RSVP

The purpose of the measurement is using RSVP protocol to provide QoS in IP based networks. During the measurement, a report must be made about the performed tasks and results (including some screenshots). The report must contain the name and code of students and can be saved in the Desktop folder!

Topology used in the measurement

- Source PC -- FORRÁS: Linux operating system and rsvpd daemon
- Destination PC -- NYELŐ: Linux operating system and rsvpd daemon
- 2 Cisco (26xx 36xx) routers

Source and Destination PCs are also used to generate background traffic.

Preparation

First, switch on the two PCs besides the routers.

- Using meres2/meres2 username/password pair on Source PC (192.168.0.2)
- Using meres4/meres4 username/password pair on Destination PC (192.168.0.4)

If we have already logged on to the machines then turn on the two routers in the back.

Note that the whole measurement can be done on both the source and destination machines at the same time or only on a single machine using ssh; For the latter case, see later.

Initially,

- On the Source: eth0 NIC is configured with 192.168.0.2 and eth1 is down.
- On the Destination: eth1 NIC is configured with 192.168.0.4 and eth0 is down.

*It’s IMPORTANT that both Source and Destination are booted from network via 192.168.0.2 (eth0) and 192.168.0.4 (eth1), respectively. Thus, these NICs cannot be configured, otherwise we must restart the machines.*
To open Terminal(s): ALT + F2 -> ENTER -> enter: `xterm` -> run button ENTER (Better alternatives are Terminal or Konsole from Applications as we can copy and paste texts with them)

**On the Source:**

Enter the following commands in bold letters into the terminal:

**su** (password: meres – you will need root privileges for some commands)

**ifconfig eth1 193.168.2.1** (setting the IP address of the interface to be used for the measurement)

*(Before you may check whether this is really the interface to be set!!! DO NOT configure the NIC has ip of 192.168.0.x!)*

**route add -net 193.168.1.0 netmask 255.255.255.0 gw 193.168.2.254**

(that is, we send packets to the 193.168.1.X ip domain via the 193.168.2.254 router interface)

**route -n** (with this command you can see the result of the setting)

After setup:

```
eth0  192.168.0.2
eth1  193.168.2.1
```

**On the Destination:**

Enter the following commands in bold letters into the terminal:

**su**

**ifconfig eth0 193.168.1.1**

**route add -net 193.168.2.0 netmask 255.255.255.0 gw 193.168.1.254**
route -n

After setup:

<table>
<thead>
<tr>
<th>Interface</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth1</td>
<td>192.168.0.4</td>
</tr>
<tr>
<td>eth0</td>
<td>193.168.1.1</td>
</tr>
</tbody>
</table>

Measurement from a single PC:
If you do not want to work directly on both the source and the destination PCs, you may use one PC and connect to the other through `ssh` and do the measurement only on one computer.

For example, from 192.168.0.2

```
ssh meres4@192.168.0.4
```

The password is the same as the user name. If you get a prompt asking whether you want to connect, type: `yes`

So, on one PC, you can perform the tasks you need to run on two machines. On the screen, it is best to place the source page on the left side, while the destination side to the right (such as terminals to which we issue instructions.)

Configuring routers

From the Source, execute `ping -c 3 193.168.1.1`

If ping is not working, there may still need to be some settings missing in the router. For example, the current ethernet0/0 interface (or serial0/0) is not turned on. In this case, the routers must be reconfigured in order to send traffic between Source and Destination.

You can check this with the `show run` command (step with the SPACE key). There at the current interface you may find the shutdown command option as well. So there the `no shutdown` command required for that interface. These commands must be issued in the `minicom`

On Source:
Open a new terminal and type into it:

```
minicom
```

Then these (commands in bold) should be entered by line:

```
voip3600> enable
Password: cisco
voip3600 # show run
```
Make sure that both Ethernet 0/0 and Serial 0/0 are up, i.e. there is no “shutdown” line in the corresponding record. Furthermore, we may need to set clock rate for Serial 0/0 if it is not set with following commands:

```
voip3600 # configure terminal
voip3600 (config) # interface ethernet 0/0
voip3600 (config-if) # no shutdown
voip3600 (config-if) # exit
voip3600 (config) # interface serial 0/0
voip3600 (config-if) # no shutdown
voip3600 (config-if) # clock rate 2000000
voip3600 (config-if) # exit
voip3600 (config) # exit
voip3600 # show run
```

(by pressing CTRL + A and Q, you can exit from the minicom, help: CTRL + A and Z)

**On Destination**

Open a new terminal and type into it:

```
minicom
```

Then wait a bit...

```
boot
```

The answer for the incoming question is "no" followed by ENTER.

Router> enable
Router # configure memory
...
Press ENTER (then, in theory, enable is not necessary anymore)

```
voip2600> enable
(Password: cisco)
voip2600 # show run
```

Similarly, make sure that both Ethernet 0/0 and Serial 0/0 are up and the clock rate is set for Serial 0/0 After these settings, we can check the links with ping. For example on the source:

```
ping 193.168.2.254 (adjacent router)
ping 193.168.1.254 (destination side router)
ping 193.168.1.1 (Destination)
```

**Measurement tasks**

1. RSVP structure, measuring traffic parameters with different background traffic a, enable the use of RSVP in CISCO routers, then use `rsvpd` on Linux machines to reserve resources between the source and the destination route (with the Tspec and Rspec parameters specified by the instructor.)
Note that the API of RSVP is not used in this measurement; instead, we are starting the process "manually". To do so, run rsvpd with the -D switch, so besides the rsvpd, the program called rtap will be started immediately too. You also have to provide the neighboring RSVP-enabled router’s IP address to the rsvpd with the -R parameter. Tip: the RSVP session must be defined on both sides of the rsvpd (rtap) program.

The usage of rsvpd can be found in RSVPmeres/doc/rsvpd. Enter the Terminal:

```
cd RSVPmeres/doc/rsvpd
mc
```

Then press F3 to view the required file:

```
F3: rsvpd.8
F3: rtap.8
```

b, Show the effect of a reservation by generating traffic between the source and the destination and generating a suitable background traffic. Do the measurement with two different background traffics of different size. To do this, follow the steps below:

• launch background traffic in the same direction as the download via the routers. To do this, you can use the `iperf` program to generate TCP background traffic.

Tip: Client side should use the following parameters: -P -t -i

• Measure the network transmission speed without QoS, and with QoS (Note: You might want to measure the background traffic as well to check your results)

Tip: You can also use `iperf` if the port used is the same that was given in the reservation.

**Solution:**

a.)

**Enable RSVP in Routers:**

To enable RSVP in the routers, you must make settings within the router. For this you have to use the minicom software to connect to the router.

You can use the following command in the minicom program:

```
ip rsvp bandwidth [interface-kbps] [single-flow-kbps] Enables RSVP for IP on an interface.
```

(http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt5/qcdrsvp.htm)

Settings must be made within both routers, that is, the source side and the destination. First, set the source side and then destination the side:

**Source side:**

Type into the terminal if minicom is not run:

```
minicom
```
voip3600> enable
Password: cisco (this is the password)
voip3600 # configure terminal
voip3600 (config) # interface Ethernet 0/0
voip3600 (config) # ip rsvp bandwidth 3750 1500
voip3600 (config) # exit
voip3600 (config) # interface Serial 0/0
voip3600 (config) # ip rsvp bandwidth 1500 1500
voip3600 (config) # exit
voip3600 (config) # exit
voip3600 # show run

(by pressing CTRL + A and Q, you can exit from the minicom, help: CTRL + A and Z)

Destination side:

Type into the terminal if minicom is not run:

    minicom

voip2600> enable
(Password: cisco)
voip2600 # configure terminal
voip2600 (config) # interface Ethernet 0/0
voip2600 (config) # ip rsvp bandwidth 3750 1500
voip2600 (config) # exit
voip2600 (config) # interface Serial 0/0
voip2600 (config) # ip rsvp bandwidth 1500 1500
voip2600 (config) # exit
voip2600 (config) # exit
voip2600 # show run

By doing this, we have enabled RSVP on both the source and the destination side.

After these settings, it is advisable to check the RSVP settings in routers with show ip rsvp commands. Type show ip rsvp? for more info!

**Starting Rsvp daemon (rsvpd) and using it to allocate resources for the source and the destination:**

**On Source:**

Starting rsvpd (and rtap):

Open a new Konsole or Terminal terminal and enter into it:

    cd RSVPmeres/rsvpd
    ./rsvpd.1 -D -R 193.168.2.254
    ENTER
    (help: '?' and view the files mentioned in the task with F3)
On Destination:
Starting rsvpd (and rtap):

Enter the terminal:

```
    cd RSVPmeres/rsvpd
    ./rsvpd.1 -D -R 193.168.1.254
```

b.)

Use iperf program to generate traffic. Start new terminals:

Enter into the terminal on the destination side (192.168.0.4):

```
    iperf -s
```

Enter into the source side (192.168.0.2):

```
    iperf -c 193.168.1.1 -i 2 -t 99999
    (we set 99999 to generate traffic for as long as possible so you do not have to worry about it stopping generation and starting the task from the beginning)
```

IMPORTANT: After this the ‘iperf’ prints the port used by the client, which is required later in the rsvpd to reserve resource for this traffic. Therefore, we cannot stop them during the experiment!

It is recommended to write rsvp commands into a text file (e.g. using Applications menu - Accessories - Mousepad) and from there copy them into the rsvpd – since the messages are coming continuously and there would not be enough time to enter them.

Step 1) On Source:

Creating session (must be done with the destination in succession before the PATH and the RESV messages):

```
    dest tcp 193.168.1.1/5001
```

Step 2) On Destination:

Creating Session:

```
    dest tcp 193.168.1.1/5001
```

Step 3) Again on the source:

Sending PATH:

```
    sender 193.168.2.1/port [t 120000 96000 120000 50000 160000]
    (you may have to wait 2-3 seconds in the destination until it prints the messages.)
```

Step 4) Again on the destination:

```
    sender
```
Then sending RESV:

```bash
reserve 193.168.1.1 ff 193.168.2.1/port [cl 120000 96000 120000 50000 160000]
```

If we see some ERROR messages in these terminals, then the configuration is not complete. The reason is we do not use RSVP API, thus it has failed to send PATH to Destination. To fix it, the following trick with _fake settings_ can be used:

First stop rsvpd in both sides, then start them again with _./rsvpd.1 -D -R_ ... command

**On Source:**

Creating session (must be done with the destination in succession before the PATH and the RESV messages):

```bash
dest tcp 193.168.2.1/5001
```

**On Destination:**

Creating Session:

```bash
dest tcp 193.168.2.1/5001
```

**Again on the source:**

Sending PATH:

```bash
sender 193.168.1.1/port [t 120000 96000 120000 50000 160000]
```

(you may have to wait 2-3 seconds in the destination until it prints the messages.)

We just switch the role of Source and Destination. Now the PATH message should be sent to Destination successfully. Stop rsvpd in both sides, then start them again with _./rsvpd.1 -D -R_ ... command and repeat Step 1-4!

Check RSVP settings again in routers to ensure that the resource allocation is success!

Now start another iperf process (multiple threads with -P switch) on the source side, you can see that these two share the bandwidth that remained after the first background traffic generation.

```bash
iperf -c 193.168.1.1 -i 2 -t 99999 -P 2
```

Breakdown of connection in rsvp daemon (on both sides):

```bash
close
```

Stop iperf:

```
CTRL + C
```

(If you want to work with new values, you have to disconnect both sides in rsvpd then stop the two-threaded iperf traffic. Then reserve the resources in rsvpd with the new values and start iperf on two threads)

(Comment:

```
// start of comment
```
If you worked with the following values (blue with the modified values compared to the previous one), then no the reservation conditions have changed, so the bandwidth depends on the 'r' value.

sender 193.168.2.1/port [t 120000 9600 170000 50000 160000]
reserve 193.168.1.1 ff 193.168.2.1/port [cl 120000 9600 170000 50000 160000]

End of comment )

If you used the following values, you got the following:

sender 193.168.2.1/port [t 170000 9600 170000 50000 160000]
reserve 193.168.1.1 ff 193.168.2.1/port [cl 170000 9600 170000 50000 160000]

This time the maximum bandwidth of the iperf running on thwo threads is 590Kbit/sec, that is 72KB, whereas the reserved bandwidth minimum is 1.38Mbit/sec on the first iperf (that we did not stop and do not have to until the end of the measurement), and checking it with minicom it is 1360Kbit/set, that is 170KB (170000 byte). Which means this is the reserved bandwidth, and other traffic may only use the remaining.

**Comparison of cases with QoS and without QoS:**

This can always stay the same.

(sender 193.168.2.1/port [t 150000 9600 150000 50000 160000])

Without Qos:

reserve 193.168.1.1 ff 193.168.2.1/port [cl 150000 9600 150000 50000 160000]

Now, we can see 1.05 Mbit/sec and 1.15Mbit/sec values that can appear because the reserved bandwidth is not utilized fully.

With QoS:

reserve 193.168.1.1 ff 193.168.2.1/port [g 180000 0 150000 9600 150000 50000 160000]

You can not see values below 1.38 Mbit/sec because you have set a minimum value of 180KB, whose speed must in any case be assured.

**Another comparison (with other values):**

This can always be the same.

(sender 193.168.2.1/port [t 130000 9600 130000 50000 160000])

Without Qos:

reserve 193.168.1.1 ff 193.168.2.1/port [cl 130000 9600 130000 50000 160000]

with QoS:

reserve 193.168.1.1 ff 193.168.2.1/port [g 180000 0 130000 9600 130000 50000 160000]
2. RSVP constructing/deconstructing process:

a, Document what RSVP messages have been going through the network when constructing/deconstructing it:

- if reservation requirements can be met
- if reservation requirements can not be met (e.g. br> 2 Mbit)

You can monitor traffic between rsvpd and CISCO routers with tcpdump, but rsvpd is also appropriate (it is worth using the log file generated by rsvpd: /var/tmp/rsvpd.log.*). The traffic between the two CISCO routers cannot be monitored in this measurement configuration, so you do not have to worry about them.

b, Measure the time between the Path and the corresponding Resv messages.

Tip: In order to have the rsvpd respond immediately with a Resv message for an incoming Path message you have to issue a properly parametrized ‘reserve` command before the Path message would arrive. In this case you may get a “no path information” message but once when the Path arrives it will reply with a Resv message.

Solution:

a.)

A.) (sender and reserve, and close-close commands):

During the construction, PATH and RESV messages went through the network, PATH messages were sent by the source towards the destination, RESV messages were sent from the destination towards the source.

During the deconstruction (with close), PTEAR and RTEAR messages passed through the network. PTEAR messages were from the source and RTEAR messages from the recipient.

(/var/tmp/rsvpd.log file contains the messages, but you may also use the status command in the rsvpd to see what kind of messages were received by the source and the destination).

(Due to the soft state mechanism, the PATH and RESV messages are constantly going through the network.)

B.)

PATH message is still sent by the source, but RESV message did not go through the network. Instead we received an error message RESV_ERR (rsvp error: resv: admission control failure) on the destination side.

Use the following parameters:

```
dest tcp 193.168.1.1/5001 (source)
dest tcp 193.168.1.1/5001 (destination)
sender 193.168.2.1/port [t 185000 96 000 185000 50000 185000 160 000] (source)
reserve 193.168.1.1 ff 193.168.2.1/port [cl 185000 96000 185000 50000 160000] (destination)
```
At this point, everything was fine, and the reservation was done. This is because when we enabled RSVP in the routers, we set the maximum reserved bandwidth (serial port) to 1500 Kbits/sec, that is 187 KB per second.

If you enter a value above 187, you will have trouble:

```
sender 193.168.2.1/port [t 190000 96000 185000 50 000 160 000] (source)
reserve 193.168.1.1 ff 193.168.2.1/port [cl 190000 96000 185000 50000 160000] (drain)
```

But we may also set a speed of more than 2 Mbps, and then we get an error every time, as the cable between the routers takes 2Mb of maximum speed per second.

b.)

Do the measurement on the source side by reading the data from /var/tmp/rsvpd.log. Before the examination start a session, then send a resv message first and only after that the path messages. When the resv message arrived, check with the `status` command and then closed it with the `close` command. Then check the file (look for something similar to the bold).

```
Register sender: 193.168.2.1/5001  T=[20(30) 50B/s 5 60]
193.168.2.1/5001  T=[20(30) 50B/s 5 60]
   Adspec( 0 hop InfBW 0us 65535B, G={br!}, CL={br!})
18:18:34.186 >>>>>>>>>>> Internal STATE: <<<<<<<< 85884 <<<<<<<<
Sender: 193.168.2.1/5001  PHOP: <(API) TTD: 243384
In_if 2=>API Outlist <1>  flags *PE ip_ttl 63
T=[20(30) 50B/s 5 60]  Adspec( 0 hop InfBW 0us 65535B, G={br!}, CL={br!})
------------------ End of Dest state dump ------------------
   In_if 2=>API Outlist <1>  flags *PE ip_ttl 63
T=[20(30) 50B/s 5 60]  Adspec( 1 hop 1.25MBW 0us 1500B, G={br!}, CL={br!})
4/255
   WF [ CL T=[20(30) 50B/s 5 60] ]
18:18:34.303|  AddFlow  eth1 193.168.1.1/5000[6]  Flg= =>handle=0
   flowspec= [ CL T=[20(30) 50B/s 5 60] ]
   Tspec=T=[20(30) 50B/s 5 60]
18:18:34.306|  AddFilt  eth1 193.168.1.1/5000[6]  Flg= =>handle=0
   Filter, Fhandle=1
18:18:34.312 >>>>>>>>> Internal STATE: <<<<<<<< 85938 <<<<<<<<
Sender: 193.168.2.1/5001  PHOP: <(API) TTD: 243384
In_if 2=>API Outlist <1>  flags *E ip_ttl 63
T=[20(30) 50B/s 5 60]  Adspec( 0 hop InfBW 0us 65535B, G={br!}, CL={br!})
WF Resv: Iface 1=>eth1 Nhop <193.168.2.254 LIH=1 TTD 243438
   Flowspec [ CL T=[20(30) 50B/s 5 60] ]
```
Kernel reservation: Iface 1 (193.168.2.1) Rhandle 0
  Flowspec [CL T=[20(30) 50B/s 5 60] ]
------------------- End of Dest state dump -------------------
18:18:42.432| Rcv Raw RESV-T EAR 193.168.1.1/5000[6] eth1<=1 <
193.168.2.254/255
  WF [CL T=[20(30) 50B/s 5 60] ]
  flowspec= [CL T=[20(30) 50B/s 5 60] ]
18:18:42.441| API Upc Resv Evt 193.168.1.1/5000[6] > API pid=5942 Asid=1
18:18:42.444 >>>>>>>>>>> Internal STATE: <<<<<<<< 94047 <<<<<<<<
  Sender: 193.168.2.1/5001 PHOP: <(API)> TTD: 243384
  In_if 2=>API Outlist <1> flags *E ip_ttl 63
  T=[20(30) 50B/s 5 60] Adspec( 0 hop InfBW 0us 65535B, G={br!}, CL={br!})
------------------- End of Dest state dump -------------------
18:18:44.734| APICls 193.168.1.1/5000[6] <API pid=5942 Asid=1
  193.168.2.1/5001 T=[20(30) 50B/s 5 60]
  Adspec( 0 hop InfBW 0us 65535B, G={br!}, CL={br!})
  193.168.2.1/5001 [ ]