

Development of a Download Application and Study of a Computer Network Configuration

Computer Networks, 1st Semester 2025/2026

Bachelor's in Informatics and Computing Engineering

Diogo Sousa Campeão (202307177)
Rodrigo Jorge do Rego Rocha (202305891)

Lecturer: Filipe Borges Teixeira

This report was made in the ambit of the Course Unit "Redes de Computadores" at the Faculty of Engineering of University of Porto (FEUP), and it consists of a study about the computer network existent on the netlab Laboratory at FEUP, as well as a Download Application for that network, using the FTP download protocol.

Table of Contents

1	Summary	1
2	Introduction	1
3	Development of a download application	2
3.1	Application Architecture and Relevant Functions	2
3.2	Flow of the Application	3
3.3	Report of a successful Download	3
4	Configuration and Study of a Network	4
4.1	Exp 1- Configure an IP Network	4
4.2	Exp 2 - Implement two bridges in a switch	5
4.3	Exp 3 - Configure a Router in Linux	5
4.4	Exp 4 - Configure a Commercial Router and Implement NAT	6
4.5	Exp 5 - DNS	6
4.6	Exp 6 - TCP connections	7
5	Conclusion	8
6	References	8
7	Appendix	i
7.1	Download Application Code	i
7.2	Configuration Commands	ix
7.2.1	Experience 1	ix
7.2.2	Experience 2	ix
7.2.3	Experience 3	ix
7.2.4	Experience 4	x
7.2.5	Experience 5	xi
7.2.6	Experience 6	xi
7.3	Captured Logs	xiii
7.3.1	Experience 1	xiii
7.3.2	Experience 2	xiv
7.3.3	Experience 3	xv
7.3.4	Experience 4	xviii
7.3.5	Experience 5	xxi
7.3.6	Experience 6	xxiii

List of Figures

1	Tux 3 Routes	xiii
2	Ping Tux4	xiii
3	Ping Tux4	xiv
4	Ping Tux2	xiv
5	Ping Broadcast on Tux3	xv
6	Ping Broadcast on Tux2	xv
7	Routes Tux2	xv
8	Routes Tux3	xvi
9	Routes Tux4	xvi
10	Ping on Tux2	xvi
11	Ping on Tux4's e1	xvi
12	Ping on Tux4's e2	xvii
13	Captures from Tux4's e1	xvii
14	Captures from Tux4's e2	xviii
15	Routes on Tux2	xviii
16	Routes on Tux3	xviii
17	Routes on Tux4	xix
18	Traceroute from Tux2 to Tux3, with no redirection rules applied	xix
19	Traceroute from Tux2 to Tux3, with a forced redirection on the Router	xix
20	Traceroute from Tux2 to Tux3, with ICMP redirection rules applied	xix
22	Ping on Tux2, with ICMP redirection rules applied	xx
21	Ping on Tux2, with a forced redirection on the Router	xx
23	Ping on Internet from Tux3, no NAT	xxi
24	Ping to ni.fe.up.pt from Tux2	xxi
25	Ping to youtube.com from Tux3	xxii
26	Ping to sigarra.up.pt on Tux4	xxii
27	Sucessful Download	xxiii

1 Summary

This report documents the development of an FTP download application and network configuration experiments performed in the Computer Networks laboratory at FEUP. Part 1 describes the architecture of the download application and reports a successful download with Wireshark logs showing FTP packets. Part 2 presents six experiments covering network configuration and analysis: for each experiment, we describe the network architecture, experiment objectives, main configuration commands, relevant logs, and analysis of the captured logs relevant for the learning objectives.

2 Introduction

This project was developed in the ambit of the Course Unit "Computer Networks" 2025/2026 [pequena descrição] and has the following structure:

- Part 1: Developing a Download application
- Part 2: Performing the following experiences:
 - Configure an IP Network
 - Implement two bridges in a switch
 - Configure a Router in Linux
 - Configure a Commercial Router and Implement NAT
 - DNS
 - TCP Connections
- Conclusions

3 Development of a download application

3.1 Application Architecture and Relevant Functions

The download application was done in a single file, *download.c*, and follows this architecture:

- **Url** - Data Structure to store the components of the RFC1738 standard Url of the form *ftp://[< user >:< password > @] < domain > / < file – path >*, storing the following attributes: Name, Password, Domain and File Path.
- **Decode the URL following the RFC1738 standard**

```
int decode_rfc1738(char *ftp_link, Url *ftp_url)
```

This function takes a string *ftp_link* and decodes it into the *Url* format (Username, Password, Domain, Path) in the RFC1738 standard and stores it in the *ftp_url* *Url*'s Data Structure respective fields. It checks its structure and separates each part of the *Url* carefully, returns error and ends the program in case of a wrongly formatted *Url*, and automatically assigns 'anonymous' in case username or password is blank.

- **Open and close Sockets**

```
int open_socket(const char *address, uint16_t port)
int close_socket(const int socket_fd)
```

These functions open and close the TCP socket responsible for the Download protocol and the connection to the server. They use the code already given in **clientTCP.c**.

- **Sending commands to the TCP server**

```
int command(int socket_fd, char *command, char *content)
int single_command(int socket_fd, char *command)
```

These functions are responsible for sending protocol commands to the server, and can be of two types: Functions with arguments (e.g. "USER *username*") or Functions without arguments (e.g. "PASV").

- **Reading a message from the server**

```
int message(int socket_fd, char *code, char *content)
```

Reads and parses FTP server responses. Extracts the 3 digit code and content. Handles both multi-line and single-line commands according to FTP protocol specifications. It implements a state-machine with 4 states:

1. State 0: Code reading - Reads the 3 digit response code and determines if the response is single-line or multi-line.
2. State 1: Multi-line reading - Accumulates line content until a escape character is encountered. Line content is printed at stdout.
3. State 2: Single-line reading - Accumulates line content until a espacio characters, then returns the code and the last line content.
4. State 3: Escape after multi-line - Detects the end of multi-line messages by checking if the next line is a digit (indicating a new response code).

- **Reading the PASV parameters from the server**

```
int pasv_decode(int socket_fd, char *code,
                char *address, uint8_t port[2])
```

Reads and parses passive mode (PASV) response from the FTP server. Extracts the IP address and port number for the Data Socket from the response format (h1, h2, h3, h4, h5, h6). Characters h1-h4 represent the IP Address of the download server and h5-h6 the MSB (Most significant Bytes) and LSB (Least significant Bytes) of the download port.

3.2 Flow of the Application

To compile the application, run `make` from the root of the project. For execution, run `cd bin/` to go to the bin/ directory, then run `./download` `ftp://[<user>:<password>@]<domain>/<path>`

No further inputs are necessary, and the downloaded file will be located in the `bin/` directory. The application has the following execution flow:

First, it parses the URL according to the RFC1738 standard. Then, it opens a Control Socket on Port 21 based on the procedures described in `getip.c` and `clientTCP.c`. The connection is successful by reading a **message** from the server with the code **220**. Upon successful socket opening, it sends the password via **USER <username>** and **PASS <password>**, receiving the codes **331** and **230**, respectively.

After the authentication is successful and the server access is granted, we need to switch to the **TYPE I** mode (which is the most common mode for transfers for FTP protocols) and get the **SIZE** of the file (this last one is not strictly necessary, but helps on tracking the progress of the download). The switching is successful via response code **200**, and the size is of the form **213 <file size>**.

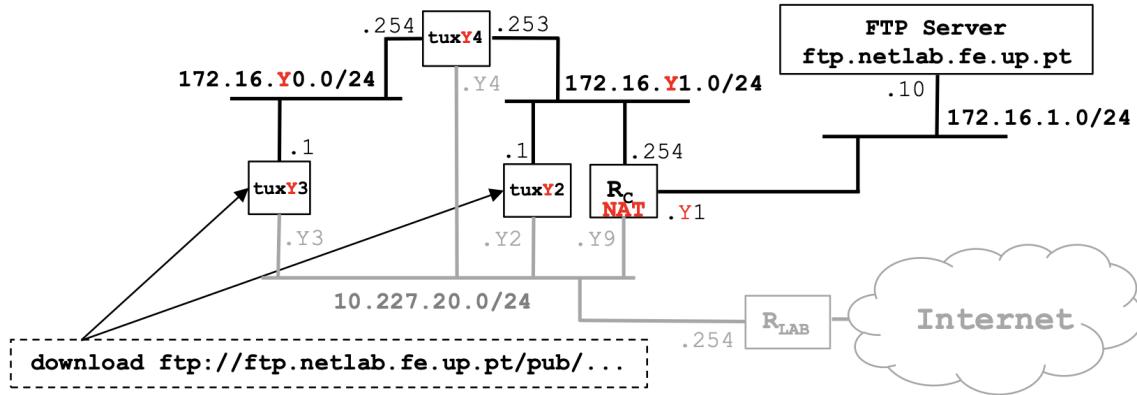
To actually transfer the files, we need to open a Data Socket in the tranfer port given by the server upon sending the **PASV** command, which is the passive mode. The server will send the response **227 (ip,ip,ip,ip,port1,port2)**. Calculating $port1 * 256 + port2$ will give us the port where the Data Socket should download the file. After this, we open a new socket on the port obtained.

By sending the **RETR** command and getting the response code **150 or 125** we start the download. The application then loops while there are bytes to read from the Data socket and writes to the file. Upon completion, it closes the Data Socket and sends the command **QUIT** to exit the FTP protocol and closes the Control Socket.

3.3 Report of a successful Download

4 Configuration and Study of a Network

This following section describes each of the experiences performed in the practical classes, explaining the procedures done and answering the questions on the Guide.



4.1 Exp 1- Configure an IP Network

The goal of this experiment is to configure the network between Tux3 and Tux4 using a Mikrotik switch, configure IP addresses, and analyze ARP tables.

The configuration commands used can be found in Appendix 7.2.1. The first step is to connect the ethernet ports from both computers to the switch. Using ifconfig, we configured IP 172.16.110.1/24 on Tux3 and 172.16.110.254/24 on Tux4. Each device has a unique MAC address (Tux3: ec:75:0c:c2:3c:7b, Tux4: ec:75:0c:c2:3c:96). The setup is now complete and we tested it by running ping between the two devices.

Using Wireshark to capture the logs, we observed that ICMP Echo Request/Reply packets are generated containing the IP addresses of source and destination, as well as the MAC addresses of the next-hop. However, since initially the computers don't know each other's MAC addresses, the source machine broadcasts an ARP request with the origin and destination IP address.

ARP (Address Resolution Protocol) packets map IP addresses to MAC addresses in a local network. This ARP request is answered by the machine with the destination IP, responding with its MAC address which is then saved in the ARP table. After the value is cached, the ARP request is not needed in future connections. In the ARP request, the source MAC and IP are from Tux3 (ec:75:0c:c2:3c:7b, 172.16.110.1), while the destination MAC is broadcast (ff:ff:ff:ff:ff:ff) and target IP is 172.16.110.254. In the ARP reply, source is Tux4 (ec:75:0c:c2:3c:96, 172.16.110.254) and destination is Tux3.

The ping command generates ARP packets (if MAC not cached), then ICMP Echo Request and Echo Reply packets. The ICMP packets have source IP 172.16.110.1 and destination IP 172.16.110.254, with source MAC ec:75:0c:c2:3c:7b (Tux3) and destination MAC ec:75:0c:c2:3c:96 (Tux4). To distinguish between packet types, we analyze the EtherType header: 0x0806 corresponds to ARP and 0x0800 to IP. For IP packets, if the Protocol field is 1, we have ICMP. The frame length can be determined from the payload size in the Ethernet header.

Finally, we noticed the ARP table contains a special value for the loopback interface (127.0.0.1), which is a software interface that allows a machine to communicate with itself. This interface is essential for testing and internal communications. These conclusions can be taken from **Appendix 7.3.1**.

4.2 Exp 2 - Implement two bridges in a switch

The goal of this experiment is to implement two bridges in the Mikrotik switch: bridge110 connecting Tux3 and Tux4, and bridge111 connecting Tux2. This allows us to learn about creating bridges using the Mikrotik switch console and verify that devices in different subnetworks cannot communicate. The configuration commands used can be found in Appendix 7.2.2.

The first step is to configure Tux2 using ifconfig commands (IP: 172.16.101.1/24), similar to the first experiment. After that, we access the Mikrotik switch console and setup the two bridges: bridge110 and bridge111. To configure bridge110, we use `/interface bridge add name=bridge110`, then remove ports from the default bridge with `/interface bridge port remove [find interface=etherN]`, and assign each device by specifying the Ethernet ports connected to the switch using `/interface bridge port add bridge=bridge110 interface=etherN`.

To check the connections between bridges, we captured logs in all computers and ran ping broadcasts in both Tux3 and Tux2. As expected, we received responses from devices in the same bridge (Tux3 can connect to Tux4), but not from devices in the other bridge (Tux3 cannot reach Tux2 and vice-versa). From this, we conclude there are two different broadcast domains: one for bridge110 (Tux3 and Tux4) and another for bridge111 (Tux2).

This is evident in the logs because broadcast packets sent from Tux3 are only visible to Tux4, not to Tux2. The logs for this experiment can be found in **Appendix 7.3.2**.

4.3 Exp 3 - Configure a Router in Linux

The goal of this experiment is to extend the configuration from Experiment 2, using Tux4 as a router between the two bridges. This gives insights on how routing works, how to connect different subnetworks, and analyze IP/MAC addresses and ARP/routing tables.

The configuration commands used can be found in Appendix 7.2.3. The setup is similar to previous experiments. We first configured a new Ethernet port on Tux4 (e2: 172.16.101.253/24) and added it to bridge111 with Tux2. To allow Tux4 to work as a router between the two subnetworks, we enabled IP forwarding using `sysctl net.ipv4.ip_forward=1` and disabled ICMP echo-ignore-broadcast.

We then added routes: in Tux3 to subnetwork 172.16.101.0/24 via gateway 172.16.100.254 (Tux4), and in Tux2 to subnetwork 172.16.100.0/24 via gateway 172.16.101.253 (Tux4). When Tux3 connects to Tux2, it connects through Tux4 (the gateway), which then forwards to Tux2. The same happens when Tux2 connects to Tux3. The setup is complete and we can now successfully ping Tux2 from Tux3 and vice-versa. The routes shown in the logs are: Tux3 has direct route to 172.16.100.0/24 and route to 172.16.101.0/24 via gateway 172.16.100.254; Tux4 has direct routes to both networks; Tux2 has direct route to 172.16.101.0/24 and route to 172.16.100.0/24 via gateway 172.16.101.253. These routes define packet forwarding: directly for local networks, through the gateway for remote networks. An entry on the forwarding table contains the destination IP address, the gateway IP (next-hop), the network mask, flags, metric, and the network interface to use.

As an additional test, we cleared the ARP tables in all computers. As expected, the results were similar, but with initial broadcast ARP requests to retrieve MAC addresses before starting the connection. The ARP messages observed at connection start are broadcasts that allow each machine to resolve MAC addresses of their next-hop connections. When Tux3 pings Tux2, Tux3 sends ARP request for 172.16.100.254 (Tux4's e1) to learn the gateway's MAC, Tux4's e1 replies with its MAC (ec:75:0c:c2:51:4d), then Tux4's e2 sends ARP request for Tux2, and Tux2 replies with its MAC (ec:75:0c:c2:17:8b).

We observed ICMP Echo Request packets from Tux3 to Tux2 and ICMP Echo Reply packets from Tux2 to Tux3, generated by ping to test connectivity. By analyzing the ICMP packets, we see they have constant IP addresses (Tux3 to Tux2), but MAC addresses change at each

hop: from Tux3 to Tux4 the source MAC is ec:75:0c:c2:3c:f5 (Tux3) and destination MAC is ec:75:0c:c2:51:4d (Tux4 e1); from Tux4 to Tux2 the source MAC is ec:75:0c:c2:17:2a (Tux4 e2) and destination MAC is ec:75:0c:c2:17:8b (Tux2). This demonstrates that IP addresses work end-to-end (Network layer) while MAC addresses work hop-by-hop (Data Link layer). The logs for this experiment can be found in **Appendix 7.3.3**.

4.4 Exp 4 - Configure a Commercial Router and Implement NAT

The goal of this experiment is to continue with the configuration from Experiment 3, adding a MikroTik router connected to bridge111 (containing Tux2 and Tux4). This router connects the network to the Internet using NAT, and routes on all Tuxes are updated accordingly. This allows us to learn how to configure the Mikrotik router, connect to the Internet, implement NAT and expand knowledge about routing. The configuration commands used can be found in Appendix 7.2.4.

The first step is to connect the router to the lab network (ether1: 172.16.1.110/24) with NAT enabled by default, then connect it to the MikroTik switch (ether2: 172.16.111.254/24). We use the switch console to connect the router to bridge111 with Tux2 and Tux4. To configure a static route in the commercial router, we use `/ip route add dst-address=172.16.110.0/24 gateway=172.16.111.253`, which tells the router **how to reach networks that aren't directly connected**. We then configured all routes in the network: Tux3 routes to 172.16.111.0/24 and 172.16.1.0/24 via gateway 172.16.110.254 (Tux4); Tux4 routes to 172.16.1.0/24 via gateway 172.16.111.254 (Router); Tux2 routes to 172.16.1.0/24 via gateway 172.16.111.254 (Router) and to 172.16.110.0/24 via gateway 172.16.111.253 (Tux4); Router routes to 172.16.110.0/24 via gateway 172.16.111.253 (Tux4). This setup allows pinging any connection in the network from any machine.

The first additional test consisted of changing Tux2's configuration: we disabled ICMP redirects using `sysctl net.ipv4.conf.if_e1.accept_redirects=0` and changed the route to 172.16.110.0/24 from gateway Tux4 to gateway Router. With this setup, when we ping Tux3 from Tux2, packets follow: **Tux2 >Router >Tux4 >Tux3 (3 hops)**. Despite a shorter route existing, since redirects are disabled, packets follow the longer path.

After enabling redirects with `accept_redirects=1`, the Router sends an ICMP Redirect message to Tux2 and packets change route to: **Tux2 >Tux4 >Tux3 (2 hops)**. These conclusions can be analyzed with traceroute.

The other test consisted of pinging the FTP server from Tux3 with and without NAT. NAT (Network Address Translation) **translates private IP addresses to a public IP address, allowing multiple devices to share a single public IP**. To configure NAT in MikroTik, we use `/ip firewall nat disable 0`. With NAT enabled, the ping works correctly because the source IP is translated to 172.16.1.110 (routable). With NAT disabled, the packet reaches the server with source IP 172.16.110.1, **but the response doesn't reach Tux3 since the server cannot route back to the private IP address**. The logs for this experiment can be found in **Appendix 7.3.4**.

4.5 Exp 5 - DNS

The goal of this experiment is to configure the Domain Name System (DNS) in all machines within the network. This makes it possible to understand how DNS works and how hostnames are resolved into IP addresses based on the netlab DNS service. The configuration commands used can be found in Appendix 7.2.5.

To configure the DNS service in a host, we edit `/etc/resolv.conf` and add `nameserver 10.227.20.3`. This configures DNS on all machines, allowing translation between human-readable names (like youtube.com) into public IP addresses used at the Network level.

When a human-readable address is accessed, DNS exchanges two types of packets using UDP port 53: a DNS Query request is sent to the nameserver to translate it, and the response is sent in a DNS Query Response with the result. Both packets transport: Header (Transaction ID for matching responses, Flags indicating query/response and recursion status, Questions count, Answers count, Authority, Additional), Queries section (Name containing the domain name, Type indicating DNS record type like A for IPv4 or AAAA for IPv6, and Class).

The DNS Query Response additionally contains the Answers section with Resource Records including Time To Live (seconds the record can live), Data Length and Data (the resolved IP addresses). The complete logs can be analyzed in **Appendix 7.3.5**.

4.6 Exp 6 - TCP connections

The goal of this experiment is to test the Download Application developed by downloading a file on Tux3 from the Netlab FTP server. This allows us to review concepts from the FTP download application, verify the network configuration is correct, and understand the final architecture and what TCP packets are exchanged. The configuration commands used can be found in Appendix 7.2.6.

The first part consisted of compiling the download application in Tux3 and downloading a file from the FTP server, verifying it arrives correctly. By analyzing the logs, we notice that the FTP application opens two TCP connections: one for control on port 21 (transporting FTP commands like USER, PASS, TYPE, PASV, RETR, QUIT and server responses with status codes), and one for data transfer on a dynamic port provided by the PASV command response, used exclusively for file content.

The application then exchanges commands (login, obtain file size,...) with the server and eventually receives information needed to connect the second port.

We can easily view the 3 phases of a TCP connection through the logs: Connection Establishment with three-way handshake (SYN with initial sequence number, SYN-ACK acknowledging and sending server's sequence number, ACK completing the handshake), Data Transfer with bidirectional exchange using sequence numbers and acknowledgments, and Connection Termination with four-way handshake (FIN, ACK, FIN, ACK).

From the logs, we observed the Automatic Repeat Request (ARQ) mechanism which is an error control process used in TCP to reliably transmit data packets.

The relevant TCP fields are: Sequence Number (indicates byte position in data stream), Acknowledgment Number (indicates next expected byte), Window Size (receiver's buffer space for flow control), and Checksum (detects transmission errors). ARQ is based on acknowledgments, timeouts and packet retransmissions. In the logs we can observe sequence numbers incrementing by segment size, ACK numbers confirming receipt, duplicate ACKs indicating potential loss, and retransmitted packets marked by Wireshark.

We also viewed how TCP Congestion Control works using the relevant congestion window (cwnd) maintained internally. Initially, the transfer window starts small (Slow Start phase) and doubles each RTT (exponential growth), then gradually increases by 1 segment per RTT (Congestion Avoidance, additive increase) while no errors occur. As soon as packet loss occurs, the window size is halved (multiplicative decrease) to reduce congestion.

The best way to analyze this is through the throughput graph which has a saw-tooth pattern due to steady increases and drastic decreases. Our throughput evolved according to TCP congestion control: rapid increase during slow start, stabilization during congestion avoidance, and drops upon packet loss followed by recovery.

The second part investigates how throughput is disturbed when two machines download files simultaneously. This experiment was not performed due to impossibility of the laboratory. In theory, we would've started a download in Tux3 and, in the middle of the transfer, started another in Tux2. If the experiment was conducted as planned, the throughput of the first TCP connection would be significantly reduced when the second TCP connection is initiated.

Both connections compete for bandwidth and TCP's fairness mechanism causes them to converge toward equal bandwidth shares. The bandwidth is approximately shared equally between both connections, with Tux3's throughput dropping by roughly 50%, demonstrating TCP's fair sharing principle in action.

5 Conclusion

In conclusion, this project allowed us to deepen our understanding of fundamental computer network concepts, such as MAC and IP addresses, bridges, switches, routers, ARP and route tables, Network Address Translation, Domain Name Systems, TCP protocol and a general overview of computer network configuration with focus on the Network layer. Moreover, developing an FTP download application provided hands-on experience with sockets, C libraries for network interfacing and FTP server connections.

This project was also very useful to gain practical skills by configuring the network with appropriate commands and connections. Furthermore, using Wireshark to capture the logs was essential to gain insights with common development tools in this field.

Overall, the project successfully met its objectives, giving us practical insights into network protocols and computer network configuration.

6 References

1. RFC 1738 - Uniform Resource Locators (URL). T. Berners-Lee, L. Masinter, M. McCahill. December 1994. Available at: <https://www.ietf.org/rfc/rfc1738.txt>
2. RFC 959 - File Transfer Protocol (FTP). J. Postel, J. Reynolds. October 1985. Available at: <https://www.ietf.org/rfc/rfc959.txt>
3. RFC 793 - Transmission Control Protocol. Information Sciences Institute. September 1981. Available at: <https://www.ietf.org/rfc/rfc793.txt>
4. Computer Networks Laboratory Guide, FEUP, 2025/2026
5. Wireshark User's Guide. Available at: <https://www.wireshark.org/docs/>
6. MikroTik RouterOS Documentation. Available at: <https://help.mikrotik.com/>
7. Stevens, W. R., Wright, G. R. (1994). TCP/IP Illustrated, Volume 1: The Protocols. Addison-Wesley Professional.
8. Kurose, J. F., Ross, K. W. (2021). Computer Networking: A Top-Down Approach (8th Edition). Pearson.

7 Appendix

7.1 Download Application Code

```

1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4 #include <netdb.h>
5 #include <netinet/in.h>
6 #include <arpa/inet.h>
7 #include <sys/socket.h>
8 #include <unistd.h>
9 #include <ctype.h>
10 #include <fcntl.h>
11
12
13 #define FTP_PORT 21
14 #define MAX_LEN 1024
15 #define TRANSFER_LEN 2048
16
17 typedef struct {
18     char username[MAX_LEN+1];
19     char password[MAX_LEN+1];
20     char domain[MAX_LEN+1];
21     char path[MAX_LEN+1];
22 } Url;
23
24 // ftp://[<user>:<password>@]<host>/<url-path>
25 int decode_rfc1738(char *ftp_link, Url *ftp_url){
26     if(strlen(ftp_link) > MAX_LEN){
27         printf("URL maximum size exceeded\n");
28         return -1;
29     }
30
31     char *schema = strtok(ftp_link, ":");
32     if (schema == NULL) {
33         printf("No schema provided.\n");
34         return -1;
35     }
36     if(strcmp(schema, "ftp")) {
37         printf("Wrong protocol selected.\nCorrect link format: ftp://[<user>:<password>@]<host>/<url-path>\n");
38         return -1;
39     }
40
41     char *url = strtok(NULL, " ");
42     if(url == NULL){
43         printf("No URL provided\n");
44         return -1;
45     }
46     if (url[0] != '/' || url[1] != '/') {
47         printf("Wrong URL format.\nCorrect link format: ftp://[<user>:<password>@]<host>/<url-path>\n");
48         return -1;
49     }
50     url = url+2;
51
52     char *at = strchr(url, '@');
53     char *start;
54     if (at) {
55         char *div = strchr(url, ':');
56
57         if (div) { // username:password@url

```

```

58     strncpy(ftp_url->username, url, div-url);
59     strncpy(ftp_url->password, div+1, at-div-1);
60
61     ftp_url->username[div-url] = '\0';
62     ftp_url->password[at-div-1] = '\0';
63 }
64 else { // username@url
65     strncpy(ftp_url->username, url, at-url);
66     strcpy(ftp_url->password, "anonymous");
67
68     ftp_url->username[at-url] = '\0';
69 }
70
71     start = at +1;
72 }
73 else { // url
74     strcpy(ftp_url->username, "anonymous");
75     strcpy(ftp_url->password, "anonymous");
76
77     start = url;
78 }
79
80 if (strchr(start, ':')){
81     printf("Bad URL.\n");
82     return -1;
83 }
84
85 char *first_slash = strchr(start, '/');
86 char *end = strchr(start, '\0');
87 if (first_slash &&
88     (first_slash-start > 0) )
89 {
90     strncpy(ftp_url->domain, start, first_slash-start);
91     strncpy(ftp_url->path, first_slash+1, end-first_slash);
92
93     ftp_url->domain[first_slash-start] = '\0';
94     ftp_url->path[end-first_slash] = '\0';
95 }
96 else {
97     printf("Bad URL\n");
98     return -1;
99 }
100
101 return 0;
102 }
103
104 int open_socket(const char *address, uint16_t port){
105     int socket_fd = -1;
106     struct sockaddr_in server_addr;
107
108     bzero((char *) &server_addr, sizeof(server_addr));
109     server_addr.sin_family = AF_INET;
110     server_addr.sin_addr.s_addr = inet_addr(address);
111     server_addr.sin_port = htons(port);
112
113     if ((socket_fd = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
114         perror("socket()");
115         exit(-1);
116     }
117
118     if (connect(socket_fd, (struct sockaddr *) &server_addr, sizeof(server_addr))
119 ) < 0) {
120         perror("connect()");
121     }
122 }
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169

```

```

120         exit(-1);
121     }
122
123     printf("Connected to %s:%d\n", address, port);
124     return socket_fd;
125 }
126
127 int close_socket(const int socket_fd){
128     if (close(socket_fd) < 0){
129         perror("close()");
130         exit(-1);
131     }
132     return 0;
133 }
134
135 int message(int socket_fd, char *code, char *content){
136     char c;
137     char line[MAX_LEN];
138     memset(line, 0, sizeof(line));
139     int code_index = 0;
140     int line_index = 0;
141     int first_line = 1;
142     int state = 0; // 0 code, 1 multi-line, 2 single-line, 3 escape
143
144     while (1) {
145         ssize_t n = read(socket_fd, &c, 1);
146         if (n == 0) {
147             fprintf(stderr, "Connection closed by peer\n");
148             return -1;
149         } else if (n < 0) {
150             perror("read()");
151             return -1;
152         }
153
154         switch(state){
155             case 0:
156                 if (code_index < 3) {
157                     code[code_index++] = c;
158                 }
159                 else if (c == '\r'){
160                     if (line_index < MAX_LEN - 1) line[line_index++] = c;
161                     code[3] = '\0';
162                     state = 1;
163                 }
164                 else if (c == '\n'){
165                     if (line_index < MAX_LEN - 1) line[line_index++] = c;
166                     code[3] = '\0';
167                     state = 2;
168                 }
169                 break;
170             case 1:
171                 if (c == '\n'){
172                     if (first_line){
173                         printf("> %s%s\n", code, line);
174                         first_line = 0;
175                     }
176                     else printf("> %s\n", line);
177
178                     memset(line, 0, sizeof(line));
179                     line_index = 0;
180                     state = 3;
181                 }
182             else {

```

```

183             if (line_index < MAX_LEN - 1) line[line_index++] = c;
184         }
185         break;
186     case 2:
187         if (c == '\n'){
188             strncpy(content, line + 1, MAX_LEN-1);
189             memset(line, 0, sizeof line);
190             return 0;
191         }
192         else {
193             if (line_index < MAX_LEN - 1) line[line_index++] = c;
194             break;
195         }
196         break;
197     case 3:
198         if (isdigit(c)){
199             memset(line, 0, sizeof(line));
200             memset(code, 0, MAX_LEN);
201
202             code_index = 1;
203             line_index = 0;
204
205             code[0] = c;
206             first_line = 1;
207
208             state = 0;
209         }
210         else {
211             if (line_index < MAX_LEN - 1) line[line_index++] = c;
212             state = 1;
213         }
214     }
215 }
216 return -1;
217 }

218 int pasv_decode(int socket_fd, char *code, char *address, uint8_t port[2]){
219     char c;
220     char line[MAX_LEN];
221     memset(line, 0, MAX_LEN);
222     int line_index = 0;
223
224     while (1){
225         ssize_t n = read(socket_fd, &c, 1);
226         if (n == 0) {
227             fprintf(stderr, "Connection closed by peer\n");
228             return -1;
229         } else if (n < 0) {
230             perror("read()");
231             return -1;
232         }
233
234         if (line_index < MAX_LEN - 1) line[line_index++] = c;
235
236         if (c == '\n') {
237             line[line_index] = '\0';
238             break;
239         }
240     }
241
242     strncpy(code, line, 3);
243     code[3] = '\0';
244

```

```

246     int h1, h2, h3, h4, p1, p2;
247     if (sscanf(line, "%*d %*[^\n](%d,%d,%d,%d,%d)", &h1, &h2, &h3, &h4, &p1,
248     &p2) == 6) {
249         sprintf(address, "%d.%d.%d.%d", h1, h2, h3, h4);
250         port[0] = p1;
251         port[1] = p2;
252         return 0;
253     } else {
254         fprintf(stderr, "Failed to parse PASV response\n");
255         return -1;
256     }
257
258     return -1;
259 }
260
261 int command(int socket_fd, char *command, char *content){
262     if (write(socket_fd, command, strlen(command)) < 0) return -1;
263     if (write(socket_fd, content, strlen(content)) < 0) return -1;
264     if (write(socket_fd, "\r\n", 2) < 0) return -1;
265     return 0;
266 }
267
268 int single_command(int socket_fd, char *command){
269     if (write(socket_fd, command, strlen(command)) < 0) return -1;
270     if (write(socket_fd, "\r\n", 2) < 0) return -1;
271     return 0;
272 }
273
274 int check_response_code(char *code, char *expected){
275     return strncmp(code, expected, strlen(expected));
276 }
277
278 int main(int argc, char **argv){
279     if (argc < 2){
280         printf("Usage: %s <ftp_url>\n", argv[0]);
281         return -1;
282     }
283
284     Url url;
285
286     if (decode_rfc1738(argv[1], &url) != 0){
287         printf("ERROR: Couldn't parse URL.\n");
288         return -1;
289     }
290
291     printf("Username : %s\n", url.username);
292     printf("Password : %s\n", url.password);
293     printf("Domain   : %s\n", url.domain);
294     printf("Path      : %s\n", url.path);
295
296     struct hostent *host = gethostbyname(url.domain);
297     if (host == NULL){
298         printf("ERROR: Host could not be found.\n");
299         return -1;
300     }
301
302     const char *ip_address = inet_ntoa(*((struct in_addr *) host->h_addr));
303
304     printf("Host name  : %s\n", host->h_name);
305     printf("IP Address : %s\n", ip_address);
306
307     int socket;
308     printf("Connecting to %s\n", ip_address);

```

```

308     socket = open_socket(ip_address, FTP_PORT);
309     printf("Socket File descriptor: %d\n\nAttempting Connection...\\n\\n", socket
310 );
311
312     char response_code[MAX_LEN], content[MAX_LEN];
313     if (message(socket, response_code, content) != 0){
314         printf("Unable to connect to %s\\n", ip_address);
315         close(socket);
316         return -1;
317     }
318     else {
319         printf("> %s %s\\n\\n", response_code, content);
320         printf("\\nConnection Successful.\\nStarting Authentication.\\n\\n");
321     }
322
323
324
325     printf("Sending USER %s\\n", url.username);
326     if(command(socket, "USER ", url.username) != 0
327         || message(socket, response_code, content) != 0
328         || check_response_code(response_code, "331") != 0){
329         printf("Could not send command. Aborting.\\n");
330         close_socket(socket);
331         return -1;
332     }
333     printf("> %s %s\\n\\n", response_code, content);
334
335     printf("Sending PASS %s\\n", url.password);
336     if(command(socket, "PASS ", url.password) != 0
337         || message(socket, response_code, content) != 0
338         || check_response_code(response_code, "230") != 0){
339         printf("Could not send command. Aborting.\\n");
340         close_socket(socket);
341         return -1;
342     }
343     printf("> %s %s\\n\\n", response_code, content);
344
345     printf("Sending TYPE I\\n");
346     if(single_command(socket, "TYPE I") !=0
347         || message(socket, response_code, content) != 0
348         || check_response_code(response_code, "200") != 0){
349         printf("Could not enter Binary mode. Aborting\\n");
350         close_socket(socket);
351         return -1;
352     }
353     printf("> %s %s\\n\\n", response_code, content);
354
355     printf("Sending SIZE %s\\n", url.path);
356     if(command(socket, "SIZE ", url.path) != 0
357         || message(socket, response_code, content) != 0
358         || check_response_code(response_code, "213") != 0){
359         printf("Could not send command. Aborting.\\n");
360         close_socket(socket);
361         return -1;
362     }
363     printf("> %s %s\\n\\n", response_code, content);
364
365     long file_size = atol(content);
366     printf("File Size: %ld bytes\\n", file_size);
367
368     char address[16];
369     uint8_t port[2];

```

```

370     printf("Sending PASV\n");
371     if(single_command(socket, "PASV") !=0
372         || pasv_decode(socket, response_code, address, port) != 0
373         || check_response_code(response_code, "227") != 0){
374         printf("Could not enter Binary mode. Aborting\n");
375         close_socket(socket);
376         return -1;
377     }
378     printf("> %s %s\n\n", response_code, content);
379
380     int data_socket = open_socket(address, port[0]*256+port[1]);
381
382     char *filename = strrchr(url.path, '/');
383     if (filename) filename++;
384     else filename = url.path;
385
386     int file = open(filename, O_WRONLY | O_CREAT, 0640);
387
388     uint8_t buffer[TRANSFER_LEN];
389     size_t bytes_read = 0, total = 0;
390
391     printf("Sending RETR %s\n", url.path);
392     if(command(socket, "RETR ", url.path) != 0
393         || message(socket, response_code, content) != 0
394         || (check_response_code(response_code, "150") != 0
395             && check_response_code(response_code, "125") != 0)){
396         printf("Could not send command. Aborting.\n");
397         close_socket(socket);
398         return -1;
399     }
400     printf("> %s %s\n\n", response_code, content);
401
402     while((bytes_read = read(data_socket, buffer, TRANSFER_LEN)) > 0){
403         if (write(file, buffer, bytes_read) < 0) {
404             perror("write()");
405             break;
406         }
407
408         total += bytes_read;
409         printf("\rProgress: %d%% (%zu / %ld)", (int)((total * 100) / file_size)
410             , total, file_size);
411         fflush(stdout);
412     }
413     printf("\n\n");
414     close(file);
415
416     printf("Transfer complete. Closing Data Socket %d\n", socket);
417     close_socket(data_socket);
418
419     if(message(socket, response_code, content) != 0
420         || check_response_code(response_code, "226") != 0){
421         printf("Download Failed.\n");
422         close_socket(socket);
423         return -1;
424     }
425     printf("> %s %s\n\n", response_code, content);
426
427     printf("Sending QUIT\n");
428     if(single_command(socket, "QUIT") != 0
429         || message(socket, response_code, content) != 0
430         || check_response_code(response_code, "221") != 0){
431         printf("Could not send command. Aborting.\n");
        close_socket(socket);
    }

```

```
432     return -1;
433 }
434 printf("> %s %s\n\n", response_code, content);
435
436 printf("Closing Control Socket %d\n", socket);
437 close_socket(socket);
438
439 printf("Socket closed. Protocol Complete.\n");
440
441 return 0;
442 }
```

7.2 Configuration Commands

7.2.1 Experience 1

```
# Configure IP on Tux3
netedu@tux113:~$ sudo ifconfig if_e1 172.16.110.1/24

# Configure IP on Tux4
netedu@tux114:~$ sudo ifconfig if_e1 172.16.110.254/24

# Ping Tux4
netedu@tux113:~$ sudo ping 172.16.110.254

# Ping Tux3
netedu@tux114:~$ sudo ping 172.16.110.1

# Inspect Forwarding and ARP
netedu@tux113:~$ sudo route -n
netedu@tux113:~$ sudo arp -a
```

7.2.2 Experience 2

```
# Configure IP on Tux2
netedu@tux112:~$ sudo ifconfig if_e1 172.16.111.1/24

# Add Bridge 110 and 111
[admin@MikroTik] > /interface bridge add name=bridge110
[admin@MikroTik] > /interface bridge add name=bridge111

# Remove ports from default bridge
[admin@MikroTik] > /interface bridge port remove [find interface=]ether8
[admin@MikroTik] > /interface bridge port remove [find interface=]ether9
[admin@MikroTik] > /interface bridge port remove [find interface=]ether10

# Add Tux3 and Tux4 to Bridge110
# (/interface bridge port ) is ommited
[admin@MikroTik] > /... add bridge=bridge110 interface=ether9
[admin@MikroTik] > /... add bridge=bridge110 interface=ether10

# Add Tux2 to Bridge111
# (/interface bridge port ) is ommited
[admin@MikroTik] > /... add bridge=bridge110 interface=ether8

# Ping Tux4 and Tux2
netedu@tux113:~$ sudo ping 172.16.110.254 (Connection exists)
netedu@tux113:~$ sudo ping 172.16.111.1 (Connection does not exist)

# Ping Broadcast in Tux3
netedu@tux113:~$ sudo ping -b 172.16.110.255

# Ping Broadcast in Tux2
netedu@tux112:~$ sudo ping -b 172.16.111.255
```

7.2.3 Experience 3

```
# Configure e2 from Tux4
netedu@tux112:~$ sudo ifconfig if_e2 172.16.111.253/24

# Add Tux2 to Bridge110
# (/interface bridge port ) is ommited
[admin@MikroTik] > /... add bridge=bridge110 interface=ether11

# Enable IP Forwarding and Disable Echo-Ignore-Broadcasts
netedu@tux114:~$ sudo sysctl net.ipv4.ip_forward=1
netedu@tux114:~$ sudo sysctl net.ipv4.icmp_echo_ignore_broadcasts=0

# Add routes
# From Tux2 to Tux3
netedu@tux113:~$ sudo route add -net 172.16.110.1/24 gw 172.16.111.253

# From Tux3 to Tux2
netedu@tux113:~$ sudo route add -net 172.16.111.0/24 gw 172.16.110.254

# View Routes
netedu@tux112:~$ sudo route -n
netedu@tux113:~$ sudo route -n
netedu@tux114:~$ sudo route -n

# From Tux3 ping other interfaces
netedu@tux113:~$ sudo ping 172.16.110.254 (Connection exists)
netedu@tux113:~$ sudo ping 172.16.111.1 (Connection exists)
netedu@tux113:~$ sudo ping 172.16.111.253 (Connection exists)

# Clean ARP tables
netedu@tux112:~$ sudo arp -d 172.16.111.254

netedu@tux113:~$ sudo arp -d 172.16.110.1

netedu@tux114:~$ sudo arp -d 172.16.110.254
netedu@tux114:~$ sudo arp -d 172.16.111.253

# From Tux3 ping Tux2 (First Packet should take more time)
netedu@tux113:~$ sudo ping 172.16.111.1
```

7.2.4 Experience 4

```
# Configure IP Address of the Router (Router Console)
[admin@MikroTik] > /ip address add address=172.16.1.110/24 interface=ether1
[admin@MikroTik] > /ip address add address=172.16.111.254/24 interface=ether2

# Remove the Router from default bridge
[admin@MikroTik] > /interface bridge port remove [find interface=]ether12

# Add the router to Bridge111
# (/interface bridge port ) is ommited
[admin@MikroTik] > /... add bridge=bridge110 interface=ether12

# Add Routes
# From Tux2 to the Internet via Router
netedu@tux112:~$ sudo route add -net 172.16.1.0/24 gw 172.16.111.254
```

```

# From Tux3 to the Internet via Router
netedu@tux113:~$ sudo route add -net 172.16.1.0/24 gw 172.16.110.254

# From Tux4 to the Internet via Router
netedu@tux114:~$ sudo route add -net 172.16.1.0/24 gw 172.16.111.254

# Redirect ping from Tux2 to Tux3, via Router instead of Tux4
netedu@tux112:~$ sudo sysctl net.ipv4.conf.if_e1.accept_redirects=0
netedu@tux112:~$ sudo sysctl net.ipv4.conf.all.accept_redirects=0

# Disable the route from Tux2 to Tux3 gateway Tux4
netedu@tux112:~$ sudo route del -net 172.16.110.0/24 gw 172.16.111.253

# Add the route from Tux2 to Tux3 gateway Router
netedu@tux112:~$ sudo route add -net 172.16.110.0/24 gw 172.16.111.254

# Ping Tux3 from Tux2 (should receive a lot of Redirect Host messages)
netedu@tux112:~$ sudo ping 172.16.110.1

# Traceroute Tux3 (should have 3 lines)
netedu@tux112:~$ sudo traceroute 172.16.110.1

# Revert the changes
netedu@tux112:~$ sudo route del -net 172.16.110.0/24 gw 172.16.111.254
netedu@tux112:~$ sudo route add -net 172.16.110.0/24 gw 172.16.111.253
netedu@tux112:~$ sudo sysctl net.ipv4.conf.if_e1.accept_redirects=1
netedu@tux112:~$ sudo sysctl net.ipv4.conf.all.accept_redirects=1

# Ping Tux3
netedu@tux112:~$ sudo ping 172.16.110.1

# Traceroute Tux3 (should have 2 lines)
netedu@tux112:~$ sudo traceroute 172.16.110.1

# Ping FTP Server (netlab.fe.up.pt)
netedu@tux112:~$ sudo ping 172.16.110.1

# Disable NAT (Router Console)
[admin@MikroTik] > /ip firewall nat disable=0

# Ping FTP Server (netlab.fe.up.pt) (Should not Work)
netedu@tux112:~$ sudo ping 172.16.110.1

# Enable NAT (Router Console)
[admin@MikroTik] > /ip firewall nat enable=0

```

7.2.5 Experience 5

```

netedu@tux112:~$ sudo ping ni.fe.up.pt
netedu@tux113:~$ sudo ping youtube.com
netedu@tux114:~$ sudo ping sigarra.up.pt

```

7.2.6 Experience 6

```

netedu@tux113:~$ make
netedu@tux113:~$ cd bin/

```

```
netedu@tux113:~$ ./download ftp://ftp.netlab.fe.up.pt/pipe.txt
```

7.3 Captured Logs

7.3.1 Experience 1

```
netedu@tux113:~$ sudo route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
0.0.0.0          10.227.20.254   0.0.0.0        UG    100    0        0 if_mng
10.227.20.0      0.0.0.0        255.255.255.0  U     100    0        0 if_mng
172.16.110.0     0.0.0.0        255.255.255.0  U     0      0        0 if_e1
```

Figure 1: Tux 3 Routes

```
netedu@tux113:~$ sudo arp -a
dns.netlab.fe.up.pt (10.227.20.2) at bc:24:11:f5:a2:54 [ether] on if_mng
posto11.netlab.fe.up.pt (10.227.20.115) at 20:7c:14:a1:64:e8 [ether] on if_mng
_gateway (10.227.20.254) at e4:8d:8c:20:25:c8 [ether] on if_mng
fog.netlab.fe.up.pt (10.227.20.223) at bc:24:11:f8:0d:71 [ether] on if_mng
ns1.netlab.fe.up.pt (10.227.20.3) at bc:24:11:e7:5e:5b [ether] on if_mng
? (172.16.110.254) at ec:75:0c:c2:3c:96 [ether] on if_e1
```

46	86.038041157	TPLink_c2:3c:7b	Broadcast	ARP	42 Who has 172.16.110.254? Tell 172.16.110.1
47	86.038182385	TPLink_c2:3c:96		ARP	60 172.16.110.254 is at ec:75:0c:c2:3c:96
48	86.038190696	172.16.110.1	172.16.110.254	ICMP	98 Echo (ping) request id=0x42af, seq=1/256, ttl=64 (reply in 49)
49	86.038295663	172.16.110.254	172.16.110.1	ICMP	98 Echo (ping) reply id=0x42af, seq=1/256, ttl=64 (request in 48)
50	86.093019210	Routerboardc_1c:8d:2c	Nearest-Customer-Bridge	STP	60 RST, Root = 32768/0:c4:ad:34:1c:8d:2b Cost = 0 Port = 0x8002
51	87.050566465	172.16.110.1	172.16.110.254	ICMP	98 Echo (ping) request id=0x42af, seq=2/512, ttl=64 (reply in 52)
52	87.050760157	172.16.110.254	172.16.110.1	ICMP	98 Echo (ping) reply id=0x42af, seq=2/512, ttl=64 (request in 51)
53	88.074568533	172.16.110.1	172.16.110.254	ICMP	98 Echo (ping) request id=0x42af, seq=3/768, ttl=64 (reply in 54)
54	88.074762011	172.16.110.254	172.16.110.1	ICMP	98 Echo (ping) reply id=0x42af, seq=3/768, ttl=64 (request in 53)
55	88.095165928	Routerboardc_1c:8d:2c	Nearest-Customer-Bridge	STP	60 RST, Root = 32768/0:c4:ad:34:1c:8d:2b Cost = 0 Port = 0x8002
56	89.098558925	172.16.110.1	172.16.110.254	ICMP	98 Echo (ping) request id=0x42af, seq=4/1024, ttl=64 (reply in 57)
57	89.098753285	172.16.110.254	172.16.110.1	ICMP	98 Echo (ping) reply id=0x42af, seq=4/1024, ttl=64 (request in 56)
58	90.097280162	Routerboardc_1c:8d:2c	Nearest-Customer-Bridge	STP	60 RST, Root = 32768/0:c4:ad:34:1c:8d:2b Cost = 0 Port = 0x8002
59	90.122541096	172.16.110.1	172.16.110.254	ICMP	98 Echo (ping) request id=0x42af, seq=5/1280, ttl=64 (reply in 60)
60	90.122781290	172.16.110.254	172.16.110.1	ICMP	98 Echo (ping) reply id=0x42af, seq=5/1280, ttl=64 (request in 59)
61	91.146563587	172.16.110.1	172.16.110.254	ICMP	98 Echo (ping) request id=0x42af, seq=6/1536, ttl=64 (reply in 62)
62	91.146763367	172.16.110.254	172.16.110.1	ICMP	98 Echo (ping) reply id=0x42af, seq=6/1536, ttl=64 (request in 61)
63	91.172487694	TPLink_c2:3c:96	TPLink_c2:3c:7b	ARP	60 Who has 172.16.110.1? Tell 172.16.110.254
64	91.172502157	TPLink_c2:3c:7b	TPLink_c2:3c:96	ARP	42 172.16.110.1 is at ec:75:0c:c2:3c:7b
65	92.098668814	Routerboardc_1c:8d:2c	Nearest-Customer-Bridge	STP	60 RST, Root = 32768/0:c4:ad:34:1c:8d:2b Cost = 0 Port = 0x8002
66	92.170571319	172.16.110.1	172.16.110.254	ICMP	98 Echo (ping) request id=0x42af, seq=7/1792, ttl=64 (reply in 67)
67	92.170762386	172.16.110.254	172.16.110.1	ICMP	98 Echo (ping) reply id=0x42af, seq=7/1792, ttl=64 (request in 66)
68	93.194563676	172.16.110.1	172.16.110.254	ICMP	98 Echo (ping) request id=0x42af, seq=8/2048, ttl=64 (reply in 69)
69	93.194759328	172.16.110.254	172.16.110.1	ICMP	98 Echo (ping) reply id=0x42af, seq=8/2048, ttl=64 (request in 68)
70	94.100890053	Routerboardc_1c:8d:2c	Nearest-Customer-Bridge	STP	60 RST, Root = 32768/0:c4:ad:34:1c:8d:2b Cost = 0 Port = 0x8002
71	94.218559067	172.16.110.1	172.16.110.254	ICMP	98 Echo (ping) request id=0x42af, seq=9/2304, ttl=64 (reply in 72)
72	94.218745920	172.16.110.254	172.16.110.1	ICMP	98 Echo (ping) reply id=0x42af, seq=9/2304, ttl=64 (request in 71)
73	95.242567626	172.16.110.1	172.16.110.254	ICMP	98 Echo (ping) request id=0x42af, seq=10/2560, ttl=64 (reply in 74)
74	95.242761199	172.16.110.254	172.16.110.1	ICMP	98 Echo (ping) reply id=0x42af, seq=10/2560, ttl=64 (request in 73)
75	96.103011611	Routerboardc_1c:8d:2c	Nearest-Customer-Bridge	STP	60 RST, Root = 32768/0:c4:ad:34:1c:8d:2b Cost = 0 Port = 0x8002
76	96.266559504	172.16.110.1	172.16.110.254	ICMP	98 Echo (ping) request id=0x42af, seq=11/2816, ttl=64 (reply in 77)
77	96.266747584	172.16.110.254	172.16.110.1	ICMP	98 Echo (ping) reply id=0x42af, seq=11/2816, ttl=64 (request in 76)
78	97.290558239	172.16.110.1	172.16.110.254	ICMP	98 Echo (ping) request id=0x42af, seq=12/3072, ttl=64 (reply in 79)
79	97.290747321	172.16.110.254	172.16.110.1	ICMP	98 Echo (ping) reply id=0x42af, seq=12/3072, ttl=64 (request in 78)
80	98.105157853	Routerboardc_1c:8d:2c	Nearest-Customer-Bridge	STP	60 RST, Root = 32768/0:c4:ad:34:1c:8d:2b Cost = 0 Port = 0x8002
81	98.314557112	172.16.110.1	172.16.110.254	ICMP	98 Echo (ping) request id=0x42af, seq=13/3328, ttl=64 (reply in 82)
82	98.314754585	172.16.110.254	172.16.110.1	ICMP	98 Echo (ping) reply id=0x42af, seq=13/3328, ttl=64 (request in 81)
83	99.338570637	172.16.110.1	172.16.110.254	ICMP	98 Echo (ping) request id=0x42af, seq=14/3584, ttl=64 (reply in 84)
84	99.338712457	172.16.110.254	172.16.110.1	ICMP	98 Echo (ping) reply id=0x42af, seq=14/3584, ttl=64 (request in 83)

Figure 2: Ping Tux4

7.3.2 Experience 2

1 0.0000000000	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60 RST. Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
2 1.278640295	172.16.100.1	172.16.100.254	ICMP	98 Echo (ping) request id=0x2d83, seq=1/256, ttl=64 (reply in 3)
3 1.278834277	172.16.100.254	172.16.100.1	ICMP	98 Echo (ping) reply id=0x2d83, seq=1/256, ttl=64 (request in 2)
4 2.002256742	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60 RST. Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
5 2.293833603	172.16.100.1	172.16.100.254	ICMP	98 Echo (ping) request id=0x2d83, seq=2/512, ttl=64 (reply in 6)
6 2.294042170	172.16.100.254	172.16.100.1	ICMP	98 Echo (ping) reply id=0x2d83, seq=2/512, ttl=64 (request in 5)
7 3.317855777	172.16.100.1	172.16.100.254	ICMP	98 Echo (ping) request id=0x2d83, seq=3/768, ttl=64 (reply in 8)
8 3.318882746	172.16.100.254	172.16.100.1	ICMP	98 Echo (ping) reply id=0x2d83, seq=3/768, ttl=64 (request in 7)
9 4.004458226	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60 RST. Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
10 4.341847685	172.16.100.1	172.16.100.254	ICMP	98 Echo (ping) request id=0x2d83, seq=4/1024, ttl=64 (reply in 11)
11 4.342065908	172.16.100.254	172.16.100.1	ICMP	98 Echo (ping) reply id=0x2d83, seq=4/1024, ttl=64 (request in 10)
12 5.365847236	172.16.100.1	172.16.100.254	ICMP	98 Echo (ping) request id=0x2d83, seq=5/1280, ttl=64 (reply in 13)
13 5.366072694	172.16.100.254	172.16.100.1	ICMP	98 Echo (ping) reply id=0x2d83, seq=5/1280, ttl=64 (request in 12)
14 6.006673189	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60 RST. Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
15 6.389853158	172.16.100.1	172.16.100.254	ICMP	98 Echo (ping) request id=0x2d83, seq=6/1536, ttl=64 (reply in 16)
16 6.390071235	172.16.100.254	172.16.100.1	ICMP	98 Echo (ping) reply id=0x2d83, seq=6/1536, ttl=64 (request in 15)
17 7.413858968	172.16.100.1	172.16.100.254	ICMP	98 Echo (ping) request id=0x2d83, seq=7/1792, ttl=64 (reply in 18)
18 7.414075279	172.16.100.254	172.16.100.1	ICMP	98 Echo (ping) reply id=0x2d83, seq=7/1792, ttl=64 (request in 17)
19 8.008884371	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60 RST. Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
20 8.437859909	172.16.100.1	172.16.100.254	ICMP	98 Echo (ping) request id=0x2d83, seq=8/2048, ttl=64 (reply in 21)
21 8.438086186	172.16.100.254	172.16.100.1	ICMP	98 Echo (ping) reply id=0x2d83, seq=8/2048, ttl=64 (request in 20)
22 9.461848481	172.16.100.1	172.16.100.254	ICMP	98 Echo (ping) request id=0x2d83, seq=9/2304, ttl=64 (reply in 23)
23 9.462074134	172.16.100.254	172.16.100.1	ICMP	98 Echo (ping) reply id=0x2d83, seq=9/2304, ttl=64 (request in 22)
24 10.011103751	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60 RST. Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
25 10.485878180	172.16.100.1	172.16.100.254	ICMP	98 Echo (ping) request id=0x2d83, seq=10/2560, ttl=64 (reply in 26)
26 10.486102037	172.16.100.254	172.16.100.1	ICMP	98 Echo (ping) reply id=0x2d83, seq=10/2560, ttl=64 (request in 25)
27 11.509841959	172.16.100.1	172.16.100.254	ICMP	98 Echo (ping) request id=0x2d83, seq=11/2816, ttl=64 (reply in 28)
28 11.510041071	172.16.100.254	172.16.100.1	ICMP	98 Echo (ping) reply id=0x2d83, seq=11/2816, ttl=64 (request in 27)

Figure 3: Ping Tux4

1 0.0000000000	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60 RST. Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
2 0.462539173	172.16.100.1	172.16.101.1	ICMP	98 Echo (ping) request id=0x2d66, seq=1/256, ttl=64 (no response found!)
3 1.487671274	172.16.100.1	172.16.101.1	ICMP	98 Echo (ping) request id=0x2d66, seq=2/512, ttl=64 (no response found!)
4 2.002216305	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60 RST. Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
5 2.511718725	172.16.100.1	172.16.101.1	ICMP	98 Echo (ping) request id=0x2d66, seq=3/768, ttl=64 (no response found!)
6 3.528268188	172.16.100.254	172.16.100.1	ICMP	126 Destination unreachable (Host unreachable)
7 3.520268714	172.16.100.254	172.16.100.1	ICMP	126 Destination unreachable (Host unreachable)
8 3.520268797	172.16.100.254	172.16.100.1	ICMP	126 Destination unreachable (Host unreachable)
9 3.520453549	172.16.100.1	172.16.101.1	ICMP	98 Echo (ping) request id=0x2d66, seq=4/1024, ttl=64 (no response found!)
10 4.004393995	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60 RST. Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
11 4.527716975	172.16.100.1	172.16.101.1	ICMP	98 Echo (ping) request id=0x2d66, seq=5/1280, ttl=64 (no response found!)
12 5.551664408	TPLink_c2:3c:f5	TPLink_c2:3c:f5	ARP	42 Who has 172.16.100.254? Tell 172.16.100.1
13 5.551773726	172.16.100.1	172.16.101.1	ICMP	98 Echo (ping) request id=0x2d66, seq=6/1536, ttl=64 (no response found!)
14 5.551839733	TPLink_c2:51:4d	TPLink_c2:3c:f5	ARP	60 172.16.100.254 is at ec:75:0c:c2:51:4d
15 6.006647957	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60 RST. Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
16 6.575708789	172.16.100.1	172.16.101.1	ICMP	98 Echo (ping) request id=0x2d66, seq=7/1792, ttl=64 (no response found!)
17 6.592194560	172.16.100.254	172.16.100.1	ICMP	126 Destination unreachable (Host unreachable)
18 6.592195061	172.16.100.254	172.16.100.1	ICMP	126 Destination unreachable (Host unreachable)
19 6.592195144	172.16.100.254	172.16.100.1	ICMP	126 Destination unreachable (Host unreachable)
20 6.592195214	172.16.100.254	172.16.100.1	ICMP	126 Destination unreachable (Host unreachable)
21 7.577402641	172.16.100.1	172.16.101.1	ICMP	98 Echo (ping) request id=0x2d66, seq=8/2048, ttl=64 (no response found!)
22 8.008856241	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60 RST. Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
23 8.576177929	TPLink_c2:51:4d	TPLink_c2:3c:f5	ARP	60 Who has 172.16.100.1? Tell 172.16.100.254
24 8.576193687	TPLink_c2:3c:f5	TPLink_c2:51:4d	ARP	42 172.16.100.1 is at ec:75:0c:c2:3c:f5
25 8.591687171	172.16.100.1	172.16.101.1	ICMP	98 Echo (ping) request id=0x2d66, seq=9/2304, ttl=64 (no response found!)
26 9.615709288	172.16.100.1	172.16.101.1	ICMP	98 Echo (ping) request id=0x2d66, seq=10/2560, ttl=64 (no response found!)
27 10.011059275	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60 RST. Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
28 10.656235182	172.16.100.254	172.16.100.1	ICMP	126 Destination unreachable (Host unreachable)
29 10.656235718	172.16.100.254	172.16.100.1	ICMP	126 Destination unreachable (Host unreachable)
30 10.656235798	172.16.100.254	172.16.100.1	ICMP	126 Destination unreachable (Host unreachable)
31 12.013267531	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60 RST. Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002

Figure 4: Ping Tux2

1	0.0000000000	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001
2	2.002224769	Routerboards_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001
3	3.872754959	172.16.120.1	172.16.120.255	ICMP	98	Echo (ping) request id=0x4a76, seq=1/256, ttl=64 (no response found!)
4	4.872912637	172.16.120.254	172.16.120.1	ICMP	98	Echo (ping) reply id=0x4a76, seq=1/256, ttl=64
5	4.004319735	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001
6	4.879824191	172.16.120.1	172.16.120.255	ICMP	98	Echo (ping) request id=0x4a76, seq=2/512, ttl=64 (no response found!)
7	4.879935316	172.16.120.254	172.16.120.1	ICMP	98	Echo (ping) reply id=0x4a76, seq=2/512, ttl=64
8	5.595265092	0.0.0.0	255.255.255.255	MNDP	160	5678 → 5678 Len=118
9	5.595265541	Routerboardc_1c:a3:2c	CDP/FTP/DTP/PagP/UDLD	CDP	94	Device ID: MikroTik Port ID: bridge120
10	5.595267521	Routerboardc_1c:a3:2c	Nearest-Bridge	LLDP	111	MA/c4:ad:34:1c:a3:2c IN/bridge120 120 Sys=MikroTik SysD=MikroTik RouterOS 6.
11	5.90838847	172.16.120.1	172.16.120.255	ICMP	98	Echo (ping) request id=0x4a76, seq=3/768, ttl=64 (no response found!)
12	5.908975356	172.16.120.254	172.16.120.1	ICMP	98	Echo (ping) reply id=0x4a76, seq=3/768, ttl=64
13	6.005677516	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001
14	6.927837759	172.16.120.1	172.16.120.255	ICMP	98	Echo (ping) request id=0x4a76, seq=4/1024, ttl=64 (no response found!)
15	6.927966510	172.16.120.254	172.16.120.1	ICMP	98	Echo (ping) reply id=0x4a76, seq=4/1024, ttl=64
16	7.951831921	172.16.120.1	172.16.120.255	ICMP	98	Echo (ping) request id=0x4a76, seq=5/1280, ttl=64 (no response found!)
17	7.951960367	172.16.120.254	172.16.120.1	ICMP	98	Echo (ping) reply id=0x4a76, seq=5/1280, ttl=64
18	8.007790925	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001
19	8.975934672	172.16.120.1	172.16.120.255	ICMP	98	Echo (ping) request id=0x4a76, seq=6/1536, ttl=64 (no response found!)
20	8.975961609	172.16.120.254	172.16.120.1	ICMP	98	Echo (ping) reply id=0x4a76, seq=6/1536, ttl=64
21	9.999833021	172.16.120.1	172.16.120.255	ICMP	98	Echo (ping) request id=0x4a76, seq=7/1792, ttl=64 (no response found!)
22	9.999957792	172.16.120.254	172.16.120.1	ICMP	98	Echo (ping) reply id=0x4a76, seq=7/1792, ttl=64
23	10.009894568	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001
24	11.023828132	172.16.120.1	172.16.120.255	ICMP	98	Echo (ping) request id=0x4a76, seq=8/2048, ttl=64 (no response found!)
25	11.023952695	172.16.120.254	172.16.120.1	ICMP	98	Echo (ping) reply id=0x4a76, seq=8/2048, ttl=64
26	12.012039985	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001
27	12.047828047	172.16.120.1	172.16.120.255	ICMP	98	Echo (ping) request id=0x4a76, seq=9/2304, ttl=64 (no response found!)
28	12.047947576	172.16.120.254	172.16.120.1	ICMP	98	Echo (ping) reply id=0x4a76, seq=9/2304, ttl=64
29	13.071835053	172.16.120.1	172.16.120.255	ICMP	98	Echo (ping) request id=0x4a76, seq=10/2560, ttl=64 (no response found!)
30	13.071957867	172.16.120.254	172.16.120.1	ICMP	98	Echo (ping) reply id=0x4a76, seq=10/2560, ttl=64
31	14.014161973	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001
32	14.095812975	172.16.120.1	172.16.120.255	ICMP	98	Echo (ping) request id=0x4a76, seq=11/2816, ttl=64 (no response found!)
33	14.095940475	172.16.120.254	172.16.120.1	ICMP	98	Echo (ping) reply id=0x4a76, seq=11/2816, ttl=64
34	15.119839273	172.16.120.1	172.16.120.255	ICMP	98	Echo (ping) request id=0x4a76, seq=12/3072, ttl=64 (no response found!)
35	15.119958303	172.16.120.254	172.16.120.1	ICMP	98	Echo (ping) reply id=0x4a76, seq=12/3072, ttl=64
36	16.016285568	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001
37	16.143851458	172.16.120.1	172.16.120.255	ICMP	98	Echo (ping) request id=0x4a76, seq=13/3328, ttl=64 (no response found!)
38	16.143974998	172.16.120.254	172.16.120.1	ICMP	98	Echo (ping) reply id=0x4a76, seq=13/3328, ttl=64
39	18.018385569	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001

Figure 5: Ping Broadcast on Tux3

1	0.0000000000	Routerboardc_1c:a3:2b	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:12 Cost = 10 Port = 0x8001
2	2.175911549	192.168.88.1	255.255.255.255	MNDP	153	5678 → 5678 Len=111
3	3.1759222102	Routerboardc_eb:24:12	CDP/FTP/DTP/PagP/UDLD	CDP	188	Device ID: MikroTik Port ID: bridge
4	4.2002183409	Routerboardc_1c:a3:2b	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:12 Cost = 10 Port = 0x8001
5	4.004200682	Routerboardc_1c:a3:2b	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:12 Cost = 10 Port = 0x8001
6	4.166376317	172.16.121.1	172.16.121.255	ICMP	98	Echo (ping) request id=0x3078, seq=1/256, ttl=64 (no response found!)
7	4.166733164	172.16.121.253	172.16.121.1	ICMP	98	Echo (ping) reply id=0x3078, seq=1/256, ttl=64
8	5.192129112	172.16.121.1	172.16.121.255	ICMP	98	Echo (ping) request id=0x3078, seq=2/512, ttl=64 (no response found!)
9	5.192375956	172.16.121.253	172.16.121.1	ICMP	98	Echo (ping) reply id=0x3078, seq=2/512, ttl=64
10	6.006353810	Routerboardc_1c:a3:2b	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:12 Cost = 10 Port = 0x8001
11	6.216140116	172.16.121.1	172.16.121.255	ICMP	98	Echo (ping) request id=0x3078, seq=3/768, ttl=64 (no response found!)
12	6.216358383	172.16.121.253	172.16.121.1	ICMP	98	Echo (ping) reply id=0x3078, seq=3/768, ttl=64
13	7.240138045	172.16.121.1	172.16.121.255	ICMP	98	Echo (ping) request id=0x3078, seq=4/1024, ttl=64 (no response found!)
14	7.240306323	172.16.121.253	172.16.121.1	ICMP	98	Echo (ping) reply id=0x3078, seq=4/1024, ttl=64
15	8.008448782	Routerboardc_1c:a3:2b	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:12 Cost = 10 Port = 0x8001
16	8.264140337	172.16.121.1	172.16.121.255	ICMP	98	Echo (ping) request id=0x3078, seq=5/1280, ttl=64 (no response found!)
17	8.264319966	172.16.121.253	172.16.121.1	ICMP	98	Echo (ping) reply id=0x3078, seq=5/1280, ttl=64
18	9.288172521	172.16.121.1	172.16.121.255	ICMP	98	Echo (ping) request id=0x3078, seq=6/1536, ttl=64 (no response found!)
19	9.288356607	172.16.121.253	172.16.121.1	ICMP	98	Echo (ping) reply id=0x3078, seq=6/1536, ttl=64
20	9.324020080	TPLink_c2:10:6b	Tell_2:31:73	ARP	60	Who has 172.16.121.1? Tell 172.16.121.253
21	9.324033331	TPLink_c2:31:73	TPLink_c2:10:6b	ARP	42	172.16.121.1 is at ec:75:ec:c2:31:73
22	10.018581982	Routerboardc_1c:a3:2b	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:12 Cost = 10 Port = 0x8001
23	10.312139634	172.16.121.1	172.16.121.255	ICMP	98	Echo (ping) request id=0x3078, seq=7/1792, ttl=64 (no response found!)
24	10.31232357	172.16.121.253	172.16.121.1	ICMP	98	Echo (ping) reply id=0x3078, seq=7/1792, ttl=64
25	11.336134830	172.16.121.1	172.16.121.255	ICMP	98	Echo (ping) request id=0x3078, seq=8/2048, ttl=64 (no response found!)
26	11.336330262	172.16.121.253	172.16.121.1	ICMP	98	Echo (ping) reply id=0x3078, seq=8/2048, ttl=64
27	12.012694405	Routerboardc_1c:a3:2b	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:12 Cost = 10 Port = 0x8001
28	12.360133589	172.16.121.1	172.16.121.255	ICMP	98	Echo (ping) request id=0x3078, seq=9/2304, ttl=64 (no response found!)
29	12.360298238	172.16.121.253	172.16.121.1	ICMP	98	Echo (ping) reply id=0x3078, seq=9/2304, ttl=64
30	13.384132232	172.16.121.1	172.16.121.255	ICMP	98	Echo (ping) request id=0x3078, seq=10/2560, ttl=64 (no response found!)
31	13.384300877	172.16.121.253	172.16.121.1	ICMP	98	Echo (ping) reply id=0x3078, seq=10/2560, ttl=64
32	14.014769453	Routerboardc_1c:a3:2b	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:12 Cost = 10 Port = 0x8001
33	14.408131749	172.16.121.1	172.16.121.255	ICMP	98	Echo (ping) request id=0x3078, seq=11/2816, ttl=64 (no response found!)
34	14.408269213	172.16.121.253	172.16.121.1	ICMP	98	Echo (ping) reply id=0x3078, seq=11/2816, ttl=64
35	15.432132008	172.16.121.1	172.16.121.255	ICMP	98	Echo (ping) request id=0x3078, seq=12/3072, ttl=64 (no response found!)
36	15.432294687	172.16.121.253	172.16.121.1	ICMP	98	Echo (ping) reply id=0x3078, seq=12/3072, ttl=64
37	16.016889401	Routerboardc_1c:a3:2b	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:12 Cost = 10 Port = 0x8001

Figure 6: Ping Broadcast on Tux2

Kernel IP routing table						
Destination	Gateway	Genmask	Flags	Metric	Ref	Use Iface
0.0.0.0	10.227.20.254	0.0.0.0	UG	100	0	0 if_mng
10.227.20.0	0.0.0.0	255.255.255.0	U	100	0	0 if_mng
172.16.100.0	172.16.101.253	255.255.255.0	UG	0	0	0 if_e1
172.16.101.0	0.0.0.0	255.255.255.0	U	0	0	0 if_e1

Figure 7: Routes Tux2

Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	10.227.20.254	0.0.0.0	UG	100	0	0	if_mng
10.227.20.0	0.0.0.0	255.255.255.0	U	100	0	0	if_mng
172.16.1.0	172.16.100.254	255.255.255.0	UG	0	0	0	if_e1
172.16.100.0	0.0.0.0	255.255.255.0	U	0	0	0	if_e1
172.16.101.0	172.16.100.254	255.255.255.0	UG	0	0	0	if_e1

Figure 8: Routes Tux3

Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	10.227.20.254	0.0.0.0	UG	100	0	0	if_mng
10.227.20.0	0.0.0.0	255.255.255.0	U	100	0	0	if_mng
172.16.100.0	0.0.0.0	255.255.255.0	U	0	0	0	if_e1
172.16.101.0	0.0.0.0	255.255.255.0	U	0	0	0	if_e2

Figure 9: Routes Tux4

1	0.00000000	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
2	1.125751666	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x332f, seq=1/256, ttl=64 (reply in 3)
3	1.126222697	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x332f, seq=1/256, ttl=63 (request in 2)
4	2.002237751	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
5	2.138759584	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x332f, seq=2/512, ttl=64 (reply in 6)
6	2.139188197	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x332f, seq=2/512, ttl=63 (request in 5)
7	3.162753218	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x332f, seq=3/768, ttl=64 (reply in 8)
8	3.163189728	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x332f, seq=3/768, ttl=63 (request in 7)
9	4.004485098	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
10	4.186749855	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x332f, seq=4/1024, ttl=64 (reply in 11)
11	4.187179973	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x332f, seq=4/1024, ttl=63 (request in 10)
12	5.210750651	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x332f, seq=5/1280, ttl=64 (reply in 13)
13	5.211190245	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x332f, seq=5/1280, ttl=63 (request in 12)
14	6.006721053	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
15	6.234751151	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x332f, seq=6/1536, ttl=64 (reply in 16)
16	6.235180495	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x332f, seq=6/1536, ttl=63 (request in 15)
17	7.258750958	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x332f, seq=7/1792, ttl=64 (reply in 18)
18	7.259182108	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x332f, seq=7/1792, ttl=63 (request in 17)
19	8.008959412	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
20	8.282751223	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x332f, seq=8/2048, ttl=64 (reply in 21)
21	8.283185616	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x332f, seq=8/2048, ttl=63 (request in 20)
22	8.418708546	TPLink_c2:3c:f5	TPLink_c2:51:4d	ARP	42	Who has 172.16.100.254? Tell 172.16.100.1
23	8.41926414	TPLink_c2:51:4d	TPLink_c2:3c:f5	ARP	60	172.16.100.254 is at ec:75:0c:c2:51:4d
24	9.306744683	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x332f, seq=9/2304, ttl=64 (reply in 25)
25	9.307184513	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x332f, seq=9/2304, ttl=63 (request in 24)
26	10.011188684	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
27	10.330763665	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x332f, seq=10/2560, ttl=64 (reply in 28)
28	10.331190450	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x332f, seq=10/2560, ttl=63 (request in 27)
29	11.354750977	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x332f, seq=11/2816, ttl=64 (reply in 30)
30	11.355185315	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x332f, seq=11/2816, ttl=63 (request in 29)
31	12.013428977	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
32	14.015672450	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002

Figure 10: Ping on Tux2

1	0.00000000	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
2	2.0917542288	172.16.100.1	172.16.100.254	ICMP	98	Echo (ping) request id=0x331c, seq=1/256, ttl=64 (reply in 3)
3	2.091759746	172.16.100.254	172.16.100.1	ICMP	98	Echo (ping) reply id=0x331c, seq=1/256, ttl=64 (request in 2)
4	4.1.933062947	172.16.100.1	172.16.100.254	ICMP	98	Echo (ping) request id=0x331c, seq=2/512, ttl=64 (reply in 5)
5	5.1.933285735	172.16.100.254	172.16.100.1	ICMP	98	Echo (ping) reply id=0x331c, seq=2/512, ttl=64 (request in 4)
6	6.0022778688	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
7	7.2.957090162	172.16.100.1	172.16.100.254	ICMP	98	Echo (ping) request id=0x331c, seq=3/768, ttl=64 (reply in 8)
8	8.2.957312320	172.16.100.254	172.16.100.1	ICMP	98	Echo (ping) reply id=0x331c, seq=3/768, ttl=64 (request in 7)
9	9.3.98109751	172.16.100.1	172.16.100.254	ICMP	98	Echo (ping) request id=0x331c, seq=4/1024, ttl=64 (reply in 10)
10	10.3.981319002	172.16.100.254	172.16.100.1	ICMP	98	Echo (ping) reply id=0x331c, seq=4/1024, ttl=64 (request in 9)
11	11.4.004473097	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
12	12.5.005056054	172.16.100.1	172.16.100.254	ICMP	98	Echo (ping) request id=0x331c, seq=5/1280, ttl=64 (reply in 13)
13	13.5.005280231	172.16.100.254	172.16.100.1	ICMP	98	Echo (ping) reply id=0x331c, seq=5/1280, ttl=64 (request in 12)
14	14.5.932997023	TPLink_c2:3c:f5	TPLink_c2:51:4d	ARP	42	Who has 172.16.100.254? Tell 172.16.100.1
15	15.5.933209338	TPLink_c2:51:4d	TPLink_c2:3c:f5	ARP	60	172.16.100.254 is at ec:75:0c:c2:51:4d
16	16.6.006732790	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
17	17.6.029051614	172.16.100.254	172.16.100.254	ICMP	98	Echo (ping) request id=0x331c, seq=6/1536, ttl=64 (reply in 18)
18	18.6.029269604	172.16.100.254	172.16.100.1	ICMP	98	Echo (ping) reply id=0x331c, seq=6/1536, ttl=64 (request in 17)
19	19.6.141379503	TPLink_c2:51:4d	TPLink_c2:3c:f5	ARP	42	172.16.100.1 is at ec:75:0c:c2:3c:f5
20	20.6.141400351	TPLink_c2:3c:f5	TPLink_c2:51:4d	ARP	60	172.16.100.17 tell 172.16.100.254
21	21.7.053053306	172.16.100.1	172.16.100.254	ICMP	98	Echo (ping) request id=0x331c, seq=7/1792, ttl=64 (reply in 22)
22	22.7.053277193	172.16.100.254	172.16.100.1	ICMP	98	Echo (ping) reply id=0x331c, seq=7/1792, ttl=64 (request in 21)
23	23.8.008976418	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002

Figure 11: Ping on Tux4's e1

1	0.000000000	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
2	0.983775513	172.16.100.1	172.16.101.253	ICMP	98	Echo (ping) request id=0x3328, seq=1/256, ttl=64 (reply in 3)
3	0.983995623	172.16.101.253	172.16.100.1	ICMP	98	Echo (ping) reply id=0x3328, seq=1/256, ttl=64 (request in 2)
4	2.002228372	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
5	2.013267438	172.16.100.1	172.16.101.253	ICMP	98	Echo (ping) request id=0x3328, seq=2/512, ttl=64 (reply in 6)
6	2.013483223	172.16.101.253	172.16.100.1	ICMP	98	Echo (ping) reply id=0x3328, seq=2/512, ttl=64 (request in 5)
7	3.037280033	172.16.100.1	172.16.101.253	ICMP	98	Echo (ping) request id=0x3328, seq=3/768, ttl=64 (reply in 8)
8	3.037502581	172.16.101.253	172.16.100.1	ICMP	98	Echo (ping) reply id=0x3328, seq=3/768, ttl=64 (request in 7)
9	4.004428356	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
10	4.061270043	172.16.100.1	172.16.101.253	ICMP	98	Echo (ping) request id=0x3328, seq=4/1024, ttl=64 (reply in 11)
11	4.061490031	172.16.101.253	172.16.100.1	ICMP	98	Echo (ping) reply id=0x3328, seq=4/1024, ttl=64 (request in 10)
12	5.085265373	172.16.100.1	172.16.101.253	ICMP	98	Echo (ping) request id=0x3328, seq=5/1280, ttl=64 (reply in 13)
13	5.085488262	172.16.101.253	172.16.100.1	ICMP	98	Echo (ping) reply id=0x3328, seq=5/1280, ttl=64 (request in 12)
14	6.086722692	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
15	6.093521530	TPLink_c2:51:4d	TPLink_c2:3:c:f5	ARP	60	Who has 172.16.100.1 Tell 172.16.100.254
16	6.093541661	TPLink_c2:3:c:f5	TPLink_c2:51:4d	ARP	42	172.16.100.1 at ec:75:0:c2:3:c:f5
17	6.109270114	172.16.100.1	172.16.101.253	ICMP	98	Echo (ping) request id=0x3328, seq=6/1536, ttl=64 (reply in 18)
18	6.109484614	172.16.101.253	172.16.100.1	ICMP	98	Echo (ping) reply id=0x3328, seq=6/1536, ttl=64 (request in 17)
19	6.141201848	TPLink_c2:3:c:f5	TPLink_c2:51:4d	ARP	42	Who has 172.16.100.254 Tell 172.16.100.1
20	6.141382228	TPLink_c2:51:4d	TPLink_c2:3:c:f5	ARP	60	172.16.100.254 is at ec:75:0:c2:51:4d
21	7.133265351	172.16.100.1	172.16.101.253	ICMP	98	Echo (ping) request id=0x3328, seq=7/1792, ttl=64 (reply in 22)
22	7.133487990	172.16.101.253	172.16.100.1	ICMP	98	Echo (ping) reply id=0x3328, seq=7/1792, ttl=64 (request in 21)
23	8.008964368	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
24	8.157288903	172.16.100.1	172.16.101.253	ICMP	98	Echo (ping) request id=0x3328, seq=8/2048, ttl=64 (reply in 25)
25	8.157504030	172.16.101.253	172.16.100.1	ICMP	98	Echo (ping) reply id=0x3328, seq=8/2048, ttl=64 (request in 24)
26	9.181261242	172.16.100.1	172.16.101.253	ICMP	98	Echo (ping) request id=0x3328, seq=9/2304, ttl=64 (reply in 27)
27	9.181486521	172.16.101.253	172.16.100.1	ICMP	98	Echo (ping) reply id=0x3328, seq=9/2304, ttl=64 (request in 26)
28	10.011195467	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002
29	12.013427293	Routerboardc_2b:fa:0c	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8002

Figure 12: Ping on Tux4's e2

1	0.000000000	Routerboardc_2b:fa:0d	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8001
2	1.396997210	172.16.100.1	239.255.255.250	UDP/XML	649	39327 - 3702 Len=607
3	1.414085335	172.16.100.1	224.0.0.22	IGMPv3	60	Membership Report / Leave group 239.255.255.250
4	1.558641725	172.16.100.254	239.255.255.250	UDP/XML	649	53201 - 3702 Len=607
5	1.562026748	172.16.100.254	224.0.0.22	IGMPv3	54	Membership Report / Leave group 239.255.255.250
6	1.891174455	172.16.100.1	224.0.0.22	IGMPv3	60	Membership Report / Leave group 239.255.255.250
7	2.004740596	Routerboardc_2b:fa:0d	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8001
8	2.139718782	172.16.100.1	239.255.255.250	UDP/XML	649	48494 - 3702 Len=607
9	2.154073267	172.16.100.1	224.0.0.22	IGMPv3	60	Membership Report / Leave group 239.255.255.250
10	2.482011393	172.16.100.254	224.0.0.22	IGMPv3	54	Membership Report / Leave group 239.255.255.250
11	2.859101578	172.16.100.1	224.0.0.22	IGMPv3	60	Membership Report / Leave group 239.255.255.250
12	3.777449468	172.16.100.1	239.255.255.250	UDP/XML	649	59472 - 3702 Len=607
13	3.790090981	172.16.100.1	224.0.0.22	IGMPv3	60	Membership Report / Leave group 239.255.255.250
14	4.004740596	Routerboardc_2b:fa:0d	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8001
15	4.158847114	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x3324, seq=1/256, ttl=64 (reply in 18)
16	4.159337328	TPLink_c2:51:4d	Broadcast	ARP	42	Who has 172.16.100.1 Tell 172.16.100.254
17	4.159544634	TPLink_c2:3:c:f5	TPLink_c2:51:4d	ARP	60	172.16.100.1 at ec:75:0:c2:3:c:f5
18	4.159556036	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x3324, seq=1/256, ttl=63 (request in 15)
19	4.770888064	172.16.100.1	224.0.0.22	IGMPv3	60	Membership Report / Leave group 239.255.255.250
20	5.159934928	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x3324, seq=2/512, ttl=64 (reply in 21)
21	5.160177335	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) reply id=0x3324, seq=2/512, ttl=63 (request in 20)
22	5.171616949	Routerboardc_2b:fa:0d	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8001
23	6.178142339	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x3324, seq=3/768, ttl=64 (reply in 24)
24	6.178366983	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x3324, seq=3/768, ttl=63 (request in 23)
25	7.202134028	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x3324, seq=4/1024, ttl=64 (reply in 26)
26	7.202373944	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x3324, seq=4/1024, ttl=63 (request in 25)
27	7.483211052	0.0.0.0	255.255.255.255	NDNP	160	5678 - 5678 Len=118
28	7.843249487	Routerboardc_2b:fa:0d	CDP/FTP/DTP/PgP/UDLD	CDP	94	Device ID: MikroTik Port ID: bridge10
29	7.843276075	Routerboardc_2b:fa:0d	Nearest-Bridge	LLDP	111	MAC:44:ad:34:2b:fa:0d IN:bridge10 120 Sys=MikroTik Sys=MikroTik RouterOS 6.4.13 (long-term) CRS326-24G-2S+
30	8.009439515	Routerboardc_2b:fa:0d	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8001
31	8.013183574	172.16.100.254	239.255.255.250	UDP/XML	649	43134 - 3702 Len=607
32	8.026086395	172.16.100.254	224.0.0.22	IGMPv3	54	Membership Report / Leave group 239.255.255.250
33	8.204980972	172.16.100.1	239.255.255.250	UDP/XML	649	57463 - 3702 Len=607
34	8.222052218	172.16.100.1	224.0.0.22	IGMPv3	60	Membership Report / Leave group 239.255.255.250
35	8.403618037	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x3324, seq=4/1024, ttl=64 (reply in 36)
36	8.422053936	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x3324, seq=4/1024, ttl=63 (request in 35)
37	8.305998521	172.16.100.254	224.0.0.22	IGMPv3	54	Membership Report / Leave group 239.255.255.250
38	8.420880364	172.16.100.1	224.0.0.22	IGMPv3	60	Membership Report / Leave group 239.255.255.250
39	8.763423795	172.16.100.254	239.255.255.250	UDP/XML	649	58900 - 3702 Len=607
40	8.778837174	172.16.100.254	224.0.0.22	IGMPv3	54	Membership Report / Leave group 239.255.255.250
41	9.250899633	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x3324, seq=6/1536, ttl=64 (reply in 42)
42	9.259352479	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x3324, seq=6/1536, ttl=63 (request in 41)
43	9.314842797	TPLink_c2:3:c:f5	TPLink_c2:51:4d	ARP	60	Who has 172.16.100.254 Tell 172.16.100.3
44	9.314864867	TPLink_c2:51:4d	TPLink_c2:3:c:f5	ARP	42	172.16.100.254 is at ec:75:0:c2:51:4d
45	9.618081029	172.16.100.254	224.0.0.22	IGMPv3	54	Membership Report / Leave group 239.255.255.250
46	10.001728020	Routerboardc_2b:fa:0d	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8001
47	10.274999804	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x3324, seq=7/1792, ttl=64 (reply in 48)
48	10.27437597	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x3324, seq=7/1792, ttl=63 (request in 47)
49	10.376249573	172.16.100.254	239.255.255.250	UDP/XML	649	35088 - 3702 Len=607
50	10.394051598	172.16.100.254	224.0.0.22	IGMPv3	54	Membership Report / Leave group 239.255.255.250
51	10.706830881	172.16.100.254	224.0.0.22	IGMPv3	54	Membership Report / Leave group 239.255.255.250
52	11.298139759	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x3324, seq=8/2048, ttl=64 (reply in 53)
53	11.298481738	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x3324, seq=8/2048, ttl=63 (request in 52)
54	12.003960194	Routerboardc_2b:fa:0d	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0/c4:ad:34:2b:fa:0d Cost = 0 Port = 0x8001

Figure 13: Captures from Tux4's e1

1	0.00000000	Routerboardc_2b:fa:0e	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0:c4:ad:34:2b:fa:0b Cost = 0 Port = 0x8002
2	2.001891377	Routerboardc_2b:fa:0e	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0:c4:ad:34:2b:fa:0b Cost = 0 Port = 0x8002
3	3.994155738	Routerboardc_2b:fa:0e	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0:c4:ad:34:2b:fa:0b Cost = 0 Port = 0x8002
4	4.553771924	172.16.101.253	239.255.255.250	UDP/XML	649	56898 - 3702 Len=607
5	4.565654151	172.16.101.253	224.0.0.22	IGMPv3	54	Membership Report / Leave group 239.255.255.250
6	5.197688058	172.16.101.253	224.0.0.22	IGMPv3	54	Membership Report / Leave group 239.255.255.250
7	7.162144731	Routerboardc_2b:fa:0e	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0:c4:ad:34:2b:fa:0b Cost = 0 Port = 0x8002
8	7.161925145	TPLink_c2:17:2a	Broadcast	ARP	42	Who has 172.16.101.17 Tell 172.16.101.253
9	7.162144732	TPLink_c2:17:8b	TPLink_c2:17:2a	ARP	60	172.16.101.1 is at ec:75:0:c2:17:8b
10	7.162157549	172.16.101.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x33c4, seq=1/256, ttl=63 (reply in 11)
11	7.162358811	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x33c4, seq=1/256, ttl=64 (request in 18)
12	7.160888617	Routerboardc_2b:fa:0e	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0:c4:ad:34:2b:fa:0b Cost = 0 Port = 0x8002
13	8.162999494	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x33c4, seq=2/512, ttl=63 (reply in 14)
14	8.163288124	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x33c4, seq=2/512, ttl=64 (request in 13)
15	9.181205751	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x33c4, seq=3/768, ttl=63 (reply in 16)
16	9.181417272	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) reply id=0x33c4, seq=3/768, ttl=64 (request in 15)
17	10.001255866	Routerboardc_2b:fa:0e	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0:c4:ad:34:2b:fa:0b Cost = 0 Port = 0x8002
18	10.205196497	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x33c4, seq=4/1024, ttl=63 (reply in 19)
19	10.205484814	172.16.100.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x33c4, seq=4/1024, ttl=64 (request in 18)
20	10.846444808	0.0.0.0	255.255.255.255	NDNP	168	5678 - 5678 Len=118
21	10.846474365	Routerboardc_2b:fa:0b	COP/VT/OTP/AgP/UDLD	COP	94	Device ID: MikroTik Port ID: bridge1@1
22	10.846532724	Routerboardc_2b:fa:0b	Nearest-Bridge	LDPD	111	MA:4:ad:34:2b:fa:0b IN/bridge1@1 120 Sys=MikroTik SysD=MikroTik RouterOS 6.43.16 (long-term) CR5326-24G-2S+
23	11.016282624	172.16.101.253	239.255.255.250	UDP/XML	649	56937 - 3702 Len=607
24	11.023051411	172.16.101.253	224.0.0.22	IGMPv3	54	Membership Report / Leave group 239.255.255.250
25	11.223326594	172.16.101.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x33c4, seq=5/1280, ttl=63 (reply in 26)
26	11.223326284	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x33c4, seq=5/1280, ttl=64 (request in 25)
27	11.765519465	172.16.101.253	239.255.255.250	UDP/XML	649	60230 - 3702 Len=607
28	11.781061021	172.16.101.253	224.0.0.22	IGMPv3	54	Membership Report / Leave group 239.255.255.250
29	11.002532728	Routerboardc_2b:fa:0e	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0:c4:ad:34:2b:fa:0b Cost = 0 Port = 0x8002
30	12.253167869	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x33c4, seq=6/1536, ttl=63 (reply in 31)
31	12.253381146	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x33c4, seq=6/1536, ttl=64 (request in 30)
32	12.299119439	TPLink_c2:17:8b	TPLink_c2:17:2a	ARP	60	Who has 172.16.101.253 Tell 172.16.101.1
33	12.299166368	TPLink_c2:17:2a	TPLink_c2:17:8b	ARP	42	172.16.101.253 is at ec:75:0:c2:17:2a
34	12.493861848	172.16.101.253	224.0.0.22	IGMPv3	54	Membership Report / Leave group 239.255.255.250
35	13.277161413	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x33c4, seq=7/1792, ttl=63 (reply in 36)
36	13.277368387	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x33c4, seq=7/1792, ttl=64 (request in 35)
37	13.379352349	172.16.101.253	239.255.255.250	UDP/XML	649	35351 - 3702 Len=607
38	13.397086386	172.16.101.253	224.0.0.22	IGMPv3	54	Membership Report / Leave group 239.255.255.250
39	14.005876248	Routerboardc_2b:fa:0e	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0:c4:ad:34:2b:fa:0b Cost = 0 Port = 0x8002
40	14.061063355	172.16.101.253	224.0.0.22	IGMPv3	54	Membership Report / Leave group 239.255.255.250
41	14.301201877	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x33c4, seq=8/2048, ttl=63 (reply in 42)
42	14.301430781	172.16.100.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x33c4, seq=8/2048, ttl=64 (request in 41)
43	15.325268648	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x33c4, seq=9/2304, ttl=63 (reply in 44)
44	15.325407457	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) reply id=0x33c4, seq=9/2304, ttl=64 (request in 43)
45	16.018302888	Routerboardc_2b:fa:0e	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0:c4:ad:34:2b:fa:0b Cost = 0 Port = 0x8002
46	16.349167854	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x33c4, seq=10/2560, ttl=63 (reply in 47)
47	16.349377324	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x33c4, seq=10/2560, ttl=64 (request in 46)
48	17.373169683	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x33c4, seq=11/2816, ttl=63 (reply in 49)
49	17.373338667	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x33c4, seq=11/2816, ttl=64 (request in 48)
50	18.018302888	Routerboardc_2b:fa:0e	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0:c4:ad:34:2b:fa:0b Cost = 0 Port = 0x8002
51	18.397211958	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x33c4, seq=12/3072, ttl=63 (reply in 52)
52	18.397421193	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x33c4, seq=12/3072, ttl=64 (request in 51)
53	19.421173793	172.16.100.1	172.16.101.1	ICMP	98	Echo (ping) request id=0x33c4, seq=13/3328, ttl=63 (reply in 54)
54	19.421382401	172.16.101.1	172.16.100.1	ICMP	98	Echo (ping) reply id=0x33c4, seq=13/3328, ttl=64 (request in 53)
55	20.012515926	Routerboardc_2b:fa:0e	Nearest-Customer-Bridge	STP	60	RST, Root = 32768/0:c4:ad:34:2b:fa:0b Cost = 0 Port = 0x8002

Figure 14: Captures from Tux4's e2

7.3.4 Experience 4

Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	10.227.20.254	0.0.0.0	UG	100	0	0	if_mng
10.227.20.0	0.0.0.0	255.255.255.0	U	100	0	0	if_mng
17.16.1.0	172.16.111.254	255.255.255.0	UG	0	0	0	if_e1
172.16.110.0	172.16.111.253	255.255.255.0	UG	0	0	0	if_e1
172.16.111.0	0.0.0.0	255.255.255.0	U	0	0	0	if_e1

Figure 15: Routes on Tux2

Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	10.227.20.254	0.0.0.0	UG	100	0	0	if_mng
10.227.20.0	0.0.0.0	255.255.255.0	U	100	0	0	if_mng
172.16.1.0	172.16.110.254	255.255.255.0	UG	0	0	0	if_e1
172.16.110.0	0.0.0.0	255.255.255.0	U	0	0	0	if_e1
172.16.111.0	172.16.110.254	255.255.255.0	UG	0	0	0	if_e1

Figure 16: Routes on Tux3

```
root@tux114:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref  Use Iface
0.0.0.0         10.227.20.254   0.0.0.0       UG    100    0      0 if_mng
10.227.20.0     0.0.0.0        255.255.255.0  U      100    0      0 if_mng
172.16.1.0      172.16.111.254 255.255.255.0  UG      0    0      0 if_e2
172.16.110.0    0.0.0.0        255.255.255.0  U      0    0      0 if_e1
172.16.111.0    0.0.0.0        255.255.255.0  U      0    0      0 if_e2
```

Figure 17: Routes on Tux4

```
root@tux112:~# traceroute 172.16.110.1
traceroute to 172.16.110.1 (172.16.110.1), 30 hops max, 60 byte packets
1  172.16.111.253 (172.16.111.253)  0.222 ms  0.207 ms  0.190 ms
2  172.16.110.1 (172.16.110.1)  0.450 ms  0.435 ms  0.428 ms
3  172.16.110.1 (172.16.110.1)  0.450 ms  0.435 ms  0.428 ms
```

Figure 18: Traceroute from Tux2 to Tux3, with no redirection rules applied

```
root@tux112:~# traceroute 172.16.110.1
traceroute to 172.16.110.1 (172.16.110.1), 30 hops max, 60 byte packets
1  172.16.111.254 (172.16.111.254)  0.348 ms  0.314 ms  0.298 ms
2  172.16.111.253 (172.16.111.253)  0.648 ms  0.641 ms  0.634 ms
3  172.16.110.1 (172.16.110.1)  0.875 ms  0.859 ms  0.853 ms
```

Figure 19: Traceroute from Tux2 to Tux3, with a forced redirection on the Router

```
root@tux112:~# traceroute 172.16.110.1
traceroute to 172.16.110.1 (172.16.110.1), 30 hops max, 60 byte packets
1  172.16.111.253 (172.16.111.253)  0.237 ms  0.196 ms  0.182 ms
2  172.16.110.1 (172.16.110.1)  0.370 ms  0.379 ms  0.359 ms
```

Figure 20: Traceroute from Tux2 to Tux3, with ICMP redirection rules applied

1	0.000000000	Routerboardc_1c:8d:32	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:07 Cost = 10 Port = 0x8001
2	2.002011017	Routerboardc_1c:8d:32	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:07 Cost = 10 Port = 0x8001
3	3.584337077	0.0.0.0	255.255.255.255	MNDP	160	5678 → 5678 Len=118
4	3.584337618	Routerboardc_1c:8d:32	CDP/FTP/DTP/PagP/UDLD	CDP	94	Device ID: MikroTik Port ID: bridge111
5	3.584368832	Routerboardc_1c:8d:32	Nearest-Bridge	LLDP	111	MA/c4:ad:34:1c:8d:32 IN:bridge111 120 SysN=MikroTik SysD=MikroTik Route
6	4.004079708	Routerboardc_1c:8d:32	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:07 Cost = 10 Port = 0x8001
7	6.006251894	Routerboardc_1c:8d:32	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:07 Cost = 10 Port = 0x8001
8	8.007986051	Routerboardc_1c:8d:32	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:07 Cost = 10 Port = 0x8001
9	9.488337905	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x689b, seq=1/256, ttl=64 (reply in 11)
10	9.488610542	Routerboardc_eb:24:07	Broadcast	ARP	60	Who has 172.16.111.253? Tell 172.16.111.254
11	9.489161776	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x689b, seq=1/256, ttl=63 (request in 9)
12	9.774762511	192.168.88.1	255.255.255.255	MNDP	153	5678 → 5678 Len=111
13	9.774762940	Routerboardc_eb:24:07	CDP/FTP/DTP/PagP/UDLD	CDP	108	Device ID: MikroTik Port ID: bridge
14	10.010397171	Routerboardc_1c:8d:32	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:07 Cost = 10 Port = 0x8001
15	10.495846574	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x689b, seq=2/512, ttl=64 (reply in 19)
16	10.496664675	Routerboardc_eb:24:07	Broadcast	ARP	60	Who has 172.16.111.17 Tell 172.16.111.254
17	10.496680798	TPLink_c2:17:51	Routerboardc_eb:24:07	ARP	42	172.16.111.1 is at ec:75:0:c:21:7:51
18	10.496216960	172.16.111.254	172.16.111.1	ICMP	126	Redirect (Redirect for host)
19	10.496384469	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x689b, seq=2/512, ttl=63 (request in 15)
20	10.519840676	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x689b, seq=3/768, ttl=64 (reply in 21)
21	11.520302928	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x689b, seq=3/768, ttl=63 (request in 20)
22	12.012322192	Routerboardc_1c:8d:32	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:07 Cost = 10 Port = 0x8001
23	12.543839688	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x689b, seq=4/1024, ttl=64 (reply in 24)
24	12.544296672	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x689b, seq=4/1024, ttl=63 (request in 23)
25	13.567838800	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x689b, seq=5/1280, ttl=64 (reply in 26)
26	13.568302016	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x689b, seq=5/1280, ttl=63 (request in 25)
27	14.014424061	Routerboardc_1c:8d:32	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:07 Cost = 10 Port = 0x8001
28	14.559774287	TPLink_c2:17:51	Routerboardc_eb:24:07	ARP	42	Who has 172.16.111.254? Tell 172.16.111.1
29	14.559888186	Routerboardc_eb:24:07	TPLink_c2:17:51	ARP	60	172.16.111.254 is at 74:4d:28:eb:24:07
30	14.591835462	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x689b, seq=6/1536, ttl=64 (reply in 31)
31	14.592270881	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x689b, seq=6/1536, ttl=63 (request in 30)
32	14.728011718	TPLink_c2:2d:dc	TPLink_c2:17:51	ARP	60	Who has 172.16.111.1? Tell 172.16.111.253
33	14.728034405	TPLink_c2:17:51	TPLink_c2:2d:dc	ARP	42	172.16.111.1 is at ec:75:0:c:21:7:51
34	15.615839726	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x689b, seq=7/1792, ttl=64 (reply in 35)
35	15.616299538	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x689b, seq=7/1792, ttl=63 (request in 34)
36	16.016513740	Routerboardc_1c:8d:32	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:07 Cost = 10 Port = 0x8001
37	16.607786681	TPLink_c2:17:51	TPLink_c2:2d:dc	ARP	42	Who has 172.16.111.253? Tell 172.16.111.1
38	16.608824494	TPLink_c2:2d:dc	TPLink_c2:17:51	ARP	60	172.16.111.253 is at ec:75:0:c:21:2d:dc
39	16.639835644	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x689b, seq=8/2048, ttl=64 (reply in 40)
40	16.640273785	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x689b, seq=8/2048, ttl=63 (request in 39)
41	17.663839684	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x689b, seq=9/2304, ttl=64 (reply in 42)
42	17.664290303	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x689b, seq=9/2304, ttl=63 (request in 41)
43	18.008532395	Routerboardc_1c:8d:32	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:07 Cost = 10 Port = 0x8001
44	18.6867835570	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x689b, seq=10/2560, ttl=64 (reply in 45)
45	18.688291075	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x689b, seq=10/2560, ttl=63 (request in 44)
46	19.711833778	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x689b, seq=11/2816, ttl=64 (reply in 47)
47	19.712264288	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x689b, seq=11/2816, ttl=63 (request in 46)
48	20.010623125	Routerboardc_1c:8d:32	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:07 Cost = 10 Port = 0x8001
49	22.012627017	Routerboardc_1c:8d:32	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:07 Cost = 10 Port = 0x8001

Figure 22: Ping on Tux2, with ICMP redirection rules applied

1	0.000000000	Routerboardc_1c:8d:32	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:07 Cost = 10 Port = 0x8001
2	1.729711260	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x6735, seq=1/256, ttl=64 (reply in 3)
3	1.730224503	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x6735, seq=1/256, ttl=63 (request in 2)
4	1.992159871	Routerboardc_1c:8d:32	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:07 Cost = 10 Port = 0x8001
5	2.754561544	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x6735, seq=2/512, ttl=64 (reply in 7)
6	2.754757311	172.16.111.254	172.16.111.1	ICMP	126	Redirect (Redirect for host)
7	2.755025361	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x6735, seq=2/512, ttl=63 (request in 5)
8	3.778529155	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x6735, seq=3/768, ttl=64 (reply in 10)
9	3.778772626	172.16.111.254	172.16.111.1	ICMP	126	Redirect (Redirect for host)
10	3.779053160	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x6735, seq=3/768, ttl=63 (request in 8)
11	3.994295673	Routerboardc_1c:8d:32	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:07 Cost = 10 Port = 0x8001
12	4.802520436	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x6735, seq=4/1024, ttl=64 (reply in 14)
13	4.8027229847	172.16.111.254	172.16.111.1	ICMP	126	Redirect (Redirect for host)
14	4.803065121	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x6735, seq=4/1024, ttl=63 (request in 12)
15	5.826523599	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x6735, seq=5/1280, ttl=64 (reply in 17)
16	5.826734438	172.16.111.254	172.16.111.1	ICMP	126	Redirect (Redirect for host)
17	5.827091534	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x6735, seq=5/1280, ttl=63 (request in 15)
18	5.964657570	Routerboardc_1c:8d:32	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:07 Cost = 10 Port = 0x8001
19	6.858523729	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x6735, seq=6/1536, ttl=64 (reply in 21)
20	6.858726254	172.16.111.254	172.16.111.1	ICMP	126	Redirect (Redirect for host)
21	6.851855121	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x6735, seq=6/1536, ttl=63 (request in 19)
22	7.874520582	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x6735, seq=7/1792, ttl=64 (reply in 23)
23	7.875065348	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x6735, seq=7/1792, ttl=63 (request in 22)
24	7.998623993	Routerboardc_1c:8d:32	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:07 Cost = 10 Port = 0x8001
25	8.898519493	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x6735, seq=8/2048, ttl=64 (reply in 27)
26	8.898724048	172.16.111.254	172.16.111.1	ICMP	126	Redirect (Redirect for host)
27	8.899065168	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x6735, seq=8/2048, ttl=63 (request in 25)
28	9.922523453	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x6735, seq=9/2304, ttl=64 (reply in 29)
29	9.923058628	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x6735, seq=9/2304, ttl=63 (request in 28)
30	10.000782895	Routerboardc_1c:8d:32	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:07 Cost = 10 Port = 0x8001
31	10.039743767	0.0.0.0	255.255.255.255	MNDP	160	5678 → 5678 Len=118
32	10.039744251	Routerboardc_1c:8d:32	CDP/FTP/DTP/PagP/UDLD	CDP	94	Device ID: MikroTik Port ID: bridge111
33	10.039779840	Routerboardc_1c:8d:32	Nearest-Bridge	LLDP	111	MA/c4:ad:34:1c:8d:32 IN:bridge111 120 SysN=MikroTik SysD=MikroTik Route
34	10.946515198	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x6735, seq=10/2560, ttl=64 (reply in 35)
35	10.947632525	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x6735, seq=10/2560, ttl=63 (request in 34)
36	11.874468657	TPLink_c2:17:51	Routerboardc_eb:24:07	ARP	42	Who has 172.16.111.254? Tell 172.16.111.1
37	11.874606664	Routerboardc_eb:24:07	TPLink_c2:17:51	ARP	60	172.16.111.254 is at 74:4d:28:eb:24:07
38	11.970518845	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x6735, seq=11/2816, ttl=64 (reply in 40)
39	11.970679898	172.16.111.254	172.16.111.1	ICMP	126	Redirect (Redirect for host)
40	11.971087013	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x6735, seq=11/2816, ttl=63 (request in 38)
41	12.002462748	Routerboardc_1c:8d:32	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:07 Cost = 10 Port = 0x8001
42	12.994514743	172.16.111.1	172.16.110.1	ICMP	98	Echo (ping) request id=0x6735, seq=12/3072, ttl=64 (reply in 43)
43	12.995846896	172.16.110.1	172.16.111.1	ICMP	98	Echo (ping) reply id=0x6735, seq=12/3072, ttl=63 (request in 42)
44	14.004695009	Routerboardc_1c:8d:32	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:07 Cost = 10 Port = 0x8001
45	16.00683166	Routerboardc_1c:8d:32	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0/74:4d:28:eb:24:07 Cost = 10 Port = 0x8001

Figure 21: Ping on Tux2, with a forced redirection on the

1	0.000000000	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001
2	2.290048127	172.16.120.1	172.16.1.10	ICMP	60	Echo (ping) request id=0x4d7a, seq=1/256, ttl=64 (no response found!)
3	2.002151189	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001
4	2.311094138	172.16.120.1	172.16.1.10	ICMP	98	Echo (ping) request id=0x4d7a, seq=2/512, ttl=64 (no response found!)
5	3.335111018	172.16.120.1	172.16.1.10	ICMP	98	Echo (ping) request id=0x4d7a, seq=3/768, ttl=64 (no response found!)
6	4.004277686	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001
7	4.359098987	172.16.120.1	172.16.1.10	ICMP	98	Echo (ping) request id=0x4d7a, seq=4/1024, ttl=64 (no response found!)
8	5.38395125	172.16.120.1	172.16.1.10	ICMP	98	Echo (ping) request id=0x4d7a, seq=5/1280, ttl=64 (no response found!)
9	6.006418966	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001
10	6.407095942	172.16.120.1	172.16.1.10	ICMP	98	Echo (ping) request id=0x4d7a, seq=6/1536, ttl=64 (no response found!)
11	6.439042691	TPLink_c2:3:c:ac	TPLink_c2:3:c:75	ARP	42	Who has 172.16.120.254? Tell 172.16.120.1
12	6.439122436	TPLink_c2:3:c:75	TPLink_c2:3:c:ac	ARP	60	172.16.120.254 is at ec:75:0c:c2:3:c:75
13	7.413099491	172.16.120.1	172.16.1.10	ICMP	98	Echo (ping) request id=0x4d7a, seq=7/1792, ttl=64 (no response found!)
14	8.008506616	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001
15	8.455099990	172.16.120.1	172.16.1.10	ICMP	98	Echo (ping) request id=0x4d7a, seq=8/2048, ttl=64 (no response found!)
16	9.479096506	172.16.120.1	172.16.1.10	ICMP	98	Echo (ping) request id=0x4d7a, seq=9/2304, ttl=64 (no response found!)
17	16.016685142	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001
18	16.503999897	172.16.120.1	172.16.1.10	ICMP	98	Echo (ping) request id=0x4d7a, seq=10/2560, ttl=64 (no response found!)
19	11.527102681	172.16.120.1	172.16.1.10	ICMP	98	Echo (ping) request id=0x4d7a, seq=11/2816, ttl=64 (no response found!)
20	12.012872885	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001
21	12.551096684	172.16.120.1	172.16.1.10	ICMP	98	Echo (ping) request id=0x4d7a, seq=12/3072, ttl=64 (no response found!)
22	13.575899919	172.16.120.1	172.16.1.10	ICMP	98	Echo (ping) request id=0x4d7a, seq=13/3328, ttl=64 (no response found!)
23	14.014975259	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001
24	14.599118242	172.16.120.1	172.16.1.10	ICMP	98	Echo (ping) request id=0x4d7a, seq=14/3584, ttl=64 (no response found!)
25	15.623100231	172.16.120.1	172.16.1.10	ICMP	98	Echo (ping) request id=0x4d7a, seq=15/3840, ttl=64 (no response found!)
26	16.017139021	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001
27	16.647113704	172.16.120.1	172.16.1.10	ICMP	98	Echo (ping) request id=0x4d7a, seq=16/4096, ttl=64 (no response found!)
28	17.671099802	172.16.120.1	172.16.1.10	ICMP	98	Echo (ping) request id=0x4d7a, seq=17/4352, ttl=64 (no response found!)
29	18.019262556	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001
30	18.695096493	172.16.120.1	172.16.1.10	ICMP	98	Echo (ping) request id=0x4d7a, seq=18/4608, ttl=64 (no response found!)
31	19.719993015	172.16.120.1	172.16.1.10	ICMP	98	Echo (ping) request id=0x4d7a, seq=19/4864, ttl=64 (no response found!)
32	20.021396857	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001
33	20.743092852	172.16.120.1	172.16.1.10	ICMP	98	Echo (ping) request id=0x4d7a, seq=20/5120, ttl=64 (no response found!)
34	21.767989297	172.16.120.1	172.16.1.10	ICMP	98	Echo (ping) request id=0x4d7a, seq=21/5376, ttl=64 (no response found!)
35	22.023536038	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001
36	22.791108253	172.16.120.1	172.16.1.10	ICMP	98	Echo (ping) request id=0x4d7a, seq=22/5632, ttl=64 (no response found!)
37	24.025682033	Routerboardc_1c:a3:2c	Nearest-Customer-Bridge	STP	60	RST. Root = 32768/0:c4:ad:34:1c:a3:2c Cost = 0 Port = 0x8001

Figure 23: Ping on Internet from Tux3, no NAT

7.3.5 Experience 5

1	0.000000000	Cisco_b6:8c:04	Nearest-Customer-Bridge	TCP	60	RST. Root = 489/1/38/37/6d:d4:1c:00 Cost = 228019 Port = 0x8084
2	0.403151809	10.5376.178.122	142.258.178.163	TCP	60	RST. Root = 489/1/38/37/6d:d4:1c:00 Cost = 228019 Port = 0x8084
3	0.408762425	142.258.178.163	10.227.28.112	TCP	66	[TCP ACKed unseen segment] 80 - 41728 [ACK Seq=1 Ack2=2 Win=1840 Len=8 Tsva=L-633391469 Tscr=3684736778
4	1.432196116	10.227.28.112	142.258.178.163	TCP	66	[TCP ACKed unseen segment] 80 - 41728 [ACK Seq=1 Ack2=2 Win=1840 Len=8 Tsva=L-633391469 Tscr=3684736778
5	1.432849918	142.258.178.163	10.227.28.112	TCP	68	Standard query response 0x8fc1 A ni.fe.up.pt A 193.136.38.172
6	1.432196116	10.227.28.112	142.258.178.163	TCP	68	Standard query response 0x8fc1 A ni.fe.up.pt A 193.136.38.172
7	1.432196116	10.227.28.112	228.8.8.251	MNDS	68	Standard query response 0x8fc1 A ni.fe.up.pt A 193.136.38.172
8	1.688182991	10.227.28.112	142.258.178.163	TCP	66	[TCP ACKed unseen segment] 80 - 41688 [ACK Seq=1 Ack2=2 Win=1840 Len=8 Tsva=L-37817627735 Tscr=3684738858
9	1.688771029	142.258.178.163	10.227.28.112	TCP	66	[TCP ACKed unseen segment] 80 - 41688 [ACK Seq=1 Ack2=2 Win=1840 Len=8 Tsva=L-37817627735 Tscr=3684738858
10	2.811199802	Cisco_b6:8c:04	Nearest-Customer-Bridge	STP	60	RST. Root = 489/1/38/37/6d:d4:1c:00 Cost = 228019 Port = 0x8084
11	2.84421289	10.227.28.112	142.258.178.163	TCP	71	Standard query 0x8fc1 A ni.fe.up.pt A 193.136.38.172
12	2.844692593	10.227.28.112	142.258.178.163	TCP	71	Standard query 0x8fc1 A ni.fe.up.pt A 193.136.38.172
13	2.877859523	10.227.28.112	142.258.178.163	TCP	87	Standard query response 0x8fc1 A ni.fe.up.pt A 193.136.38.172
14	2.8785998	10.227.28.112	142.258.178.163	TCP	87	Standard query response 0x8fc1 A ni.fe.up.pt A 193.136.38.172
15	2.873373757	10.227.28.112	193.136.38.172	TCP	98	Echo (ping) request id=0x244a, seq=1/256, ttl=64 (no response found!)
16	2.453659387	Cisco_b6:8c:04	Cisco_b6:8c:04	LOOP	68	Rep
17	3.021396857	10.227.28.112	142.258.178.163	TCP	68	Echo (ping) request id=0x244a, seq=2/512, ttl=64 (no response found!)
18	3.358445683	fe80::1e7f:dfff:fe70:195d1	228.8.8.172	MNDS	68	Echo (ping) request id=0x244a, seq=3/1024, ttl=64 (no response found!)
19	3.358445989	10.227.28.33	224.8.8.251	MNDS	68	Echo (ping) request id=0x244a, seq=4/2048, ttl=64 (no response found!)
20	4.025747271	Cisco_b6:8c:04	Nearest-Customer-Bridge	STP	68	RST. Root = 489/1/38/37/6d:d4:1c:00 Cost = 228019 Port = 0x8084
21	4.128251281	10.227.28.112	193.136.38.172	TCP	98	Echo (ping) request id=0x244a, seq=5/4096, ttl=64 (no response found!)
22	4.128251281	10.227.28.112	142.258.178.163	TCP	98	Echo (ping) request id=0x244a, seq=6/8192, ttl=64 (no response found!)
23	4.128251281	10.227.28.112	142.258.178.163	TCP	98	Echo (ping) request id=0x244a, seq=7/16384, ttl=64 (no response found!)
24	5.273318667	142.258.178.163	10.227.28.112	TCP	66	[TCP ACKed unseen segment] 80 - 41714 [ACK Seq=1 Ack2=2 Win=1840 Len=8 Tsva=L-2832974127 Tscr=3684741642
25	5.423691598	Cisco_b6:8c:04	CDP/VT/DT/P/Agp/UDL	CDP	68	Device ID: Switch Port Id: FastEthernet0/4
26	6.050198103	10.227.28.112	142.258.178.163	TCP	68	Standard query 0x8000 PTR _ftp._tcp.local, "QDN" question PTR _nfs._tcp.local, "QDN" question PTR _afs._tcp.local, "QDN" question PTR _smb._tcp..
27	6.050198103	10.227.28.112	142.258.178.163	TCP	68	Standard query 0x8000 PTR _ftp._tcp.local, "QDN" question PTR _nfs._tcp.local, "QDN" question PTR _afs._tcp.local, "QDN" question PTR _smb._tcp..
28	6.344613669	fe80::1e7f:dfff:fe70:18cc5	228.8.8.172	TCPv6	62	Router Solicitation
29	7.192245748	10.227.28.112	193.136.38.172	TCP	98	Echo (ping) request id=0x244a, seq=8/1536, ttl=64 (no response found!)
30	7.638023586	Routerboardc_20:25:c8	Dell_70:8c:5b	ARP	68	Who has 10.227.28.112? Tell 10.227.28.256
31	7.63804384	Dell_70:8c:5b	Routerboardc_20:25:c8	ARP	68	10.227.28.112 is at 00:0dd:78:8c:5b
32	7.63804384	10.227.28.112	142.258.178.163	TCP	98	Echo (ping) request id=0x244a, seq=9/2560, ttl=64 (no response found!)
33	8.216249466	10.227.28.112	193.136.38.172	TCP	98	Echo (ping) request id=0x244a, seq=10/5120, ttl=64 (no response found!)
34	9.059813346	Cisco_b6:8c:04	Nearest-Customer-Bridge	STP	68	Rep
35	9.059813667	Cisco_b6:8c:04	CDP/VT/DT/P/Agp/UDL	CDP	68	DYNAMIC Trunk Protocol
36	9.240248272	10.227.28.112	193.136.38.172	TCP	98	Echo (ping) request id=0x244a, seq=11/7680, ttl=64 (no response found!)
37	9.240248272	10.227.28.112	142.258.178.163	TCP	66	[TCP Dup ACK 3 1] 80 - 41728 [ACK Seq=1 Ack2=2 Win=1840 Len=8 Tsva=L-2832974127 Tscr=3684741642
38	9.624196116	10.227.28.112	34.197.221.82	TCP	66	[TCP Dup ACK 3 1] 80 - 41728 [ACK Seq=1 Ack2=2 Win=1840 Len=8 Tsva=L-2832974127 Tscr=3684741642
39	9.624723887	34.197.221.82	10.227.28.112	TCP	66	[TCP ACKed unseen segment] 80 - 50410 [ACK Seq=1 Ack2=2 Win=1840 Len=8 Tsva=L-2870852593 Tscr=10590866472
40	9.624723887	34.197.221.82	10.227.28.112	TCP	66	[TCP ACKed unseen segment] 80 - 50488 [ACK Seq=1 Ack2=2 Win=1840 Len=8 Tsva=L-2870852593 Tscr=10590866472
41	10.066337958	Cisco_b6:8c:04	Nearest-Customer-Bridge	STP	68	RST. Root = 489/1/38/37/6d:d4:1c:00 Cost = 228019 Port = 0x8084
42	10.403151824	10.227.28.112	142.258.178.163	TCP	98	Echo (ping) request id=0x244a, seq=23/2894, ttl=64 (no response found!)
43	10.403151824	10.227.28.112	142.258.178.163	TCP	66	[TCP ACKed unseen segment] 80 - 41728 [ACK Seq=1 Ack2=2 Win=1840 Len=8 Tsva=L-283391158 Tscr=3684747818
44	10.408473267	142.258.178.163	10.227.28.112	TCP	66	[TCP Dup ACK 3 1] 80 - 41728 [ACK Seq=1 Ack2=2 Win=1840 Len=8 Tsva=L-283391158 Tscr=3684747818
45	11.288242322	10.227.28.112	193.136.38.172	TCP	98	Echo (ping) request id=0x244a, seq=24/2560, ttl=64 (no response found!)
46	11.672196388	10.227.28.112	142.258.178.163	TCP	66	[TCP Dup ACK 3 1] 80 - 41728 [ACK Seq=1 Ack2=2 Win=1840 Len=8 Tsva=L-287762778 Tscr=3684748842
47	11.672196388	142.258.178.163	10.227.28.112	TCP	66	[TCP Dup ACK 3 1] 80 - 41728 [ACK Seq=1 Ack2=2 Win=1840 Len=8 Tsva=L-287762778 Tscr=3684748842
48	11.672196388	10.227.28.112	142.258.178.163	TCP	66	[TCP Dup ACK 3 1] 80 - 41728 [ACK Seq=1 Ack2=2 Win=1840 Len=8 Tsva=L-287762778 Tscr=3684748842
49	11.928771867	142.258.178.163	10.227.28.112	TCP	66	[TCP Dup ACK 3 1] 80 - 41728 [ACK Seq=1 Ack2=2 Win=1840 Len=8 Tsva=L-287762778 Tscr=3684748842
50	12.008018784	Cisco_b6:8c:04	Nearest-Customer-Bridge	STP	68	Rep
51	12.176145914	fe80::1a60:1f36:5602:1ed8	ff02::1	MNDS	187	Standard query 0x8000 PTR _lpp._tcp.local, "QDN" question PTR _lps._tcp.local, "QDN" question
52	12.176145244	10.227.28.85	224.0.8.251	MNDS	87	

1	0.000000000	Cisco_b6:8c:05	Nearest-Customer-Bridge	STP	68	RST. Root = 0:096/1/30:37:a6:d4:1:c:00	Cost = 220819	Port = 0x8005
2	1.641290252	16.227.20.113	18.227.20.3	DNS	71	Standard query 0x7019 A youtube.com		
3	1.642117982	16.227.20.113	18.227.20.3	DNS	71	Standard query 0x7019 A youtube.com		
4	1.642117982	16.227.20.113	18.227.20.113	DNS	87	Standard query response 0xd0d1 AAA youtube.com AAAA 2a01:1450:4003:80e::200e		
5	1.642118081	16.227.20.3	18.227.20.113	DNS	99	Standard query response 0xd0d1 AAAA youtube.com AAAA 2a01:1450:4003:80e::200e		
6	1.642456297	16.227.20.113	142.251.140.238	ICMP	98	Echo (ping) request id=0x0bc3, seq=1/256, ttl=64 (request in 7)		
7	1.656944598	142.251.140.238	18.227.20.113	ICMP	98	Echo (ping) reply id=0x0bc3, seq=1/256, ttl=12 (request in 6)		
8	1.657205038	16.227.20.113	18.227.20.3	DNS	88	Standard query 0x052a PTR 238.140.251.142.in-addr.arpa PTR lcmda-ab-in-f14.1e100.net PTR dia01s03-in-f14.1e100.net		
9	1.657972661	16.227.20.3	18.227.20.113	DNS	186	Standard query response 0x052a PTR 238.140.251.142.in-addr.arpa PTR lcmda-ab-in-f14.1e100.net PTR dia01s03-in-f14.1e100.net		
10	2.01338186	Cisco_b6:8c:05	Nearest-Customer-Bridge	STP	68	RST. Root = 0:096/1/30:37:a6:d4:1:c:00	Cost = 220819	Port = 0x8005
11	2.644159637	16.227.20.113	142.251.140.238	ICMP	98	Echo (ping) request id=0x0bc3, seq=2/512, ttl=64 (reply in 12)		
12	2.644159637	16.227.20.113	18.227.20.3	DNS	98	Echo (ping) reply id=0x0bc3, seq=2/512, ttl=12 (request in 11)		
13	2.659381943	16.227.20.113	18.227.20.113	DNS	88	Standard query 0x071 PTR 238.140.251.142.in-addr.arpa		
14	2.658122710	16.227.20.3	18.227.20.113	DNS	186	Standard query response 0x071 PTR 238.140.251.142.in-addr.arpa PTR dia01s03-in-f14.1e100.net PTR lcmda-ab-in-f14.1e100.net		
15	2.6584658742	fe80::5e91:ddff:fe70:99ff	ff02::2	ICMPv6	70	Router Solicitation from 5c:91:dd:70:99:ff		
16	3.645318735	16.227.20.113	142.251.140.238	ICMP	98	Echo (ping) request id=0x0bc3, seq=3/768, ttl=64 (reply in 17)		
17	3.655918977	142.251.140.238	18.227.20.113	ICMP	98	Echo (ping) reply id=0x0bc3, seq=3/768, ttl=12 (request in 16)		
18	3.668291996	16.227.20.113	18.227.20.3	DNS	88	Standard query 0x6745 PTR 238.140.251.142.in-addr.arpa		
19	3.661133883	16.227.20.3	18.227.20.113	DNS	186	Standard query response 0x6745 PTR 238.140.251.142.in-addr.arpa PTR dia01s03-in-f14.1e100.net PTR lcmda-ab-in-f14.1e100.net		
20	4.679200091	fe80::5e91:ddff:fe83:8d46	ff02::2	ICMPv6	78	Router Solicitation from bc:24:11:83:8d:46		
21	4.679200091	Cisco_b6:8c:05	Nearest-Customer-Bridge	STP	60	RST. Root = 0:096/1/30:37:a6:d4:1:c:00	Cost = 220819	Port = 0x8005
22	4.337652948	fe80::5e91:ddff:fe70:99ff	ff02::1:b	ICMP	107	Standard query 0x0800 PTR _ipns._tcp.local, "QM" question		
23	4.3376529421	16.227.20.14	224.0.2.251	DNS	87	Standard query 0x0800 PTR _ipns._tcp.local, "QM" question		
24	4.647289211	16.227.20.113	142.251.140.238	ICMP	98	Echo (ping) request id=0x0bc3, seq=4/1024, ttl=64 (reply in 25)		
25	4.6510980134	142.251.140.238	18.227.20.113	ICMP	98	Echo (ping) reply id=0x0bc3, seq=4/1024, ttl=12 (request in 24)		
26	4.662721481	16.227.20.113	18.227.20.3	DNS	88	Standard query 0x5f0 PTR 238.140.251.142.in-addr.arpa		
27	4.663802898	16.227.20.3	18.227.20.113	DNS	186	Standard query response 0x5f0 PTR 238.140.251.142.in-addr.arpa PTR lcmda-ab-in-f14.1e100.net PTR dia01s03-in-f14.1e100.net		
28	4.664089587	16.227.20.83	229.255.255.250	ICMPv6	78	Router Solicitation from bc:24:11:83:8d:46		
29	4.664089587	16.227.20.83	18.227.20.113	DNS	649	Standard query 0x0800 PTR 238.140.251.142.in-addr.arpa		
30	4.664089583	fe80::5e91:ddff:fe70:99ff	ff02::1:c	ICMP	669	41632 - 3782 Len=607		
31	4.695577962	fe80::5e91:ddff:fe70:99ff	ff02::1:c	ICMPv6	98	Multicast Listener Report Message v2		
32	5.339287782	16.227.20.82	239.255.255.250	UDP/XML	649	40984 - 3782 Len=607		
33	5.339288153	fe80::5e91:ddff:fe70:99ff	ff02::1:c	ICMPv6	669	34336 - 3782 Len=607		
34	5.344184147	fe80::5e91:ddff:fe70:99ff	ff02::1:c	ICMPv6	98	Multicast Listener Report Message v2		
35	5.649275684	142.251.140.238	18.227.20.113	ICMP	98	Echo (ping) request id=0x0bc3, seq=5/1280, ttl=64 (reply in 36)		
36	5.6639168167	142.251.140.238	18.227.20.113	ICMP	98	Echo (ping) reply id=0x0bc3, seq=5/1280, ttl=12 (request in 35)		
37	5.6639168167	16.227.20.113	18.227.20.113	DNS	88	Standard query 0x5f1 PTR 238.140.251.142.in-addr.arpa		
38	5.665123351	16.227.20.113	18.227.20.113	DNS	186	Standard query response 0x5f1 PTR 238.140.251.142.in-addr.arpa PTR dia01s03-in-f14.1e100.net PTR lcmda-ab-in-f14.1e100.net		
39	5.669865466	16.227.20.83	239.255.255.250	UDP/XML	649	43708 - 3782 Len=607		
40	5.669865466	fe80::5e91:ddff:fe70:99ff	ff02::1:c	ICMPv6	669	42877 - 3782 Len=607		
41	5.784085714	fe80::5e91:ddff:fe70:99ff	ff02::1:c	ICMPv6	98	Multicast Listener Report Message v2		
42	6.202984398	fe80::5e91:ddff:fe70:99ff	ff02::1:c	ICMPv6	62	Router Solicitation		
43	6.831468477	fe80::5e91:ddff:fe70:99ff	ff02::1:c	ICMPv6	98	Multicast Listener Report Message v2		
44	6.940814783	Cisco_b6:8c:05	Nearest-Customer-Bridge	STP	68	RST. Root = 0:096/1/30:37:a6:d4:1:c:00	Cost = 220819	Port = 0x8005
45	6.940814783	fe80::5e91:ddff:fe70:99ff	ff02::1:c	ICMPv6	98	Multicast Listener Report Message v2		
46	6.95815297	16.227.20.113	142.251.140.238	ICMP	98	Echo (ping) request id=0x0bc3, seq=6/1536, ttl=64 (reply in 49)		
47	6.958162586	Routerboardc_20:25:c8	Dell_70:97:44	ARP	60	Who has 10.227.20.113 Tell 10.227.20.254		
48	6.958176912	Dell_70:97:44	Routerboardc_20:25:c8	ARP	42	10.227.20.113 is at bc:24:11:70:97:44		
49	6.964894878	142.251.140.238	18.227.20.113	ICMP	98	Echo (ping) reply id=0x0bc3, seq=6/1536, ttl=12 (request in 46)		
50	6.965259819	16.227.20.113	18.227.20.3	DNS	88	Standard query 0x4d2c PTR 238.140.251.142.in-addr.arpa		
51	6.966661869	16.227.20.3	18.227.20.113	DNS	102	Standard query response 0x4d2c PTR 238.140.251.142.in-addr.arpa PTR lcmda-ab-in-f14.1e100.net PTR dia01s03-in-f14.1e100.net		
52	6.915751448	Dell_70:97:44	Routerboardc_20:25:c8	ARP	42	Who has 10.227.20.254 Tell 10.227.20.113		
53	6.915751448	Routerboardc_20:25:c8	Dell_70:97:44	ARP	42	10.227.20.254 is at bc:24:11:70:97:44		
54	6.916444905	Dell_70:97:44	Routerboardc_20:25:c8	ARP	60	Who has 10.227.20.113 Tell 10.227.20.254		
55	6.916421456	Proxoserve_e7:5e:5b	Dell_70:97:44	ARP	60	Who has 10.227.20.113 is at bc:24:11:70:97:44		
56	6.857922585	Proxoserve_e7:5e:5b	Dell_70:97:44	ARP	60	Who has 10.227.20.113 is at 5:c9:91:dd:70:97:44		
57	6.857943283	Dell_70:97:44	Proxoserve_e7:5e:5b	ARP	42	10.227.20.113 is at 5:c9:91:dd:70:97:44		

Figure 25: Ping to youtube.com from Tux3

1	0.000000000	10.227.20.114	10.227.20.94	UDP	86	6000 - 6000 Len=44		
2	0.000993349	10.227.20.94	10.227.20.114	UDP	85	6000 - 6000 Len=44		
3	0.500335941	10.227.20.114	10.227.20.94	UDP	86	6000 - 6000 Len=44		
4	0.500458185	10.227.20.114	10.227.20.94	UDP	151	6000 - 6000 Len=109		
5	0.501117843	10.227.20.94	10.227.20.114	UDP	85	6000 - 6000 Len=43		
6	0.504485813	Proxoserve_e7:f1:a9	Broadcast	ARP	60	Who has 10.227.20.1877 Tell 10.227.20.224		
7	0.611118953	10.227.20.159..11	10.227.20.114	ICMP	179	Destination unreachable (Port unreachable)		
8	0.629267875	Cisco_b6:8c:06	Cisco_b6:8c:06	Loop	60	Reply		
9	0.660324776	Proxoserve_e7:a5:24	Broadcast	ARP	60	Who has 10.227.20.2117 Tell 10.227.20.2		
10	0.661414255	10.227.20.114	10.227.20.113	ICMP	73	Standard query 0x0c17d A sigarra.up.pt		
11	0.661414255	10.227.20.114	10.227.20.113	DNS	73	Standard query 0x0c17d AAA sigarra.up.pt		
12	0.6921117110	10.227.20.114	10.227.20.113	DNS	116	Standard query response 0xc071 AAAA sigarra.up.pt SOA ns1.up.pt		
13	1.000701619	10.227.20.114	10.227.20.94	ICMP	86	6000 - 6000 Len=44		
14	1.001743235	10.227.20.94	10.227.20.114	ICMP	85	6000 - 6000 Len=44		
15	1.006948825	10.227.20.114	10.227.20.114	DNS	89	Standard query response 0x167d A sigarra.up.pt A 193.137.35.140		
16	1.007136556	10.227.20.114	193.137.35.140	ICMP	98	Echo (ping) request id=0x2a79, seq=2/256, ttl=64 (reply in 17)		
17	1.010893517	193.137.35.140	10.227.20.114	ICMP	98	Echo (ping) reply id=0x2a79, seq=2/256, ttl=12 (request in 16)		
18	1.011359504	10.227.20.114	10.227.20.113	DNS	87	Standard query 0x7d4a PTR 140.35.137.193.in-addr.arpa		
19	1.059645455	10.227.20.114	10.227.20.114	DNS	114	Standard query response 0x7d4a PTR 140.35.137.193.in-addr.arpa		
20	1.275590734	Cisco_b6:8c:06	Nearest-Customer-Bridge	STP	60	NSC Root = 0:096/1/30:37:a6:d4:1:c:00	Cost = 220806	
21	1.501820595	10.227.20.114	10.227.20.94	UDP	86	6000 - 6000 Len=44		
22	1.501817777	10.227.20.94	10.227.20.114	UDP	85	6000 - 6000 Len=43		
23	1.955172053	10.227.20.114	64.233.184.94	TCP	66	45478 - 80 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=4219205138 TSecr=2888546358		
24	1.955750589	64.233.184.94	10.227.20.114	ICMP	66	[TCP ACKed unseen segment] 80 - 45478 [ACK] Seq=1 Ack=2 Win=1048 Len=0 TSval=2888546391 TSecr=4219205138		
25	2.001349314	10.227.20.114	10.227.20.94	UDP	86	6000 - 6000 Len=44		
26	2.002801736	10.227.20.94	10.227.20.114	UDP	85	6000 - 6000 Len=43		
27	2.00871943	10.227.20.114	193.137.35.140	ICMP	98	Echo (ping) request id=0x2a79, seq=2/512, ttl=64 (reply in 28)		
28	2.010881001	193.137.35.140	10.227.20.114	ICMP	98	Echo (ping) reply id=0x2a79, seq=2/512, ttl=249 (request in 27)		
29	2.011192293	10.227.20.114	10.227.20.113	DNS	87	Standard query 0xc5f5 PTR 140.35.137.193.in-addr.arpa		
30	2.012018245	10.227.20.114	10.227.20.114	DNS	141	Standard query response 0xc5f5 PTR 140.35.137.193.in-addr.arpa PTR sigarra.up.pt		
31	2.501676211	10.227.20.114	10.227.20.94	ICMP	86	6000 - 6000 Len=44		
32	2.501815431	10.227.20.114	10.227.159..11	ICMP	151	6000 - 6000 Len=109		
33	2.502515029	10.227.20.114</						

7.3.6 Experience 6

8	12.726188120	172.16.128.1	172.16.1.18	TCP	74	43654 - 21 [FIN] Seq=8 Win=64248 MSS=1468 SACK_PERM Tsvl=>2335152582 Tsecr=>0 WS=128
9	12.727837118	172.16.128.1	172.16.1.18	TCP	66	43654 - 21 [ACK] Seq=1 Ack=8 Win=65168 Len=0 MSS=1468 SACK_PERM Tsvl=>2335152582 Tsecr=>2335152582 WS=128
10	12.727837114	172.16.128.1	172.16.1.18	FTP	86	Response: 220 (vsFTPd 3.0.5)
11	12.731453713	172.16.128.1	172.16.1.18	FTP	66	43654 - 21 [ACK] Seq=1 Ack=21 Win=64256 Len=0 Tsvl=>2335152583 Tsecr=>281001323
12	12.731488194	172.16.128.1	172.16.1.18	TCP	66	43654 - 21 [ACK] Seq=1 Ack=21 Win=64256 Len=0 Tsvl=>2335152582 Tsecr=>281001323
13	12.731553817	172.16.128.1	172.16.1.18	TCP	73	Request: USER
14	12.731553815	172.16.128.1	172.16.1.18	TCP	66	21 - 43654 [ACK] Seq=21 Ack=6 Win=65280 Len=0 Tsvl=>2810015282 Tsecr=>2335152587
15	12.732013311	172.16.128.1	172.16.1.18	FTP	77	Request: anonymous
16	12.732453634	172.16.1.18	172.16.1.18	TCP	66	21 - 43654 [ACK] Seq=21 Ack=17 Win=65288 Len=0 Tsvl=>2810015282 Tsecr=>2335152588
17	12.732453632	172.16.1.18	172.16.1.18	FTP	77	Request: anonymous Please specify the password.
18	12.732659628	172.16.128.1	172.16.1.18	FTP	73	Request: PASS
19	12.733458266	172.16.128.1	172.16.1.18	FTP	66	21 - 43654 [ACK] Seq=55 Ack=22 Win=65288 Len=0 Tsvl=>2810015170 Tsecr=>2335152589
20	12.733458265	172.16.128.1	172.16.1.18	TCP	77	Request: anonymous Please specify the password.
21	12.733458263	172.16.128.1	172.16.1.18	FTP	66	21 - 43654 [ACK] Seq=55 Ack=33 Win=65288 Len=0 Tsvl=>2810015170 Tsecr=>2335152629
22	12.779848186	172.16.1.18	172.16.1.18	FTP	89	Response: 230 Login successful.
23	12.779816387	172.16.128.1	172.16.1.18	FTP	72	Request: TYPE I
24	12.780016386	172.16.128.1	172.16.1.18	FTP	66	21 - 43654 [ACK] Seq=78 Ack=39 Win=65280 Len=0 Tsvl=>281001617 Tsecr=>2335152635
25	12.8286747812	172.16.128.1	172.16.1.18	FTP	66	Request: PWD
26	12.8286949168	172.16.1.18	172.16.128.1	TCP	66	21 - 43654 [ACK] Seq=8 Ack=41 Win=65288 Len=0 Tsvl=>281001617 Tsecr=>2335152676
27	12.821113752	172.16.1.18	172.16.128.1	TCP	97	Response: 200 Switching to binary mode.
28	12.821113750	172.16.1.18	172.16.128.1	TCP	66	21 - 43654 [ACK] Seq=8 Ack=46 Win=65288 Len=0 Tsvl=>2810016159 Tsecr=>2335152677
29	12.862491855	172.16.1.18	172.16.128.1	TCP	66	Request: pub/pipe.txt
30	12.862518826	172.16.128.1	172.16.1.18	FTP	75	Request: 213 189
31	12.863136075	172.16.1.18	172.16.128.1	FTP	70	Request: PASV
32	12.863136222	172.16.1.18	172.16.128.1	FTP	66	21 - 43654 [ACK] Seq=118 Ack=64 Win=65288 Len=0 Tsvl=>2810015170 Tsecr=>2335152718
33	12.863199681	172.16.128.1	172.16.1.18	FTP	66	21 - 43654 [ACK] Seq=118 Ack=64 Win=65288 Len=0 Tsvl=>2810015170 Tsecr=>2335152719
34	12.984468763	172.16.1.18	172.16.128.1	FTP	66	21 - 43654 [ACK] Seq=118 Ack=64 Win=65288 Len=0 Tsvl=>2810015170 Tsecr=>2335152719
35	12.984468762	172.16.1.18	172.16.128.1	FTP	66	21 - 43654 [ACK] Seq=118 Ack=64 Win=65288 Len=0 Tsvl=>2810015170 Tsecr=>2335152719
36	12.984468725	172.16.1.18	172.16.128.1	FTP	66	21 - 43654 [ACK] Seq=118 Ack=64 Win=65288 Len=0 Tsvl=>2810015170 Tsecr=>2335152719
37	12.985441675	172.16.1.18	172.16.128.1	FTP	116	Response: 227 Entering Passive Mode (172,16,1,18,198,198).
38	12.985441674	172.16.1.18	172.16.128.1	FTP	70	Request: PWD
39	12.985441673	172.16.1.18	172.16.128.1	FTP	66	33368 - 48633 [FIN] Seq=198 Win=65288 Len=0 Tsvl=>2335152762 Tsecr=>281001782 WS=128
40	12.986237955	172.16.1.18	172.16.128.1	TCP	66	33368 - 48633 [FIN] Seq=198 Win=65288 Len=0 Tsvl=>2335152762 Tsecr=>281001782 WS=128
41	12.986398095	172.16.128.1	172.16.1.18	FTP	71	Request: RETR
42	12.986398093	172.16.1.18	172.16.128.1	FTP	66	1 - 43654 [ACK] Seq=168 Ack=1 Win=65288 Len=0 Tsvl=>281001744 Tsecr=>2335152762
43	12.947482749	172.16.1.18	172.16.128.1	FTP	66	Request: pub/pipe.txt
44	12.947997153	172.16.1.18	172.16.128.1	FTP	66	21 - 43654 [ACK] Seq=168 Ack=85 Win=65288 Len=0 Tsvl=>281001744 Tsecr=>2335152803
45	12.948749628	172.16.1.18	172.16.128.1	FTP	137	Response: 138 Opening binary mode for data connection for pub/pipe.txt (189 bytes).
46	12.948749627	172.16.1.18	172.16.128.1	FTP-DATA	66	Request: 226 Transfer complete.
47	12.948798993	172.16.1.18	172.16.128.1	FTP	66	33368 - 48638 [ACK] Seq=198 Win=64128 Len=0 Tsvl=>2335152845 Tsecr=>281001745
48	12.948998688	172.16.1.18	172.16.128.1	FTP	66	48638 - 33368 [FIN] Seq=198 Win=65288 Len=0 Tsvl=>2335152845 Tsecr=>281001745
49	12.949012322	172.16.1.18	172.16.128.1	FTP	66	48638 - 33368 [FIN] Seq=198 Win=65288 Len=0 Tsvl=>2335152845 Tsecr=>281001745
50	12.949012381	172.16.1.18	172.16.128.1	FTP	66	48638 - 33368 [ACK] Seq=191 Ack=2 Win=65288 Len=0 Tsvl=>2335152845
51	12.949821263	172.16.1.18	172.16.128.1	FTP	66	Response: 226 Transfer complete.
52	12.949980874	172.16.128.1	172.16.1.18	FTP	66	43654 - 21 [ACK] Seq=85 Ack=263 Win=64256 Len=0 Tsvl=>2335152806 Tsecr=>281001745
53	12.950000038	172.16.1.18	172.16.128.1	FTP	66	Request: QUIT
54	12.991480954	172.16.1.18	172.16.128.1	FTP	66	21 - 43654 [ACK] Seq=263 Ack=89 Win=65288 Len=0 Tsvl=>281001748 Tsecr=>2335152806
55	12.991442751	172.16.128.1	172.16.1.18	FTP	68	Request: QUIT
56	12.991442750	172.16.1.18	172.16.128.1	FTP	66	1 - 43654 [ACK] Seq=263 Ack=91 Win=65288 Len=0 Tsvl=>281001748 Tsecr=>2335152847
57	12.992055257	172.16.1.18	172.16.128.1	FTP	66	21 - 43654 [FIN, ACK] Seq=277 Ack=91 Win=65288 Len=0 Tsvl=>281001748 Tsecr=>2335152847
58	12.992055256	172.16.1.18	172.16.128.1	FTP	66	43654 - 21 [FIN, ACK] Seq=278 Ack=78 Win=64256 Len=0 Tsvl=>2335152848 Tsecr=>281001748
59	12.992052094	172.16.1.18	172.16.128.1	FTP	66	RST, Root = 32768/8/141:34:1:1:a3:35 Cost = 0 Port = 0x8001
60	12.992052093	172.16.1.18	172.16.128.1	FTP	66	RST, Root = 32768/8/141:34:1:1:a3:35 Cost = 0 Port = 0x8001
61	12.91016282051	Nearest-Customer-Bridge	172.16.1.18	ARP	66	RST, Root = 32768/8/141:34:1:1:a3:35 Cost = 0 Port = 0x8001
62	16.016479337	Routerboard_c1c1a335	172.16.1.18	ARP	66	RST, Root = 32768/8/141:34:1:1:a3:35 Cost = 0 Port = 0x8001
63	17.740300409	TPLink_c23c175	172.16.1.18	ARP	66	RST, Root = 32768/8/141:34:1:1:a3:35 Cost = 0 Port = 0x8001
64	17.881535894	TPLink_c23c175	172.16.1.18	ARP	66	RST, Root = 32768/8/141:34:1:1:a3:35 Cost = 0 Port = 0x8001
65	18.910578774	Routerboard_c1c1a335	172.16.1.18	ARP	66	RST, Root = 32768/8/141:34:1:1:a3:35 Cost = 0 Port = 0x8001
66	28.826777959	Routerboard_c1c1a335	172.16.1.18	ARP	66	RST, Root = 32768/8/141:34:1:1:a3:35 Cost = 0 Port = 0x8001

Figure 27: Sucessful Download