Chapter 3 – Using Maintenance & Troubleshooting Tools & Applications Objectives

- Describe & utilise Cisco IOS diagnostic tools.
- Explain the need for specialist tools in the troubleshooting process.
- Configure software to allow packet captures.
- Explain how to create a network baseline using Netflow, SNMP, IP SLA & NBAR.

Filtering Cisco IOS Outputs

- Cisco IOS offers multiple show commands useful for gathering information. However, many of these show commands produce a large quantity of output:

```
R1# show ip route 10.1.193.3
Routing entry for 10.1.193.0/30
Known via "connected", distance 0, metric 0 (connected, via interface)
Redistributing via eigrp 1
Routing Descriptor Blocks:
* directly connected, via Serial0/0/1
Route metric is 0, traffic share count is 1

R1# show ip route 10.1.193.10
% Subnet not in table
```

•If gateway of last resort (default route) is present in the IP routing table, but no entry matches the IP address you entered, the router again responds with the **% Subnet not in table** message even though packets for that destination are forwarded using the gateway of last resort

Filtering Cisco IOS Outputs

```
R1# show ip route 10.1.193.0 255.255.255.0 longer-prefixes
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
10.0.0.0/8 is variably subnetted, 46 subnets, 6 masks
C 10.1.193.2/32 is directly connected, Serial0/0/1
C 10.1.193.0/30 is directly connected, Serial0/0/1
D 10.1.193.6/32 [90/20517120] via 10.1.192.9, 2d01h, FastEthernet0/1
[90/20517120] via 10.1.192.1, 2d01h, FastEthernet0/0
D 10.1.193.4/30 [90/20517120] via 10.1.192.9, 2d01h, FastEthernet0/1
[90/20517120] via 10.1.192.1, 2d01h, FastEthernet0/0
D 10.1.193.5/32 [90/41024000] via 10.1.194.6, 2d01h, Serial0/0/0.122
```

Filtering Cisco IOS Outputs

- Unfortunately, show commands do not always have the option that allows you filter the output down to exactly what you need. You can still perform a more generic way of filtering:

```
R1# show ip interface brief
Interface              IP-Address        OK?    Method  Status       Protocol
FastEthernet0/0        192.168.1.11      YES    NVRAM up             up
Serial0/0              unassigned        YES    NVRAM admin down     down
FastEthernet0/1        192.168.0.11      YES    NVRAM up             up
Serial0/1              unassigned        YES    NVRAM admin down     down
NVI0                   unassigned        YES    unset    up          up
Loopback0              10.1.1.1          YES    NVRAM up             up
```

- The output of Cisco IOS show commands can be filtered by appending a pipe character (|) to the show command followed by one of the keywords **include**, **exclude**, or **begin**:

```
R1# show ip interface brief | exclude unassigned
Interface        IP-Address      OK?     Method  Status               Protocol
FastEthernet0/0  192.168.1.11    YES     NVRAM   up                   up
FastEthernet0/1  192.168.0.11    YES     NVRAM   up                   up
Loopback0        10.1.1.1         YES    NVRAM    up                  up
```

Filtering Cisco IOS Outputs

```
R1# show processes cpu | include IP Input
71 3149172    7922812         397 0.24% 0.15% 0.05% 0 IP Input
```

```
SW1# show running-config | begin line vty
line vty 0 4
transport input telnet ssh
line vty 5 15
transport input telnet ssh
```

Filtering Cisco IOS Outputs

- Cisco IOS Software Release (12.3(2)T) introduced the *section* option, which allows you to select and display a specific section or lines from the configuration that match a particular regular expression and any following associated lines:

```
R1# show running-config | section router eigrp
router eigrp 1
network 10.1.192.2 0.0.0.0
network 10.1.192.10 0.0.0.0
network 10.1.193.1 0.0.0.0
no auto-summary
```

Saving Cisco IOS Outputs

- Other useful options that can be used with the pipe character after the show command are redirect, tee, and append.
- The output of a show command can be redirected, copied or appended to a file by using the pipe character, followed by the options **redirect**, **tee**, or **append** and a URL that denotes the file.

```
R1# show ip int brief | redirect tftp://192.168.37.2/show-stuff.txt
```

The redirect option *does not* display the output on the console

```
R1# show ip interface brief | tee flash:show-stuff.txt
```

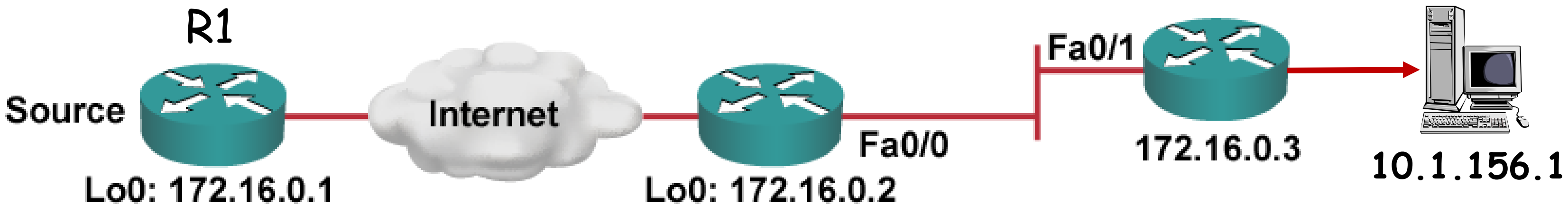
The tee option *displays* the output on the console and *sends* it to the file

```
R1# dir flash:
Directory of flash:/
1  -rw-     23361156           Mar 2 2009 16:25:54 -08:00 c1841-advipservicesk9mz.1243.bin
2 -rw- 680 Mar 7 2009 02:16:56 -08:00 show-stuff.txt
```

```
R1# show ip interface brief | append flash:show-stuff.txt
```

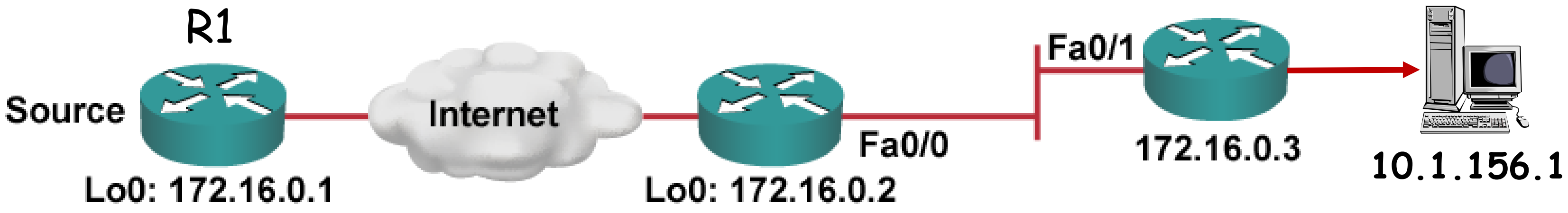
The append option allows you to *add* the command output to an existing file

Testing Network Connectivity with Ping



```
R1# ping 10.1.156.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.156.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

R1# ping 10.1.156.1 source lo0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.156.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.192.2
.....
Success rate is 0 percent (0/5)
```

Testing Network Connectivity with Ping



•By setting the *df-bit* option and combining it with the *size* option, you can force routers along the path to drop the packets if they would have to *fragment* them.

• By varying the size and looking at which point the packets start being dropped, you can determine the *MTU.*

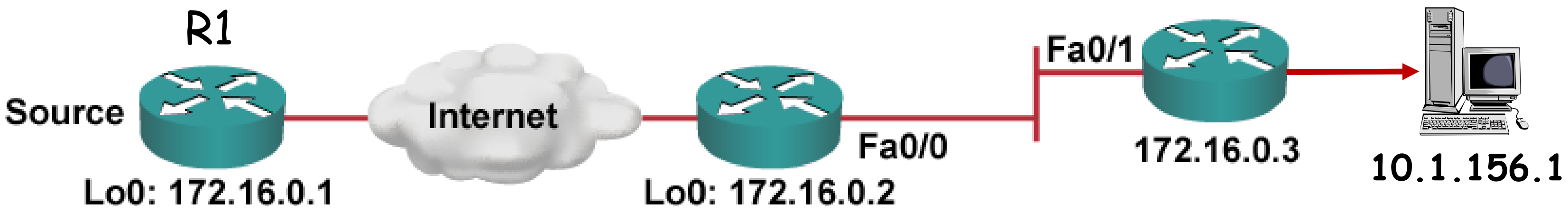
```
R1# ping 10.1.156.1 size 1476 df-bit
Type escape sequence to abort.
Sending 5, 1476-byte ICMP Echos to 10.1.156.1, timeout is 2 seconds:
Packet sent with the DF bit set
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 184/189/193 ms

R1# ping 10.1.156.1 size 1477 df-bit
Type escape sequence to abort.
Sending 5, 1477-byte ICMP Echos to 10.1.156.1, timeout is 2 seconds:
Packet sent with the DF bit set
M.M.M
Success rate is 0 percent (0/5)
```

Testing Network Connectivity with Ping

```
R1# ping
Protocol [ip]:
Target IP address: 10.1.156.1
Repeat count [5]: 1
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface:
Type of service [0]:
Set DF bit in IP header? [no]: yes
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Time stamp, Verbose[none]:
Sweep range of sizes [n]: y
Sweep min size [36]: 1400
Sweep max size [18024]: 1500
Sweep interval [1]:
Type escape sequence to abort.
Sending 101, [1400..1500]-byte ICMP Echos to 10.1.156.1, timeout is 2 seconds:
Packet sent with the DF bit set
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!M.M.M.M.M.M.M.M.M.M.M.M.
Success rate is 76 percent (77/101), round-trip min/avg/max = 176/184/193 ms
```

- The router is instructed to send packets starting at a size of 1400 bytes, sending a single packet per size and increasing the size one byte at a time until a size of 1500 bytes is reached.
- M = Could not fragment.

Testing Network Connectivity with Telnet



- Telnet is an excellent companion to ping for testing *transport* layer connections from the command line.
- Telnet server applications use port 23, but you can specify a specific port number on the client and connect to any TCP port that you want to test.

```
R1# telnet 10.1.156.1 80
Trying 10.1.156.1 , 80 ... Open
GET
<html><body><h1>It works!</h1></body></html>
[Connection to 192.168.37.2 closed by foreign host]

R1# telnet 10.1.156.1 25
Trying 10.1.156.1, 25 ...
% Connection refused by remote host
```

Collecting Real Time Information – Cisco Debug Commands

- Because debugging output is assigned high priority in the CPU process, it can render the system *unusable*.
- Therefore, use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff.

```
R1 #debug ip packet
IP: s=172.69.13.44 (Fddi0), d=10.125.254.1 (Serial2), g=172.69.16.2, forward
IP: s=172.69.1.57 (Ethernet4), d=10.36.125.2 (Serial2), g=172.69.16.2, forward
IP: s=172.69.1.6 (Ethernet4), d=255.255.255.255, rcvd 2
IP: s=172.69.1.55 (Ethernet4), d=172.69.2.42 (Fddi0), g=172.69.13.6, forward
IP: s=172.69.89.33 (Ethernet2), d=10.130.2.156 (Serial2), g=172.69.16.2, forward
```

```
R1 #debug ip rip
RIP: received v2 update from 10.1.1.2 on Serial0/0/0
30.0.0.0/8 via 0.0.0.0 in 1 hops
RIP: sending v2 update to 224.0.0.9 via FastEthernet0/0 (20.1.1.1)
RIP: build update entries
10.0.0.0/8 via 0.0.0.0, metric 1, tag 0
30.0.0.0/8 via 0.0.0.0, metric 2, tag 0
RIP: sending v2 update to 224.0.0.9 via Serial0/0/0 (10.1.1.1)
```

R1 #sh debug - display active debugs

R1 #u all – switch off all currently active debugs

Diagnosing Hardware Issues Using Cisco IOS Commands

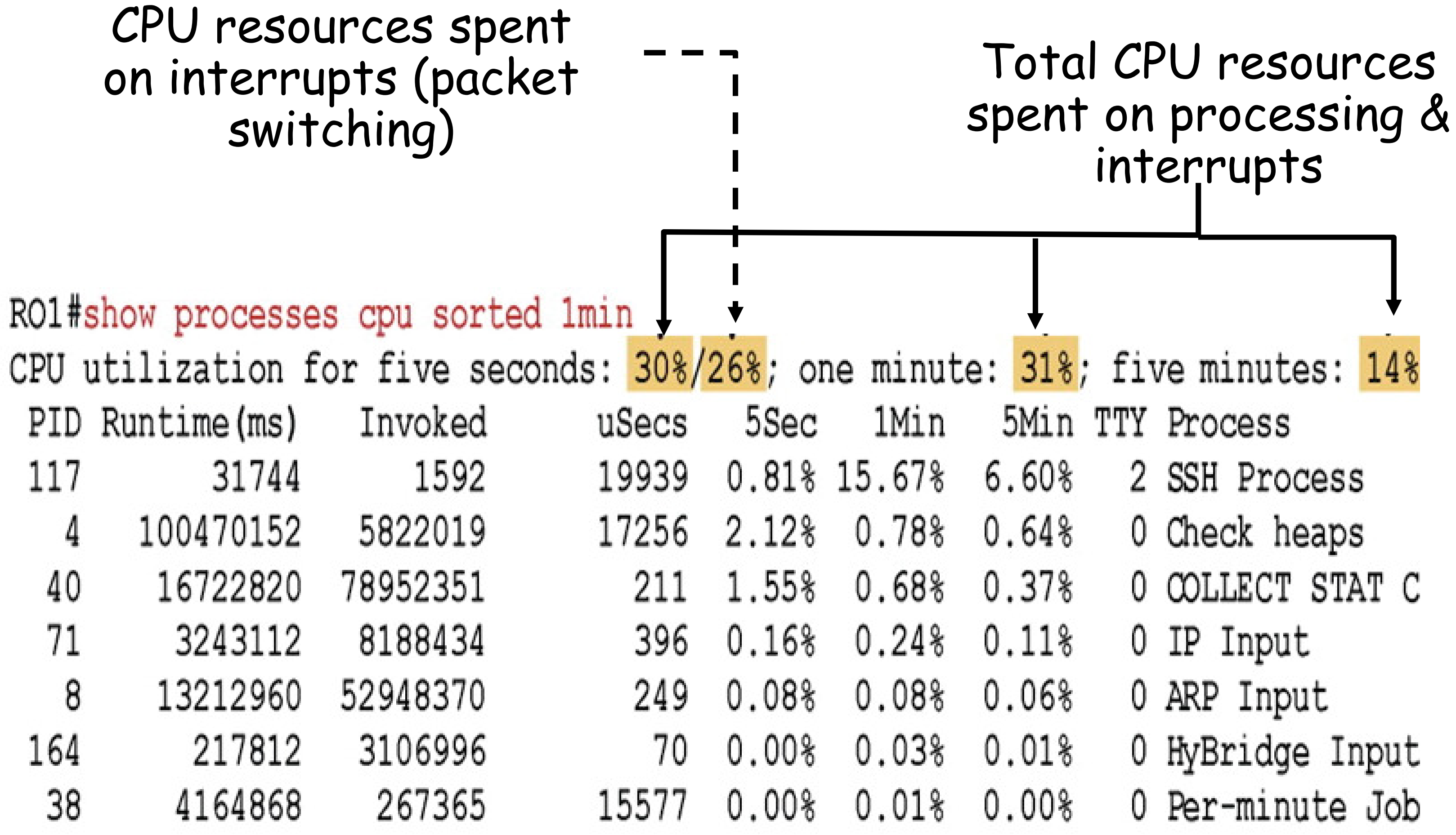
- Due to its nature, diagnosing hardware problems is highly product and platform dependent.
- However, you can use a number of generic commands to diagnose performance-related hardware issues on all Cisco IOS platforms:

1. Show processes cpu
2. Show memory
3. Show interface

Checking CPU Utilisation

- The same CPU that is used to run the operating system processes is also responsible for packet switching.

- The CPU is interrupted to suspend the current process that it is executing, switch one or more packets, and resume the execution of scheduled processes.



CPU resources spent on interrupts (packet switching)

Total CPU resources spent on processing & interrupts

```
R01#show processes cpu sorted 1min
CPU utilization for five seconds: 30%/26%; one minute: 31%; five minutes: 14%
 PID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min TTY Process
 117       31744      1592      19939  0.81% 15.67%  6.60%   2 SSH Process
   4   100470152   5822019      17256  2.12%  0.78%  0.64%   0 Check heaps
  40    16722820  78952351        211  1.55%  0.68%  0.37%   0 COLLECT STAT C
  71     3243112   8188434        396  0.16%  0.24%  0.11%   0 IP Input
   8    13212960  52948370        249  0.08%  0.08%  0.06%   0 ARP Input
 164      217812   3106996         70  0.00%  0.03%  0.01%   0 HyBridge Input
  38     4164868    267365      15577  0.00%  0.01%  0.00%   0 Per-minute Job
```

Checking Memory Utilisation

- Similar to CPU cycles, memory is a finite resource shared by the various processes that together form the Cisco IOS operating system.

- Memory is divided into different pools and used for different purposes: the processor pool contains memory that can be used by the *scheduled* processes, and the I/O pool is used to *temporarily* buffer packets during packet switching.

```
R1# show memory

                 Head      Total(b)   Used(b)   Free(b)  Lowest(b)  Largest(b)
Processor   820B1DB4  26534476  19686964  6847512   6288260    6712884
I/O         3A00000   6291456   3702900   2588556 2511168      2577468
```

- It is useful to create a baseline of the memory usage on routers and switches and graph the utilisation over time.

- If a router or switch does not have enough free memory to satisfy the request of a process, it will log a memory allocation failure, signified by a **%SYS-2-MALLOCFAIL**

Checking Interfaces

```
R1# show interfaces FastEthernet 0/0
FastEthernet0/0 is up, line protocol is up
<…output omitted…>
Last input 00:00:00, output 00:00:01, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/1120/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 2000 bits/sec, 3 packets/sec
5 minute output rate 0 bits/sec, 1 packets/sec
110834589 packets input, 1698341767 bytes
Received 61734527 broadcasts, 0 runts, 0 giants, 565 throttles
30 input errors, 5 CRC, 1 frame, 0 overrun, 25 ignored
0 watchdog
0 input packets with dribble condition detected
35616938 packets output, 526385834 bytes, 0 underruns
0 output errors, 0 collisions, 1 interface resets
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier
0 output buffer failures, 0 output buffers swapped out
```

Checking Interfaces - Filtered

- If you repeatedly want to display selected statistics to see how
- the counters are increasing, it is useful to filter the output.
- Using a *regular* expression to include only the lines in which you are interested can prove quite useful in this case

```
R1# show interfaces FastEthernet 0/0 | include ^Fast|errors|packets
FastEthernet0/0 is up, line protocol is up
5 minute input rate 3000 bits/sec, 5 packets/sec
5 minute output rate 2000 bits/sec, 1 packets/sec
2548 packets input, 257209 bytes
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
0 input packets with dribble condition detected
610 packets output, 73509 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
```

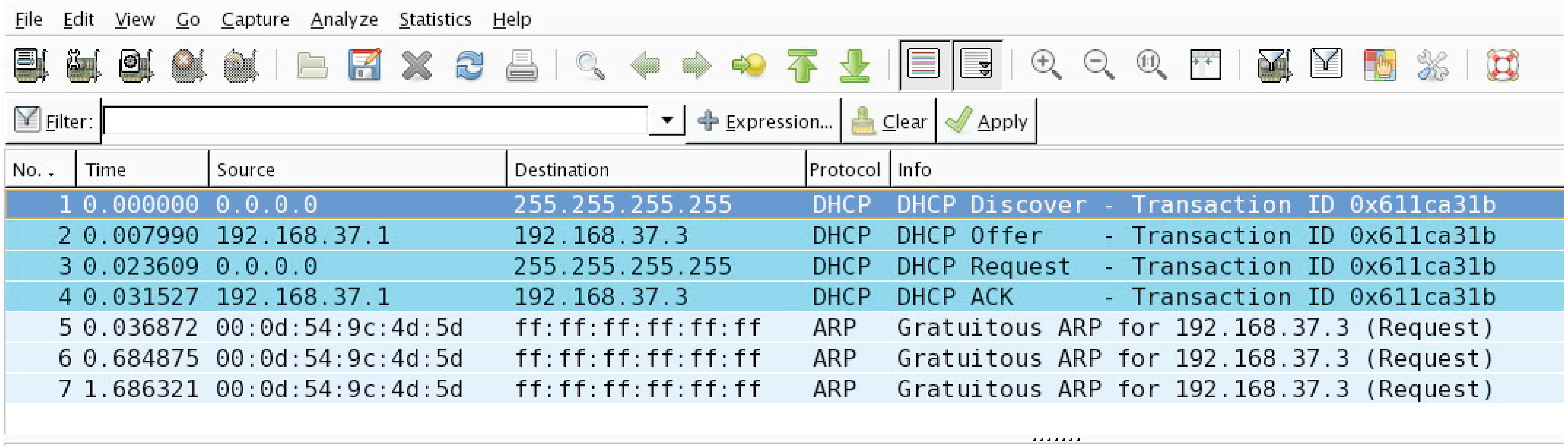
- Caret (^) = match string if it occurs at the beginning of a line.
- Pipe(|) = logical OR

Traffic Capturing Tools

- Information *gathering* is essential to both troubleshooting and maintenance.
- Information is either gathered on a *need basis*, such as during a troubleshooting effort, or *continuously* as part of baseline creation.
- A troubleshooter need to be able to:
  1. Enable Switched Port Analyser (SPAN) and Remote SPAN (RSPAN) to facilitate the use of packet sniffers.
  2. Configure routers and switches for communication with Simple Network Management Protocol (SNMP) or NetFlow-based network management systems to facilitate the collection of device and traffic statistics that are part of a network baseline.
  3. Configure routers and switches to send SNMP traps to provide fault notification to SNMP based network management systems.

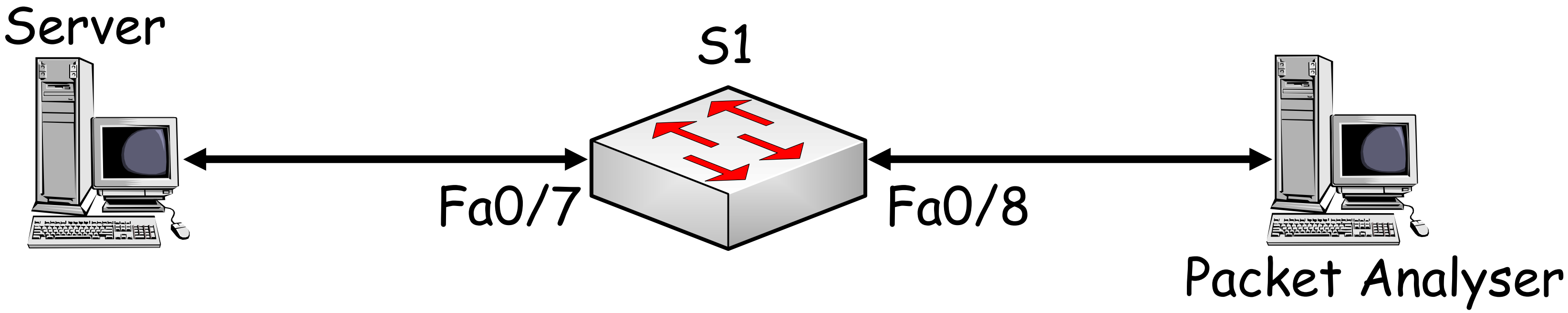
Traffic Capturing Tools – Protocol Analyser

- Packet sniffers, or protocol analysers, are important and useful tools for network engineers. Using these tools, you can look for and observe protocol errors like retransmissions or session resets.
- Packet sniffers are powerful tools because they capture large amounts of very detailed data – use *filtering* so that only the traffic you are interested in is displayed.



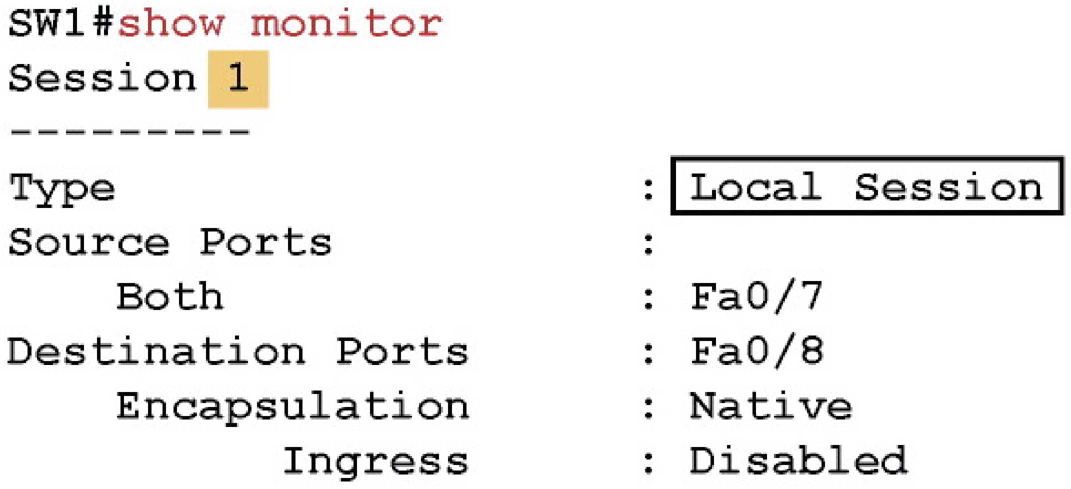
No.	Time	Source	Destination	Protocol	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	DHCP Discover - Transaction ID 0x611ca31b
2	0.007990	192.168.37.1	192.168.37.3	DHCP	DHCP Offer - Transaction ID 0x611ca31b
3	0.023609	0.0.0.0	255.255.255.255	DHCP	DHCP Request - Transaction ID 0x611ca31b
4	0.031527	192.168.37.1	192.168.37.3	DHCP	DHCP ACK - Transaction ID 0x611ca31b
5	0.036872	00:0d:54:9c:4d:5d	ff:ff:ff:ff:ff:ff	ARP	Gratuitous ARP for 192.168.37.3 (Request)
6	0.684875	00:0d:54:9c:4d:5d	ff:ff:ff:ff:ff:ff	ARP	Gratuitous ARP for 192.168.37.3 (Request)
7	1.686321	00:0d:54:9c:4d:5d	ff:ff:ff:ff:ff:ff	ARP	Gratuitous ARP for 192.168.37.3 (Request)

Traffic Capturing Tools – Protocol Analyser using SPAN



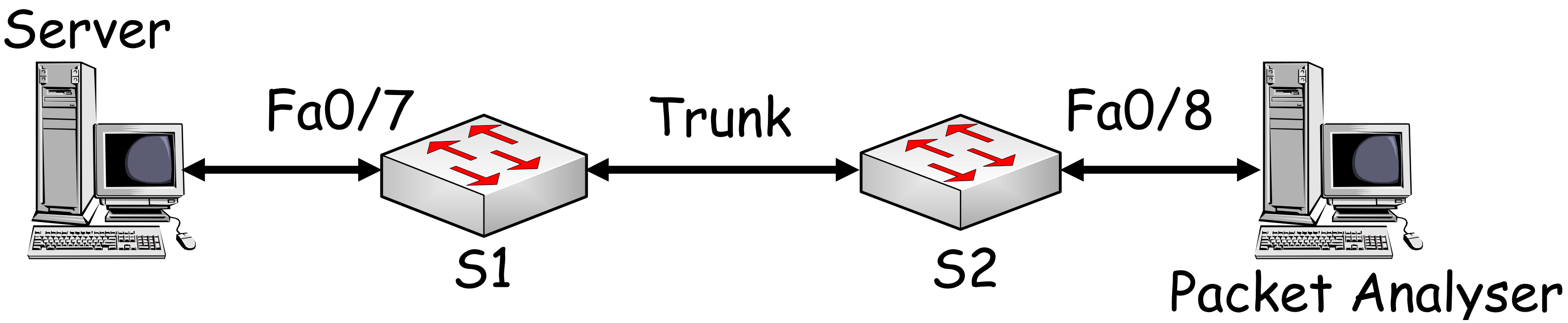
- The Switched Port Analyzer (SPAN) feature of Cisco Catalyst switches allows copying the traffic from one or more switch interfaces or VLANs to another interface on the same switch:

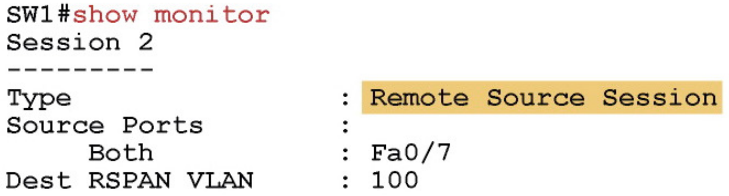
```
S1 (config) #monitor session 1 source int fa0/7
S1 (config) #monitor session 1 destination int fa0/8
```



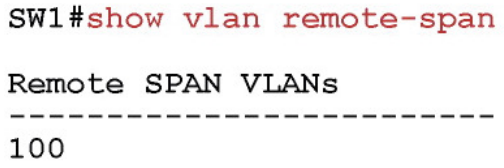
```
SW1#show monitor
Session 1
---------
Type                        : Local Session
Source Ports                :
    Both                    : Fa0/7
Destination Ports           : Fa0/8
    Encapsulation           : Native
          Ingress           : Disabled
```

Traffic Capturing Tools – Protocol Analyser using RSPAN



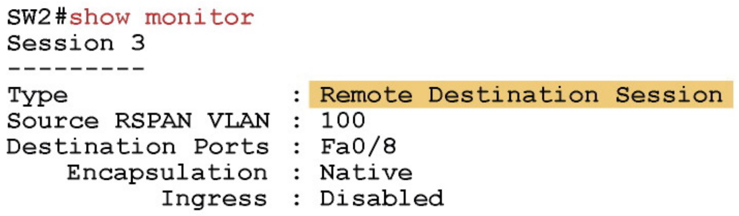


```
SW1#show monitor
Session 2
---------
Type                    : Remote Source Session
Source Ports            :
      Both              : Fa0/7
Dest RSPAN VLAN         : 100
```

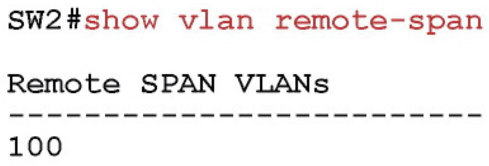


```
SW1#show vlan remote-span

Remote SPAN VLANs
-------------------------
100
```

```
S1(config) #vlan 100
S1(config-vlan) #remote-span
S1(config) #monitor session 2 source int fa0/7
S1(config) #monitor session 2 destination remote vlan 100
```



```
SW2#show monitor
Session 3
---------
Type              : Remote Destination Session
Source RSPAN VLAN : 100
Destination Ports : Fa0/8
    Encapsulation : Native
          Ingress : Disabled
```



```
SW2#show vlan remote-span

Remote SPAN VLANs
-------------------------
100
```

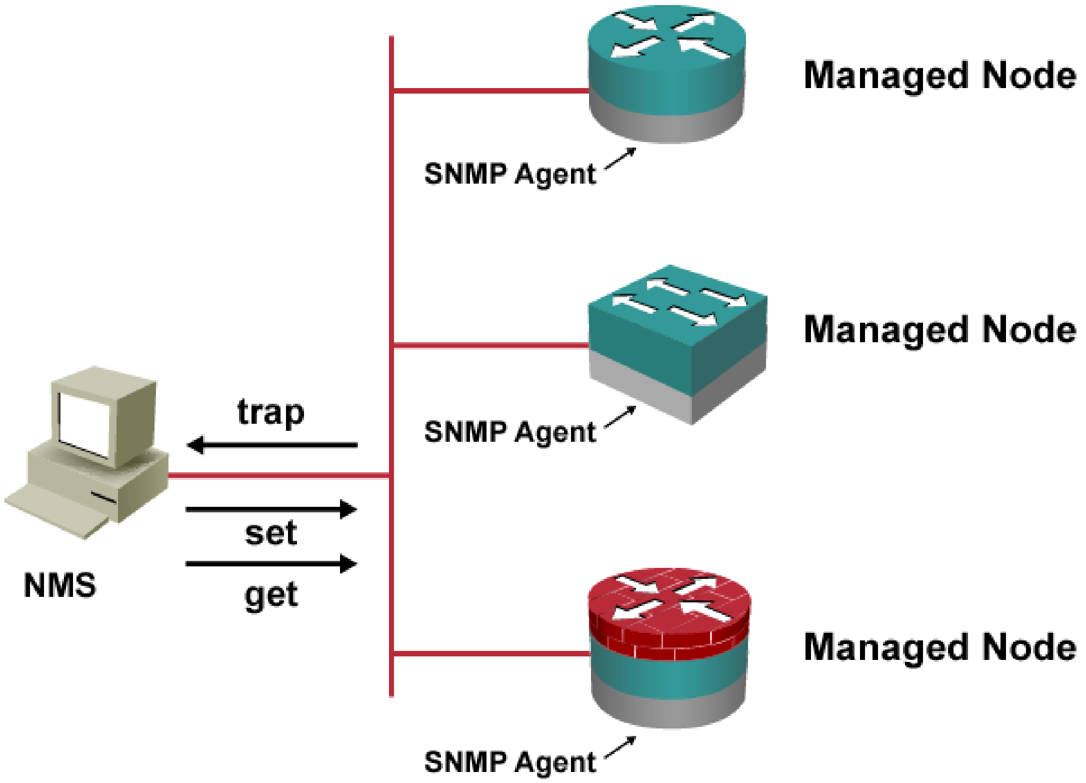
```
S2(config) #vlan 100
S2(config-vlan) #remote-span
S2(config) #monitor session 3 destination int  fa0/8
S2(config) #monitor session 3 source remote vlan 100
```

Traffic Capturing Tools - Simple Network Management Protocol (SNMP)

- SNMP forms part of the internet protocol suite as defined by the IETF.
- SNMP is used by network management systems to monitor network-attached devices for conditions that warrant administrative attention.
- It consists of a set of standards for network management, including an Application Layer protocol, a database schema, and a set of data objects.
- The current version is SNMPv3.
- SNPv1 and v2 are considered obsolete, and are extremely insecure. It is recommended they *NOT* be used on a publicly attached network.

SNMPv1 and SNMPv2 Architecture

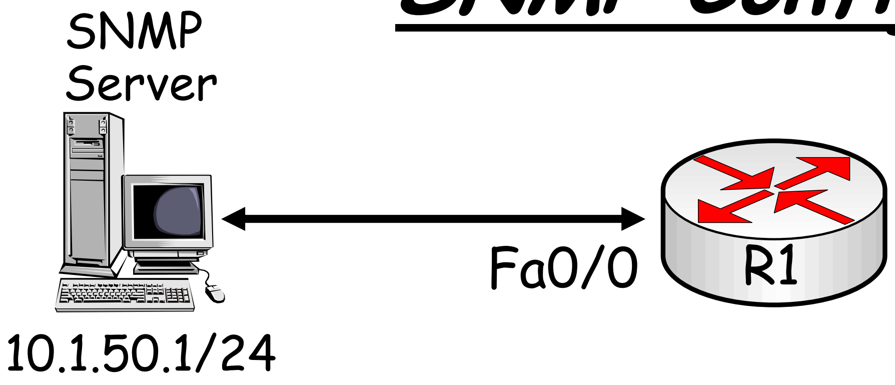
- SNMP asks agents embedded in network devices for information or tells the agents to do something.



SNMP: Security is Not My Problem

- Typically, SNMP uses UDP ports *161* for the agent and *162* for the manager.
- The Manager may send Requests from any available ports (source port) to port 161 in the agent (destination port).
- The agent response will be given back to the source port. The Manager will receive traps on port 162.

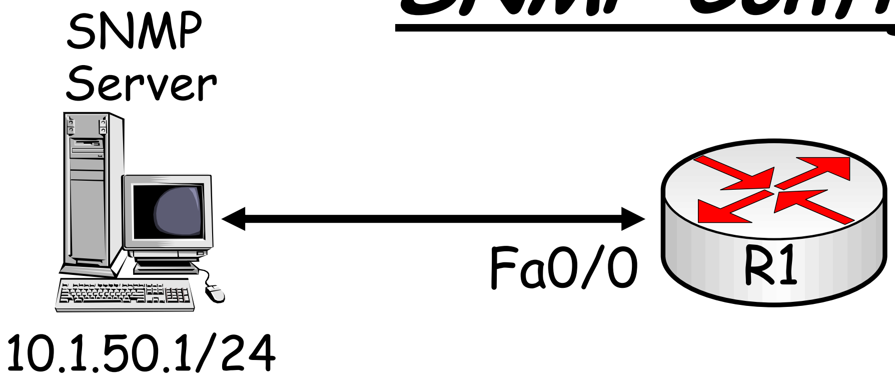
SNMP Configuration



```
R1(config) #snmp-server community cisco ro
R1(config) #snmp-server community san-fran rw
R1(config) #snmp-server location TSHOOT Lab Facility
R1(config) #snmp-server contact support@mgmt.tshoot.local
R1(config) #snmp-server ifindex persist
```

- The **snmp-server ifindex persist** guarantees that the SNMP interface index for each interface will stay the same, even if the device is rebooted.

SNMP Configuration



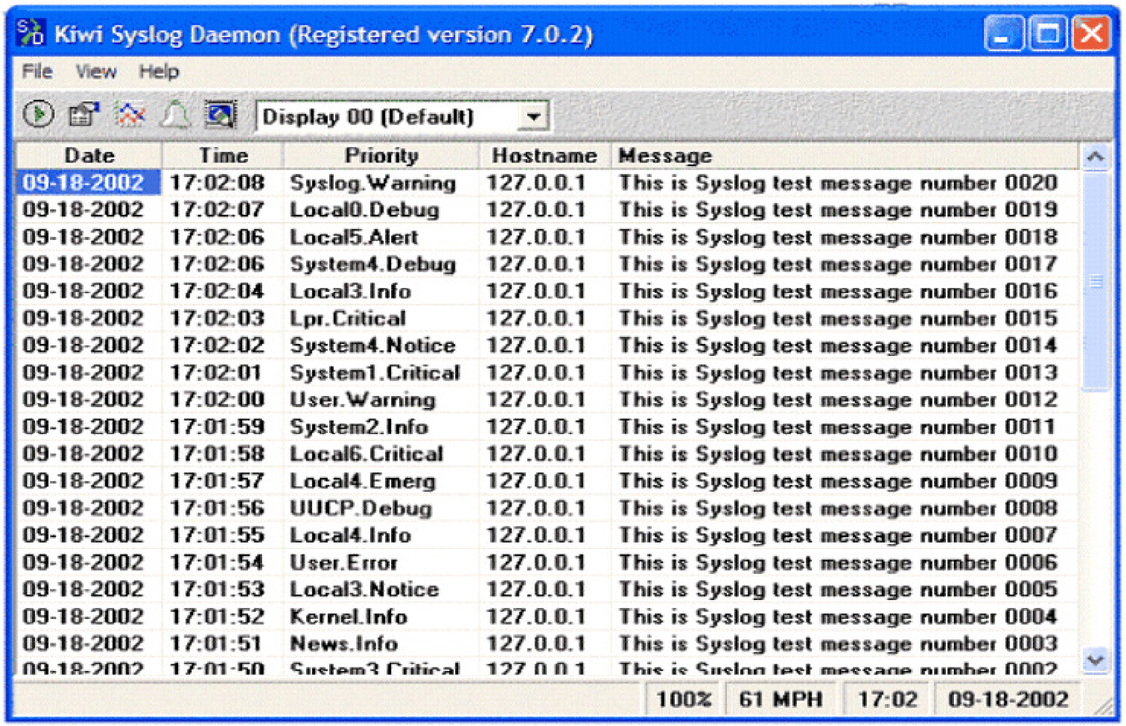
```
R1(config) #snmp-server host 10.1.50.1 version 2c cisco

R1(config) #snmp-server enable traps
```

```
R1 # sh run | include traps
snmp-server enable traps vrrp
snmp-server enable traps dsl
snmp-server enable traps tty
snmp-server enable traps eigrp
snmp-server enable traps envmon
<output omitted>
```

Log Monitoring Tools – Syslog

- Syslog is a standard for forwarding *log* messages in an IP network.

- Syslog messages may be sent via UDP or the TCP. The data is sent in clear text



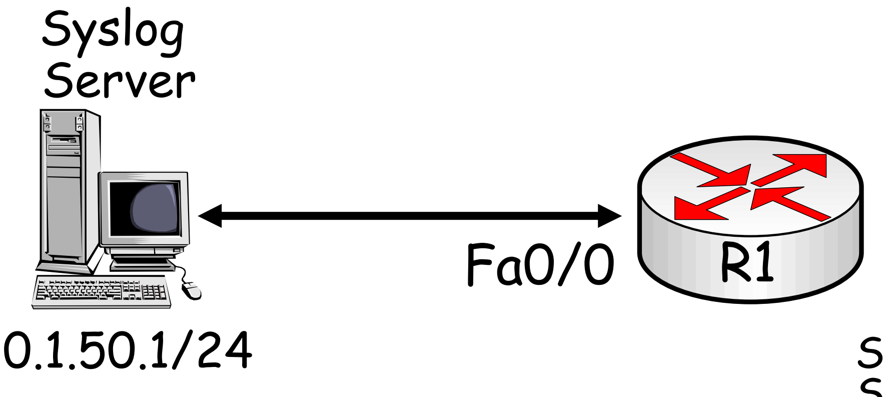
```
Date        Time      Priority          Hostname   Message
09-18-2002  17:02:08  Syslog.Warning    127.0.0.1  This is Syslog test message number 0020
09-18-2002  17:02:07  Local0.Debug      127.0.0.1  This is Syslog test message number 0019
09-18-2002  17:02:06  Local5.Alert      127.0.0.1  This is Syslog test message number 0018
09-18-2002  17:02:06  System4.Debug     127.0.0.1  This is Syslog test message number 0017
09-18-2002  17:02:04  Local3.Info       127.0.0.1  This is Syslog test message number 0016
09-18-2002  17:02:03  Lpr.Critical      127.0.0.1  This is Syslog test message number 0015
09-18-2002  17:02:02  System4.Notice    127.0.0.1  This is Syslog test message number 0014
09-18-2002  17:02:01  System1.Critical  127.0.0.1  This is Syslog test message number 0013
09-18-2002  17:02:00  User.Warning      127.0.0.1  This is Syslog test message number 0012
09-18-2002  17:01:59  System2.Info      127.0.0.1  This is Syslog test message number 0011
09-18-2002  17:01:58  Local6.Critical   127.0.0.1  This is Syslog test message number 0010
09-18-2002  17:01:57  Local4.Emerg      127.0.0.1  This is Syslog test message number 0009
09-18-2002  17:01:56  UUCP.Debug        127.0.0.1  This is Syslog test message number 0008
09-18-2002  17:01:55  Local4.Info       127.0.0.1  This is Syslog test message number 0007
09-18-2002  17:01:54  User.Error        127.0.0.1  This is Syslog test message number 0006
09-18-2002  17:01:53  Local3.Notice     127.0.0.1  This is Syslog test message number 0005
09-18-2002  17:01:52  Kernel.Info       127.0.0.1  This is Syslog test message number 0004
09-18-2002  17:01:51  News.Info         127.0.0.1  This is Syslog test message number 0003
09-18-2002  17:01:50  System3.Critical  127.0.0.1  This is Syslog test message number 0002
```

```
100%   61 MPH   17:02   09-18-2002
```

- Syslog is usually not native to Windows-based systems, but syslog software is available for Windows and Macintosh platforms.

- Syslog software is available via commercial software packages or freeware.

Syslog Configuration



Severity 0 = system unusable
Severity 1 = alerts, immediate action needed
Severity 2 = critical conditions
Severity 3 = error conditions
Severity 4 = warnings
Severity 5 = notifications
Severity 6 = informational messages
Severity 7 = debugging

```
R1 (config)#logging on
R1 (config)#logging host 10.0.50.1
R1 (config)#logging trap <severity>
```

```
R1 (config)#service timestamps log uptime / datetime
R1 (config)#service sequence-numbers
R1 #clock set <hh:mm:ss day month year>
```

Monitor Logging:

```
R1 (config)#show logging
R1 (config)#clear logging
```

Traffic Capturing Tools - Netflow

- A NetFlow-enabled device, such as a router or Layer 3 switch, will collect information about the IP traffic that is flowing through the device, classifying it by flow.

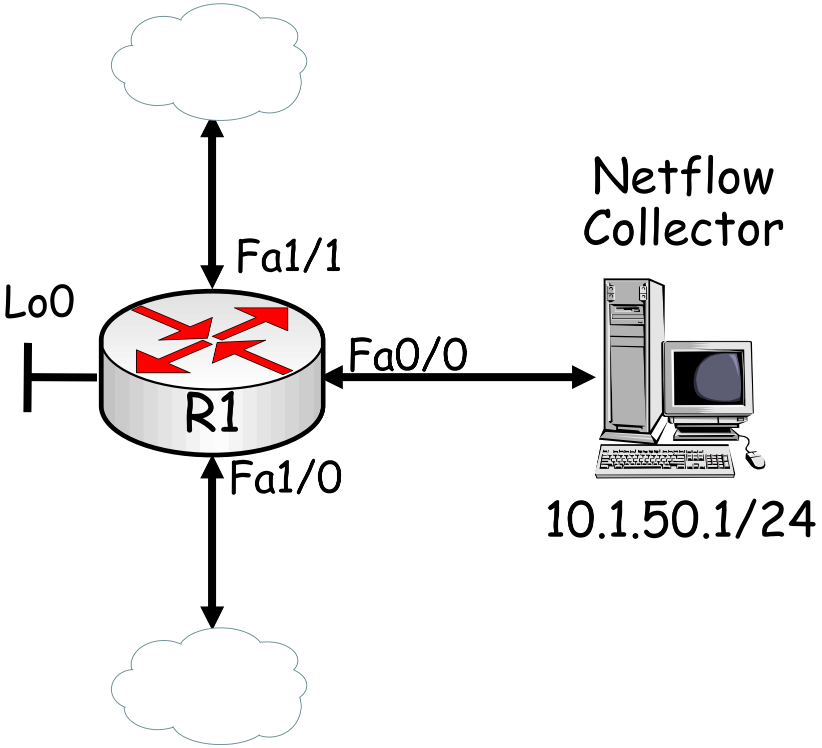
- For each individual flow, the number of packets and bytes is tracked and accounted. This information is kept in a *flow cache*.

- Flows are expired from the cache when the flows are terminated or time out.

- Netflow cache can be configured as a *standalone* feature on router interfaces and examined using CLI commands.

- In addition to keeping a local cache and temporary accounting of the flows on the device itself, the flow information can be *exported* to a NetFlow *collector*.

Netflow Configuration



- The address used as a *source* needs to match the IP address defined on the collector for the router.

- The Netflow *version* and udp *port* number need on the router to match the version and port number on the collector.

```
R1(config)#int fa1/0
R1(config-if)#ip flow ingress
R1(config-if)# int fa1/1
R1(config-if)# ip flow ingress

R1(config) # ip flow-export source lo0
R1(config) #ip flow-export version 5
R1(config) #ip flow export destination 10.1.50.1 9996
```

Netflow Monitoring

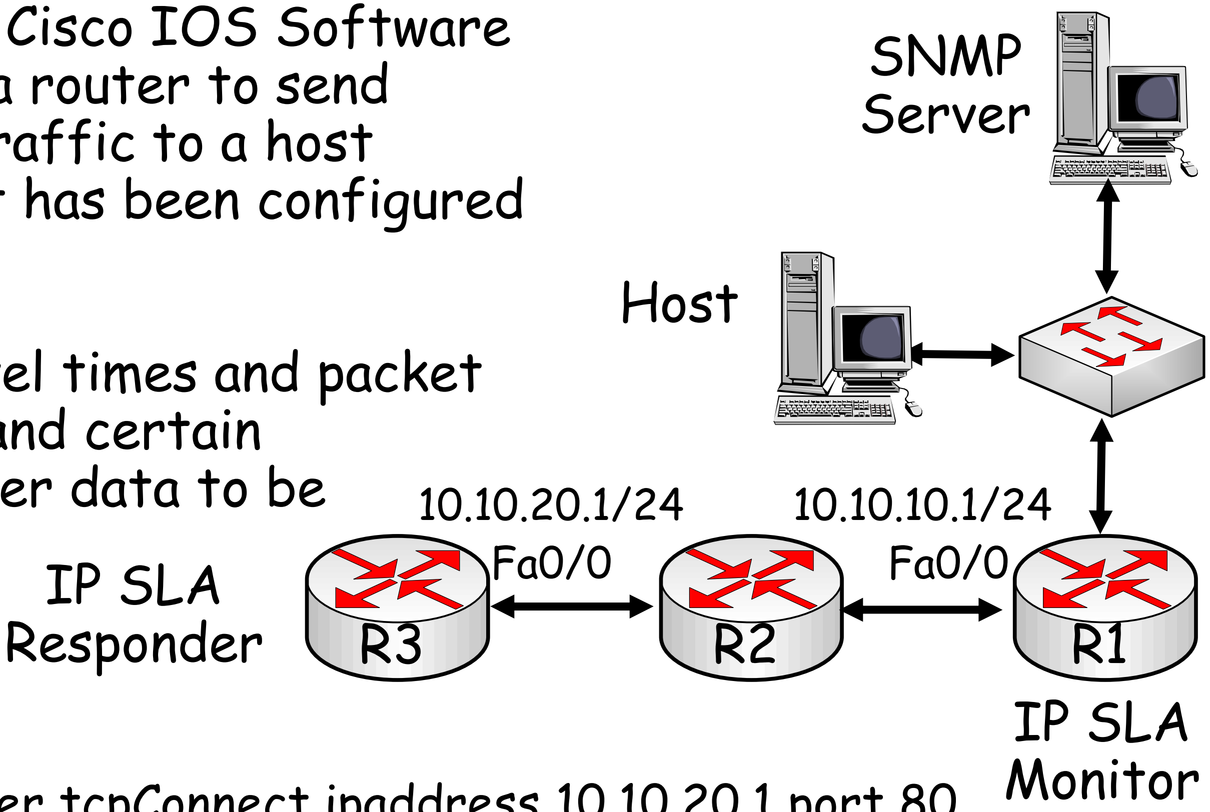
```
R1# show ip cache flow
SrcIf         SrcIPaddress    DstIF    DstIPaddress    Pr    SrcP    DstP    Pkts
Se0/0/0.121   10.1.194.10      Null    224.0.0.10      58    0000     0000   27
Se0/0/0.121   10.1.194.14     Null     224.0.0.10      58    0000     0000   28
Fa0/0         10.1.192.5      Null     224.0.0.10      58     0000   0000    28
Fa0/1         10.1.192.13     Null     224.0.0.10      58    0000     0000   27
Fa0/1          10.1.152.1     Local    10.1.220.2      01    0000    0303    1
Se0/0/1       10.1.193.6      Null     224.0.0.10      58    0000    0000    28
Fa0/1         10.1.152.1      Se0/0/1 10.1.163.193     11    0666    E75E    1906
Se0/0/1        10.1.163.193   Fa0/0    10.1.152.1      11    E75E    0666    1905
```

- The command **show ip cache flow | include 10.1.163.193** could have been used to limit the output to only those flows that have 10.1.163.193 as the source or destination IP address.

Cisco IP Service Level Agreement (SLA) Responder

- The IP SLA feature of Cisco IOS Software allows you to configure a router to send synthetic (generated) traffic to a host computer or router that has been configured to respond.

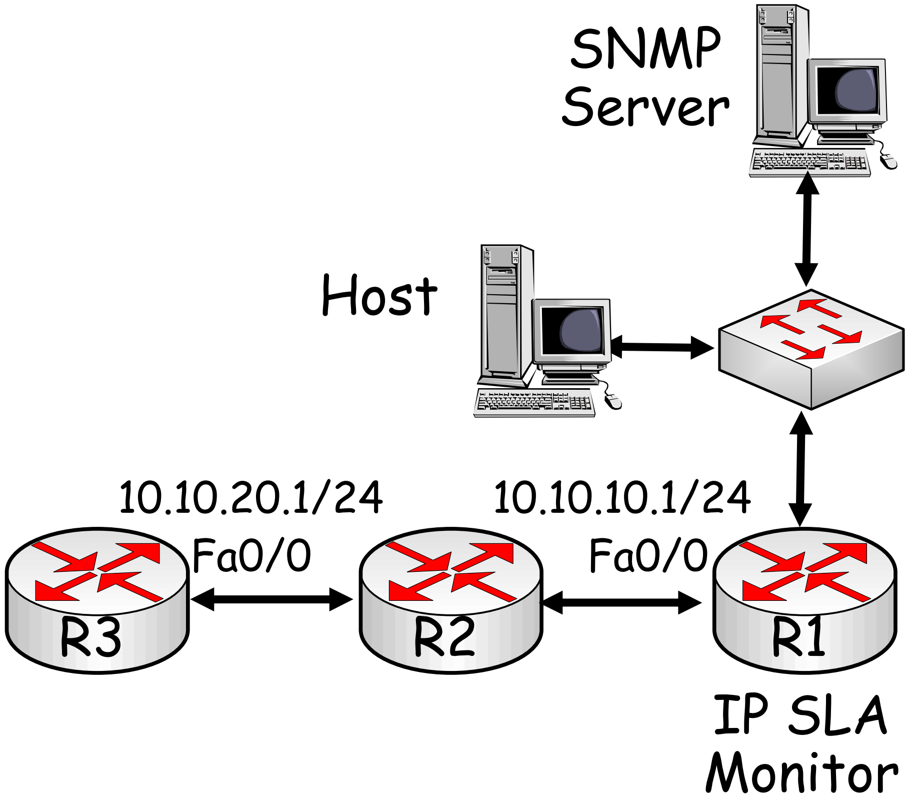
- One-way or return travel times and packet loss data are gathered and certain measurements allow jitter data to be collected as well.



```
R3(config)# ip sla responder tcpConnect ipaddress 10.10.20.1 port 80

R1(config)#ip sla monitor 1
R1(config-rtr)# type  tcpConnect 10.10.20.1
R1(config-rtr-tcp)# frequency 120
R1(config-rtr-monitor-tcp)# tos 64
R1(config)# ip sla monitor schedule 1 start-time now life forever
```

Cisco IP Service Level Agreement (SLA) Responder

- IP SLA monitor supports the ICMP echo feature, without the need to configure the target router as a responder.
- Allows IP SLA to operate with none-Cisco devices.



```
R1(config)# ip sla monitor 2
R1(config-rtr)# type echo protocol ipIcmpEcho 10.10.20.1
R1(config-rtr-echo)# frequency 120
R1(config-rtr-monitor-echo)# tos 32
R1(config)# ip sla monitor schedule 2 start-time now life forever
```

Verify IP SLA

```
R1# show ip sla monitor statistics
Round trip time (RTT) Index 1
Latest RTT: 168 ms
Latest operation start time: *16:10:52.453 UTC Sun Mar 3 2010
Latest operation return code: OK
Number of successes: 13
Number of failures: 1
Operation time to live: Forever

R3# show ip sla responder
IP SLA Monitor Responder is: Enabled
Number of control message received: 15 Number of errors: 1
Recent sources:
10.10.101 [00:38:01.807 UTC Fri Mar 3 2010]
10.10.10.1[00:37:01.783 UTC Fri Mar 3 2010]
...OUTPUT OMITTED...
tcpConnect Responder:
IP Address      Port
10.10.20.1      80
```

Network-Based Application Recognition (NBAR)

- Used in conjunction with QoS class-based features, NBAR is an intelligent classification engine that:
  - *Classifies* modern client-server and web-based applications.
  - *Discovers* what traffic is running on the network.
  - *Analyzes* application traffic patterns in real time.
- NBAR functions:
  - Performs identification of *applications* and *protocols* (Layer 4–7).
  - Performs protocol discovery.
  - Provides traffic *statistics.*
- New applications are easily supported by loading a Packet Description Language Module (PDLM).

Configuring and Monitoring NBAR Protocol Discovery

R1(config-if)#ip nbar protocol-discovery

- Configures NBAR to discover traffic for all protocols known to NBAR on a particular interface
- Requires that CEF be enabled before protocol discovery
- Can be applied with or without a service policy enabled

```
R1#show ip nbar protocol-discovery

 Ethernet0/0
               Input                     Output
   Protocol    Packet Count              Packet Count
               Byte Count                Byte Count
               5 minute bit rate (bps)   5 minute bit rate (bps)
   ----------  ------------------------  ------------------------
   realaudio   2911                      3040
               1678304                   198406
               19000                     1000
   http        19624                     13506
               14050949                  2017293
               0                         0
<output omitted>
```

Chapter 3 – Using Maintenance & Troubleshooting Tools & Applications Objectives

- Describe & utilise Cisco IOS diagnostic tools.
- Explain the need for specialist tools in the troubleshooting process.
- Configure software to allow packet captures.
- Explain how to create a network baseline using Netflow, SNMP, IP SLA & NBAR.

Any Questions?