Computer Networks Lab

Project Report

Submitted by:

Nadia Dessouky
900212701

Omar Anwar
900205016

Project GitHub Link

Contents

First Part: RTP and RTCP Protocols

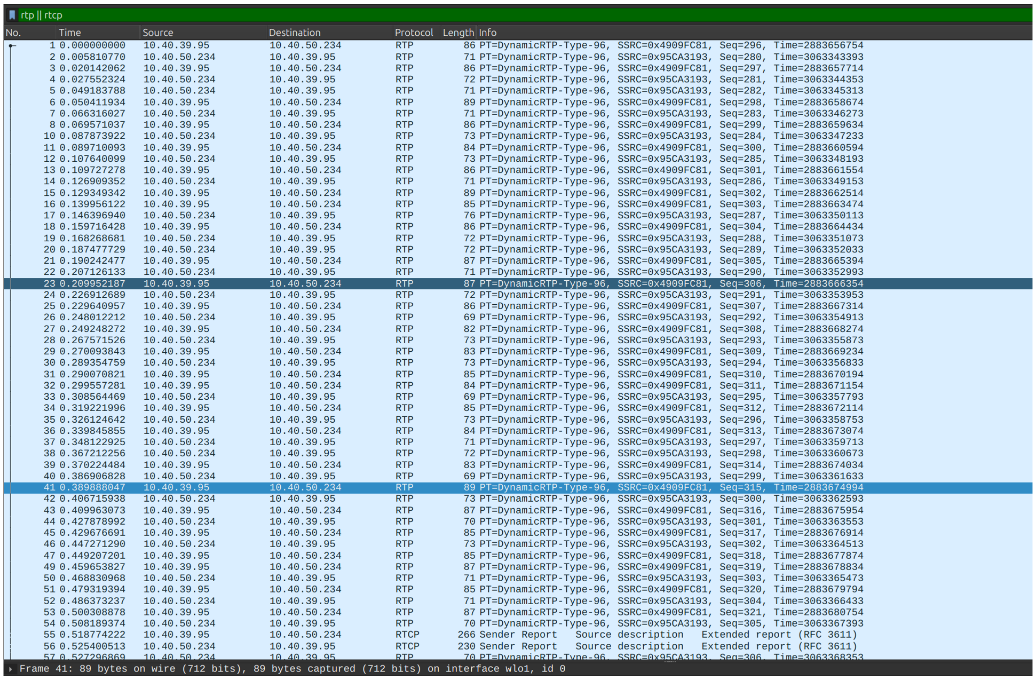
Required Questions:

1) *Explain the RTP and RTCP protocols in details.*

Created by the Internet Engineering Task Force (IETF), the Real-time Transport Protocol (RTP) is essential for transmitting audio and video streams across IP networks in real time. It is widely used in services such as video conferencing, live streaming, and Voice over IP (VoIP). RTP sits on top of the User Datagram Protocol (UDP), opting for fast delivery and low latency over reliability. To support real-time communication, RTP includes crucial functionalities like sequence numbering to preserve packet order, timestamps to enable synchronization, source identification, and payload type recognition. While RTP does not ensure packet delivery or prevent duplication, it is designed to work alongside the Real-time Transport Control Protocol (RTCP), which monitors network conditions, provides performance feedback, and helps align multiple media streams. By allowing some packet loss and avoiding retransmissions, RTP maintains smooth and timely playback, making it highly suitable for delay-sensitive applications.

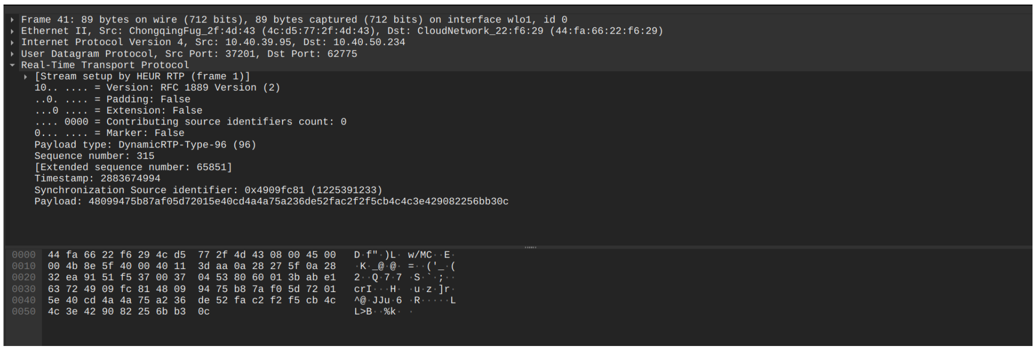
The Real-time Transport Control Protocol (RTCP), developed as a companion to the Real-time Transport Protocol (RTP), is responsible for overseeing and managing the quality of media transmission over IP networks. While RTP focuses on delivering the actual audio and video content, RTCP complements it by gathering performance metrics such as packet loss, delay variation (jitter), and network latency. This statistical feedback allows users and systems to evaluate connection health and adapt streaming behavior accordingly. RTCP transmits several types of control messages—such as Sender Reports (SR), Receiver Reports (RR), Source Description (SDES), Goodbye (BYE), and Application-Defined (APP) packets—that help synchronize media streams, identify participants, and manage session dynamics. Though RTCP does not include built-in security mechanisms like encryption or authentication, it is often deployed in conjunction with protocols such as Secure RTP (SRTP) to protect media data. By collaborating with RTP, RTCP plays a vital role in delivering smooth, synchronized, and high-quality real-time media for applications like live streaming and video conferencing.

2) *Filter the RTP and RTCP packets in Wireshark.*



3) *Identify the payload type, whether audio or video.*

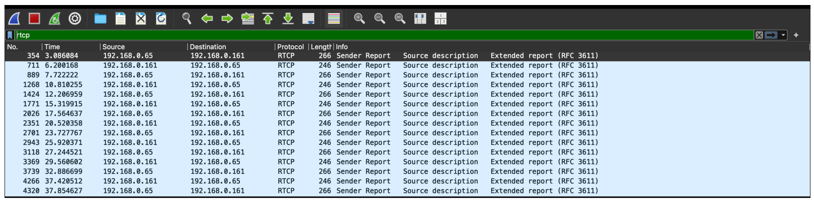
From the image below, it can be seen that the payload type is 96, which could be video or audio. However, the payload size is 32 bytes, which is small for video so it must be audio. This aligns with what we did as we did an audio call.



4) *Inspect the following different reports used by RTCP:*

- *Sender Report (SR):*

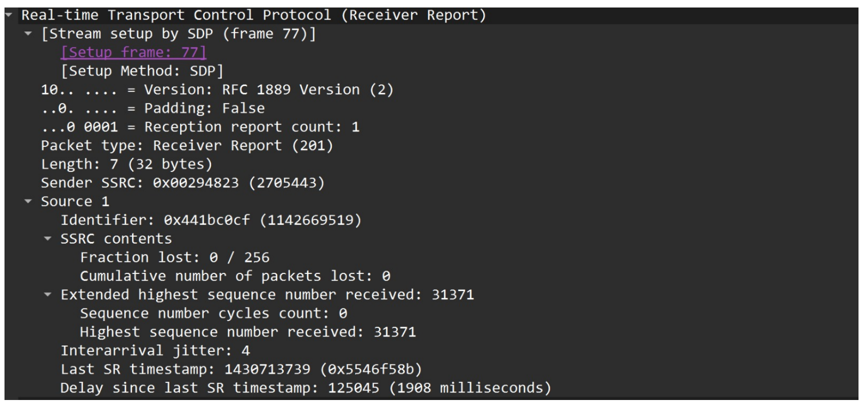
  Provides transmission and reception statistics from active senders, including NTP and RTP timestamps, useful for synchronization and delay calculations.



No.	Time	Source	Destination	Protocol	Length	Info
354	3.086084	192.168.0.65	192.168.0.161	RTCP	266	Sender Report Source description Extended report (RFC 3611)
711	6.200168	192.168.0.161	192.168.0.65	RTCP	246	Sender Report Source description Extended report (RFC 3611)
889	7.722222	192.168.0.65	192.168.0.161	RTCP	266	Sender Report Source description Extended report (RFC 3611)
1268	10.810255	192.168.0.161	192.168.0.65	RTCP	246	Sender Report Source description Extended report (RFC 3611)
1424	12.206959	192.168.0.65	192.168.0.161	RTCP	266	Sender Report Source description Extended report (RFC 3611)
1771	15.319915	192.168.0.161	192.168.0.65	RTCP	246	Sender Report Source description Extended report (RFC 3611)
2026	17.564637	192.168.0.65	192.168.0.161	RTCP	266	Sender Report Source description Extended report (RFC 3611)
2351	20.520358	192.168.0.161	192.168.0.65	RTCP	246	Sender Report Source description Extended report (RFC 3611)
2701	23.727767	192.168.0.65	192.168.0.161	RTCP	266	Sender Report Source description Extended report (RFC 3611)
2943	25.920371	192.168.0.161	192.168.0.65	RTCP	246	Sender Report Source description Extended report (RFC 3611)
3118	27.244521	192.168.0.65	192.168.0.161	RTCP	266	Sender Report Source description Extended report (RFC 3611)
3369	29.560602	192.168.0.161	192.168.0.65	RTCP	246	Sender Report Source description Extended report (RFC 3611)
3739	32.886699	192.168.0.65	192.168.0.161	RTCP	266	Sender Report Source description Extended report (RFC 3611)
4266	37.420512	192.168.0.161	192.168.0.65	RTCP	246	Sender Report Source description Extended report (RFC 3611)
4320	37.854627	192.168.0.65	192.168.0.161	RTCP	266	Sender Report Source description Extended report (RFC 3611)

- *Receiver Report (RR):*

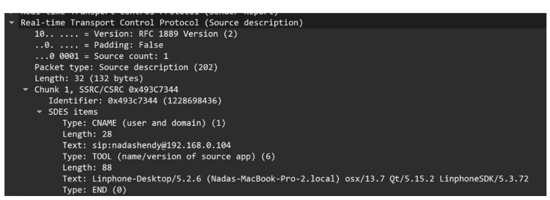
  Sent by participants not actively sending media, it includes statistics like packet loss, jitter, and feedback on sender streams.



```
Real-time Transport Control Protocol (Receiver Report)
  [Stream setup by SDP (frame 77)]
    [Setup frame: 77]
    [Setup Method: SDP]
  10.. .... = Version: RFC 1889 Version (2)
  ..0. .... = Padding: False
  ...0 0001 = Reception report count: 1
  Packet type: Receiver Report (201)
  Length: 7 (32 bytes)
  Sender SSRC: 0x00294823 (2705443)
  Source 1
    Identifier: 0x441bc0cf (1142669519)
    SSRC contents
      Fraction lost: 0 / 256
      Cumulative number of packets lost: 0
    Extended highest sequence number received: 31371
      Sequence number cycles count: 0
      Highest sequence number received: 31371
    Interarrival jitter: 4
    Last SR timestamp: 1430713739 (0x5546f58b)
    Delay since last SR timestamp: 125045 (1908 milliseconds)
```

- *Source Description (SDES):*

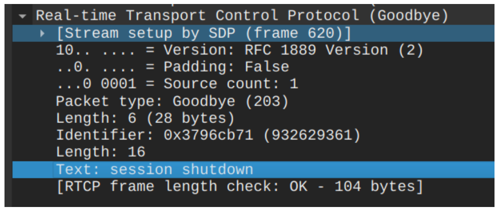
  Carries metadata about participants, such as canonical names (CNAMEs), to help identify sources of RTP streams.

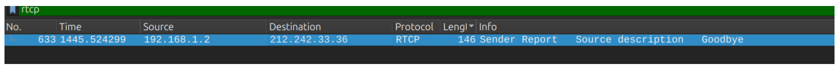


```
Real-time Transport Control Protocol (Source description)
  10.. .... = Version: RFC 1889 Version (2)
  ..0. .... = Padding: False
  ...0 0001 = Source count: 1
  Packet type: Source description (202)
  Length: 32 (132 bytes)
  Chunk 1, SSRC/CSRC 0x493C7344
    Identifier: 0x493c7344 (1228698436)
    SDES items
      Type: CNAME (user and domain) (1)
      Length: 28
      Text: sip:nadashendy@192.168.0.104
      Type: TOOL (name/version of source app) (6)
      Length: 88
      Text: Linphone-Desktop/5.2.6 (Nadas-MacBook-Pro-2.local) osx/13.7 Qt/5.15.2 LinphoneSDK/5.3.72
      Type: END (0)
```

- *End of Participation (BYE):*

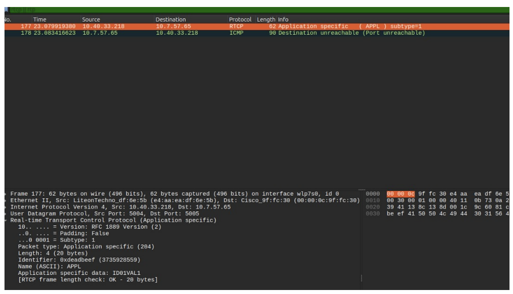
  Signals that a participant is leaving the RTP session, allowing other participants to update session state accordingly.





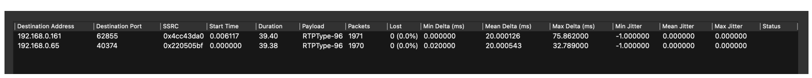
- *Application Specific (APP):*

  Used for custom, experimental, or proprietary messages that are not defined by standard RTCP packet types.



5) *Examine and compute the end-to-end (E2E) delays.*

This is the packet stream analysis:



Destination Address	Destination Port	SSRC	Start Time	Duration	Payload	Packets	Lost	Min Delta (ms)	Mean Delta (ms)	Max Delta (ms)	Min Jitter	Mean Jitter	Max Jitter	Status
192.168.0.161	62855	0x4cc43da0	0.006117	39.40	RTPType-96	1971	0 (0.0%)	0.000000	20.000126	75.862000	-1.000000	0.000000	0.000000	
192.168.0.65	40374	0x220505bf	0.000000	39.38	RTPType-96	1970	0 (0.0%)	0.020000	20.000543	32.789000	-1.000000	0.000000	0.000000	

An Alternative Solution:

From the RTCP Receiver Report and Sender Reports above:

- Last SR Timestamp (LSR): 1438701737
- Delay Since Last SR (DLSR): 125045, which corresponds to 1908 milliseconds

The round-trip time (RTT) can be estimated using the formula:

$$\text{RTT} = A - \text{LSR} - \text{DLSR}$$

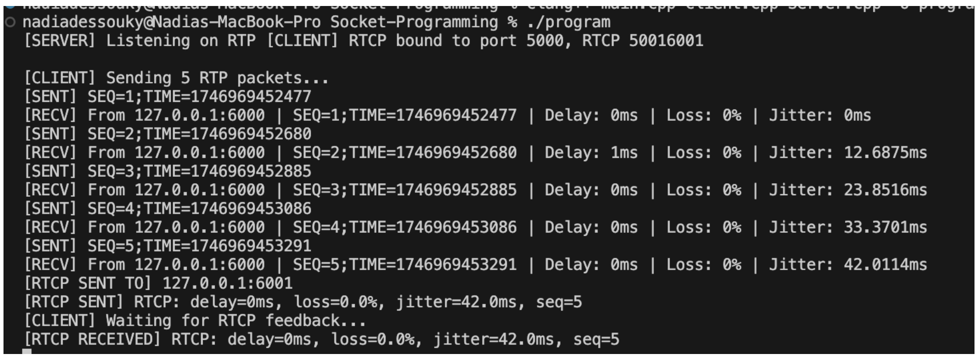
$$\text{End-to-End Delay} \approx \frac{\text{RTT}}{2} \approx \frac{1908}{2} = 954 \text{ ms}$$

6) *VoIP Test Between VM1 and VM2 Using Cisco IP Communicator*

To perform a VoIP test between two virtual machines (VM1 and VM2) using Cisco IP Communicator, the following steps are carried out: To test VoIP communication, install and set up Cisco IP Communicator on both virtual machines (VM1 and VM2), making sure they were connected to the same network and could call each other either through a call controller or direct IP dialing. Open Wireshark on the correct network interface to capture all traffic between the two VMs. On VM1,initiate a call to VM2, which should be answered and kept active briefly while speaking to generate RTP traffic. After ending the call, stop the packet capture for analysis. The recorded data showed important VoIP components such as SIP signaling messages (like INVITE, 180 Ringing, 200 OK, ACK, and BYE) used to set up and end the call. Once the call began, we observed RTP packets carrying the audio in both directions between the VMs. In addition, RTCP packets—used for monitoring call quality—were present, including sender and receiver reports that gave information about packet loss, jitter, and timing. Finally, we could estimate end-to-end delay by looking at RTP packet timestamps or using RTCP data to assess how long it took for audio data to travel across the network.

Socket Programming for RTP and RTCP Protocols

GitHub Link



```
nadiadessouky@Nadias-MacBook-Pro Socket-Programming % ./program
[SERVER] Listening on RTP [CLIENT] RTCP bound to port 5000, RTCP 50016001

[CLIENT] Sending 5 RTP packets...
[SENT] SEQ=1;TIME=1746969452477
[RECV] From 127.0.0.1:6000 | SEQ=1;TIME=1746969452477 | Delay: 0ms | Loss: 0% | Jitter: 0ms
[SENT] SEQ=2;TIME=1746969452680
[RECV] From 127.0.0.1:6000 | SEQ=2;TIME=1746969452680 | Delay: 1ms | Loss: 0% | Jitter: 12.6875ms
[SENT] SEQ=3;TIME=1746969452885
[RECV] From 127.0.0.1:6000 | SEQ=3;TIME=1746969452885 | Delay: 0ms | Loss: 0% | Jitter: 23.8516ms
[SENT] SEQ=4;TIME=1746969453086
[RECV] From 127.0.0.1:6000 | SEQ=4;TIME=1746969453086 | Delay: 0ms | Loss: 0% | Jitter: 33.3701ms
[SENT] SEQ=5;TIME=1746969453291
[RECV] From 127.0.0.1:6000 | SEQ=5;TIME=1746969453291 | Delay: 0ms | Loss: 0% | Jitter: 42.0114ms
[RTCP SENT TO] 127.0.0.1:6001
[RTCP SENT] RTCP: delay=0ms, loss=0.0%, jitter=42.0ms, seq=5
[CLIENT] Waiting for RTCP feedback...
[RTCP RECEIVED] RTCP: delay=0ms, loss=0.0%, jitter=42.0ms, seq=5
```

Second Part: HSRP Protocol Configuration

HSRP Configuration

Using Cisco's Packet Tracer network simulation tool, build the given network topology and apply the following configurations: (Include all necessary screenshots in your report)

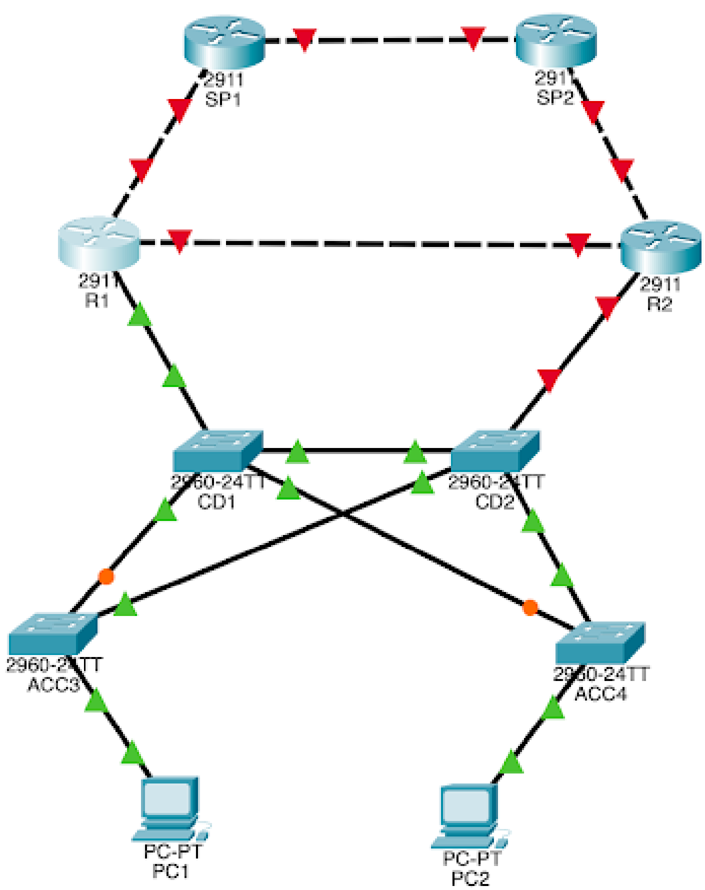


Figure 1: Network Topology for HSRP Configuration

Basic HSRP

(a) **Configure basic HSRP for the 10.10.10.0/24 network using the IP addresses shown in the topology diagram.**

R1:

```
R1(config)#interface GigabitEthernet0/1
R1(config-if)#ip address 10.10.10.2 255.255.255.0
R1(config-if)#standby 1 ip 10.10.10.1
R1(config-if)#no shutdown
```

Figure 2: R1 HSRP Configuration

R2:

```
R2(config)#interface GigabitEthernet0/1
R2(config-if)#ip address 10.10.10.3 255.255.255.0
R2(config-if)#standby 1 ip 10.10.10.1
R2(config-if)#no shutdown
```

Figure 3: R2 HSRP Configuration

(b) **Wait for HSRP to come up on both routers and then check which is the active router.**

R1:

```
R1#show standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp  Pri P State    Active          Standby         Virtual IP
Gig0/1      1    100   Standby  10.10.10.3      local           10.10.10.1
```

Figure 4: HSRP Status on R1

R2:

```
R2#show standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp  Pri P State    Active          Standby         Virtual IP
Gig0/1      1    100   Active   local           10.10.10.2      10.10.10.1
```

Figure 5: HSRP Status on R2

(c) **Verify that the PCs can ping their default gateway using the HSRP address 10.10.10.1.**

PC1:

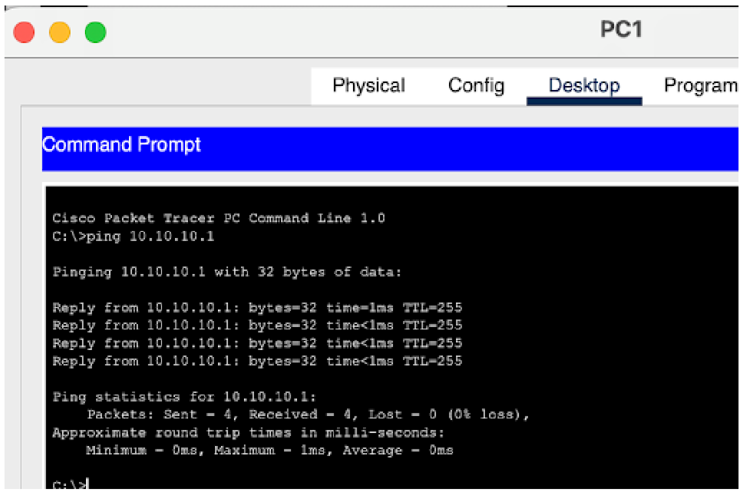


Figure 6: PC1 Pinging Default Gateway

PC2:

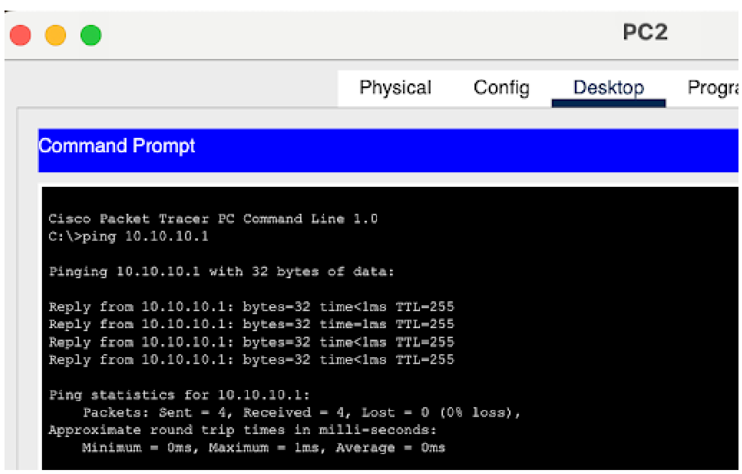


Figure 7: PC2 Pinging Default Gateway

(d) **Verify that the PCs have upstream connectivity via their HSRP default gateway. Ping SP1 at 203.0.113.1**

PC1:

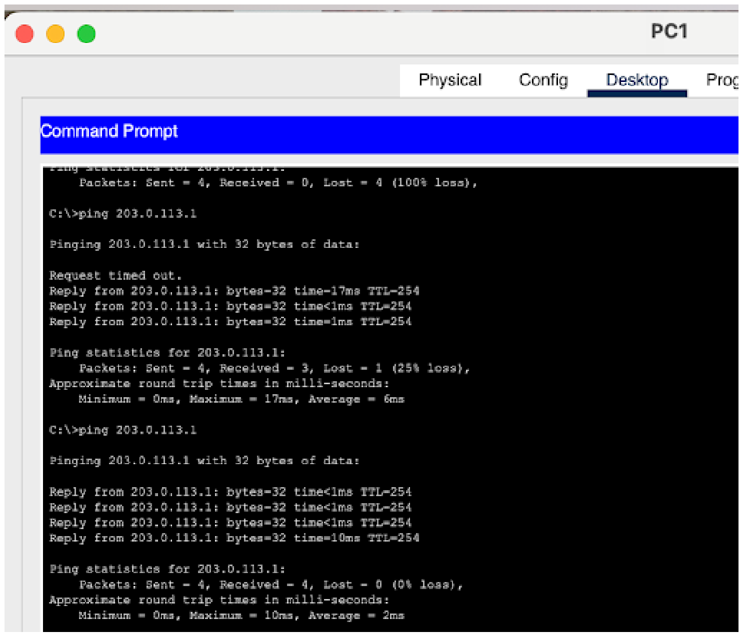


Figure 8: PC1 Pinging SP1

PC2:

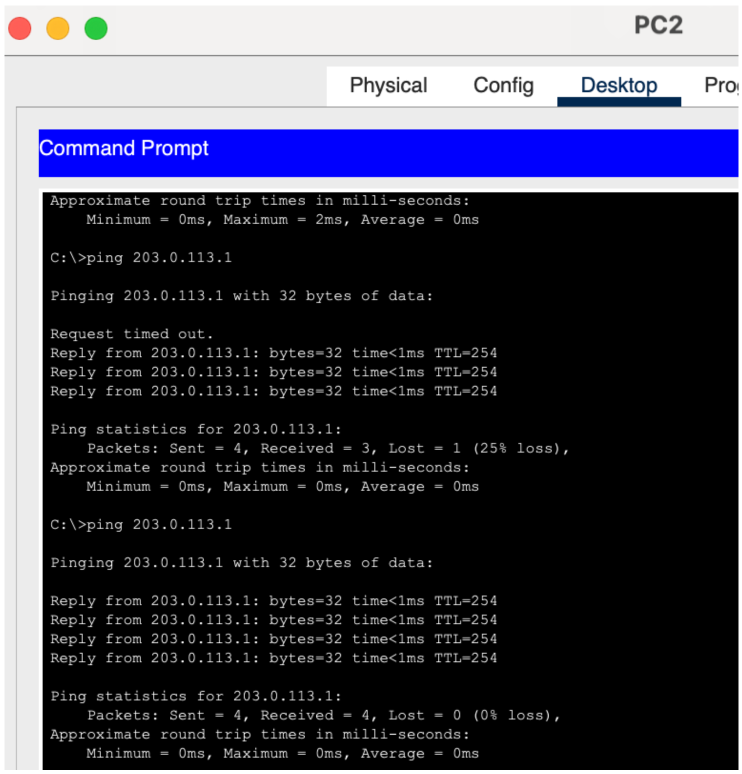


Figure 9: PC2 Pinging SP1

(e) **What is the MAC address on the physical interface of the active router?**

MAC address: 000a.f31c.2102

```
R2#show interface g0/1
GigabitEthernet0/1 is up, line protocol is up (connected)
  Hardware is CN Gigabit Ethernet, address is 000a.f31c.2102 (bia 000a.f31c.2102)
  Internet address is 10.10.10.3/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s, media type is RJ45
  output flow-control is unsupported, input flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00,
  Last input 00:00:08, output 00:00:05, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/40 (size/max)
  5 minute input rate 96 bits/sec, 0 packets/sec
  5 minute output rate 94 bits/sec, 0 packets/sec
     1979 packets input, 62133 bytes, 0 no buffer
     Received 34 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 watchdog, 1017 multicast, 0 pause input
     0 input packets with dribble condition detected
     1964 packets output, 60292 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

Figure 10: Mac Address of interface g0/1

(f) **What is the MAC address of the HSRP virtual interface?**

The HSRP virtual interface is using the MAC address: 0000.0C07.AC01

```
R2#show standby
GigabitEthernet0/1 - Group 1
  State is Active
    6 state changes, last state change 02:17:24
  Virtual IP address is 10.10.10.1
  Active virtual MAC address is 0000.0C07.AC01
    Local virtual MAC address is 0000.0C07.AC01 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 0.014 secs
  Preemption disabled
  Active router is local
  Standby router is 10.10.10.2
  Priority 100 (default 100)
  Group name is hsrp-Gig0/1-1 (default)
```

Figure 11: HSRP virtual interface g0/1

(g) **Verify the PCs are using the virtual MAC address for their default gateway.**

The default gateway for both PC2 and PC1 as we can see is 10.10.10.1:
PC1:

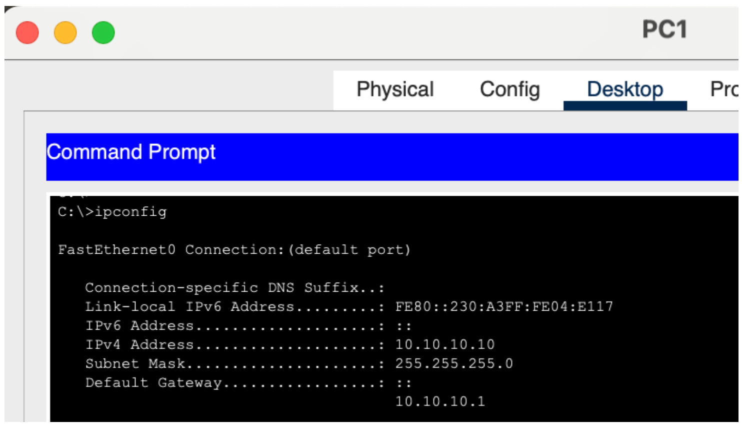


Figure 12: PC1 Default Gateway Configuration

PC2:

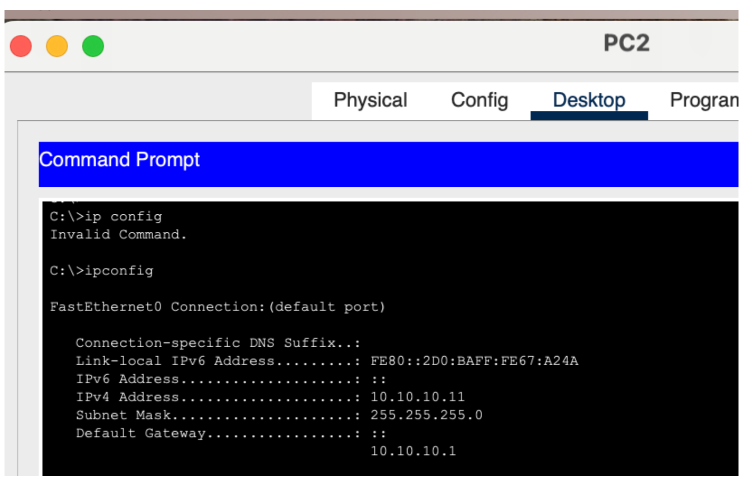


Figure 13: PC2 Default Gateway Configuration

Checking The arp cache to verify the default gateway uses the virtual MAC address for the gateway:

PC1:

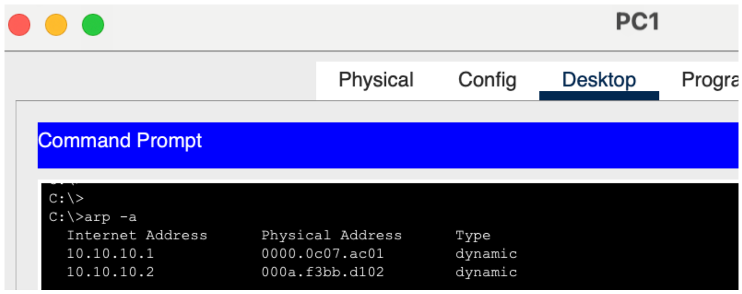


Figure 14: PC1 ARP Cache

PC2:

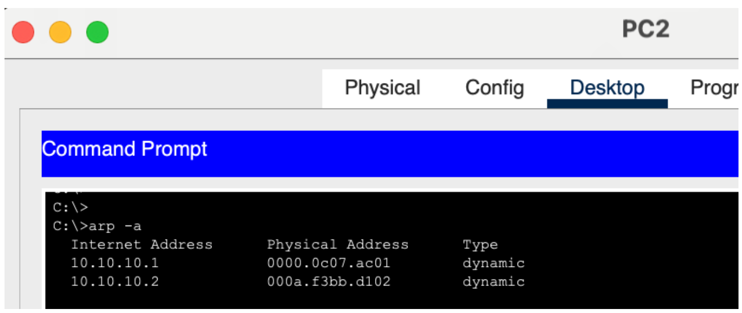


Figure 15: PC2 ARP Cache

Priority and Pre-emption

(a) **Configure HSRP so that R1 will be the preferred router. Use a single command.**

We will make R1 be the preferred router by increasing its priority. We will make R1 have priority 150 and R2 have priority 100 (it is the default one).

```
R1(config)#interface GigabitEthernet0/1
R1(config-if)#standby 1 priority 150
R1(config-if)#exit
```

Figure 16: Configuring R1 Priority

(b) **Which router do you expect will be active now? Verify this.**

Even though we raised R1's priority above R2's, R2 will stay active because, by default, HSRP doesn't allow a higher-priority router to take over unless you explicitly enable preemption. Moreover, without the standby 1 preempt command on R1, the router that first became active (R2) remains in control even after R1's priority is increased.

R1:

```
R1#show standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp  Pri P State    Active          Standby         Virtual IP
Gig0/1      1    150   Standby  10.10.10.3      local           10.10.10.1
```

Figure 17: R1 Status After Priority Change

R2:

```
R2#show standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp  Pri P State    Active          Standby         Virtual IP
Gig0/1      1    100   Active   local           10.10.10.2      10.10.10.1
```

Figure 18: R2 Status After Priority Change

(c) **Ensure that R1 is the active router. Do not reboot.**

Doing preempt to ensure that R1 is the active router without rebooting:

```
R1(config)#interface GigabitEthernet0/1
R1(config-if)#standby 1 preempt
```

Figure 19: R1 Preempt Configuration

R1:

```
%HSRP-6-STATECHANGE: GigabitEthernet0/1 Grp 1 state Standby -> Active


R1#show standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp  Pri P State    Active          Standby         Virtual IP
Gig0/1      1    150 P Active   local           10.10.10.2      10.10.10.1
```

Figure 20: R1 Status After Preempt

R2:

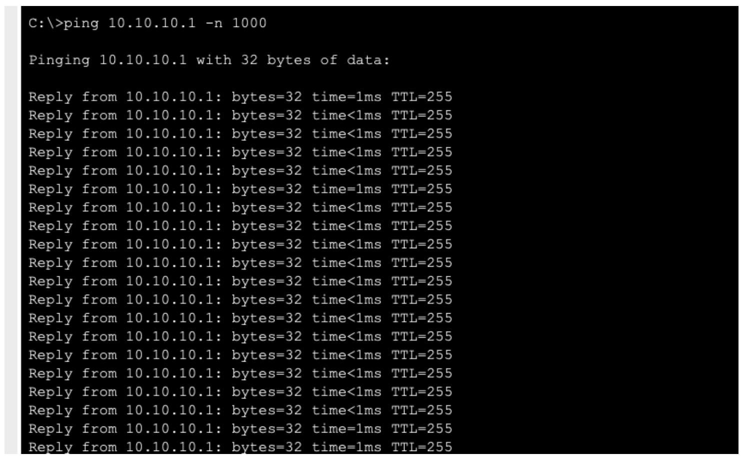
```
R2#
%HSRP-6-STATECHANGE: GigabitEthernet0/1 Grp 1 state Speak -> Standby
show standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp  Pri P State    Active          Standby         Virtual IP
Gig0/1      1    100   Standby  10.10.10.2      local           10.10.10.1
```

Figure 21: R2 Status After R1 Preempt

Test HSRP

(a) **Run a continuous ping to the HSRP IP address from PC1 with the ping 10.10.10.1 -n 1000 command.**

Here was the output without rebooting, all pings are received:

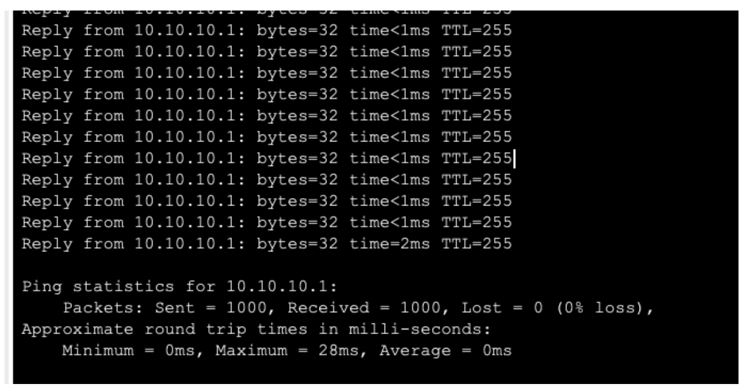


```
C:\>ping 10.10.10.1 -n 1000

Pinging 10.10.10.1 with 32 bytes of data:

Reply from 10.10.10.1: bytes=32 time=1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time=1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time=1ms TTL=255
Reply from 10.10.10.1: bytes=32 time=1ms TTL=255
```

Figure 22: PC1 Continuous Ping



```
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time=2ms TTL=255

Ping statistics for 10.10.10.1:
    Packets: Sent = 1000, Received = 1000, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 28ms, Average = 0ms
```

Figure 23: PC1 Continuous Ping Continued

(b) **Save the configuration on R1, then reboot.**

I will run the ping again and reboot: it will be seen in the answer to the two following questions.

```
R1#wr
Building configuration...
[OK]
R1#reload
Proceed with reload? [confirm]
System Bootstrap, Version 15.1(4)M4, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 2010 by cisco Systems, Inc.
Total memory size = 512 MB - On-board = 512 MB, DIMM0 = 0 MB
CISCO2911/K9 platform with 524288 Kbytes of main memory
Main memory is configured to 72/-1(On-board/DIMM0) bit mode with ECC disabled

Readonly ROMMON initialized

program load complete, entry point: 0x80803000, size: 0x1b340
program load complete, entry point: 0x80803000, size: 0x1b340

IOS Image Load Test
___________________
Digitally Signed Release Software
program load complete, entry point: 0x81000000, size: 0x3bcd3d8
Self decompressing the image :
########################################################################## [OK]
Smart Init is enabled
smart init is sizing iomem
                  TYPE      MEMORY_REQ
     Onboard devices &
          buffer pools      0x022F6000
-----------------------------------------------
                TOTAL:      0x022F6000
Rounded IOMEM up to: 36Mb.
Using 6 percent iomem. [36Mb/512Mb]

              Restricted Rights Legend
```

Figure 24: R1 Reboot

(c) **View the ping output on PC1. You should see a few dropped pings as R2 transitions to active following the outage of R1.**

```
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time=2ms TTL=255
Reply from 10.10.10.1: bytes=32 time=1ms TTL=255
Request timed out.
Request timed out.
Request timed out.
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time=1ms TTL=255
```

Figure 25: PC1 Ping During Router Failover

There is successful replies from 10.10.10.1, then three "Request timed out" lines, then replies again. Those timeouts line up exactly with when R1 went down and R2 moved into the Active role. Here's a brief window where no router answers the virtual IP, so a few ICMP packets are lost before R2 finishes the HSRP transition and begins replying.

(d) **Verify R2 has transitioned to HSRP active.**

Immediately after R1 fails, R2 logs: STATECHANGE: Standby → Active Standby=unknown shows that R2 does not have standby router now

```
%HSRP-6-STATECHANGE: GigabitEthernet0/1 Grp 1 state Standby -> Active

R2#
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2, changed state to up
show standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp  Pri P State    Active          Standby         Virtual IP
Gig0/1      1    100   Active   local           unknown         10.10.10.1
```

Figure 26: R2 Status after transtion

(e) **Wait for R1 to complete booting and HSRP to come up. Verify R1 transitions to HSRP active because pre-emption is enabled.**

Speak → Standby: As soon as its interface came back up, R1 started sending HSRP hellos (”Speak”) and heard R2’s hellos. It then moved into the Standby state because R2 was still the Active router at that moment.

Standby → Active: Because we had enabled preemption and given R1 a higher priority (150), R1 immediately claimed the Active role once it realized it was eligible. That transition is logged as:

```
%HSRP-6-STATECHANGE: GigabitEthernet0/1 Grp 1 state Speak -> Standby

%HSRP-6-STATECHANGE: GigabitEthernet0/1 Grp 1 state Standby -> Active
```

Figure 27: R1 Reclaiming Active Role

Right after R1 came back up, its very first show standby brief showed Standby = unknown because it hadn’t yet heard R2’s hellos so it knew only that it itself was Active (local), but not yet who its backup was.

```
R1>enable
R1#show standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp  Pri P State    Active          Standby         Virtual IP
Gig0/1      1    150 P Active   local           unknown         10.10.10.1
```

Figure 28: R1 Standby Status in Transition

Once R1 and R2 exchanged HSRP hellos:

R1 saw R2 advertising itself in Group 1

R1 recorded R2’s real IP (10.10.10.3) as the Standby

Now show standby brief shows Standby = 10.10.10.3

```
R1#show standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp  Pri P State    Active          Standby         Virtual IP
Gig0/1      1    150 P Active   local           10.10.10.3      10.10.10.1
```

Figure 29: R1 Final HSRP Status

(f) **Hit 'Ctrl-C' to cancel the ping on PC1. If you scroll back, you should see a dropped ping or two as R1 transitioned back to HSRP active.**

```
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time=2ms TTL=255
Reply from 10.10.10.1: bytes=32 time=1ms TTL=255
Request timed out.
Request timed out.
Request timed out.
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time=1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time=1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time=1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Reply from 10.10.10.1: bytes=32 time=1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
```

Figure 30: Effect of Dropped Pings During Second Transition (Part 1)

First gap of "Request timed out" → When we rebooted R1, R2 had to take over as Active. During that switchover, we lost a few pings.

Second gap of timeouts further down → after R1 finished booting, its higher priority + preemption kicked in and it reclaimed the Active role, causing another brief interruption.

Here is also a screenshot of the effect of the dropped pings:

```
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255
Reply from 10.10.10.1: bytes=32 time<1ms TTL=255

Ping statistics for 10.10.10.1:
    Packets: Sent = 1000, Received = 987, Lost = 13 (2% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 18ms, Average = 0ms
```

Figure 31: Effect of Dropped Pings During Second Transition (Part 2)

References

[1] GeeksforGeeks, "Real time transport protocol (RTP)," GeeksforGeeks, https://www.geeksforgeeks.org/real-time-transport-protocol-rtp/ (accessed May 6, 2025).

[2] GeeksforGeeks, "Real-time transport control protocol (RTCP)," GeeksforGeeks, https://www.geeksforgeeks.org/real-time-transport-control-protocol-rtcp/ (accessed May 6, 2025).