Lab Exercise Sheet 1

IP Version 4 Topology

Document and analyze your experimental procedures by using your Wireshark and terminal recordings. Note all relevant intermediate steps. Mark and explain all relevant information, such as protocol header fields, MAC addresses, IP addresses, port numbers. If you have little experience with Linux, you may need to do some research.

Group number:	
First name:	
Last name:	
Student number:	

This lab exercise sets the foundation of the course Practical Computer Networks and Application in this semester! Therefore the setup of the presented network with its topology is mandatory for successful participation in this course!

In this lab, you will perform variant experiments on Linux machines running the operating system Debian. For the lab exercise, you need to disable the Network Manager if it is not already disabled by default! Your configuration on the hosts needs to be done **statically**!

Furthermore, the experiments need to be conducted using the Command-Line Interface (CLI) and Wireshark. In the lab exercise, you find boxes with questions regarding the tasks performed. You need to **demonstrate and answer** the questions in the exercise session to pass the lab exercise!

Examination

For every Lab Exercise you need to prepare slides documenting your setup! The slides should contain your network configurations, steps performed and screenshots of relevant captures! Send your slides to your lecturer and present your results in the lecture!

1. Setting up the network

a) Network Topology for this lab exercise

Figure 1 shows the setup of the network topology in this lab exercise. The network consists of **4 Host machines** in a private network. One machine is configured as the router of the private network with the network address `192.168.i.0/24`. The number `i` in the IP address is a placeholder for your **group number**!

The machines have the following configuration:

Host machines:

Router 1 – `eth0` – `192.168.i.1`

Host 1 – `eth0` – `192.168.i.10`

Host 2 – `eth0` – `192.168.i.20`

Host 3 – `eth0` – `192.168.i.30`

Router 1 interfaces:

`eth0` – `192.168.i.1`

`eth1` – `10.16.0.i`

Router 2:

Address – `10.16.0.200`

Router 1 needs to be configured as a router in the network with the IP address `192.168.i.1` on the interface `eth0`[1]. The second network interface `eth1` needs to be configured with the IP address `10.16.0.i`. The interface `eth1` with IP address `10.16.0.i` is the route to a **second** private network with the network address `10.16.0.0/16`.

b) Configuring the machines

Configure the machines using the command-line tool `ip`! ¸Please get familiar with the `ip` tool and its options (especially the `link`, `addr` and `route` options!) and configure the interface `eth0` of machines **Host** [**1-3**] with **IP address**, **Subnet Mask**, **Broadcast**, and **Gateway Address** according to the network topology shown in figure 1! Configure the interface `eth0` of **Router 1** in such a way, that it becomes the router in the network `192.168.i.0/24`. Make sure that **Host** [**1-3**] can reach the router and vice versa. Furthermore, configure the interface `eth1` of **Router 1** in such a way, that **Host** [**1-3**] have a route to the network `10.16.0.0/16`! Document your setup and list the commands and steps performed!

[1] The interfaces names can differ on Linux distributions and systems! In the literature the old predictable naming scheme using `ethX` is often found. In newer versions `systemd` uses a different naming scheme!
`https://www.freedesktop.org/wiki/Software/systemd/PredictableNetworkInterfaceNames/`

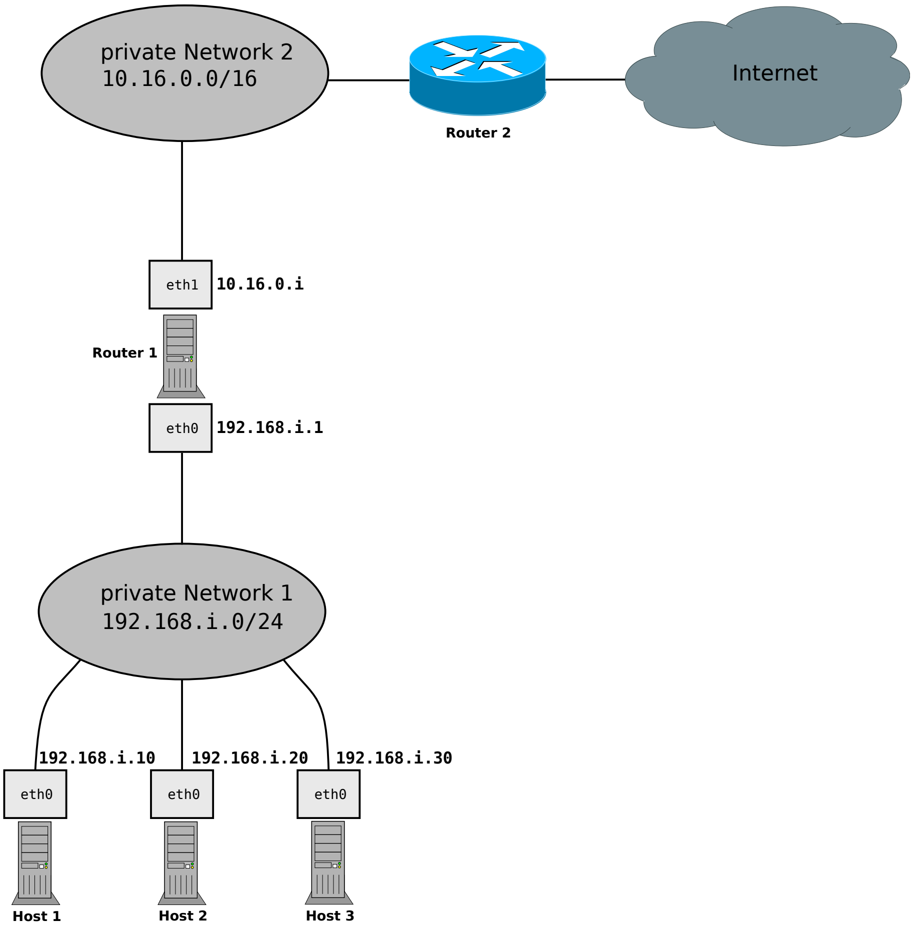


Figure 1: Network Topology of the Lab

c) Test your network setup

Test the configuration of your setup with the command-line tool `ping`! Send ICMP-requests from **Router 1** to the machines **Host [1-3]**. Also Send ICMP-requests from **Host [1-3]** to **Router 1**. Test the communication between **Host [1-3]** and the network `10.16.0.0/16` by sending ICMP-requests to machine **Router 2** (address `10.16.0.200`). Configure the name resolution on the hosts and use `8.8.8.8` as the name server! Document the tests and list the results and steps performed!

Demonstration Exercise 1

You should be able to demonstrate and explain the following things:

- Ping between **Host [1-3]**!
- The routing between **Host [1-3]** and **Router 1**!
- Ping from **Host [1-3]** to **Router 1** and vice versa!
- Ping from **Host [1-3]** to **Router 2**!

2. **Command-line tools for network administration**

   a) For the administration of hosts, servers, and other devices in a network, command-line tools are helpful for configuration and debugging network issues.

   Get familiar with these command-line tools:

   - `ip`
   - `traceroute`
   - `ping`
   - `curl`
   - `ss`
   - `nc` (`netcat`)
   - `nmap`
   - `dig`

   b) Inspect the man pages of the commands and get familiar with the functionality of each tool. Test your understanding of the tools by solving the following tasks by using the commands in your network setup. Document the commands with command-line options used together with their output for the following tasks:

   i. Show the IP address of the hosts' interface `eth0`!

   ii. Show the list of routes on **Host 1**!

   iii. Show the ARP table of **Router 1**!

   iv. Show the route of a request for the website hosted on **Router 2**!

   v. Send the string `"Hello!"` to the webserver hosted on **Router 2**!

   vi. List all open TCP ports and processes on **Host 1**!

   vii. List all listening UDP ports on **Host 1**!

   viii. Scan and list all open ports in the range of 1 - 1023 on the router **Router 1** that are reachable by **Host 1**!

   ix. Scan and list all open ports and services of all clients in the private network `192.168.i.0/24`!

3. **Monitoring network traffic with Wireshark**

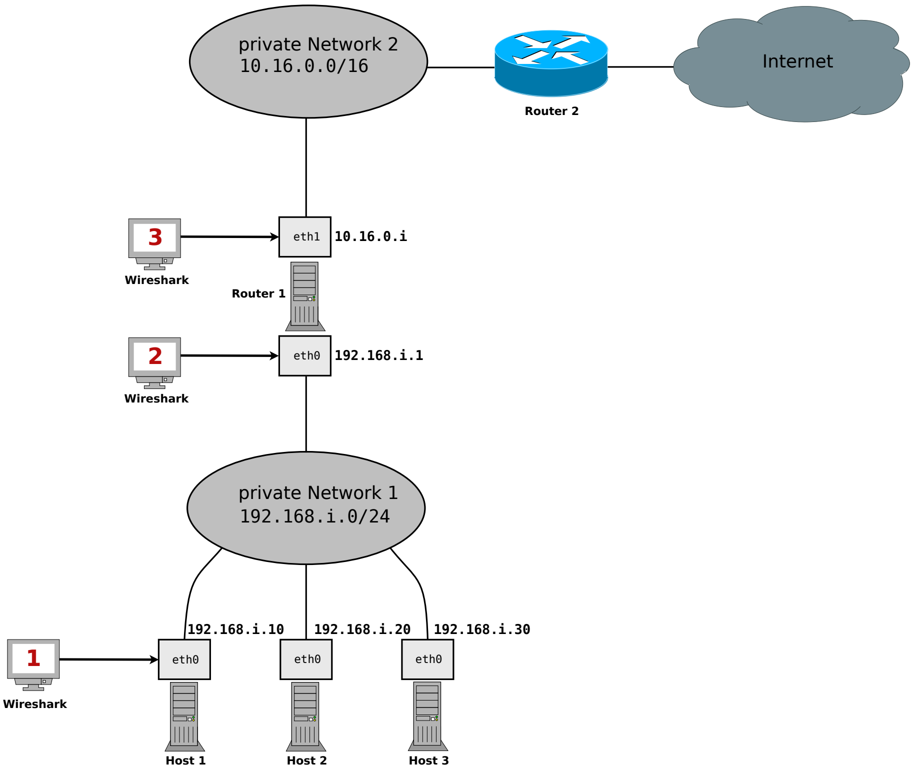


Figure 2: Junction points for monitoring

Monitor the network traffic with the application Wireshark! Figure 2 shows the junction points **1, 2, and 3**. These points mark the interfaces on which the network traffic should be analyzed. Junction point 1 is the interface `eth0` of **Host 1**, point 2 is the interface `eth0` of **Router 1**, and point 3 is the interface `eth1` of **Router 1**.

Ping the IP address of **Router 2** and monitor and analyze the ICMP-packets sent and received on the junction points! Set the filter in Wireshark to `icmp`. Document the requests and responses and list the results and steps performed!

Demonstration Exercise 3

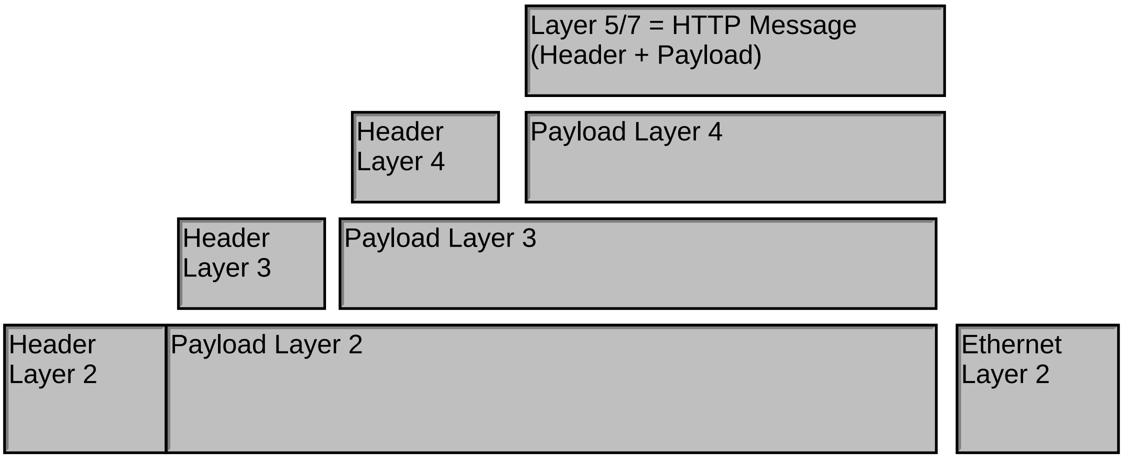
You should be able to demonstrate and explain the following things:

- The ICMP packets on the junction points **1, 2, and 3**!
- The contents of the ICMP packets (MAC-,IP addresses, type field, time-to-live)!
- The communication on junction point 1 of the `ping` command using the option `ping -s 65507` to **Router 1**!
- The message of the `ping` command when using `ping -M do -s 1473` to **Router 1**!

4. **Monitoring network protocols**

Monitor and analyze the **Three-Way-Handshake** of the TCP-protocol using Wireshark. Monitor the traffic of an HTTP-request to **Router 2** on **junction point 1** (interface `eth0` of **Host 1**) and on **junction point 2** (interface `eth0` of **Router 1**). Clear the ARP-Cache on **Host 1** and **Router 1** before sending the request and monitor the ARP-, TCP- and HTTP-Requests in the communication. Mark the ARP-Requests using the filter `arp`, the TCP segments of the Three-Way-Handshake in the communication using the filter `tcp`, and the HTTP-Requests using the `http` filter in Wireshark! Document the requests and responses and list the results and steps performed!

Show the protocol stack of the first HTTP response (starting with OSI layer 2). Fill in the correct number of Bytes of the headers, trailer, and payloads. Also, name the protocols used inside the single layers.



Additionally calculate the protocol overhead in Bytes and the protocol overhead ratio in % for the transmission of the HTTP response.

Demonstration Exercise 4

You should be able to demonstrate and explain the following questions:

- How is the ARP protocol involved and what is its purpose in the communication?
- How does the Three-Way-Handshake work and what flags are used?
- How are the protocols ARP, TCP, and HTTP working together in the communication? In what order are ARP, TCP, and HTTP messages exchanged?
- The relationship between the OSI model and the impact of header information on the transmission!