A Lab 1 – ARP protocol

In this laboratory you should become familiar with the ARP protocol and in the end fully understand it.

Objectives

1. Examine the ARP table on your local computer.
2. Make a pair with your colleague and try to capture and analyze the ARP communication between your computers in Wireshark.
3. Delete the record obtained in objective 2. Add the addresses (IP and MAC) of your colleague to the ARP table using the static method and then examine the communication in Wireshark.
4. Display the graph of captured packets in Wireshark.
5. Create the reference topology in Packet Tracer (see Fig. A.1).
6. Generate and examine the ARP communication in Packet Tracer Simulation mode. Explore the switch's MAC table.

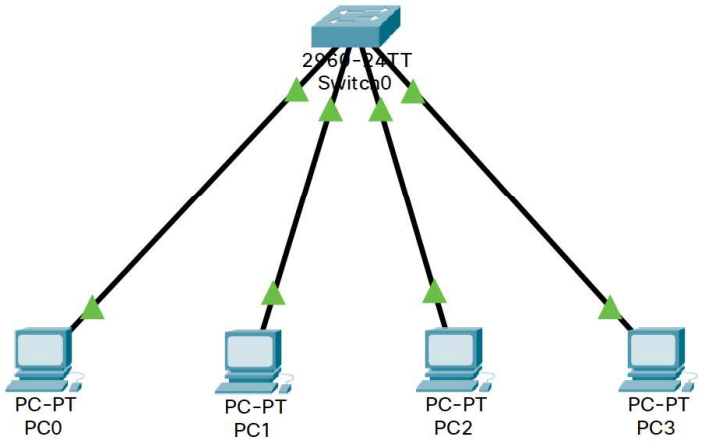

Fig. A.1: Lab 1 reference topology.

A.1 Introduction

ARP (Address Resolution Protocol), defined in RFC 826 [1], is used in computer networks to map the logical IPv4 address (32-bit address used for addressing on the ISO/OSI network layer) to the MAC address (48-bit physical address used for addressing on the ISO/OSI link layer). It operates between the OSI network and link layers. When the communication starts, an IP packet is created by adding the IP header, which consists of source and destination IP address (and many other items), to the data. But when the device wants to communicate, it must know the MAC address (on the Ethernet networks) of the destination device as well. There are 2 scenarios:

1. The device wants to communicate with another device on the local network.
2. The device wants to communicate with a device outside the local network.

In both cases, the device initiating the communication needs to know the MAC address of the destination. In the first scenario, the desired MAC address belongs to the **destination device** on the local network. In the second scenario, the destination MAC address must be set to the physical address of the **default gateway** (commonly router).

ARP packets are divided into 2 groups: *ARP request* and *ARP response*. ARP request is used by the device initiating the communication to resolve the IPv4 address of the destination device to its physical (MAC) address. You can imagine the request as: "*Hey, I am host A and I want to communicate with Host B with this IP address. What is your MAC address?*"ARP response is sent as the response to the ARP request, where the device (which the request was sent to) sends its physical address. You can imagine the response as: "*Hey Host A, I am Host B, I recognized my IP address and here is my MAC address.*"ARP packets are encapsulated in the Ethernet frames (see Fig. A.2).

The records (IP-MAC bindings) are stored in the **ARP table** (cache memory). The table basically consist of:

- **Internet Address** – The logical (IP) address of the destination device.
- **Physical Address** – The MAC address of the destination device.
- **Type** – The way the record was inserted into the table. There are 2 ways: *dynamic* and *static*. Dynamic record is a record learnt by the ARP process (ARP request and reply). Static record is a record manually inserted by the user.

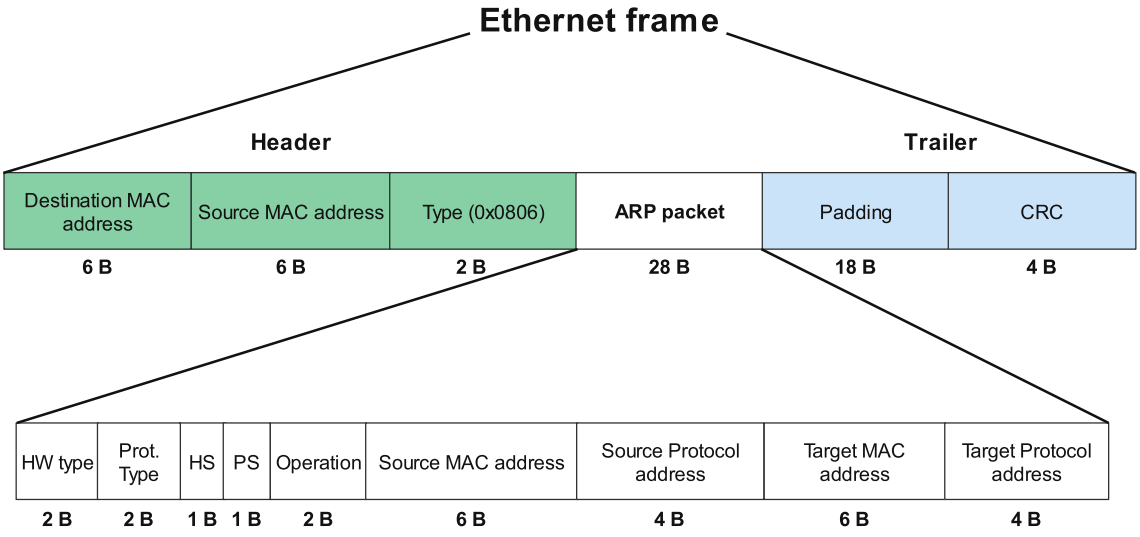

Fig. A.2: ARP packet encapsulated inside the Ethernet frame.

Principle of communication

1. The source device (Host A) wants to communicate with the destination device (Host B). Host A searches its ARP table to find the IP address of the Host B that is bound with its MAC (Media Access Control) address. If the record is found, Host A sends a message directly to the Host B.
2. If the record is not present, the ARP request is generated. Host A sets its own MAC address as the Source MAC address in both the Ethernet frame and the ARP packet (which is encapsulated inside the Ethernet frame) and sets its own IP address as the Source Protocol address. Ethernet Destination MAC address is set to Broadcast, Target MAC address is left blank (default value). Target Protocol Address is set to the Host B's IP address.
3. As the Destination MAC address is set to broadcast, all the devices on the local network receive and process the frame (see Fig. A.3). Each device de-encapsulates the ARP packet and compares the Target Protocol Address with its own IP (Internet Protocol) address. Only Host B finds a match. Every other device will drop the frame.
4. Host B adds a record to its own ARP table with the addresses of the Host A (if it is not already present) and generates ARP response. It sets original source addresses as the destination addresses and sets its own addresses (both logical and physical) as the source addresses.
5. ARP response is now sent as unicast directly to the Host A (see Fig. A.4).
6. Host A receives the ARP response, fills its own ARP table with the received addresses (Source MAC and Source Protocol) and sends the original message directly to the Host B.

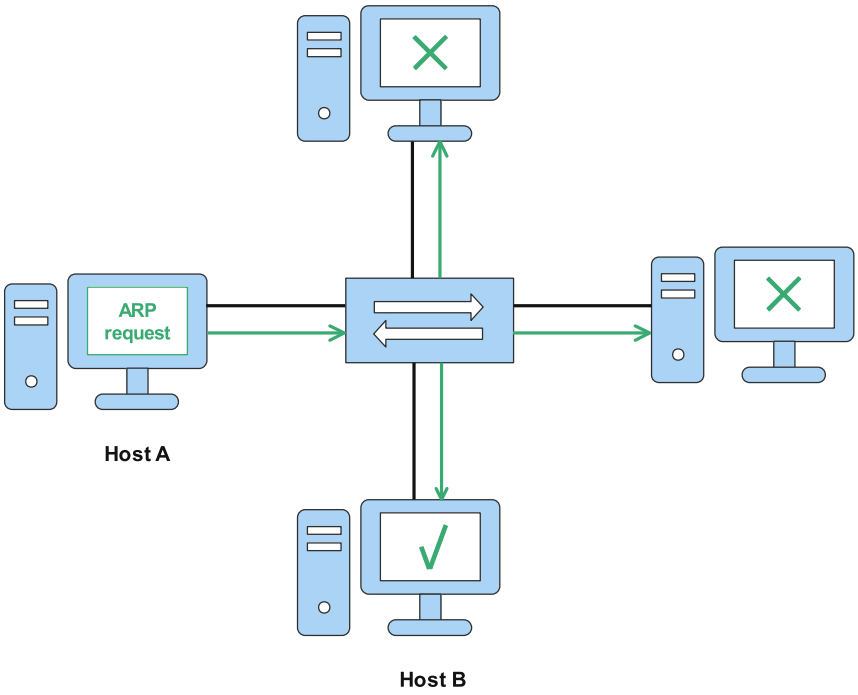

Fig. A.3: ARP request.

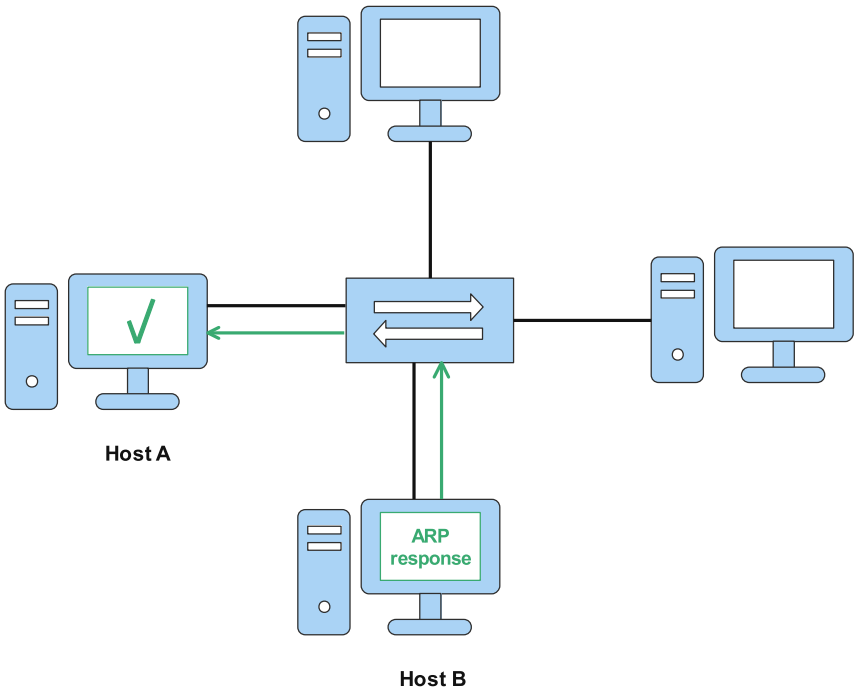

Fig. A.4: ARP response.

A.2 Wireshark

Wireshark is an open source network packet analyzer where you can capture and analyze live traffic. At the beginning, you choose the interface where you want to capture the traffic and then all the captured packets are visible. The main window is divided to the following sections (see Fig. A.5):

1. The **Menu** and **Main toolbar** are used to do some specific actions based on the chosen tool. Commonly used items are the *Start capturing packets* (symbol of blue fin under the File) and *Stop capturing packets* (the symbol of square beside the fin).
2. The **Filter Toolbar** allows users to set filter on the specific protocol, source/destination address, port etc. and display only the desired packets.
3. The **Packet List Pane** displays all the captured packets. It is further divided to the following sections by default:
   - *No.* – The number of a packet in order it was captured.
   - *Time* – The timestamp of the packet.
   - *Source* – The source IP address.
   - *Destination* – The destination IP address.
   - *Protocol* – The protocol name abbreviation.
   - *Length* – The length of a packet.
   - *Info* – Information about the packet content.
4. The **Packet Details Pane** displays the details about selected packet (chosen in the **Packet list pane**) in rows, ie. link layer protocol, IP protocol etc. You can also click on each row to display further details about each protocol (like information inside the header).
5. The **Packet Bytes Pane** displays the data inside the selected packet from the **Packet List pane** and highlights the bytes corresponding to the items selected in the **Packet Details Pane** [2].

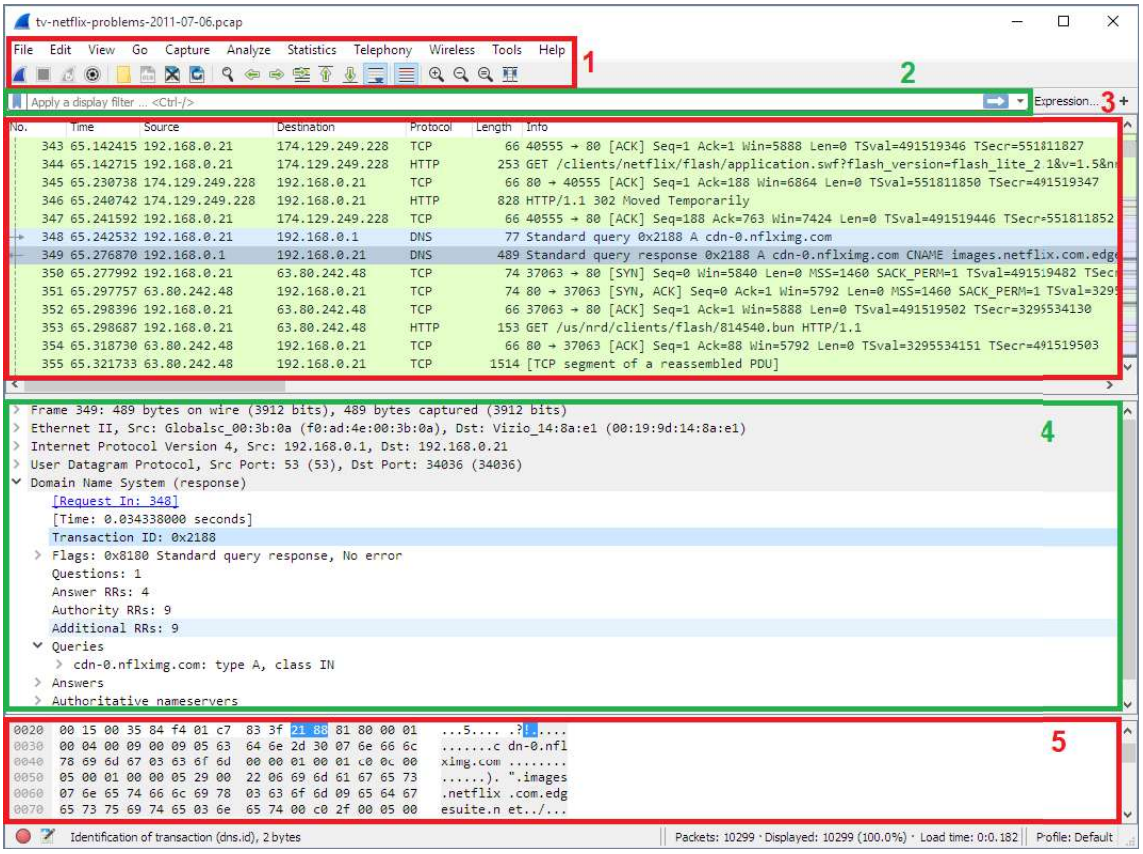

Fig. A.5: Wireshark Main window.

A.2.1 Objective 1

1. Open the **CMD** (Command Prompt) on your local computer.
2. Execute the `arp -a` command, which is used to display the contents of the ARP table.
3. You will see the records divided to the sections by interfaces, which have the IP address assigned. Explore what these interfaces are by issuing the `ipconfig` command. Here you get a list of interfaces with their name and IP address assigned.
4. As mentioned in the introduction, you can see that in the ARP table there are 3 columns: *Internet Address*, *Physical Address* and *Type*. Now focus on the Ethernet interface (the name should be something like *Ethernet adapter Ethernet* in the output of the `ipconfig` command) which is identified by its IP address. Among some static records, there should be at least one dynamic record – **the default gateway** (check the `ipconfig` output to verify the address). It usually uses the first host address of the subnet range. Its presence means that your computer has already communicated with the router (e.g., to obtain the dynamic IP address).

5. You can notice that there is one special static record with the *ff-ff-ff-ff-ff-ff* MAC address.
   What is this record used for?

A.2.2 Objective 2

1. For this objective, you should team up with your colleague next to you. Once you're in a team, you can continue to the next step.
2. If you have closed CMD from the previous objective, open it again and execute command `ipconfig /all`. This command gives you detailed information about the network configuration of your local interfaces.
3. Find the Ethernet interface (same as in the objective 1) and check the obtained data. You should be especially interested in the *Physical Address* and the *IPv4 Address*. Write the addresses somewhere for the future use.
4. Now get the addresses of your colleague and check your ARP table, if the record of your neighbor is not present in the table[1].
5. Let's open Wireshark. At first, the list of available interfaces, where you can capture the traffic, is displayed. Choose your Ethernet interface.
6. Once you have the interface selected, you can probably see dozens of packets being captured. Now you will apply the filter for the ARP protocol. In the **Filter Toolbar** (see Fig. A.5), type `arp` and press enter. If the background color of the Filter Toolbar changes to green, the filter is correctly applied.
7. Keep the Wireshark running and return to the CMD. Now **only one member** of the team will generate the ARP request to obtain the MAC address of the second member, but the traffic will be caught on both computers. This is for the ability to communicate over Ethernet. For this purpose, you will use the **ping**[2] utility.
8. In the CMD, execute `ping <IP>` where **<IP>** is the IP address of your colleague. Now return to the Wireshark. You should both see two ARP messages similar to the Fig. A.6. As mentioned in the introduction, ARP requests are sent to all the hosts on the local network, so you will see also requests from other colleagues. For this purpose, you can edit the filter to display only the ARP communication of your pair. Use the following command:
   `arp.src.proto_ipv4 == <IP> or arp.dst.proto_ipv4 == <IP>` where

[1]NOTE: If the record is present, you should delete it. For this purpose, close the CMD and open it with admin privileges. Once opened, use the command `arp -d <IP>` where **<IP>** is the IPv4 address of your colleague.

[2]Ping is a software utility used to test if the host is reachable over the IP network. It sends out *ICMP Echo request* message and awaits *ICMP Echo reply* message. For more information visit the [3].

<IP> represents your IPv4 address.

Fig. A.6: ARP request and reply captured in Wireshark.

9. In the Fig. A.6, you can see the ARP request is sent as a broadcast and info contains: "*Who has 192.168.0.4? Tell 192.168.0.1*". A host needs to be identified by its IP address first. If a match is found, the response is sent. The ARP response is sent as a unicast (directly to the source of ARP request) and info contains: "*192.168.0.4 is at 00:50:79:66:68:03*". As you can see, some host found a match with its own IP address and sent a response where he mentions his MAC address. The length of both packets is 64 bytes (which corresponds to the Fig. A.2). Now let's examine both packets more in depth.
10. **ARP request**
    Click on the ARP request packet (see Fig. A.7). There will be 3 lines displayed in the **Packet Details Pane**. We will be interested in the last 2 lines, i.e. Ethernet II and Address Resolution Protocol (see Fig. A.2). Now expand the Ethernet II line. There are following items we are interested in:
    - **Destination Address** – The destination MAC address identifying the destination of a frame.
    - **Source Address** – The source MAC address identifying the source host.
    - **Type** – The value expressing what protocol is encapsulated inside the frame.

    Determine your values of Destination, Source and Type. Next, expand the Address Resolution Protocol line. The following items are important:
    - **Opcode** – The value identifying the type of ARP message: *request* or *response*.
    - **Sender MAC address** – The MAC address of a host sending a request.
    - **Sender IP address** – The IP address of a host sending a request.
    - **Target MAC address** – The MAC address of a host the request is destined for.
      What is the value of this item?[3]
    - **Target IP address** – The IP address of a host the request is destined for.

    Compare the values with your own addresses and addresses of your colleague. *What is the destination MAC address and who will receive the frame?*

[3] You can probably see a different value than the value displayed in the Fig. A.7. Your field should contain only the zeroes. This is the default value when the destination address is not known. For more information you can visit the [4].

Why are the Destination MAC address and Target MAC address different?

```
Frame 41: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface -, id 0
Ethernet II, Src: Private_66:68:00 (00:50:79:66:68:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
› Destination: Broadcast (ff:ff:ff:ff:ff:ff)
› Source: Private_66:68:00 (00:50:79:66:68:00)
  Type: ARP (0x0806)
  Padding: 000000000000000000000000000000000000
  Frame check sequence: 0x00000000 [unverified]
  [FCS Status: Unverified]
Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: Private_66:68:00 (00:50:79:66:68:00)
  Sender IP address: 192.168.0.1
  Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)
  Target IP address: 192.168.0.4
```

Fig. A.7: ARP request detailed in Wireshark.

11. **ARP response**

    Now move from the ARP request to the ARP response packet (see Fig. A.8). The lines remain the same. Expand both Ethernet II and Address Resolution Protocol lines and compare your own values with the values from a request.

```
Frame 42: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface -, id 0
Ethernet II, Src: Private_66:68:03 (00:50:79:66:68:03), Dst: Private_66:68:00 (00:50:79:66:68:00)
› Destination: Private_66:68:00 (00:50:79:66:68:00)
› Source: Private_66:68:03 (00:50:79:66:68:03)
  Type: ARP (0x0806)
  Padding: 000000000000000000000000000000000000
  Frame check sequence: 0x00000000 [unverified]
  [FCS Status: Unverified]
Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: Private_66:68:03 (00:50:79:66:68:03)
  Sender IP address: 192.168.0.4
  Target MAC address: Private_66:68:00 (00:50:79:66:68:00)
  Target IP address: 192.168.0.1
```

Fig. A.8: ARP response detailed in Wireshark.

A.2.3 Objective 3

1. Open CMD and display the contents of the ARP table.
   Is the expected record present in the table?
   What is the type of a record?
2. Now you will both remove the obtained record. For this purpose you must run CMD with administrator privileges[4]. **Write somewhere both logical and physical addresses before deletion for the future use!** Execute command `arp -d <IP>` where **<IP>** represents the IPv4 address of your colleague[5]. In the end, both of you should have removed the records obtained from objective 2.
3. Now you are going to add the record manually. Use the command `arp -s <IP> <MAC>`. Both IP (**<IP>**) and MAC (**<MAC>**) addresses are the adresses of your colleague. For the MAC address use dash as the delimeter. Once issued, display the contents of the ARP table.
   What is the type of the newly added record?
4. Go back to the Wireshark. Check if you are still capturing packets (fin is grey, square shines red) and the filter from objective 2 is still active. If not, apply it again.
5. Use `ping` command in the CMD to test the availability of your colleague.
   Were any new ARP packets captured? Why?
   Static records are pretty rarely used but can be useful when communicating with a device, whose address does not change throughout long period communication. Dynamic records have their timeout, mostly 2 minutes. If the record is not used for communication during this period, it is automatically discarded. When used, timeout is increased by another 2 minutes. Maximum time the record can be valid is 10 minutes, then it is discarded. This is different for static records. There is no timeout for these, they remain valid until manually deleted or system reset (ARP table is stored in cache memory which is cleared with reset).

[4]NOTE: Write `cmd` to the Windows search bar, then right click on it and press "Run as administrator".

[5]NOTE: To clear the whole ARP table you would execute `arp -d *`.

A.2.4 Objective 4

1. In a Wireshark, stop capturing packets (by clicking on the red square).
2. Wireshark gives you the possibility to display captured packets in graph. You can achieve this in **Statistics > I/O Graphs**.
3. New window appears. You can probably see more than 1 graph. If not, there's nothing wrong. One of the displayed graphs should display *All Packets*, which represents all the captured packets on your interface without regard to protocol types. Below the displayed graphs, there is a section with settings for each graph. You will be interested in the following values:
   - **Enabled** – If checked, the graph is displayed.
   - **Graph Name** – The name of a graph.
   - **Display Filter** – You can limit the graph only to the certain packets based on protocol, IP address etc.
   - **Y Axis** – As X axis represents the time, you can choose what the Y axis displays. You can select *Packets*, *Bytes*, *Bits* matching the filter per time interval and others.

   Below the settings section, there are some additional items. Some of them are:
   - [+] – Add a new graph.
   - [−] – Remove a graph.
   - **Interval** – The interval period for the graph.
4. Now remove all the graphs by selecting them and clicking on the [−].
5. Add a new graph by clicking on the [+]. Tick **Enabled**, change the **Graph Name** to "arp"and set the **Display Filter** to `arp`. You are free to change the **Color** to the color you want. Now select the **Bytes** value in the **Y Axis** column. Set **Interval** to **1 sec**[6].
6. In the Fig. A.10, you can see the output with the previous settings[7]. The input data are displayed in the Fig. A.9. Bytes are exported to the Tab. A.1 and packets to the Tab. A.2. The first deflection represents PCs checking for the duplicate address when statically assigned (this is called *gratuitous ARP*, you don't have to see these necessarily as you have probably dynamic address assigned from the DHCP server). There should be only one message for the correct assignment and that is ARP request. No response should be received. The last 3 peaks represent ARP request and reply communication each. You should see similar peak. If you zoom in (using the mouse wheel), you can see that peak has 128 bytes.

[6]NOTE: If your graph is too much elongated, use the **Reset** button to make recalculation.

[7]For better readability, the following graphs are generated using Matlab. Most of the aspects of the Wireshark's graphs are preserved, including axis names, scale and the function values. Functions are shifted in time so they start at zero time.

Why is it 128 bytes?

7. Now change the **Y Axis** to **Packets** (see Fig. A.11).
   How many packets are transferred during 1 peak?

No.	Time	Source	Destination	Protocol	Length	Info
5	35.819496	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.1 (Request)
6	36.832330	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.1 (Request)
7	37.358683	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.2 (Request)
8	37.843343	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.1 (Request)
9	38.359146	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.2 (Request)
10	38.858842	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.3 (Request)
11	39.358935	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.2 (Request)
12	39.873878	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.3 (Request)
13	40.405112	Private_66:68:03	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.4 (Request)
14	40.888502	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.3 (Request)
15	41.416994	Private_66:68:03	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.4 (Request)
16	42.426050	Private_66:68:03	Broadcast	ARP	64	Gratuitous ARP for 192.168.0.4 (Request)
17	49.137627	Private_66:68:00	Broadcast	ARP	64	Who has 192.168.0.2? Tell 192.168.0.1
18	49.137627	Private_66:68:01	Private_66:68:00	ARP	64	192.168.0.2 is at 00:50:79:66:68:01
29	55.993312	Private_66:68:00	Broadcast	ARP	64	Who has 192.168.0.3? Tell 192.168.0.1
30	55.993312	Private_66:68:02	Private_66:68:00	ARP	64	192.168.0.3 is at 00:50:79:66:68:02
41	63.153478	Private_66:68:00	Broadcast	ARP	64	Who has 192.168.0.4? Tell 192.168.0.1
42	63.153478	Private_66:68:03	Private_66:68:00	ARP	64	192.168.0.4 is at 00:50:79:66:68:03

Fig. A.9: Input data for the I/O graph.

Tab. A.1: Table of bytes sent during the ARP communication.

Seconds	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Bytes	64	128	128	128	128	128	64	0	0	0	0	0	0	128
Seconds	14	15	16	17	18	19	20	21	22	23	24	25	26	27
Bytes	0	0	0	0	0	0	128	0	0	0	0	0	0	128

Tab. A.2: Table of packets sent during the ARP communication.

Seconds	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Packets	1	2	2	2	2	2	1	0	0	0	0	0	0	2
Seconds	14	15	16	17	18	19	20	21	22	23	24	25	26	27
Packets	0	0	0	0	0	0	2	0	0	0	0	0	0	2

A.3 Packet Tracer

Packet Tracer is a proprietary software developed by the Cisco company used for simulating the networks. Cisco is one of the most well known companies in the network engineering. The software is primarily intended for the CCNA (Cisco Certified Network Associate) academy, but upon registration everyone can get it for free with

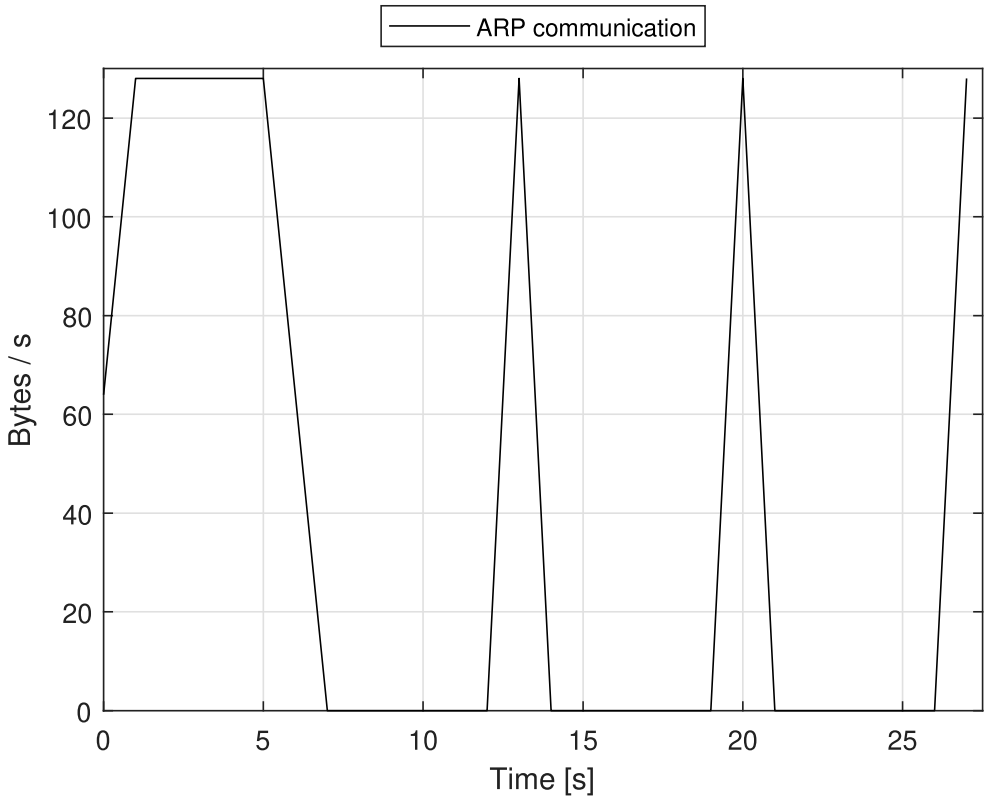

Fig. A.10: Bytes sent during ARP communication.

the basic course named *Getting Started with Cisco Packet Tracer*. You are able to work here with the **Cisco** devices only.

Device configuration

The devices in Packet Tracer can be configured via the **CLI**, **Config** and **Desktop**.

- **CLI** - CLI (Command Line Interface) is the most common way of configuring intermediary devices including switches, routers etc. You must have some knowledge of commands, but when you get used to it, this can be the fastest way of configuring devices. You can use **question mark** (**?**) in each mode to display a comprehensive list of commands. You can also use question mark while typing the keywords to get list of available keywords matching the starting string or after typing them to get the list of arguments and optional parameters.
- **Config** - This is the fast way of **basic** configuration. This allows you to set device name, IP addresses on interfaces etc. The extended configuration still needs to be performed using the CLI. Wireless (home) routers are the exception, because they don't have CLI and all the configuration is performed using

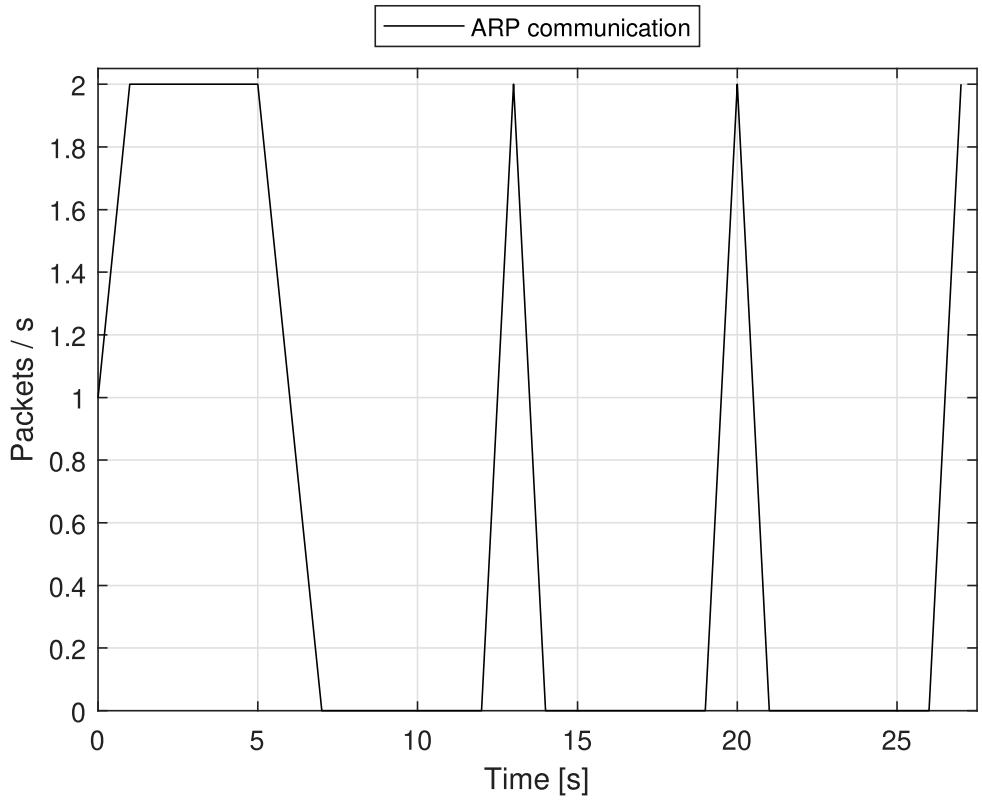

Fig. A.11: Packets sent during ARP communication.

the GUI (Graphic User Interface).

- **Desktop** - This is the way of configuring end devices like PCs and servers. You find here many icons representing user network routine. The examples are IP Configuration, Terminal, Web Browser and Email.

Cisco CLI hierarchy

There are 3 basic modes in which you operate while working with the Cisco CLI [5]:

- **User EXEC mode** – Only basic network monitoring commands are available. Mode is identified by the `>` prompt:
  `Switch>`
- **Privileged EXEC mode** – This mode should be always password protected as you can get to all the commands (including displaying running configuration, routing table etc.) and to all the modes. Mode is identified by the `#` prompt:
  `Switch#`
- **Configuration mode** - You get to the configuration mode from the **Privileged EXEC mode**. Configuration mode is not password protected as you

access it from the most authorized mode. You write all the configuration commands here concerning the global router settings or enter the interface configuration mode (to set individual interfaces) and other modes. Configuration mode is identified by the `(config)` keyword.

```
Switch(config)#
```

Packet Tracer modes

There are 2 modes in which you can operate in Packet Tracer: **Realtime** and **Simulation.**

1. **Realtime** – In the Realtime mode, the communication occurs as it would be in the real word. This means that after using network utilities (like ping, traceroute etc.) or visiting the web sites you see the output immediately.
2. **Simulation** – In the Simulation mode, individual packets are being tracked. You can examine the content of packets while they move from 1 device to another.

Environment description

Packet Tracer is divided to the following sections (see Fig. A.12):

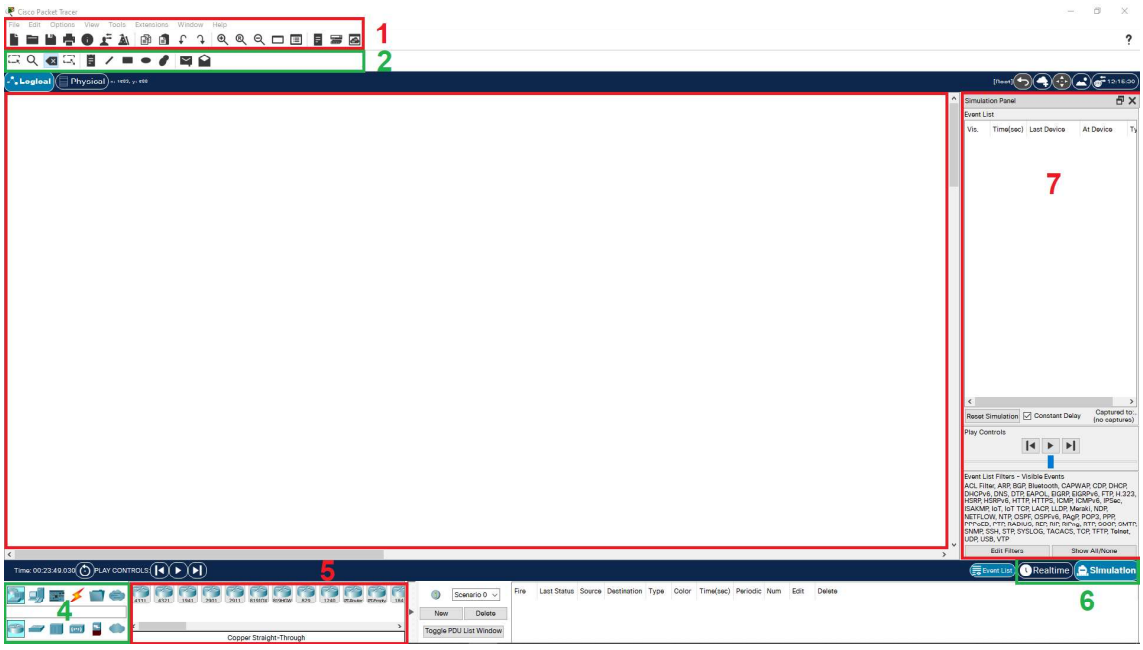

Fig. A.12: Packet Tracer environment.

1. The **Menu** and **Main toolbar** are used to do some basic actions like saving and opening the file, setting the global software preferences etc.

2. This **toolbar** is used for the Workspace in which you can:
   - **Select** more devices.
   - **Inspect** devices tables (routing table, ARP table etc.).
   - **Delete** individual elements.

   Then you can draw shapes to differentiate network areas.
3. In the **Workspace** the topologies are created.
4. **Network components** contain all the available elements (devices, cables...) in Packet Tracer. There are 2 rows [6]:
   - Upper row contains device groups based on their type.
   - Bottom row contains subgroups of the selected device group. For example the **Network Devices** group contains **Routers**, **Switches**, **Hubs** subgroups etc.
5. **Device-specific selection box** contains specific individual devices available in the selected group/subgroup.
6. **Realtime/Simulation tabs** allow you to switch the mode in which you want to simulate.
7. If the **Simulation** mode is selected, new window **Simulation Panel** appears. It contains the following sections:
   - **Event List** – The packet flow appears here.
   - **Reset Simulation** – Clears all the current packet flow (displayed in the Event List)
   - **Play Controls** – Here you can manually click on the **Go Back to Previous Event** and **Capture then forward** buttons to move simulation one step back or forward (step is meant moving packet from one device to another). You can also use **Play** button to run the simulation automatically without the need for manual clicking. The speed of simulation is controlled by the slider below.
   - **Event List Filters – Visible Events** – Displays protocols which are being traced. In the **Edit Filters** you can manually select which protocols (packets) you want to trace. By clicking on the **Show All/None** button you select all the protocols to be traced or none.

A.3.1 Objective 5

1. Open the Packet Tracer program.
2. In the **Network components** section, select the **Network Devices** group. Then select the **Switches** subgroup and in the **Device-specific selection box** select the switch named **2960**. You just click on it and then click to the **Workplace** where you want it to be placed or just follow the principle **Drag and drop**, where you drag the switch to the Workspace and then drop it.
3. Now let's add the PCs. Select the **End Devices** group and add **4 PCs** to the Workspace.
4. Now you will connect the devices. Select the **Connections** group and choose **Copper Straight-Through** cable (the solid black line icon). Notice your cursor changes. Click on the first PC and choose the **FastEthernet0** interface. Then click on the switch. Notice it has 24 Fast Ethernet ports and 2 Gigabit Ethernet ports. Select FastEthernet0/1[8]. Connect the rest 3 PCs the same way as the first computer. Use consecutive port numbers. Wait for all the lights to shine green and continue to the next step.
5. Save your current progress by clicking on the **File > Save As ...** and choose the appropriate name for the lab.

A.3.2 Objective 6

1. Once you have the topology created, you will generate the communication. But first you must configure device addresses. Click on the first PC, select **Desktop** tab and open **IP Configuration** window. Here you can set IPv4 and IPv6 configuration. We will use IPv4 address space **192.168.1.0/24** which means you have 254 available addresses for hosts (1 is reserved as the network address and 1 as the broadcast address). Set the host **IPv4 Address** and **Subnet Mask** to the first address within the host range (subnet mask should be filled in automatically based on the classful addressing)[9]. Configure the rest of the computers the same way. Use the consecutive addresses (see Tab. A.3).
2. Save your current progress by clicking on the **File > Save**.

[8]NOTE: After you select the interface, the devices are interconnected. You can notice that there is a green triangle by the PC and orange triangle by the switch. Green color means everything is working. Orange color means that something is happening in the background before putting it to the working or other state. In this case, orange color represents STP process running. STP is beyond the scope of this lab.

[9]NOTE: As you will be communicating only within the local network, you don't have to specify the **default gateway**. But it is necessary to specify it when communicating with other than local networks.

3. Open the first PC, select **Desktop** tab and open the **Command Prompt** window. Issue the `ipconfig /all` command (press Spacebar key until the prompt appears again). Notice the MAC address is now in the different format than on your local PC. Create a table similar to the Tab. A.3 and write the IPv4 and MAC addresses of all PCs to it.
   Be careful to write the Ethernet MAC address and not the Bluetooth MAC address!

Tab. A.3: Table of PC addresses.

PC	IPv4 Address	MAC address
PC0	192.168.1.1	0001.C9BC.63CC
PC1	192.168.1.2	000A.F323.6BDC
PC2	192.168.1.3	0001.9781.2486
PC3	192.168.1.4	0030.F2D1.A4E1

4. Switch to the **Simulation** mode. Use **Show All/None** to clear the filter list. Click on the **Edit Filters** button and tick **ARP** int the **IPv4** tab. Close the filter window.
5. You are going to generate ARP communication between PC0 and PC2. But before it, check if the ARP tables are empty on both computers by issuing the `arp -a` command in the Command Prompt.
6. On the PC0 issue `ping 192.168.1.3`. Notice that the ARP message was generated in the **Event List**. The following items are available:
   - **Time (sec)** – Timestamp of a packet.
   - **Last Device** – Name of the device the packet came from.
   - **At Device** – Name of the device where the packet currently is.
   - **Type** - Packet protocol.

   You can display the contents of the packet by clicking on the row in the Event List. The **OSI Model** is displayed. Notice that only 2 lowest layers contain data. You can see source and destination addresses of the Ethernet frame and encapsulated ARP packet inside the L2. Under the OSI Model, there is a description what currently happens on the device. You can also click on the **Outbound PDU Details** to display the contents of the packet.
   What is the source and target IP address?
   What is the source and target MAC address?
   What is the destination MAC address?
   What is the Opcode value?

7. Before continuing further, you are going to examine the **MAC address table** of the switch. Click on the switch end select **CLI** tab. If no prompt appears, click on the screen and press enter. You are now in the **User mode** identified by the

```
Switch>
```

prompt. Enter the `enable` command to enter the privileged EXEC mode. Now issue the `show mac-address-table` command to display the contents of the MAC table. You can notice that all the PCs are already present as seen in the Tab. A.4. Can you guess why? Table contains the following values:

- **Vlan** – Vlan the port is assigned to.
- **Mac Address** – MAC address of the device connected to the port.
- **Type** – Type of the record (static or dynamic).
- **Ports** – Port the frame was received on.

Tab. A.4: MAC table.

Vlan	Mac Address	Type	Ports
1	0001.9781.2486	DYNAMIC	Fa0/3
1	0001.c9bc.63cc	DYNAMIC	Fa0/1
1	000a.f323.6bdc	DYNAMIC	Fa0/2
1	0030.f2d1.a4e1	DYNAMIC	Fa0/4

Switch builds a MAC table to build the topology (map connected devices) so it can forward traffic directly to the host based on the destination MAC address. When the packet arrives at the switch, it checks its MAC table if the inbound port is already mapped to the source MAC address. If not, the record is created. Next the switch checks the destination MAC address. If the record of the destination MAC address and outbound port is present in the table, frame is forwarded **only** to this port. If not, the frame is flooded out the **all** ports (including those present in the MAC table) **except the inbound port**. The same process happens if the destination MAC address is set to **broadcast**. While ARP table stores records of the logical addresses mapped to the physical addresses, MAC table stores records of the ports mapped to the physical addresses. To see the process of filling the MAC table, clear the current content by issuing the `clear mac-address-table`.

8. Click **Capture then forward**. Now you can see that the frame arrived at the switch. Examine the MAC table.
   What is the current content?

Now explore the ARP packet (in the Event List). Notice that the **Inbound PDU Details** tab appeared. This is because some devices (eg. router) change the MAC address of the frame. If you compare the Inbound and Outbound PDU Details, they are the same in this case.

9. Click **Capture then forward**. The switch floods the frame out all the ports except the inbound port. As you can see, all the PCs receive the frame, but only one accepts it (see Fig. A.13) since it recognizes its IP address in the "**Target IP Address**"field.

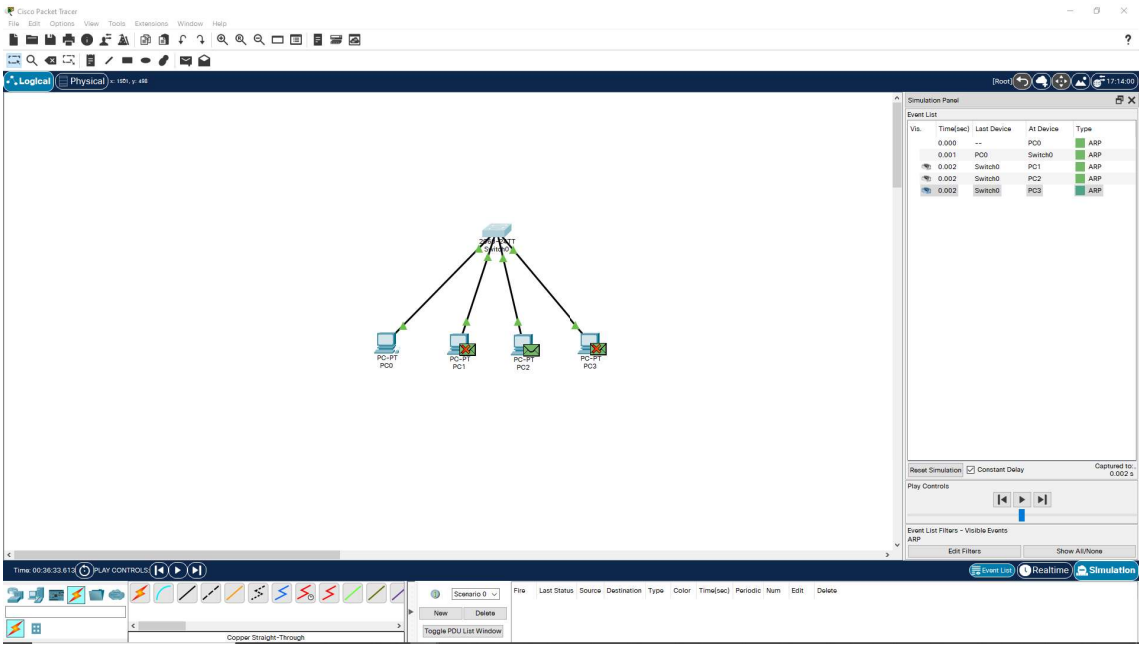

Fig. A.13: Switch sending the broadcast frame.

10. Click **Capture then forward**. You can see that PC2 sent packet to the switch. Examine PC2's ARP table.
    Does it contain some record? What device does the address combination correspond to?
    Now examine the switch MAC table and check if the content changed. Then display the ARP packet. Check the **Inbound PDU Details** as it has the content generated by the PC2.
    What is the source and target IP address?
    What is the source and target MAC address?
    What is the destination MAC address?
    What is the Opcode value?
11. Click **Capture then forward**.
    Where did the switch send packet? Did the switch flood packet the same way as the ARP request?

12. Click **Capture then forward** again. Return to the Command Prompt of PC0. You should see that the ICMP replies were successfully received[10]. Display the contents of the ARP table.
    Does it contain some record? What device does the address combination correspond to?
13. Save this topology for future use. You are going to continue with the files from previous lab in each lab.

A.4 Final questions

1. What is the destination MAC address for the ARP request?
2. What is the Opcode (Operation) value for the ARP request and response?
3. What values does the ARP table contain?
4. What is the size of ARP packet?
5. What is the difference between static and dynamic records in ARP table?
6. At which OSI layers does the ARP operate?
7. What is the difference between ARP and MAC table?
8. What does the switch do with a packet whose destination address is not contained in the MAC table?

[10] You did not see any ICMP packets as the filter is set to ARP only.