



THE UNIVERSITY OF
MELBOURNE

ELEN90061: COMMUNICATIONS NETWORKS

Workshop 1

Name: Xiang Xu

Student ID: 1454161

Name: Xiufu Sun

Student ID: 1372750

Part 1 – Emulating a LAN using GNS3

Questions

Q 1.1. Reflect on IP vs MAC addresses. What is the difference? Did you assign a MAC address to the VPCSs in the emulation? Where do they come from?

Answer:

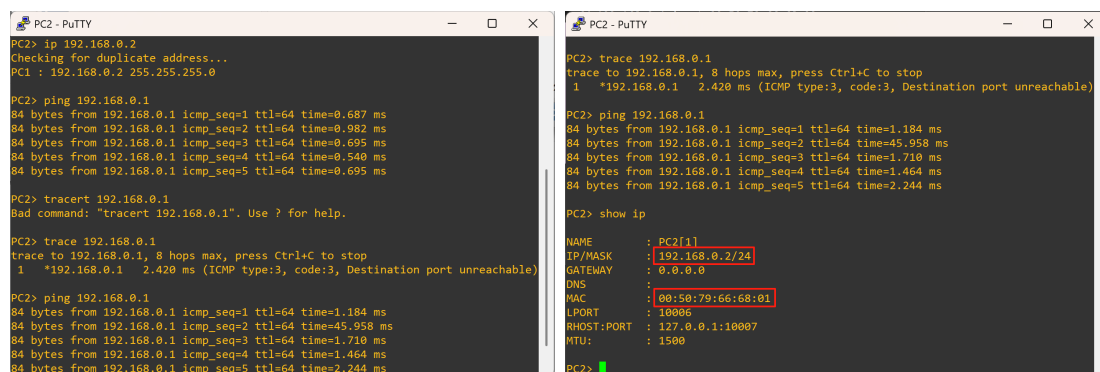
1. The Difference of IP and MAC:

- The addressing protocol layers are different. IP addresses are applied at the third layer of OSI, namely the network layer, while MAC addresses are used at the second layer of OSI, that is, the data link layer.
- The lengths are different. An IP address is 32 bits, while a MAC address is 48 bits.
- IP addresses can be freely changed within a local area network, but MAC addresses generally cannot be changed.
- In the external network (where the IP address is not a reserved field), the IP address is provided by the service provider, and the MAC address is determined by the manufacturer of the network card.

2. Did I manually assign a MAC address to the VPCS: **Absolutely not**. When we create and start a VPCS instance, the simulator software will automatically generate a MAC address for it.

3. Where do they come from:

- MAC: In GNS3, it is randomly generated by the software. In reality, the MAC address of a network card is fixed at the time of its manufacture. It cannot be changed, unless the firmware is reprogrammed.
- IP: In a local area network (LAN), IP addresses can be set manually (as shown in Figure 1), or they can be assigned by a router with the DHCP function enabled. On the external network, IP addresses are provided by the service provider.



```
PC2> ip 192.168.0.2
Checking for duplicate address...
PC1 : 192.168.0.2 255.255.255.0

PC2> ping 192.168.0.1
84 bytes from 192.168.0.1 icmp_seq=1 ttl=64 time=0.687 ms
84 bytes from 192.168.0.1 icmp_seq=2 ttl=64 time=0.982 ms
84 bytes from 192.168.0.1 icmp_seq=3 ttl=64 time=0.695 ms
84 bytes from 192.168.0.1 icmp_seq=4 ttl=64 time=0.540 ms
84 bytes from 192.168.0.1 icmp_seq=5 ttl=64 time=0.695 ms

PC2> traceroute 192.168.0.1
Bad command: "traceroute 192.168.0.1". Use ? for help.

PC2> trace 192.168.0.1
trace to 192.168.0.1, 8 hops max, press Ctrl+C to stop
1 *192.168.0.1 2.420 ms (ICMP type:3, code:3, Destination port unreachable)

PC2> ping 192.168.0.1
84 bytes from 192.168.0.1 icmp_seq=1 ttl=64 time=1.184 ms
84 bytes from 192.168.0.1 icmp_seq=2 ttl=64 time=45.958 ms
84 bytes from 192.168.0.1 icmp_seq=3 ttl=64 time=1.710 ms
84 bytes from 192.168.0.1 icmp_seq=4 ttl=64 time=1.464 ms
84 bytes from 192.168.0.1 icmp_seq=5 ttl=64 time=2.244 ms

PC2> show ip
NAME : PC2[1]
IP/MASK : 192.168.0.2/24
GATEWAY : 0.0.0.0
DNS :
MAC : 08:50:79:66:68:01
I/P/PORT : 18006
R/HOST/PORT : 127.0.0.1:10007
MTU : 1500
```

Figure 1: Use the ip command to set the IP address and subnet mask for the VPCS

Q 1.2. How does the switch know where to send the packets? Would it be able to manage it if you had connected 3 or 4 VPCSs to the switch? If yes, how?

Answer:

The switch operates entirely based on the MAC address table, which can be manually configured or learned autonomously by the switch. The process of forwarding data to the interface after dynamic learning is:

1. Learning: The switch will automatically learn the source MAC address of the data frame.
2. Forwarding: It will search for the destination MAC address of the data frame in the MAC table. If the search is successful, the data will be forwarded from the corresponding interface.
3. For unknown or broadcast addresses (FF:FF:FF:FF:FF:FF): If the search fails, the same data will be forwarded from all interfaces (flooding).
4. Refresh: By default, the switch refreshes the MAC address table every 300 seconds.

After connecting 3 or 4 VPCS, the switch can still be managed, and the MAC address table has become larger. As mentioned earlier in the management process, the switch first learns the source MAC address and the corresponding port number, and then sends the data out through the corresponding port by looking up the table.

Q 2.1. Which protocols do you observe in Wireshark during ping? Focusing on ping packets, which layers do you see in action? What does Wireshark tell us about them? Can you identify, for example, the MAC layer and find the MAC addresses? Which fundamental characteristics of the layered architecture do you observe in action?

Answer:

Observed Protocol: It can be seen from the captured data packets (as shown in Figure 2) that there are two protocols: ARP and ICMP.

1. ARP (Address Resolution Protocol)
 - VPCS2 (192.168.0.2) will broadcast an ARP request packet within the local area network: "Who has 192.168.0.1? Tell 192.168.0.2."
 - After receiving this broadcast, VPCS1 (192.168.0.1) will unicast an ARP reply packet to respond with its MAC address (00:50:79:66:68:00): "192.168.0.1 is at 00:50:79:66:68:00".
2. ICMP (Internet Control Message Protocol)
 - This is the protocol used by the ping command itself. As can be seen from Figure 3, it sends an Echo Request and receives an Echo Reply. This is based on the MAC address.

Layers in Action: The ping command mainly demonstrates at least three layers of the OSI seven-layer model: Physical Layer, Link Layer and Network Layer.

1. Physical Layer

- The first line "Frame..." of the ICMP message represents the physical layer, which shows the process of transmitting bits.

2. Link Layer

- "Ethernet II" represents the link layer (MAC address addressing), indicating the source MAC address, destination MAC address, and what the next layer protocol is.

3. Network Layer

- The third line of the message, Internet Protocol Version 4 (IPv4), represents the network layer, which is responsible for IP addressing and routing. This layer of the message contains the source IP address and the destination IP address.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Private 66:68:01	Broadcast	ARP	64	Who has 192.168.0.1? Tell 192.168.0.2
2	0.000000	Private 66:68:00	Private 66:68:01	ARP	64	192.168.0.1 is at 00:50:79:66:68:00
3	0.010449	192.168.0.2	192.168.0.1	ICMP	98	Echo (ping) request id=0x9bc5, seq=1/256, ttl=64 (r
4	0.011482	192.168.0.1	192.168.0.2	ICMP	98	Echo (ping) reply id=0x9bc5, seq=1/256, ttl=64 (r
5	1.030136	192.168.0.2	192.168.0.1	ICMP	98	Echo (ping) request id=0x9cc5, seq=2/512, ttl=64 (r
6	1.030136	192.168.0.1	192.168.0.2	ICMP	98	Echo (ping) reply id=0x9cc5, seq=2/512, ttl=64 (r
7	2.085839	192.168.0.2	192.168.0.1	ICMP	98	Echo (ping) request id=0x9dc5, seq=3/768, ttl=64 (r
8	2.085839	192.168.0.1	192.168.0.2	ICMP	98	Echo (ping) reply id=0x9dc5, seq=3/768, ttl=64 (r
9	3.103620	192.168.0.2	192.168.0.1	ICMP	98	Echo (ping) request id=0x9ec5, seq=4/1024, ttl=64 (r
10	3.103620	192.168.0.1	192.168.0.2	ICMP	98	Echo (ping) reply id=0x9ec5, seq=4/1024, ttl=64 (r
11	4.120751	192.168.0.2	192.168.0.1	ICMP	98	Echo (ping) request id=0x9fc5, seq=5/1280, ttl=64 (r
12	4.122526	192.168.0.1	192.168.0.2	ICMP	98	Echo (ping) reply id=0x9fc5, seq=5/1280, ttl=64 (r

Frame 2: 64 bytes on wire (512 bits), 64 bytes captured (512 bi	0000 00 50 79 66 68 01 00 50 79 66 68 00 00 06 00 01	Pyfh P
Ethernet II, Src: Private 66:68:00 (00:50:79:66:68:00), Dst: Pr	0010 00 00 06 04 00 02 00 50 79 66 68 00 c0 a8 00 01P
Address Resolution Protocol (reply)	0020 00 50 79 66 68 01 c0 a8 00 02 00 00 00 00 00 00	Pyfh...
Hardware type: Ethernet (1)	0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Protocol type: IPv4 (0x0800)		
Hardware size: 6		
Protocol size: 4		
Opcode: reply (2)		
Sender MAC address: Private 66:68:00 (00:50:79:66:68:00)		
Sender IP address: 192.168.0.1		
Target MAC address: Private 66:68:01 (00:50:79:66:68:01)		
Target IP address: 192.168.0.2		

Figure 2: The packets captured when executing the ping command

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Private 66:68:01	Broadcast	ARP	64	Who has 192.168.0.1? Tell 192.168.0.2
2	0.000000	Private 66:68:00	Private 66:68:01	ARP	64	192.168.0.1 is at 00:50:79:66:68:00
3	0.010449	192.168.0.2	192.168.0.1	ICMP	98	Echo (ping) request id=0x9bc5, seq=1/256, ttl=64 (r
4	0.011482	192.168.0.1	192.168.0.2	ICMP	98	Echo (ping) reply id=0x9bc5, seq=1/256, ttl=64 (r
5	1.030136	192.168.0.2	192.168.0.1	ICMP	98	Echo (ping) request id=0x9cc5, seq=2/512, ttl=64 (r
6	1.030136	192.168.0.1	192.168.0.2	ICMP	98	Echo (ping) reply id=0x9cc5, seq=2/512, ttl=64 (r
7	2.085839	192.168.0.2	192.168.0.1	ICMP	98	Echo (ping) request id=0x9dc5, seq=3/768, ttl=64 (r
8	2.085839	192.168.0.1	192.168.0.2	ICMP	98	Echo (ping) reply id=0x9dc5, seq=3/768, ttl=64 (r
9	3.103620	192.168.0.2	192.168.0.1	ICMP	98	Echo (ping) request id=0x9ec5, seq=4/1024, ttl=64 (r
10	3.103620	192.168.0.1	192.168.0.2	ICMP	98	Echo (ping) reply id=0x9ec5, seq=4/1024, ttl=64 (r
11	4.120751	192.168.0.2	192.168.0.1	ICMP	98	Echo (ping) request id=0x9fc5, seq=5/1280, ttl=64 (r
12	4.122526	192.168.0.1	192.168.0.2	ICMP	98	Echo (ping) reply id=0x9fc5, seq=5/1280, ttl=64 (r

Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bi	0000 00 50 79 66 68 00 00 50 79 66 68 01 00 00 45 00	Pyfh P
Ethernet II, Src: Private 66:68:01 (00:50:79:66:68:01), Dst: Pr	0010 00 54 c5 50 00 00 00 33 ba c0 a8 00 02 c0 a8	T #
Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.2	0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	T #
Internet Control Message Protocol	0030 00 00 11 11 11 11 11 11 11 11 11 11 11 11
Type: 8 (Echo (ping) request)	0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Code: 0	0050 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c012345
Checksum: 0x84c5 [correct]	0060 3e 3f	>
Checksum Status: Good		
Identifier (BE): 39877 (0x9bc5)		
Identifier (LE): 52897 (0x9cc5)		
Sequence Number (BE): 1 (0x0001)		
Sequence Number (LE): 256 (0x0100)		
Data (56 bytes)		

Figure 3: ICMP packets and their messages

The rest of the message is the information inside the data packet.

What Wireshark tells us:

- MAC address: The MAC address can be seen on the Ethernet II line.
- IP address: It can be seen in the third layer "Internet Protocol Version 4, Src:..."
- The size of the transmitted data: The first line of the physical layer shows how many bits were transmitted.

Fundamental characteristics of the layered architecture:

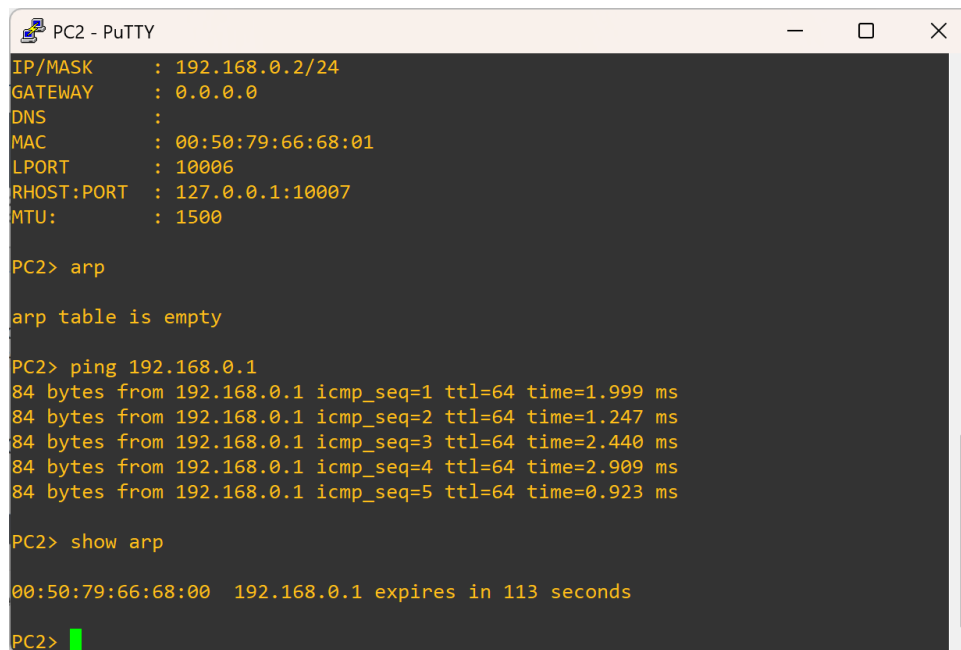
1. Encapsulation: When data is passed from the upper layer to the lower layer at the sending end, each layer adds a protocol header (and tail) to the data from the upper layer to form a new data unit. At the receiving end, the reverse decapsulation process is carried out. For example, in the packet of the ping command, the network layer adds IP addresses, and the link layer adds MAC addresses and the Frame Check Sequence (FCS) at the tail.
2. Independence of Layers: Each layer uses the services of the layer below it and provides services to the layer above it, but the functions, implementations, and technologies of each layer are independent. Changes in one layer do not directly affect other layers as long as the interfaces between them remain unchanged. For example, the network layer only cares about IP addresses, the link layer only concerns MAC addresses, and the ARP protocol is designed to solve the mapping problem between IP and MAC.

Q 2.2. Now focus on ARP protocol and tables. What do you observe about ARP? Comment on how it works. Discuss based on lectures and what you find online.

Answer:

The Address Resolution Protocol (ARP) is a protocol located between the data link layer and the network layer in the OSI model. Its purpose is to resolve the mapping relationship between the second-layer physical address (MAC address) and the third-layer logical address (IP address). As shown in Figure 2, the ARP protocol has two packets, which are divided into the inquiry and response parts. The following is the process of VPCS2 sending data to VPCS1:

1. Check the local ARP Table: If the corresponding MAC address is already in the table, there is no need to initiate an ARP request.
2. Broadcast ARP request: If no mapping is found in the ARP Table, VPCS will send an ARP request packet, which includes the IP and MAC of the sender (the local machine), the IP of the target, and a MAC address of all zeros, indicating an unknown MAC. This packet is a broadcast packet (the frame header MAC is all 1s). In this simulation, VPCS2 (192.168.0.2) broadcasts an ARP request packet within the local area network: "Who has 192.168.0.1? Tell 192.168.0.2."
3. Unicast ARP reply: All VPCS in the local area network receive the broadcast packet and check the IP. If the IP is the local machine's IP, it will unicast an ARP reply packet, which contains the local machine's MAC address. In this simulation, VPCS1 (192.168.0.1) unicasts an ARP reply packet to respond with its MAC address (00:50:79:66:68:00): "192.168.0.1 is at 00:50:79:66:68:00".



```
PC2 - PuTTY
IP/MASK      : 192.168.0.2/24
GATEWAY      : 0.0.0.0
DNS          :
MAC          : 00:50:79:66:68:01
LPORT        : 10006
RHOST:PORT   : 127.0.0.1:10007
MTU:         : 1500

PC2> arp

arp table is empty

PC2> ping 192.168.0.1
84 bytes from 192.168.0.1 icmp_seq=1 ttl=64 time=1.999 ms
84 bytes from 192.168.0.1 icmp_seq=2 ttl=64 time=1.247 ms
84 bytes from 192.168.0.1 icmp_seq=3 ttl=64 time=2.440 ms
84 bytes from 192.168.0.1 icmp_seq=4 ttl=64 time=2.909 ms
84 bytes from 192.168.0.1 icmp_seq=5 ttl=64 time=0.923 ms

PC2> show arp

00:50:79:66:68:00 192.168.0.1 expires in 113 seconds

PC2>
```

Figure 4: Ping command and ARP Table

4. Cache ARP Table: After receiving the reply packet, VPCS will store the mapping of IP and MAC in the ARP Table. It can be seen in Figure 4 that after the ping command, a new mapping is stored in the ARP Table.

It is worth noting that each entry has a Time To Live (TTL), and will be automatically deleted after expiration. This is necessary because although the MAC address of a device usually remains unchanged, its IP address may change (for example, due to DHCP reassignment). Regular refreshing can ensure the accuracy of the mapping.

Part 3 — Packet Capture and Analysis

Questions

Q 3.1. What does the CSV file contain? Does it contain the packet contents?

Answer: The header of the CSV file indicates that it contains the following contents:

"No.", "Time", "Source", "Destination", "Protocol", "Length", "Info"

Do not include the packet contents.

Q 3.2. Compute the sequence of packet inter-arrival times (IAT). Provide min, max, mean, std. Also compute instantaneous and average rate of the flow.

Answer: The computed statistics of the original IAT sequence are:

- Minimum: 4.0×10^{-6} s
- Maximum: 58.9890 s
- Mean: 0.0837 s

- Standard deviation: 2.1517 s

Flow rate statistics:

- Average rate: 11.95 packets/s
- Average instantaneous rate: 64088 packets/s

Q 3.3. “Clean” the time series by keeping IATs below the 95th percentile. Report stats (min, max, mean, std) on the cleaned series.

Answer:

With the 95th percentile threshold of 0.003965 s, the cleaned dataset has:

- Sample size: 1996 (95% of original)
- Minimum: 4.0×10^{-6} s
- Maximum: 0.0039 s
- Mean: 1.39×10^{-4} s
- Standard deviation: 4.70×10^{-4} s

Q 3.4. Plot the time series and histogram of the cleaned IATs. Add a title and label axes with correct units.

Answer:

The time series and histogram of the cleaned IATs are shown in Figures 5 and 6.

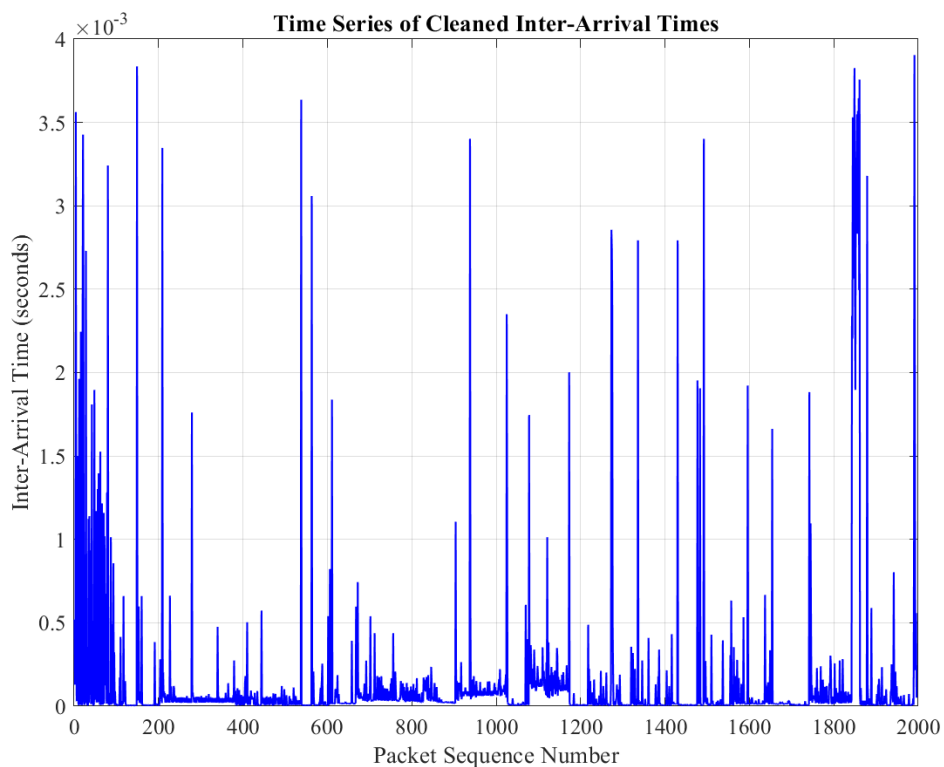


Figure 5: Time Series of Cleaned Inter-Arrival Times

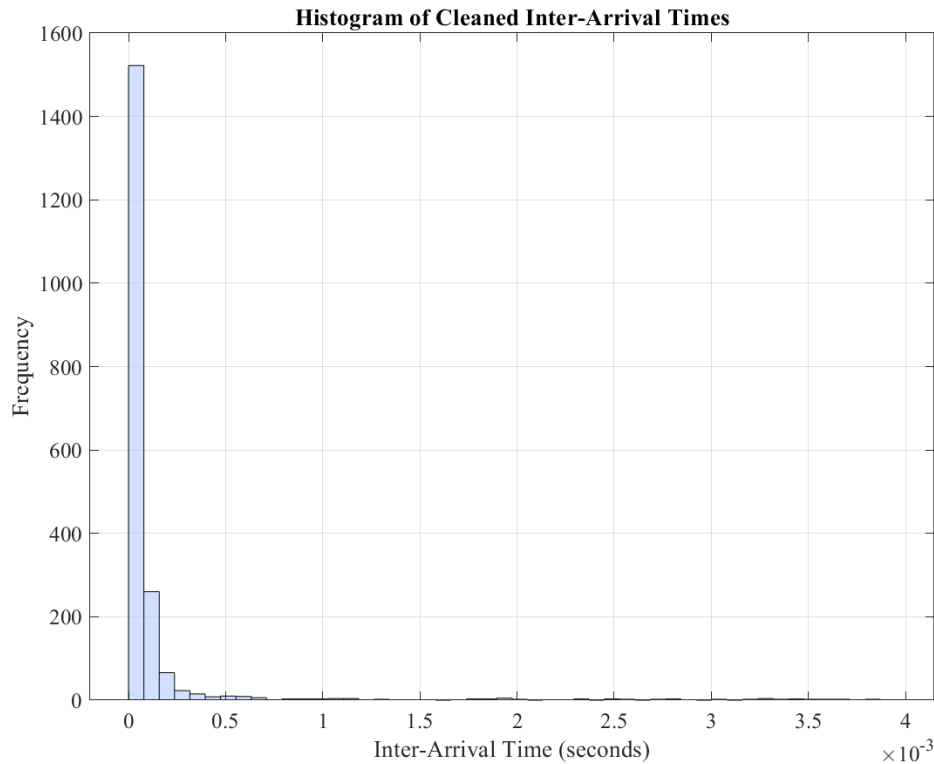


Figure 6: Histogram of Cleaned Inter-Arrival Times

Q 3.5. Fit an exponential law to the IAT distribution (e.g., MATLAB `expfit` or Python `scipy.optimize.curve_fit`).

Answer:

The exponential fit yielded:

$$\lambda = 7203.31 \text{ s}^{-1}, \quad \frac{1}{\lambda} = 1.39 \times 10^{-4} \text{ s}.$$

Q 3.6. Plot the IAT distribution and superimpose the fitted exponential. Discuss whether “the packet arrival time series can be accurately modelled by a Poisson process.”

Answer:

Figure 7 shows the observed IAT distribution with the exponential fit.

In theory, if the packet arrivals follow a Poisson process, then the inter-arrival times (IATs) should follow an exponential distribution with parameter λ , where both the mean and standard deviation are equal to $1/\lambda$. This implies a coefficient of variation (CV) of 1.

As shown in Figure 7, the exponential curve fits the general decay trend of the measured IAT distribution. At first glance, this resembles a Poisson process. However, the empirical results show a coefficient of variation of 3.385, which is significantly larger than the theoretical value of 1. This indicates much higher variability than expected under a Poisson model.

Therefore, although the IAT distribution superficially appears exponential, the packet arrival process does not truly follow a Poisson process due to its strong burstiness and variability.

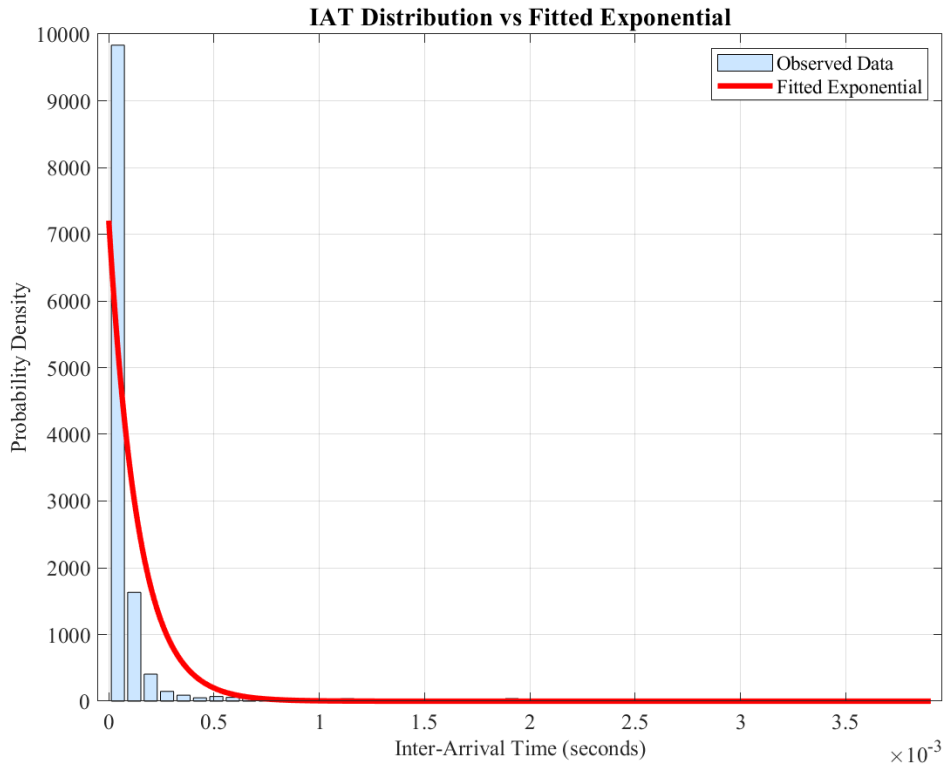


Figure 7: Observed IAT Distribution vs Fitted Exponential

Q 3.7. Repeat Questions 3.2–3.6 for two additional traces with modified link characteristics (e.g., added delays, random drops). Compare and discuss.

Answer:

The comparative results are summarized below:

- **Original trace:** mean 1.39×10^{-4} s, std 4.70×10^{-4} s, CV = 3.385.
- **With delays:** mean 1.80×10^{-4} s, std 4.71×10^{-4} s, CV = 2.620.
- **With drops:** mean 1.35×10^{-4} s, std 4.48×10^{-4} s, CV = 3.315.

Figures 8–10 illustrate the distributions for each case.

Analysis shows that introducing delays increases the average IAT slightly and reduces the relative variability (lower CV), while packet drops reduce the sample size and preserve high burstiness (CV > 3). In all cases, the CV remains well above 1, which confirms that the packet arrivals cannot be accurately described by a Poisson process.

Overall, link impairments such as delays or drops affect the shape of the distribution and the statistics, but none of the traces exhibit the statistical properties expected of a true Poisson process.

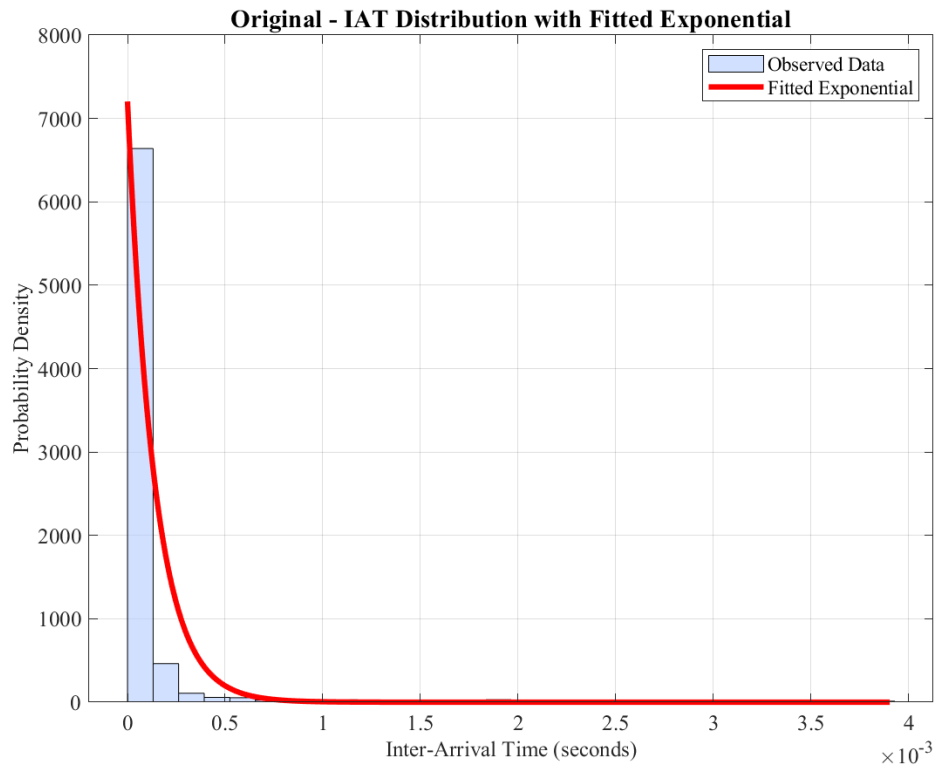


Figure 8: Original IAT Distribution

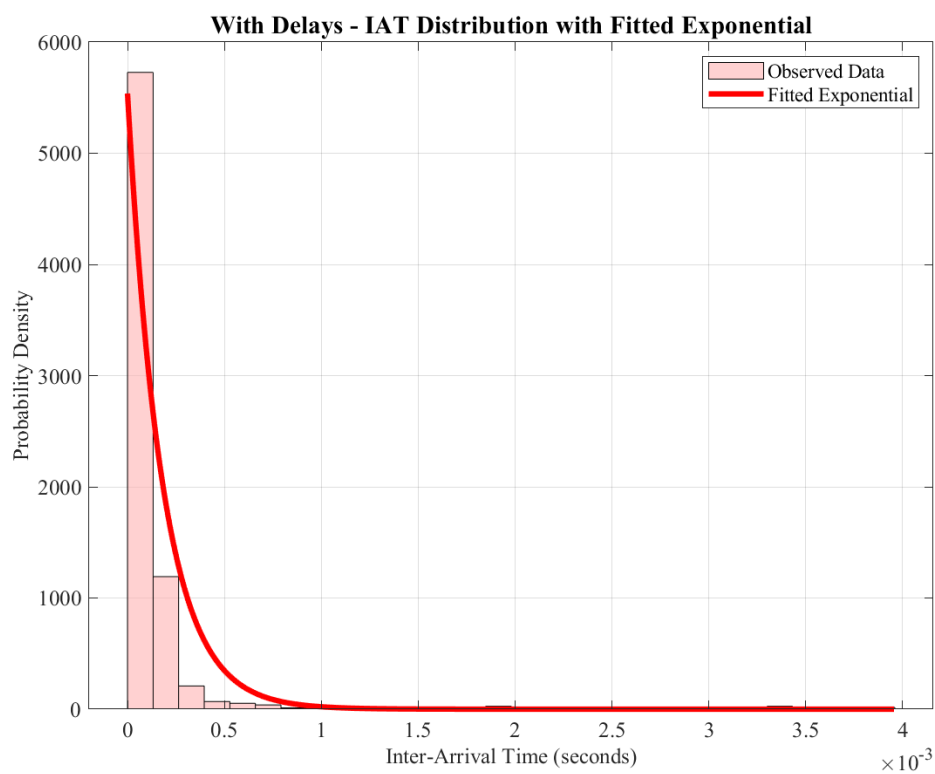


Figure 9: IAT Distribution with Added Delays

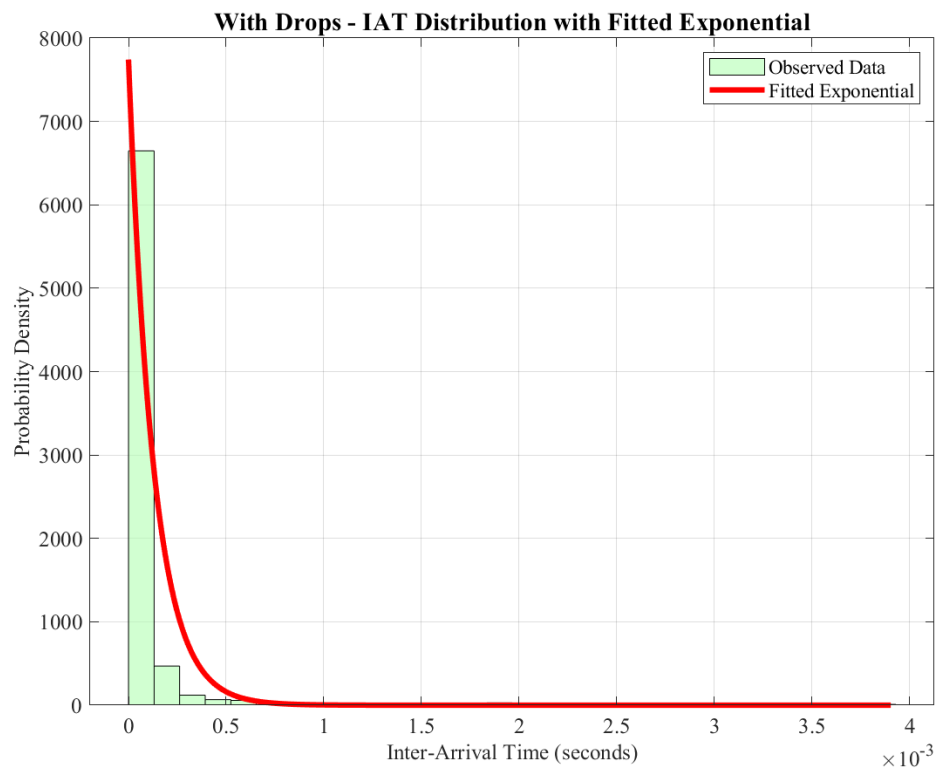


Figure 10: IAT Distribution with Random Packet Drops