Practical Computer Networks and Applications

Exercise 1 – IP Version 4 Networks

Prof. Dr. Baun, Prof. Dr. Ebinger, Prof. Dr. Hahm, Prof. Dr. Kappes,
Dipl. Inf. (FH) Maurizio Petrozziello, Henry Cocos, M.Sc.
{baun,cocos,peter.ebinger,oliver.hahm,kappes,petrozziello}@fb2.fra-uas.de

Frankfurt University of Applied Sciences
Faculty of Computer Science and Engineering
Nibelungenplatz 1
60318 Frankfurt am Main

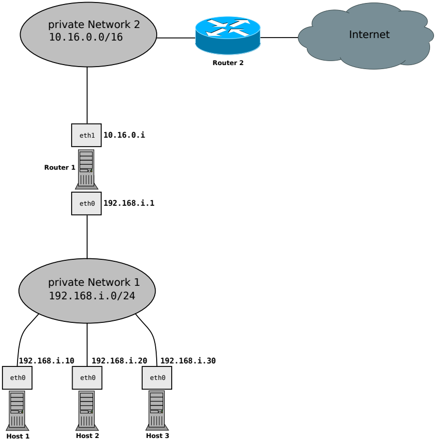
Contents

Exercise 1

Networking in Linux

Wireshark

Network Topology – Exercise 1



Network Topology of lab exercise 1

Private Network 1:

`192.168.i.0/24`

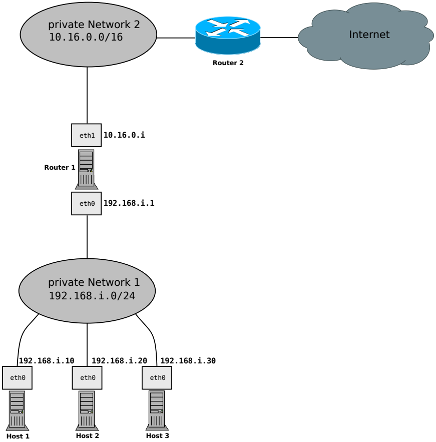
Private Network for host machines and router

Private Network 2:

`10.16.0.0/16`

Private Network spanning all networks

Network Topology – Private Network 2



Network Topology of lab exercise 1

Private Network 2:

`10.16.0.0/16`

Router 2:

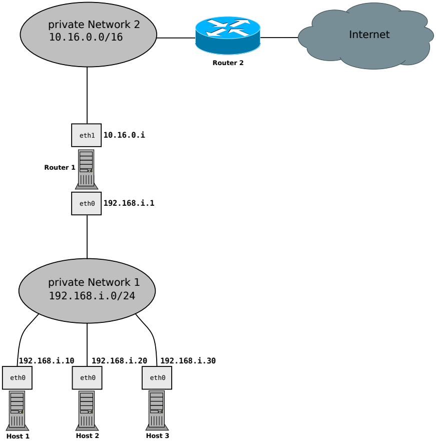
Address – `10.16.0.200`

Router 2 is the gateway for all routers in private network 1!

The route to **Router 2** needs to be configured on **Router 1**!

Is running a web server on **port 80**!

Network Topology – Private Network 1



Network Topology of lab exercise 1

Private Network 1:

`192.168.i.0/24`

Router 1:

1. Interface `eth0` – `192.168.i.1`
2. Interface `eth1` – `10.16.0.i`

Host Network:

Router 1 – `192.168.i.1`

Host 1 – `192.168.i.10`

Host 2 – `192.168.i.20`

Host 3 – `192.168.i.30`

Network Topology – Exercise 1 – Objectives

In the lab exercise you need to accomplish…

- a successful static configuration of the machines!
- working static routing on the machines!
- reachability of all machines (all hosts including **Router 1** and **2**)!
- captures of various protocols using Wireshark!

Contents

Exercise 1

Networking in Linux

Wireshark

Network Interface Names in Linux

The network interfaces in Linux can be configured with the tool `ip`

- In the literature and the internet you often find the interface names `eth0`, `eth1`,. . . , `ethX`!
- In practice the device names differ and often the naming schemes `enp1sXfX` or `enoX` can be seen![1]

[1] This is a good source of information: `https://www.freedesktop.org/wiki/Software/systemd/PredictableNetworkInterfaceNames/`

Name Resolution in Linux

- The name resolution is configured in the file `/etc/resolv.conf`!
- The entry of that file is used to resolve domain names into IP addresses![2]
- The format of the entries is `nameserver <IP ADDRESS>`!
- The most commonly known entry is `8.8.8.8` and refers to a Google name server!

[2] More detailed information on the name resolution in the lecture slides!

Commandline-tools for Networking in Linux (1/4)

The `ip` command from the `iproute2` package is a very powerful tool and replaces the deprecated `ifconfig` tool!
Its primary use cases are:

- statistics of the network links
- network configuration tasks (address assignment, ARP cache inspection, routing tables, etc.)
- configuration of static routes

Importance of the `ip` command

Please get familiar with the `ip` command and its options, since it plays a pivotal role in the configuration of the machines for lab exercise 1!

Commandline-tools for Networking in Linux (2/4)

The `ip`[3] command:

- `ip link` ... – configuration of interfaces
- `ip addr` ... – configuration of addresses
- `ip route` ... – configuration of routes

[3] The manpage of `ip` gives you the full list of functions and options!

Commandline-tools for Networking in Linux (3/4)

The `ping`[4] command is used to send ICMP packets to a host machine in the network! Its primary use cases are:

- gathering statistics on the network
- testing the reachability of a host and the network link
- first step in debugging of network errors

[4] The manpage of `ping` gives you the full list of functions and options!

Commandline-tools for Networking in Linux (4/4)

- `traceroute` – tracks the route of packets to the destination
- `curl` – a tool to transfer data over multiple protocols (HTTP, FTP, etc.)
- `ss` (`socket statistics`) – generates statistics on transport layer protocols
- `nc` (`netcat`) – listens and analyzes transport layer protocols
- `nmap` – a tool for network analysis and port scanning
- `dig` – displays the domain name lookups and the available name servers

IP forwarding in Linux routers

In order to enable routing in Linux ip forwarding needs to be activated!
This can be done by setting the Kernel parameter:

```
net.ipv4.ip_forward=1
```

alternatively

```
/proc/sys/net/ipv4/ip_forward
```

Setting Kernel parameters

The options presented above can be set by using `sysctl -w` followed by the parameter to set (here `net.ipv4.ip_forward=1`). Alternatively the parameter can be set by using `cat` and writing the value `1` to the file `/proc/sys/net/ipv4/ip_forward`!

Contents

Exercise 1

Networking in Linux

Wireshark

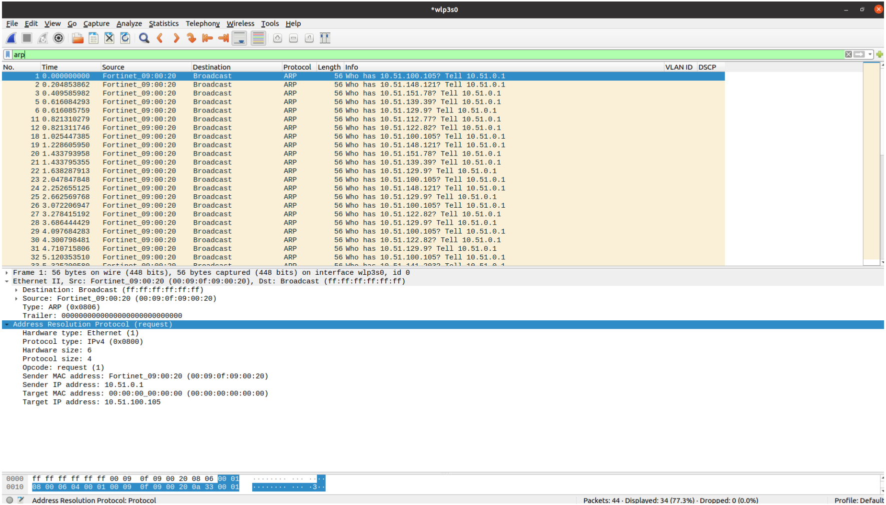
Wireshark

Wireshark is an open-source tool for network analysis

Wireshark features the following functions:

- Graphical user interface
- Collection of transmited data
- Detailed view of each packet and protocol
- Enables a detailed analysis of network traffic

Wireshark


*wlp3s0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
arp
```

```
No.      Time          Source              Destination          Protocol Length Info                                       VLAN ID DSCP
       1 0.000000000   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.100.105? Tell 10.51.0.1
       2 0.204853862   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.148.121? Tell 10.51.0.1
       3 0.409585982   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.151.78? Tell 10.51.0.1
       5 0.616084293   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.139.39? Tell 10.51.0.1
       6 0.616085759   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.129.9? Tell 10.51.0.1
      11 0.821310279   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.112.77? Tell 10.51.0.1
      12 0.821311746   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.122.82? Tell 10.51.0.1
      18 1.025447385   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.100.105? Tell 10.51.0.1
      19 1.228605950   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.148.121? Tell 10.51.0.1
      20 1.433793958   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.151.78? Tell 10.51.0.1
      21 1.433795355   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.139.39? Tell 10.51.0.1
      22 1.638287913   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.129.9? Tell 10.51.0.1
      23 2.047847848   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.100.105? Tell 10.51.0.1
      24 2.252655125   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.148.121? Tell 10.51.0.1
      25 2.662569768   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.129.9? Tell 10.51.0.1
      26 3.072206947   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.100.105? Tell 10.51.0.1
      27 3.278415192   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.122.82? Tell 10.51.0.1
      28 3.686444429   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.129.9? Tell 10.51.0.1
      29 4.097684283   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.100.105? Tell 10.51.0.1
      30 4.300798481   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.122.82? Tell 10.51.0.1
      31 4.710715806   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.129.9? Tell 10.51.0.1
      32 5.120353510   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.100.105? Tell 10.51.0.1
      33 5.325200580   Fortinet_09:00:20   Broadcast            ARP          56 Who has 10.51.141.203? Tell 10.51.0.1
```

```
▸ Frame 1: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface wlp3s0, id 0
▾ Ethernet II, Src: Fortinet_09:00:20 (00:09:0f:09:00:20), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
   ▸ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
   ▸ Source: Fortinet_09:00:20 (00:09:0f:09:00:20)
     Type: ARP (0x0806)
     Trailer: 0000000000000000000000000000
▾ Address Resolution Protocol (request)
     Hardware type: Ethernet (1)
     Protocol type: IPv4 (0x0800)
     Hardware size: 6
     Protocol size: 4
     Opcode: request (1)
     Sender MAC address: Fortinet_09:00:20 (00:09:0f:09:00:20)
     Sender IP address: 10.51.0.1
     Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
     Target IP address: 10.51.100.105
```

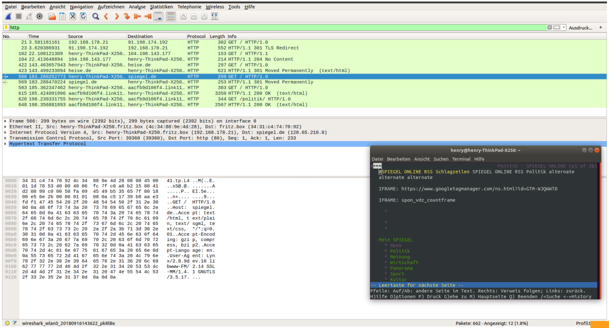
```
0000  ff ff ff ff ff ff 00 09  0f 09 00 20 08 06 00 01   ········ ··· ····
0010  08 00 06 04 00 01 00 09  0f 09 00 20 0a 33 00 01   ········ ··· ·3··
```

```
Address Resolution Protocol: Protocol          Packets: 44 · Displayed: 34 (77.3%) · Dropped: 0 (0.0%)          Profile: Default
```

Wireshark Desktop

An Example on Using Wireshark

The picture shows Wireshark collecting data for a HTTP-connection using `lynx` to access `www.heise.de`.


Datei Bearbeiten Ansicht Navigation Aufzeichnen Analyse Statistiken Telephonie Wireless Tools Hilfe
http
Ausdruck...
+

```
No.   Time          Source                 Destination            Protocol Length Info
   21 3.581181161   192.168.178.21         91.198.174.192         HTTP      302 GET / HTTP/1.0
   23 3.620386931   91.198.174.192         192.168.178.21         HTTP      552 HTTP/1.1 301 TLS Redirect
  102 22.108121309  henry-ThinkPad-X250…   104.198.143.177        HTTP      153 GET / HTTP/1.1
  104 22.413648894  104.198.143.177        henry-ThinkPad-X250…   HTTP      214 HTTP/1.1 204 No Content
  422 143.463657943 henry-ThinkPad-X250…   heise.de               HTTP      297 GET / HTTP/1.0
  423 143.499233094 heise.de               henry-ThinkPad-X250…   HTTP      621 HTTP/1.1 301 Moved Permanently  (text/html)
  566 183.260252773 henry-ThinkPad-X250…   spiegel.de             HTTP      299 GET / HTTP/1.0
  569 183.288470224 spiegel.de             henry-ThinkPad-X250…   HTTP      253 HTTP/1.1 301 Moved Permanently
  583 185.362347462 henry-ThinkPad-X250…   aacfb9d106f4.link11…   HTTP      303 GET / HTTP/1.0
  615 185.424091996 aacfb9d106f4.link11…   henry-ThinkPad-X250…   HTTP     3350 HTTP/1.1 200 OK  (text/html)
  626 198.239331755 henry-ThinkPad-X250…   aacfb9d106f4.link11…   HTTP      344 GET /politik/ HTTP/1.0
  648 198.356881893 aacfb9d106f4.link11…   henry-ThinkPad-X250…   HTTP     2567 HTTP/1.1 200 OK  (text/html)
```

```
▸ Frame 566: 299 bytes on wire (2392 bits), 299 bytes captured (2392 bits) on interface 0
▸ Ethernet II, Src: henry-ThinkPad-X250.fritz.box (4c:34:88:9e:4d:28), Dst: fritz.box (34:31:c4:74:70:92)
▸ Internet Protocol Version 4, Src: henry-ThinkPad-X250.fritz.box (192.168.178.21), Dst: spiegel.de (128.65.210.8)
▸ Transmission Control Protocol, Src Port: 39360 (39360), Dst Port: http (80), Seq: 1, Ack: 1, Len: 233
▸ Hypertext Transfer Protocol
```

```
0000  34 31 c4 74 70 92 4c 34  88 9e 4d 28 08 00 45 00   41.tp.L4 ..M(..E.
0010  01 1d 78 53 40 00 40 06  fc 7f c0 a8 b2 15 80 41   ..xS@.@. .......A
0020  d2 08 99 c0 00 50 fa 09  45 49 b5 35 65 7f 80 18   .....P.. EI.5e...
0030  00 e5 6e 2b 00 00 01 01  08 0a c5 17 39 b8 aa e3   ..n+.... ....9...
0040  fd f1 47 45 54 20 2f 20  48 54 54 50 2f 31 2e 30   ..GET /  HTTP/1.0
0050  0d 0a 48 6f 73 74 3a 20  73 70 69 65 67 65 6c 2e   ..Host:  spiegel.
0060  64 65 0d 0a 41 63 63 65  70 74 3a 20 74 65 78 74   de..Acce pt: text
0070  2f 68 74 6d 6c 2c 20 74  65 78 74 2f 70 6c 61 69   /html, t ext/plai
0080  6e 2c 20 74 65 78 74 2f  73 67 6d 6c 2c 20 74 65   n, text/ sgml, te
0090  78 74 2f 63 73 73 2c 20  2a 2f 2a 3b 71 3d 30 2e   xt/css,  */*;q=0.
00a0  30 31 0d 0a 41 63 63 65  70 74 2d 45 6e 63 6f 64   01..Acce pt-Encod
00b0  69 6e 67 3a 20 67 7a 69  70 2c 20 63 6f 6d 70 72   ing: gzi p, compr
00c0  65 73 73 2c 20 62 7a 69  70 32 0d 0a 41 63 63 65   ess, bzi p2..Acce
00d0  70 74 2d 4c 61 6e 67 75  61 67 65 3a 20 65 6e 0d   pt-Langu age: en.
00e0  0a 55 73 65 72 2d 41 67  65 6e 74 3a 20 4c 79 6e   .User-Ag ent: Lyn
00f0  78 2f 32 2e 38 2e 39 64  65 76 2e 31 36 20 6c 69   x/2.8.9d ev.16 li
0100  62 77 77 77 2d 46 4d 2f  32 2e 31 34 20 53 53 4c   bwww-FM/ 2.14 SSL
0110  2d 4d 4d 2f 31 2e 34 2e  31 20 47 4e 55 54 4c 53   -MM/1.4. 1 GNUTLS
0120  2f 33 2e 35 2e 31 37 0d  0a 0d 0a                  /3.5.17. ...
```

henry@henry-ThinkPad-X250: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe

```
#<<<                                              Politik - SPIEGEL ONLINE (p1 of 38)
   #SPIEGEL ONLINE RSS Schlagzeilen SPIEGEL ONLINE RSS Politik alternate
   alternate alternate

   IFRAME: https://www.googletagmanager.com/ns.html?id=GTM-WJQWWTD

   IFRAME: spon_vdz_countframe

     *


     *
     *


   Mein SPIEGEL
     * Menü
     * Politik
     * Meinung
     * Wirtschaft
     * Panorama
     * Sport
     * Kultur
-- Leertaste für nächste Seite --
Pfeile: Auf/Ab: andere Seite im Text. Rechts: Verweis folgen; Links: zurück.
H)ilfe O)ptionen P) Druck G)ehe zu M) Hauptseite Q) Beenden /=Suche <-=History
```

```
wireshark_wlan0_20180916143622_pkRlBx          Pakete: 662 · Angezeigt: 12 (1.8%)          Profil:
```

Data collected with Wireshark using `lynx`

Configuration of the machines

Please follow these rules:

- **Make your configurations statically! Use the tool** `ip` **exclusively!**
- **Save your static configuration on file! Use an USB-Drive for the extraction!**
- **Test your setup and document it accurately! Demonstrate it in the lab exercise!**
- **Make slides of your configurations! Use the command-line snippets, screenshots and Wireshark captures for your documentation!**

Non persistent configuration on machines

Please be aware, that the configurations on the machines are static and will be deleted after a reboot! Make sure to save your progress on an external drive!