Exercise for Lecture Software Defined Networking

Prof. Dr. David Hausheer, Julius Rückert
Christian Koch, Jeremias Blendin, Leonhard Nobach, Matthias Wichtlhuber

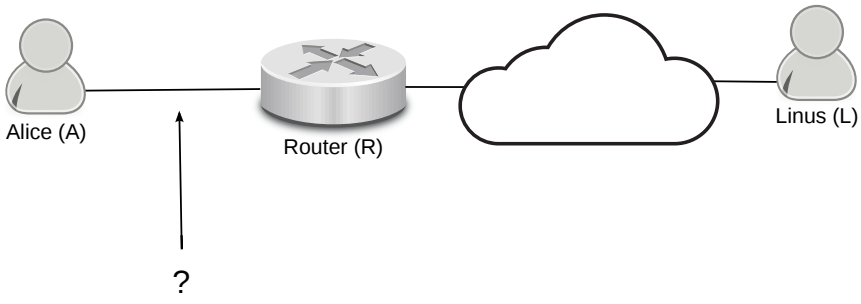
TECHNISCHE UNIVERSITÄT DARMSTADT

Winter Term 2015/16
Exercise No. 1
Published: 25.10.2016
Submission exclusively via Moodle, Deadline: 1.11.2016 before 4pm
Contact: Please use the Moodle forum to post questions and remarks on the exercise.
Web: `http://www.ps.tu-darmstadt.de/teaching/ws1617/sdn/`
Submission: `https://moodle.tu-darmstadt.de/course/view.php?id=8385`
– Example Solution –

Problem 1.1 - Cross-Layer Networking Basics

Note: This task is not directly related to the content in the slides, however you should be able to solve it with your previous knowledge of communication networks. Depending on your previous skills, it might require some online research. Nevertheless, the understanding of this topic will definitely help you with solving future exercises, labs, and exam questions related to SDN and OpenFlow!
Tip: Recap the following topics: the ISO/OSI layer model, Ethernet, MAC addresses, the Internet Protocol (IP, at least IPv4), ARP, ICMP, TCP (only the basics) and UDP.

Alice has a Linux PC. She connects to the Router via Ethernet, and her first action is to "ping" Linus's PC.



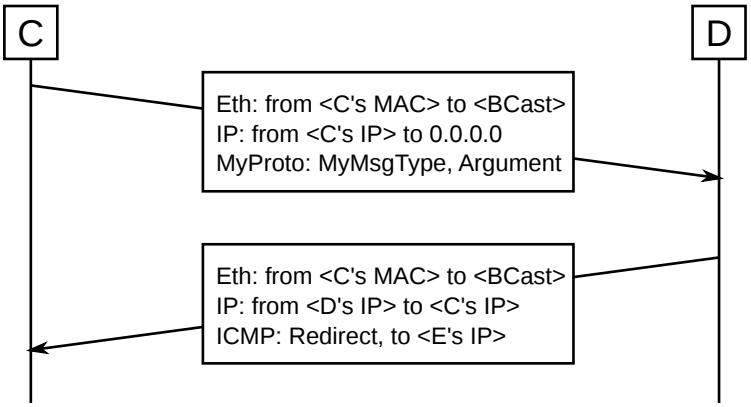
Assume the following:

- Alice uses no domain name, she just enters Linus's IP address for pinging
- Linus answers the ping, and the (Internet) connection between Linus and the Router is working as expected.
- Alice does not use DHCP, her IP address and default route was entered manually.
- No Layer 2 information between Alice and the router is known in advance.
- No other communication unrelated to the ping is taking place.

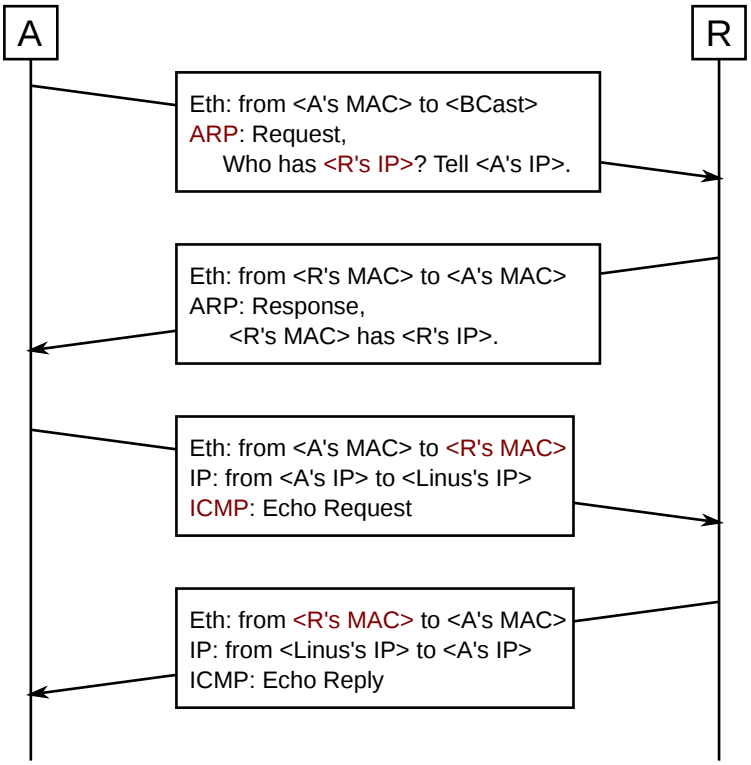
Draw the communication on the Ethernet link between Alice and the Router in a sequence diagram.

- Every message exchanged in your diagram shall contain all protocols of the ISO/OSI layer model that were used, starting with the data link layer.
- For every protocol, specify **data link and network addresses** that are used. As you don't know the actual addresses of anyone, you may write e.g. <IP address of X> or <X's IP> as a placeholder. For broadcast addresses, you also may use <BCast>.
- For the highest protocol used, also write down the message type/opcode that is used, e.g. ICMP **Redirect**.

Below is an **example** diagram, explaining the idea of the format that you shall use (Note: The example diagram makes no sense beyond explaining the *format*). Stop after the first ping answer has been received by Alice.



Solution:



- As Alice has statically configured her IP addressing information, she knows the IP address of her **default gateway**, which is the interior IP address of R.

- All traffic not belonging to the local subnet is commonly forwarded to this default gateway, so is the traffic for Linus. However, to start sending IP packets for Linus to R, she requires Layer 2 information about R (its MAC address).

- Alice therefore **sends an ARP request** to ask who has the default gateway's IP address, and gets an ARP response from R, containing R's MAC address.

- Alice now knows the MAC address of its default gateway, which is required for further IP communication.

- To communicate with Linus, Alice now sends a packet with Linus's IP address in the IP destination field, **but with R's MAC address** in the Ethernet destination field.

- Returning packets from Linus also always contain R's MAC address in the Ethernet source field, but Linus's IP address in the IP source field.

- *Ping* sends out ICMP Echo request packets. Linus therefore answers with an ICMP Echo Reply.

Problem 1.2 - Routing and Basics

Note: If you have fundamental problems in answering these or other basic networking questions from this exercise, we highly recommend to prepare yourself for the later lectures and exercises by refreshing your knowledge in the relevant topics, e.g., using the book by Andrew S. Tanenbaum on "Computer Networks".

a) The lecture briefly summarized a variant of the protocol stack used to describe the functionality required to provide end-to-end communication in computer networks. The stack is organized into layers, where alternative implementations of the same abstract functionality can be realized by alternative mechanisms and protocols. These alternatives can be functionally equivalent or expose specific sets of functions to layers higher up in the stack.
Please name the basic functions that IP(v4), TCP, and UDP provide to higher layers.

Solution:
Note: The following answer is longer than what was expected from you.

IPv4 (Source: `https://tools.ietf.org/html/rfc791`):
"The internet protocol provides for transmitting blocks of data called datagrams from sources to destinations, where sources and destinations are hosts identified by fixed length addresses."

- Addressing and routing: "The function or purpose of [the] Internet Protocol is to move datagrams through an interconnected set of networks. This is done by passing the datagrams from one internet module to another until the destination is reached. The internet modules reside in hosts and gateways in the internet system. The datagrams are routed from one internet module to another through individual networks based on the interpretation of an internet address. Thus, one important mechanism of the internet protocol is the internet address."

- Fragmentation ("and reassembly of long datagrams"): "In the routing of messages from one internet module to another, datagrams may need to traverse a network whose maximum packet size is smaller than the size of the datagram. To overcome this difficulty, a fragmentation mechanism is provided in the internet protocol."

TCP (Source: `https://tools.ietf.org/html/rfc793`):
"[..] the primary purpose of the TCP is to provide reliable, securable logical circuit or connection service between pairs of processes."
"Reliability: [..] The TCP must recover from data that is damaged, lost, duplicated, or delivered out of order by the internet communication system."
Note: This definition might be confusing as it uses a broad definition of reliability that covers reliability, order, and integrity of data deliveries.

- Reliability delivery, order, and integrity of packets

- Flow control: Alignment of sender and receiver connection/processing speed

- Connection semantics: Allow for logical connection between processes (not hosts)

UDP (Source: `https://tools.ietf.org/html/rfc768`): "This protocol provides a procedure for application programs to send messages to other programs with a minimum of protocol mechanism. The protocol is transaction oriented, and delivery and duplicate protection are not guaranteed. " Yet, it specifies mechanisms for:

- Packet integrity (checksum over header and data)
- Multiplexing: Use of ports allows for multiplexing of packets among multiple processes

b) Briefly explain the difference between **routing** and **forwarding**.

Solution:

- *Routing* (algorithm): A successive exchange of connectivity information between routers. Each router builds its own routing table based on collected information
- *Forwarding* (process): A switch- or router-local process which forwards packets towards the destination using the information given in the local routing table.

c) Name two examples (each) of **link state** and **distance vector** routing algorithms. State the abbreviation as well as the full names.

Solution:
Link state:

- OSPF (Open Shortest Path First)
- IS-IS (Intermediate System to Intermediate System)

Distance vector:

- RIP (Routing Information Protocol)
- IGRP (Interior Gateway Routing Protocol)

d) Briefly name and explain key differences between **link state** and **distance vector** routing approaches.

Solution:
Example answer: "Link-state algorithms (also known as shortest path first algorithms) flood routing information to all nodes in the internetwork. Each router, however, sends only the portion of the routing table that describes the state of its own links. In link-state algorithms, each router builds a picture of the entire network in its routing tables. Distance vector algorithms (also known as Bellman-Ford algorithms) call for each router to send all or some portion of its routing table, but only to its neighbors. In essence, link-state algorithms send small updates everywhere, while distance vector algorithms send larger updates only to neighboring routers. Distance vector algorithms know only about their neighbors."

Source: `http://docwiki.cisco.com/wiki/Routing_Basics`

e) What is the name of the most widely used routing protocol that exchanges routes **between** autonomous systems? Do not abbreviate the name.

Solution: Border Gateway Protocol (Exterior Gateway Protocol would be okay, too.)

f) MPLS - Multiprotocol Label Switching

Although not explicitly in the focus of this lecture, MPLS is mentioned and used throughout the course on various occasions. As a commonly applied mechanism, you are required to be aware of its purpose and high-level principals. Make yourself familiar with the core concepts of MPLS and briefly answer the following questions.

Hint: You may use the main RFC of MPLS as starting point `https://tools.ietf.org/html/rfc3031` or, e.g., the book by Andrew S. Tanenbaum on "Computer Networks".

I What is an FEC in the context of MPLS?

Solution:
FEC stands for "Forwarding Equivalence Class". The idea of FECs is to enable the grouping of individual network flows into classes and defining the forwarding paths as well as the service classes based on these classes. It is possible that each flow is mapped to its own FEC, yet, large network providers would usually target grouping most flows into a small number of FECs and only handle specific flows separately.

II What is the technical motivation of using MPLS inside a network domain instead of normal routing based on the destination IP address?

Solution:
It allows for a simpler routing of packets through the network. Simpler in the sense that packets belonging to the same FEC can be consistently forwarded through the network and the routing process inside the network does not require the maintenance of large routing tables and perform IP address matching.
The mapping to an FEC happens once when a packet enters an MPLS network, while intermediate routers can take forwarding decisions solely based on labels included as part of the packet's MPLS header. This makes the otherwise more complex destination address-based routing of IP packets simpler in that (potentially large) routing tables need to be only maintained at the edge of the network and that only lookups in (smaller) MPLS forwarding tables are required inside the network. Additionally, this allows to provide and enforce quality-of-service guarantees for individual FECs throughout the whole network. For these features, MPLS comes with a number of additional mechanisms and protocols to, e.g., establish forwarding paths and reserve resources for them.

III To which layer of the IP protocol stack does it belong?

Solution:
It is independent of both the network as well as the data link layer. As it introduces an additional header between layer 2 and 3, where it encapsulates network layer/IP packets, it is often said to belong to layer 2.5 (in-between the two others).

Problem 1.3 - SDN and OpenFlow Basics

a) What steps does SDN take to improve the programmability of networks?

Solution:
It resolves the tightly vertical integration of network devices and introduces the concept of a *Network Operating System (NOS)*. The NOS exposes a well-defined API to implement specific features upon. As you will learn in the course of the lecture, these features are also commonly termed *control applications*. The NOS itself is a (distributed) system that provides a consistent and up-to-date network view to the control applications. It uses an open and vendor-agnostic protocol to access state information from the (hardware) network forwarding elements and control the forwarding behavior of these elements.

b) In the lecture, first examples for functionality that can be realized using OpenFlow flow table entries are shown. One example describes a simplified firewall behavior (see slide 38 of lecture 1).
Briefly describe the behavior of an OpenFlow switch that solely uses this entry. Assume that afterwards, packets are passed to the switch's normal Ethernet switching pipeline.

Solution:
Incoming packets are filtered and TCP packets with destination port 22 are dropped. All other packets are passed on to the normal Ethernet switching pipeline.

c) Other rules could be specified based on other header fields or their combinations to realize a firewall that completely runs on an OpenFlow switch, without involving the network controller in the packet processing. Which types of firewall rules cannot be realized using the set of functionality introduced so far?

Solution:

- Stateful firewall rules cannot be specified. Flow table entries are stateless (besides the statistics).
- Besides, firewall rules that concern layers that cannot be matched using OpenFlow cannot realized (application-layer firewalls).