Network Layer — Data Plane (Lab5)

Contents

Introduction

The goal of this lab is to learn how to setup and analyze NAT, DHCP as well as how to analyze IP datagram and IP fragmentation. The lab has several **milestones**. Make sure you reach each one before advancing to the next.

For delivery, submit a PDF report where you answer **only** those steps that are marked with **REPORT(%)**. Additionally, you should submit the codes or capture files, if they were **explicitly** asked for. The percent point gives you an indication of the score of that question. Lab5 counts for **4.5 points** of your final score in this course.

System Setup

In this lab, you will use a system setup as shown in Figure 1. It is an extension to lab2, but the servers now have public IP addresses in addition to two private networks. Your job is to setup NAT in order to access the servers from the private networks. Additionally, you will setup DHCP to dynamically assign IP addresses to clients in your private networks.

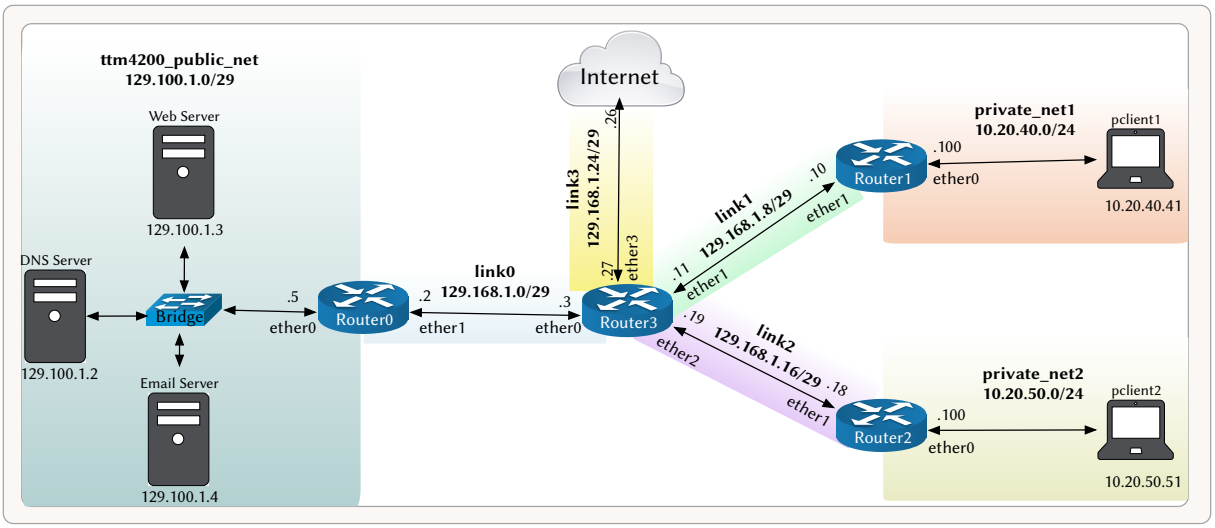

Figure 1: System Setup for Lab5

> *To get a deeper understanding of the system setup, read through the "docker-compose.yml" file.*

Depending on how they used the previous labs, running the usual "docker-compose up -d –build" can lead to errors here (trying to create containers whose name is already in use) -> "docker-compose kill" followed by "docker rm -v $(docker ps -a -q -f status=exited)" did the trick for me

- Create and start the containers using the docker-compose file.

> *If you didn't remove the containers from previous labs (`docker-compose down`), you will get the error: "trying to create containers whose name is already in use". You can use this command:* `docker-compose kill && docker rm -v $(docker ps -a -q -f status=exited)`

- You can connect to any container using SSH:

```
ssh ttm4200@127.0.0.1 -p [Mapped Port in the Docker Host]
```

  To simplify the configuration, the containers' SSH ports are mapped to the Docker Host as in the following list, but feel free to change them according to your preferences (do not forget to rebuild the containers):

  - "dnsserver" → 2000
  - "webserver" → 3000
  - "mailserver" → 4000
  - "router0" → 9000
  - "router1" → 10000
  - "router2" → 20000
  - "router3" → 30000
  - "pclient1" → 41000
  - "pclient2" → 51000

- To copy files from or to containers, use `scp` or the mounted volumes. Read through the "docker-compose.yml" to find out which directories are mounted.
- The servers are already configured, so you don't have to change anything.

  > *The DNS server is configured to allow queries from any network. Additionaly, all containers are configured to use this DNS server as their nameserver.*

- To save your configuration in "router1" or "router2", run the script: `sudo sh ~/work_dir/save.sh` inside the corresponding container.

1 Milestone 1 – Network Addressing and Routing

For simplicity, the IP addresses have already been set in the "docker-compose" file. Additionally, the network has been configured with static routes to have full connectivity.

Q1. **REPORT(5%):** Check the routing table in "router3" (`ip route list`) and explain the purpose of **every entry** based on Figure 1.

2 Milestone 2 – Network Address Translation (NAT)

2.1 NAT Setup

In this section, you will set up NAT using *iptables* (https://linux.die.net/man/8/iptables).

- In the containers "router1" and "router2", activate IP Masquerading to enable NAT/Masquerading for the private networks "private_net1" and "private_net2".

  > *Hint: https://askubuntu.com/questions/95199/two-network-cards-and-ip-forwarding/95261#95261, and pay attention to the names of external network interface and internal network interface, as shown in Figure 1 _*

- Check the NAT settings by checking connectivity to a public network (`ping 129.100.1.2`).
- Start a packet capture with `tcpdump` on your "pclient1" and "webserver", and dump the capture to a file (e.g. "nat_pclient1.pcap" and "nat_webserver.pcap").
- In the "pclient1" container, enter the following into the command line: `wget www.ttm4200.com`. This will retrieve the content from the webserver.
- Stop `tcpdump` capturing, both at "pclient1" and "webserver" and copy the capture files to your VM.

2.2 Analyzing Packets Before NAT

Open the "nat_pclient1.pcap" file in Wireshark. To display only those frames that are sent to/from webserver, apply a display filter of `tcp.port == 80` in the filter field. Then answer these questions and validate your answer with an **annotated** screenshot from Wireshark:

Q2. **REPORT(2%):** Find the HTTP GET sent from the client to the webserver. What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP GET?

Q3. **REPORT(2%):** Find the corresponding 200 OK HTTP message received from the webserver. What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP 200 OK message?

Q4. **REPORT(2%):** Before a GET command can be sent to an HTTP server, TCP must first set up a connection using the three-way SYN/ACK handshake. What are the source and destination IP addresses and source and destination ports for the TCP SYN segment?

Q5. **REPORT(2%):** What are the source and destination IP addresses and source and destination ports of the SYN ACK sent in response to the SYN.

2.3 Analyzing Packets After NAT

Open the "nat_webserver.pcap" file in Wireshark, apply the display filter `tcp.port == 80`. Then answer these questions and validate your answer with an **annotated** screenshot from Wireshark:

Q6. **REPORT(4%):** Find the HTTP GET message that was sent from the client to the webserver. What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP GET? Which of these fields are the same, and which are different, compared to your answer to question (Q2) above? Explain why?

Q7. **REPORT(4%):** Are any fields in the HTTP GET message changed? Which of the following fields in the IP datagram carrying the HTTP GET are changed: *Version*, *Header Length*, *Flags*, *Checksum*? If any of these fields is changed, give a reason (in one sentence) stating why this field had to change.

Q8. **REPORT(2%):** Find the 200 OK HTTP message. What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP 200 OK message? Which of these fields are the same, and which are different, compared to your answer to question (Q3) above?

Q9. **REPORT(2%):** Find the TCP SYN segments. What are the source and destination IP addresses and source and destination ports? Which of these fields are the same, and which are different than your answer to question (Q4) above?

Q10. **REPORT(2%):** Find the TCP SYN ACK segments, What are the source and destination IP addresses and source and destination ports? Which of these fields are the same, and which are different than your answer to question (Q5) above?

Q11. **REPORT(5%):** Submit your capture files "nat_pclient1.pcap" and "nat_webserver.pcap" along with the report. We will check it and make sure that it is unique and corresponds to your answers above

- To use the current NAT setup for later, save your configuration by running `sudo sh ~/work_dir/save.sh` in "router1" and "router2" containers. Later, you can just run `sudo sh ~/work_dir/restore.sh` to restore the setup without rebuilding the docker images.

3 Milestone 3 – Dynamic Host Configuration Protocol (DHCP)

3.1 DHCP Setup

In this section, you will set up a DHCP server and a client. We will use ISC DHCP, which offers an open-source implementation of a DHCP server and client. You will set up an authoritative DHCP server for the local network "private_net2" (see figure 1). The server-side, *isc-dhcp-server* is already installed in "router2", but you need to configure it. You can read this article (https://en.wikiversity.org/wiki/Configure_ISC-DHCP_server) on how to configure a ISC-DHCP server.

- Configure the DHCP server's properties in the "/etc/dhcp/dhcpd.conf" file, you can start with this template:

```
#declare this DHCP server as authoritative
==fill here==;
#specify a subnet and its netmask
subnet ==fill here== netmask ==fill here== {

   #specify a range of addresses the server can offer
   #(exclude the router ip from this range)
   range ==fill here==  ==fill here== ;

   option domain-name-servers 129.100.1.2;
   option domain-name "ttm4200.com";
   option routers 10.20.50.100;
}
default-lease-time 600;
max-lease-time 7200;
```

- Activate DHCP on the interface "ether0" by editing the "/etc/default/isc-dhcp-server" configuration file (comment "INTERFACESv6").
- Check the syntax of the server configuration file for errors (`dhcpd -t`) and if no problems are found restart the server (`sudo service isc-dhcp-server restart`).
- Start packet capturing in "router2" on "ether0" interface (`tcpdump -i ether0 -w dhcp.pcap`)
- On the client-side, *dhclient* is already installed in "pclient2". You will use this client for retrieving a DHCP lease.
- In "pclient2", release any existing DHCP client lease by using `sudo dhclient -r -v`. Afterwards, request a new IP address from a DHCP server `sudo dhclient -v`. Then, **again**, release the lease.

> *You might lose SSH connection to "pclient2". You can reconnect using* `docker attach pclien`. *Ignore the error message "System has not been booted with systemd as init system (PID 1). Can't operate.", which is a systemd-related error message stems from being inside a container (https://stackoverflow.com/questions/39169403/systemd-and-systemctl-within-ubuntu-docker-images#answer-39169889).*

- Stop the packet capture and copy the capture file to your VM.

3.2 Analyzing DHCP Packets

Open the capture file "dhcp.pcap" in Wireshark and display only DHCP frames (apply the display filter `dhcp` or similar). Then answer these questions and validate your answer with an **annotated** screenshot from Wireshark:

Q12. **REPORT(2%):** Are DHCP messages sent over UDP or TCP?

Q13. **REPORT(2%):** What values in the DHCP discover message differentiate this message from the DHCP request message?

Q14. **REPORT(2%):** What is the value of the Transaction-ID in each of the four (Discover/Offer/Request/ACK) DHCP messages? What is the purpose of the Transaction-ID field?

Q15. **REPORT(5%):** What are the source and destination IP addresses for each of the four DHCP messages (Discover/Offer/Request/ACK DHCP)? Based on your reading, are these IPs what you expected? Can you explain why?

Q16. **REPORT(4%):** By consulting the captured packets, is there a relay agent between the client and the DHCP server? What values in the packet indicate the presence or absence of a relay agent?

Q17. **REPORT(2%):** What is the router and subnetmask in the DHCP offer message?

Q18. **REPORT(2%):** How long is the IP address lease time? Does it correspond to the "default-lease-time" or "max-lease-time" in your DHCP server settings?

Q19. **REPORT(4%):** Does the DHCP server acknowledge the DHCP release message? What would happen if the client's DHCP release message is lost?

Q20. **REPORT(5%):** Submit your capture files "dhcp.pcap" along with the report. We will check it and make sure that it is unique and corresponds to your answers above.

- To use the current DHCP setup for later, save your configuration by running `sudo sh ~/work_dir/save.sh` in the "router2" container. Later, you can just run `sudo sh ~/work_dir/restore.sh` to restore the setup without rebuilding the docker images.

4 Milestone 4 – Analyzing IP Datagrams

In this milestone, you will analyze the various fields in the IP datagram and IP fragmentation.

4.1

- In "pclient2" container, start packet capturing with `tcpdump` and dump the capture to a file (e.g. "ip_analysis.pcap"). Afterwards, send `traceroute` datagrams of 150, 2250, and 4550 bytes to the webserver:

```
traceroute www.ttm4200.com 150
traceroute www.ttm4200.com 2250
traceroute www.ttm4200.com 4550
```

> *HINT: You might need to request a new IP address from the DHCP server.*

4.2

Open the capture file in Wireshark and select the first UDP message sent by “pclient2”, then answer these questions and validate your answer with an **annotated** screenshot from Wireshark:

Q21. **REPORT(2%):** Within the IP packet header, what is the value in the upper layer protocol field?

Q22. **REPORT(2%):** How many bytes are there in the IP header? How many bytes are there in the payload of the IP datagram? Explain how you determined the number of payload bytes.

Q23. **REPORT(2%):** Has this IP datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented.

Q24. **REPORT(5%):** By looking at the series of UDP messages sent by “pclient2”, determine which fields in the IP datagram change from one datagram to the next? Which fields stay constant? Why?

4.3

Select the first UDP message sent by “pclient2” after the packet size was changed to 2250 bytes. Then answer these questions and validate your answer with an **annotated** screenshot from Wireshark:

Q25. **REPORT(2%):** Has that message been fragmented across more than one IP datagram? How can you tell?

Q26. **REPORT(4%):** Find the first fragment of the fragmented IP datagram. What information in the IP header indicates that the datagram has been fragmented? What information in the IP header indicates whether this is the first fragment or a latter fragment? How long is this IP datagram?

Q27. **REPORT(4%):** Find the second fragment of the fragmented IP datagram. What information in the IP header indicates that this is not the first datagram fragment? Are there more fragments? How can you tell?

Q28. **REPORT(4%):** What fields change in the IP header between the first and the second fragment?

Q29. **REPORT(4%):** How many bytes are there in the first fragment? How many bytes are there in the second fragment? Why is the sum of the two fragments greater than the packet size you have sent (2200 bytes)?

4.4

Select the first UDP message sent by "pclient2 after the packet size was changed to 4550 bytes. Then answer these questions and validate your answer with an **annotated** screenshot from Wireshark:

Q30. **REPORT(2%):** How many fragments were created from the original datagram?

Q31. **REPORT(4%):** What fields change in the IP header among the fragments?

Q32. **REPORT(5%):** Submit your capture files “ip_analysis.pcap” along with the report. We will check it and make sure that it is unique and corresponds to your answers above.

Contributors

Abdulmajid Murad, David Palma, Stanislav Lange, Mathias Pettersen, Sebastian Fuglesang.