

Lab 3.1 Preparing for QoS

Learning Objectives

- Create complete configurations to be used with later Quality of Service labs
- Use Pagent tools to create traffic flows for test purposes
- Load and store Pagent configurations
- View statistics on traffic flows during network tests

Topology Diagram

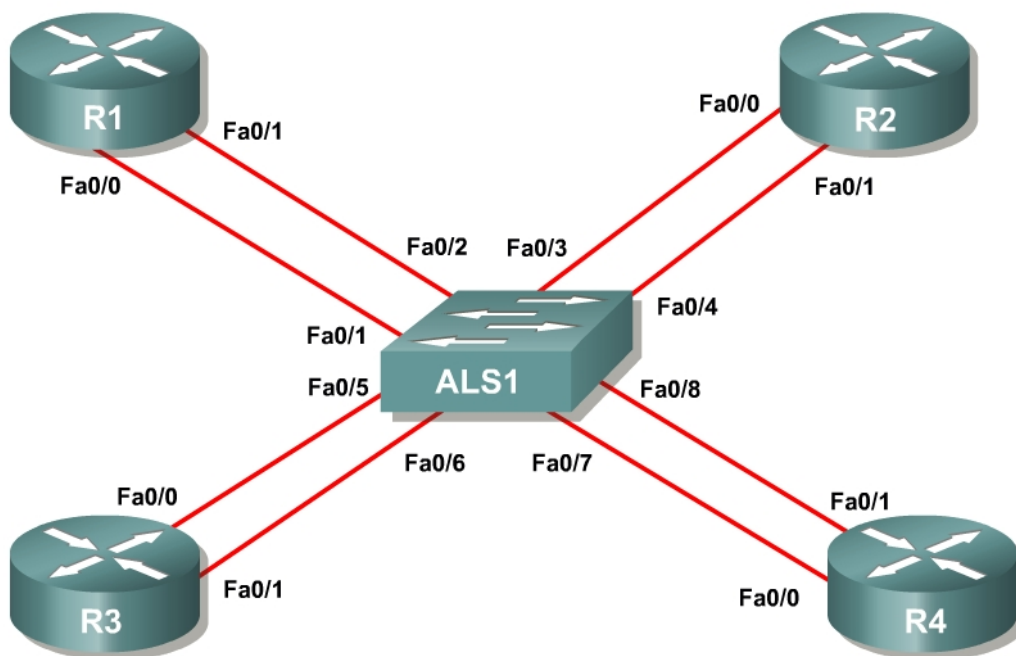


Figure 1-1: Ethernet Connectivity Diagram

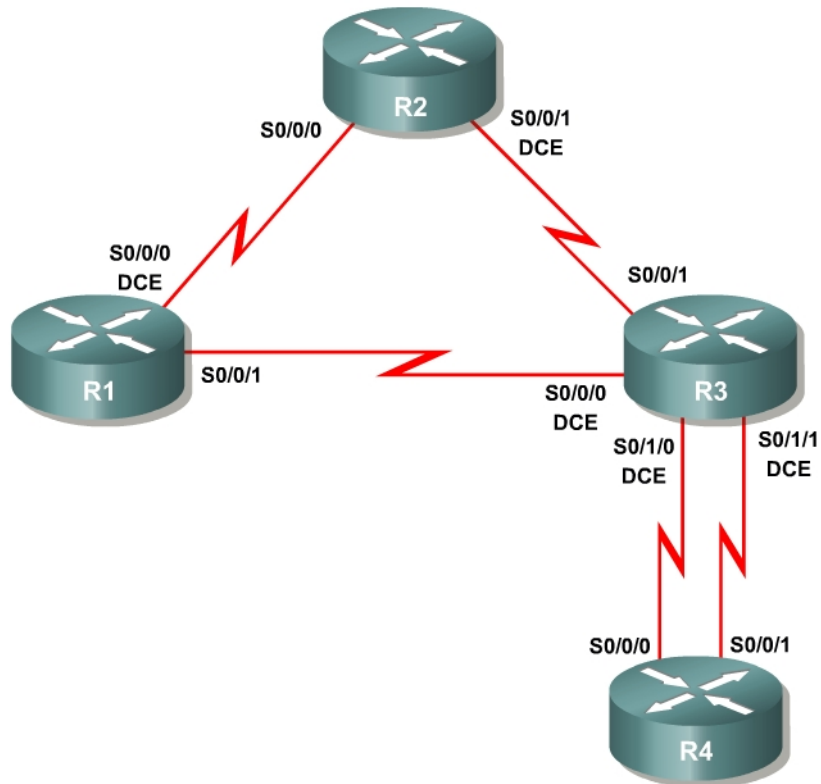


Figure 1-2: Serial Connectivity Diagram

Overview

The Quality of Service (QoS) labs for Modules 3, 4, and 5 have been designed to rely on traffic generation and measuring tools for testing purposes. Traffic generation will be used to create streams of traffic that will flow through your network unidirectionally.

The authors **highly recommend** that you use the Cisco Pagent image and toolset for the QoS labs in the QoS modules. Pagent is a set of traffic generation and testing tools that runs on top of a Cisco IOS image. Booting a router with Pagent can be done by acquiring the image through the Cisco Networking Academy program, loading it into the router's flash memory, and entering a license key when prompted during system boot.

When using the lab configuration suggested in the "CCNP: Optimizing Converged Networks Lab Configuration Guide," you should load the Pagent image on R4.

Key point: Each router booted with Pagent requires a machine-specific license key. It is important to have the license key for R4 before beginning this lab.

This lab guides you through creating configurations for the QoS labs and includes two different configurations.

You will employ the Basic Pagent Configuration in labs that demonstrate each QoS tool separately through two or three routers. You will use the Advanced Pagent Configuration in labs that integrate QoS components across four routers, with R4 acting as both the traffic generator and as a router. The interfaces involved in traffic generation will be isolated from normal routing to ensure that you can use R4 in both roles.

For purposes of this lab, it is assumed that you already have obtained, installed, and activated a Pagent IOS image with a license key on the TrafGen/R4 router.

Finally, labs in these modules may be completed without using any traffic generation. The same configuration steps in each lab will be followed. However, without packet generation tools, you will not see real-time command output.

Step 1: Preliminaries

Erase the startup configurations on any routers involved in this lab. You may need to reactivate Pagent because the activation key is stored in the running configuration of the router.

Traffic generated from TGN, the traffic generation component of Pagent, requires almost all header fields to be hardcoded. Since the packets will be generated over Ethernet, you need to set the destination MAC address of the packets so that they are not broadcast. Remember that this is only the destination for the first hop, not the final destination MAC address. Use the **show interfaces** command to discover the following values.

Example:

```
R1# show interfaces fastethernet0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is MV96340 Ethernet, address is 0019.0623.4380 (bia 0019.0623.4380)
<OUTPUT OMITTED>
```

Record the following value since you will need it at various points throughout this lab:

R1 FastEthernet 0/0, MAC Address: _____

Step 2: Create Basic Pagent IOS and TGN Configurations

This step guides you through creating the Basic Pagent Configuration. In this lab, traffic will flow solely through R1, which will function as the entire network “cloud.” That is, generated traffic will go through R1 and directly back to TrafGen. In the actual QoS labs, the generated traffic will go to the first hop router, traverse the network topology, and then end back at the TrafGen router (or another destination) as shown in the following diagram:

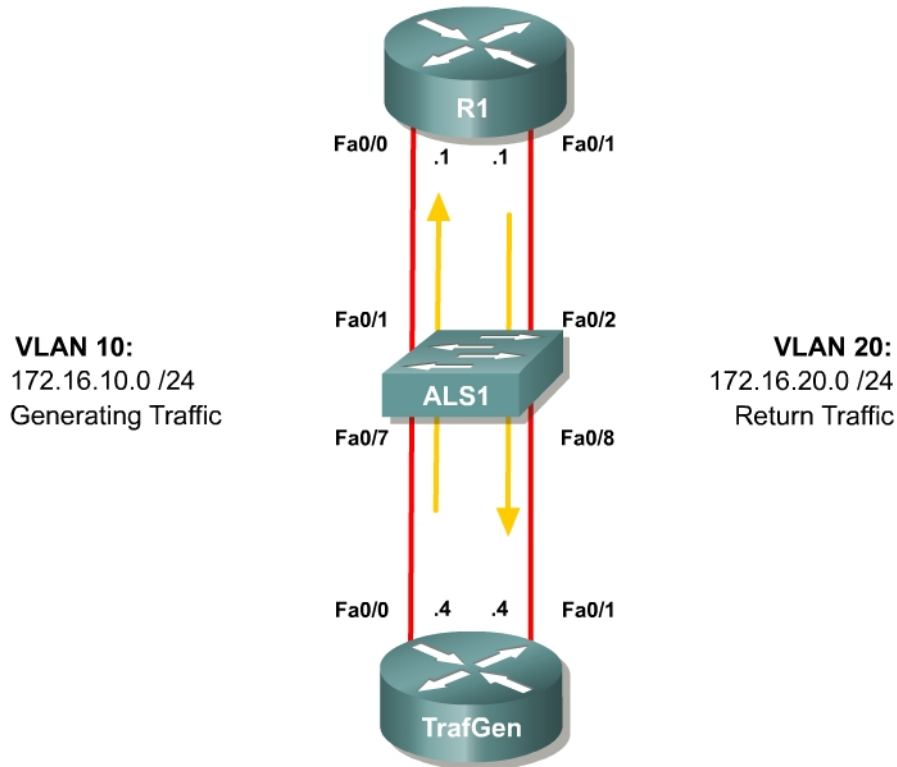


Figure 2-1: Basic Pagent Configuration

- VLAN 10 will be used to send traffic from TrafGen to R1.
- VLAN 20 will be used for traffic returning to the TrafGen router after passing through the last router in the network topology.

You need to assign switchports into the VLANs shown in the diagram.

In order to test connectivity in this scenario, configure TrafGen to send traffic to R1 and then directly back to TrafGen.

Configure the switch to provide Ethernet connectivity for VLANs 10 and 20 as shown in the diagram. Do not configure the FastEthernet 0/2 interface on the switch yet.

```

ALS1# configure terminal
ALS1(config)# interface fastethernet0/1
ALS1(config-if)# switchport access vlan 10
ALS1(config-if)# switchport mode access
ALS1(config-if)# interface fastethernet0/7
ALS1(config-if)# switchport access vlan 10
ALS1(config-if)# switchport mode access
ALS1(config-if)# interface fastethernet0/8
ALS1(config-if)# switchport access vlan 20
ALS1(config-if)# switchport mode access

```

This configuration will be used to begin labs that use the Basic Pageant Configuration. Since the network topology's exit point will change from lab to lab, only TrafGen's FastEthernet 0/1 interface will be placed in VLAN 20 for your template to load at the beginning of each lab that uses the Basic Pageant Configuration. Save this configuration on the switch to a file in flash memory named *flash:basic.cfg*.

```
ALS1# copy run flash:basic.cfg
Destination filename [basic.cfg]?

1391 bytes copied in 0.730 secs (1905 bytes/sec)
ALS1#
```

For this lab only, R1's FastEthernet 0/1 will be the exit point for the network topology while traffic is forwarded back to TrafGen. Therefore add the FastEthernet 0/2 interface on the switch to access VLAN 20.

```
ALS1(config)# interface fastethernet 0/2
ALS1(config-if)# switchport access vlan 20
ALS1(config-if)# switchport mode access
```

At this point, your switch configuration should be complete.

Put TrafGen into configuration mode.

```
Router> enable
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

Copy and paste the following configuration into TrafGen. Adjust the **interface** statements for your lab setup if necessary. You will use the same configuration to begin every QoS lab that uses the Basic Pageant Configuration.

```
hostname TrafGen
!
! Replace this interface with the outgoing interface for generated traffic
interface fastethernet0/0
ip address 172.16.10.4 255.255.255.0
no shutdown
!
! Replace this interface with the incoming interface for generated traffic
! (return traffic)
interface fastethernet0/1
ip address 172.16.20.4 255.255.255.0
no shutdown
```

Copy and paste the following configuration into R1. This configuration is for this guide only. Normally non-Pageant routers should be configured according to the lab. Replace the interface names as necessary if the physical topology of your lab is different.

```
hostname R1
interface fastethernet0/0
ip address 172.16.10.1 255.255.255.0
no shutdown
```

```
interface fastethernet0/1
ip address 172.16.20.1 255.255.255.0
no shutdown
```

TGN is the bulk packet generator tool of Pagent. On the TrafGen router, enter the TGN configuration prompt by using the privileged EXEC command **tgn**.

```
TrafGen# tgn
TrafGen(TGN:OFF, Fa0/0:none) #
```

Copy and paste the following configuration to a text editor. Replace \$R1-MAC\$ in the highlighted line in the configuration below with R1's MAC address from Step 1. If you are using a different source interface for generated traffic, replace all instances of "fastethernet0/0" with the appropriate port. If you are using an outbound serial interface, you do not need to specify an l2-dest and should remove the highlighted line entirely. To exit the TGN prompt, use the **end** command.

```
fastethernet0/0
add tcp
rate 1000
l2-dest $R1-MAC$
l3-src 172.16.10.4
l3-dest 172.16.20.4
l4-dest 23
length random 16 to 1500
burst on
burst duration off 1000 to 2000
burst duration on 1000 to 3000
add fastethernet0/0 1
l4-dest 80
data ascii 0 GET /index.html HTTP/1.1
add fastethernet0/0 1
l4-dest 21
add fastethernet0/0 1
l4-dest 123
add fastethernet0/0 1
l4-dest 110
add fastethernet0/0 1
l4-dest 25
add fastethernet0/0 1
l4-dest 22
add fastethernet0/0 1
l4-dest 6000
!
end
```

Now that you have configured TGN, starting and stopping traffic generation in a lab is very simple. To start traffic generation, use the privileged EXEC command **tgn start**. To stop traffic generation, use the privileged EXEC command **tgn stop**. Or, enter the TGN prompt using the privileged exec command **tgn**, and then use the **start** and **stop** commands. Either method is acceptable, since both perform the same task.

```
TrafGen# tgn start
TrafGen# tgn stop
TrafGen# tgn
```

```
TrafGen(TGN:OFF,Fa0/0:8/8)# start
TrafGen(TGN:ON,Fa0/0:8/8)# stop
TrafGen(TGN:OFF,Fa0/0:8/8)# end
TrafGen#
```

On R1, use the **show interfaces** command for both the inbound and outbound interfaces to make sure that packets are being generated correctly and routed appropriately. This test should be done while traffic generation is on. For the inbound interface (receiving newly generated packets), make sure the inbound packet counters are incrementing. For the outbound interface (routing the generated packets back to TrafGen), make sure the outbound packet counters are incrementing.

```
TrafGen# tgn start
```

```
R1# show interfaces fastethernet 0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is MV96340 Ethernet, address is 0019.0623.4380 (bia 0019.0623.4380)
  Internet address is 172.16.10.1/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 2/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:16, output 00:00:01, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 874000 bits/sec, 139 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    46701 packets input, 36522488 bytes
<OUTPUT OMITTED>
```

```
R1# show interfaces fastethernet 0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is MV96340 Ethernet, address is 0019.0623.4380 (bia 0019.0623.4380)
  Internet address is 172.16.10.1/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 2/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:26, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 952000 bits/sec, 152 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    55017 packets input, 43066713 bytes
<OUTPUT OMITTED>
```

```
R1# show interfaces fastethernet 0/1
FastEthernet0/1 is up, line protocol is up
  Hardware is MV96340 Ethernet, address is 0019.0623.4381 (bia 0019.0623.4381)
  Internet address is 172.16.20.1/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
```

```

    reliability 255/255, txload 4/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Full-duplex, 100Mb/s, 100BaseTX/FX
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:00:19, output 00:00:00, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 1666000 bits/sec, 270 packets/sec
    48 packets input, 17808 bytes
    Received 47 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 watchdog
    0 input packets with dribble condition detected
    97245 packets output, 75956525 bytes, 0 underruns
<OUTPUT OMITTED>

R1# show interfaces fastethernet 0/1
FastEthernet0/1 is up, line protocol is up
  Hardware is MV96340 Ethernet, address is 0019.0623.4381 (bia 0019.0623.4381)
  Internet address is 172.16.20.1/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
    reliability 255/255, txload 4/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:29, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 1794000 bits/sec, 292 packets/sec
    48 packets input, 17808 bytes
    Received 47 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 watchdog
    0 input packets with dribble condition detected
    106314 packets output, 82995904 bytes, 0 underruns
<OUTPUT OMITTED>

```

Step 3: Store Basic Pagent Configurations

First, store the Basic Pagent Configuration in flash memory with a filename of *basic-ios.cfg* using the **copy running-config flash:basic-ios.cfg** command. When you require the Basic Pagent Configuration, your first step should be to replace the configuration in NVRAM with this file. Then you would reload your router and load the Pagent configurations.

Caution: Make sure you do not erase the flash file system when you replace the configuration. If you do, you will have to stop the lab and install a Pagent IOS image on the router before continuing.

```

TrafGen# copy running-config flash:basic-ios.cfg
Destination filename [basic-ios.cfg]?

```

```
Erase flash: before copying? [confirm] n
Verifying checksum... OK (0x3FD3)
2875 bytes copied in 0.600 secs (4792 bytes/sec)
```

As you may have guessed, TGN configurations are stored separately from the running configuration, so they are not saved to the router when you type **copy run start** or **write memory** to save the running configuration to the NVRAM of the router. To save a TGN configuration, use the TGN command **save-config location**. To load a TGN configuration from a file, use the TGN command **load-config location**. The following example shows the TGN configuration being saved to a file on the flash named *basic-tgn.cfg*, and shows it being loaded back in. Use this filename if you want to be able to load the configuration from the menu in the previous step.

```
TrafGen# tgn
TrafGen(TGN:OFF,Fa0/0:8/8)# save-config flash:basic-tgn.cfg
Save complete.
TrafGen(TGN:OFF,Fa0/0:8/8)# load-config flash:basic-tgn.cfg
Please wait until 'Load Complete' message.

TrafGen(TGN:OFF,Fa0/0:none)#
Load Complete.
TrafGen(TGN:OFF,Fa0/0:8/8)#
```

Clear the current TGN configuration before you proceed to the next step. Use the TGN command **clear config**, as shown in the following output.

```
TrafGen(TGN:OFF,Fa0/0:8/8)# clear config
TrafGen(TGN:OFF,Fa0/0:none)#
```

Along with the ALS1's *basic.cfg* file, the configurations saved in this step will be loaded initially at the beginning of each of the labs which use the Basic Pagent Configuration.

Step 4: Create Advanced Pagent IOS, TGN, and NQR Configurations

Keep in mind that the Basic Pagent Configuration will be used in the labs that demonstrate individual QoS tools; the Advanced Pagent Configuration will be used in labs that integrate QoS topics across a larger topology. You will use R4 as both a transit router on which you will configure some QoS tools and as the Pagent host on VLANs 10 and 20 with which you will generate and capture traffic. The interfaces you configure to generate and capture Pagent traffic will be isolated from the default routing table. They will be contained in another routing table, essentially virtualizing the router into two devices. One virtual device will be acting as a host generating traffic on one interface and receiving it back on another after the traffic passes through the network topology. The other virtual router will act as R4 in the topology, associating with the other routers through routing protocols. If you are confused about this concept, discuss it with classmates and study the topology diagram in Figure 5-1 and the conceptual diagram in Figure 5-2. Do not proceed until you understand the concept.

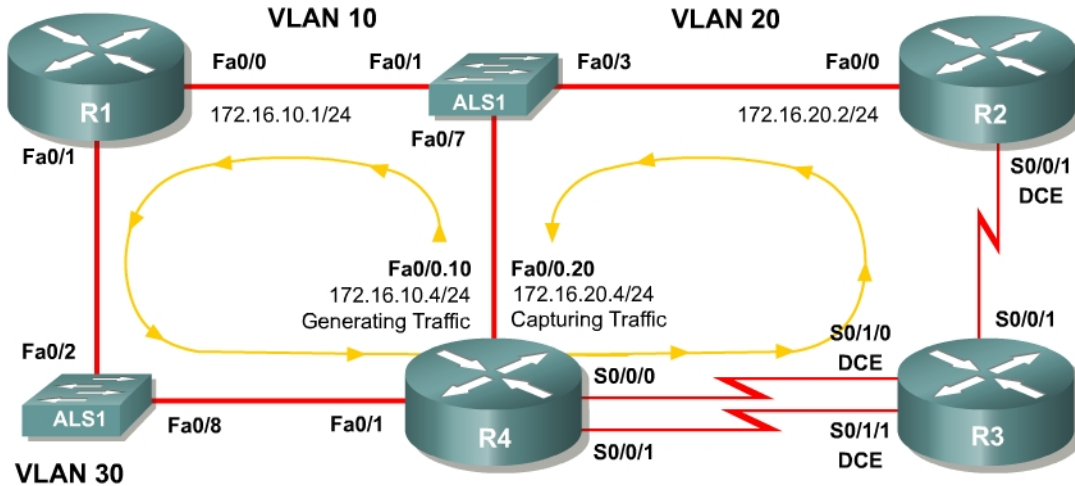


Figure 5-1: Advanced Pagent Configuration

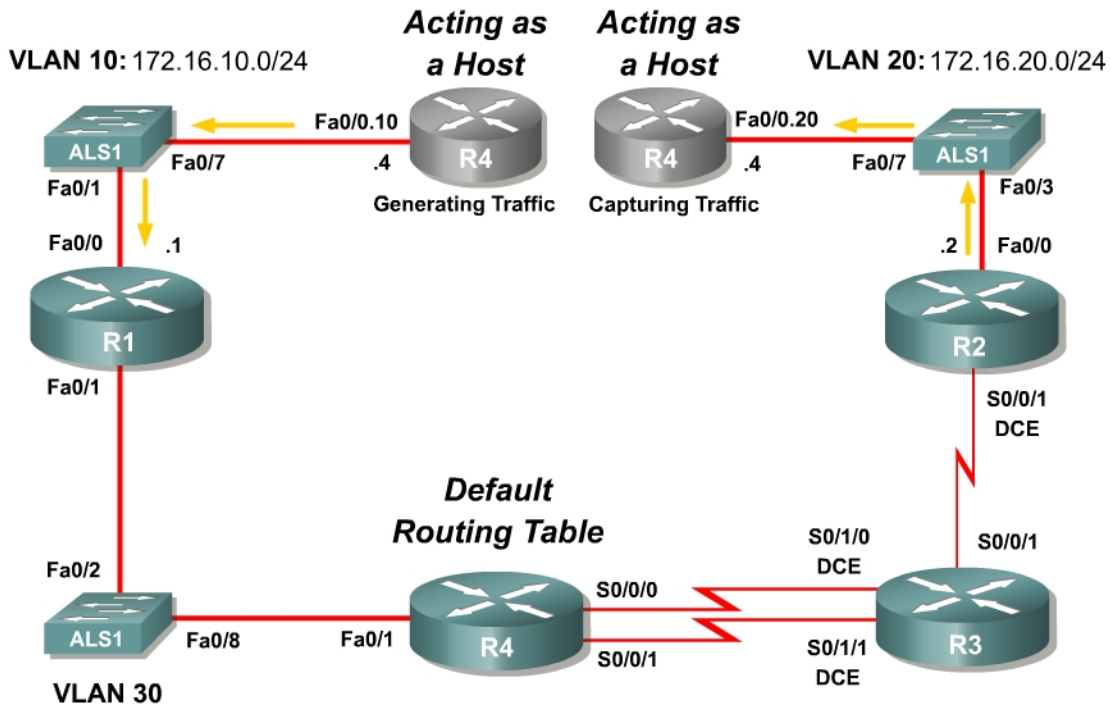


Figure 5-2: Advanced Pagent Configuration, Conceptual Diagram

The recommended configuration uses trunking. If you are using a pod in which you may not manipulate switchports to trunking mode, you may consider using more than one subnet on a single VLAN as shown in Appendix C. Appendix C configurations are NetLab-compatible.

Use the **erase startup-config** command followed by the **reload** command to reset the R4 with a blank configuration. You will need to re-enter the Pagent license key that you first entered in Step 1.

Next, copy and paste the following Advanced Pagent Configuration onto R4 (TrafGen) at the configure prompt. This configuration only includes the commands relevant to Pagent's setup but not those that relate to specific connectivity between R4 and the routers with which it will communicate. This configuration isolates the traffic generation to a separate routing table from the main routing table using virtual routing and forwarding tables, or VRFs. VRFs are outside the scope of this course. To learn more about VRFs, consult cisco.com.

```
hostname R4
!
ip vrf PAGENT
!
interface fastethernet0/0.10
description Interface generating traffic
encapsulation dot1q 10
ip vrf forwarding PAGENT
ip address 172.16.10.4 255.255.255.0
!
interface fastethernet0/0.20
description Interface capturing traffic
encapsulation dot1q 20
ip vrf forwarding PAGENT
ip address 172.16.20.4 255.255.255.0
!
interface fastethernet0/0
no shutdown
```

Configure the switch connected to R4's Fast Ethernet 0/0 port to trunk VLANs 10 and 20 to R4. Also, configure switchports connected to R1 and R2 as access ports and in the VLANs diagrammed above. Finally, place Fast Ethernet interfaces 0/2 and 0/8 on the switch in VLAN 30. Fast Ethernet interfaces 0/2 and 0/8 will be in VLAN 30 for all of the QoS labs that require the Advanced Pagent Configuration.

Copy and paste the following configuration onto the switch in global configuration mode to accomplish these tasks.

```
hostname ALS1
!
vtp mode transparent
vtp domain CISCO
!
vlan 10,20,30
!
interface fastethernet0/1
switchport mode access
switchport access vlan 10
!
interface fastethernet 0/2
switchport mode access
switchport access vlan 30
!
interface fastethernet0/3
switchport mode access
switchport access vlan 20
!
```

```

interface fastethernet0/7
! switchport trunk encapsulation dot1q
! Remove the exclamation point in the previous line
! if the switch supports multiple trunk encapsulations
switchport mode trunk
!
interface fastethernet 0/8
switchport mode access
switchport access vlan 30
!
end

```

On R4, you will now configure TGN. The configuration you will use is almost identical to the basic one, except modified because we are using subinterfaces. You will not need to put in R1's MAC address because the packets are being encapsulated differently. Use the privileged EXEC command **tgn** to get into the TGN prompt.

```

fastethernet0/0
add tcp
rate 1000
datalink ios-dependent fastethernet0/0.10
l2-arp-for 172.16.10.1
l3-src 172.16.10.4
l3-dest 172.16.20.4
l4-dest 23
length random 16 to 1500
burst on
burst duration off 1000 to 2000
burst duration on 1000 to 3000
add fastethernet0/0 1
l4-dest 80
data ascii 0 GET /index.html HTTP/1.1
add fastethernet0/0 1
l4-dest 21
add fastethernet0/0 1
l4-dest 123
add fastethernet0/0 1
l4-dest 110
add fastethernet0/0 1
l4-dest 25
add fastethernet0/0 1
l4-dest 22
add fastethernet0/0 1
l4-dest 6000
!
end

```

Refer to Step 2 to find out how to use TGN.

Step 5: Store Advanced Pagent Configurations

Store the advanced TGN configuration to the file in flash memory named *advanced-tgn.cfg*, and save the advanced IOS configuration to the file in flash named *advanced-ios.cfg*.

```

R4# tgn save-config flash:advanced-tgn.cfg
Save complete.
R4# copy running-config flash:advanced-ios.cfg

```

```
Destination filename [advanced-ios.cfg]?
Erase flash: before copying? [confirm]n
Verifying checksum... OK (0xDCE7)
1103 bytes copied in 1.228 secs (898 bytes/sec)
```

This configuration will be used to begin labs that use the Advanced Pagent Configuration. Save this configuration on the switch to a file in flash memory named *flash:advanced.cfg*.

```
ALS1# copy run flash:advanced.cfg
Destination filename [advanced.cfg]?

1458 bytes copied in 0.730 secs (1997 bytes/sec)
```

Step 6: Display Traffic Statistics

In many labs using the advanced configuration, you can use NQR to gather traffic statistics. NQR is a Pagent tool that allows you to send and then capture packets. It combines TGN (the traffic generation tool you have already been using) and PKTS (a packet capturing tool you have not set up). Configuration of NQR is similar to that of TGN except that you select one interface for generating the packets and another for capturing them. Unlike the TGN configuration for this course, NQR labs may vary from lab to lab so this configuration is just an example, not a template to be used in all labs.

Before you configure NQR, apply the configurations in Appendix D to each of your routers to set up an end-to-end routing topology using Open Shortest Path First (OSPF).

NQR can be run at the same time as TGN. They work together for QoS testing, in that TGN can generate the bulk background traffic but statistics can be run for the more limited NQR traffic. For this part of the lab, shut off TGN so that its traffic will not interfere with the NQR traffic. If you decide to try this part of the lab, you will also have to configure all of the routers the same way as shown in the Figure 5-1, with IP addresses and a routing protocol (including R4). These configurations are shown in Appendix D. Otherwise, just look at the commands below to get an idea of how NQR works.

To get into the NQR configuration prompt, type **nqr** at the privileged EXEC prompt. After you get into the NQR configuration prompt, copy and paste in the following configuration. Please see appendix C for NETLAB compatible version

```
fastethernet0/0
add tcp
datalink ios-dependent fastethernet0/0.10
l2-arp-for 172.16.10.1
l3-src 172.16.10.4
l3-dest 172.16.20.4
l4-dest 23
fastethernet0/0.20 ios-dependent capture
```

This configuration instructs NQR to generate traffic destined towards TCP port 23 (Telnet), similar to the stream configured for TGN. You may notice that an interface was selected for capturing packets.

To start the traffic stream, type **start** (just like TGN). To stop the traffic stream, type **stop**.

Note that in NQR, once traffic generation is stopped, it will keep collecting data and the status will change to "WAIT" before it is done. Once complete, you can view traffic statistics on loss, delay, reordering, jitter, and so forth.

```
R4#nqr
R4 (NQR:OFF, Fa0/0:none) # fastethernet0/0
R4 (NQR:OFF, Fa0/0:none) # add tcp
R4 (NQR:OFF, Fa0/0:1/1) # datalink ios-dependent fastethernet0/0.10
R4 (NQR:OFF, Fa0/0:1/1) # l2-arp-for 172.16.10.1
R4 (NQR:OFF, Fa0/0:1/1) # l3-src 172.16.10.4
R4 (NQR:OFF, Fa0/0:1/1) # l3-dest 172.16.20.4
R4 (NQR:OFF, Fa0/0:1/1) # l4-dest 23
R4 (NQR:OFF, Fa0/0:1/1) # fastethernet0/0.20 ios-dependent capture
R4 (NQR:OFF, Fa0/0:1/1) #
R4 (NQR:OFF, Fa0/0:1/1) # start
R4 (NQR:ON, Fa0/0:1/1) # stop
R4 (NQR:WAIT, Fa0/0:1/1) #
R4 (NQR:OFF, Fa0/0:1/1) #
```

Verify packet drop and reordering statistics using the command **show pkt-seq-drop-stats**. You should have zero dropped packets (as seen in the following output) since there is no other traffic running through the network. If all packets are dropped, you have a problem: either they are not getting routed correctly through the network or something else is taking up the bandwidth (which should not happen since TGN was turned off).

```
R4 (NQR:OFF, Fa0/0:1/1) # show pkt-seq-drop-stats

Summary of packet sequence/drop stats of traffic streams
  ts#  template interface      sent   recvd  dropped  out-of-seq  max-seq
   1   TCP      Fa0/0.10*         31     31      0           0         31
```

You can also look at delay and jitter statistics with the commands **show delay-stats** and **show jitter-stats** respectively.

```
R4 (NQR:OFF, Fa0/0:1/1) # show delay-stats

Summary of delay-stats of traffic streams
  ts#  template interface      min-delay  max-delay  avg-delay  stdev-delay
   1   TCP      Fa0/0.10*  0.009561  0.009771  0.009653  0.000060

R4 (NQR:OFF, Fa0/0:1/1) # show jitter-stats

Summary of jitter-stats of traffic streams
  ts#  template interface      min-jitter  max-jitter  avg-jitter  stdev-jitter
   1   TCP      Fa0/0.10*  0.000001  0.000144  0.000054  0.000036
```

Appendix A: Basic Pagent Configurations

IOS Configuration on R4 — Stored in **flash:basic-ios.cfg**

```
!  
hostname TrafGen  
!  
username cisco password cisco  
username pagent privilege 15 password pagent  
username pagent autocommand menu pagentmenu  
!  
! Replace this interface with the outgoing interface for generated traffic  
interface fastethernet0/0  
  ip address 172.16.10.4 255.255.255.0  
  no shutdown  
!  
! Replace this interface with the incoming interface for generated traffic  
! (return traffic)  
interface fastethernet0/1  
  ip address 172.16.20.4 255.255.255.0  
  no shutdown  
!  
line con 0  
  login local  
!  
end
```

TGN Configuration on R4 — Stored in **flash:basic-tgn.cfg**

```
fastethernet0/0  
add tcp  
rate 1000  
12-dest $R1-MAC$  
13-src 172.16.10.4  
13-dest 172.16.20.4  
14-dest 23  
length random 16 to 1500  
burst on  
burst duration off 1000 to 2000  
burst duration on 1000 to 3000  
add fastethernet0/0 1  
14-dest 80  
data ascii 0 GET /index.html HTTP/1.1  
add fastethernet0/0 1  
14-dest 21  
add fastethernet0/0 1  
14-dest 123  
add fastethernet0/0 1  
14-dest 110  
add fastethernet0/0 1  
14-dest 25  
add fastethernet0/0 1  
14-dest 22  
add fastethernet0/0 1  
14-dest 6000
```

IOS Configuration on ALS1 — Stored in **flash:basic.cfg**. You may have to add additional switchports to VLAN 20 in future labs based on specific lab topologies.

```
!  
hostname ALS1  
!
```

```

interface fastethernet 0/1
  description R1 FastEthernet0/0
  switchport access vlan 10
  switchport mode access
!
interface fastethernet 0/7
  description TrafGen FastEthernet 0/0
  switchport access vlan 10
  switchport mode access
!
interface fastethernet 0/8
  description TrafGen FastEthernet0/1
  switchport access vlan 20
  switchport mode access
!
end

```

Appendix B: Advanced Pagent Configurations

IOS Configuration on R4 (TrafGen) — Stored in **flash:advanced-ios.cfg**. This does not include the sample configuration for NQR (shown in Appendix D)

```

hostname R4
!
username cisco password cisco
username pagent privilege 15 password pagent
username pagent autocommand menu pagentmenu
!
ip vrf PAGENT
!
interface fastethernet0/0
  no shutdown
!
interface fastethernet0/0.10
  encapsulation dot1q 10
  ip vrf forwarding PAGENT
  ip address 172.16.10.4 255.255.255.0
!
interface fastethernet0/0.20
  encapsulation dot1q 20
  ip vrf forwarding PAGENT
  ip address 172.16.20.4 255.255.255.0
!
line con 0
  login local
!
end

```

TGN Configuration on R4 (TrafGen) — Stored in **flash:advanced-tgn.cfg**

```

fastethernet0/0
add tcp
rate 1000
datalink ios-dependent fastethernet0/0.10
l2-arp-for 172.16.10.1
l3-src 172.16.10.4
l3-dest 172.16.20.4
l4-dest 23
length random 16 to 1500
burst on
burst duration off 1000 to 2000

```

```

burst duration on 1000 to 3000
add fastethernet0/0 1
  l4-dest 80
  data ascii 0 GET /index.html HTTP/1.1
add fastethernet0/0 1
  l4-dest 21
add fastethernet0/0 1
  l4-dest 123
add fastethernet0/0 1
  l4-dest 110
add fastethernet0/0 1
  l4-dest 25
add fastethernet0/0 1
  l4-dest 22
add fastethernet0/0 1
  l4-dest 6000

```

IOS Configuration on ALS1 — Stored in **flash:advanced.cfg**

```

hostname ALS1
!
vtp mode transparent
vtp domain CISCO
!
vlan 10,20,30
!
interface fastethernet0/1
  switchport mode access
  switchport access vlan 10
!
interface fastethernet 0/2
  switchport mode access
  switchport access vlan 30
!
interface fastethernet0/3
  switchport mode access
  switchport access vlan 20
!
interface fastethernet0/7
! switchport trunk encapsulation dot1q
! Remove the exclamation point in the previous line
! if the switch supports multiple trunk encapsulations
  switchport mode trunk
!
interface fastethernet 0/8
  switchport mode access
  switchport access vlan 30
!
end

```

Appendix C: NetLab-compatible Advanced Pagent Configurations

IOS Configuration on R4 — Stored in **flash:advanced-ios.cfg**

```

hostname R4
!
username cisco password cisco
username pagent privilege 15 password pagent
username pagent autocommand menu pagentmenu
!

```

```

ip vrf PAGENT
!
!
interface fastethernet0/0
 ip vrf forwarding PAGENT
 ip address 172.16.20.4 255.255.255.0
 ip address 172.16.10.4 255.255.255.0 secondary
 no shutdown
!
!
line con 0
 login local
!
end

```

TGN Configuration on R4 — Stored in **flash:advanced-tgn.cfg**

```

fastethernet0/0
add tcp
rate 1000
12-dest $ R1 Fa0/0's MAC$
13-src 172.16.10.4
13-dest 172.16.20.4
14-dest 23
length random 16 to 1500
burst on
burst duration off 1000 to 2000
burst duration on 1000 to 3000
add fastethernet0/0 1
14-dest 80
data ascii 0 GET /index.html HTTP/1.1
add fastethernet0/0 1
14-dest 21
add fastethernet0/0 1
14-dest 123
add fastethernet0/0 1
14-dest 110
add fastethernet0/0 1
14-dest 25
add fastethernet0/0 1
14-dest 22
add fastethernet0/0 1
14-dest 6000

```

NOTE: NETLAB+ would automatically load the following configuration to the switch for this exercise. Notice that VLAN 10 connects two IP subnets 172.16.10.0 and 172.16.20.0.

IOS Configuration on ALS1 – Stored in **flash:advanced.cfg**

```

!
hostname ALS1
!
vtp mode transparent
vtp domain CISCO
!
vlan 10,30
!
interface FastEthernet0/1

```

```

description Connection to R1 (FastEthernet0/0)
switchport access vlan 10
switchport mode access
!
interface FastEthernet0/2
description Connection to R1 (FastEthernet0/1)
switchport access vlan 30
switchport mode access
!
interface FastEthernet0/3
description Connection to R2 (FastEthernet0/0)
switchport access vlan 10
switchport mode access
!
interface FastEthernet0/7
description Connection to R4 (FastEthernet0/0) - for Pagent Generation
switchport access vlan 10
switchport mode access
!
interface FastEthernet0/8
description Connection to R4 (FastEthernet0/1)
switchport access vlan 30
switchport mode access
!
end

```

NQR Configuration

```

FastEthernet0/0
add tcp
12-dest $R1 Fa0/0's MAC$
13-src 172.16.10.4
13-dest 172.16.20.4
14-dest 23
fastEthernet0/0 capture

```

Appendix D: Sample Advanced Pagent Configuration

Copy and paste these configurations into their respective routers in the configure prompt. This configuration is only for trying out the advanced topology. These configurations may vary from lab to lab. For the configuration that can be used as a template, consult Appendix B. The switch configurations are not shown.

R1:

```

!
hostname R1
!
interface fastEthernet0/0
ip address 172.16.10.1 255.255.255.0
no shutdown
!
interface fastEthernet0/1
ip address 172.16.14.1 255.255.255.0
no shutdown
!
router ospf 1

```

```
network 172.16.0.0 0.0.255.255 area 0
!  
end
```

R2:

```
!  
hostname R2  
!  
interface FastEthernet0/0  
ip address 172.16.20.2 255.255.255.0  
no shutdown  
!  
interface Serial0/0/1  
ip address 172.16.23.2 255.255.255.0  
clockrate 64000  
no shutdown  
!  
router ospf 1  
network 172.16.0.0 0.0.255.255 area 0  
!  
end
```

R3:

```
!  
hostname R3  
!  
interface Serial0/0/1  
ip address 172.16.23.3 255.255.255.0  
no shutdown  
!  
interface Serial0/1/0  
ip address 172.16.34.3 255.255.255.0  
clock rate 64000  
no shutdown  
!  
router ospf 1  
network 172.16.0.0 0.0.255.255 area 0  
!  
end
```

R4:

```
!  
hostname R4  
!  
interface Serial0/0/0  
ip address 172.16.34.4 255.255.255.0  
no shutdown  
!  
interface FastEthernet0/1  
ip address 172.16.14.4 255.255.255.0  
no shutdown  
!  
router ospf 1  
network 172.16.0.0 0.0.255.255 area 0  
!  
ip route vrf PAGENT 172.16.14.0 255.255.255.0 172.16.10.1  
ip route vrf PAGENT 172.16.23.0 255.255.255.0 172.16.10.1  
ip route vrf PAGENT 172.16.34.0 255.255.255.0 172.16.10.1  
!  
end
```

R4 NQR:

```
fastethernet0/0
add tcp
datalink ios-dependent fastethernet0/0.10
12-arp-for 172.16.10.1
13-src 172.16.10.4
13-dest 172.16.20.4
14-dest 23
fastethernet0/0.20 ios-dependent capture
```