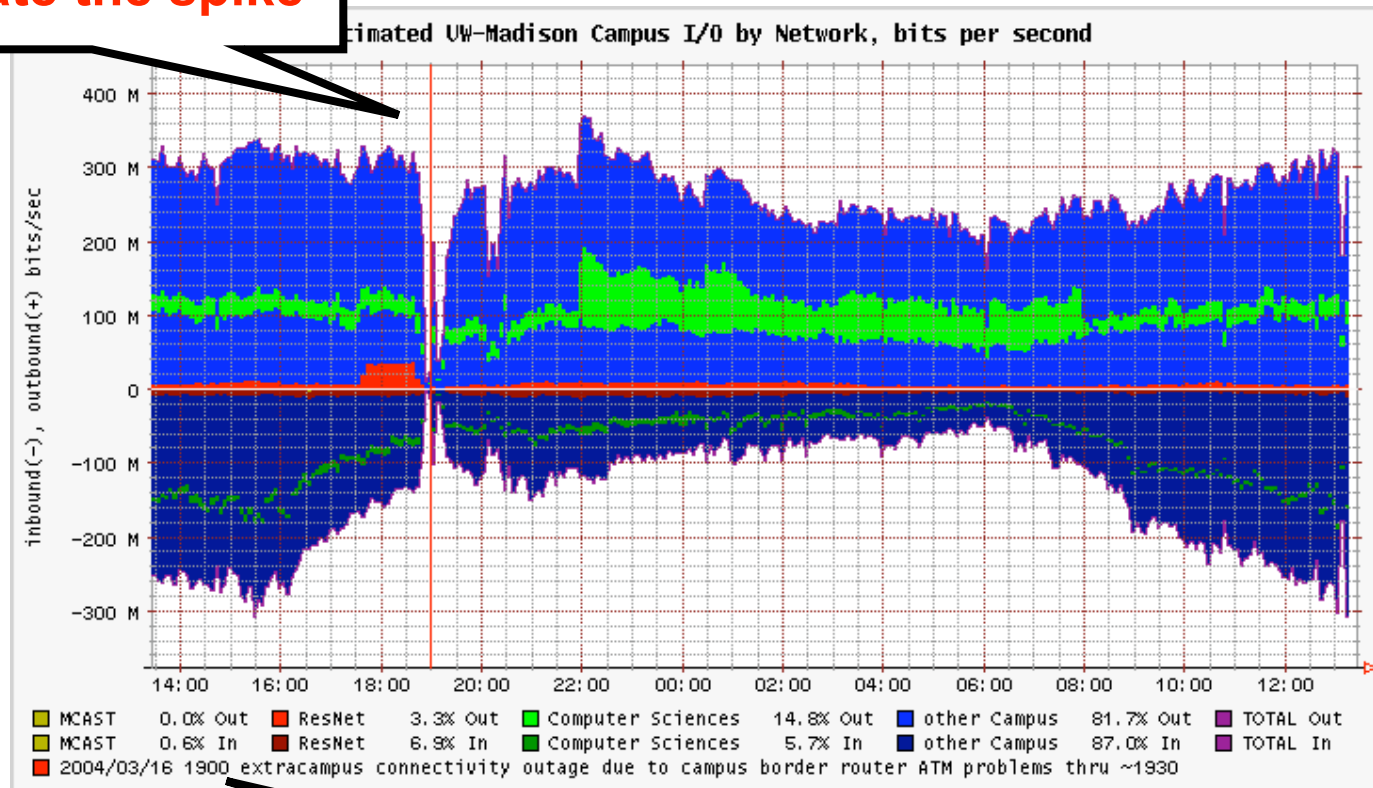
# Open Source Tools for NetFlow Analysis Visualization—FlowScan

- 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



# Open Source Tools for NetFlow Analysis Visualization—FlowScan

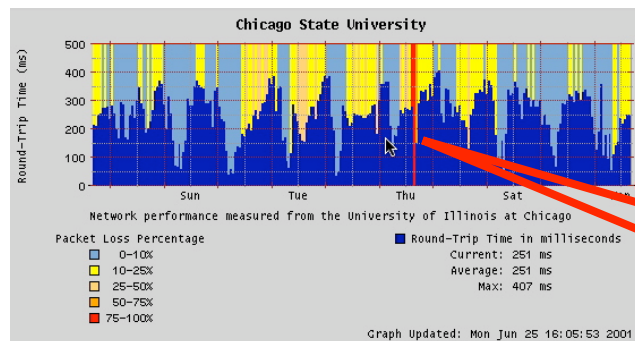
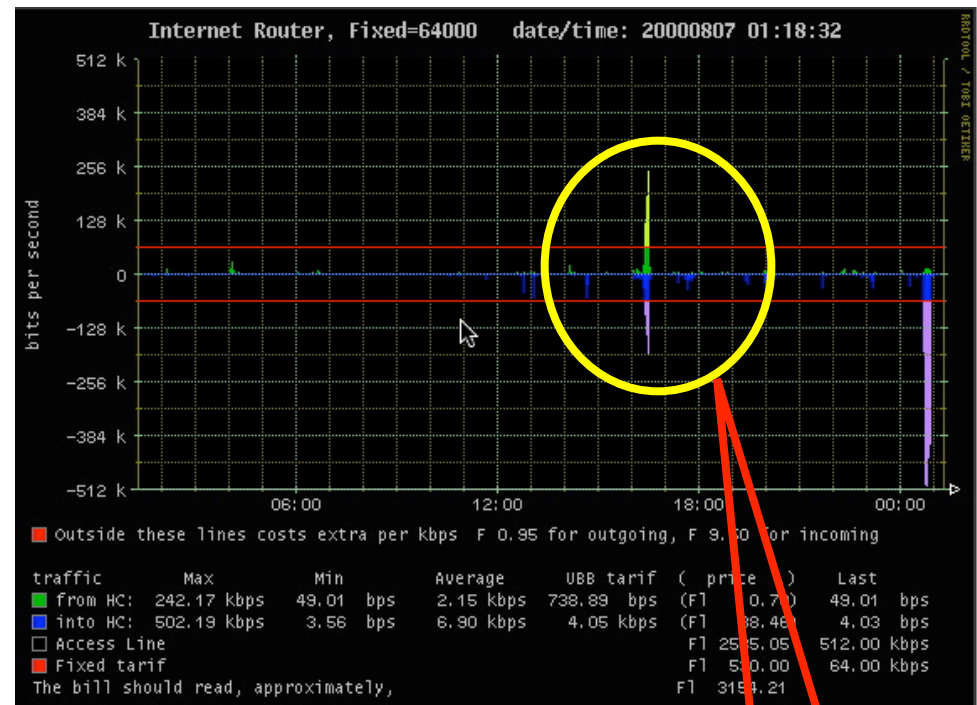
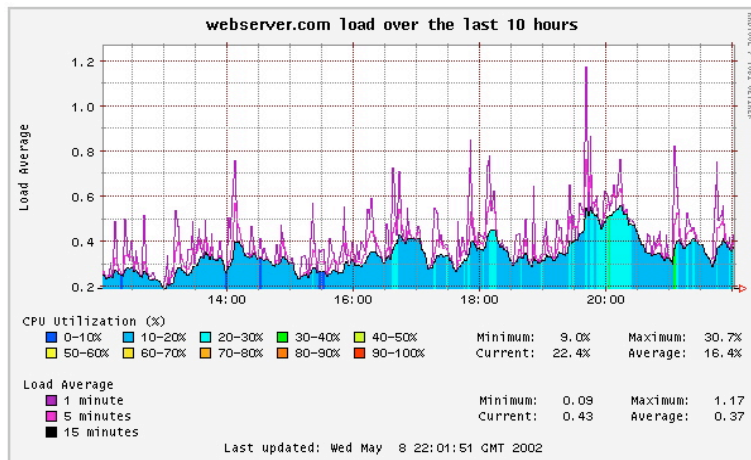
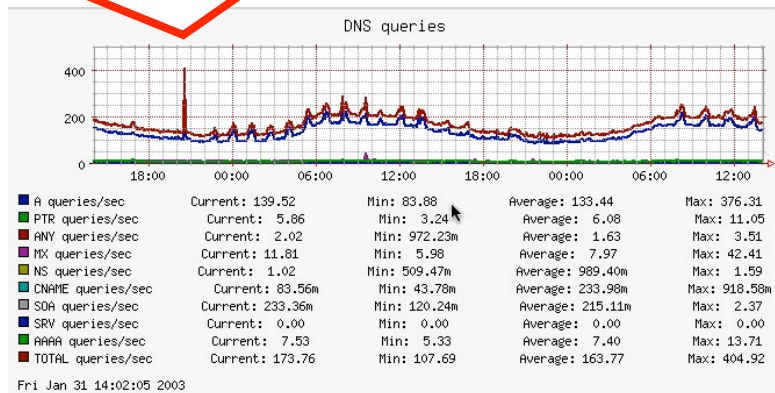
Investigate the spike



An identified cause of the outage

# Other Visualization Techniques Using SNMP Data with RRDTool

## Anomaly for DNS Queries

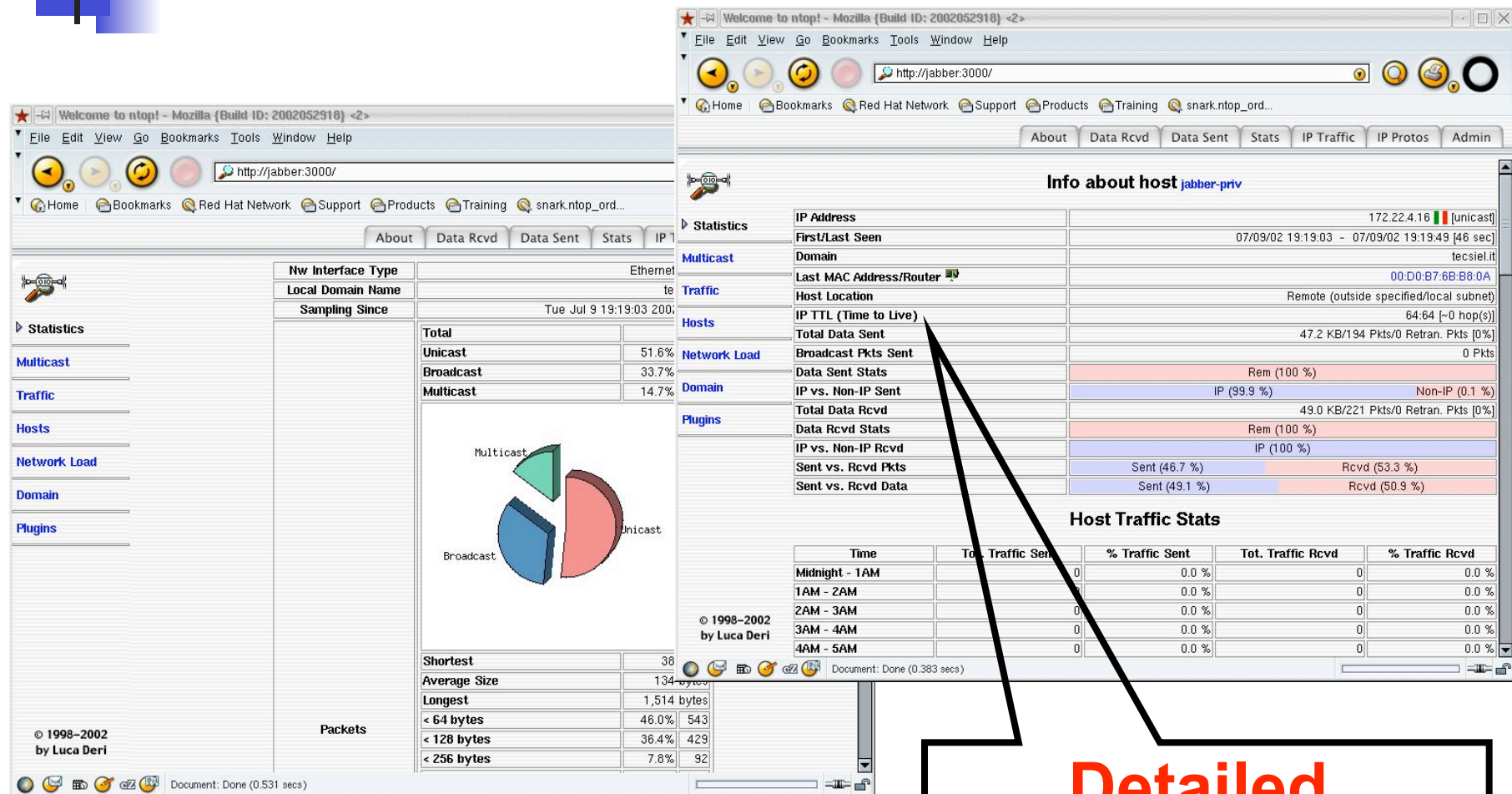


**Thru'put Spike**

**RTT Spike**

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

# Displaying RMON—ntop Examples

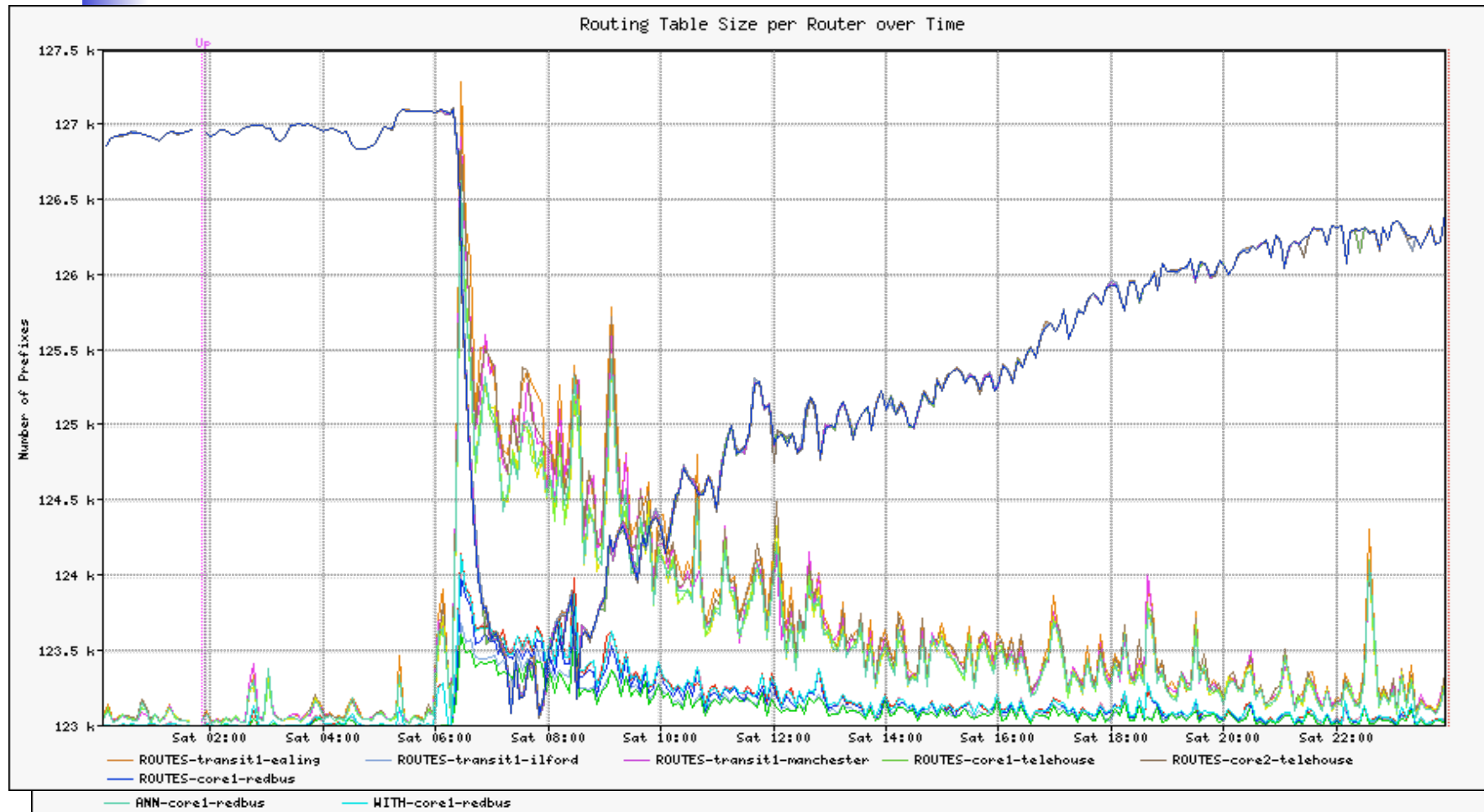


**Detailed Analysis i.e. TTL**

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

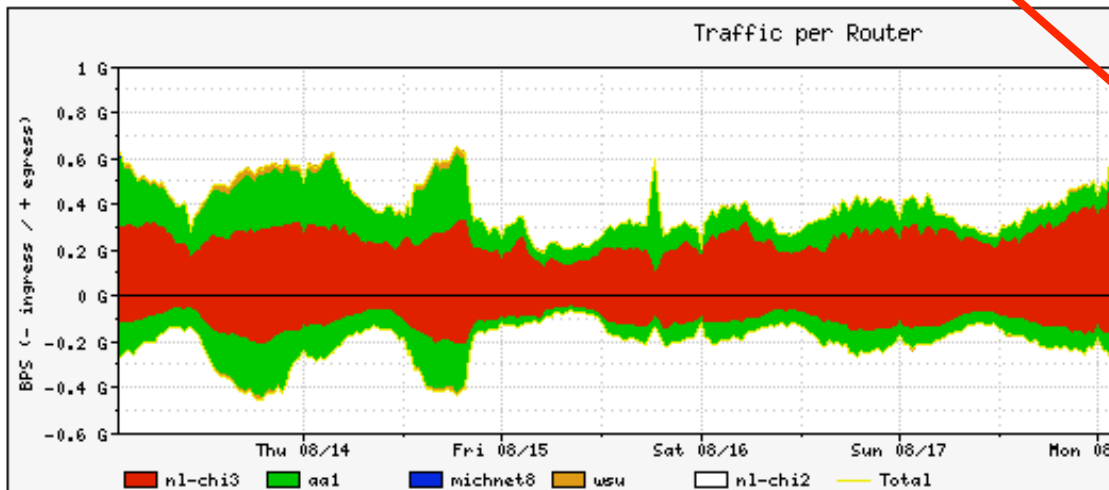
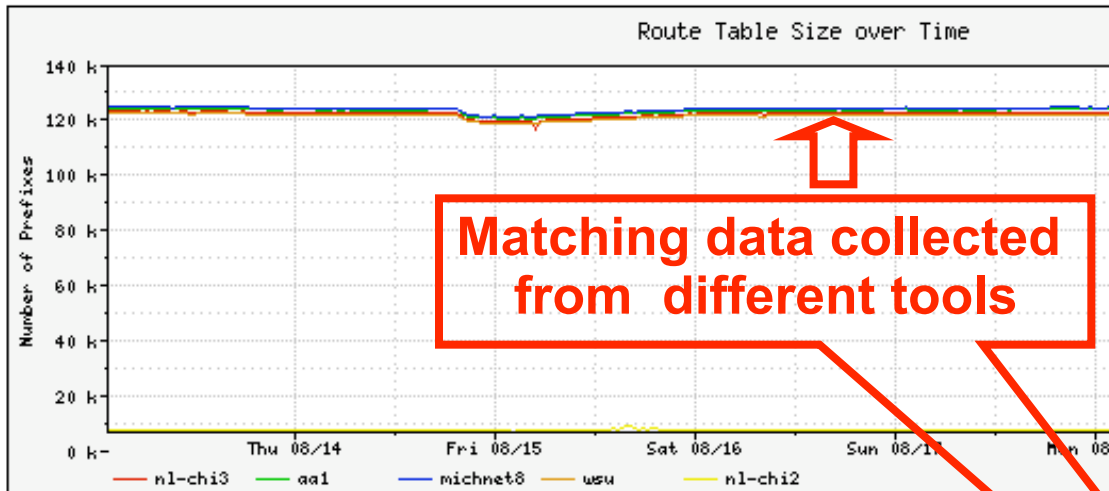


# BGP Example—SQL Slammer





# Correlating NetFlow and Routing Data



tcsh — tcsh

danny@rambler% cat prefixes

Prefix Length	*Current	Daily Max	Daily Average
/24	65,900	68,497	67,259
/23	9,904	10,157	10,027
/22	9,053	9,211	9,110
/21	6,035	6,106	6,045
/20	8,485	8,560	8,487
/19	8,175	8,221	8,161
/18	3,007	3,031	3,005
/17	1,693	1,705	1,690
/16	7,293	7,396	7,326
/15	473	473	469
/14	263	263	262
/13	98	98	97
/12	55	55	54
/11	12	12	11
/10	6	6	5
/9	4	4	3
/8	19	19	18
Current_Total: 120,475			
Max_Total: 123,814			
Average_Total: 122,029			
Current v. Average: 98.73% (1554 prefixes)			
* Current Based on my Snapshot @9P MDT 8.14.2003			
[~]			
danny@rambler%			



# 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
- Logging of ACLs is generally contraindicated due to CPU overhead—NetFlow provides more info, 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 info based upon device type, function, other criteria
- Syslog-ng from [http://www.balabit.com/products/syslog\\_ng/](http://www.balabit.com/products/syslog_ng/) adds a lot of useful functionality—HOW-TO located at <http://www.campin.net/newlogcheck.html>



# Local Log Files

---

- Local log files are useful for:
  - Detecting problems on the router
  - Monitoring the system usage by *friendly* users
  - Monitoring normal events
- Local log files are *not* useful for:
  - Monitoring the activity of attackers who have compromised your system

A good attack will erase the evidence of their activity from local log files!



# Remote Log Files

---

- Logging to a remote host has several advantages:
  - Initial attacker activity is available in the log
    - Remote logging is difficult to stop prior to the compromise
    - Attacker can stop remote logging once the system is compromised
    - The *lack* of remote logging can be an indication of a problem
  - Remote logs from multiple systems can be consolidated
    - You can monitor and coordinate events between multiple systems





# What is Syslog?

---

- Operating systems and applications generate a multitude of log messages about a variety of things
  - Syslog was developed as a generic logging server to accept, categorize, and record log messages
  - As systems became more complex, a method was needed to forward log messages to a remote syslog server and consolidate messages from multiple hosts
  - BSD Syslog Protocol
    - Outlined in RFC 3164
    - Specifies the format and content of remote syslog messages



## Syslog Facilities (1 of 2)

- Each message has a *facility* used to categorize the type of message generated
- The router specifies the facility to which each message belongs

Facility	Description
Any	Any facility
Authorization	Any authorization attempt
Change-log	Any change to the configuration
Conflict-log	Messages generated when configuration conflicts with the hardware
Cron	Cron daemon
Daemon	Various system daemons



## Syslog Facilities (2 of 2)

Facility	Description
Firewall	Firewall filtering subsystem
Interactive-commands	Commands executed in the CLI
Kernel	Messages generated by the JUNOS software kernel
PFE	Messages generated by the PFE
User	Messages from user processes
Local0 – Local7	<i>Local-use</i> facilities



# Syslog Severity

---

- Each message has a *severity* used to prioritize its importance
  - Setting a facility and severity level causes the router to log all messages for that severity at the specified level and above
    - For example, logging at the `critical` level also causes `alert` and `emergency` messages to be logged



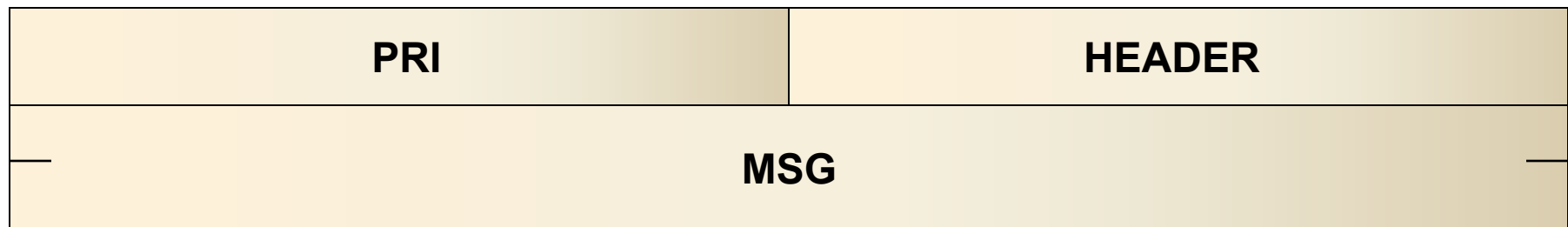
emergency alert **critical** error warning notice info debug

**More severe**



# Syslog Packet Format

- Format and content of messages defined in RFC 3164
  - Messages sent on UDP port 514
  - No minimum size
  - Maximum size is 1024 bytes
- Messages consist of three text strings
  - PRI (Priority)
  - HEADER
  - MSG (Message)





# Overriding the Remote Facility

- By default, syslog messages are sent with their normal BSD-specified facility and various local facilities
  - You can override the message facility

```
[edit system]
lab@R1# show
syslog {
    host 10.1.10.2 {
        authorization info;
        change-log info;
        interactive-commands info;
        facility-override local7;
        log-prefix Security;
    }
}
```



# Security Cautions

---

- Caution:
  - Syslog messages can contain sensitive information in cleartext
    - User authentication messages when logging the authorization facility
    - Passwords entered into the configuration when logging the interactive-commands facility
  - Consider sending syslog messages only on the out-of-band management network
  - Compromise of the remote syslog server might give an attacker enough information to compromise the router!

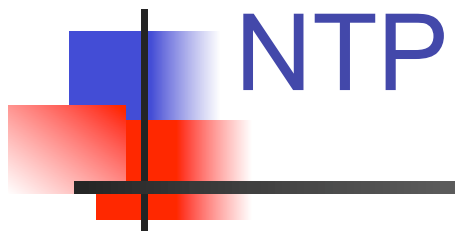


# Good things to log

---

- All login attempts
  - Successful or not
- All commands typed
  - So you know who did what and when
  - Helps Identifying “training issues” as well 😊
- Availability issues
  - Interface status change (not dialup)
  - More critical for core routers
  - BGP peering changes
  - OSPF neighbor changes?







# Benefits of Deploying NTP

---

- Very valuable on a global network with network elements in different time zones
- Easy to correlate data from a global or a sizable network with a consistent time stamp
- NTP based timestamp allows to trace security events for chronological forensic work
- Any compromise or alteration is easy to detect as network elements would go out of sync with the main 'clock'
- Did you there is an NTP MIB? Some think that we may be able to use "NTP Jitter" to watch what is happening in the network.



# Local System Time

---

- In a security situation, you must have a consistent concept of time across the network (it does not have to be the correct time, just consistent)
- Choose UTC/GMT or Head office Time Zone
- NTP was developed to synchronize large numbers of network devices to a consistent, accurate time reference
- Local and remote log files are stamped with the local system time
  - Event correlation is easier if all devices are synchronized
  - Law enforcement officials might need copies of these logs



# Network Time Protocol

---

- NTPv3: *Network Time Protocol (Version 3) Specification, Implementation and Analysis* (RFC 1305—March 1992)
  - Defines a protocol to keep accurate, synchronized time between network devices
    - Uses UDP port 123
  - Additional features incorporated in NTPv4
    - DES encryption
    - Not an IETF standard, but widely supported
    - Backwards compatible with NTP v3

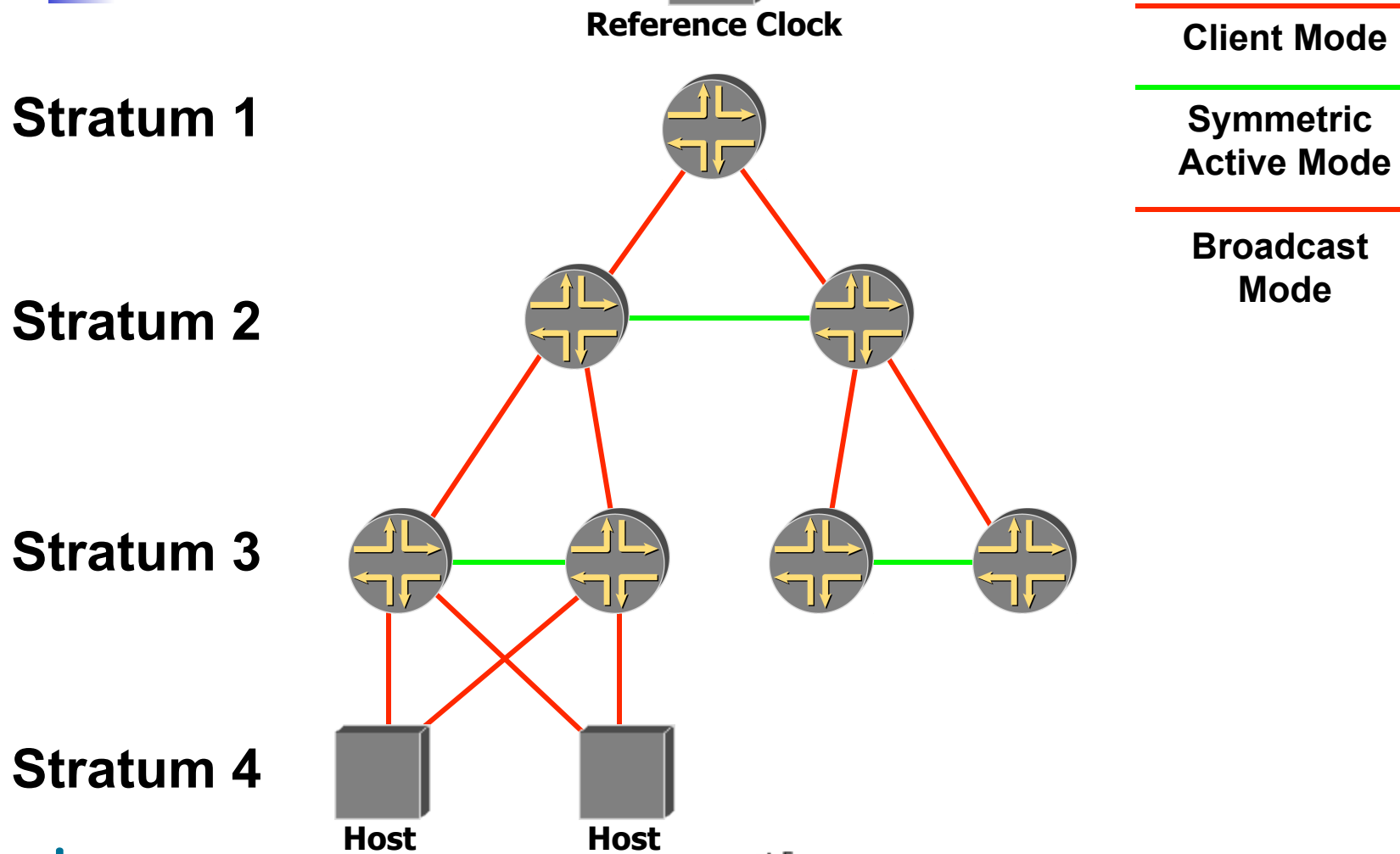


# Three NTP Modes

---

- Three modes:
  - Client mode
    - Client synchronizes local time one way to remote server
  - Symmetric active mode
    - Equal peer systems synchronize each other's local time
  - Broadcast mode
    - Server sends periodic broadcast/multicast messages on broadcast-capable media
    - Clients receive broadcast/multicast messages and synchronize local time

# NTP Hierarchy





# Stratum

---

Stratum	Min accuracy
1	$1.0 \times 10^{-11}$
2	$1.6 \times 10^{-8}$
3	$4.6 \times 10^{-6}$



# NTP Security

---

- NTP security
  - NTP relies on the number of connected hosts to:
    - Receive accurate time information
    - Isolate participants whose clock is incorrect
  - NTP supports MD5 and DES authentication
  - NTP without authentication on the public network is subject to spoofing
    - Create the appropriate filters to block incoming unsolicited information
    - Consider using the management network for NTP traffic





# NTP Boot Server

---

- NTP particulars:
  - NTP will not synchronize with a peer whose time is very different
    - Tiny offsets are adjusted normally
    - Small offsets are *slewed* (adjusted slowly)
    - Larger offsets are *stepped* (set anew)
    - Huge offsets are rejected outright
  - To synchronize the initial time:
    - Use an NTP boot server
    - When the router is booted a request is issued to the boot server to get the initial reference time

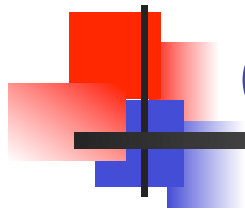
```
[edit system]
```

```
lab@R1# show
```

```
ntp {
```

```
boot-server 10.1.10.2;
```





# Client Configuration

---

```
[edit system]
lab@R1# show
ntp {
    boot-server 10.1.10.2;
    server 10.1.10.2 version 3
prefer;
    server 10.1.9.2;
```



# Symmetric Active Mode Configuration

---

```
[edit system]
lab@R1# show
ntp {
    boot-server 10.1.10.2;
    peer 10.1.10.2 version 3 prefer;
    peer 10.1.9.2;
}
```



# Broadcast Mode Configuration

---

```
[edit system]
```

```
lab@R1# show
```

```
ntp {
```

```
    boot-server 10.1.10.2;
```

```
    server 10.1.10.2 version 3
```

```
prefer;
```

```
    peer 10.1.9.2;
```

```
    broadcast 224.0.1.1;
```

```
    broadcast 10.1.2.255 version 3;
```



# Broadcast Client Configuration

---

```
[edit system]
lab@R1# show
ntp {
    boot-server 10.1.10.2;
    broadcast-client
}
```

```
[edit system]
lab@R1# show
ntp {
    boot-server 10.1.10.2;
    multicast-client
}
```



# Authentication

---

- Authentication of time synchronization
  - All NTP modes can use authenticated connections
    - Prevents spoofing
  - Supports two types of encrypted/hashed authentication algorithms
    - DES
    - MD5
  - Easy to configure
    - No real reason not to use authentication



# Utilizing Packet Capture

- SPAN/RSPAN (6500/7600, 4K, 2900,,), copy/capture VACLs (6500/7600), IP Traffic Export (software-based routers) are all used to get packets to analysis systems
- SPAN/RSPAN and copy/capture VACLs do not have measurable performance impact; IP Traffic Export can delay processing of traffic outbound from the router, based upon the volume of traffic to be replicated
- A \*NIX box running tcpdump is a common method of capturing packets, with analysis performed offline using additional open-source tools such as Ethereal
- The Cisco NAM-2 captures packets via SPAN/RSPAN or copy/capture VACLs on the 6500/7600; it can perform basic on-board analysis, but captures are typically saved and downloaded for use in Ethereal, Network General Sniffer, etc.
- Packet capture is generally undertaken after a macro-level indication of an issue via SNMP, NetFlow, etc.



## Utilizing Packet Capture (cont.)

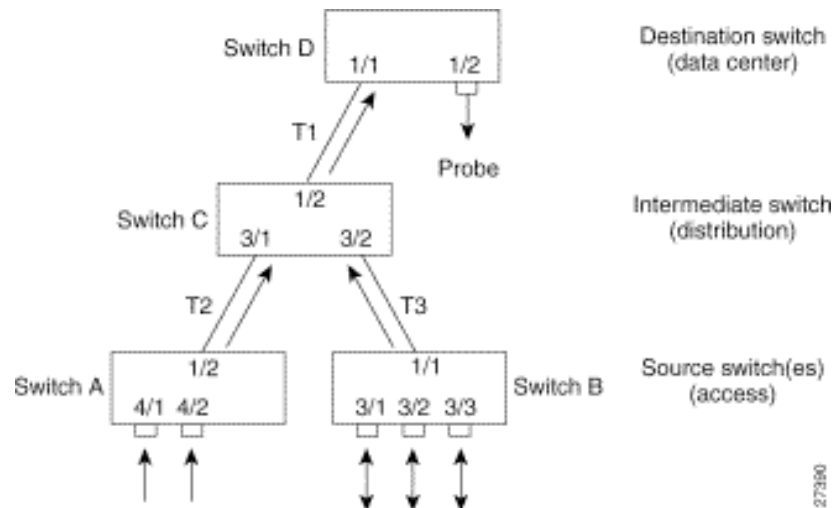
- Packet capture should take place at key points in the topology such as distribution gateways, IDC switch meshes, desktop access switch meshes, and in some cases, the core
- It is important to be as specific as possible when capturing packets; at high rates of speed, the amount of information can be overwhelming
- There's lots of garbage out there - 'weird' packets are often perfectly explicable, in context
- It's extremely important to ensure that traffic is captured bidirectionally - or, if this isn't possible, the observer must know about the unidirectionality of the capture and take it into account when analyzing the captured traffic

Conversely, it's important to avoid capturing duplicate traffic, especially in complex topologies



# Packet Capture Example - CatOS

## RSPAN



Switch	Ports	RSPAN VLAN	Direction	RSPAN CLI Commands
A (source)	4/1, 4/2	901	Ingress	<b>set rspan source 4/1-2 901 rx</b>
B (source)	3/1, 3/2, 3/3	901	Bidirectional	<b>set rspan source 3/1-3 901</b>
C (intermediate)	-	901	-	No RSPAN CLI command needed
D (destination)	1/2	901	-	<b>set rspan destination 1/2 901</b>

# Packet Capture Example - tcpdump

```
tcpdump -lllvvnxxxXX -s 1500 -i en1
```

```
tcpdump: listening on en1, link-type EN10MB (Ethernet), capture size 1500 bytes
```

```
..
```

```
07:10:25.740130 IP (tos 0x0, ttl 64, id 15460, offset 0, flags [none], length: 59) 10.25.7.122.58607  
> 172.17.168.183.53: [udp sum ok] 15197+ A? delta.mac.com. (31)
```

```
0x0000: 0005 31a0 3414 000d 93f0 c5bc 0800 4500 ..1.4.....E.  
0x0010: 003b 3c64 0000 4011 d8bd 0a19 077a ab46 .;<d..@.....z.F  
0x0020: a8b7 e4ef 0035 0027 bfb9 3b5d 0100 0001 .....5.'...;]....  
0x0030: 0000 0000 0000 0564 656c 7461 036d 6163 .....delta.mac  
0x0040: 0363 6f6d 0000 0100 01 .....com.....
```

```
07:10:25.829524 IP (tos 0x0, ttl 56, id 14524, offset 0, flags [DF], length: 256) 172.17.168.183.53  
> 10.25.7.122.58607: [udp sum ok] 15197 q: A? delta.mac.com. 2/4/4 delta.mac.com. CNAME  
idisk.mac.com., idisk.mac.com. A 17.250.248.77 ns: mac.com. NS nserver4.apple.com., mac.com. NS  
nserver.apple.com., mac.com. NS nserver2.apple.com., mac.com. NS nserver3.apple.com. ar:  
nserver.apple.com. A 17.254.0.50, nserver2.apple.com. A 17.254.0.59, nserver3.apple.com. A  
17.112.144.50, nserver4.apple.com. A 17.112.144.59 (228)
```

```
0x0000: 000d 93f0 c5bc 0005 31a0 3414 0800 4500 .....1.4...E.  
0x0010: 0100 38bc 4000 3811 a3a0 ab46 a8b7 0a19 ..8.@.8....F....  
0x0020: 077a 0035 e4ef 00ec c78e 3b5d 8180 0001 .z.5.....;]....  
0x0030: 0002 0004 0004 0564 656c 7461 036d 6163 .....delta.mac  
0x0040: 0363 6f6d 0000 0100 01c0 0c00 0500 0100 .com.....  
0x0050: 0006 ea00 0805 6964 6973 6bc0 12c0 2b00 .....idisk...+.  
0x0060: 0100 0100 000d da00 0411 faf8 4dc0 1200 .....M...  
0x0070: 0200 0100 0222 ab00 1108 6e73 6572 7665 .....".nserve  
0x0080: 7234 0561 7070 6c65 c016 c012 0002 0001 r4.apple.....  
0x0090: 0002 22ab 000a 076e 7365 7276 6572 c058 ..".nserver.X  
0x00a0: c012 0002 0001 0002 22ab 000b 086e 7365 .....".nse  
0x00b0: 7276 6572 32c0 58c0 1200 0200 0100 0222 rver2.X....."  
0x00c0: ab00 0b08 6e73 6572 7665 7233 c058 c06c ....nserver3.X.1  
0x00d0: 0001 0001 0001 86fa 0004 11fe 0032 c082 .....2..  
0x00e0: 0001 0001 0001 86fa 0004 11fe 003b c099 .....;..  
0x00f0: 0001 0001 0001 86fa 0004 1170 9032 c04f .....p.2.O  
0x0100: 0001 0001 0002 9995 0004 1170 9032 c04f .....p.;
```

# Packet Capture Example - Ethereal

Packets: 1-1000 of 1470

Pkt	Time(s)	Size	Source	Destination	Protocol	Info
1	0.000	437	nam-6506.embu-mlab...	dhcp-171-69-125-166....	HTTP	HTTP/1.1 302 Found
2	0.006	68	nam-6506.embu-mlab...	dhcp-171-69-125-166....	TCP	http > 3953 [ACK] Seq=2086005762 Ack=305177...
3	0.048	70	core2-e0-1.embu-mla...	ALL-ROUTERS.MCAS...	HSRP	Hello (state Active)
4	0.057	68	embu-callmgr1.embu-...	192.168.79.42	MGCP	200 2303453
5	0.069	1222	nam-6506.embu-mlab...	dhcp-171-69-125-166....	HTTP	HTTP/1.1 200 OK
6	0.069	1222	nam-6506.embu-mlab...	dhcp-171-69-125-166....	HTTP	Continuation
7	0.075	1222	nam-6506.embu-mlab...	dhcp-171-69-125-166....	HTTP	Continuation
8	0.075	1222	nam-6506.embu-mlab...	dhcp-171-69-125-166....	HTTP	Continuation
9	0.075	1222	nam-6506.embu-mlab...	dhcp-171-69-125-166....	HTTP	Continuation
10	0.084	1222	nam-6506.embu-mlab...	dhcp-171-69-125-166....	HTTP	Continuation

**Packet** Number: 7 - Time: May 16, 2003 12:47:17.357 - Packet Length: 1222 bytes - Capture Length: 1218 bytes

- + **ETH** Ethernet II, Src: 00:d0:d3:9d:73:d0, Dst: 00:30:94:fd:c6:17
- + **VLAN** 802.1q Virtual LAN
- + **IP** Internet Protocol, Src Addr: nam-6506.embu-mlab.cisco.com (192.168.76.12), Dst Addr: dhcp-171-69-125-166.cisco.com (171...
- + **TCP** Transmission Control Protocol, Src Port: http (80), Dst Port: 3953 (3953), Seq: 2086008082, Ack: 3051775911, Len: 1160
- **HTTP** Hypertext Transfer Protocol
- HTTP** Data (1160 bytes)

```

0000  00 30 94 fd c6 17 00 d0 d3 9d 73 d0 81 00 00 3c  .0.....s....<
0010  08 00 45 00 04 b0 0d 40 40 00 3f 06 f4 67 c0 a8  ..E....@?.?.g..
0020  4c 0c ab 45 7d a6 00 50 0f 71 7c 55 f5 12 b5 e6  L..E)..P.q|U....
0030  67 a7 50 10 43 98 0a 57 00 00 25 22 20 62 6f 72  g.P.C..W..%" bor
0040  64 65 72 3d 22 30 22 20 63 65 6c 6c 73 70 61 63  der="0" cellspac
0050  69 6e 67 3d 22 30 22 20 63 65 6c 6c 70 61 64 64  ing="0" cellpadd
  
```

Source: <http://www.ethereal.com>



# References

---

- DoS detection:

- "Tackling Network DoS on Transit Networks": David Harmelin, DANTE, March 2001 (Describes a detection method based on NetFlow) [<http://www.dante.net/pubs/dip/42/42.html>]
- "Inferring Internet Denial-of-Service Activity": David Moore et al, May 2001; (Described a new method to detect dos attacks, based on the return traffic from the victims, analysed on A /8 network; very interesting reading) [<http://www.caida.org/outreach/papers/backscatter/index.xml>]
- "The Spread of the Code Red Worm": David Moore, CAIDA, July 2001 (Using the above to detect how this worm spread across the Internet) [<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.enteract.com/~robt/Docs/Articles/tracking-spoofed.html>]

# Packet Capture Examples

Packets: 1-1000 of 1470

Pkt	Time(s)	Size	Source	Destination	Protocol	Info
1	0.000	437	nam-6506.embu-mlab...	dhcp-171-69-125-166...	HTTP	HTTP/1.1 302 Found
2	0.006	68	nam-6506.embu-mlab...	dhcp-171-69-125-166...	TCP	http > 3953 [ACK] Seq=2086005762 Ack=305177...
3	0.048	70	core2-e0-1.embu-mla...	ALL-ROUTERS.MCAS...	HSRP	Hello (state Active)
4	0.057	68	embu-callmgr1.embu...	192.168.79.42	MGCP	200 2303453
5	0.069	1222	nam-6506.embu-mlab...	dhcp-171-69-125-166...	HTTP	HTTP/1.1 200 OK
6	0.069	1222	nam-6506.embu-mlab...	dhcp-171-69-125-166...	HTTP	Continuation
7	0.075	1222	nam-6506.embu-mlab...	dhcp-171-69-125-166...	HTTP	Continuation
8	0.075	1222	nam-6506.embu-mlab...	dhcp-171-69-125-166...	HTTP	Continuation
9	0.075	1222	nam-6506.embu-mlab...	dhcp-171-69-125-166...	HTTP	Continuation
10	0.084	1222	nam-6506.embu-mlab...	dhcp-171-69-125-166...	HTTP	Continuation

Packet Number: 7 - Time: May 16, 2003 12:47:17.357 - Packet Length: 1222 bytes - Capture Length: 1218 bytes

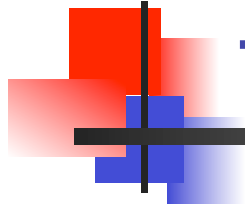
- + **ETH** Ethernet II, Src: 00:d0:d3:9d:73:d0, Dst: 00:30:94:fd:c6:17
- + **VLAN** 802.1q Virtual LAN
- + **IP** Internet Protocol, Src Addr: nam-6506.embu-mlab.cisco.com (192.168.76.12), Dst Addr: dhcp-171-69-125-166.cisco.com (171...)
- + **TCP** Transmission Control Protocol, Src Port: http (80), Dst Port: 3953 (3953), Seq: 2086008082, Ack: 3051775911, Len: 1160
- **HTTP** Hypertext Transfer Protocol
- HTTP** Data (1160 bytes)

```
0000  00 30 94 fd c6 17 00 d0 d3 9d 73 d0 81 00 00 3c  .0.....s....<
0010  08 00 45 00 04 b0 0d 40 00 3f 06 f4 67 c0 a8  ..E....00.?.g..
0020  4c 0c ab 45 7d a6 00 50 0f 71 7c 55 f5 12 b5 e6  L..E)...P.q|U....
0030  67 a7 50 10 43 98 0a 57 00 00 25 22 20 62 6f 72  g.P.C..W..%" bor
0040  64 65 72 3d 22 30 22 20 63 65 6c 6c 73 70 61 63  der="0" cellspac
0050  69 6e 67 3d 22 30 22 20 63 65 6c 6c 70 61 64 64  ing="0" cellpadd
```

Wealth of  
information, L1-L7  
raw data for  
analysis

Source: <http://www.ethereal.com>, Cisco Systems, Inc.





# Tell Me Where to Start From?

---

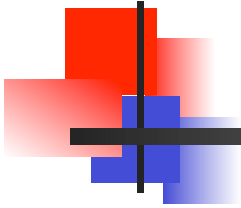
1. NetFlow or jflow enablement on the network elements
2. NetFlow or jflow data correlation and analysis
3. SNMP / RMON [SNMP more prevalent]
  1. CPU / Memory util
  2. Link usage and display with MRTG
4. SysLog collection and analysis
5. Monitoring to Routing, DNS queries, etc. [BGP, DNS]
6. Local and remote packet capture facility [Most have it today with sniffer, ethereal]



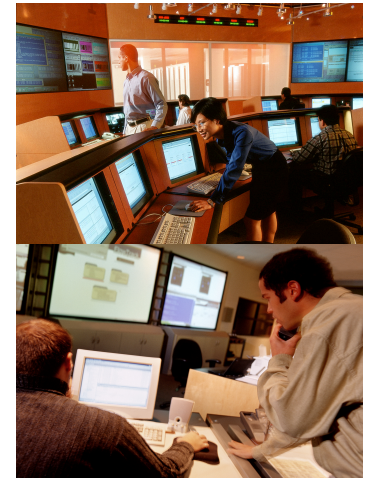
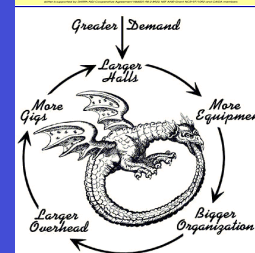
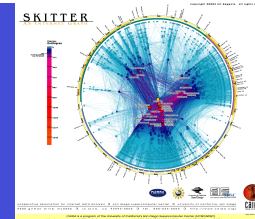
# Homework from Total Visibility

---

- Define telemetry strategy—**ASAP**
  - **Local and remote**
- Need to start deployment **today** where the most bang for the buck is offered. However, the end goal is to achieve the holistic view
- Telemetry: Deploy, Understand and Practice
  - For any security event – Proactive Telemetry or telemetry during the incident, if 'SECOPS' trained then they can use it with familiarity of 'back of their hand'
- Telemetry builds foundation to be successful with all the other 5 of 6 steps methodology



# MPLS / L3VPN Security







# Before we start...

---

- Mainly of interest to providers/ISP/Carriers
  - Some interest in enterprise
- To support MPLS in your network you **MUST** have
  - Fully working IP network. If it's broken MPLS won't fix it.
  - Hardware and Software support. Depends on vendors
    - Juniper
      - All our routers (M-Series, T-Series, J-Series, E-Series)
    - Cisco
      - CEF support

# Things I want you to know

- MPLS is a tool to solve problems
  - Not everyone has the same problems or pain
- In other words reason to deploy (choose 1+)
  - Traffic Engineering
  - Traffic Protection
  - Provider provisioned VPN's
    - Layer 3 and/or Layer 2
- Or in other words
  - Save money
  - Make money



# What is MPLS?

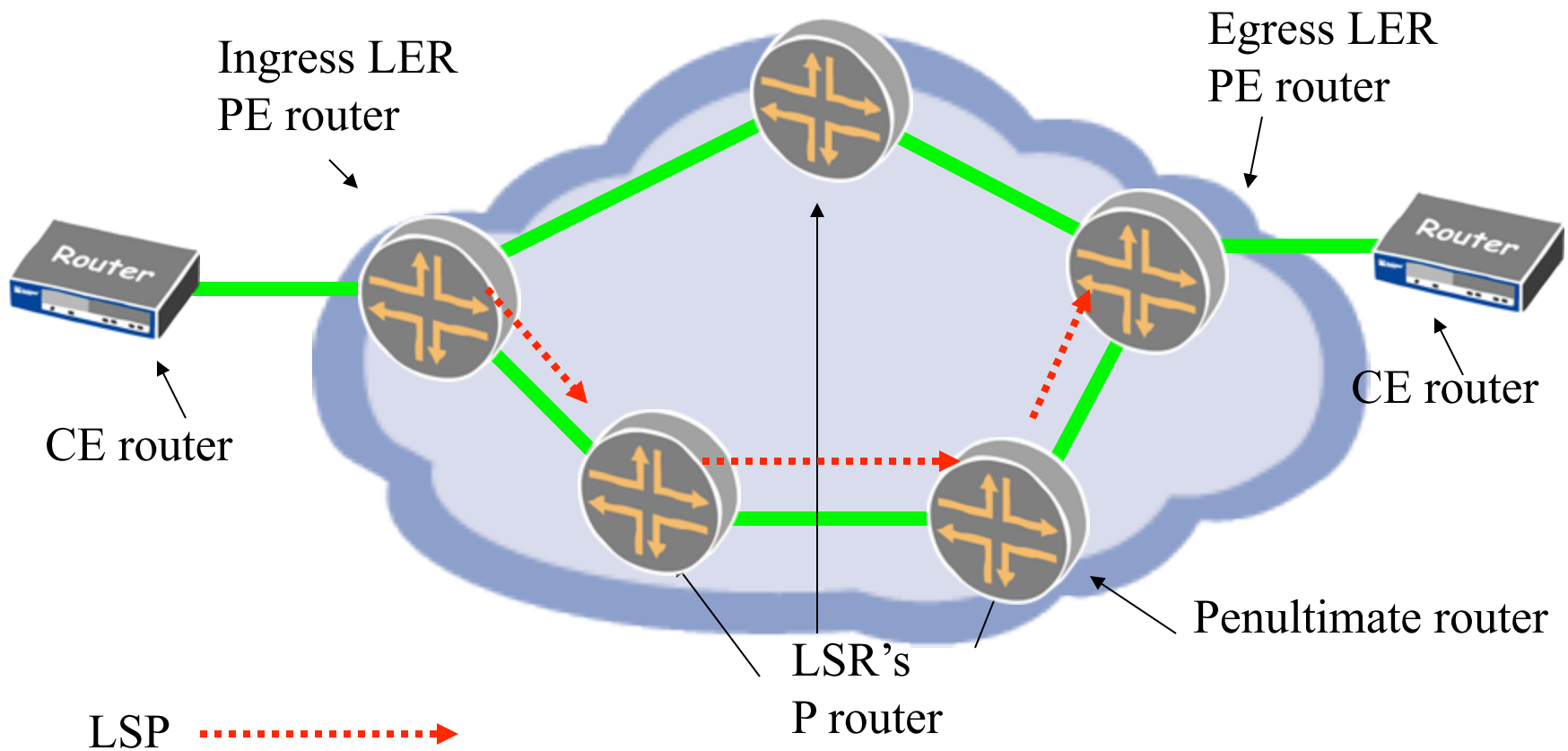
It's a tunnel!

- Multiprotocol Label Switching
- Connection Orientated Virtual Circuits over IP implemented with label switching
- Grew out of
  - Cisco's Tag switching
  - Ipsilon (Nokia) IP switching
  - IBM ARIS
  - 3Com's FAST IP
- Expanding area's of application
  - Cost savings
  - New services
- Promise of Multiprotocol Unification (Core NOT edge)
- Defined by RFC 3031, RFC 3032



# MPLS Terminology

-An LSP is a unidirectional flow of traffic



# Push, Pop, Swap

- Push



- Pop

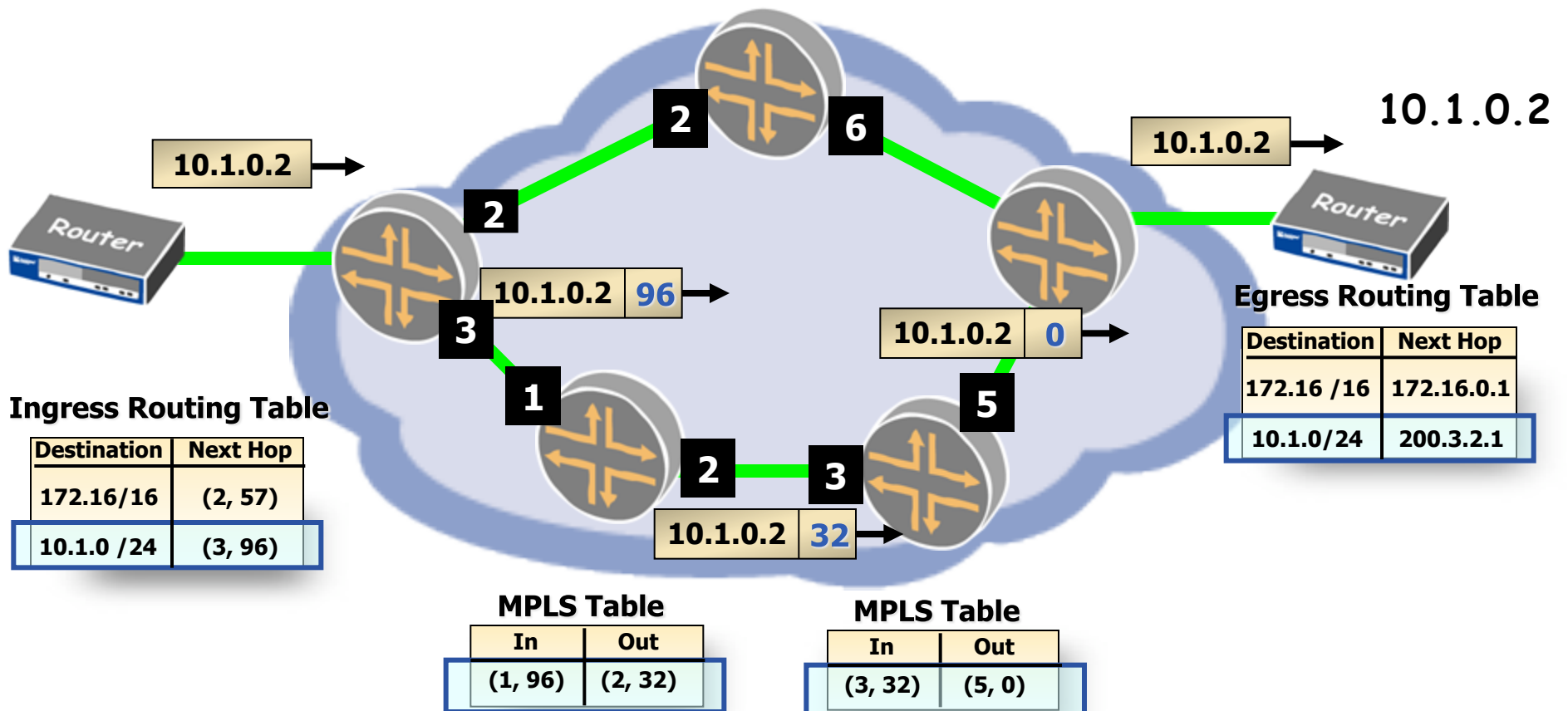


- Swap



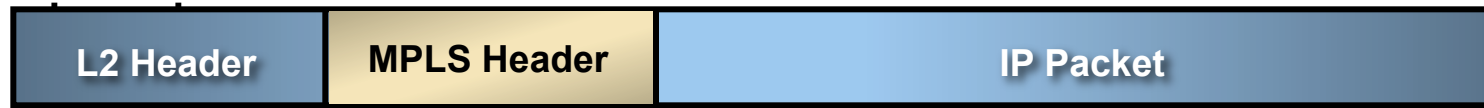
# MPLS Forwarding Plane

MPLS Table	
In	Out
(2, 57)	(6, 0)



# Labeled Packets

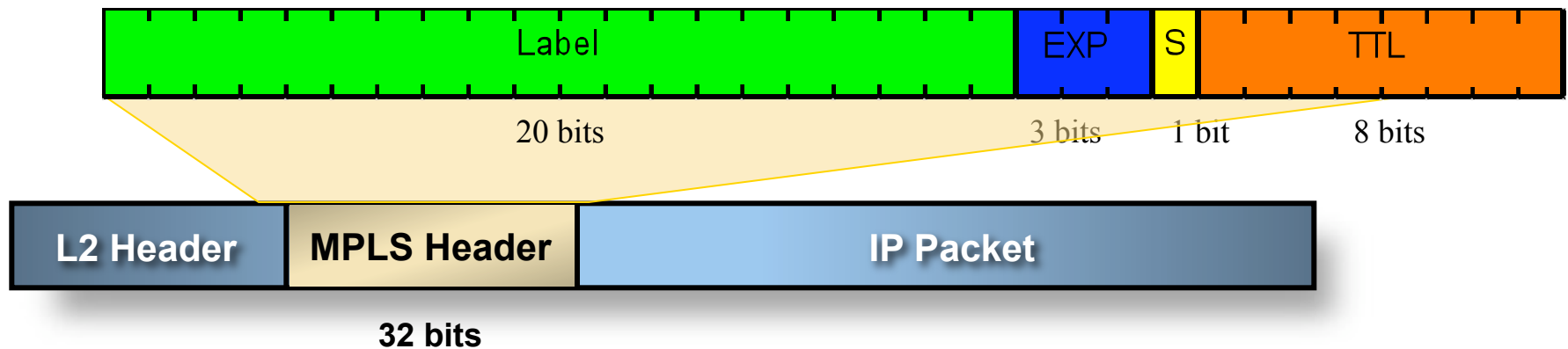
- MPLS header is prepended to packet with a *push* operation at ingress node
  - Label is added immediately after Layer 2 encapsulation



32-Bit  
MPLS shim Header

- Packet is restored at the end of the LSP with a *pop* operation
  - Normally the label stack is popped at penultimate node

# The Label



- Label
  - Used to identify virtual circuit
- EXP
  - Experimental. Currently this is used to identify class of service (CoS)
- S (Stack Bit)
  - Used to indicate if there is another label inside this packet or is it the original encapsulated data
- TTL



# Example - Ethernet



0 0 1 0 1 1 1 1 0 1 0 0 0 1 0 1 1 1

My Web Page

TCP | port = 80 (www)

IP Header | Protocol = TCP

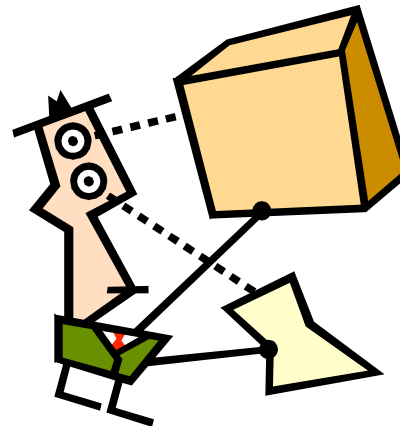
Label = 23 | EXP = BE | S = 0 | TTL = 254

Label = 47 | EXP = BE | S = 1 | TTL = 240

Dest. MAC   Src. MAC   Type = 8347

# FEC – Forwarding Equivalency class

- All traffic with the same FEC will follow the same path and experience same level of service
- E.g. of FEC
  - Destination IP address
  - BGP next hop
  - VPN membership
  - Source address
  - Any combination of above



Packet

Label

# Signaling

- Protocols that are used to setup maintain and tear down LSP's.
- Can behave differently depending on function
- Let's describe a language / concepts to understand these differences in operation

Tell the routers what label to use on each hop!



# Signaling Protocols

- LDP
  - Label Distribution Protocol
- RSVP-TE
  - Resource Reservation Protocol with Traffic Engineering Extensions
- MBGP
  - Multi-protocol BGP

Which you use depends on why you are using MPLS! Maybe you need all of them!



# Which to choose...

RFC's mandate LDP support for L3 VPN's

- Traffic Engineering, Traffic Protection
  - RSVP
  - Link State protocol
- VPN's
  - LDP or RSVP (all LSR's)
  - MBGP (PE's only)
- Why use LDP at all?
  - Configuration scaling
  - LDP configuration is "per box"
  - RSVP configuration is "per LSP"



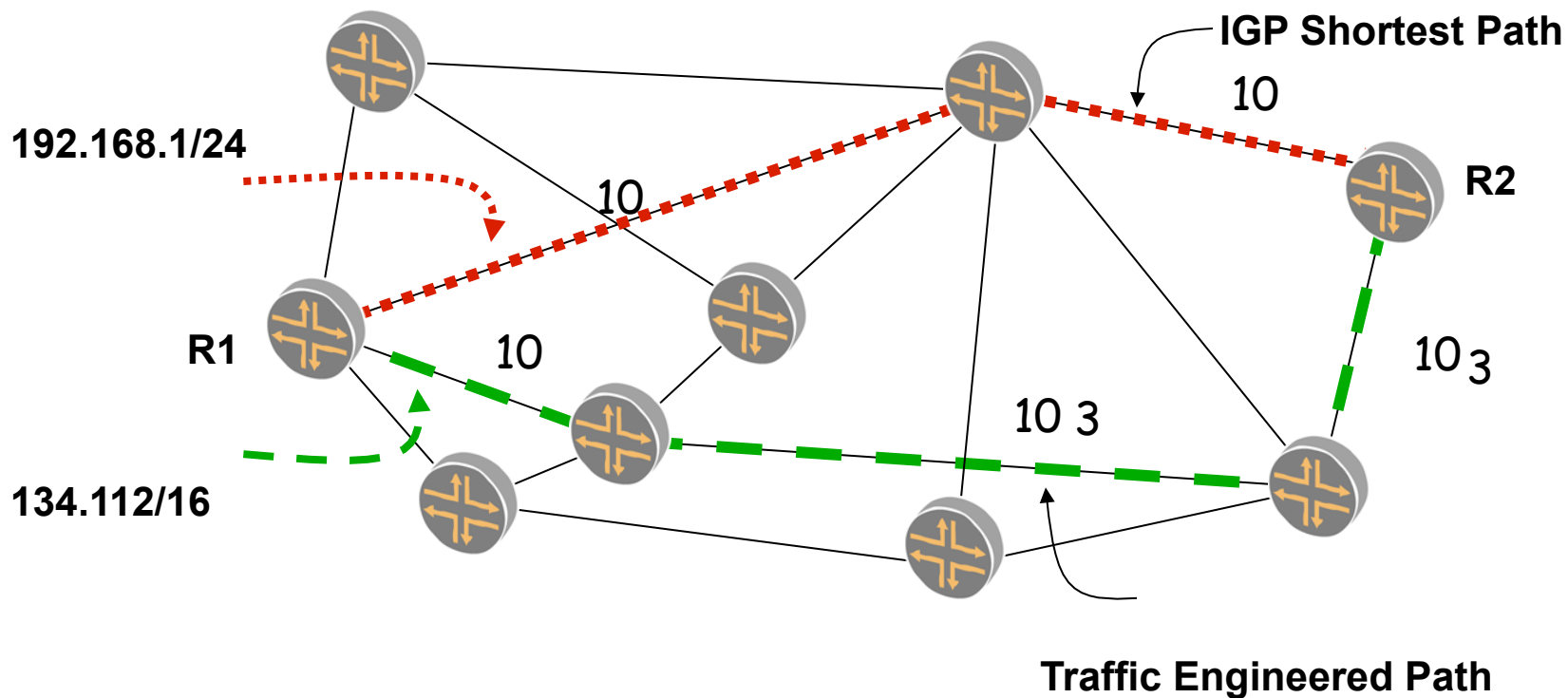


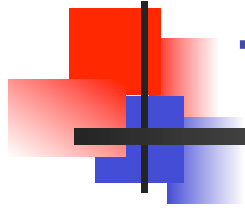
# Traffic Engineering Defined

---

- Sub Optimal routing
- Network Engineering is putting bandwidth where the traffic is. Traffic Engineering is putting the traffic where the bandwidth is!
- To meet one of two requirements
  - To better utilize network capacity and resources.
  - To put traffic on a path that can support it's requirements
- Incorporate Traffic Protection to achieve SONET-like failure recovery.

# MPLS-Based Traffic Engineering





# Traffic Engineering Options

---

- Can we do this another way
  - IGP metrics ☹
  - Flow = all traffic with same destination
- MPLS because
  - Granularity of flows
  - Flow = all traffic with same FEC
  - One network for all services
  - Less expensive



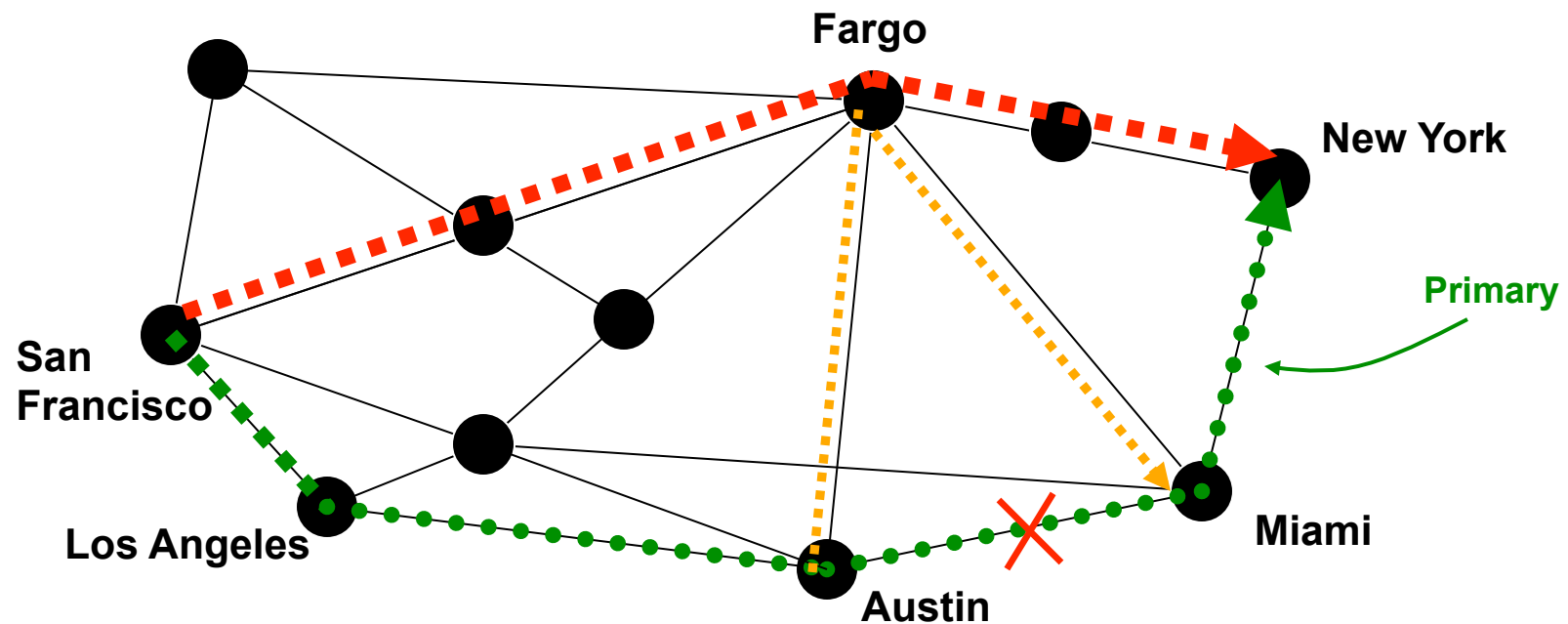


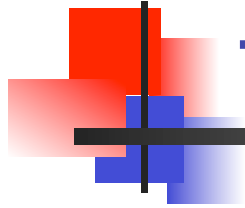
# Traffic Protection

---

- Working definition
  - Reduce time of disruption
  - Reduce Packet Loss
  - “SONET like” sub millisecond recovery under failure conditions
- Can we do this another way
  - SONET/SDH
  - Lower IGP timers
- MPLS because
  - No extra capital – config change only
  - Pick which traffic needs it
  - One network for all services
  - Less expensive

# Traffic Protection – example





# Traffic Protection Variations

---

- Fast reroute
- Link Protection
- Link-Node Protection

# Layer 3 VPN

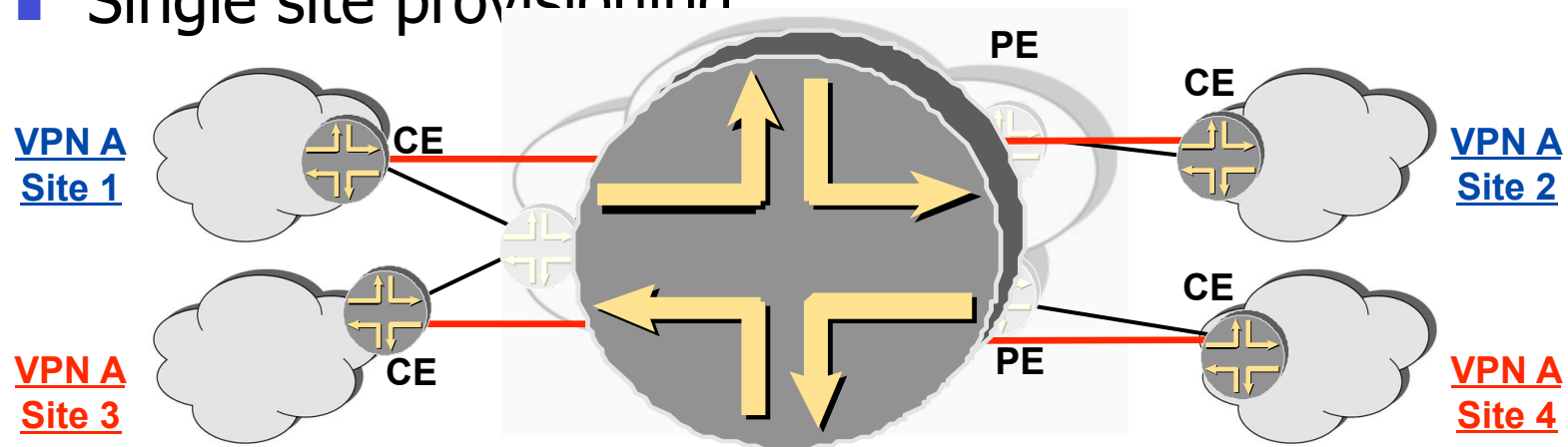
(2547bis BGP/MPLS VPN)

## Provider provisioned VPN

- ISP runs backbone for customer
  - Customer can be another ISP!
- Attractive to
  - Customer who do not want to run their own backbone
- Not attractive to
  - Customer who doesn't trust carrier
  - Customers who's jobs are threatened

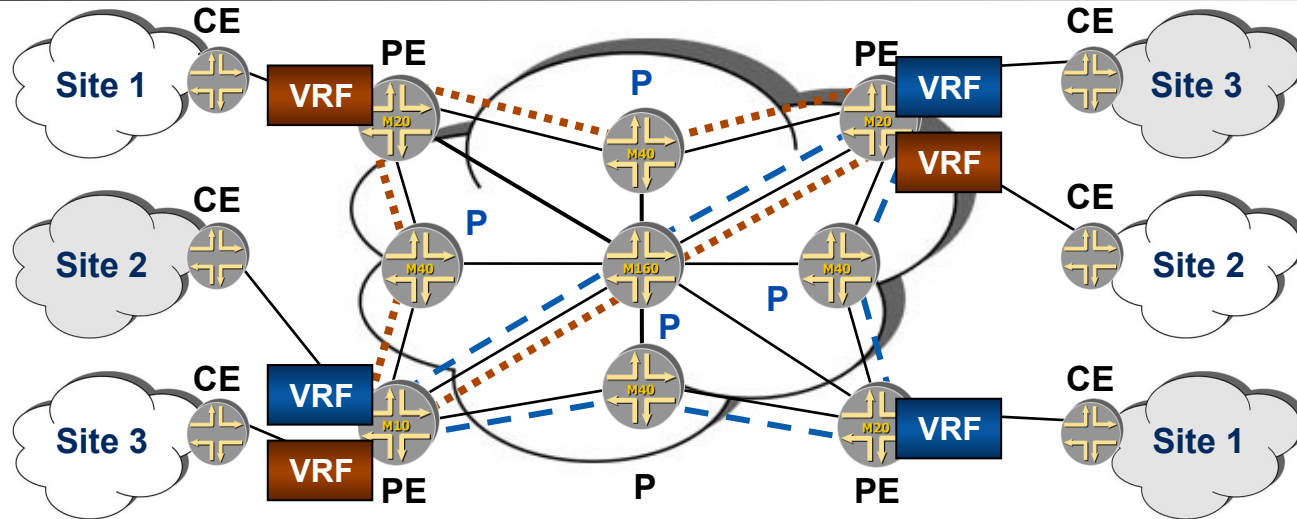
# Customer View of L3VPN

- Make the cloud look like a router
- Single site provisioning



# Layer 3 PP-VPNs: RFC 2547bis (1 of 2)

## Service Provider Network



### ■ Application: Outsource VPN

- PE router maintains VPN-specific forwarding tables for each of its directly connected VPNs
- Conventional IP routing between CE and PE routers
- VPN routes distributed using MP-BGP
  - Uses extended communities
- VPN traffic forwarded across provider backbone using MPLS

# Layer 3 PP-VPNs: RFC 2547bis (2 of 2)

---

- LDP or RSVP is used to set up PE-to-PE LSPs
- MP-BGP is used to distribute information about the VPN
  - Routing and reachability for the VPN
  - Labels for customer sites (tunneled in PE-PE LSP)
- Constrain connectivity by route filtering
  - Flexible, policy-based control mechanism



# L3 VPN Options

---

- Can we do it another way
  - Separate Physical routers
  - Separate Logical Routers
- MPLS because
  - Scaling
  - Single site provisioning
  - Less expensive





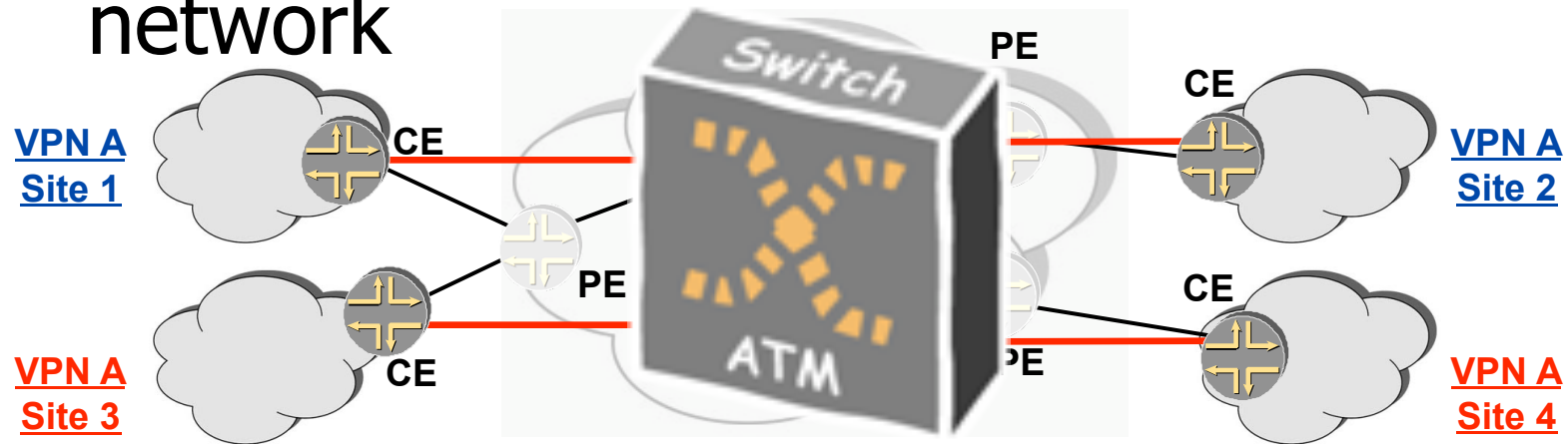
# Layer 2 VPN's

---

- Provider provisioned VPN
  - ISP runs backbone for customer
    - Customer can be another ISP!
- Attractive to
  - Customers who want to preserve current CE technology
  - Customers who don't trust provider with L3
  - Carriers who want to offer another service
- Not Attractive to
  - Customers who do not want to run their own backbone

# Customer View of L2VPN

- Make the cloud look like a ATM/FR network





# L2 VPN Options

---

- Can we do it another way?
  - Traditional ATM/FR/leased line infrastructure
- MPLS because
  - One network for all services
  - Less expensive
  - Scaling
  - Single site provisioning \*



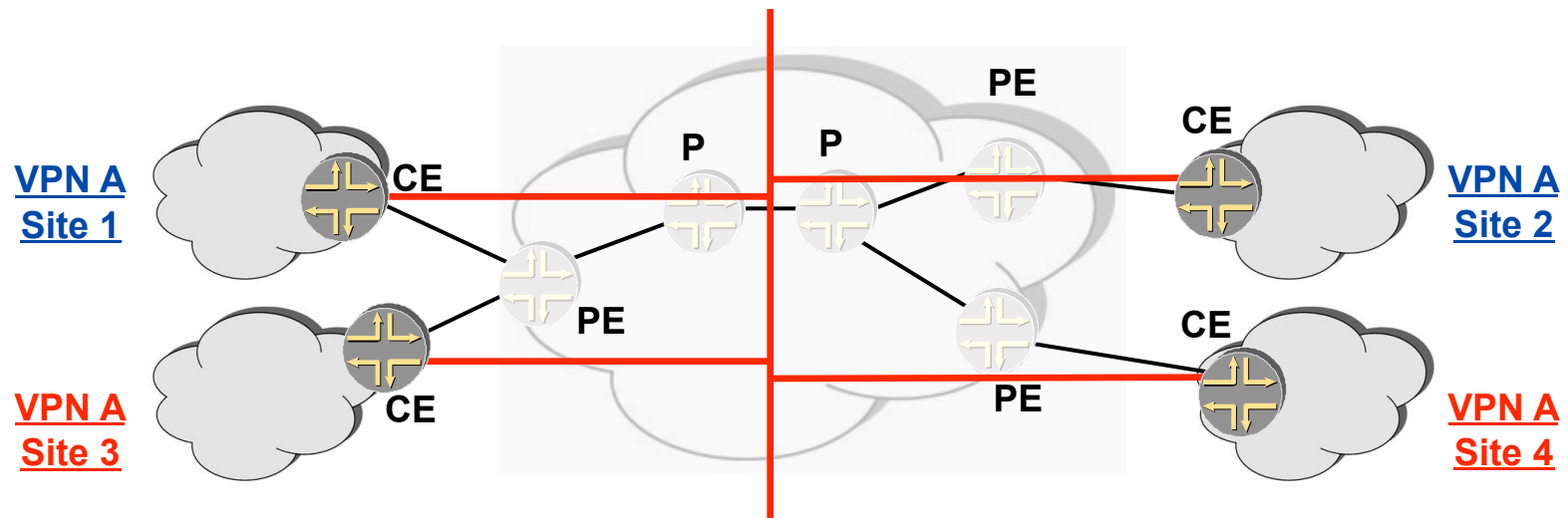
# VPLS

---

- Virtual Private LAN Service
- Attractive to
  - Customers who like ethernet as CE
  - Lots of locations close together with 'high' WAN bandwidth requirements (kiosks)
  - No routing required
- Not attractive to
  - Customers who like control and visibility of core.  
"what can I ping to identify fault-domain?"
  - Controlling broadcasts

# VPLS

- Make the cloud look like an ethernet switch





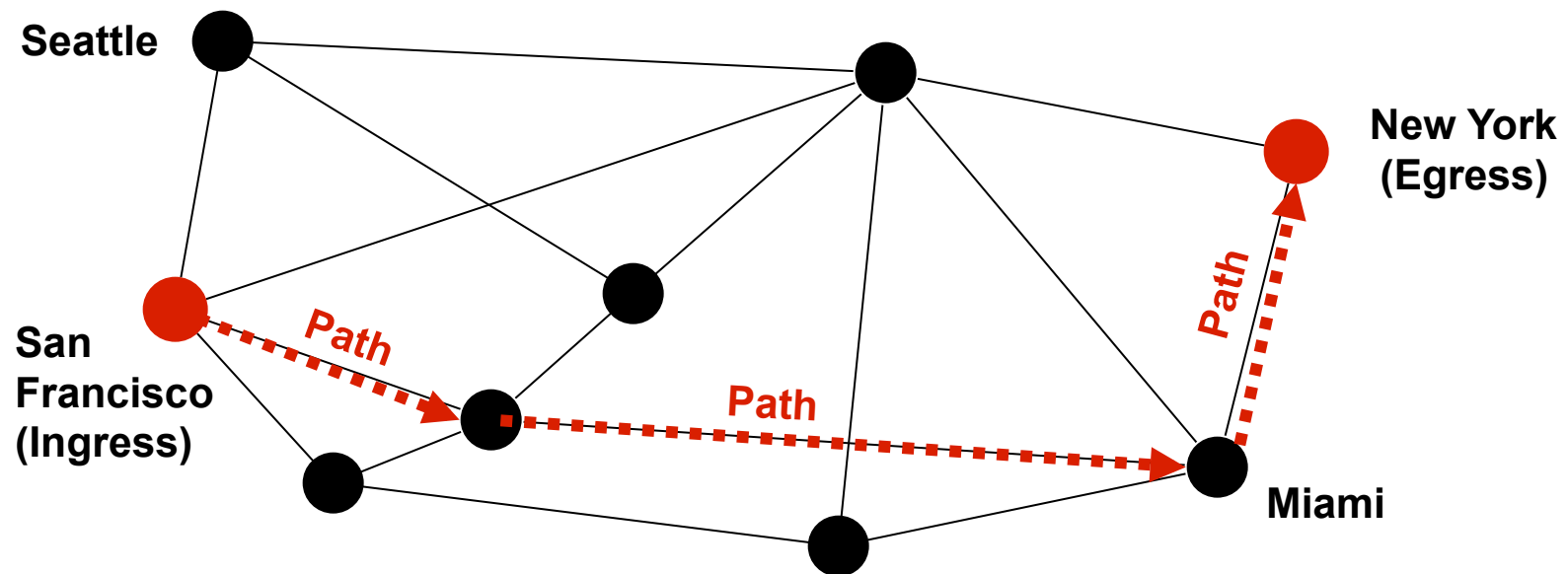
# VPLS Options

---

- Can we do it another way?
  - Separate physical switches tying all customer sites
  - VLAN's over layer 2 backbone
- MPLS because
  - Scaling
  - One network for all services
  - Less expensive

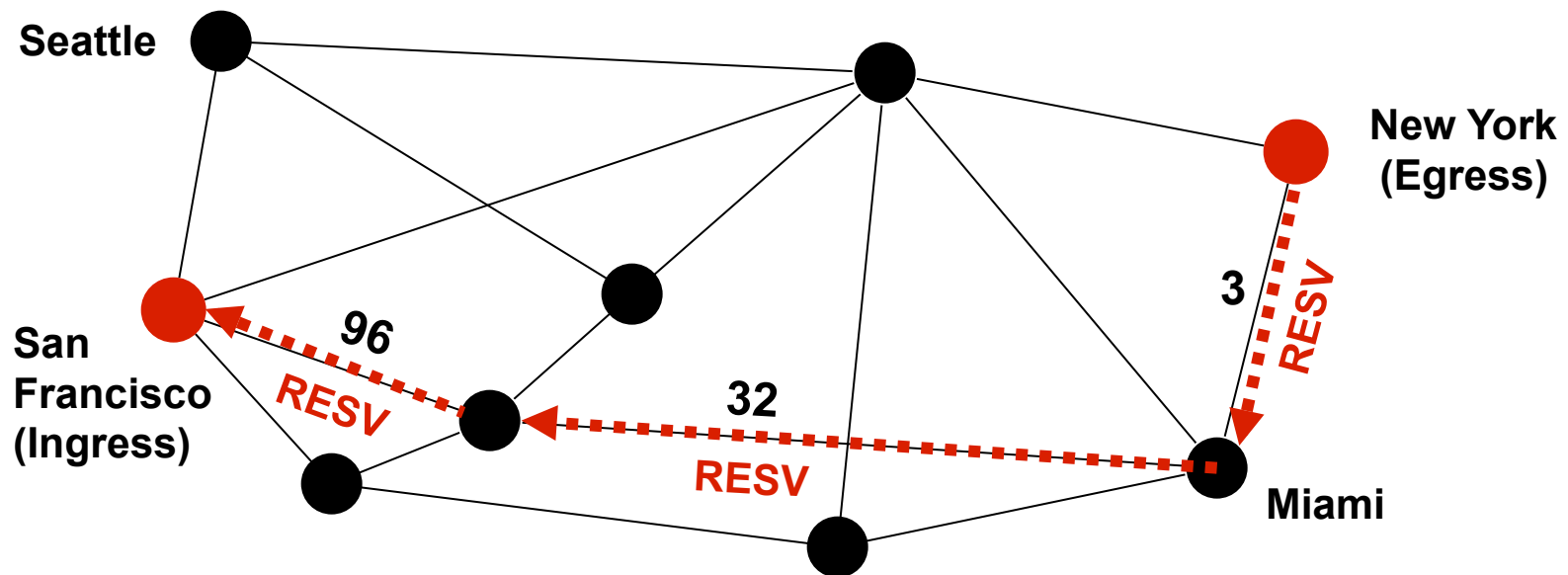
# RSVP Signaling Example: Path

RSVP sets up path from San Francisco to New York



# RSVP Signaling Example: Reservation

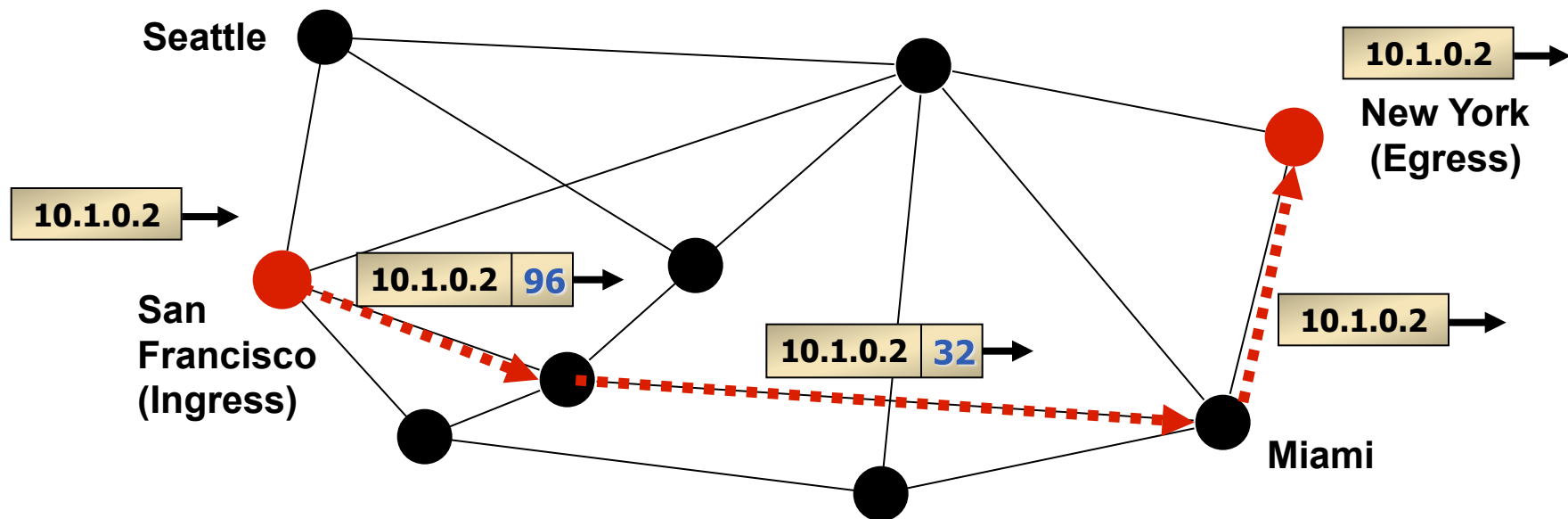
- The resv message visits each router on the path in reverse order
  - Labels assigned hop to hop in the upstream direction



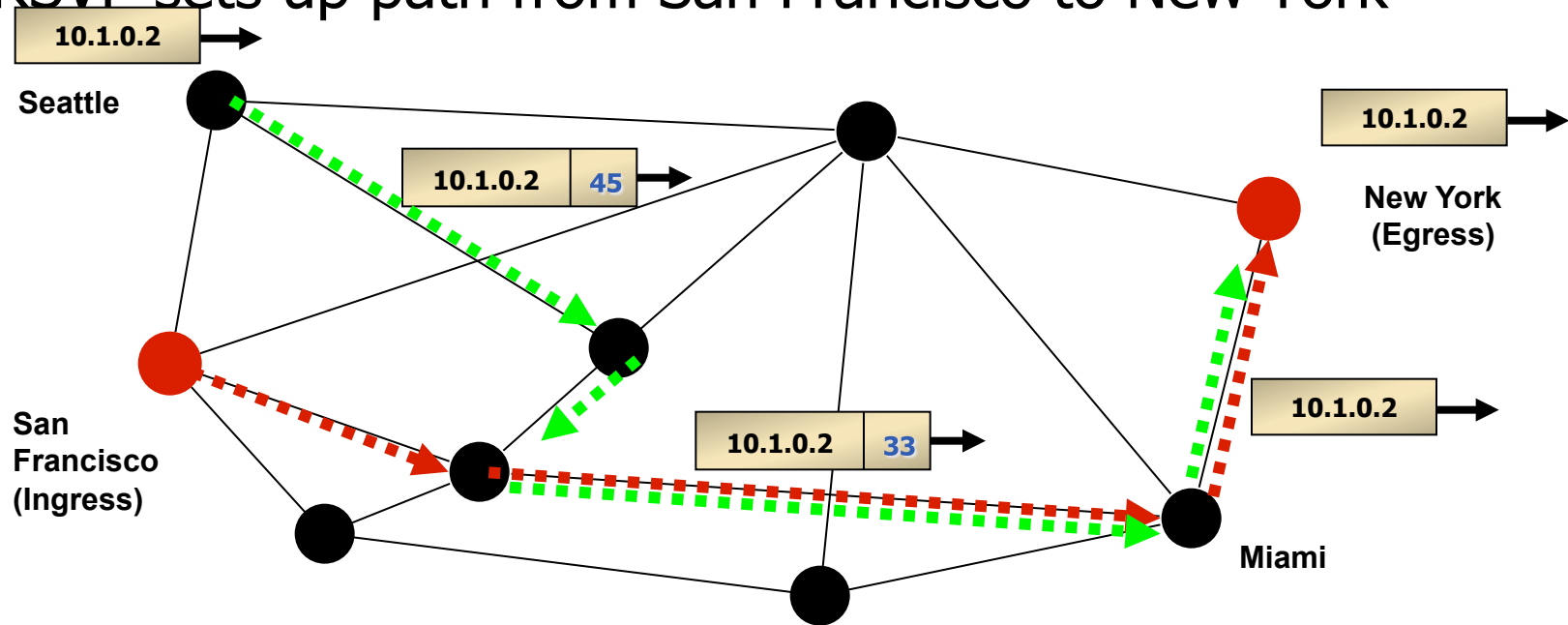


# RSVP Signaling Example: Forwarding

RSVP sets up path from San Francisco to New York

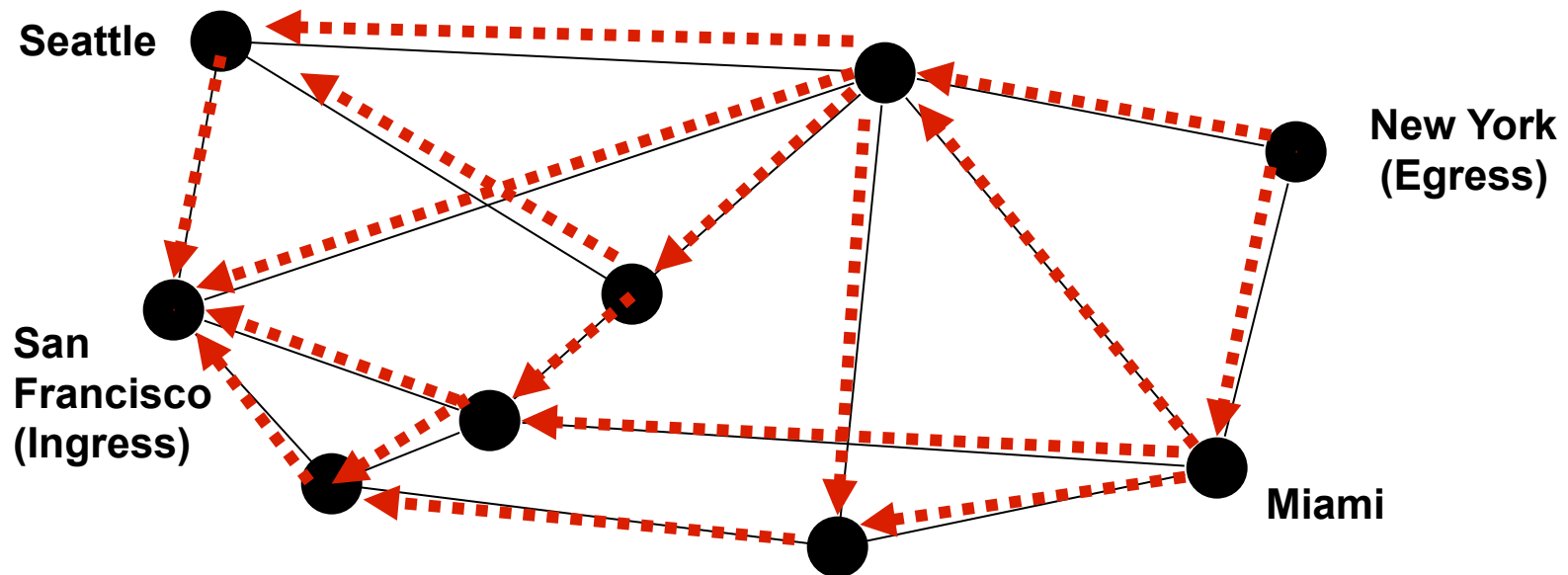


## RSVP sets up path from San Francisco to New York



# LDP Signaling Example: Label Binding

- Label Mappings are made for entries in the routing table
  - Labels assigned hop to hop in the upstream direction





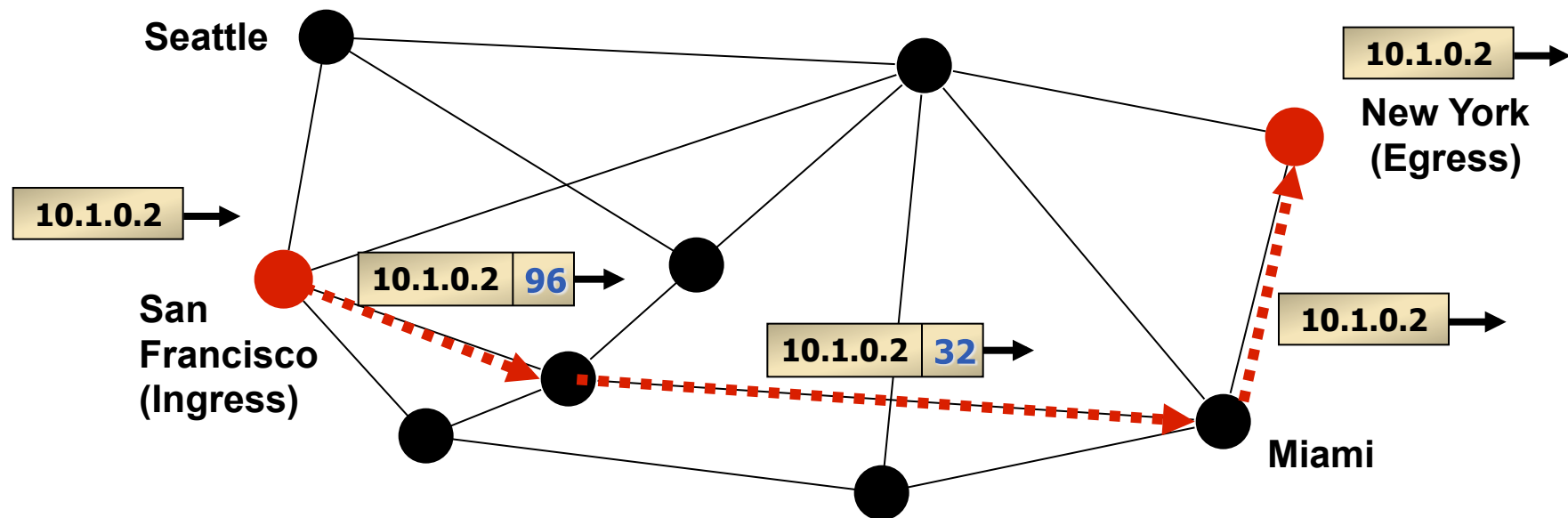
## For those who care!

---

- The last slide assumed LDP was operating in
  - Unsolicited Downstream mode
    - Not downstream-on-demand
  - Ordered Mode
    - Not Independent Mode
  - Liberal label retention
    - Not conservative

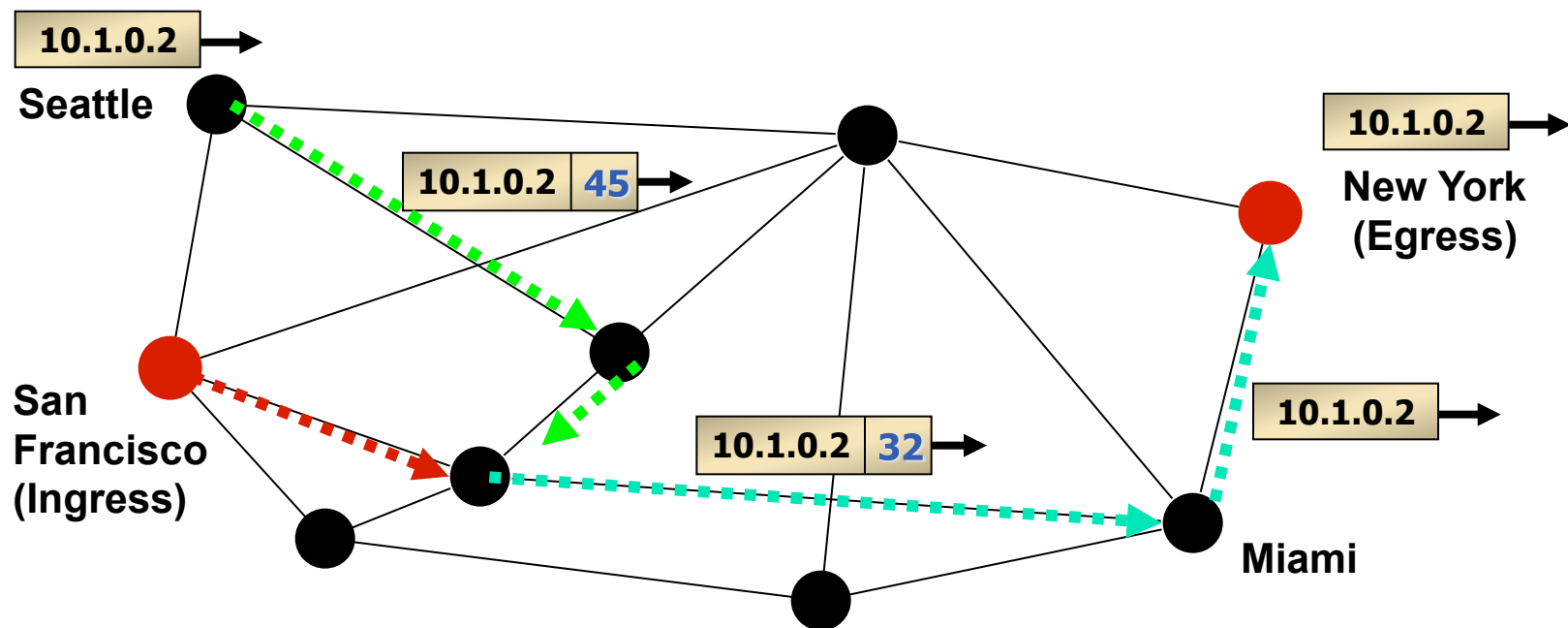
# LDP Signaling Example: Forwarding

LDP path available to egress



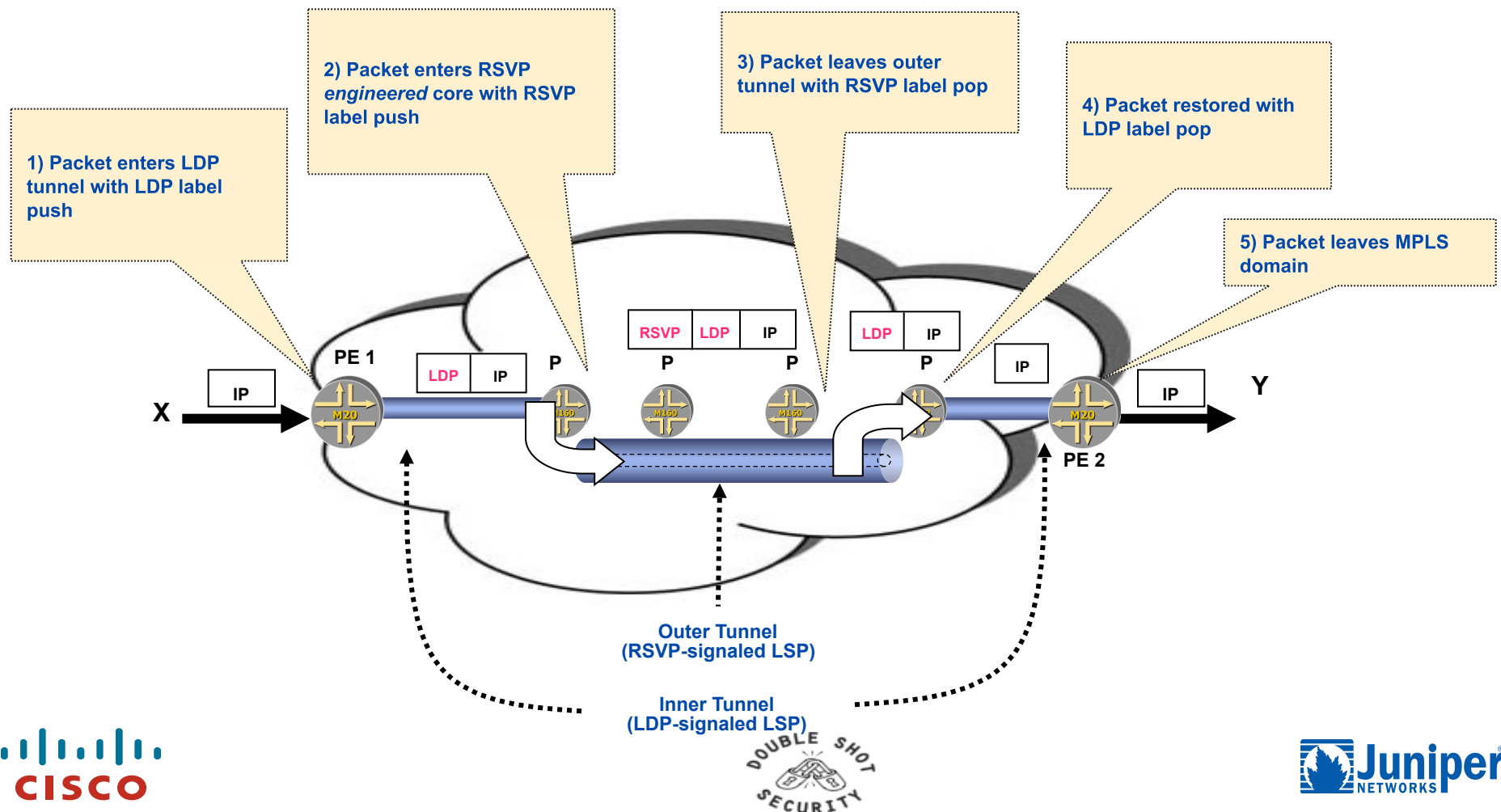
# LDP Signaling Example: Forwarding 2

LSP Merging occurs



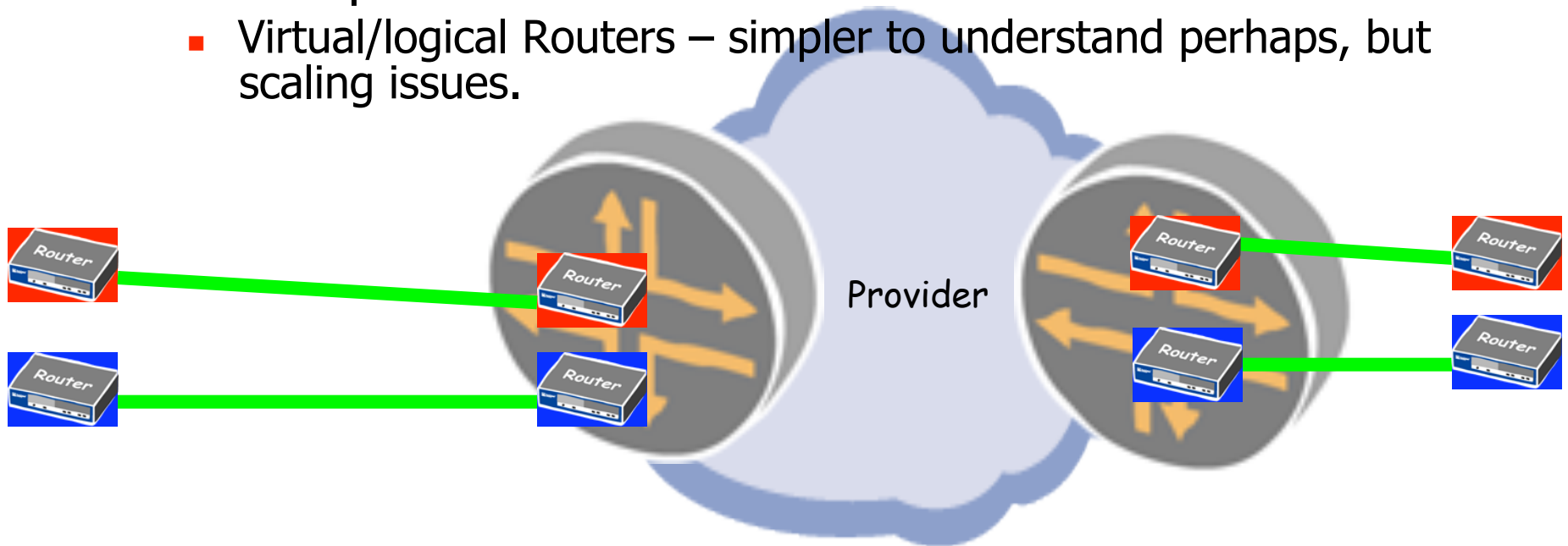
# Label Stacking

- Label stacking improves scalability
  - Similar to ATM's VP and VC hierarchy



# Layer 3 VPN's

- Now RFC 4364
  - RFC2547bis
- BGP/MPLS IP VPN's
- Other options
  - Virtual/logical Routers – simpler to understand perhaps, but scaling issues.





# Influencing Deployment

- Cost ~2 x IP connectivity
- Expected to be 1:1 in 2-3 years

YOU, are expected  
to deliver  
more for less ☺

**Predicted Revenue for IP VPN Services – Asia Pacific**

Year	Revenues	Growth
2003	\$1.69b	24.9%
2004	\$2.11b	25.4%
2005	\$2.72b	28.7%
2006	\$3.36b	23.4%
2007	\$4.06b	20.9%
2008	\$4.62b	13.7%
2009	\$5.14b	11.5%



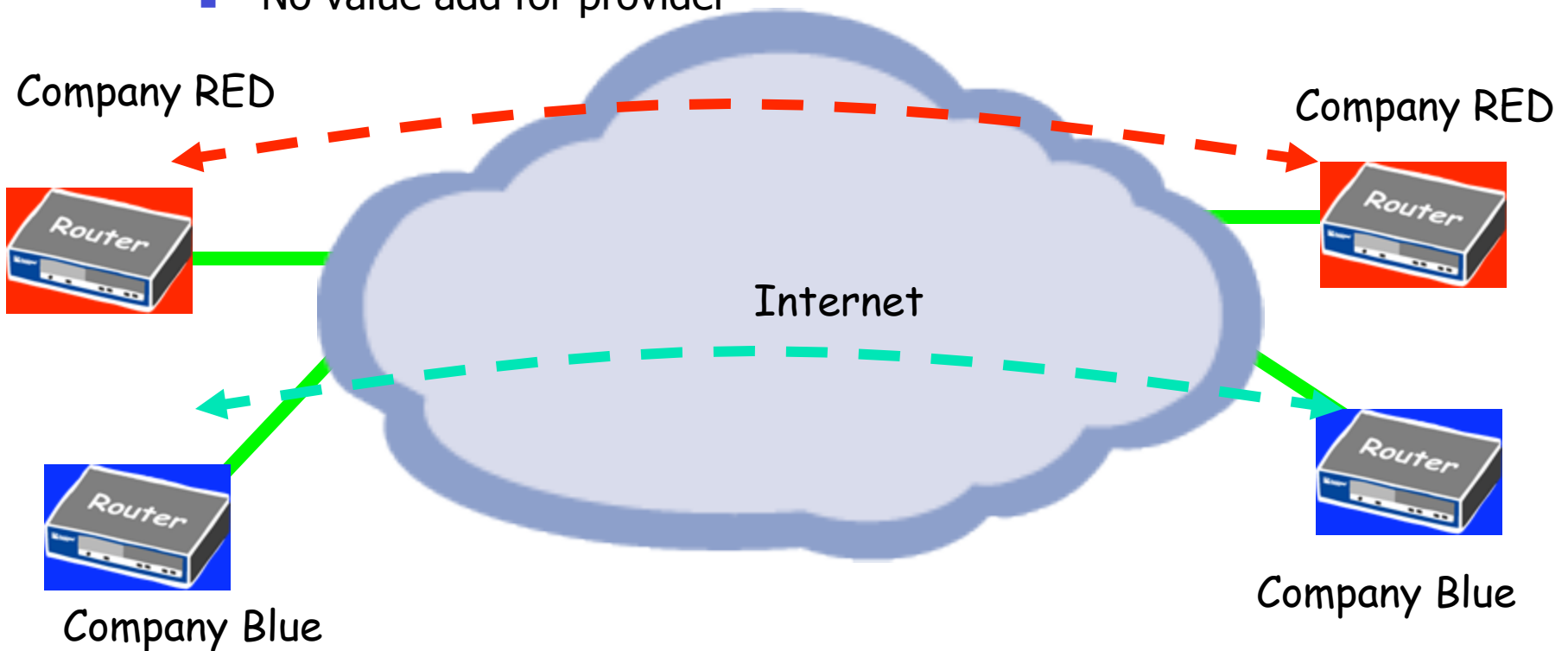
# Layer 3 VPN's (2547bis BGP/MPLS VPN's)

## Provider provisioned VPN

- ISP runs backbone for customer
  - Customer can be another ISP!
- Attractive to
  - Customer who do not want to run their own backbone
- Not attractive to
  - Customer who doesn't trust carrier
  - Customers who's jobs are threatened

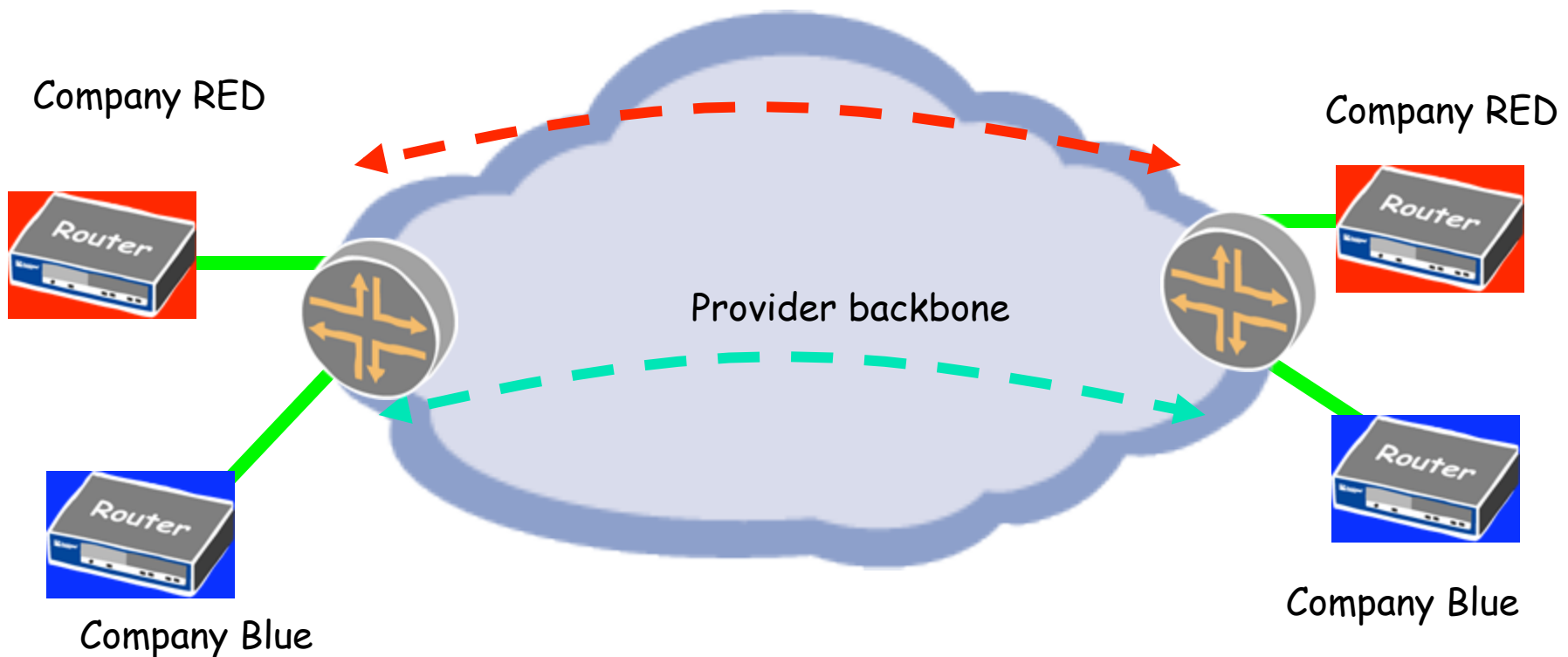
# Traditional VPN's

- CPE based
- Customer controlled
- No value add for provider



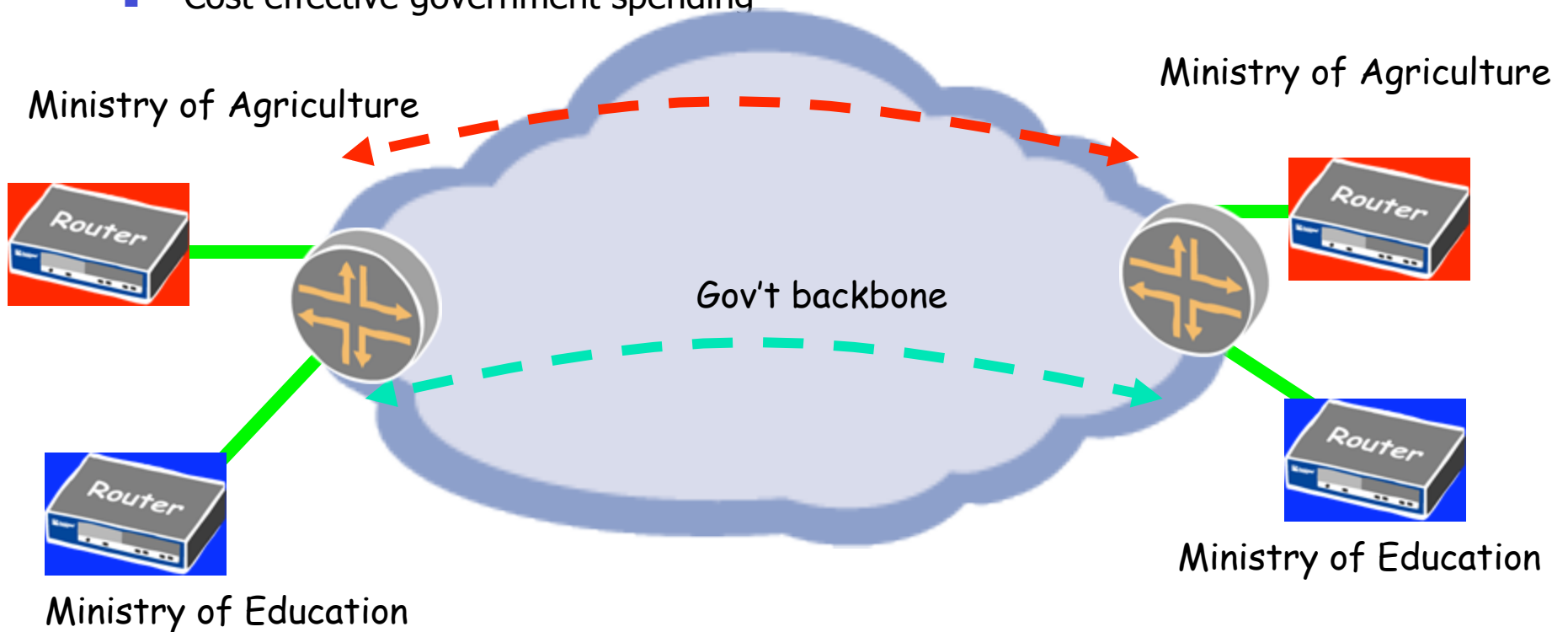
# Provider provisioned VPN's - PPVPN

- PE based
- Customer outsource backbone
- Value add for provider
- Single Site Provisioning (BGP, + Route refresh + Route Target Filtering)



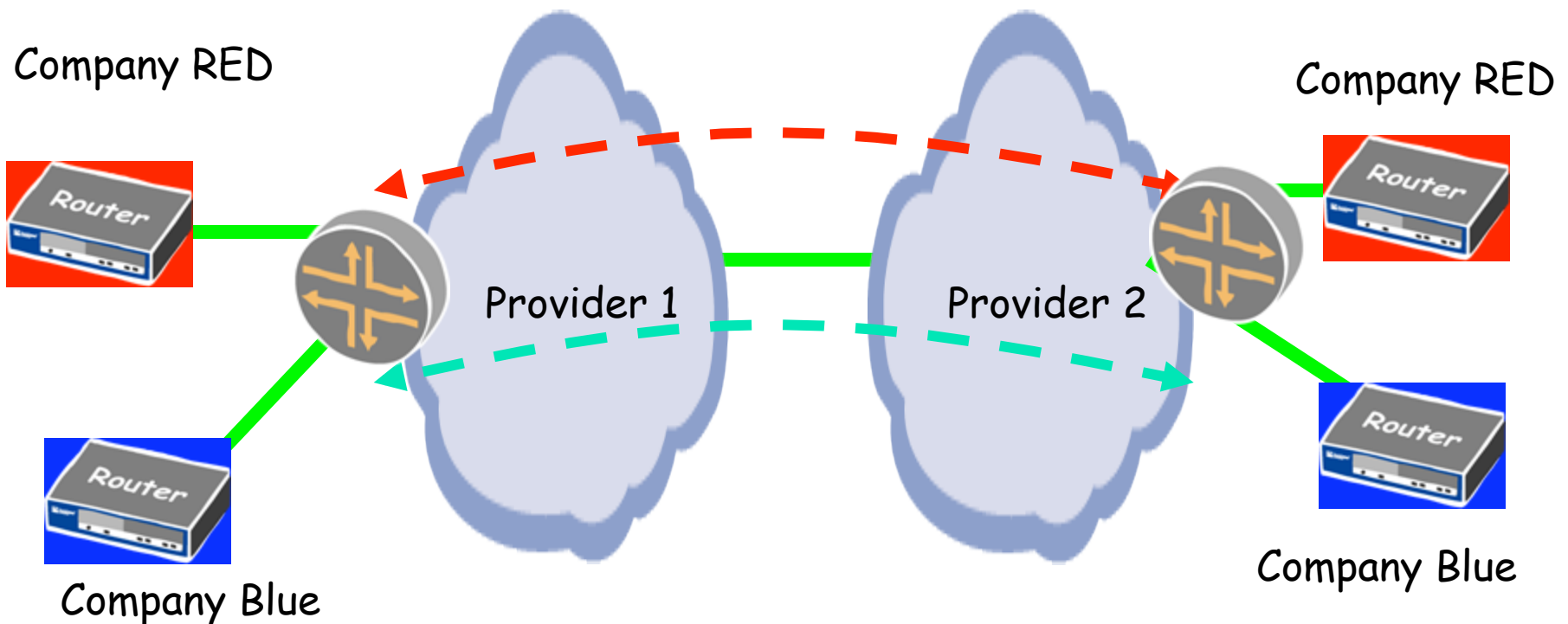
# Sharing Network backbones

- Infrastructure built by one department
  - Shared by other departments
  - Cost effective government spending
- Examples
    - Gov't backbones
    - Industry Aligned



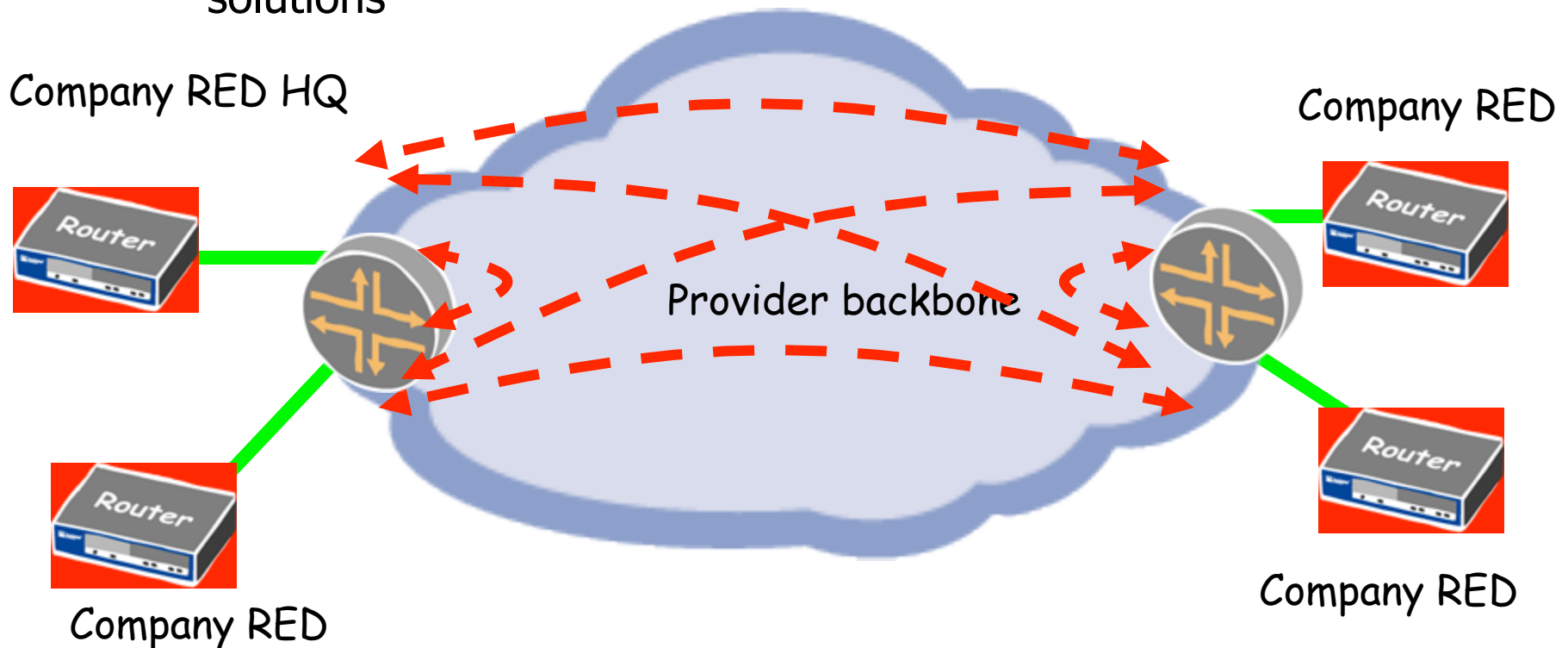
# InterAS VPN's

- Requires Co-operation
- Opportunity for global coverage



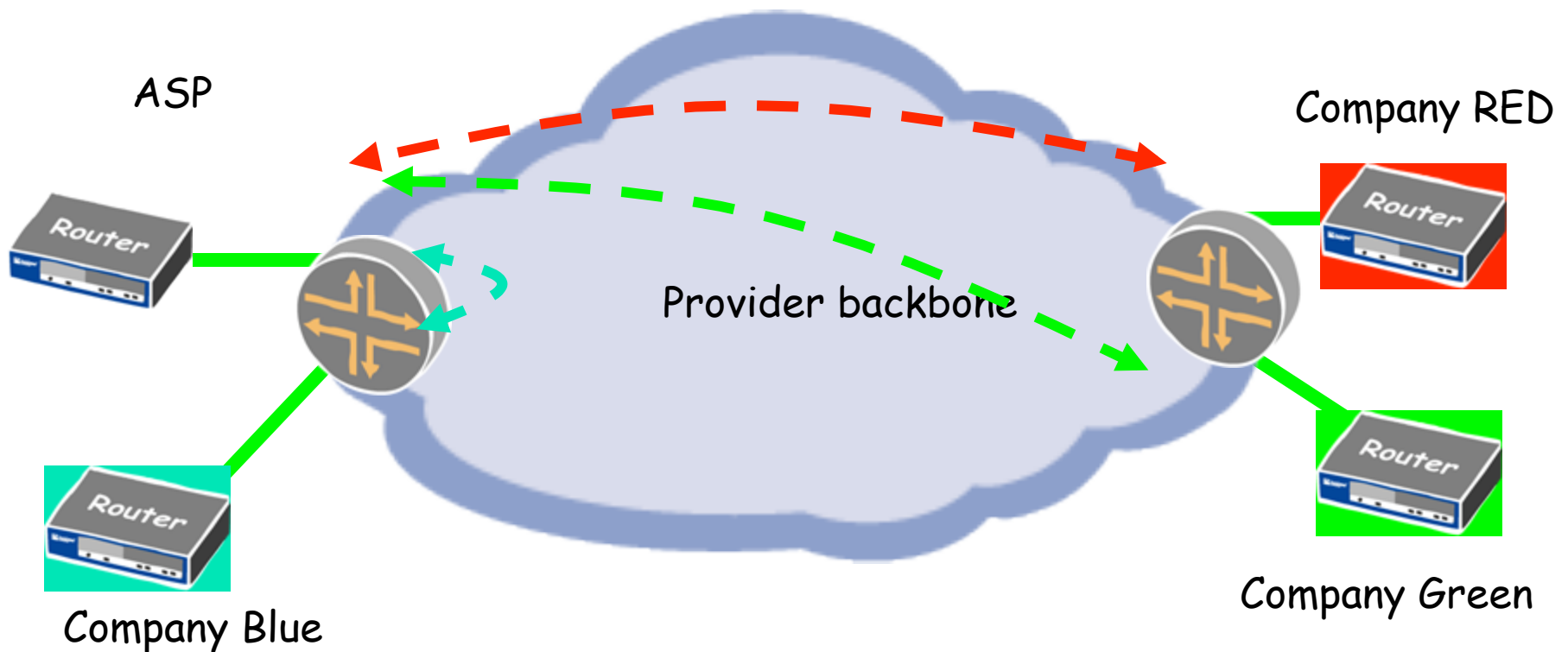
# Site Connectivity

- Partial or Full Mesh is supported
- Full Mesh is more cost effective and competitive with traditional solutions



# Overlapping VPN's

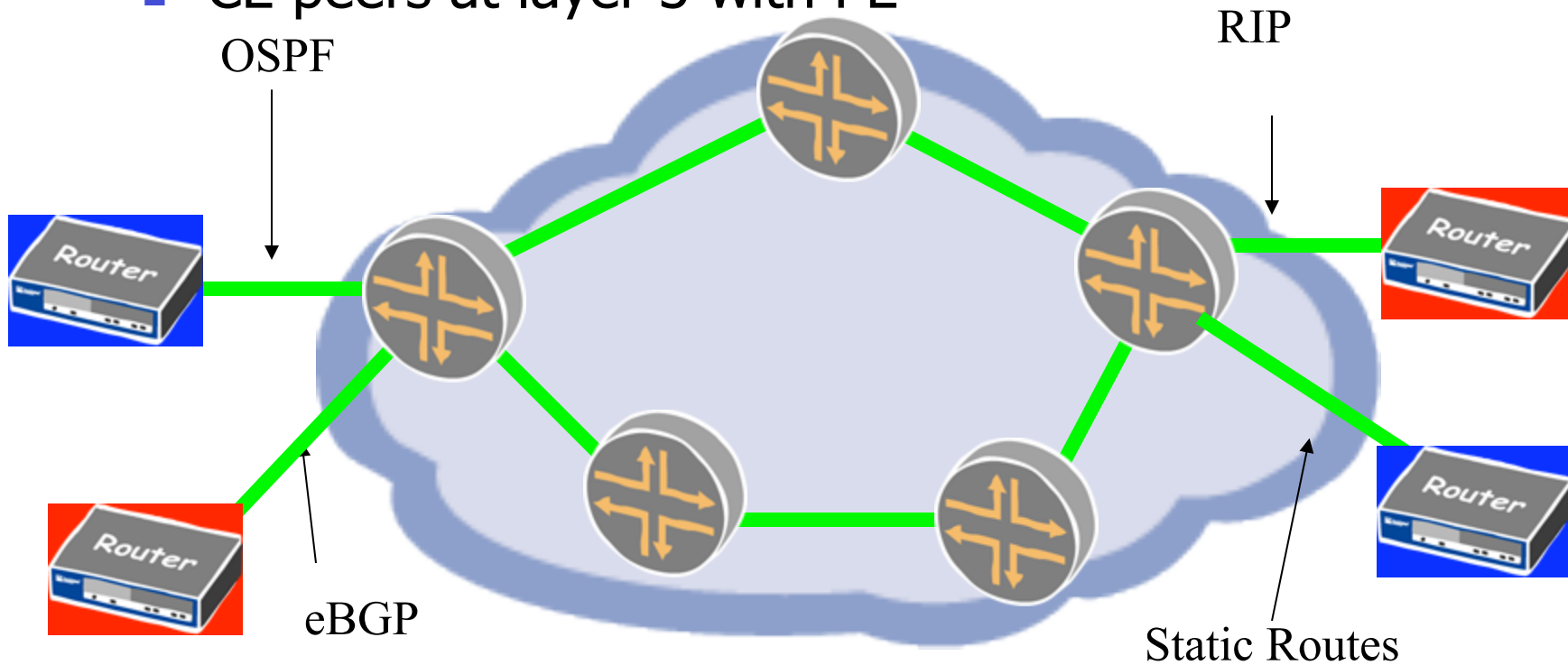
- Suites application / service providers





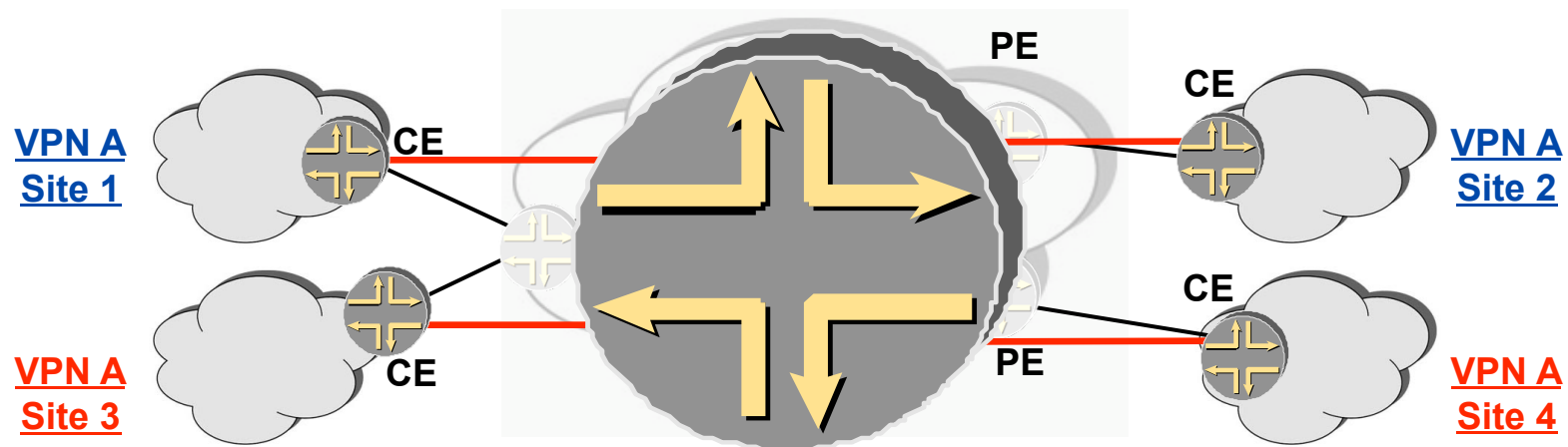
# CE-PE interaction

- Any L2 connection, Any routing protocol
- CE peers at layer 3 with PE



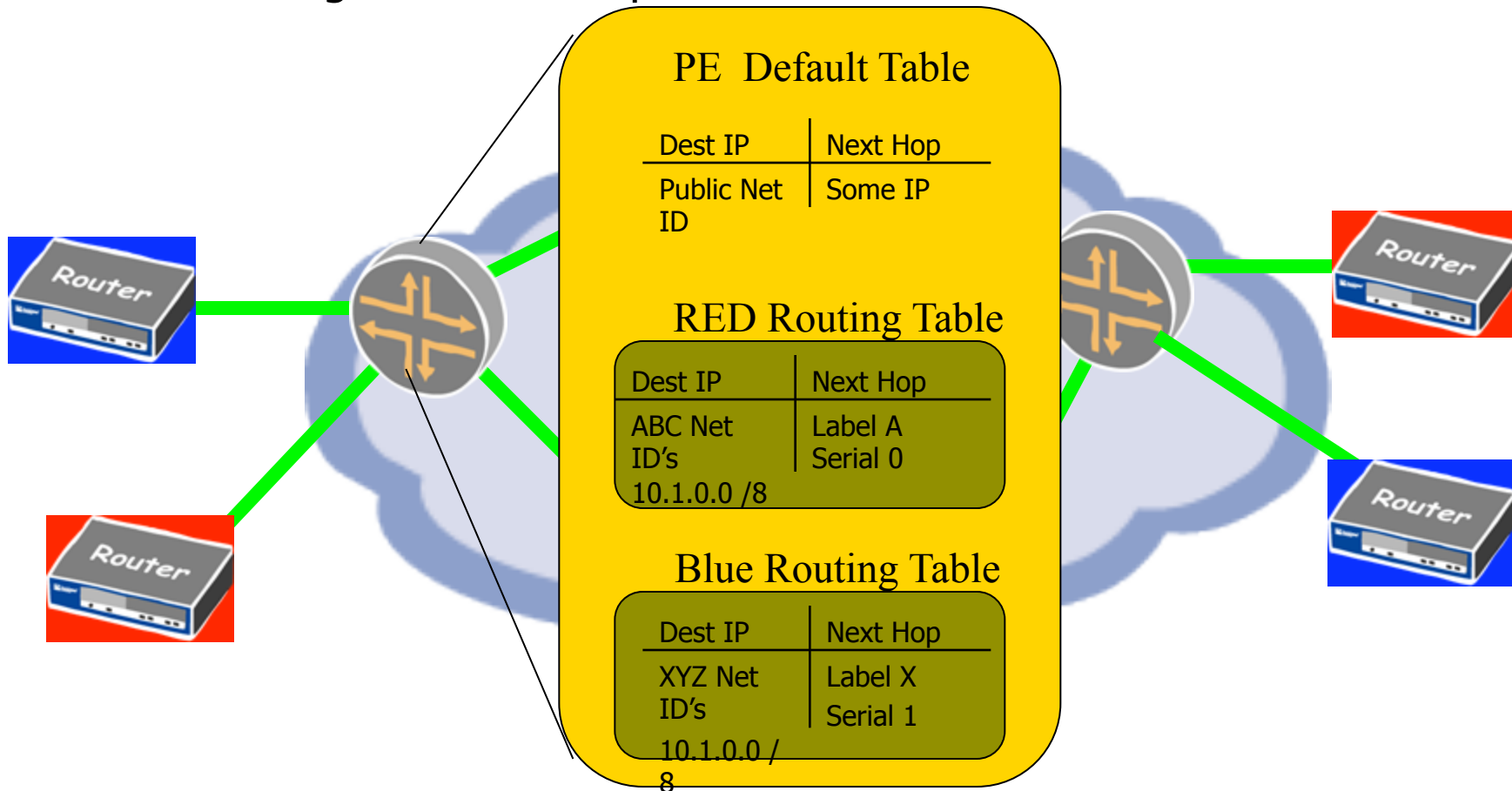
# Customer View of L3VPN

- Make the cloud look like a router
- Single site provisioning



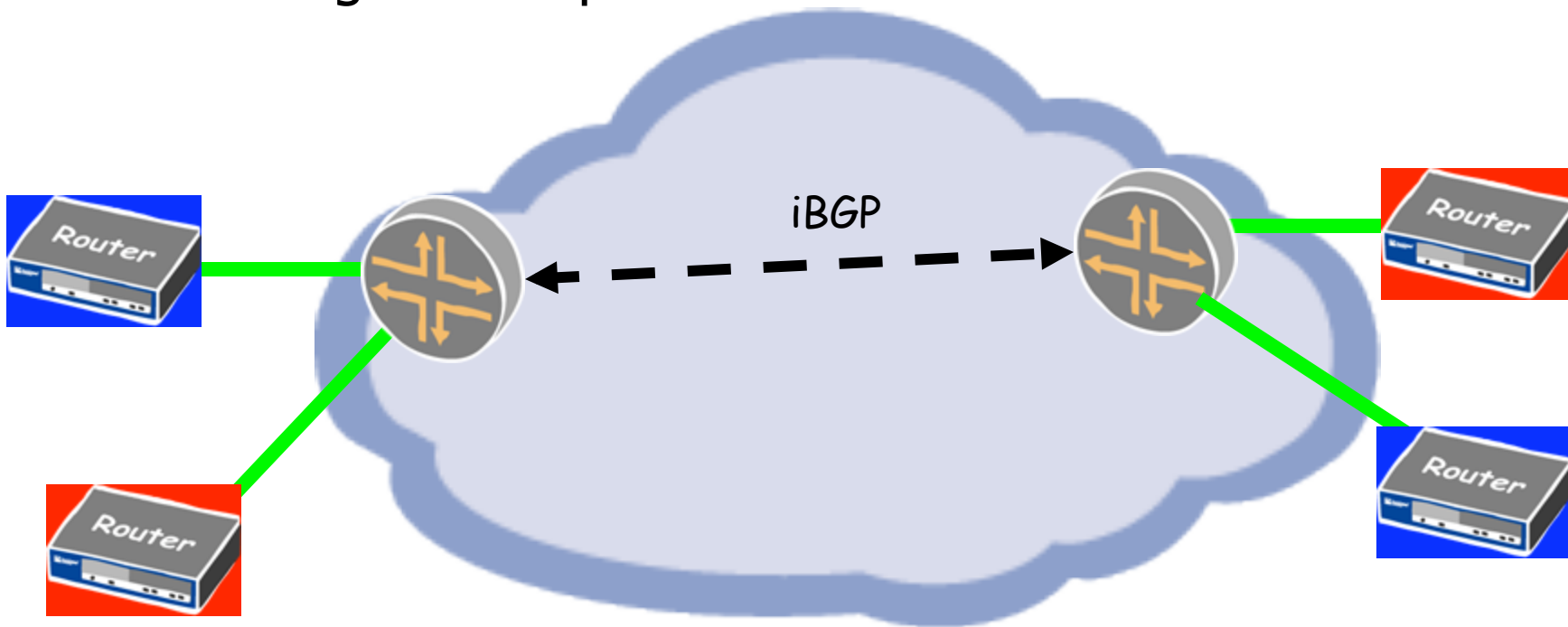
# VRF – Virtual Routing and Forwarding instance

- VRF per VPN on PE
- Logical Interface packet arrives on defines the VRF used



# PE-PE interaction

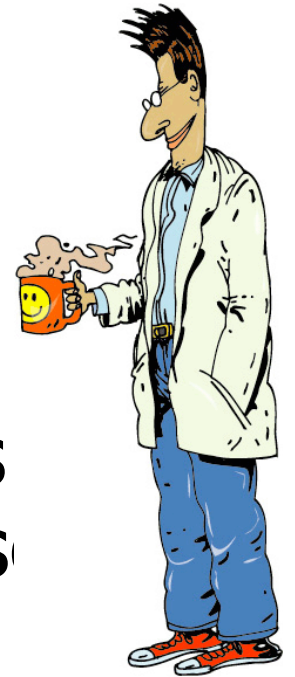
- iBGP between PE's carries routing information
- Assigns label per VPN



# Route Distinguishers

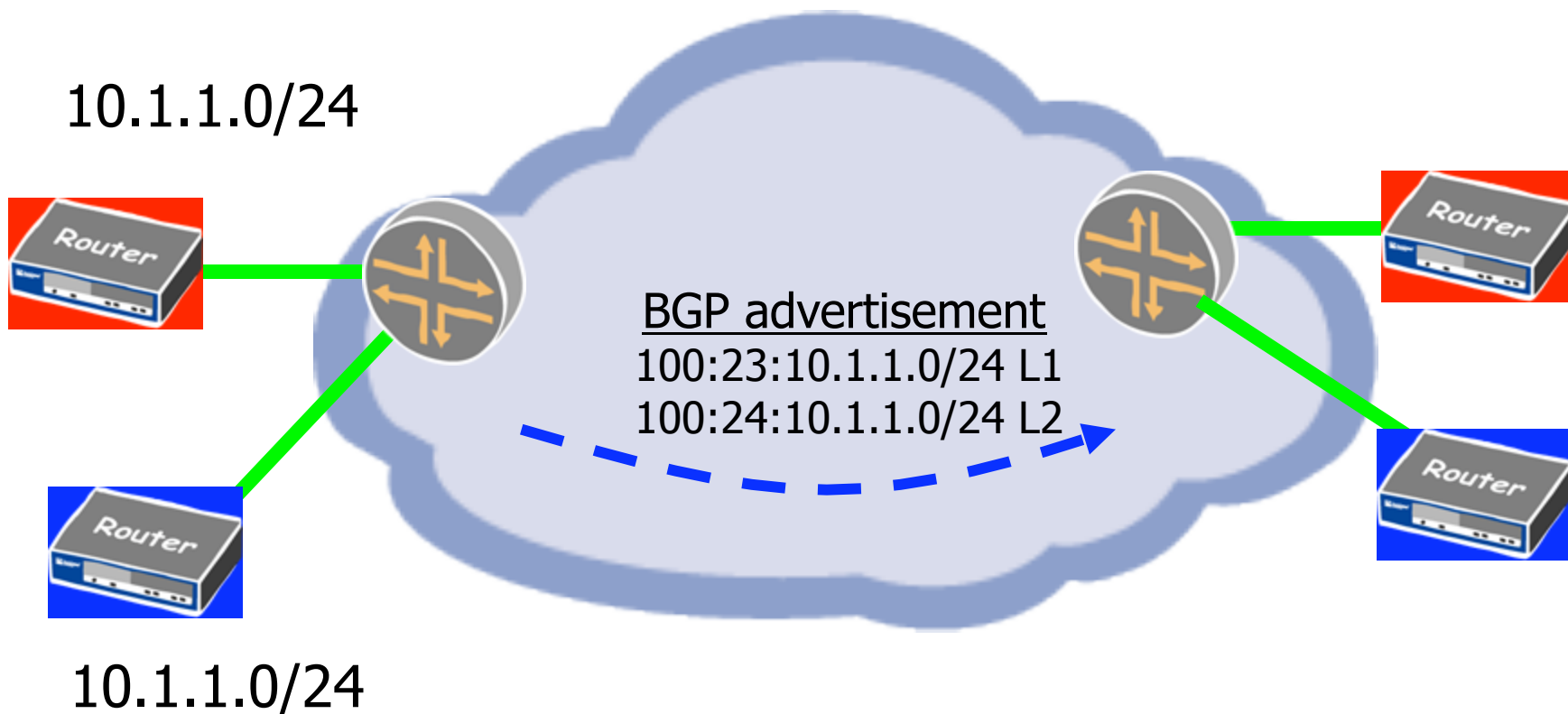
RD's have nothing to do with defining VPN membership

- Used to disambiguate possibly duplicate routes from VRF's
  - i.e. guarantee unique addressing space
  - AS:nn e.g. 100:23
  - IPv4:nn e.g. 192.168.1.1:23
- Creates a guaranteed unique address BGP can advertise in a single database
- VPNIPv4 addresses



# RD's in action

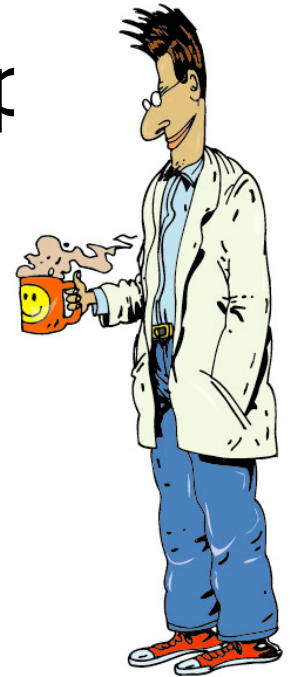
- Per VPN via BGP label assignment
- PE – PE set up via LDP or RSVP (saves state)

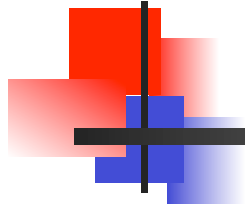


# Route Targets

RT's tell you  
which routes go into  
which VPN's

- PE receives VPN IPv4 NLRI's
- Routes then placed into VRF based up RT
  - Extended BGP community,
  - AS:nn 100:45
  - IPv4:nn e.g. 192.168.1.1:45
- A route may have one or more RT





# Route Targets in action

---

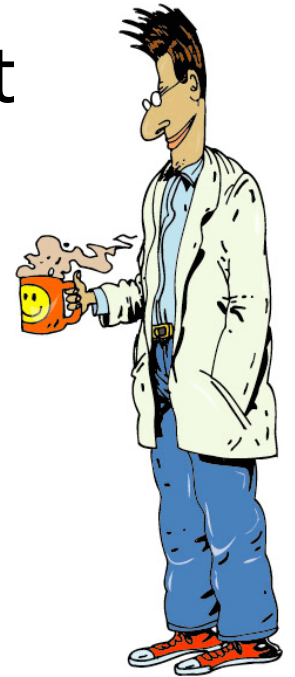
- When routes are advertised, they are exported with one or more RT's
- A VRF can import routes with matching RT's
- Security of this architecture depends on YOUR provisioning integrity



# Why RD's and RT's?

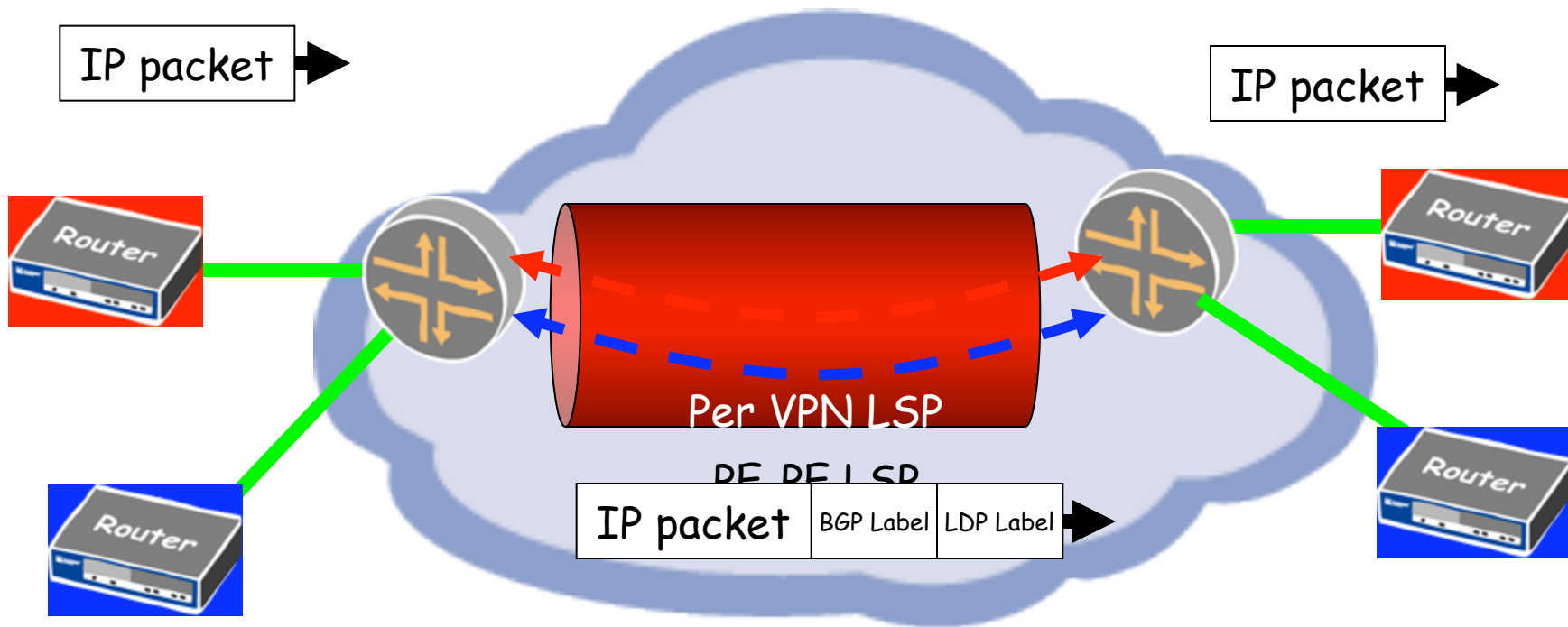
RT's tell you  
which routes go into  
which VPN's

- Overhead is better when
  - Advertisements get bigger, as opposed to
  - More advertisements
- Allows for overlapping VPN's
- Can be the same
  - But don't lock yourself in



# LSP establishment

- Per VPN via BGP label assignment
- PE – PE set up via LDP or RSVP (saves state)





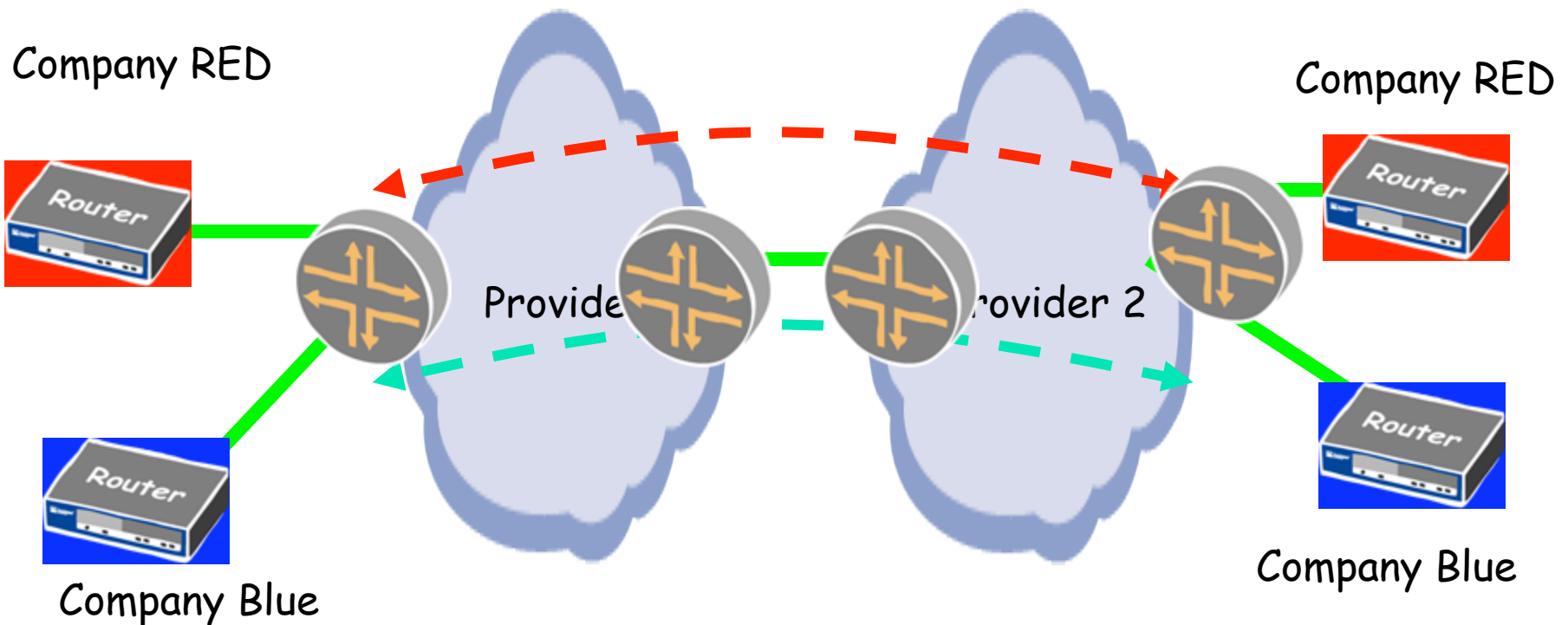
# Connectivity

---

- Hub and spoke
  - Outsourcing internet access and Applications
- Full Mesh

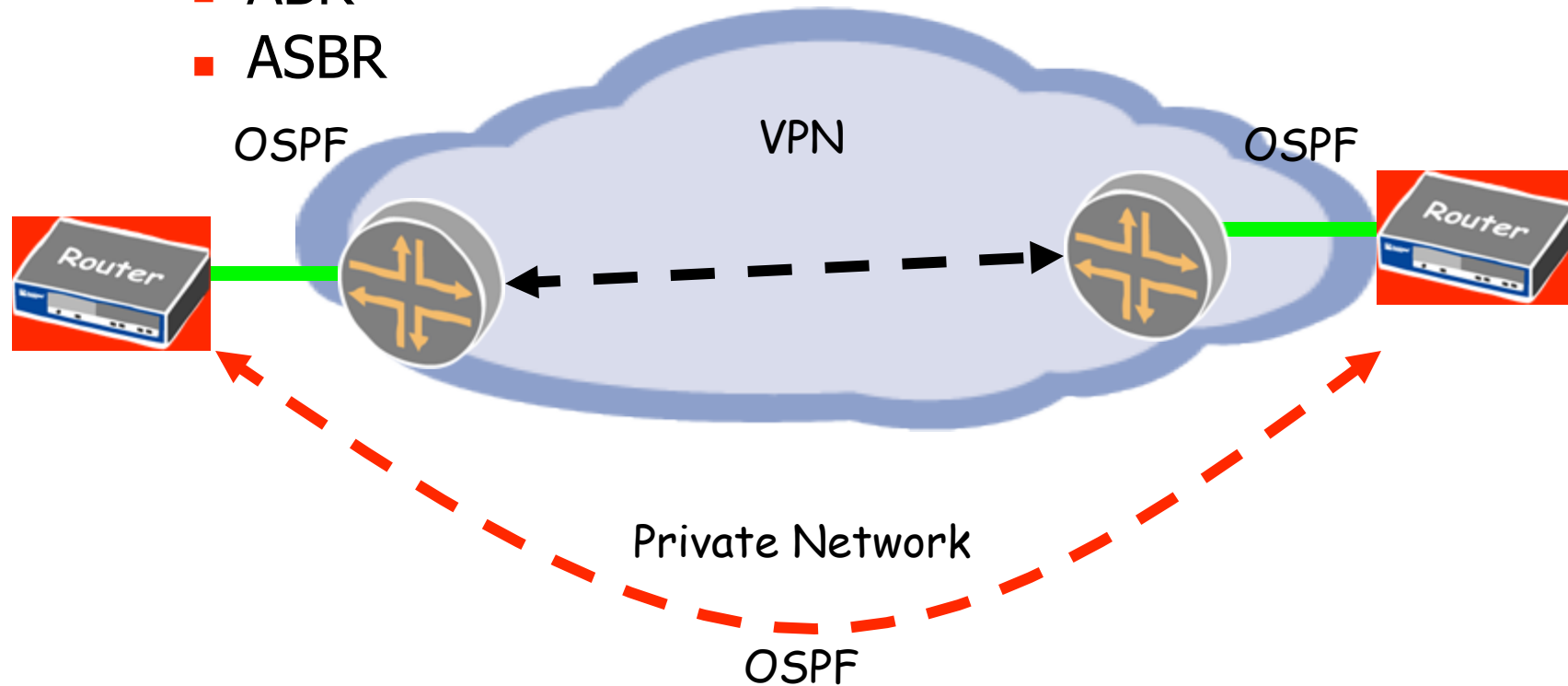
# InterAS VPN's

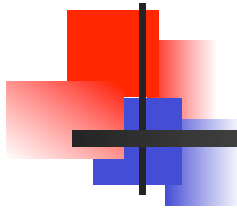
- VRF-to-VRF
- MBGP between ASBR (not OSPF)
- MBGP between PE's



# VPN as backup

- Do you want PE to appear as
  - Intra Area Router (Sham Links)
  - ABR
  - ASBR





# Issues

---

- BGP scaling
  - RR, often separate from IP RR
- Inter-AS scaling
  - MBGP between PE's is desirable
- Management
  - Usual MPLS, OAM, root cause automation.
  - Overlap NOC with VPN? Addressing?
- QoS
  - Carriers mapping 4+ queues



# Configuring L3VPN's

---



# Enable MPLS and LDP

---

JUNOS

-----

```
protocols {
    mpls {
        interface all;
    }
}
protocols {
    ldp {
        interface all;
    }
}
interfaces {
    fe-0/0/0 {
        unit 0 {
            family mpls;
        }
    }
}
```

IOS

-----

```
ip cef
mpls ip
mpls label protocol ldp
!
interface fast 0/0
mpls ip
mpls label protocol ldp

!
```







# PE-PE MP-IBGP Peering

- PE-to-PE MP-IBGP sessions require VPN-IPv4 NLRI

JUNOS

-----

```
group int {  
    type internal;  
    local-address 192.168.24.1;  
    family inet {  
        unicast;  
    }  
    family inet-vpn {  
        unicast;  
    }  
    neighbor 192.168.16.1;  
}
```

IOS

-----

```
router bgp 150  
neighbor 192.168.16.1 activate  
!  
address-family vpnv4  
neighbor 192.168.16.1 activate  
neighbor 192.168.16.1 send-community  
extended
```



# MP-IBGP Peering: PE-PE

```
lab@Amsterdam> show bgp neighbor
Peer: 192.168.16.1+179 AS 65412 Local: 192.168.24.1+1048 AS 65412
  Type: Internal      State: Established      Flags: <>
  Last State: OpenConfirm  Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference LocalAddress HoldTime AddressFamily Rib-group Refresh>
  Address families configured: inet-unicast inet-vpn-unicast
  Local Address: 192.168.24.1 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.168.16.1      Local ID: 192.168.24.1      Active Holdtime: 90
  Keepalive Interval: 30
  NLRI advertised by peer: inet-unicast inet-vpn-unicast
  NLRI for this session: inet-unicast inet-vpn-unicast
  Peer supports Refresh capability (2)
  Table inet.0 Bit: 10000
    Send state: in sync
    Active prefixes: 0
    Received prefixes: 0
    Suppressed due to damping: 0
  Table bgp.l3vpn.0 Bit: 30000
    Send state: in sync
    Active prefixes: 8
    Received prefixes: 8
    Suppressed due to damping: 0
  Table vpn-a.inet.0 Bit: 40000
    Send state: in sync
    Active prefixes: 7
    Received prefixes: 8
```



# Assigning the Route Distinguisher

- Manually assign the RD per VRF table

JUNOS

-----

```
[edit routing-instances vpn-a]
lab@HK# show
instance-type vrf;
interface fe-0/0/0.0;
route-distinguisher 192.168.16.1:1;
```

...

IOS

---

```
ip vrf ODD_Customer
rd 150:101
```

...

- Enable router to dynamically assign a unique Type 1 RD to every configured VRF table

```
[edit routing-options]
```

```
lab@HK# show
```

...

```
route-distinguisher-id 192.168.16.1;
autonomous-system 65412;
```





## A Sample VRF Table Configuration

Create a VRF table called *vpn-a* with BGP running between the PE and CE routers using the `vrf-target` statement:

JUNOS

-----

```
[edit routing-instances vpn-a]
lab@HK# show
Vrf-table-label; ←-----
instance-type vrf;
interface fe-0/0/0.0;
route-distinguisher 192.168.16.1:1;
vrf-target {
    import target:150:111;
    export target:150:111;
}
```

IOS

-----

```
ip vrf vpn-a
rd 150:101

interface Serial 0/1
ip vrf forwarding vpn-a
ip address 200.1.9.1 255.255.255.0

ip vrf vpn-a
route-target export 150:111
route-target import 150:111
```



# Further Reading

---

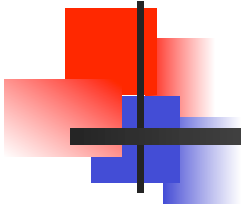
1. [http://www.juniper.net/solutions/literature/white\\_papers/](http://www.juniper.net/solutions/literature/white_papers/)
2. [http://www.juniper.net/solutions/literature/white\\_papers/200012.pdf](http://www.juniper.net/solutions/literature/white_papers/200012.pdf)
3. [www.mplsrc.com](http://www.mplsrc.com)



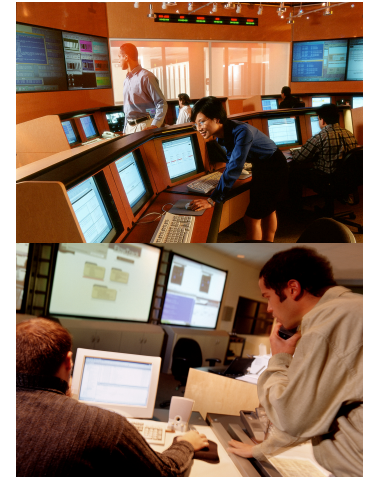
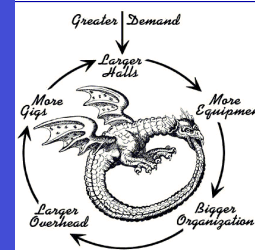
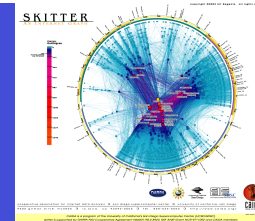
# MPLS / L3 VPN Security

---

- RFC2547bis
- BGP / IP VPN
- Other Options
  - Virtual / logical routers – simpler to understand perhaps, but scaling issues



# Understanding IPSec and SSL VPN





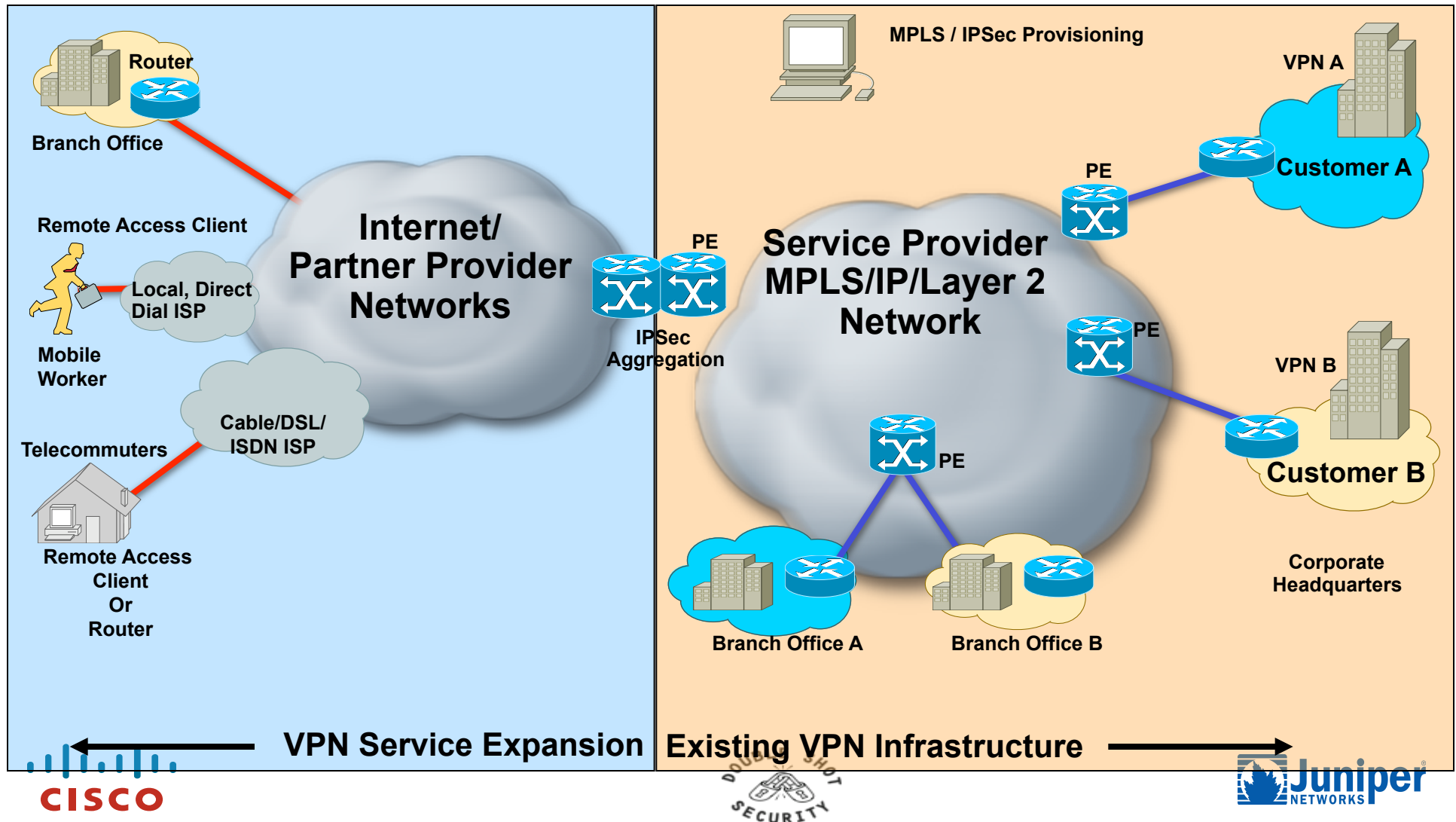
# Managed Security Services

---

- Managed IP VPN (Private)
- Managed SSL/IPSec VPN (Encrypted)



# Managed IP VPN Security Services





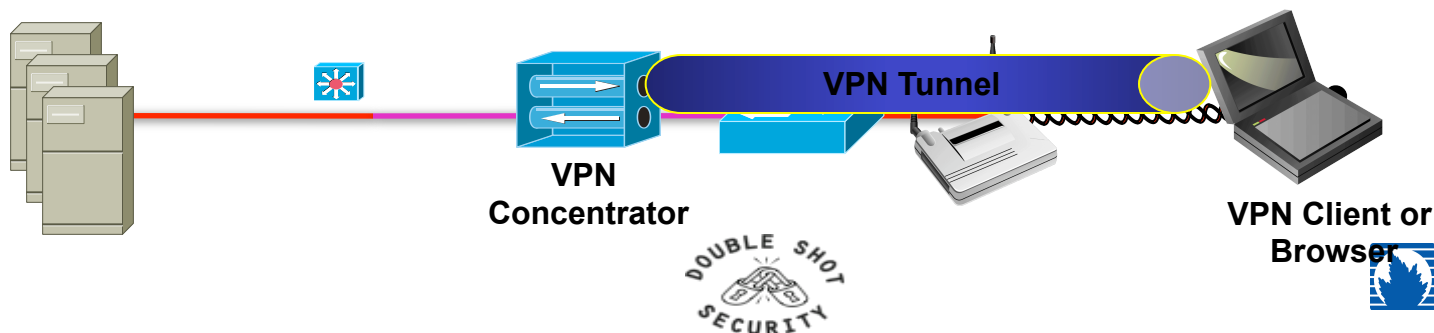
# Applying Cryptography to IP VPN

- IP Security [IPSec]
- Secure Socket Layer /Transport Layer Security
- Applying cryptography helps integrate:
  - Availability
  - Integrity
  - Confidentiality
  - Anti-replay
- That means using firewalls for access control and using SSL/TLS & IPsec for confidentiality and data origin authentication.

# Virtual Private Network (VPN) Overview

## IP Security (IPSec) and SSL

- Mechanism for secure communication over IP
  - Authenticity (Unforged/trusted party)
  - Integrity (Unaltered/tampered)
  - Confidentiality (Unread)
- Remote Access (RA) VPN Components
  - Client (mobile or fixed)
  - Termination device (high number of endpoints)





# SSL/TLS

---

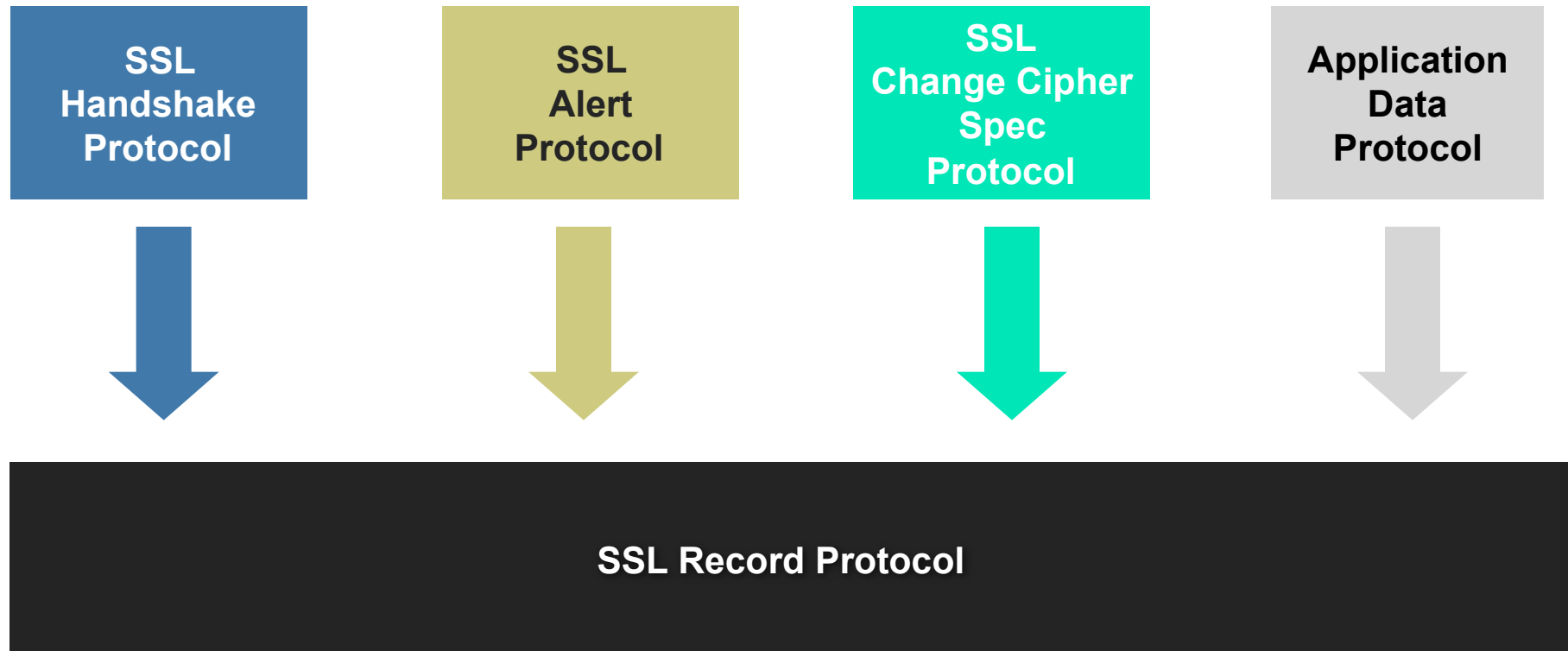
- **SSL and TLS**
  - SSL v3.0 specified in an I-D in 1996 (draft-freier-ssl-version3-02.txt)
  - TLS v1.0 specified in RFC 2246 in 1999
  - TLS v1.0 = *SSL v3.1*  $\approx$  SSL v3.0
- **OSI layer placement**
  - Above TCP/IP and below application layer
  - Most common use with HTTP  $\rightarrow$  HTTPS
- **Goals of protocol**
  - Secure communication between applications
  - Authentication + privacy + integrity



# SSL Composition

---

**SSL Is a Combination of a Primary Record Protocol  
with Four 'Client' Protocols**



# SSL Protocols

**SSL  
Handshake  
Protocol**



**Allow for Authentication and Generation of Encryption Material Through Negotiation of Parameters And Exchange of Calculated Values**

**SSL  
Alert  
Protocol**



**Used to Convey Administrative Alerts for Managing SSL Connections and Sessions**

**SSL  
Change Cipher  
Spec  
Protocol**



**Used to Signal Transition to New Cipher and Keys Generally Towards the End of a Handshake Negotiation**

**SSL  
Record  
Protocol**



**Provides for Transmission of Data in Encrypted and Compressed Form with Integrity Checking**

# How Does SSL Negotiation Work?

## SSL Session Is Negotiated Through Four Sets Of Messages

### 1st Set of Messages

Used to Start a Negotiation and to Offer and Agree upon Basic Negotiation Options



### 2nd Set of Messages

Used by the Server to Prove Its ID to the Client and to Send Its Certificate



### 3rd Set of Messages

Optionally Used By the Client to Prove Its ID and to Send Its Certificate, if Needed, and to Pass Initial Keying Material for Subsequent Key Generation to the Server



### 4th Set of Messages

Used by the Server and Client to Indicate Beginning of Use of New Keying Material

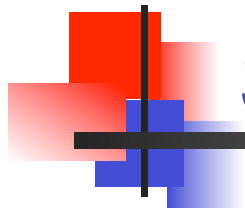


# SSL/TLS Properties

---

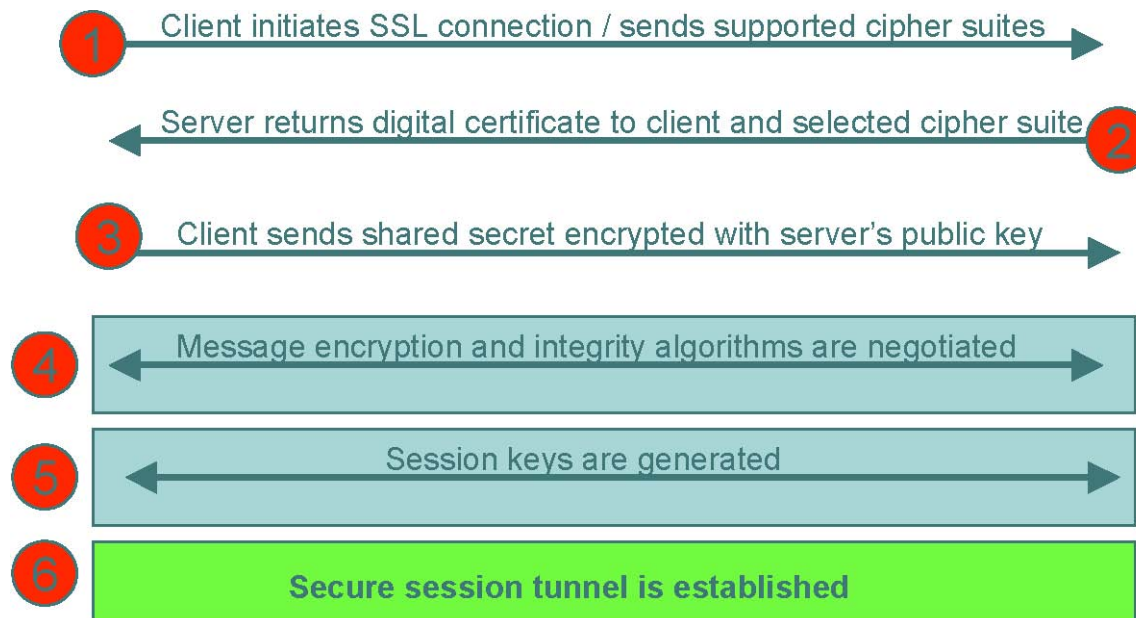
- Connection is private
  - Encryption is used after an initial handshake to define a secret key.
  - Symmetric cryptography used for data encryption ( DES or RC4).
- Peer's identity can be authenticated
  - Asymmetric cryptography is used (RSA or DSS).
- Connection is reliable
  - Message transport includes a message integrity check using a keyed MAC.
  - Secure hash functions (such as SHA and MD5) are used for MAC computations.





# SSL Handshake Process

## SSL Handshake Process





# SSL Protocol Elements

---

- Handshake Protocol
  - Negotiates crypto algorithms and keys
- Alert Protocol
  - Indicates errors or end of a session
- Record Protocol
  - Functions as layer beneath all SSL messages
  - Indicates which integrity and encryption protection is applied to data
  - Each record individually encrypted and hashed
  - Connections closed with a 'Close Notify'
  - Previously established session can be resumed by providing session ID in 'Client Hello'
    - Abbreviated version of handshake protocol
    - Reuses previously established crypto parameters



# SSL Client Authentication

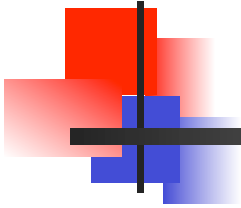
---

- Client authentication (certificate based) is optional and not often used
- Many application protocols incorporate their own client authentication mechanism such as username/password or S/Key
- These authentication mechanisms are more secure when run over SSL

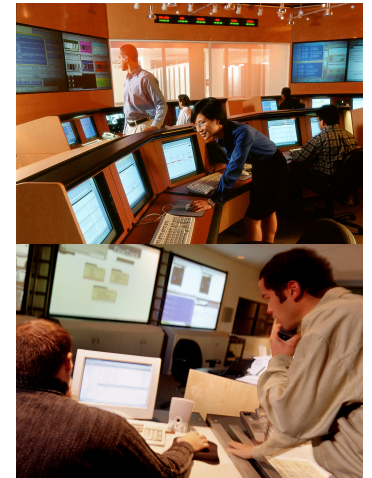
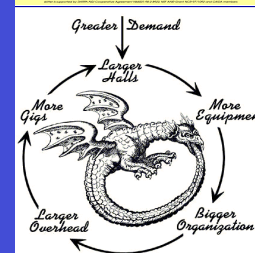
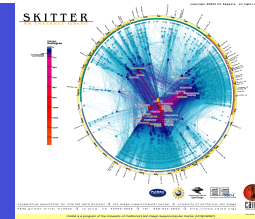


# SSL / TLS Port Numbers

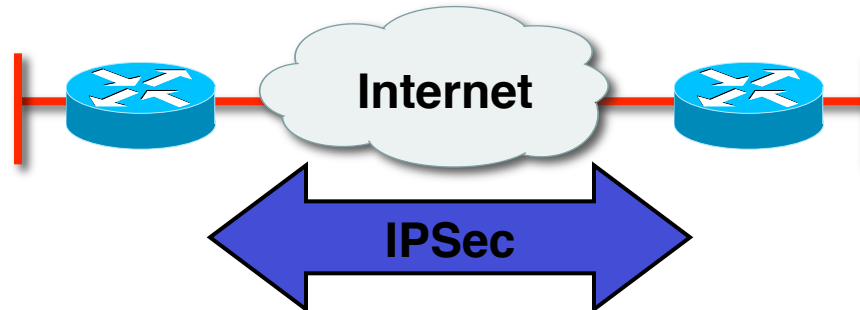
Protocol	Defined Port Number	SSL/TLS Port Number
HTTP	80	443
NNTP	119	563
SMTP	110	995
FTP-Data	20	989
FTP-Control	21	990
Telnet	23	992



# IPSec Explained for MSS



# What Is IPSec?



- IETF standard that enables encrypted communication between peers:
  - Consists of open standards for securing private communications
  - Network layer encryption ensuring data confidentiality, integrity, and authentication
  - Scales from small to very large networks
  - Available in Cisco IOS software version 11.3(T) and later
  - Included in PIX Firewall version 5.0 and later



# IPSec Composition

---

## IPSec Combines Three Main Protocols into a Cohesive Security Framework

**IKE**



**Provides Framework for the Negotiation of Security Parameters and Establishment of Authenticated Keys**

**ESP**



**Provides Framework for the Encrypting, Authenticating and Securing Data**

**AH**

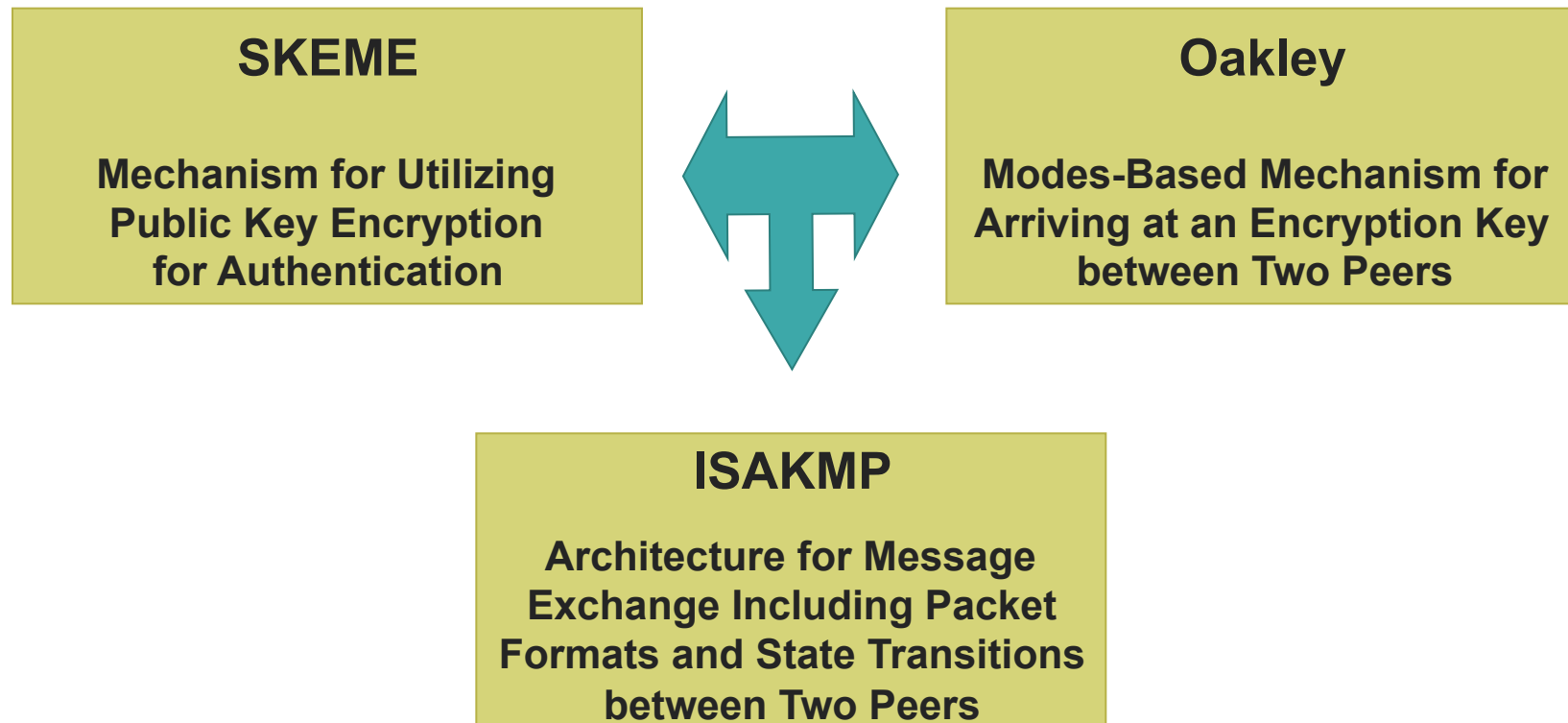


**Provides Framework for the Authenticating and Securing Data**



# What Is IKE?

IKE (Internet Key Exchange) (RFC 2409)  
Is a Hybrid Protocol







# Why IKE?

---

**IKE Solves the Problems of Manual and Unscalable Implementation of IPSec by Automating the Entire Key Exchange Process**

- Negotiation of SA characteristics
- Automatic key generation
- Automatic key refresh
- Manageable manual configuration



# How Does IKE Work?

---

## IKE Is a TWO Phase Protocol

### Phase 1 Exchange

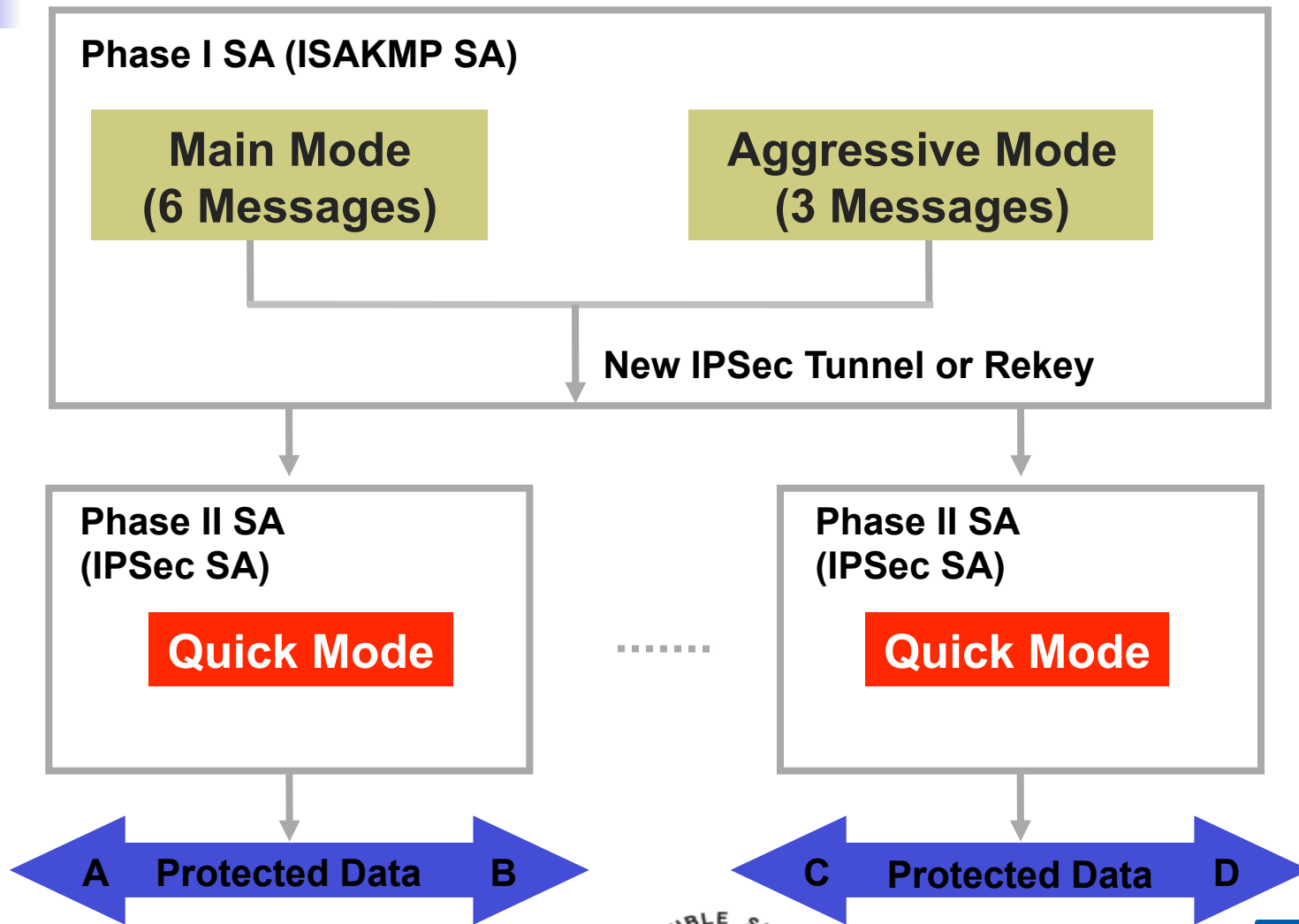
Peers Negotiate a Secure, Authenticated Channel with Which to  
Communicate 'Main Mode'  
or 'Aggressive Mode' Accomplish a Phase I Exchange

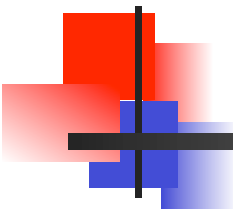


### Phase 2 Exchange

Security Associations Are Negotiated on Behalf of IPSec Services;  
'Quick Mode' Accomplishes a Phase II Exchange

# How Does IKE Work?





# IKE v2: Replacement for Current IKE Specification

---

- **Feature preservation**
  - Most of the features and characteristics of the baseline parent IKE v1 protocol are being preserved in v2
- **Compilation of features and extensions**
  - Quite a few features that were added on top of the baseline IKE protocol functionality in v1 are being reconciled into the mainline v2 framework
- **New features**
  - A few new mechanisms and features are being introduced in the IKE v2 protocol as well



# IKE v2: What Is Not Changing

---

- Features in v1 that have been debated but are ultimately being preserved in v2
  - Most payloads reused
  - Use of nonces to ensure uniqueness of keys
- v1 extensions and enhancements being merged into mainline v2 specification
  - Use of a 'configuration payload' similar to MODECFG for address assignment
  - 'X-auth' type functionality retained through EAP
  - Use of NAT Discovery and NAT Traversal techniques



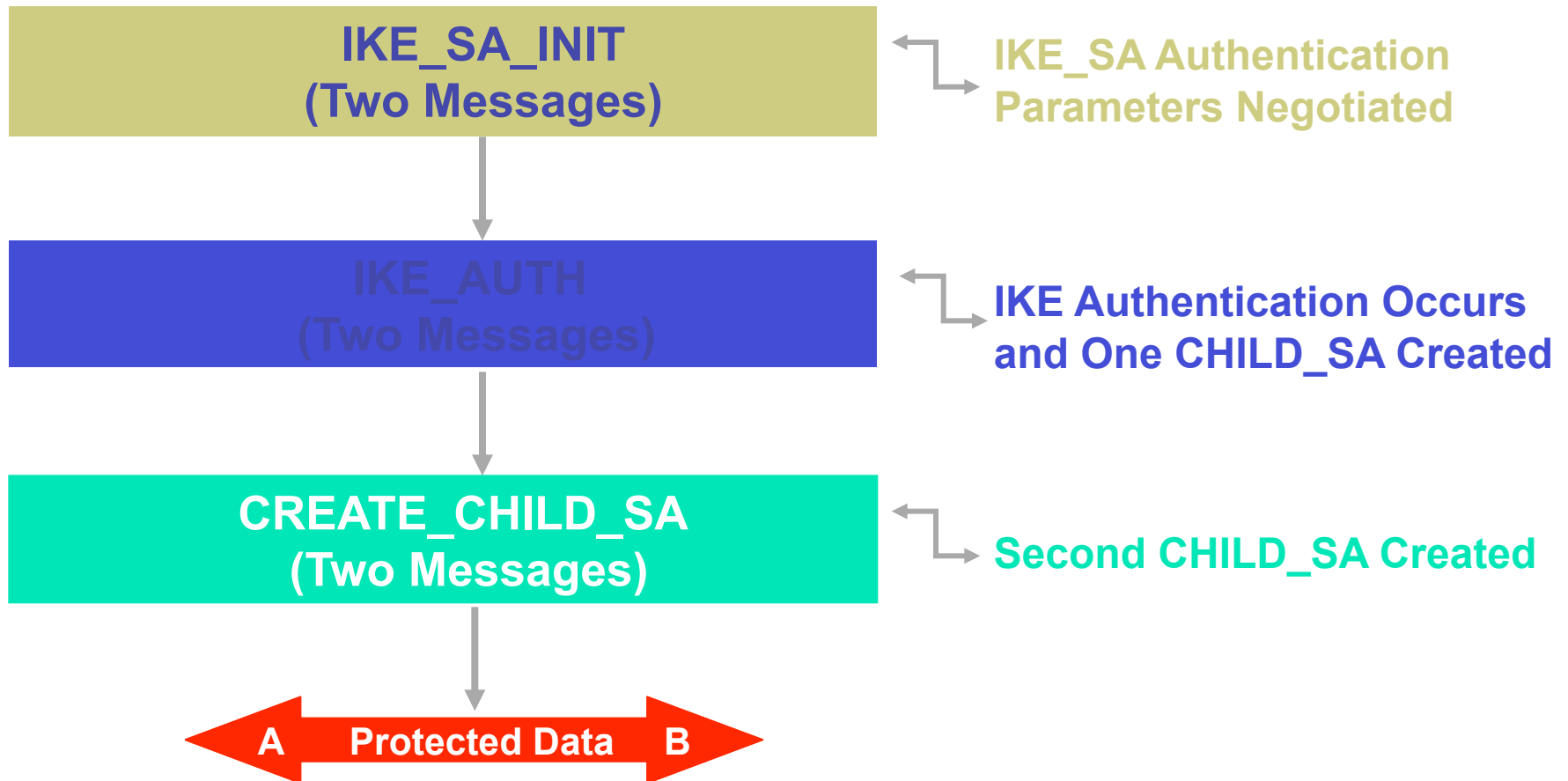
# IKE v2: What Is Changing

---

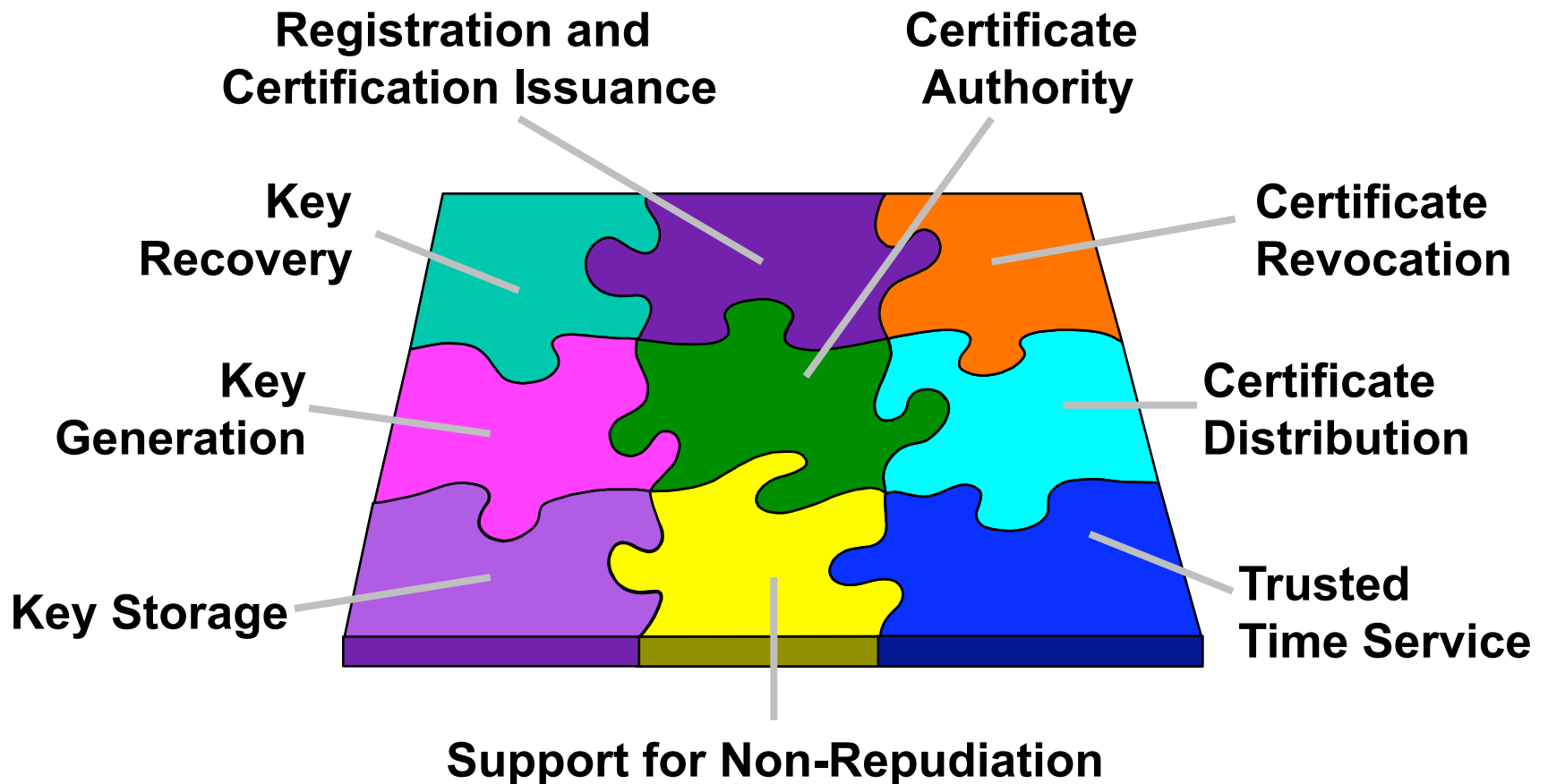
## Significant Changes Being Made to the Baseline Functionality of IKE

- EAP adopted as the method to provide legacy authentication integration with IKE
- Public signature keys and pre-shared keys, the only methods of IKE authentication
- Use of 'stateless cookie' to avoid certain types of DOS attacks on IKE
- Continuous phase of negotiation

# How Does IKE v2 Work?

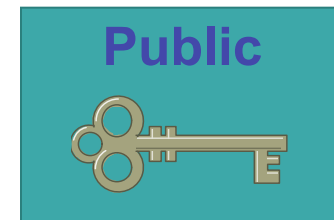


# PKI: IKE Authentication Architecture





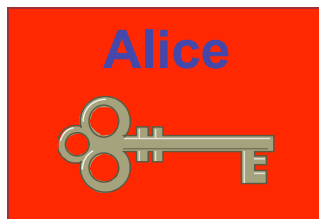
# Digital Signatures



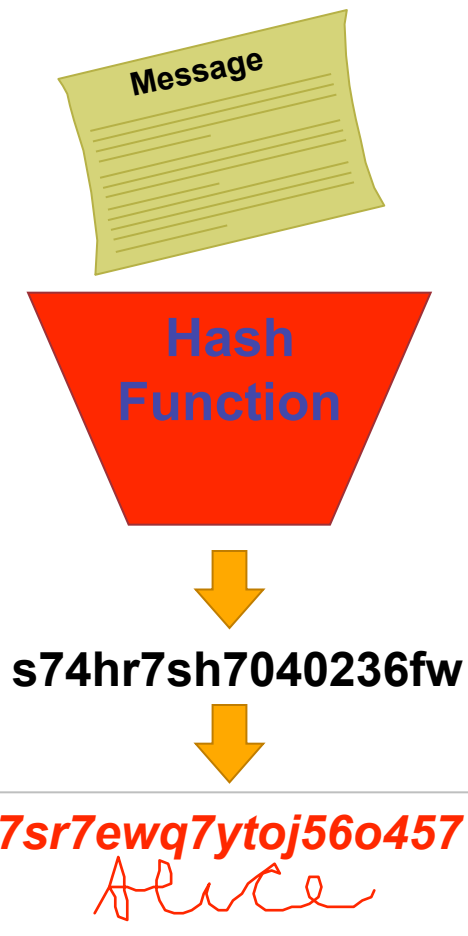
- Entity authentication
- Data origin authentication
- Integrity
- Non-repudiation

# Digital Signatures

- One-way function; easy to produce hash from message, “impossible” to produce message from hash



Hash of Message

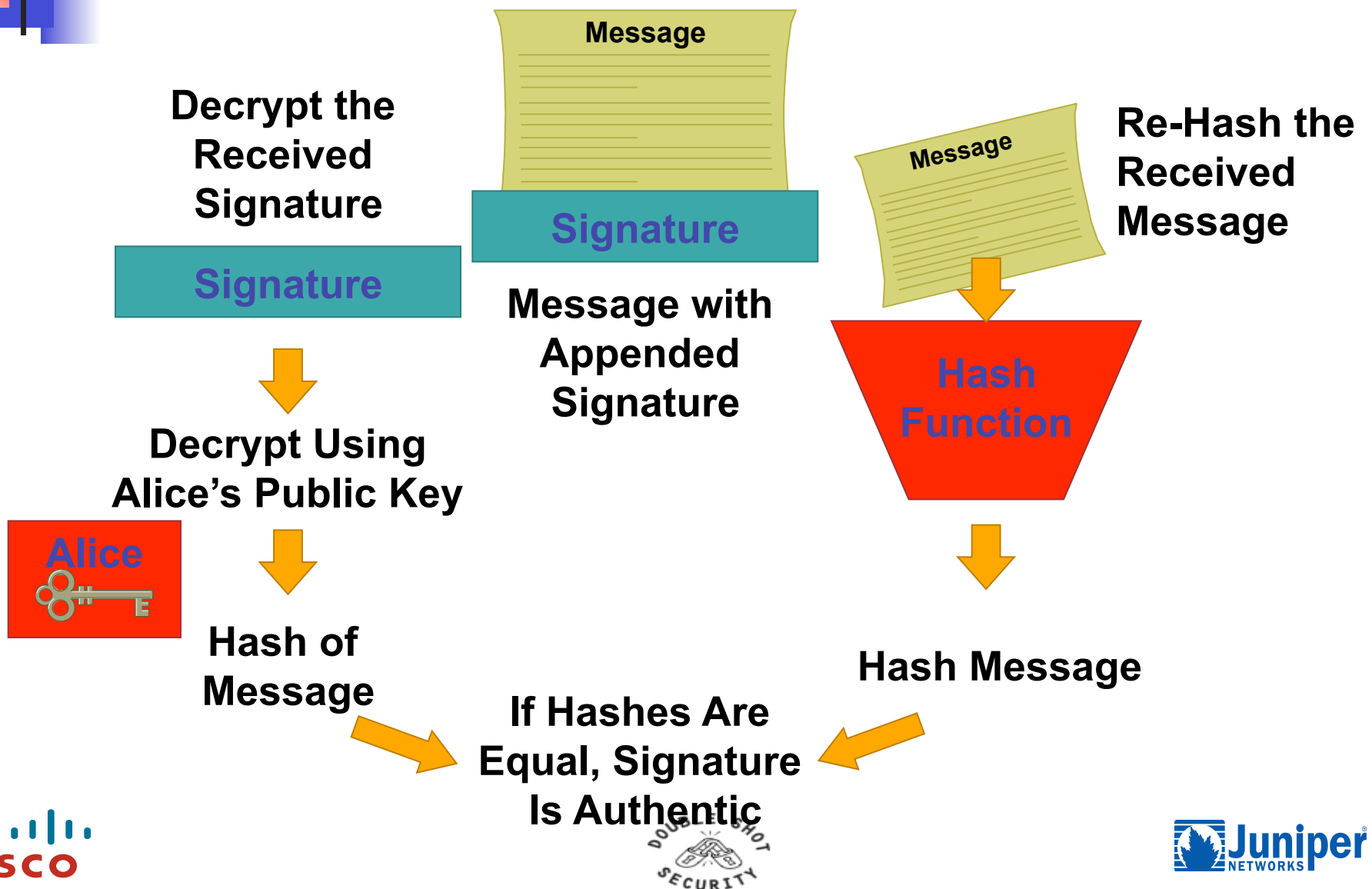


Sign Hash with Private Key

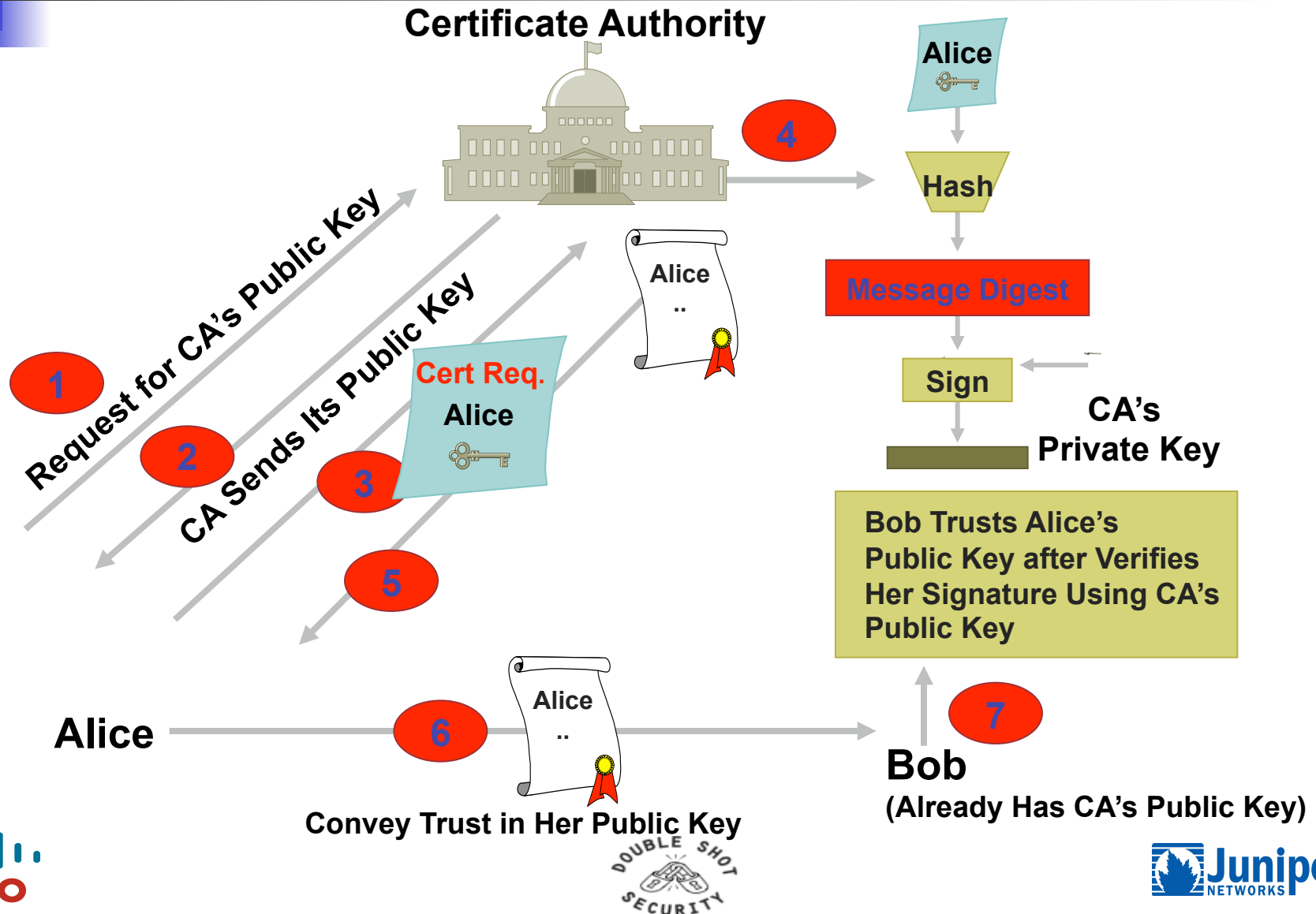
Signature = “Encrypted”

Hash of Message

# Signature Verification



# Digital Certification





# Certificate Authority

---

- The trust basis of a PKI system
- Verify user identity, issues certificates by binding user's identity to a public key with a digital certificate
- Revokes certificates and publish Certificate Revocation List (CRL)
- In-house implementation or outsourcing





# Registration Authority (RA)

---

- An RA provides interface between the user and CA
  - For example, a CGI script
- Publishes CRL



# Certificate Revocation List (CRL)

- Certificates can be revoked by CA
  - Key compromise
  - Cessation of operation
- CRL is a list of the serial numbers of revoked certificates
- Makes PKI scalable
- CRL is published by CA or RA
- CRL also has a lifetime and is updated frequently by CA

# Authentication Header

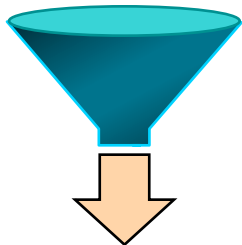


- Ensures data integrity
- Provides origin authentication—ensures packets definitely came from peer router
- Uses keyed-hash mechanism
- Does NOT provide confidentiality (no encryption)
- Provides optional replay protection



# AH Authentication and Integrity

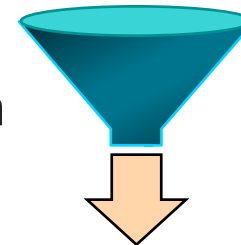
IP header + Data



Hash

Authentication  
data (00ABCDEF)

IP header + Data

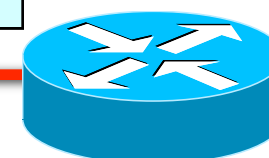


Hash

Authentication  
data (00ABCDEF)



Router A



Router B

# Encapsulating Security Payload



- Data confidentiality (encryption)
- Limited traffic flow confidentiality
- Data integrity
- Optional data origin authentication
- Anti-replay protection
- Does not protect IP header

# Tunnel and Transport Modes

- Transport mode for end-to-end session
- Tunnel mode for everything else

A

Tunnel mode

B

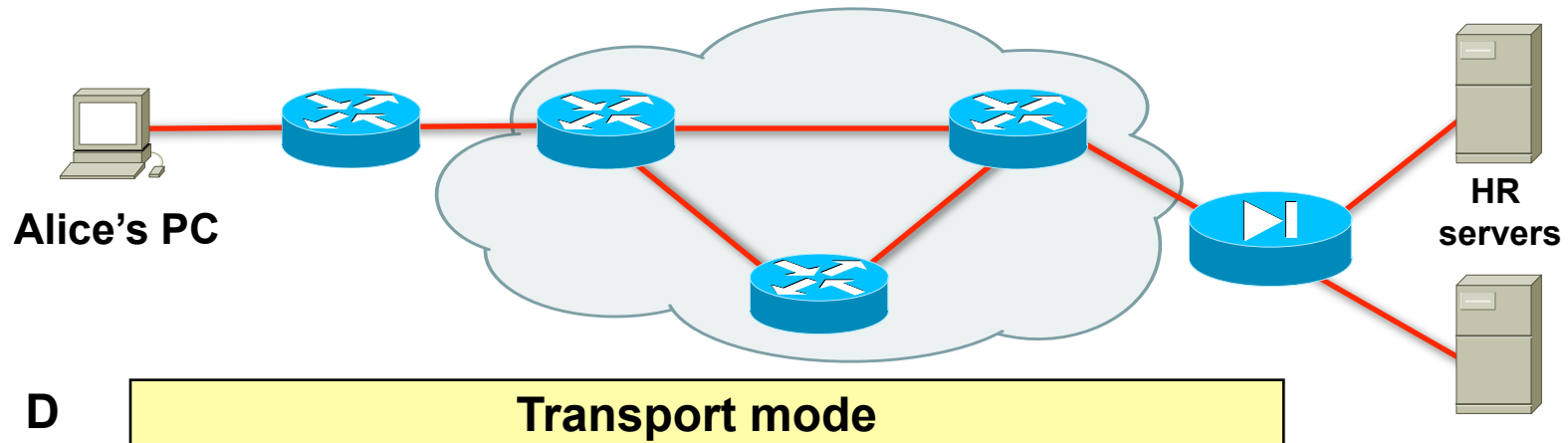
Tunnel mode

C

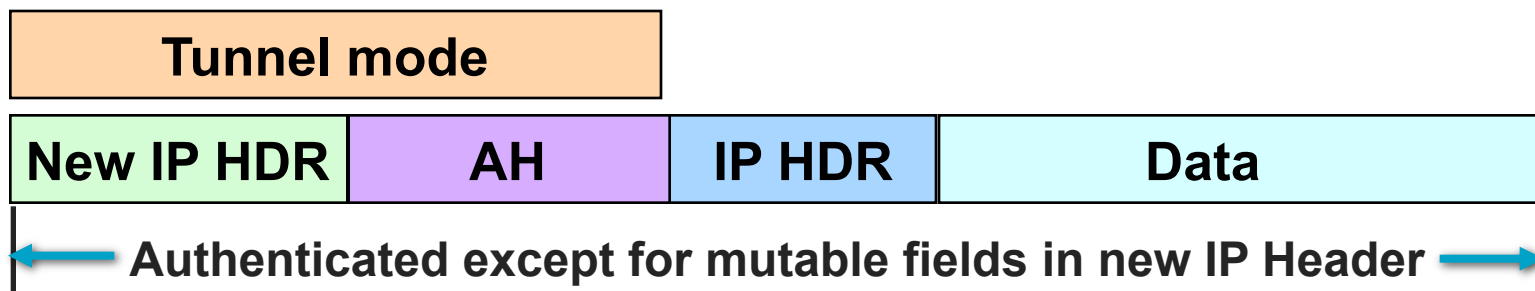
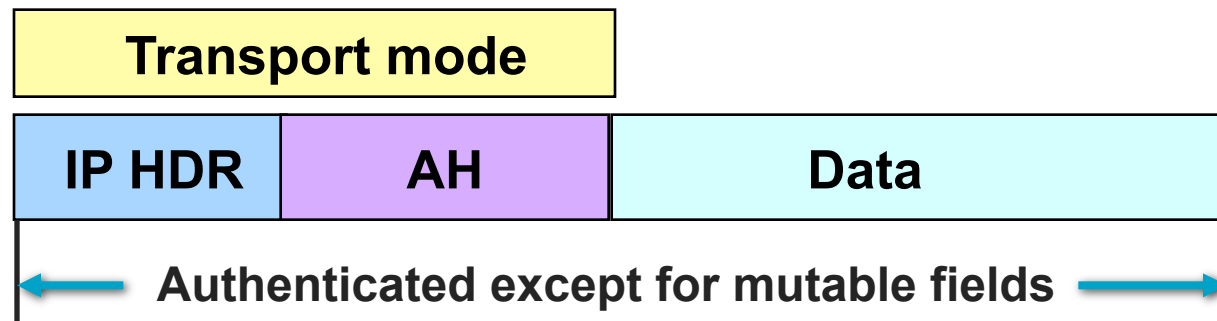
Tunnel mode

D

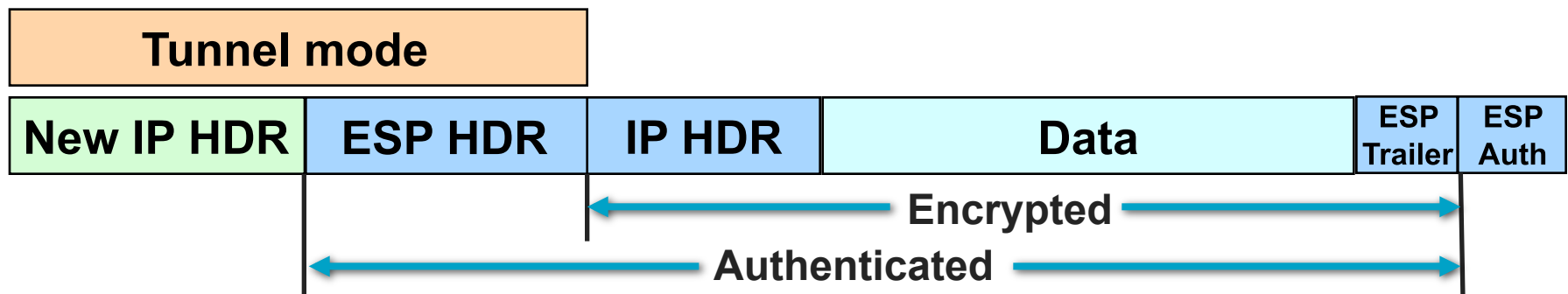
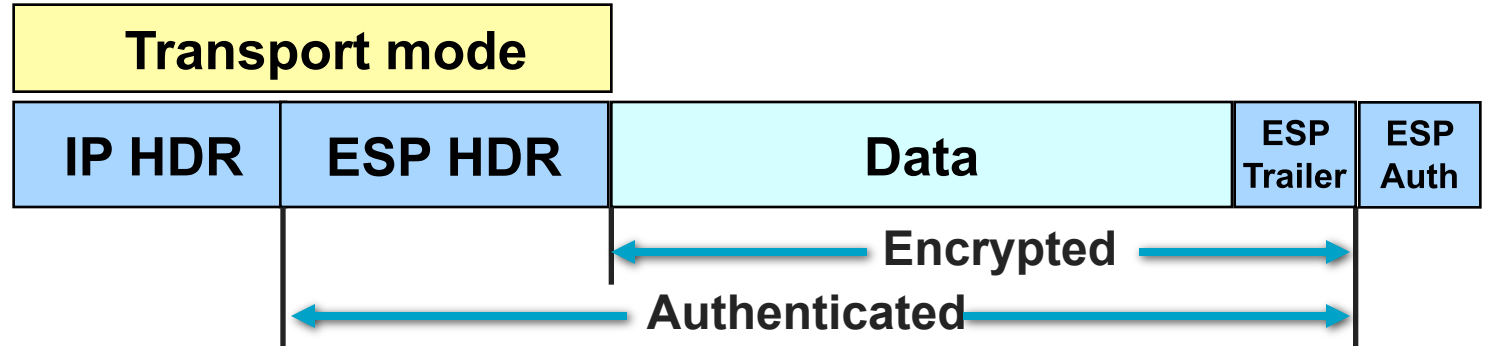
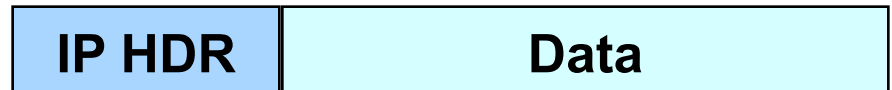
Transport mode



# AH Tunnel vs. Transport Mode

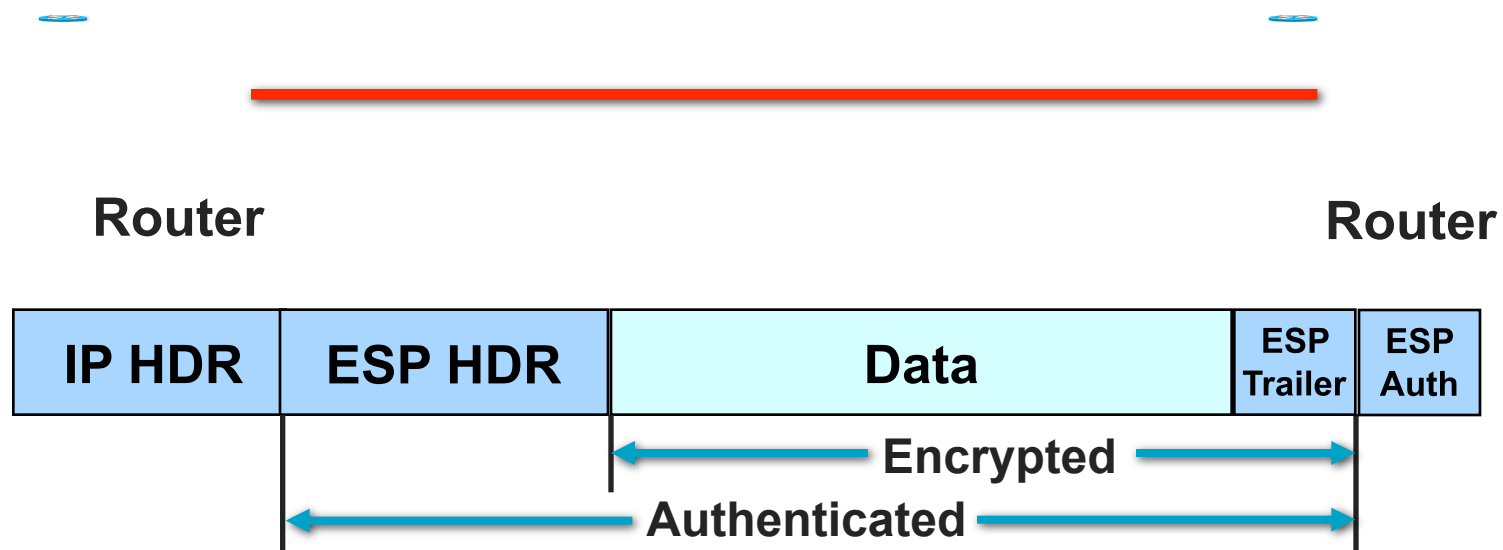


# ESP Tunnel vs. Transport Mode



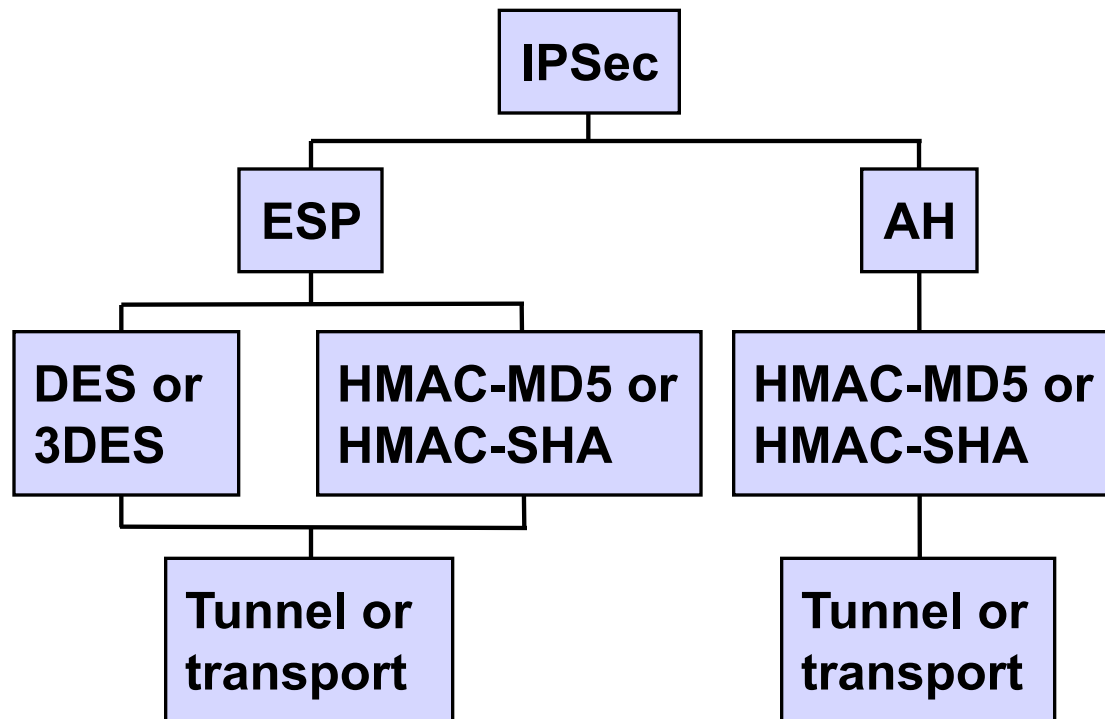
# ESP Encryption with a Keyed-HMAC

- Provides ESP confidentiality with encryption
- Provides integrity with a keyed HMAC



# IPSec Transforms

**An IPSec transform specifies either an AH or an ESP protocol and its corresponding algorithms and mode.**





# Transform Sets

---

**Transform1 + Transform2 + Transform3**  
**esp-des**  
**ah-md5-hmac**  
**esp-md5-hmac + esp-des**  
**esp-sha-hmac + esp-3des**  
**ah-sha-hmac + esp-3des + esp-sha-hmac**

- A transform set is a combination of IPSec transforms that enact a security policy for traffic
- Up to three transforms can be in a set
- Sets are limited to up to one AH and up to two ESP transforms

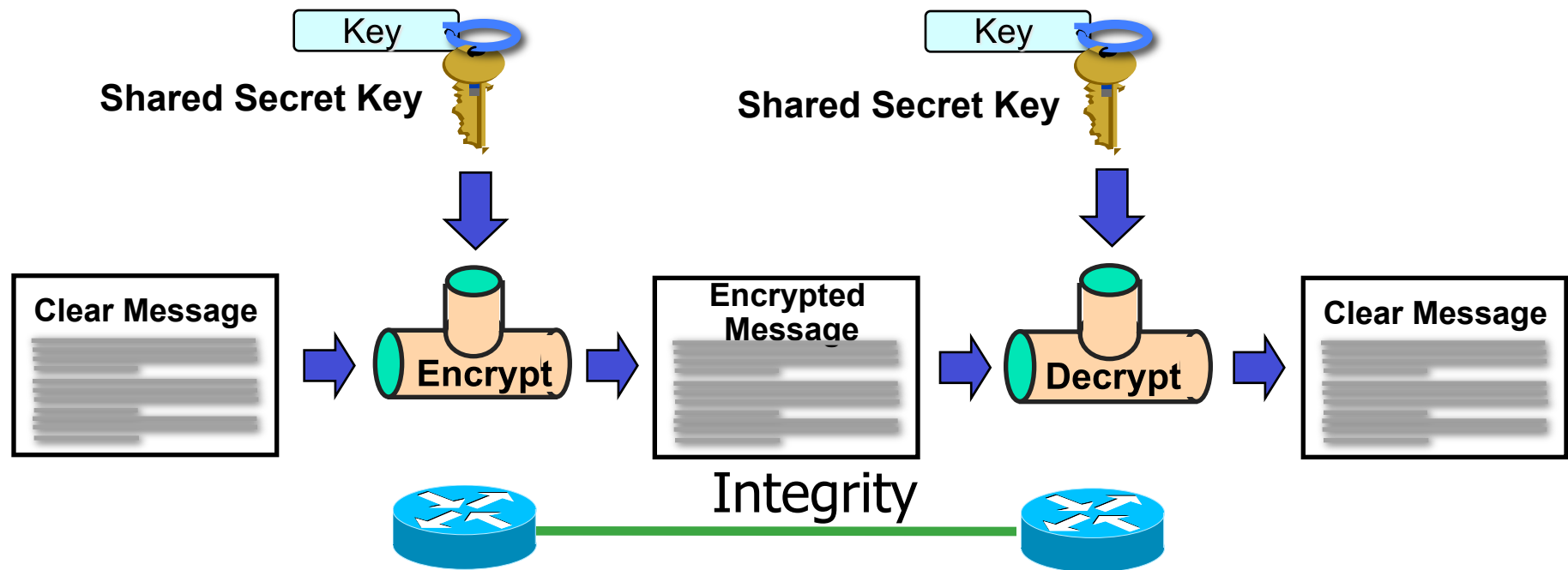




# Data Encryption Standard (DES)

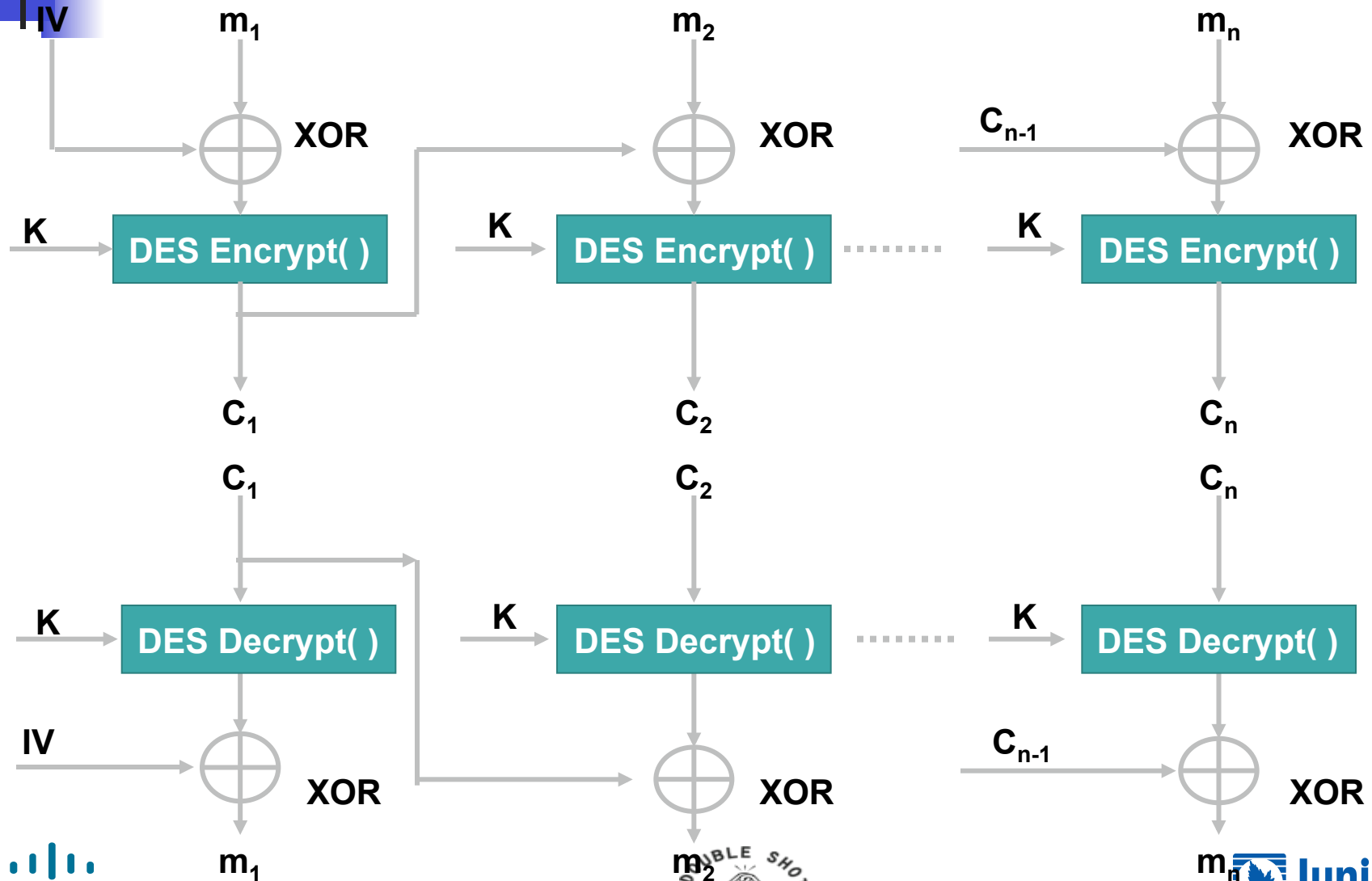
- Symmetric key encryption algorithm
- Block cipher: works on 64-bit data block, use 56-bit key (last bit of each byte used for parity)
- Mode of operation: how to apply DES to encrypt blocks of data
  - Electronic Code Book (ECB)
  - Cipher Block Chaining (CBC)
  - K-bit Cipher FeedBack (CFB)
  - K-bit Output FeedBack (OFB)

# DES Encryption



- Encryption turns cleartext into ciphertext.
- Decryption restores cleartext from ciphertext.
- Keys enable encryption and decryption.

# DES CBC Mode



# Triple-DES



- 168-bit total key length
- Mode of operation decides how to process DES three times
- Normally: encrypt, decrypt, encrypt
- More secure than DES but slower
- So is 3DES optimally the fastest, the easiest to implement and the securest algorithm out there?



# AES Encryption

---

- Published in November 2001
- Rijndael algorithm developed by Dr. Joan Daemen and Dr. Vincent Rijmen
- Symmetric Block Cipher
- 128 bit blocks
- 3 key lengths: 128, 192, and 256 bits
- Symmetric and parallel
- Low memory requirement



# AES Key Length

---

- **Key Length (in bits) Number of Combinations**

- 40 240 = 1,099,511,627,776
- 56 256 =  $7.2 \times 10^{16}$
- 64 264 =  $1.8 \times 10^{19}$
- 112 2112 =  $5.2 \times 10^{33}$
- 128 2128 =  $3.4 \times 10^{38}$
- 192 2192 =  $6.2 \times 10^{57}$
- 256 2256 =  $1.1 \times 10^{77}$

# Diffie-Hellman Key Agreement

Peer A



Peer B



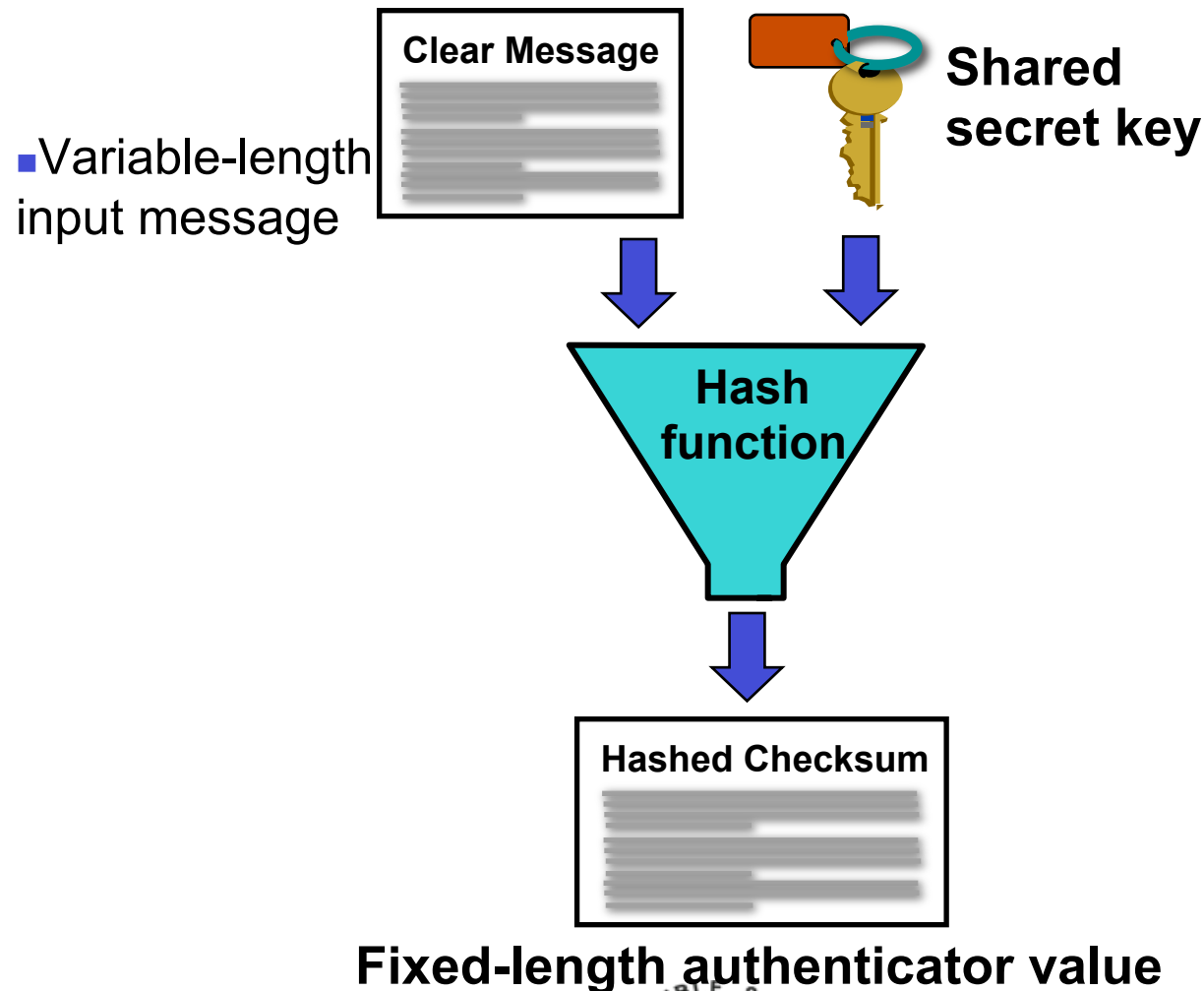
1. Generate large integer  $p$ .  
Send  $p$  to Peer B.  
Receive  $q$ .  
Generate  $g$ .
2. Generate private key  $X_A$
3. Generate public key  
 $Y_A = g^{X_A} \bmod p$
4. Send public key  $Y_A$
5. Generate shared secret  
number  $ZZ = Y_B^{X_A} \bmod p$
6. Generate shared secret key  
from  $ZZ$  (56-bit for DES,  
168-bit for 3DES)



1. Generate large integer  $q$ .  
Send  $q$  to Peer A.  
Receive  $p$ .  
Generate  $g$ .
2. Generate private key  $X_B$
3. Generate public key  
 $Y_B = g^{X_B} \bmod p$
4. Send public key  $Y_B$
5. Generate shared secret  
number  $ZZ = Y_A^{X_B} \bmod p$
6. Generate shared secret key  
from  $ZZ$  (56-bit for DES,  
168-bit for 3DES)

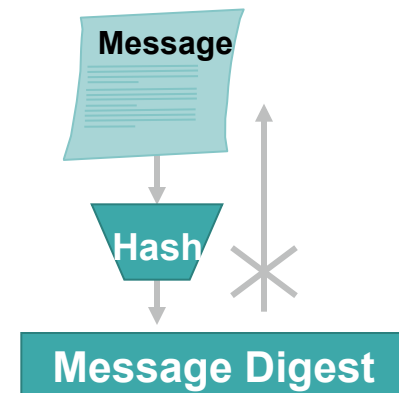
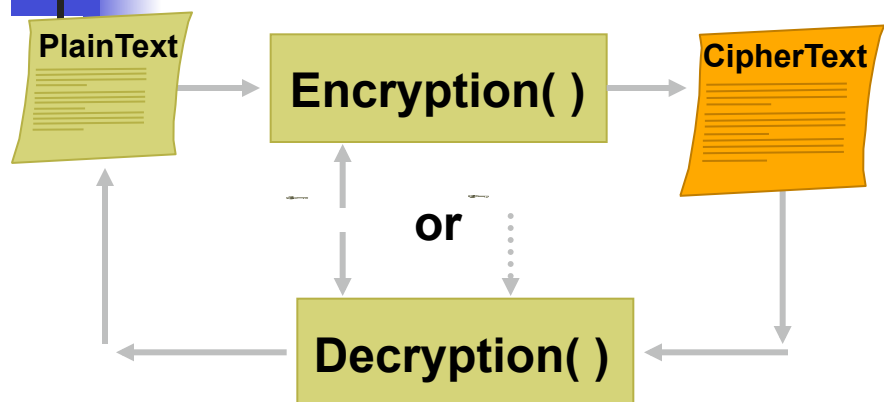


# Hashed Message Authentication Codes (HMAC)





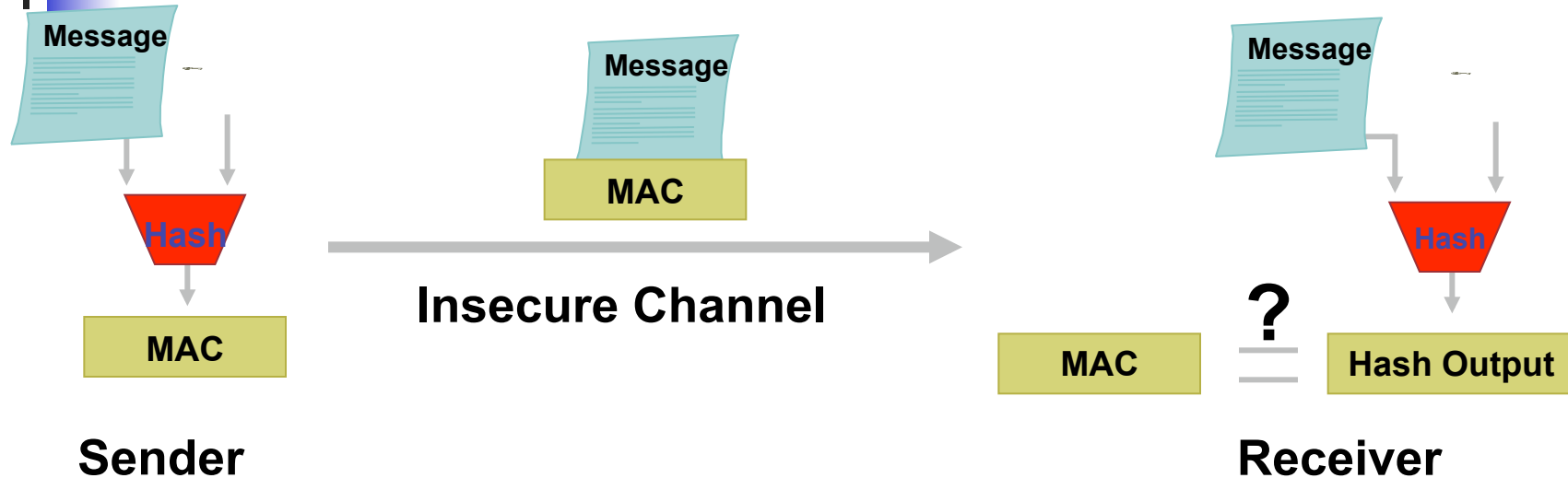
# Encryption vs. Hashing



- Encryption keeps communications private
- Encryption and decryption can use same or different keys
- Achieved by various algorithms, e.g. DES, CAST
- Need key management

- Hash transforms message into fixed-size string
- One-way hash function
- Strongly collision-free hash
- Message digest can be viewed as "digital fingerprint"
- Used for message integrity check and digital certificates
- Hash is generally faster than encryption

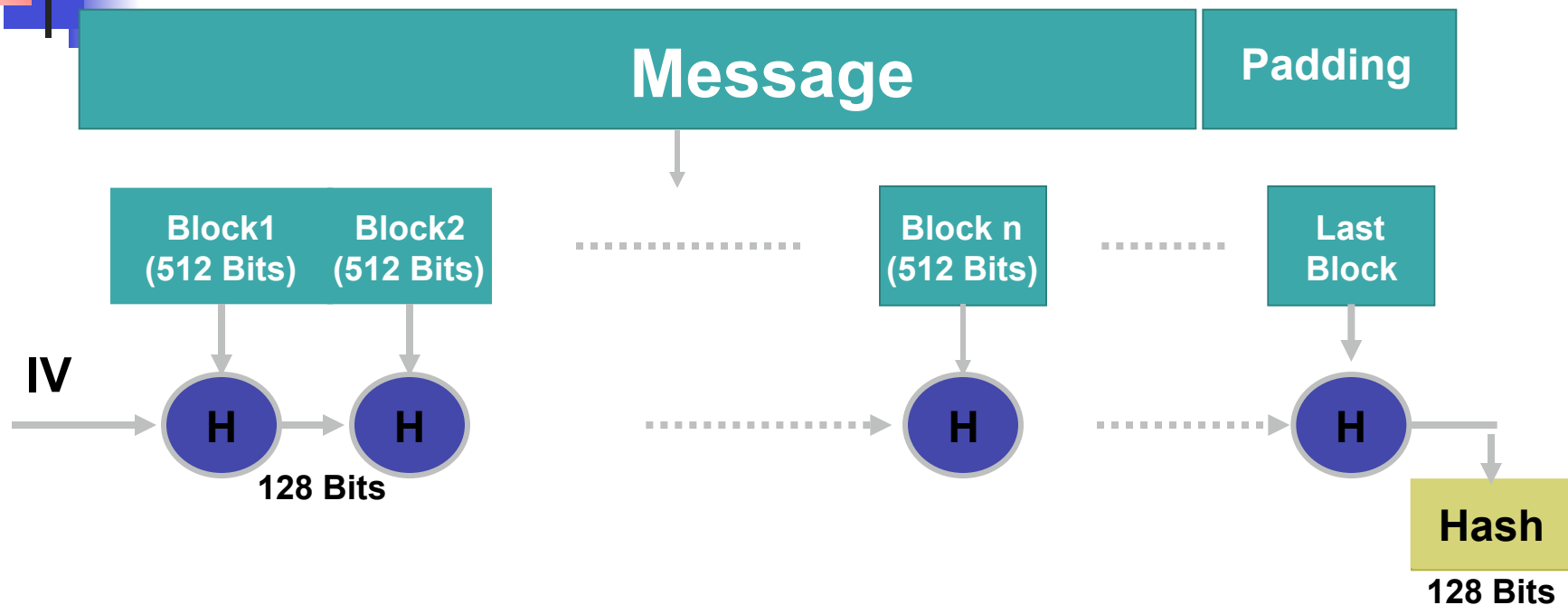
# Message Authentication and Integrity Check Using Hash



**Secret Key Only Known by Sender and Receiver**

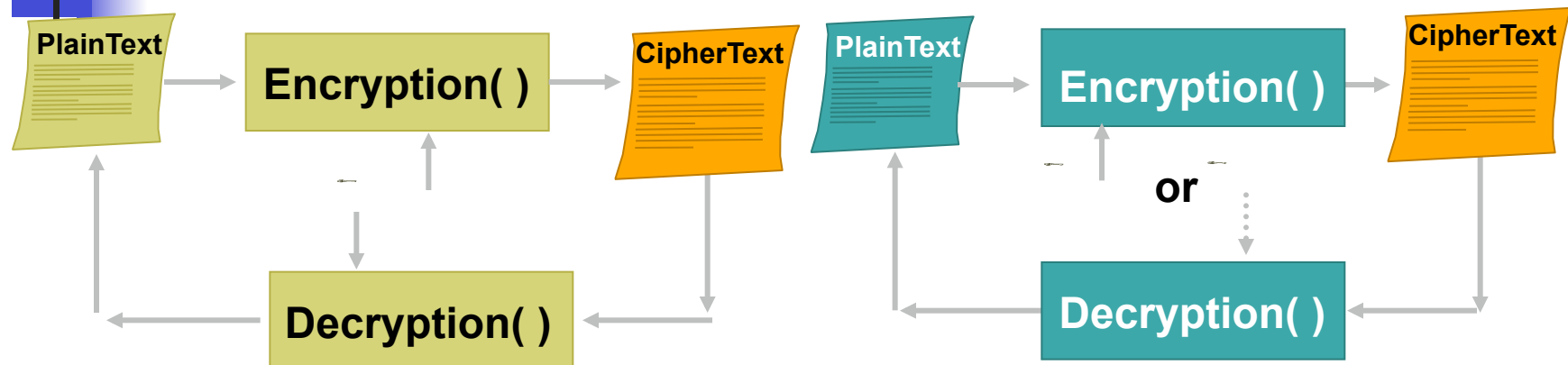
- MAC (Message Authentication Code): cryptographic checksum generated by passing data thru a message authentication algorithm
- MAC is often used for message authentication and integrity check
- HMAC—keyed hashed-based MAC

# Commonly Used Hash Functions (MD5 and SHA)



- Both MD5 and SHA are derived based on MD4
- MD5 provides 128-bit output, SHA provide 160-bit output; (only first 96 bits used in IPSec)
- Both of MD5 and SHA are considered **one-way strongly collision-free** hash functions

# Symmetric vs. Asymmetric Encryption Algorithms



- Secret-key cryptography
- Encryption and decryption use the same key
- Typically used to encrypt the content of a message
- Examples: DES

- Public-key cryptography
- Encryption and decryption use different keys
- Typically used in digital certification and key management
- Examples: Diffie-Hellman, RSA

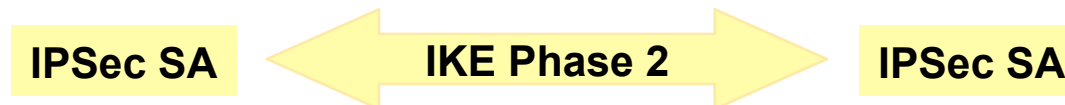
# Five Steps of IPSec



1. Host A sends interesting traffic to Host B.
2. Router A and B negotiate an IKE phase one session.



3. Router A and B negotiate an IKE phase two session.



4. Information is exchanged via IPSec tunnel.



5. IPSec tunnel is terminated.

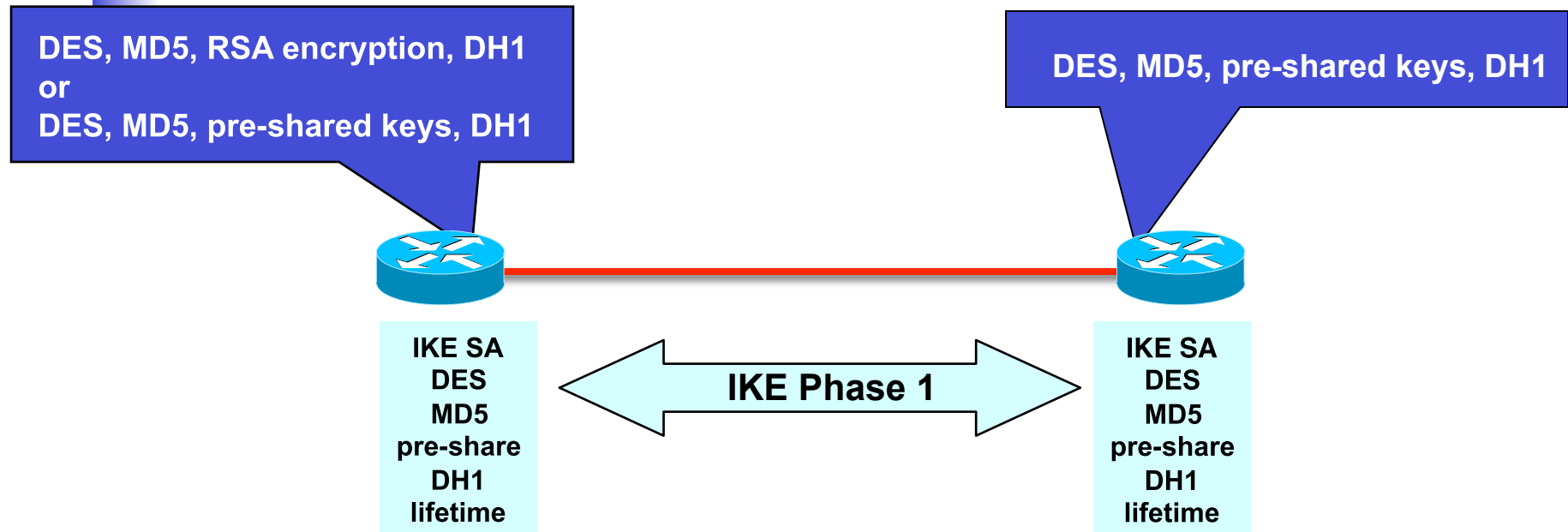
# Step 1—Interesting Traffic



```
access-list 101 permit ip 10.0.1.0 0.0.0.255 10.0.2.0 0.0.0.255
```

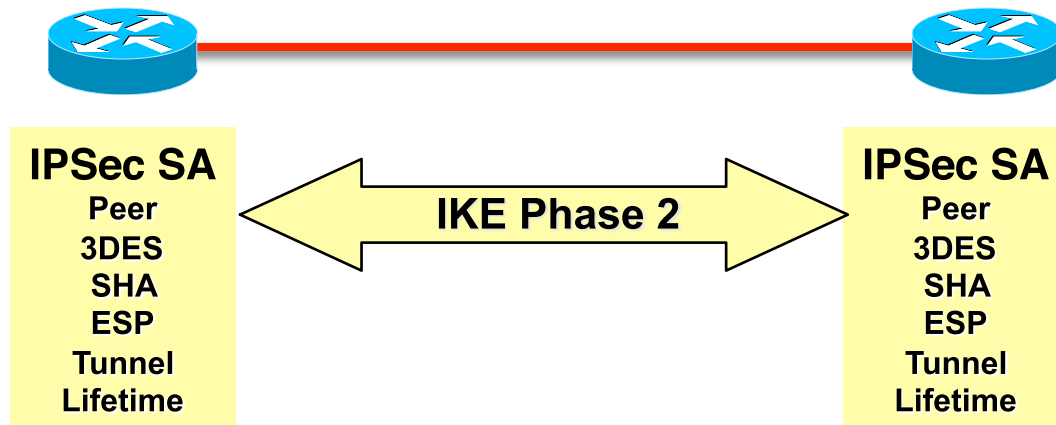
- Access lists determine traffic to encrypt
- Permit—traffic must be encrypted
- Deny—traffic sent unencrypted

# Step 2—IKE Phase One



- Authenticates IPSec peers
- Negotiates matching policy to protect IKE exchange
- Exchanges keys via Diffie-Hellman
- Establishes IKE security association

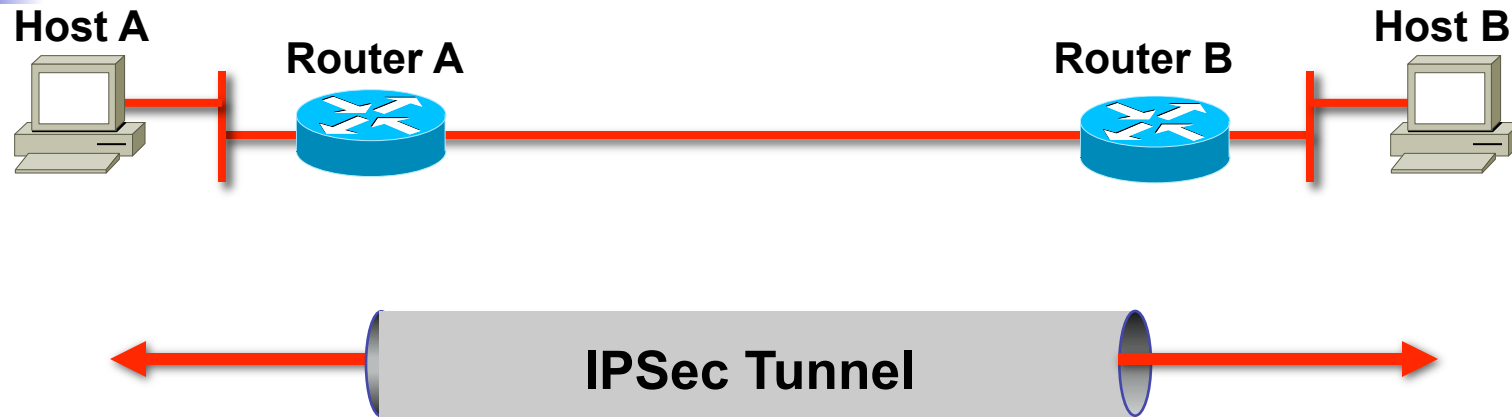
## Step 3—IKE Phase Two



- Negotiates IPsec SA parameters protected by an existing IKE SA
- Establishes IPsec security associations
- Periodically renegotiates IPsec SAs to ensure security
- Optionally performs an additional Diffie-Hellman exchange
- If perfect forward secrecy is specified, a new Diffie-Hellman exchange is performed with each quick mode.



# Step 4—IPSec Encrypted Tunnel



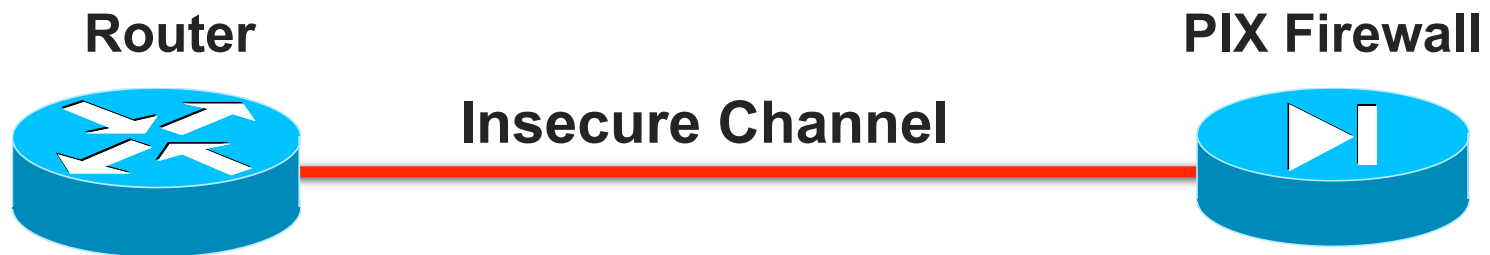
- Information is exchanged via IPSec tunnel.
- Packets are encrypted and decrypted.
- Uses encryption specified in IPSec SA.

# Step 5—Tunnel Termination



- Tunnel is terminated by
  - TCP session termination
  - SA lifetime timeout
  - Packet counter exceeded
- Removes IPSec SA

# Security Association



- Agreement between two entities on method to communicate securely
- IPSec SA is unidirectional
- Two-way communication consists of two SAs

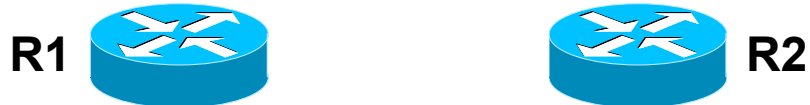


# IPSec SA

---

<b>Destination Address</b>	<b>192.168.2.1</b>
<b>Security Parameter Index (SPI)</b>	<b>7A390BC1</b>
<b>IPSec Transform</b>	<b>AH, HMAC-MD5</b>
<b>Key</b>	<b>7572CA49F7632946</b>
<b><i>Additional SA Attributes (for example, lifetime)</i></b>	<b>One Day or 100MB</b>

# SA Parameter Example for Cisco Routers



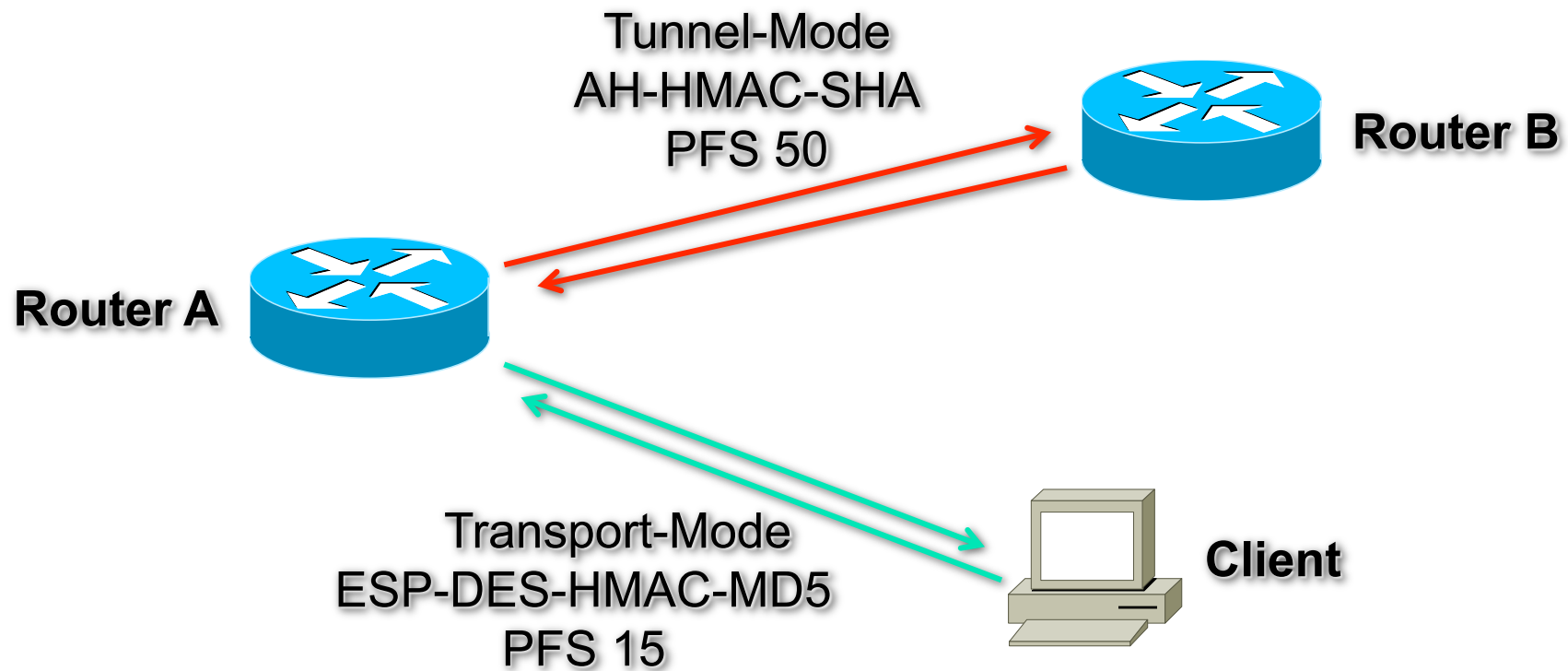
- outbound esp sas:
- spi: 0x1B781456(460854358)
- transform: esp-des ,
- in use settings ={Tunnel, }
- slot: 0, conn id: 18, crypto map:mymap
- sa timing: (k/sec)
- replay detection support: N

- inbound esp sas:
- spi: 0x8AE1C9C(145628316)
- transform: esp-des ,
- in use settings ={Tunnel, }
- slot: 0, conn id: 17, crypto map:mymap
- sa timing: (k/sec)
- replay detection support: N

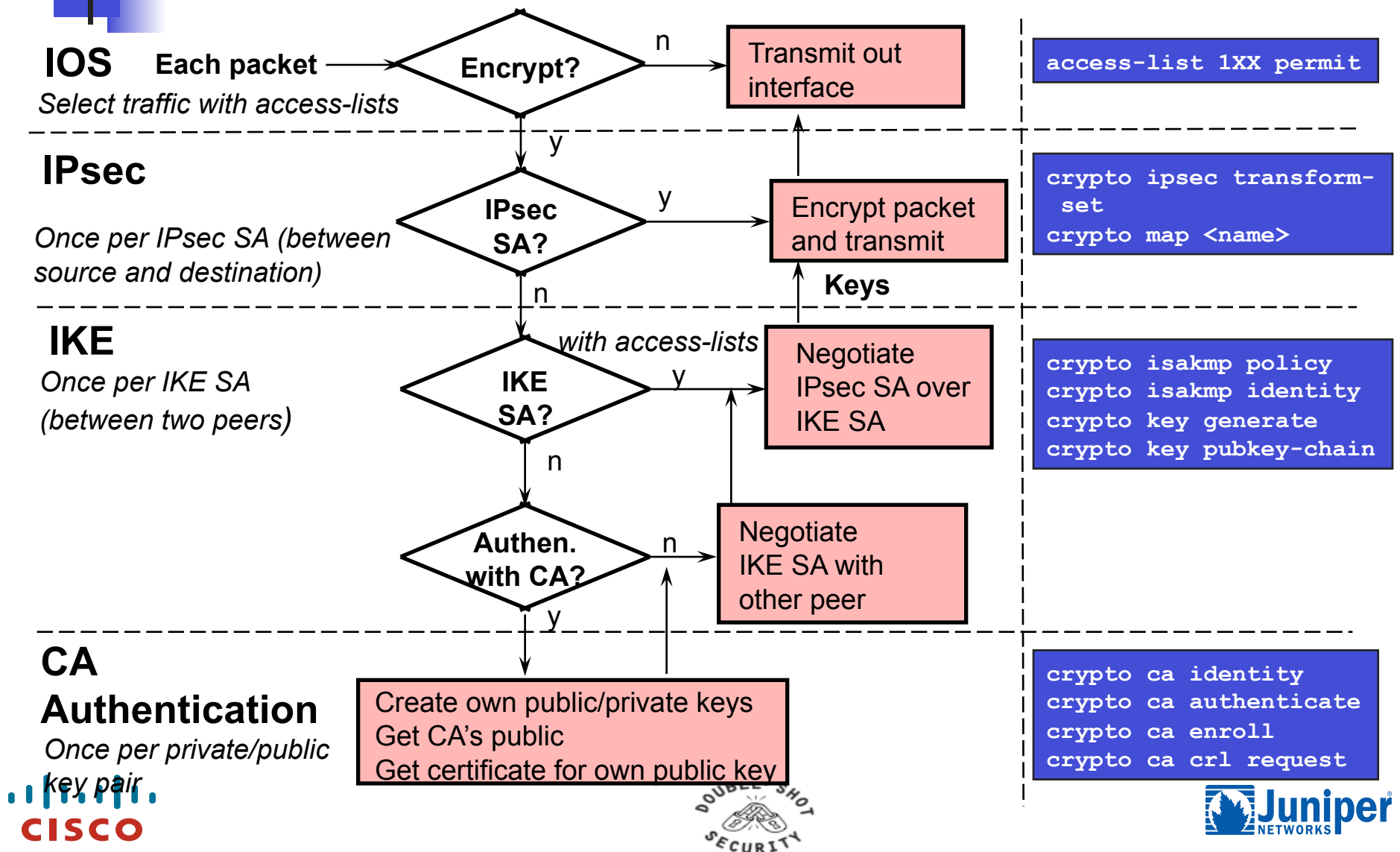
- inbound esp sas:
- spi: 0x1B781456(460854358)
- transform: esp-des ,
- in use settings ={Tunnel, }
- slot: 0, conn id: 18, crypto map:mymap
- sa timing: (k/sec)
- replay detection support: N

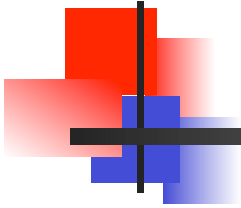
- outbound esp sas:
- spi: 0x8AE1C9C(145628316)
- transform: esp-des ,
- in use settings ={Tunnel, }
- slot: 0, conn id: 17, crypto map:mymap
- sa timing: (k/sec)
- replay detection support: N

# SAs Enable Your Chosen Policy

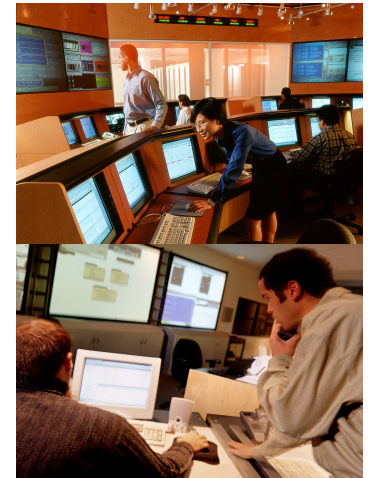
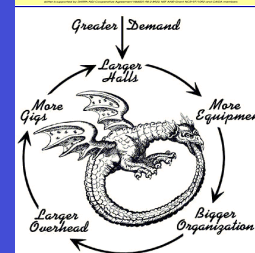
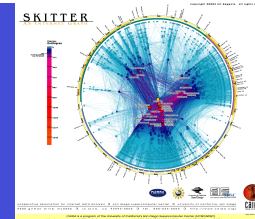


# IKE and IPsec Flowchart for Cisco Routers



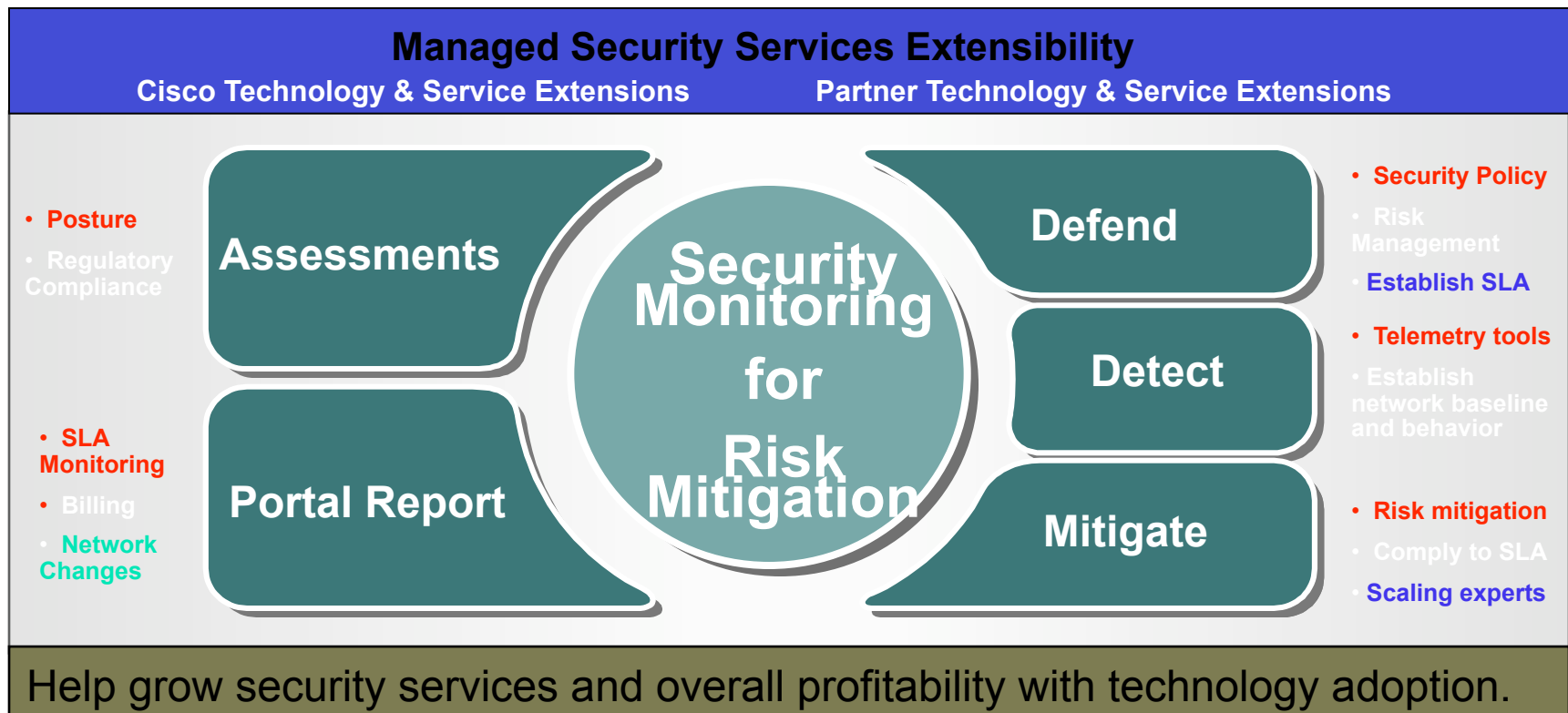
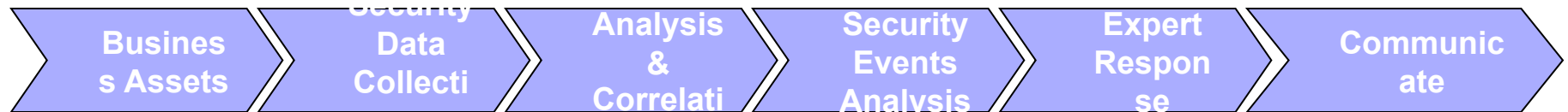


# Managed Security Services

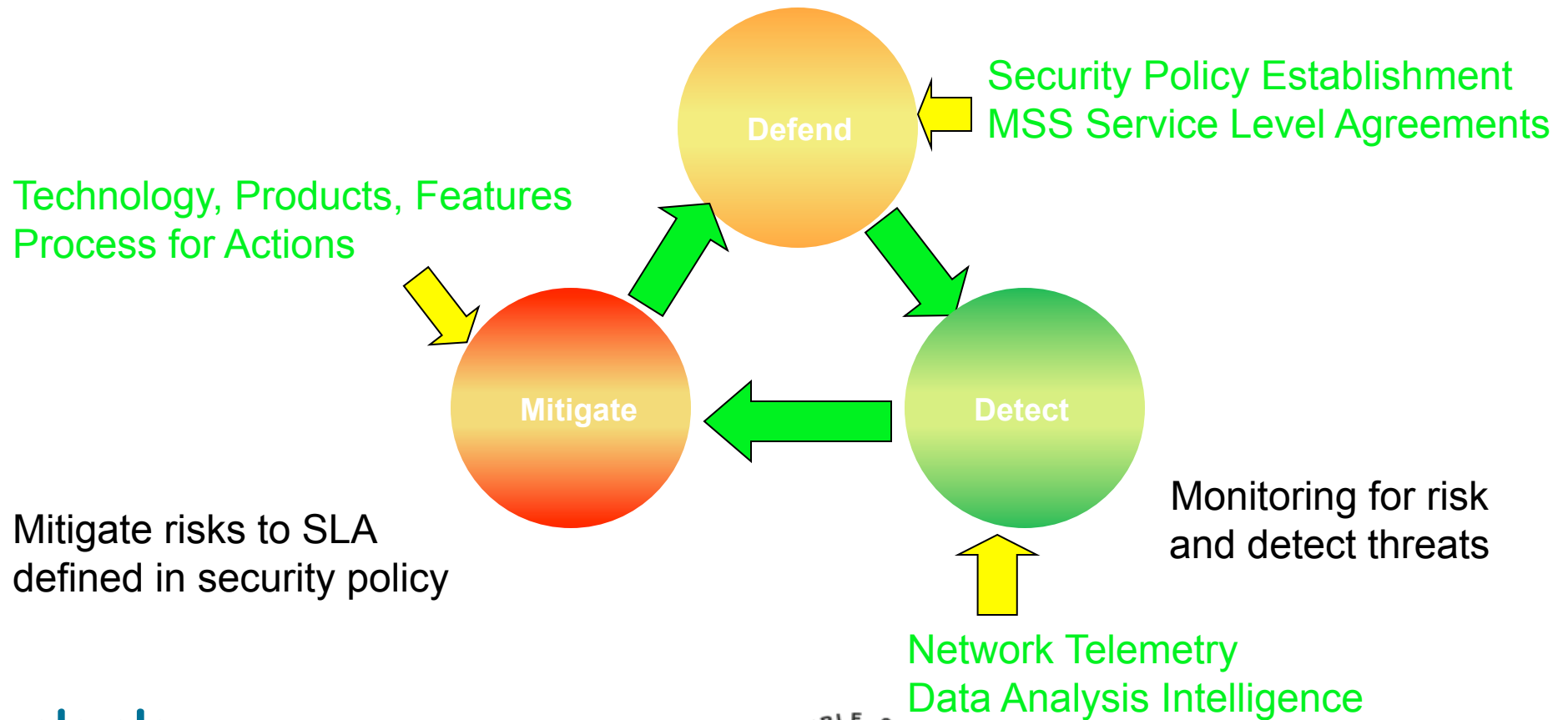




# Managed Security Services Architecture



## Security policy representation with risk mitigation plans



# MSS Offerings Strategy and Roadmap

## *Enhanced Security Services and Options*

Managed Content Security

Managed End Point Protection – NAC/CSA

Managed Security Services – IPS

**Enhanced Security Services and Options**

*Service Continuum*

Managed Security Services - IDS

Managed Security Service - VPN

Managed Security Services – Firewall

**Managed Router Service Connecting to the 'I'**

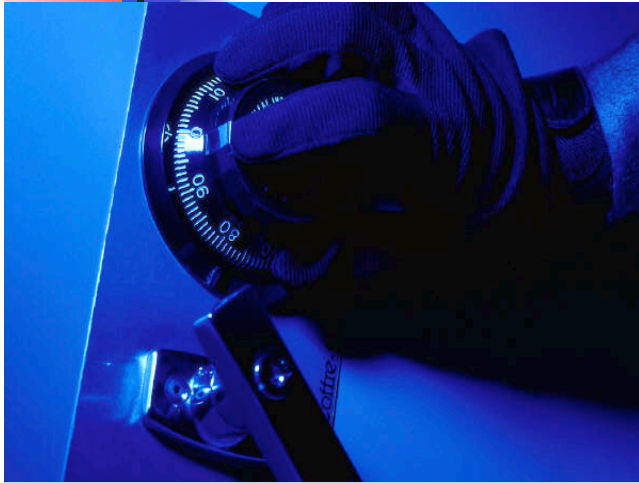
*Secure Access Services*

Revenue  
And  
Customer  
Retention

cisco



# Managed Firewall Service



**Integrated firewall  
results in  
operational savings**

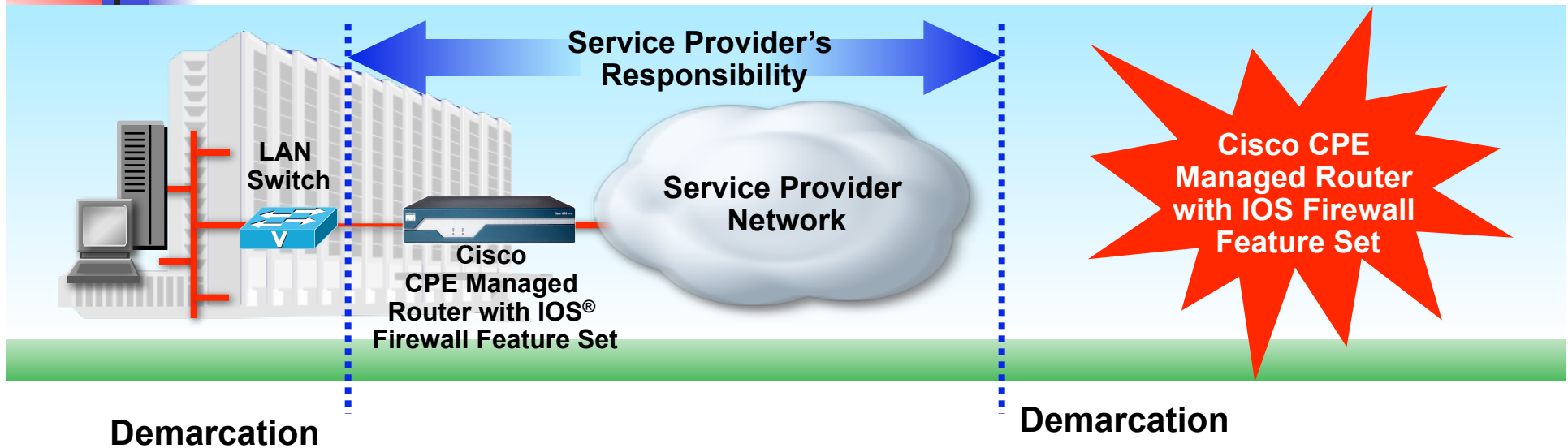
## Pain Points for Customers

- Protect internal and external networks
- Protect against embedded vulnerabilities
- Allow secure access to users

## Service Opportunity for SP

- Value-add on transport provision
- ISR run firewall with hardware acceleration

# Managed Firewall Service In Details



- Managed router service allows remote configuration
- SP enables firewall on a managed router
- Basic firewall allows split tunneling and dynamic site-to-site connections
- Advanced firewall allows application filtering to comply with security policies
- Provide Managed IOS Firewall Service without truck-roll
- Obtain new service revenue by already deployed and managed CPE

# Managed VPN Service



**Integrated VPN  
results in new  
users and locations**

**CISCO**

## Pain Points for Customers

- Secure scalable connectivity
- Address regulatory requirements

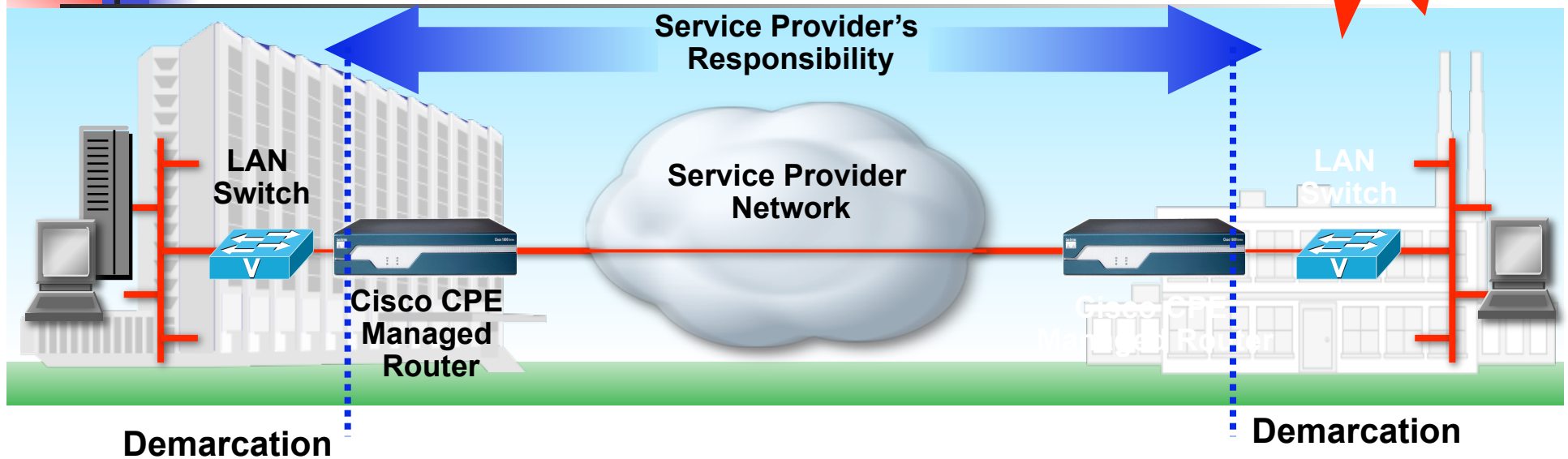
## Service Opportunity for SP

- Extend network to new users and locations
- Remote management ensure optimal performance and scale



# Managed Site-to-Site VPN Service

**Cisco CPE  
Managed Router  
with Encryption**



- Managed router service allows remote configuration
- SP enables Site-to-Site MPLS or IPSec VPN features in Cisco Router
- Extend support for VPN Acceleration from SP to customer premise through AES wide-key support in both
- Provide Managed Site-to-Site VPN Service without truck-roll
- Obtain new service revenue from already deployed and managed CPE

# Managed IDS/IPS Service



**Integrated IPS  
results in increased  
network visibility**

**CISCO**

## Pain Points for Customers

- Attack mitigation and threat prevention distributed at all network entry points
- Identify, classify and stop malicious traffic in real-time

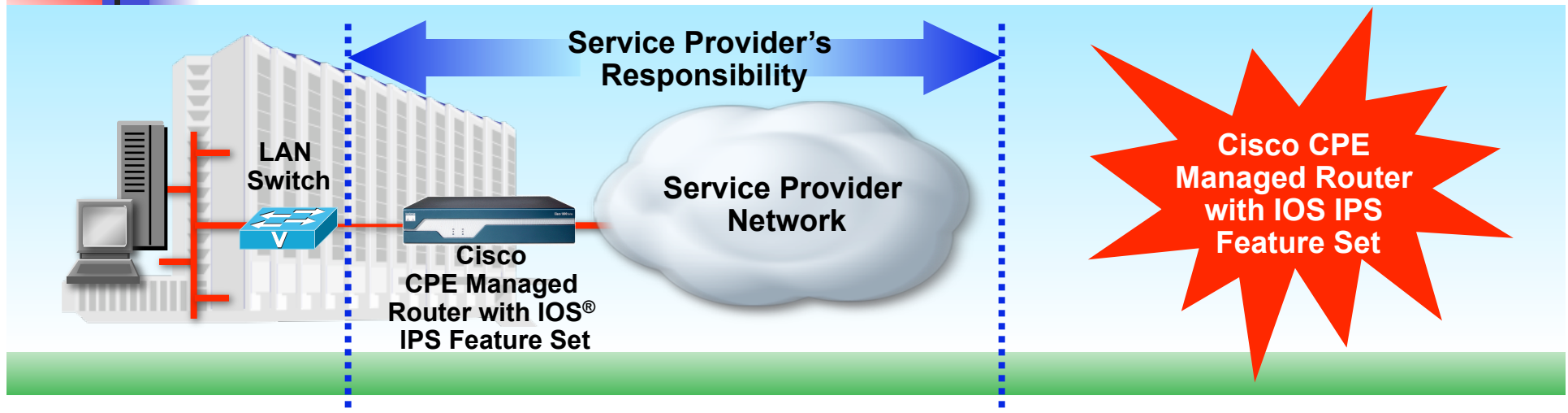
## Service Opportunity for SP

- Add value by real time monitoring and threat mitigation for customers
- Create loyalty by customizing solutions for unique customer security policy
- Provide remote management for security service and provide comprehensive reporting on the security events





# Managed IDS/IPS Service In Details



- SP configures IOS IDS/IPS Feature Set in Cisco Router by configuration management feature of Managed Router Service
- Provide Managed IOS Firewall Service and Intrusion Protection Service without truck-roll
- Provide Inline IPS option- customizable signatures can be dynamically loaded
- Obtain new service revenue by already deployed and managed CPE

# Managed Endpoint Protection Service



CISCO

## Pain Points for Customers

- Protection Beyond the Perimeter
- Detect and mitigate

## Service Opportunity for SP

- Emerging service opportunity to further penetrate customer network and LAN environment
- Leverage the network to intelligently enforce access privileges based on endpoint security posture



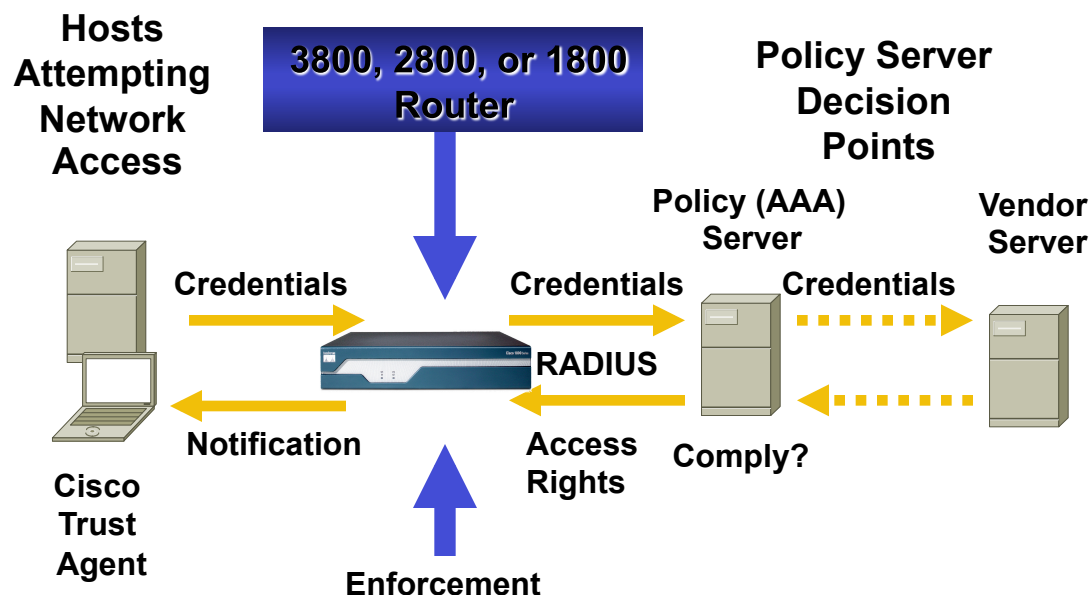
# Introducing NAC



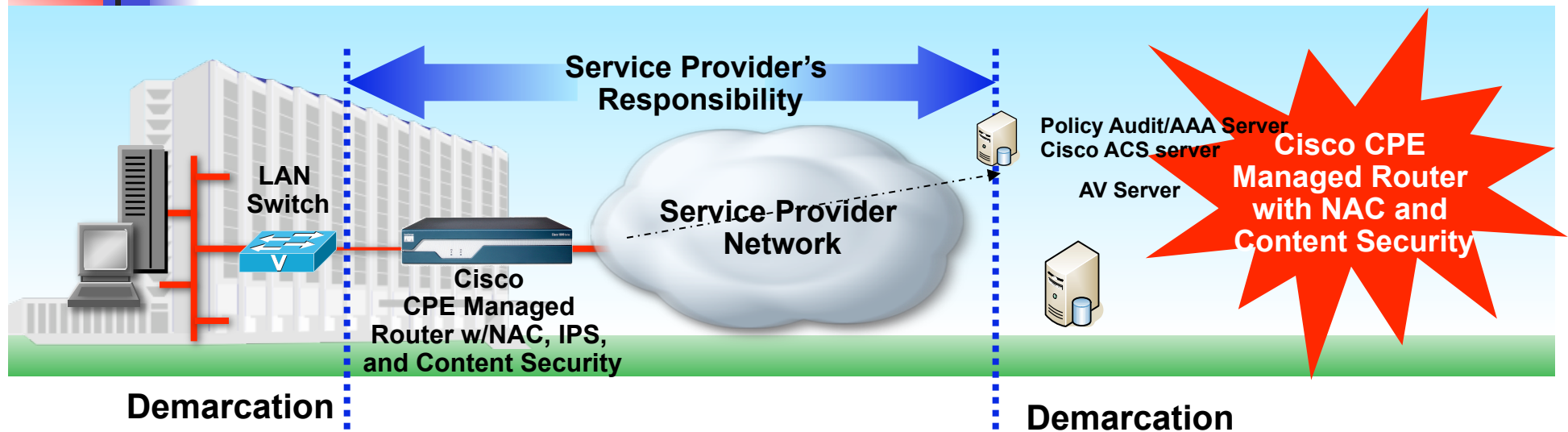
Coalition of  
industry leading  
partners

CISCO

## Network Admission Control



# Managed Endpoint Protection Service In Details

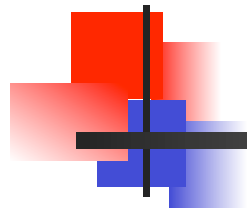


Cisco Network Admission Control technology within CPE Router

- ISR as a CPE is a security policy enforcement point and provides visibility to the network behavior
- Managed ISR supports access control and identity
- Network collaborates with applications
- Layer 2 and Layer 3 collaboration provides in per user level policy management
- Facilitate security policy audit and compliance

■ NAC is an enhanced service so obtain new service revenue by already deployed and managed CPE





# Managed Services Lab

---

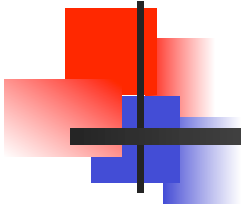




# Conclusions

---

- What is the first step after the workshop?
  - Be part of the community
  - Consider iNOC/DBA
  - Utilize NetFlow / SNMP / SysLog and Open Source tools for Telemetry
  - Deploy RTBH for Source and Destination based dropping
- What Managed Security Services your organization can offer?
  - Managed VPN Services: MPLS and IPSec
  - Managed Firewall Services



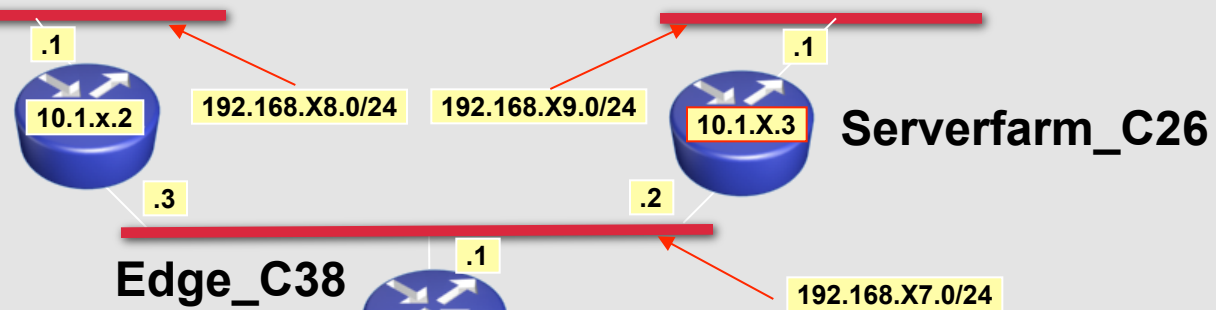
## Q and A



# APRICOT 2008 ISP Security L3VPN Lab

## Hub Site

Trigger\_J63



BGP AS 65412

OSPF area 0

IBGP backbone



Attacker

172.16.0.0/24

.X1

## Remote Site

Remote\_J23

192.168.X.0/24

.1

.100

FreeBSD

