

Chapter 4 Lab Work: Communication

Nguyen Ngoc Lam - 20162316

Tuesday 12th May, 2020

Contents

- 1 What is the code part that shows that the Server assigns the correlation ID to the response?
- 2 You base on both code of Client and Server program to explain which code shows that the Client sends request to Server through `rpc_queue` and create a new queue to wait for the reply of the Server.
- 3 Try to add the delay to the Server program
- 4 Important note for the video streaming part.
- 5 What is the IP address of your 2 machines? How to ping each other?
- 6 Can you watch the video in the client machine? Evaluate the quality of the video streaming service.
- 7 What is the result of the ping test? Can you see an increase of 100 milliseconds?
- 8 Disable the buffering function of VLC in Client machine. Then, evaluate the video quality at the Client machine. How can you conclude the impact of fix delay on video streaming service?
- 9 Evaluate the video quality at the Client machine. How can you conclude the impact of delay variation on video streaming service?
- 10 Evaluate the video quality at the Client machine. How can you conclude the impact of fix loss rate on video streaming service? Try to increase the value of loss rate to see the impact more clear.
- 11 Evaluate the video quality at the Client machine. How can you conclude the impact of loss rate variation on video streaming service? Try to increase this value to see the impact more clear.
- 12 Evaluate the video quality at the Client machine. How can you conclude the impact of packet duplication on video streaming service? Try to increase this value to see the impact more clear.

13 Evaluate the video quality at the Client machine. How can you conclude the impact of packet corruption on video streaming service?

1 What is the code part that shows that the Server assigns the correlation ID to the response?

```
channel.queueDeclare(RPC_QUEUE_NAME, false, false, false, null);
channel.queuePurge(RPC_QUEUE_NAME);
channel.basicQos(1);
System.out.println(" [x] Awaiting RPC requests");
Object monitor = new Object();
DeliverCallback deliverCallback = (consumerTag, delivery) -> {
    AMQP.BasicProperties replyProps = new AMQP.BasicProperties.Builder()
        .correlationId(delivery.getProperties().getCorrelationId()).build();
    String response = "";
    try {
```

Figure 1: Assigning correlation ID

The part the Server assigns the correlation ID is highlighted in the figure

1

2 You base on both code of Client and Server program to explain which code shows that the Client sends request to Server through rpc_queue and create a new queue to wait for the reply of the Server.

<pre>ConnectionFactory factory = new ConnectionFactory(); factory.setUri(uri); try (Connection connection = factory.newConnection(); Channel channel = connection.createChannel()) { channel.queueDeclare(RPC_QUEUE_NAME, false, false, false, null); channel.queuePurge(RPC_QUEUE_NAME); channel.basicQos(1); System.out.println(" [x] Awaiting RPC requests"); Object monitor = new Object(); DeliverCallback deliverCallback = (consumerTag, delivery) -> { AMQP.BasicProperties replyProps = new AMQP.BasicProperties.Builder() .correlationId(delivery.getProperties().getCorrelationId()).build(); String response = ""; try { String message = new String(delivery.getBody(), "UTF-8"); int n = Integer.parseInt(message); System.out.println(" [x] Task: " + message + " [x]"); response = ""; try { Thread.sleep(1000); } catch (InterruptedException e) { e.printStackTrace(); } finally { channel.basicPublish("", delivery.getProperties().getReplyTo(), replyProps, response.getBytes("UTF-8")); channel.basicAck(delivery.getProperties().getDeliveryTag(), false); } // Finally the consumer must send back the RPC server some thread synchronized (monitor) { monitor.notify(); } } catch (Exception e) { e.printStackTrace(); } }; channel.consume(RPC_QUEUE_NAME, deliverCallback, true); }</pre>	<pre>public RPCClient() throws IOException, TimeoutException { final String uri = URI.create("amqp://localhost"); ConnectionFactory factory = new ConnectionFactory(); factory.setUri(uri); Connection connection = factory.newConnection(); channel = connection.createChannel(); public String call(String message) throws IOException, InterruptedException { final String corrId = UUID.randomUUID().toString(); String replyQueueName = channel.queueDeclare().getQueue(); AMQP.BasicProperties props = new AMQP.BasicProperties.Builder().correlationId(corrId).replyTo(replyQueueName).build(); channel.basicPublish("", "rpc_queue", props, message.getBytes("UTF-8")); final BlockResponseHandler response = new ArrayBlockingQueue<>(); String reply = channel.waitForResponse(replyQueueName, props, (consumerTag, delivery) -> { String response = new String(delivery.getBody(), "UTF-8"); response.offered(response.getBytes("UTF-8")); }); consumerTag -> { String result = response.toString(); channel.basicAck(deliveryTag, true); return result; } } }</pre>
---	--

(a) Receive request and response to request Server side

(b) Sending request, create a new queue to wait the reply on Client side

Figure 2: Client sends request to Server through rpc_queue

3 Try to add the delay to the Server program

The response from Server will be delayed thus making the Client waits for the response from previous request longer.

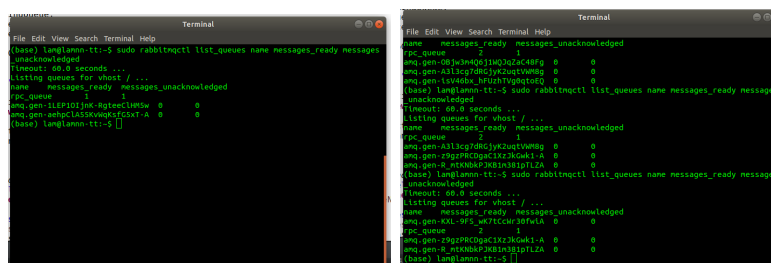
```

[x] Requesting fib(12)
[.] Got '144'
[x] Requesting fib(13)
[.] Got '233'
[x] Requesting fib(14)
[.] Got '377'
[x] Requesting fib(15)
[.] Got '610'
[x] Requesting fib(16)
[.] Got '987'
[x] Requesting fib(17)
[.] Got '1597'
[x] Requesting fib(18)
[.] Got '2584'
[x] Requesting fib(19)
[.] Got '4181'
[x] Requesting fib(20)
[.] Got '6765'
[x] Requesting fib(21)
[.] Got '10946'
[x] Requesting fib(22)
[.] Got '17711'
[x] Requesting fib(23)
[.] Got '28657'
[x] Requesting fib(24)
[.] Got '46368'
[x] Requesting fib(25)
[.] Got '75025'
[x] Requesting fib(26)
[.] Got '121393'
[x] Requesting fib(27)
[.] Got '196418'
[x] Requesting fib(28)
[.] Got '317811'
[x] Requesting fib(29)
[.] Got '514229'
[x] Requesting fib(30)
[.] Got '832040'
[x] Requesting fib(31)
[.] Got '1346269'

```

Figure 3: Client Result

When checking for queues, for n instances of Client are running, there are n queues for each instances plus the `rpc_queue`. At all time, there are $n-1$ ready messages and 1 unacknowledged messages at `rpc_queue` since the number on `ArrayBlockingQueue` equals to 1. The message on n unique queues for n instances will be transfered to `rpc_queue` very fast thus it could be said that the ready messages and unacknowledged messages number equals to 0.



(a) When there are 2 instances of Client (b) When there are 3 instances of Client

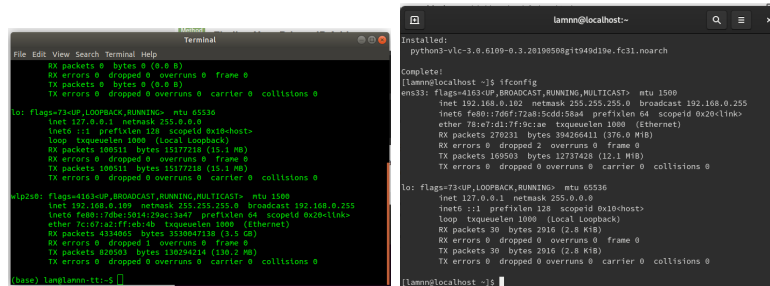
Figure 4: Status of queues

4 Important note for the video streaming part.

1. For this part, I will use my own laptop and my home computer connected over the home network.

5 What is the IP address of your 2 machines? How to ping each other?

The server will have the address: 192.168.0.109 The client will have the address: 192.168.0.102



The image shows two terminal windows side-by-side. The left window, titled 'Terminal', displays the output of the 'ifconfig' command for the 'lo' and 'eth0' interfaces. The 'lo' interface has the IP address 127.0.0.1, and the 'eth0' interface has the IP address 192.168.0.109. The right window, titled 'lamn@localhost:~', shows the output of the 'ifconfig' command for the 'lo' and 'eth0' interfaces. The 'lo' interface has the IP address 127.0.0.1, and the 'eth0' interface has the IP address 192.168.0.102.

```
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 108511 bytes 15177218 (15.1 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 108511 bytes 15177218 (15.1 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.109 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::70be:5014:29ac:1a47 prefixlen 64 scopeid 0x20<link>
    ether 78:e7:d1:7f:9c:ae txqueuelen 1000 (Ethernet)
    RX packets 276231 bytes 394266411 (376.0 MiB)
    RX errors 0 dropped 2 overruns 0 frame 0
    TX packets 169503 bytes 12737428 (12.1 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

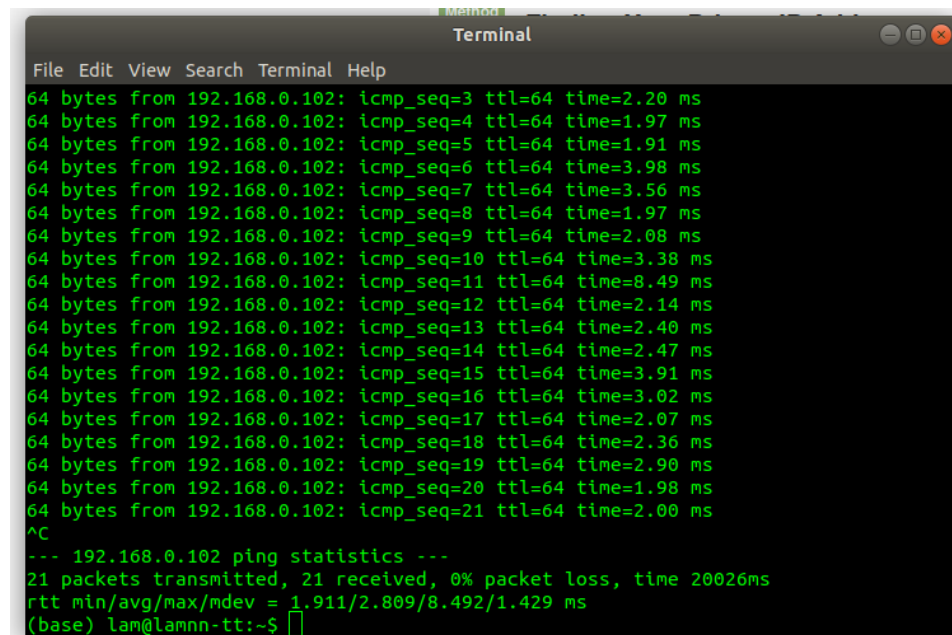
```
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 30 bytes 2916 (2.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 30 bytes 2916 (2.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

(a) Of the Server

(b) Of the Client

Figure 5: IP address

To ping each other we just use the standard ping function of Linux.



The image shows a terminal window titled 'Terminal' displaying the output of a ping command from the server to the client. The output shows 21 successful ping requests, each receiving 64 bytes from 192.168.0.102. The ping statistics at the bottom show 21 packets transmitted, 21 received, 0% packet loss, and a time of 20026ms. The round trip time (rtt) statistics are: min/avg/max/mdev = 1.911/2.809/8.492/1.429 ms.

```
64 bytes from 192.168.0.102: icmp_seq=3 ttl=64 time=2.20 ms
64 bytes from 192.168.0.102: icmp_seq=4 ttl=64 time=1.97 ms
64 bytes from 192.168.0.102: icmp_seq=5 ttl=64 time=1.91 ms
64 bytes from 192.168.0.102: icmp_seq=6 ttl=64 time=3.98 ms
64 bytes from 192.168.0.102: icmp_seq=7 ttl=64 time=3.56 ms
64 bytes from 192.168.0.102: icmp_seq=8 ttl=64 time=1.97 ms
64 bytes from 192.168.0.102: icmp_seq=9 ttl=64 time=2.08 ms
64 bytes from 192.168.0.102: icmp_seq=10 ttl=64 time=3.38 ms
64 bytes from 192.168.0.102: icmp_seq=11 ttl=64 time=8.49 ms
64 bytes from 192.168.0.102: icmp_seq=12 ttl=64 time=2.14 ms
64 bytes from 192.168.0.102: icmp_seq=13 ttl=64 time=2.40 ms
64 bytes from 192.168.0.102: icmp_seq=14 ttl=64 time=2.47 ms
64 bytes from 192.168.0.102: icmp_seq=15 ttl=64 time=3.91 ms
64 bytes from 192.168.0.102: icmp_seq=16 ttl=64 time=3.02 ms
64 bytes from 192.168.0.102: icmp_seq=17 ttl=64 time=2.07 ms
64 bytes from 192.168.0.102: icmp_seq=18 ttl=64 time=2.36 ms
64 bytes from 192.168.0.102: icmp_seq=19 ttl=64 time=2.90 ms
64 bytes from 192.168.0.102: icmp_seq=20 ttl=64 time=1.98 ms
64 bytes from 192.168.0.102: icmp_seq=21 ttl=64 time=2.00 ms
^C
--- 192.168.0.102 ping statistics ---
21 packets transmitted, 21 received, 0% packet loss, time 20026ms
rtt min/avg/max/mdev = 1.911/2.809/8.492/1.429 ms
(base) lam@lamnn-tt:~$
```

Figure 6: Ping from Server to Client

6 Can you watch the video in the client machine? Evaluate the quality of the video streaming service.

When trying to connect to the server, we can see the video with good quality (no delay, no jitter, the stream is smooth)

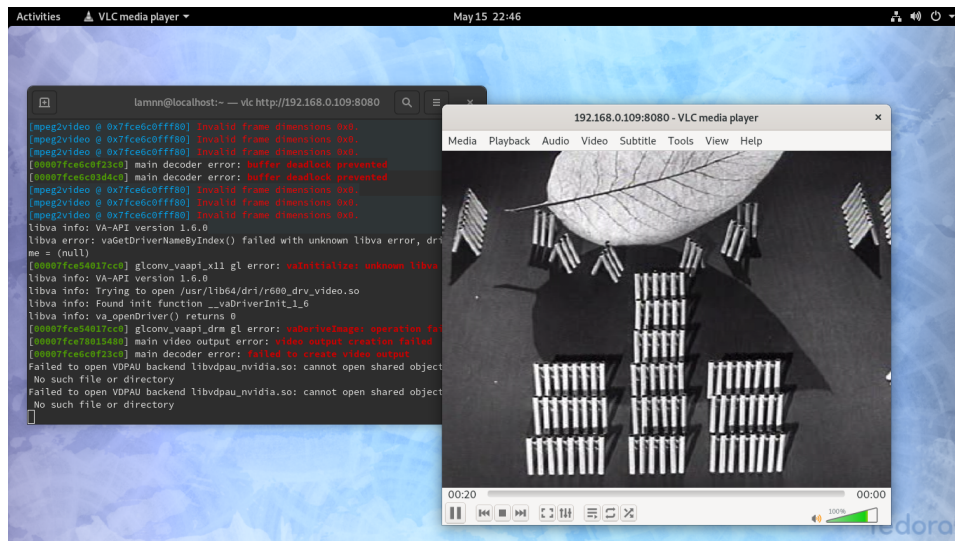


Figure 7: Video streaming

7 What is the result of the ping test? Can you see an increase of 100 milliseconds?

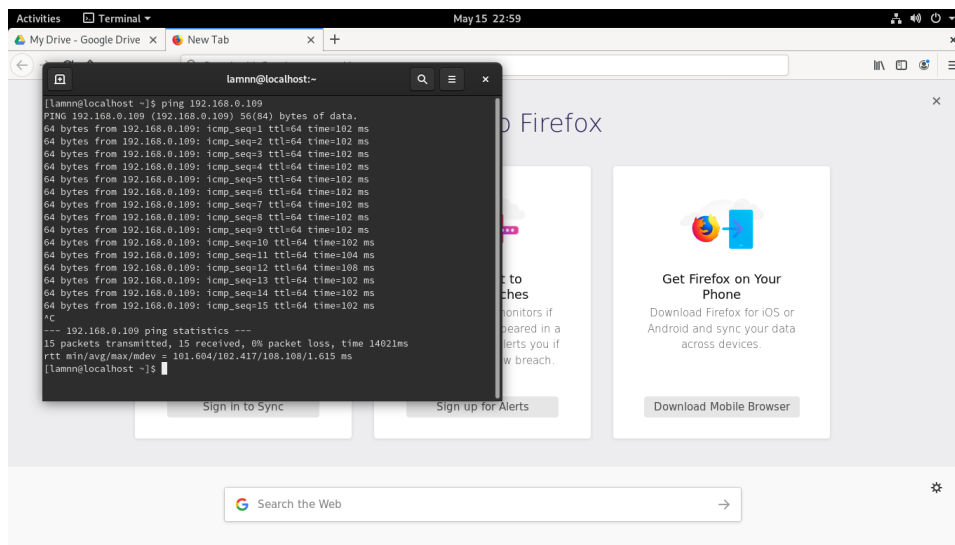


Figure 8: Ping result when has delay=100ms

As figure 8 shows, the time has been added 100ms from the previous time.

8 Disable the buffering function of VLC in Client machine. Then, evaluate the video quality at the Client machine. How can you conclude the impact of fix delay on video streaming service?

The frame periodically froze. In conclusion, the impact of fix delay on video streaming service is constantly freeze the frames.

9 Evaluate the video quality at the Client machine. How can you conclude the impact of delay variation on video streaming service?

The frame randomly froze. In conclusion, the impact of delay variation on video streaming service is sometimes freeze the frames and for different duration.

- 10 Evaluate the video quality at the Client machine. How can you conclude the impact of fix loss rate on video streaming service? Try to increase the value of loss rate to see the impact more clear.**

Sometimes, the video jumps from one scene to another scene without the transition. In conclusion, the impact of fix loss rate on video streaming service is the stream is periodically jitter. While it is annoying, if rate is low, it was bearable

- 11 Evaluate the video quality at the Client machine. How can you conclude the impact of loss rate variation on video streaming service? Try to increase this value to see the impact more clear.**

There was a long duration of jitter. There was a 10 seconds black screen at some point. In conclusion, the impact of loss rate variation on video streaming service is the stream is big gap in video. As a customer, this maybe unacceptable.

- 12 Evaluate the video quality at the Client machine. How can you conclude the impact of packet duplication on video streaming service? Try to increase this value to see the impact more clear.**

There was a slight freeze at some frame.

- 13 Evaluate the video quality at the Client machine. How can you conclude the impact of packet corruption on video streaming service?**

There is some frame that appeared after the frame that suppose to follow it. It makes the movement of the character seems to be backward.