第二章 服务质量与流量控制

实验 2-1 QoS 基础

学习目的

- 掌握使用NQA分析SLA的方法
- 掌握进行优先级映射和流量监管的方法
- 掌握配置流量整形的方法
- 掌握实现基于队列和基于流分类的拥塞管理方法
- 掌握配置WRED实现拥塞避免的方法

拓扑图

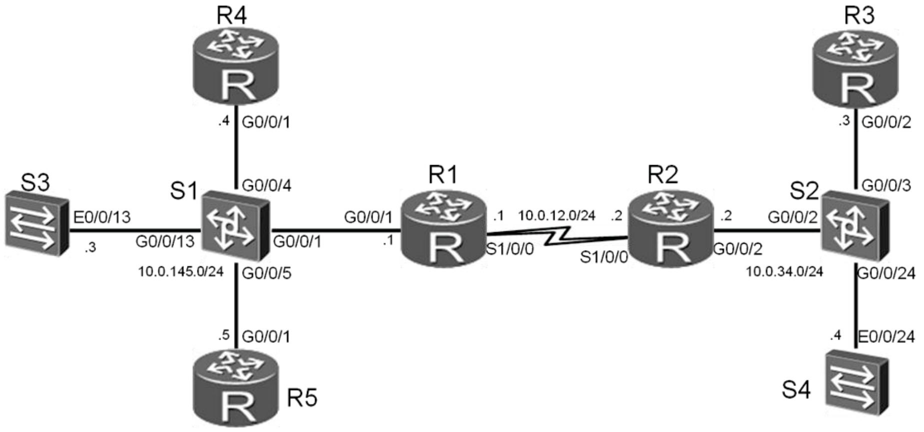

图2-1 QoS基础

场景

你是公司的网络管理员。公司网络分成两部分，其中R1与S1在公司总部，R2与S2在公司分部，之间通过专线实现互联。

随着网络的发展，内网带宽逐渐增大，而专线的带宽一直没有升级，所以网络中出现了比较严重的重要业务反应较慢，或无法正常使用的情况。

使用QoS的差分服务，你可以调整相应的QoS特性，保证重要的业务数据能更好的发送到目标。

实验中，S3和S4使用NQA相互发送数据，模拟大量数据流的发送。R3、R4与R5模拟客户端和服务器，测试重要应用是否可以正常使用。

学习任务

步骤一.　基础配置与 IP 编址

给所有路由器和交换机S3，S4配置IP地址和掩码。

配置时需要将R1接口S1/0/0的波特率配置为72000，作为广域网链路，因带宽不足而出现拥塞。

```
<Huawei>system-view
Enter system view, return user view with Ctrl+Z.
[Huawei]sysname R1
[R1]interface Serial 1/0/0
[R1-Serial1/0/0]ip address 10.0.12.1 255.255.255.0
[R1-Serial1/0/0]baudrate 72000
[R1-Serial1/0/0]interface GigabitEthernet 0/0/1
[R1-GigabitEthernet0/0/1]ip address 10.0.145.1 255.255.255.0

<Huawei>system-view
Enter system view, return user view with Ctrl+Z.
[Huawei]sysname R2
[R2]interface s1/0/0
[R2-Serial1/0/0]ip address 10.0.12.2 255.255.255.0
[R2-Serial1/0/0]interface GigabitEthernet 0/0/2
[R2-GigabitEthernet0/0/2]ip address 10.0.34.2 255.255.255.0

<Huawei>system-view
Enter system view, return user view with Ctrl+Z.
[Huawei]sysname R3
[R3]interface GigabitEthernet 0/0/2
[R3-GigabitEthernet0/0/2]ip address 10.0.34.3 255.255.255.0

<Huawei>system-view
Enter system view, return user view with Ctrl+Z.
[Huawei]sysname R4
```

```
[R4]interface GigabitEthernet 0/0/1
[R4-GigabitEthernet0/0/1]ip address 10.0.145.4 255.255.255.0

<Huawei>system-view
Enter system view, return user view with Ctrl+Z.
[Huawei]sysname R5
[R5]interface GigabitEthernet 0/0/1
[R5-GigabitEthernet0/0/1]ip address 10.0.145.5 255.255.255.0

<Huawei>system-view
Enter system view, return user view with Ctrl+Z.
[Huawei]sysname S3
[S3]interface vlan
[S3]interface Vlanif 1
[S3-Vlanif1]ip address 10.0.145.3 255.255.255.0

<Huawei>system-view
Enter system view, return user view with Ctrl+Z.
[Huawei]sysname S4
[S4]interface Vlanif 1
[S4-Vlanif1]ip address 10.0.34.4 255.255.255.0
```

配置完成后，测试直连链路的连通性。

```
[R1]ping -c 1 10.0.12.2
  PING 10.0.12.2: 56  data bytes, press CTRL_C to break
    Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=36 ms

  --- 10.0.12.2 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 36/36/36 ms

[R1]ping -c 1 10.0.145.3
  PING 10.0.145.3: 56  data bytes, press CTRL_C to break
    Reply from 10.0.145.3: bytes=56 Sequence=1 ttl=255 time=35 ms

  --- 10.0.145.3 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 35/35/35 ms
```

```
[R1]ping -c 1 10.0.145.4
  PING 10.0.145.4: 56  data bytes, press CTRL_C to break
    Reply from 10.0.145.4: bytes=56 Sequence=1 ttl=255 time=6 ms

  --- 10.0.145.4 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 6/6/6 ms

[R1]ping -c 1 10.0.145.5
  PING 10.0.145.5: 56  data bytes, press CTRL_C to break
    Reply from 10.0.145.5: bytes=56 Sequence=1 ttl=255 time=6 ms

  --- 10.0.145.5 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 6/6/6 ms

[R2]ping -c 1 10.0.34.3
  PING 10.0.34.3: 56  data bytes, press CTRL_C to break
    Reply from 10.0.34.3: bytes=56 Sequence=1 ttl=255 time=5 ms

  --- 10.0.34.3 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 5/5/5 ms

[R2]ping -c 1 10.0.34.4
  PING 10.0.34.4: 56  data bytes, press CTRL_C to break
    Reply from 10.0.34.4: bytes=56 Sequence=1 ttl=255 time=36 ms

  --- 10.0.34.4 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 36/36/36 ms
```

步骤二. 配置静态路由与 NQA

在所有路由器和交换机S3，S4上配置静态路由。

```
[R1]ip route-static 10.0.34.0 255.255.255.0 10.0.12.2

[R2]ip route-static 10.0.145.0 255.255.255.0 10.0.12.1

[R3]ip route-static 0.0.0.0 0.0.0.0 10.0.34.2

[R4]ip route-static 0.0.0.0 0.0.0.0 10.0.145.1

[R5]ip route-static 0.0.0.0 0.0.0.0 10.0.145.1

[S3]ip route-static 0.0.0.0 0.0.0.0 10.0.145.1

[S4]ip route-static 0.0.0.0 0.0.0.0 10.0.34.2
```

配置完成后，测试网络连通性。

```
[S3]ping -c 1 10.0.34.4
  PING 10.0.34.4: 56  data bytes, press CTRL_C to break
    Reply from 10.0.34.4: bytes=56 Sequence=1 ttl=252 time=40 ms

  --- 10.0.34.4 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 40/40/40 ms

[R4]ping -c 1 10.0.34.3
  PING 10.0.145.4: 56  data bytes, press CTRL_C to break
    Reply from 10.0.145.4: bytes=56 Sequence=1 ttl=255 time=3 ms

  --- 10.0.145.4 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 3/3/3 ms

[R5]ping -c 1 10.0.34.3
  PING 10.0.34.3: 56  data bytes, press CTRL_C to break
    Reply from 10.0.34.3: bytes=56 Sequence=1 ttl=253 time=44 ms
  --- 10.0.34.3 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 44/44/44 ms
```

S3去往S4，R4，R5去往R3可以连通，证明网络通信正常。

公司总部和分部之间的链路为72K串行链路，因而在实际情况中很容易造成拥塞。

实验中使用NQA在网络中产生流量。S4作为NQA服务器端，S3作为NQA客户端。

定义UDP，Jitter两种NQA测试例，分别用来模拟企业网中的数据流量和语音流量。

通过设置NQA测试例中的一些参数来实现两种流量中任何一种单独存在的情况下不会产生拥塞，二者共存的情况下会产生拥塞，来模拟实际环境。

在S4设上配置NQA服务器端，UDP监听的IP地址设为10.0.34.4，端口号设为6000。

```
[S4]nqa-server udpecho 10.0.34.4 6000
```

收到UDP回应,本机IP

在S3上配置UDP类型的NQA测试例模拟数据流量，其中**tos**设为28，包大小为5800字节，包间隔设为1s，周期设为3s，超时设为1s，并开启该测试。

```
[S3]nqa test-instance admin udp
[S3-nqa-admin-udp]test-type udp
[S3-nqa-admin-udp]destination-address ipv4 10.0.34.4
[S3-nqa-admin-udp]destination-port 6000
[S3-nqa-admin-udp]tos 28
[S3-nqa-admin-udp]datasize 5000
[S3-nqa-admin-udp]interval seconds 1
[S3-nqa-admin-udp]frequency 3
[S3-nqa-admin-udp]timeout 1
[S3-nqa-admin-udp]start now
```

udp

目的

端口

af32

每隔1s发送，共3次

NQA

Network Quality Analyzer

网络质量分析

查看UDP测试结果。

```
[S3]display nqa results test-instance admin udp
1 . Test 2 result   The test is finished
   Send operation times: 3              Receive response times: 3
   Completion:success                   RTD OverThresholds number: 0
   Attempts number:1                    Drop operation number:0
   Disconnect operation number:0       Operation timeout number:0
   System busy operation number:0      Connection fail number:0
   Operation sequence errors number:0  RTT Stats errors number:0
   Destination ip address:10.0.34.4
   Min/Max/Average Completion Time: 930/950/943
   Sum/Square-Sum  Completion Time: 2830/2669900
   Last Good Probe Time: 2008-01-28 23:10:02.4
   Lost packet ratio: 0 %
```

此时不丢包，链路没有产生拥塞。关闭UDP测试。

```
[S3]nqa test-instance admin udp
[S3-nqa-admin-udp]stop
```

在S3上配置Jitter类型的NQA测试例模拟语音流量，其中**tos**设为46，包大小为90字节，包间隔设为20ms，周期设为3s，超时设为1s，并开启该测试。

```
[S3]nqa test-instance admin jitter
[S3-nqa-admin-jitter]test-type jitter
[S3-nqa-admin-jitter]destination-address ipv4 10.0.34.4
[S3-nqa-admin-jitter]destination-port 6000
[S3-nqa-admin-jitter]tos 46
[S3-nqa-admin-jitter]datasize 90
[S3-nqa-admin-jitter]interval milliseconds 20
[S3-nqa-admin-jitter]frequency 3
[S3-nqa-admin-jitter]timeout 1
[S3-nqa-admin-jitter]start now
```

查看Jitter测试结果。

```
[S3]display nqa results test-instance admin jitter

 NQA entry(admin, jitter) :testflag is active ,testtype is jitter
  1 . Test 1 result   The test is finished
   SendProbe:60                        ResponseProbe:60
   Completion:success                  RTD OverThresholds number:0
   Min/Max/Avg/Sum RTT:40/70/54/3260   RTT  Square Sum:179800
   NumOfRTT:60                         Drop operation number:0
   Operation sequence errors number:0  RTT Stats errors number:0
   System busy operation number:0      Operation timeout number:0
   Min Positive SD:10                  Min Positive DS:10
   Max Positive SD:10                  Max Positive DS:10
   Positive SD Number:5                Positive DS Number:11
   Positive SD Sum:50                  Positive DS Sum:110
   Positive SD Square Sum:500          Positive DS Square Sum:1100
   Min Negative SD:10                  Min Negative DS:10
   Max Negative SD:10                  Max Negative DS:20
   Negative SD Number:4                Negative DS Number:10
   Negative SD Sum:40                  Negative DS Sum:110
   Negative SD Square Sum:400          Negative DS Square Sum:1300
   Min Delay SD:20                     Min Delay DS:19
   Avg Delay SD:27                     Avg Delay DS:26
   Max Delay SD:35                     Max Delay DS:34
   Packet Loss SD:0                    Packet Loss DS:0
   Packet Loss Unknown:0               jitter out value:0.0937500
   jitter in value:0.2291667           NumberOfOWD:60
   OWD SD Sum:1630                     OWD DS Sum:1570
   TimeStamp unit: ms
```

此时不丢包，链路没有产生拥塞。关闭Jitter测试。

```
[S3]nqa test-instance admin jitter
[S3-nqa-admin-jitter]stop
```

步骤三.　配置优先级映射

现在通过**ping**命令来模拟公司中一些不太重要的流量，并且针对这部分流量，将其DSCP优先级映射为BE，不做QoS保证。

配置R1的接口G0/0/1与S1/0/0信任报文的DSCP优先级。

```
[R1]interface GigabitEthernet 0/0/1
[R1-GigabitEthernet0/0/1]trust dscp override
[R1-GigabitEthernet0/0/1]interface Serial 1/0/0
[R1-Serial1/0/0]trust dscp
```

在接口G0/0/1上的**trust**命令中需要加上**override**参数，使得接下来在R1上配置优先级映射后，将DSCP值修改为映射后的值。

在R4上使用**ping**命令产生去往R3的流量，并且将**tos**设为26。

```
[R4]ping -tos 26 10.0.34.3
```

在R1上配置优先级映射关系，将该流量的DSCP报文优先级26映射为0，

```
[R1]qos map-table dscp-dscp
[R1-maptbl-dscp-dscp]input 26 output 0
```

查看R1上的优先级映射信息。

```
[R1]display qos map-table dscp-dscp
Input DSCP     DSCP
-------------------
 0             0
 1             1
 2             2
 3             3
 4             4
 5             5
 6             6
 7             7
 8             8
 9             9
10             10
11             11
12             12
13             13
14             14
15             15
16             16
17             17
18             18
19             19
20             20
21             21
22             22
23             23
24             24
25             25
26             0
27             27
28             28
29             29
30             30
```

此时可以观察到，现在已将DSCP报文优先级26映射成为了0，而其余DSCP值都是默认映射值。

步骤四.　配置整形与监管

开启S3上的NQA的UDP与Jitter测试，模拟公司总部与分部之间的72K链路产生拥塞。

```
[S3]nqa test-instance admin udp
[S3-nqa-admin-udp]start now
[S3-nqa-admin-udp]nqa test-instance admin jitter
[S3-nqa-admin-jitter]start now
```

在R4上使用**ping**命令实现模拟去往R3的流量，设置包大小为700字节，发10个包。

```
[R4]ping -s 700 -c 10 10.0.34.3
  PING 10.0.34.3: 700  data bytes, press CTRL_C to break
    Request time out
    Request time out
    Request time out
    Request time out
    Request time out
    Request time out
    Request time out
    Request time out
    Reply from 10.0.34.3: bytes=700 Sequence=9 ttl=253 time=1944 ms
    Request time out

  --- 10.0.34.3 ping statistics ---
    10 packet(s) transmitted
    1 packet(s) received
    90.00% packet loss
    round-trip min/avg/max = 1944/1944/1944 ms
```

此时公司总部与分部之间的链路发生严重拥塞，丢包现象严重，即使通过的数据包延迟也非常大。此时R4无法与R3建立正常通信。

下面将介绍分别通过使用流量监管和流量整形的方法来消除链路上的拥塞，使得公司总部的客户端R4与分部的客户端R3能够建立正常通信。

首先通过流量监管来消除拥塞。在S1上，针对拥塞流量入接口G0/0/13上配置流量监管，CIR设为64kbit/s。

```
[S1]interface GigabitEthernet 0/0/13
[S1-GigabitEthernet0/0/13]qos lr inbound cir 64
```

查看S1上流量监管的配置信息。

```
[S1]display qos lr inbound interface GigabitEthernet 0/0/13
GigabitEthernet0/0/13 lr inbound:
  cir: 64 Kbps, cbs: 8000 Byte
```

现在再回到R4上使用**ping**命令实现模拟去往R3的流量，设置包大小为700字节，发10个包。

```
[R4]ping -s 700 -c 10 10.0.34.3
  PING 10.0.34.3: 700  data bytes, press CTRL_C to break
    Reply from 10.0.34.3: bytes=700 Sequence=1 ttl=253 time=1412 ms
    Reply from 10.0.34.3: bytes=700 Sequence=2 ttl=253 time=255 ms
    Reply from 10.0.34.3: bytes=700 Sequence=3 ttl=253 time=736 ms
    Reply from 10.0.34.3: bytes=700 Sequence=4 ttl=253 time=1746 ms
    Reply from 10.0.34.3: bytes=700 Sequence=5 ttl=253 time=246 ms
    Reply from 10.0.34.3: bytes=700 Sequence=6 ttl=253 time=746 ms
    Reply from 10.0.34.3: bytes=700 Sequence=7 ttl=253 time=1736 ms
    Reply from 10.0.34.3: bytes=700 Sequence=8 ttl=253 time=258 ms
    Reply from 10.0.34.3: bytes=700 Sequence=9 ttl=253 time=766 ms
    Reply from 10.0.34.3: bytes=700 Sequence=10 ttl=253 time=1736 ms

  --- 10.0.34.3 ping statistics ---
    10 packet(s) transmitted
    10 packet(s) received
    0.00% packet loss
round-trip min/avg/max = 246/963/1746 ms
```

此时流量监管产生效果，不丢包，R4和R3之间能够建立起正常通信。

删除S1上流量监管配置。

```
[S1]interface GigabitEthernet 0/0/13
[S1-GigabitEthernet0/0/13]undo qos lr inbound
```

现在通过流量整形的方式来达到消除拥塞的目的。在S3上，针对拥塞流量出接口E0/0/13上配置流量整形，CIR设为64kbit/s。

```
[S3]interface Ethernet0/0/13
[S3-Ethernet0/0/13]qos lr outbound cir 64
```

在R4上使用**ping**命令实现模拟去往R3的流量，设置包大小为700字节，发10个包。

```
[R4]ping -s 700 -c 10 10.0.34.3
  PING 10.0.34.3: 700  data bytes, press CTRL_C to break
    Reply from 10.0.34.3: bytes=700 Sequence=1 ttl=253 time=240 ms
    Reply from 10.0.34.3: bytes=700 Sequence=2 ttl=253 time=284 ms
    Reply from 10.0.34.3: bytes=700 Sequence=3 ttl=253 time=334 ms
    Reply from 10.0.34.3: bytes=700 Sequence=4 ttl=253 time=224 ms
    Reply from 10.0.34.3: bytes=700 Sequence=5 ttl=253 time=344 ms
    Reply from 10.0.34.3: bytes=700 Sequence=6 ttl=253 time=275 ms
    Reply from 10.0.34.3: bytes=700 Sequence=7 ttl=253 time=534 ms
    Reply from 10.0.34.3: bytes=700 Sequence=8 ttl=253 time=184 ms
    Reply from 10.0.34.3: bytes=700 Sequence=9 ttl=253 time=204 ms
    Reply from 10.0.34.3: bytes=700 Sequence=10 ttl=253 time=314 ms

  --- 10.0.34.3 ping statistics ---
    10 packet(s) transmitted
    10 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 184/293/534 ms
```

此时流量监管产生效果，不丢包，R4和R3之间能够建立起正常通信。

删除S3上的流量整形配置，

```
[S3]interface Ethernet0/0/13
[S3-Ethernet0/0/13]undo qos lr outbound
```

现在再回到R4上使用**ping**命令实现模拟去往R3的流量，设置包大小为700字节，发10个包。

```
[R4]ping -s 700 -c 10 10.0.34.3
  PING 10.0.34.3: 700  data bytes, press CTRL_C to break
    Reply from 10.0.34.3: bytes=700 Sequence=1 ttl=253 time=1918 ms
    Request time out
    Reply from 10.0.34.3: bytes=700 Sequence=3 ttl=253 time=1762 ms
    Request time out
    Request time out
    Request time out
    Request time out
    Request time out
    Request time out
    Request time out

  --- 10.0.34.3 ping statistics ---
    10 packet(s) transmitted
    2 packet(s) received
    80.00% packet loss
    round-trip min/avg/max = 1762/1840/1918 ms
```

删除配置之后，丢包严重，并且通过的数据包延迟也非常大。R4与R3之间无法建立起正常通信。

步骤五. 配置基于队列的拥塞管理与拥塞避免

为了解决公司总部与分部之间产生的网络拥塞，现在通过配置基于队列的拥塞管理和拥塞避免的方式解决。

在R1上创建WRED丢弃模板**data**，使其基于DSCP优先级进行丢弃，将阀值上限设为90，下限设为50，丢弃概率设为30。

```
[R1]drop-profile data
[R1-drop-profile-data]wred dscp
[R1-drop-profile-data]dscp af32 low-limit 50 high-limit 90 discard-percentage
30
```

在R1上创建队列模板queue-profile1，将数据流量放入WFQ队列，并和丢弃模板data绑定，将需要高优先级，低延迟保证的语音流量放入PQ队列。

```
[R1]qos queue-profile queue-profile1
[R1-qos-queue-profile-queue-profile1]queue 3 drop-profile data
[R1-qos-queue-profile-queue-profile1]schedule wfq 3 pq 5
```

在R1的S1/0/0上应用队列模板。

```
[R1]interface Serial 1/0/0
[R1- Serial0/0/1]qos queue-profile queue-profile1
```

查看配置的队列模板信息。

```
[R1]display qos queue-profile queue-profile1
Queue-profile: queue-profile1
Queue  Schedule  Weight  Length(Bytes/Packets) Gts(CIR/CBS)
-----------------------------------------------------------------
3      WFQ      10                0/0              -/-
5      PQ       -                 0/0              -/-
```

此时数据流量与语音流量分别使用了WFQ与PQ队列。

查看配置的丢弃模板信息。

```
[R1]display drop-profile data
Drop-profile[1]: data
DSCP          Low-limit  High-limit  Discard-percentage
-----------------------------------------------------------------
default          30         100         10
1                30         100         10
2                30         100         10
3                30         100         10
```

```
4               30        100       10
5               30        100       10
6               30        100       10
7               30        100       10
cs1             30        100       10
9               30        100       10
af11            30        100       10
11              30        100       10
af12            30        100       10
13              30        100       10
af13            30        100       10
15              30        100       10
cs2             30        100       10
17              30        100       10
af21            30        100       10
19              30        100       10
af22            30        100       10
21              30        100       10
af23            30        100       10
23              30        100       10
cs3             30        100       10
25              30        100       10
af31            30        100       10
27              30        100       10
af32            50        90        30
29              30        100       10
af33            30        100       10
31              30        100       10
cs4             30        100       10
33              30        100       10
af41            30        100       10
```

可以观察到配置上限，下限阀值与丢弃概率产生的效果，其余作没有配置的对应的都是默认值。

步骤六.　配置基于流的拥塞管理与拥塞避免

为了解决公司总部与分部之间产生的网络拥塞，现在通过配置基于流的拥塞管理和拥塞避免的方式解决。

现在将公司总部的客户端R4与分部的客户端R3之间的流量定义为重要流量，通过对其做QoS保证，使得R4与R3能够建立正常的通信。

删除步骤五中R1接口S1/0/0上队列模板的调用。

```
[R1]interface GigabitEthernet 0/0/1
[R1-GigabitEthernet0/0/1]undo qos queue-profile
```

在R4上使用**ping**命令测试去往R3的连通性，设置源地址为10.0.145.4，包大小为700字节，发10个包。

```
[R4]ping -a 10.0.145.4 -s 700 -c 10 10.0.34.3
  PING 10.0.34.3: 700  data bytes, press CTRL_C to break
    Reply from 10.0.34.3: bytes=700 Sequence=1 ttl=253 time=1279 ms
    Request time out
    Reply from 10.0.34.3: bytes=700 Sequence=3 ttl=253 time=1587 ms
    Reply from 10.0.34.3: bytes=700 Sequence=4 ttl=253 time=1827 ms
    Request time out
    Reply from 10.0.34.3: bytes=700 Sequence=6 ttl=253 time=1717 ms
    Request time out
    Request time out
    Request time out
    Request time out

  --- 10.0.34.3 ping statistics ---
    10 packet(s) transmitted
    4 packet(s) received
    60.00% packet loss
    round-trip min/avg/max = 1279/1602/1827 ms
```

此时公司总部与分部之间的链路发生严重拥塞，丢包现象严重，R4无法与R3建立正常通信。

在R1上创建ACL3001匹配从10.0.145.4去往10.0.34.3的流量。

```
[R1]acl number 3001
[R1-acl-adv-3001]rule 0 per ip source 10.0.145.4 0.0.0.0 destination 10.0.34.3
0.0.0.0
```

创建流分类**class-ef**，匹配ACL3001，创建流行为**behavior-ef**，配置队列调度方式为EF，带宽为10Kbps。

```
[R1]traffic classifier class-ef
[R1-classifier-class-ef]if-match acl 3001
[R1-classifier-class-ef]traffic behavior behavior-ef
[R1-behavior-behavior-ef]queue ef bandwidth 8
```

创建流分类**class-af32**，匹配DSCP值为AF32的数据流量，创建流行为

behavior-af32，配置队列调度方式为AF，带宽为30Kbps，与丢弃模板data绑定。

```
[R1]traffic classifier class-af32
[R1-classifier-class-af32]if-match dscp af32
[R-classifier-class-af321]traffic behavior behavior-af32
[R1-behavior-behavior-af32]queue af bandwidth 30
[R1-behavior-behavior-af32]drop-profile data
```

创建流策略**policy-1**，关联流分类**class-ef**和流动作**behavior-ef**，流分类**class-af32**和流动作**behavior-af32**，并在R1的接口S1/0/0上应用。

```
[R1]traffic policy policy-1
[R1-trafficpolicy-policy-1]classifier class-ef behavior behavior-ef
[R1-trafficpolicy-policy-1]classifier class-af32 behavior behavior-af32
[R1-trafficpolicy-policy-1]interface Serial 1/0/0
[R1-Serial1/0/0]traffic-policy policy-1 outbound
```

在R4上使用**ping**命令测试去往R3的连通性，设置每个包大小为700，源地址为10.0.145.4，个数为10。

```
[R4]ping -a 10.0.145.4 -s 700 -c 10 10.0.34.3
  PING 10.0.34.3: 700  data bytes, press CTRL_C to break
    Reply from 10.0.34.3: bytes=700 Sequence=1 ttl=253 time=694 ms
    Reply from 10.0.34.3: bytes=700 Sequence=2 ttl=253 time=391 ms
    Reply from 10.0.34.3: bytes=700 Sequence=3 ttl=253 time=361 ms
    Reply from 10.0.34.3: bytes=700 Sequence=4 ttl=253 time=671 ms
    Reply from 10.0.34.3: bytes=700 Sequence=5 ttl=253 time=211 ms
    Reply from 10.0.34.3: bytes=700 Sequence=6 ttl=253 time=611 ms
    Reply from 10.0.34.3: bytes=700 Sequence=7 ttl=253 time=688 ms
    Reply from 10.0.34.3: bytes=700 Sequence=8 ttl=253 time=391 ms
    Reply from 10.0.34.3: bytes=700 Sequence=9 ttl=253 time=301 ms
    Reply from 10.0.34.3: bytes=700 Sequence=10 ttl=253 time=651 ms

  --- 10.0.34.3 ping statistics ---
    10 packet(s) transmitted
    10 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 211/497/694 ms
```

将R1去往R3的流量设置为EF队列后，现在R1可以与R3建立正常通信。

附加实验：思考并验证

QoS是使用差分服务来实现对不同业务的服务质量的保证，保证带宽和延迟。试想一下，带宽的增加是否可用彻底解决服务质量问题，从而不需要使用QoS？

实验完成后，回想理论课程中关于QoS的逻辑处理过程。将路由器实现QoS的过程总结一下。

最终设备配置

```
<R1>display current-configuration
[V200R001C00SPC200]
#
 sysname R1
#
acl number 3001
 rule 0 permit ip source 10.0.145.4 0 destination 10.0.34.3 0
#
drop-profile data
wred dscp
  dscp af32 low-limit 50 high-limit 90 discard-percentage 30
#
qos queue-profile queue-profile1
  queue 3 drop-profile data
  schedule wfq 3 pq 5
#
qos map-table dscp-dscp
  input 26 output 0
#
traffic classifier class-ef operator or
 if-match acl 3001
traffic classifier class-af32 operator or
 if-match dscp af32
#
traffic behavior behavior-ef
 queue ef bandwidth 10 cbs 250
traffic behavior behavior-af32
 queue af bandwidth 30
 drop-profile data
```

```
traffic behavior behavir-af32
 queue af bandwidth 30
#
traffic policy policy-1
 classifier class-ef behavior behavior-ef
 classifier class-af32 behavior behavior-af32
#
interface Serial1/0/0
 link-protocol ppp
 ip address 10.0.12.1 255.255.255.0
 trust dscp
 traffic-policy policy-1 outbound
 baudrate 72000
#
interface GigabitEthernet0/0/1
 ip address 10.0.145.1 255.255.255.0
 trust dscp override
#
 ip route-static 10.0.34.0 255.255.255.0 10.0.12.2
#
Return
```

```
<R2>display current-configuration
[V200R001C00SPC200]
#
 sysname R2
#
interface Serial1/0/0
 link-protocol ppp
 ip address 10.0.12.2 255.255.255.0
#
interface GigabitEthernet0/0/2
 ip address 10.0.34.2 255.255.255.0
#
 ip route-static 10.0.145.0 255.255.255.0 10.0.12.1
#
return
```

```
<R3>display current-configuration
[V200R001C00SPC200]
#
 sysname R3
#
```

```
interface GigabitEthernet0/0/2
 ip address 10.0.34.3 255.255.255.0
#
 ip route-static 0.0.0.0 0.0.0.0 10.0.34.2
#
return
```

```
<R4>display current-configuration
[V200R001C00SPC200]
#
 sysname R4
#
interface GigabitEthernet0/0/1
 ip address 10.0.145.4 255.255.255.0
#
 ip route-static 0.0.0.0 0.0.0.0 10.0.145.1
#
return
```

```
<R5>display current-configuration
[V200R001C00SPC200]
#
 sysname R5
#
interface GigabitEthernet0/0/1
 ip address 10.0.145.5 255.255.255.0
#
 ip route-static 0.0.0.0 0.0.0.0 10.0.145.1
#
return
```

```
<S3>display current-configuration
#
!Software Version V100R006C00SPC800
 sysname S3
#
interface Vlanif1
 ip address 10.0.145.3 255.255.255.0
#
 ip route-static 0.0.0.0 0.0.0.0 10.0.145.1
#
nqa test-instance admin udp
 test-type udp
 destination-address ipv4 10.0.34.4
 destination-port 6000
 tos 28
 frequency 3
 interval seconds 1
 timeout 1
 datasize 5800
 start now
nqa test-instance admin jitter
 test-type jitter
 destination-address ipv4 10.0.34.4
 destination-port 6000
 tos 46
 frequency 3
 interval milliseconds 20
 timeout 1
 datasize 90
 start now
#
return

<S4>display current-configuration
#
!Software Version V100R006C00SPC800
 sysname S4
#
interface Vlanif1
 ip address 10.0.34.4 255.255.255.0
#
 nqa-server udpecho 10.0.34.4 6000
#
 ip route-static 0.0.0.0 0.0.0.0 10.0.34.2
#
return
```

实验 2-2 使用流策略实现流行为控制

学习目的

- 掌握配置端到端QoS的方法
- 掌握使用流策略实现流行为控制的方法

拓扑图

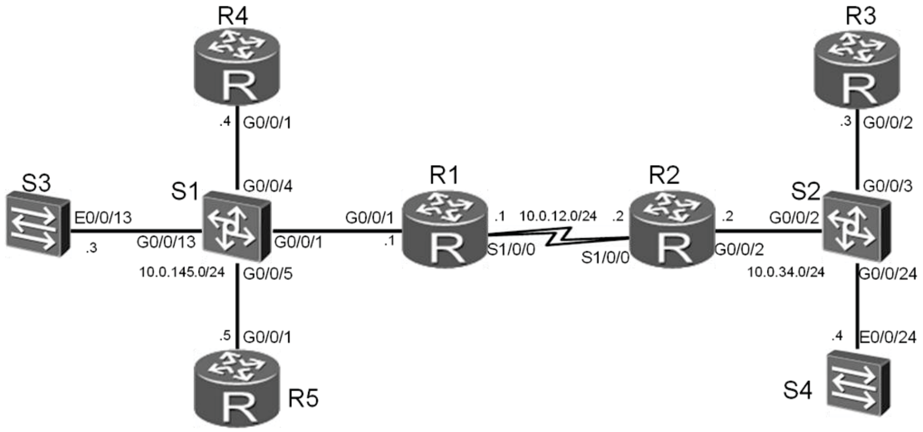

图2-2 使用流策略实现流行为控制

场景

你是公司的网络管理员。公司网络分成两部分，其中R1与S1在公司总部，R2与S2在公司分部，之间通过专线实现互联。随着网络的发展，内网带宽逐渐增大，而专线的带宽一直没有升级，所以网络中出现了比较严重的重要业务反应较慢，或无法正常使用的情况。

部署端到端QoS，你可以调整相应的QoS特性，保证重要的业务数据能更好的发送到目标，并通过流策略实现对流行为的控制。

学习任务

步骤一.　基础配置与 IP 编址

给所有路由器和交换机S3，S4配置IP地址和掩码。

```
<R1>system-view
Enter system view, return user view with Ctrl+Z.
[R1]interface Serial 1/0/0
[R1-Serial1/0/0]ip address 10.0.12.1 255.255.255.0
[R1-Serial1/0/0]interface GigabitEthernet 0/0/1
[R1-GigabitEthernet0/0/1]ip add 10.0.145.1 255.255.255.0

<R2>system-view
Enter system view, return user view with Ctrl+Z.
[R2]interface Serial 1/0/0
[R2-Serial1/0/0]ip address 10.0.12.2 255.255.255.0
[R2-Serial1/0/0]interface GigabitEthernet 0/0/2
[R2-GigabitEthernet0/0/2]ip address 10.0.34.2 255.255.255.0

<R3>system-view
Enter system view, return user view with Ctrl+Z.
[R3]interface GigabitEthernet 0/0/2
[R3-GigabitEthernet0/0/2]ip address 10.0.34.3 255.255.255.0

<R4> system-view
Enter system view, return user view with Ctrl+Z.
[R4]interface GigabitEthernet 0/0/1
[R4-GigabitEthernet0/0/1]ip address 10.0.145.4 255.255.255.0

<R5>system-view
Enter system view, return user view with Ctrl+Z.
[R5]interface GigabitEthernet 0/0/1
[R5-GigabitEthernet0/0/1]ip address 10.0.145.5 255.255.255.0

<S3>system-view
Enter system view, return user view with Ctrl+Z.
[S3]interface Vlanif 1
[S3-Vlanif1]ip address 10.0.145.3 255.255.255.0

<S4>system-view
Enter system view, return user view with Ctrl+Z.
```

```
[S4]interface Vlanif 1
[S4-Vlanif1]ip address 10.0.34.4 255.255.255.0
```

配置完成后，测试直连链路的连通性。

```
[R1]ping -c 1 10.0.12.2
  PING 10.0.12.2: 56  data bytes, press CTRL_C to break
    Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=36 ms

  --- 10.0.12.2 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 36/36/36 ms

[R1]ping -c 1 10.0.145.3
  PING 10.0.145.3: 56  data bytes, press CTRL_C to break
    Reply from 10.0.145.3: bytes=56 Sequence=1 ttl=255 time=35 ms

  --- 10.0.145.3 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 35/35/35 ms

[R1]ping -c 1 10.0.145.4
  PING 10.0.145.4: 56  data bytes, press CTRL_C to break
    Reply from 10.0.145.4: bytes=56 Sequence=1 ttl=255 time=6 ms

  --- 10.0.145.4 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 6/6/6 ms

[R1]ping -c 1 10.0.145.5
  PING 10.0.145.5: 56  data bytes, press CTRL_C to break
    Reply from 10.0.145.5: bytes=56 Sequence=1 ttl=255 time=6 ms

  --- 10.0.145.5 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 6/6/6 ms
```

```
[R2]ping -c 1 10.0.34.3
  PING 10.0.34.3: 56  data bytes, press CTRL_C to break
    Reply from 10.0.34.3: bytes=56 Sequence=1 ttl=255 time=5 ms

  --- 10.0.34.3 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 5/5/5 ms

[R2]ping -c 1 10.0.34.4
  PING 10.0.34.4: 56  data bytes, press CTRL_C to break
    Reply from 10.0.34.4: bytes=56 Sequence=1 ttl=255 time=36 ms

  --- 10.0.34.4 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 36/36/36 ms
```

步骤二. 配置静态路由

在所有路由器和交换机S3，S4上配置静态路由。

```
[R1]ip route-static 10.0.34.0 255.255.255.0 10.0.12.2

[R2]ip route-static 10.0.145.0 255.255.255.0 10.0.12.1

[R3]ip route-static 0.0.0.0 0.0.0.0 10.0.34.2

[R4]ip route-static 0.0.0.0 0.0.0.0 10.0.145.1

[R5]ip route-static 0.0.0.0 0.0.0.0 10.0.145.1

[S3]ip route-static 0.0.0.0 0.0.0.0 10.0.145.1

[S4]ip route-static 0.0.0.0 0.0.0.0 10.0.34.2
```

配置完成后，测试网络连通性。

```
[S3]ping -c 1 10.0.34.4
  PING 10.0.34.4: 56  data bytes, press CTRL_C to break
```

```
    Reply from 10.0.34.4: bytes=56 Sequence=1 ttl=252 time=40 ms

  --- 10.0.34.4 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 40/40/40 ms

[R4]ping -c 1 10.0.34.3
  PING 10.0.145.4: 56  data bytes, press CTRL_C to break
    Reply from 10.0.145.4: bytes=56 Sequence=1 ttl=255 time=3 ms

  --- 10.0.145.4 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 3/3/3 ms

[R5]ping -c 1 10.0.34.3
  PING 10.0.34.3: 56  data bytes, press CTRL_C to break
    Reply from 10.0.34.3: bytes=56 Sequence=1 ttl=253 time=44 ms

  --- 10.0.34.3 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 44/44/44 ms
```

步骤三. 配置 DSCP 优先级的重标记

公司网络中有语音，视频，数据三种业务，但是由于公司总部与分部之间的专线仍然没有得到升级，所以不可避免网络出现了拥塞。

通过配置端到端的QoS来实现语音报文的优先发送，视频报文的带宽保证。

将R4与R3之间的流量模拟为语音报文，将R5与R3之间的流量模拟为视频报文，将S3与S4之间的报文模拟为数据报文。接下来将针对语音报文和视频报文做一系列相关的QoS策略，对数据报文默认尽最大努力传输。

现在将语音报文的DSCP值标记为EF，视频报文的DSCP值标记为AF32。

在S1上创建ACL3001，3002，分别匹配R4去往R3，R5去往R3的流量。

```
[S1]acl number 3001
```

```
[S1-acl-adv-3001]rule 0 permit ip source 10.0.145.4 0 destination 10.0.34.3 0
[S1-acl-adv-3001]acl number 3002
[S1-acl-adv-3002]rule 0 permit ip source 10.0.145.5 0 destination 10.0.34.3 0
```

在S1上创建流分类class-voice-s1，匹配ACL3001。创建流行为behavior-voice-s1，将DSCP优先级重标记为EF。

创建流策略policy-voice-s1，关联流分类class-voice-s1与流行为behavior-voice-s1，在接口G0/0/4的入方向上调用该流策略。

```
[S1]traffic classifier class-voice-s1
[S1-classifier-class-voice-s1]if-match acl 3001
[S1-classifier-class-voice-s1]traffic behavior behavior-voice-s1
[S1-behavior-behavior-voice-s1]remark dscp ef
[S1-behavior-behavior-voice-s1]traffic policy policy-voice-s1
[S1-trafficpolicy-policy-voice-s1]classifier class-voice-s1 behavior
behavior-voice-s1
[S1-trafficpolicy-policy-voice-s1]interface GigabitEthernet 0/0/4
[S1-GigabitEthernet0/0/4]traffic-policy policy-voice-s1 inbound
```

在S1上创建流分类class-video-s1，匹配ACL3002。创建流行为behavior-video-s1，将DSCP优先级重标记为AF32。创建流策略policy-video-s1，关联流分类class-video-s1与流行为behavior-video-s1，在接口G0/0/5的入方向上应用该流策略。

```
[S1]traffic classifier class-video-s1
[S1-classifier-class-video-s1]if-match acl 3002
[S1-classifier-class-video-s1]traffic behavior behavior-video-s1
[S1-behavior-behavior-video-s1]remark dscp af32
[S1-behavior-behavior-video-s1]traffic policy policy-video-s1
[S1-trafficpolicy-policy-video-s1]classifier class-video-s1 behavior
behavior-video-s1
[S1-trafficpolicy-policy-video-s1]interface GigabitEthernet 0/0/5
[S1-GigabitEthernet0/0/5]traffic-policy policy-video-s1 inbound
```

在S2上创建ACL3001，3002，分别匹配R3去往R4，R3去往R5的流量。

```
[S2]acl number 3001
[S2-acl-adv-3001]rule 0 permit ip source 10.0.34.3 0 destination 10.0.145.4 0
[S2-acl-adv-3001]acl number 3002
[S2-acl-adv-3002]rule 0 permit ip source 10.0.34.3 0 destination 10.0.145.5 0
```

在S2上创建流分类class-voice-s2，匹配ACL3001。创建流行为behavior-voice-s2，将DSCP优先级重标记为EF。

```
[S2]traffic classifier class-voice-s2
```

```
[S2-classifier-class-voice-s2]if-match acl 3001
[S2-classifier-class-voice-s2]traffic behavior behavior-voice-s2
[S2-behavior-behavior-voice-s2]remark dscp ef
```

在S2上创建流分类class-video-s2，匹配ACL3002。创建流行为behavior-video-s2，将DSCP优先级重标记为AF32。

```
[S2]traffic classifier class-video-s2
[S2-classifier-class-video-s2]if-match acl 3002
[S2-classifier-class-video-s2]traffic behavior behavior-video-s2
[S2-behavior-behavior-video-s2]remark dscp af32
```

在S2上创建流策略policy-voice-video-s2，关联流分类class-voice-s2与流行为behavior-voice-s2，关联流分类class-video-s2与流行为behavior-video-s2，在接口G0/0/3的入方向上应用该流策略。

```
[S2]traffic policy policy-voice-video-s2
[S2-trafficpolicy-policy-voice-video-s2]classifier class-voice-s2 behavior
behavior-voice-s2
[S2-trafficpolicy-policy-voice-video-s2]classifier class-video-s2 behavior
behavior-video-s2
[S2]interface GigabitEthernet 0/0/3
[S2-GigabitEthernet0/0/3]traffic-policy policy-voice-video-s2 inbound
```

步骤四. 配置流量整形和监管

在公司总部和分部的核心交换机上部署流量整形，缓解流量拥塞。

在S1的接口G0/0/1出方向上配置流量整形，CIR设为128kbit/s。

```
[S1]interface GigabitEthernet 0/0/1
[S1-GigabitEthernet0/0/1]qos lr outbound cir 128
```

查看流量整形配置信息。

```
[S1]display qos lr outbound interface GigabitEthernet 0/0/1
GigabitEthernet0/0/1 lr outbound:
  cir: 128 Kbps, cbs: 16000 Byte
```

在S2的接口G0/0/2出方向上配置流量整形，CIR设为128kbit/s。

```
[S2]interface GigabitEthernet 0/0/2
[S2-GigabitEthernet0/0/2]qos lr outbound cir 128
```

查看流量整形配置信息。

```
[S2]display qos lr outbound interface GigabitEthernet 0/0/2
GigabitEthernet0/0/2 lr outbound:
  cir: 128 Kbps, cbs: 16000 Byte
```

在公司总部和分部的出口路由器上部署流量监管，进一步缓解流量拥塞。

在R1的接口G0/0/1入方向上配置流量监管，CIR设为72kbit/s。

```
[R1]interface GigabitEthernet 0/0/1
[R1-GigabitEthernet0/0/1]qos car inbound cir 72
```

在R2的接口G0/0/2入方向上配置流量监管，CIR设为72kbit/s。

```
[R2]interface GigabitEthernet 0/0/2
[R2-GigabitEthernet0/0/2]qos car inbound cir 72
```

步骤五. 配置基于流策略的拥塞管理与拥塞避免

在公司总部与分部的出口路由器上部署基于流策略的拥塞管理与拥塞避免。保证语音流量低延迟，优先发送，保证视频流量拥有足够的带宽。

配置R1的接口G0/0/1信任DSCP优先级。

```
[R1]interface GigabitEthernet 0/0/1
[R1-GigabitEthernet0/0/1]trust dscp
```

在R1上创建WRED丢弃模板video-r1，使其基于DSCP优先级进行丢弃，将阀值下限设为50，上限设为90，丢弃概率设为30，

```
[R1]drop-profile video
[R1-drop-profile-video-r1]wred dscp
[R1-drop-profile-video-r1]dscp af32 low-limit 50 high-limit 90
discard-percentage 30
```

在R1上创建流分类class-af32-r1，匹配DSCP值为AF32的视频流量。创建流行为behavior-af32-r1，配置队列调度方式为AF，最大带宽占接口带宽百分比设为40，并与丢弃模板video-r1绑定。

```
[R1]traffic classifier class-af32-r1
[R1-classifier-class-af32-r1]if-match dscp af32
[R1-classifier-class-af32-r1]traffic behavior behavior-af32-r1
[R1-behavior-behavior-af32-r1]queue af bandwidth pct 40
[R1-behavior-behavior-af32-r1]drop-profile video-r1
```

在R1上创建流分类class-ef-r1，匹配DSCP值为EF的语音流量。创建流行为behavior-ef-r1，配置队列的调度方式为EF，最大带宽占接口带宽百分比设为30。

```
[R1]traffic classifier class-ef-r1
[R1-classifier-class-ef-r1]if-match dscp ef
[R1-classifier-class-ef-r1]traffic behavior behavior-ef-r1
[R1-behavior-behavior-ef-r1]queue ef bandwidth pct 30
```

在R1上创建流策略policy-r1，关联流分类class-af32-r1与流行为behavior-af32-r1，关联流分类class-ef-r1与流行为behavior-ef-r1，并在接口S1/0/0的出方向上应用。

```
[R1]traffic policy policy-r1
[R1-trafficpolicy-policy-r1]classifier class-af32-r1 behavior behavior-af32-r1
[R1-trafficpolicy-policy-r1]classifier class-ef-r1 behavior behavior-ef-r1
[R1-trafficpolicy-policy-r1]interface Serial 1/0/0
[R1-Serial1/0/0]traffic-policy policy-r1 outbound
```

在公司总部R1上配置完后，在公司分部R2上也作相应配置。

配置R2的接口G0/0/2信任DSCP优先级。

```
[R2]interface GigabitEthernet 0/0/2
[R2-GigabitEthernet0/0/2]trust dscp
```

在R2上创建WRED丢弃模板video-r2，使其基于DSCP优先级进行丢弃，将阈值下限设为50，上限设为90，丢弃概率设为30，

```
[R2]drop-profile video-r2
[R2-drop-profile-video-r2]wred dscp
[R2-drop-profile-video-r2]dscp af32 low-limit 50 high-limit 90
discard-percentage 30
```

在R1上创建流分类class-af32-r2，匹配DSCP值为AF32的视频流量。创建流行为behavior-af32-r2，配置队列调度方式为AF，最大带宽占接口带宽百分比设为40，并与丢弃模板video-r2绑定。

```
[R2]traffic classifier class-af32-r2
[R2-classifier-class-af32-r2]if-match dscp af32
[R2-classifier-class-af32-r2]traffic behavior behavior-af32-r2
[R2-behavior-behavior-af32-r2]queue af bandwidth pct 40
[R2-behavior-behavior-af32-r2]drop-profile video-r2
```

在R1上创建流分类class-ef-r2，匹配DSCP值为EF的语音流量。创建流行为behavior-ef-r2，配置队列的调度方式为EF，最大带宽占接口带宽百分比设为30。

```
[R2]traffic classifier class-ef-r2
[R2-classifier-class-ef-r2]if-match dscp ef
[R2-classifier-class-ef-r2]traffic behavior behavior-ef-r2
[R2-behavior-behavior-ef-r2]queue ef bandwidth pct 30
```

在R1上创建流策略policy-r2，关联流分类class-af32-r2与流行为behavior-af32-r2，关联流分类class-ef-r1与流行为behavior-ef-r2，并在接口S1/0/0的出方向上应用。

```
[R2]traffic policy policy-r2
[R2-trafficpolicy-policy-r2]classifier class-af32-r2 behavior behavior-af32-r2
[R2-trafficpolicy-policy-r2]classifier class-ef-r2 behavior behavior-ef-r2
[R2]interface Serial 1/0/0
[R2-Serial1/0/0]traffic-policy policy-r2 outbound
```

步骤六. 配置基于流策略实现流行为控制

公司总部现在出于优化的的目的将针对部分流量做控制，丢弃掉UDP端口号4000至5000部分的视频流量。

在R1上创建ACL3003，匹配从R5去往R3，UDP端口范围为4000至5000部分的流量。

```
[R1]acl number 3003
[R1-acl-adv-3003]rule 0 permit udp source-port range 4000 5000 source 10.0.145.5
0 destination 10.0.34.3 0
```

在R1上创建流分类class-drop，匹配ACL3003，

```
[R1]traffic classifier class-drop
[R1-classifier-class-drop]if-match acl 3003
```

在R1上创建流行为behavior-drop，配置命令**deny**，执行禁止动作，

```
[R1]traffic behavior behavior-drop
[R1-behavior-behavior-drop]deny
```

在R1上创建流策略policy-drop，关联流分类class-drop与流行为behavior-drop，并在接口G0/0/5的入方向上应用。

```
[R1]traffic policy policy-drop
[R1-trafficpolicy-policy-drop]classifier class-drop behavior behavior-drop
[R1-trafficpolicy-policy-drop]interface GigabitEthernet 0/0/1
[R1-GigabitEthernet0/0/1]traffic-policy policy-drop inbound
```

查看配置信息。

```
[R1]dis traffic policy user-defined policy-drop
  User Defined Traffic Policy Information:
    Policy: policy-drop
    Classifier: class-drop
    Operator: OR
    Behavior: behavior-drop
    Deny
```

附加实验：思考并验证

实验完成后，回顾QoS的知识框架，总结QoS中各项策略的使用范围与应用场景。

最终设备配置

```
<R1>display current-configuration
[V200R001C00SPC200]
#
 sysname R1
#
acl number 3003
 rule 0 permit udp source 10.0.145.5 0 source-port range 4000 5000 destination
10.0.34.3 0
#
drop-profile video-r1
wred dscp
  dscp af32 low-limit 50 high-limit 90 discard-percentage 30
#
traffic classifier class-drop operator or
 if-match acl 3003
traffic classifier class-ef-r1 operator or
 if-match dscp ef
traffic classifier class-af32-r1 operator or
 if-match dscp af32
#
traffic behavior behavior-af32-r1
 queue af bandwidth pct 40
 drop-profile video-r1
traffic behavior behavior-ef-r1
```

```
 queue ef bandwidth pct 30
traffic behavior behavior-drop
 deny
#
traffic policy policy-drop
 classifier class-drop behavior behavior-drop
traffic policy policy-r1
 classifier class-af32-r1 behavior behavior-af32-r1
 classifier class-ef-r1 behavior behavior-ef-r1
#
interface Serial1/0/0
 link-protocol ppp
 ip address 10.0.12.1 255.255.255.0
 traffic-policy policy-r1 outbound
#
interface GigabitEthernet0/0/1
 ip address 10.0.145.1 255.255.255.0
 trust dscp
 qos car inbound cir 72 cbs 13536 pbs 22536 green pass yellow pass red discard
 traffic-policy policy-drop inbound
#
 ip route-static 10.0.34.0 255.255.255.0 10.0.12.2
#
return

<R2>display current-configuration
[V200R001C00SPC200]
#
 sysname R2
#
drop-profile video-r2
wred dscp
  dscp af32 low-limit 50 high-limit 90 discard-percentage 30
#
traffic classifier class-ef-r2 operator or
 if-match dscp ef
traffic classifier class-af32-r2 operator or
 if-match dscp af32
#
traffic behavior behavior-af32-r2
 queue af bandwidth pct 40
 drop-profile video-r2
traffic behavior behavior-ef-r2
```

```
 queue ef bandwidth pct 30
#
traffic policy policy-r2
 classifier class-af32-r2 behavior behavior-af32-r2
 classifier class-ef-r2 behavior behavior-ef-r2
#
interface Serial1/0/0
 link-protocol ppp
 ip address 10.0.12.2 255.255.255.0
 traffic-policy policy-r2 outbound
#
interface GigabitEthernet0/0/2
 ip address 10.0.34.2 255.255.255.0
 trust dscp
 qos car inbound cir 72 cbs 13536 pbs 22536 green pass yellow pass red discard
#
 ip route-static 10.0.145.0 255.255.255.0 10.0.12.1
#
return
```

```
<R3>display current-configuration
[V200R001C00SPC200]
#
 sysname R3
#
interface GigabitEthernet0/0/2
 ip address 10.0.34.3 255.255.255.0
#
 ip route-static 0.0.0.0 0.0.0.0 10.0.34.2
#
return
```

```
<R4>display current-configuration
[V200R001C00SPC200]
#
 sysname R4
#
interface GigabitEthernet0/0/1
 ip address 10.0.145.4 255.255.255.0
#
 ip route-static 0.0.0.0 0.0.0.0 10.0.145.1
#
return
```

```
<R5>display current-configuration
[V200R001C00SPC200]
#
 sysname R5
#
interface GigabitEthernet0/0/1
 ip address 10.0.145.5 255.255.255.0
#
 ip route-static 0.0.0.0 0.0.0.0 10.0.145.1
#
return
```

```
<S1>display current-configuration
#
!Software Version V100R006C00SPC800
 sysname S1
#
acl number 3001
 rule 0 permit ip source 10.0.145.4 0 destination 10.0.34.3 0
acl number 3002
 rule 0 permit ip source 10.0.145.5 0 destination 10.0.34.3 0
#
traffic classifier class-video-s1 operator and
 if-match acl 3002
traffic classifier class-voice-s1 operator and
 if-match acl 3001
#
traffic behavior behavior-video-s1
 remark dscp af32
traffic behavior behavior-voice-s1
 remark dscp ef
#
traffic policy policy-video-s1
 classifier class-video-s1 behavior behavior-video-s1
traffic policy policy-voice-s1
 classifier class-voice-s1 behavior behavior-voice-s1
#
interface GigabitEthernet0/0/1
 qos lr outbound cir 128 cbs 16000
#
interface GigabitEthernet0/0/4
 traffic-policy policy-voice-s1 inbound
```

```
#
interface GigabitEthernet0/0/5
 traffic-policy policy-video-s1 inbound
#
return
```

```
<S2>display current-configuration
#
!Software Version V100R006C00SPC800
 sysname S2
#
acl number 3001
 rule 0 permit ip source 10.0.34.3 0 destination 10.0.145.4 0
acl number 3002
 rule 0 permit ip source 10.0.34.3 0 destination 10.0.145.5 0
#
traffic classifier class-video-s2 operator and
 if-match acl 3002
traffic classifier class-voice-s2 operator and
 if-match acl 3001
#
traffic behavior behavior-video-s2
 remark dscp af32
traffic behavior behavior-voice-s2
 remark dscp ef
#
traffic policy policy-voice-video-s2
 classifier class-voice-s2 behavior behavior-voice-s2
 classifier class-video-s2 behavior behavior-video-s2
#
interface GigabitEthernet0/0/2
 qos lr outbound cir 128 cbs 16000
#
interface GigabitEthernet0/0/3
 traffic-policy policy-voice-video-s2 inbound
#
return
```

```
<S3>display current-configuration
#
!Software Version V100R006C00SPC800
 sysname S3
#
```

```
interface Vlanif1
 ip address 10.0.145.3 255.255.255.0
#
 ip route-static 0.0.0.0 0.0.0.0 10.0.145.1
#
return

<S4>display current-configuration
#
!Software Version V100R006C00SPC800
 sysname S4
#
interface Vlanif1
 ip address 10.0.34.4 255.255.255.0
#
 ip route-static 0.0.0.0 0.0.0.0 10.0.34.2
#
return
```

第三章 综合实验

实验 3-1 综合实验 1（选做）

学习目的

- 掌握MST的配置方法
- 掌握VLAN间路由的配置方法
- 掌握RIP的配置方法
- 掌握OSPF的配置方法
- 掌握路由引入的配置方法
- 掌握路由策略的配置方法
- 掌握防火墙的配置方法
- 掌握交换机上QOS的配置方法

拓扑图

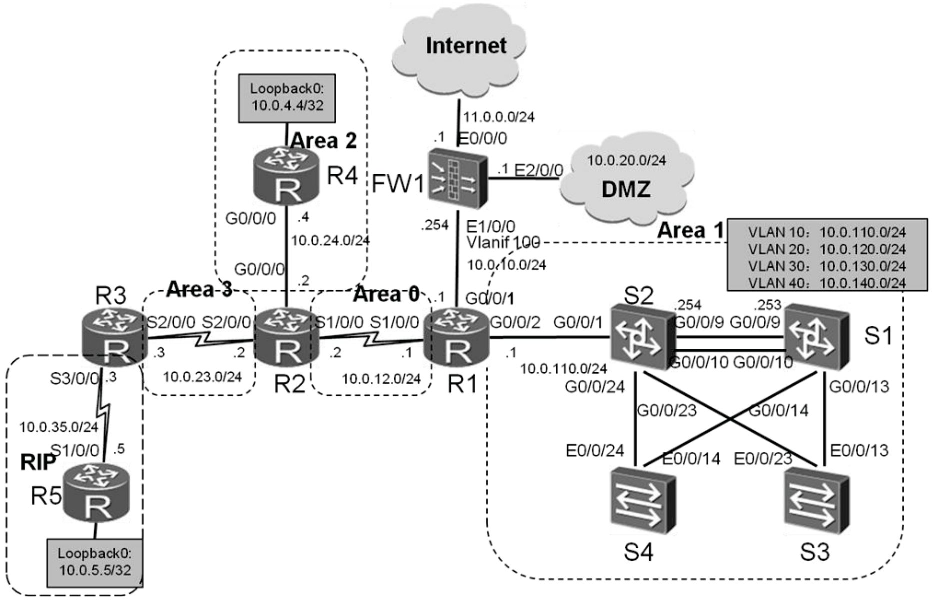

图 3-1 综合实验1（选做）

场景

你是公司的网络管理员。公司的网络由一个总部网络区域，一个分部网络区域和一个分支办公室网络区域组成。

其中总部网络区域由一台防火墙、一台路由器和四台交换机组成。防火墙控制公司内部网络与外部网络之间的互访，将网络划分为Trust、Untrust和DMZ三个区域。四台交换机使用MST技术实现网络冗余，提高网络可靠性。并使用QOS技术在交换网络上对数据流进行优化。

总部网络和分支办公室网络的路由器使用专线相连，同处于OSPF路由域中。为了优化OSPF路由域，将总部网络和分支办公室网络配置为OSPF末节区域型网络。由于分支机构使用的网络协议为RIP，需要在OSPF边界使用路由引入，以实现RIP路由域和OSPF路由域的互通。

学习任务

由于本综合实验的目的在于检测学员对前面实验学习掌握的程度，所以仅给出大概步骤和验证方式，不给出具体的操作命令。

步骤一. 基础配置与 IP 编址

给所有设备配置IP地址和掩码，配置完成后测试直连设备的连通性。

步骤二. MST 配置

交换机S1与S2之间连线为Eth-trunk链路。

将交换机与交换机之间连线的接口模式改为Trunk模式，并允许VLAN 10/20/30和40通过。

在所有交换机上都创建VLAN 10、20、30、40、100，同时配置MST生成两个实例。VLAN 10、VLAN 20和VLAN 100以S1为根，VLAN 30和VLAN 40以S2为根。

步骤三. VLAN 间路由配置

将S1的G0/0/22和G0/0/1接口加入VLAN 100，将S2的G0/0/1接口加入VLAN 10。

在S1和S2上为VLAN 10、20、30、40创建相应Vlanif接口，实现VLAN间通信。

步骤四. OSPF 配置

在R1、R2、R3、R4和S1、S2上配置OSPF路由协议。将R1和R2之间的链路配置属于OSPF区域0。总部网络配置属于OSPF区域1，分支办公室网络配置属于OSPF区域2，同时将区域1和区域2配置为OSPF末节区域。将R2与R3相连网络配置属于区域3 ,并将区域3配置为NSSA区。R1连接FW1的网络不运行OSPF。

步骤五. 路由引入配置

在R3和R5上配置RIP路由协议。在R3上使用路由引入，配置RIP路由域和OSPF路由域互相引入，实现RIP路由域和OSPF路由域之间的互通。在将RIP路由引入OSPF路由域的时候使用路由策略控制只引入R5连接的网络，而不把R3与R2直接的网络引入OSPF路由域。

在FW1上创建VLAN 100及相应的Vlanif接口，并按照拓扑图所示配置IP地址。在R1上配置一条缺省路由，下一跳地址为FW1的Vlanif 100接口地址。同时将这条路由信息引入OSPF，并让R5学习到。

同时在FW1上创建静态路由10.0.0.0/16，下一跳地址为R1的G0/0/1接口地址。使FW1能够和企业内网所有设备通信。

步骤六. 防火墙配置

按拓扑图所示，将FW1上相应端口分别加入Trust、Untrust和DMZ区域中。实现Trust区域可以访问所有区域的内容，Untrust区域只能访问DMZ区域中的服务器10.0.20.1的80号端口。DMZ区域不能主动访问所有区域。

步骤七. 网络优化配置

连接在交换机S4上的用户有些需要限制数据传输速度，有些需要提高数据传输优先级。E0/0/1接口属于VLAN 10，E0/0/2属于VLAN 30。请将S4的E0/0/1接口传输速度限制为128Kb。将E0/0/2接口的报文DSCP值修改为45，并设置E0/0/2信任报文的DSCP值。

附加实验：思考并验证

综合实验更贴近实际场景，对比之前的实验与这个实验，简述有哪些差异。

最终设备配置

```
[R1]display current-configuration
```

```
[R2]display current-configuration
```

```
[R3]display current-configuration
```

```
[R4]display current-configuration
```

```
[R5]display current-configuration
```

```
[S1]display current-configuration
```

```
[S2]display current-configuration
```

```
[S3]display current-configuration
```

```
[S4]display current-configuration
```

```
[FW1]display current-configuration
```