Arūnas ŠALTIS

TELECOMMUNICATION TECHNOLOGIES I, II

Project No
VP1-2.2-ŠMM-07-K-01-047

The Essential Renewal of Undergraduates Study Programs of VGTU Electronics Faculty

Vilnius "Technika" 2012

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

Arūnas ŠALTIS

TELECOMMUNICATION TECHNOLOGIES I, II

A Laboratory Manual

Vilnius "Technika" 2012

A. Šaltis. Telecommunication Technologies 1, 2: A Laboratory Manual. Vilnius: Technika, 2012. 127 p. [4,65 author's sheets, 2012 06 04].

Telecommunication Technologies 1, 2: A Laboratory Manual is intended for students who have chosen Telecommunication Engineering as their major at Vilnius Gediminas Technical University. In this book they will find the descriptions of the main Laboratory exercises, taught during Telecommunication Technologies 1, 2, which analyze and examine the local networks protocols, such as STP, VLAN, WLAN, as well as wide area networks protocols, for example, IP, ARP. Each Laboratory exercise consists of methodology to prepare for the exercise, the description of the exercise itself and questions, which allow having more insight into the studied area.

The publication has been recommended by the Study Committee of VGTU Electronics Faculty.

Reviewed by:
Doc Dr Lukas Pavilanskas, VGTU Department of Telecommunication Engineering,
Dr Antanas Vindašius, VGTU Department of Telecommunication Engineering

This publication has been produced with the financial assistance of Europe Social Fund and VGTU (Project No VP1-2.2-ŠMM-07-K-01-047). The book is a part of the project "The Essential Renewal of Undergraduates Study Programs of VGTU Electronics Faculty".

This is an educational methodology book, No 1343-S, issued by VGTU Press TECHNIKA *http://leidykla.vgtu.lt*

Language editor *Dalia Blažinskaitė*
Typesetter *Laura Petrauskienė*

eISBN 978-609-457-154-1
doi:10.3846/1343-S

CONTENTS

PREFACE

Telecommunications network is one of the most important components in shaping the information systems that we use in our everyday life. Mobile networks and the Internet are the things that a man living in a modern pace cannot do without. It is therefore very important to prepare young professionals who would be fully aware of modern telecommunications.

Telecommunications in a common understanding could be explained as a network, which services we use. But in fact, telecommunications network is very complex; it consists of many different systems and subsystems, protocols and interfaces. Therefore, while teaching the students about it, it is important not only to explain how networks operate, it is even more important to provide the essential knowledge how each, even the smallest, component of the network works, and most importantly, what it does and why it is so important to the network. These tasks are usually dealt with during telecommunications technology disciplines.

This book contains descriptions of laboratory works for VGTU students, who study telecommunications technology disciplines 1 and 2. These disciplines are the cornerstone of the telecommunications teaching. Therefore, the laboratory works as an educational tool is widely used during this subject.

The real number of the laboratory works for these disciplines is greater than shown in the book, as it is simply not possible to contain everything into the limited amount of pages. Therefore this book attempts to include descriptions of those laboratory works, which are most important in everyday engineering activities, and could be used as a textbook or manual.

All laboratory works that are described in the book should be carried out with real devices, the ones, which young professionals may use in their professional activities. Those students, who work with real equipment, not only get acquainted with the operation

principles of the technology he or she examines, but also gain experience in configuring devices.

Note to students

Since the equipment (personal computers with Linux, Cisco company switches, routers and WLAN Access Point, Switch and the HP Company) used in the lab is real, below you may find a preliminary introduction, on how to work with this equipment, and an explanation about the text styles used in the book.

The configuration of all the equipment, except for an Access Point, is performed by using a command line interface (CLI). These command lines in the book are identified by a special font. The text, which is not in bold, indicates the name of network node and the CLI mode (only for Cisco and HP equipment). Bold text indicates what command a student must enter the using the keyboard. For example:

```
vlan-r1(config)#line con 0
```

Where: `vlan-r1` – node name, (`config`) – CLI mode, **`line con 0`** – the command and its arguments are being entered.

Hence, these laboratory exercises require to use a lot of typing, it is therefore important while configuring the software to know those features of the CLI, which would mitigate the work.

Firstly, it is not necessary to enter the whole command, it is enough only to type in the first few letters of the command and press the Tab key. If the beginning of the written command is enough to identify the whole of it, that is, if there are no other commands with the same beginning, then it will be completed automatically. In Cisco equipment it is enough to enter only the first few letters of the command for the operating system to understand the meaning of it. For example, the command:

```
vlan-r1#conf t
```

is adequate to the command:

```
vlan-r1#configure terminal
```

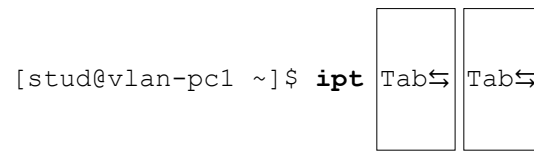
In Cisco equipment, if it is not clear what is the exact name of the command, it is enough only to type in the beginning of the command. Then after pressing the "?", all commands with such a beginning will be displayed. For example:

```
Vlan-r1#di?

disable disconnect
```

If "?" is pressed, when no letters have been typed in, then all commands possible at that time will be displayed. Also, by pressing the "?", it is possible to learn about the parameters of the command.

In Linux OS all available commands are displayed by double clicking the Tab key. For example:




```
iptab     iptables-restore iptraf     iptunnel

iptables  iptables-save    iptstate
```

Secondly, in the CLI mode it is very convenient to use the command line history. This is a special cache that stores previously entered commands. This buffer is reviewed with the arrow keys "Up" and "Down". After clicking the arrow "Up" once, the command that has been previously entered and executed is displayed. Pressing the key again displays yet another, even more previously executed command.

In Cisco and HP equipment there are several CLI modes, which determine what actions can be done. For example, even if the command is correct, but it has been typed in the wrong mode, the error message appears.

The explanations of Cisco CLI modes:

`vlan-r1>` – limited configuration mode. In this mode it is not possible to make any changes. The transition to the privileged mode is done by typing in the command **`enable`**.

`vlan-r1#` – privileged mode, also called the EXEC mode. This mode allows viewing of the statistics and the operational configuration of the device. From this mode it is possible to switch into configuration mode using the command **`configure terminal`**.

`vlan-r1(config)#` – global configuration mode, where node configuration can be performed.

`vlan-r1(config-f)#` – interface configuration mode, where only the settings of a particular interface (or its group) can be changed.

All switches and routers are configured by using serial port terminal software, when the equipment is connected to a PC using the RS232 interface. This enables the configuration of network equipment, even in these cases, when due to incorrect network configuration it is not possible to connect to the equipment using telnet or ssh programs.

Linux OS is configured with standard terminal emulator programs, such as gnome-terminal.

1. SERIAL COMMUNICATIONS

1.1. Transmission modes

Transmitter and receiver are the key elements of data transmission systems. Most of the data transmission characteristics depend on the structure of the transmitter. For instance, frequency clock generator of the transmitter is directly responsible for information transfer rate. The higher frequency of the generator, at a higher speed the data is transmitted.

The data itself from the transmitter to the receiver can be transmitted in parallel or sequential manner. In the parallel transmission the bits are grouped by amount of *n* and are transmitted through the same amount of lines of communication with the receiver to the transmitter with each clock tick.

However, a serial way of communication in telecommunication networks is used more often. In this case, during one data rate clock tick of the generator only one information symbol is transferred (usually a symbol of one bit) (Figure 1.1). Such method of transmission is more practical for long distances, besides, it allows for higher transmission rate by avoiding the problem of Clock screw, when different parallel lines delay the signal to a different time.

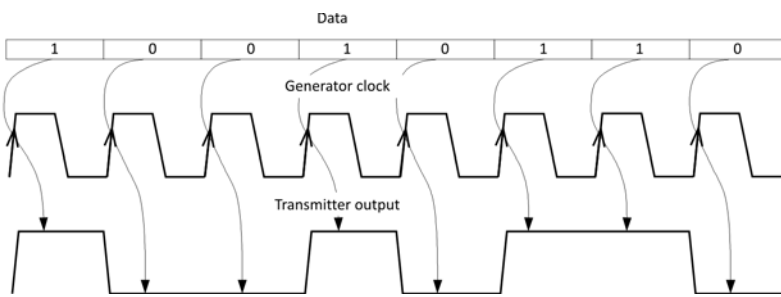


Figure 1.1. Principles of serial data transmission

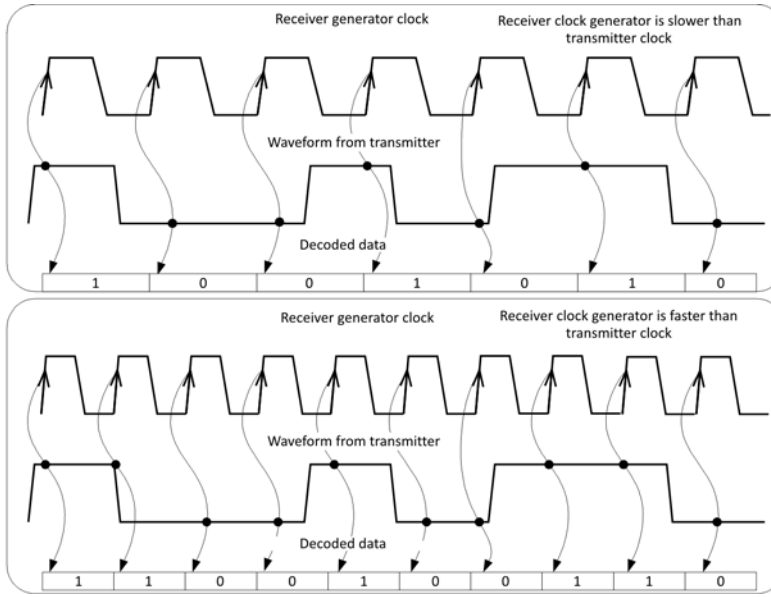


Figure 1.2. Transmitter synchronization problem, when transmitted data was 0b10010110

1.2. Asynchronous and Synchronous Transmission

The main task of communication systems receivers is to unambiguously interpret the received data. It is therefore important that the receiver and transmitter use the same channel codes and source codes. But it is no less important to ensure the synchronization of receiver and transmitter. For the receiver to be able to unequivocally accept the data it is necessary that frequency of clock generators of transmitter and the receiver operate simultaneously. In other words, the receiver must know when the bit sent by the transmitter starts and when it ends (Figure 1.1). When the generators of the transmitter and receiver are not synchronized, it allows for the receiving error to occur (Figure 1.2). Depending on the fact if generator of the

receiver rushes ahead or lags behind, the decoding accordingly will yield more or less data bits. In most cases the data decoded by the receiver in such a manner is useless; because it differs from the data sent by transmitter. The problem of synchronization occurs because of frequency and phase instability that generators in real systems have. Therefore it is not sufficient only to identify the same rate on the receiver and transmitter, it is necessary for the receiver to continuously track the data rate clock ticks that are generated by the generator of the transmitter.

Technology, which is intended for unambiguous interpretation of individual bits, is called bit synchronization technology. Depending on how the technology works, serial data transmission is divided into: synchronous data transmission and asynchronous data transmission.

In case of synchronous transmission the data rate clock generators of the transmitter and receiver operate synchronously. And at the same time it is possible not only to transfer data from transmitter to receiver, but also to transfer the data rate clock ticks of transmitter's generator, which is usually done by a separate line of communication.

In case of asynchronous data transmission the generators of transmitter and receiver operate independently. Still synchronized ticks as special symbols must be inserted into the transmitted data sequence in order to harmonize the phase and rate of the receiver.

It is also necessary to mention that the information in data network is transmitted not in the individual bits but in frames, less in bytes, so it is important in the bit stream to ensure for correct identification when the frame begins and when it ends. This task is solved by synchronization of frame or byte.

It should be noted that bit synchronization works in the physical level, and the synchronization of frame in data link level.

1.3. Synchronous transmission

In case of synchronous transmission the data transmitter transfers not only the data but also the data rate clock ticks of generator. Such scheme of transmission is the most accurate, because the generator of the receiver is managed directly by the generator of the transmitter, which allows for higher data transfer rates.

Synchronous transmission may be implemented in two different ways: the transmitter sends data rate clock ticks to the receiver via separate communication line (Figure 1.3, a); or data rate clock ticks from the transmitter may be transferred to the receiver by the same communication line along with data (Figure 1.3, b).

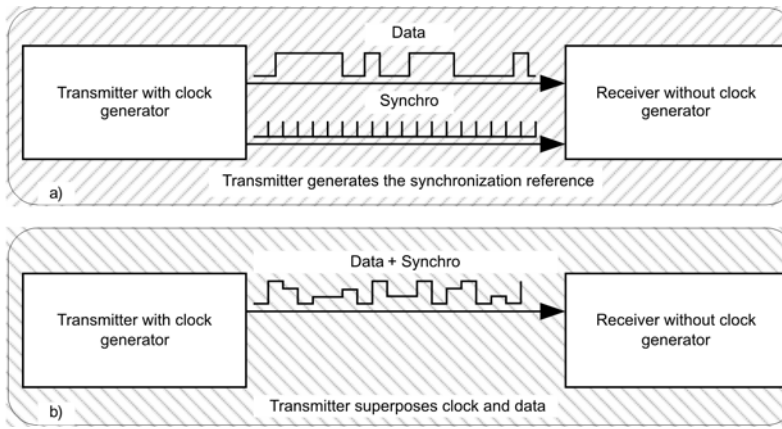


Figure 1.3. Synchronous transmission modes

The first option is usually used to connect a broadband modem to the router using the following interfaces as V.35, V.36, and X.21. The second one is more often found in communication systems, which use Manchester, HDB3, MLT-3, ISL-5 and etc. as the line codes. Manchester code is used for Ethernet 10Base-T, 10BASE5 networks, and MLT-5 code is used in 100BaseTX networks and

HDB3 - E1 time-division multiplexing (TDMA) systems. This second method often allows for less strict sync as no synchronous ticks are transmitted when transmitter does not send the data, in which case synchronization between the transmitter and the receiver is lost.

In case of synchronous transmission data is always transmitted in frames, i.e. never a single byte or bit. Therefore, in order to distinguish one frame from the other a start and a stop flags are used (Figure 1.4).

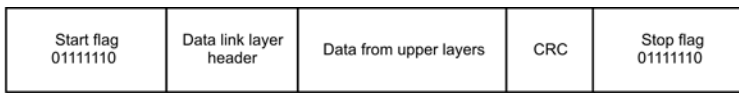


Figure 1.4. Start and stop flags for synchronous data transmission

1.4. Asynchronous transmission

In case of asynchronous transmission signal has information only about the beginning and the end of the symbol and an individual synchronization of each bit is not performed. In order for the receiver to receive data without errors, it is essential that data rates of the transmitter and the receiver are set at the same speed before the transmission. It is also important that other transmitter and receiver parameters, such as the amount of data sent in one symbol or parity bit, must be the same both in the transmitter and the receiver.

If the transmitter does not send the data, line is set to IDLE state, which may continue until the transmitter will need to send data (Figure 1.5). Before the transfer of data the transmitter sends to the receiver a special START bit, which value is opposite to IDLE state and which sets an initial phase for the receiver's generator. As soon as the receiver receives a START bit, i.e., finds out that the transmitter has started sending the data bits and knows the number

of bits to be transmitted, it can receive data without errors. When all the bits of the symbol are transferred, then the STOP bit, which value is opposite to the START bit, is transmitted. STOP bit is required in cases when the data is transmitted one after another, as it allows determining where the end of one data symbol and the next starts is. STOP bit can be one or two bits.

Asynchronous transmission does not require strict synchronization of transmitter's and receiver's data rate clock generators. Data may be transmitted at random intervals or continuously.

In asynchronous transmission parity bit is used for detection of errors. It shows what numbers (even or odd) of logical "1" were in the transferred symbol (byte). Parity bit is transmitted before the STOP bit.

There are two algorithms which explain the value of parity bit. According to ODD algorithm, when the number of "1" in the symbol is even, the parity bit is equal to "1", and according to EVEN algorithm, when the number of "1" in the symbol is even, the parity bit is equal to "0". The parity is an additional asynchronous transmission function and its use is not mandatory.

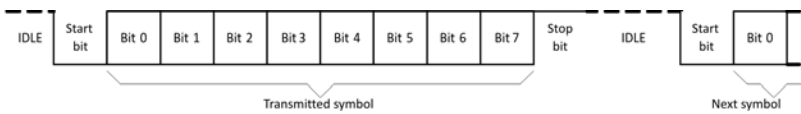


Figure 1.5. Asynchronous transmission

Asynchronous transmission is more marketable, because it is cheaper to implement and is used more intensive comparing to synchronous.

1.5. RS232 interface

RS232 is an asynchronous data transmission interface widely used in telecommunications networks. It is designed to connect

DTE (Data Terminal Equipment) and DCE (Data Circuit-terminating Equipment) in point-to-point configuration. In asynchronous serial transmission RS232 standard defines the electrical characteristics and timing of signals, the meaning of signals, and the pinout and physical size of connectors.

Earlier this standard has been widely used for data transmission, but nowadays its use for data transmission between DTE and DCE equipment is abundantly declining because of the high-voltage, high transmission speed of connectors and low transmission speed of data, and more over this standard is replaced by Universal Serial Bus (USB).

However, the RS232 is unique in configuration of network devices and its diagnostics. It is also widely used in data transmission of low power and speed between devices such as embedded devices.

For transmission of signals RS232 uses a bipolar signal, when logic "1" represents voltage from -5V to -15V and logic "0" represents voltage from +5V to +15V. For the reference, computers and other telecommunication devices use voltage of ±12V (Figure 1.6).

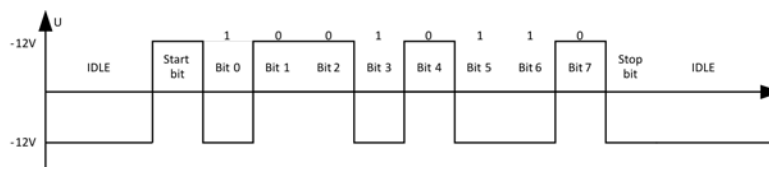


Figure 1.6. RS232 signal wave form, then symbol "i" (binary ASCII code 0b01101001) is transmitted

Since the RS232 uses a single ended data transmission, 3 frame pins are enough for full duplex data transfer: Tx – to transmit data, Rx – to receive data, and GND – ground. In addition to these contacts, RS232 defines 6 more pins, which are designed to transmit additional signaling of computer and modem. RTS (Request To Send) /

CTS (Clear To Send) pins used to the flow control, in other words, appoint the modem to inform the computer when the date buffers are filled in. DTR (Data Terminal Ready) pin is used to indicate for the modem that the computer is turned on. DSR (Data Set Ready) gives a signal to computer that the modem is turned on. DCD pin allows for the modem to inform the computer that it has a connection to a remote modem. And finally RI pin gives a signal to computer in order to inform about the call to the telephone line, which is connected to the modem.

RS232 settings are often described by codes such as 9600 8N1. The first number in this code explains the speed b/s, the second number shows how many data bits create a symbol, a letter "N" indicates that parity bit is not used, and the last number reflects the amount of STOP bits that are transmitted.

The purposes of all signals used by RS232 standard along with DB9 pin are displayed in Table 1.1.

Table 1.1. RS232 signals

Abbreviation	Name	Direction	DB9 pin
TxD	Transmitted Data	DTE→DCE	3
RxD	Received Data	DCE→DTE	2
DTR	Data Terminal Ready	DTE→DCE	4
DSR	Data Set Ready	DCE→DTE	6
RTS	Request To Send	DTE→DCE	7
CTS	Clear To Send	DCE→DTE	8
DCD	Data Cartier Detect	DCE→DTE	1
RI	Ring Indicator	DCE→DTE	9
GND	Common Ground		5

1.6. Laboratory exercise. Analysis of RS232 interface

Goal of Exercise: get acquainted with RS232 interface, measure the characteristics of the signals.

Tasks of Exercise: learn how to configure the software to work with RS232 port, measure the signal amplitude and timing characteristics of a variety of data rates, to examine the RS232 signals when parity and various combinations of the STOP bit are used.

Equipment: one personal computer with Linux OS and special software, testing model and digital oscilloscope.

Workflow:

1. Preparation:

   1.1. Run two serial port terminal programs GTKTerm[1] on the PC. In case of Fedora Core Gnome environment run Applications → Accessories → GTKTerm. Find Configuration → Port and set the following parameters on one copy of the program: Port: ttyS0, Speed: 9600, Parity: None, Bits: 8, Stopbits: 1, Flow Control: None. Set ttyS1 port instead of ttyS0 on the second copy of the program.

   1.2. Note: in order to make sure that the configuration has been executed correctly type any key on the keyboard. If the characters appear on the screen of other terminal, the configuration is correct. The characters are coded using ASCII, i.e., each character corresponds to a unique combination of eight bits.

   1.3. Note: oscilloscope shows the data transmitted by ttyS0 port, therefore all following actions must be completed using GTKTerm program, which is connected to ttyS0 interface (Figure 1.7).

[1] Laboratory uses the upgraded GTKTerm software. The source code is available at http://tc.el.vgtu.lt/stud/software/GTKTerm.

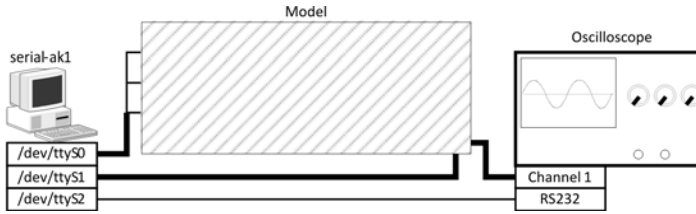


Figure 1.7. RS232 laboratory diagram

1.4. Adjust the oscilloscope. Press and hold Shift + u keys on GTKTerm window. In the correct case, 8 pulses should be seen throughout the oscilloscope screen.

1.5. Set oscilloscope into digital mode by clicking MEM button, this will facilitate the receiving of oscillograms. Oscilloscope will show or hold the image when clicking START / STOP buttons.

2. Examination of RS232 signals. Both examinations are executed with 8 bits length data and one STOP bit.

2.1. Experiment I. Analysis of character transmitted by 9600 b/s rate. Measure one freely chosen character oscillogram. Run character generator in order for the chosen character by automatically generated on GTKTerm program: Laboratory → Character Generator. Type the chosen character in the field ASCII character. Set "1" in the field Number of characters. Click START button for constant generation of character.

2.2. While using oscilloscope measure the amplitude and the shortest pulse duration of the given character. Write the results in Table 1.2.

Table 1.2. RS232 experiments results

Experiment #	Amplitude, *V*		Duration of pulse, *ms*
	Logic "0"	Logic "1"	
I			
II			

2.3. From the oscillogram find out the positions of logical ones and zeros; the position of START and STOP bits; the position of each bit from the transmitted character. While using osci2png[2] software send the oscillogram to the PC. Note: this program runs from the command line.

2.4. Experiment II. Analysis of character transmitted by 1200 b/s rate. Set the rate of the port at 1200 b/s in both GTKTerm programs. Repeat Experiment I. During this experiment make sure again that the transmitted character is well seen on the screen. If need be, adjust the oscilloscope.

2.5. Experiment III. Analysis of one and two Stop bits. Set "2" in the field "Number of characters". Note: generator will generate two successive characters. Turn the generator on. Make sure that both transmitted characters are well seen on the screen. If need be, adjust the oscilloscope. Send the oscillogram to the PC. Stop the generator.

2.6. Set the number of STOP bits equal to "2" on both GTKTerm programs. Turn the generator on again. Find out how this signal is different from the signal obtained in task 2.5.

2.7. Experiment IV. Analysis of parity bits. Set the number of STOP bits equal to "1" on both GTKTerm programs. Turn parity control EVEN on. Set type "Incremeted" in Next character field. Note: this means that the next symbol in the sequence of characters will be increased by one. Turn the generator on. Send the oscillogram to the PC. Find out in which case and when while transmitting odd and even number of units the parity bit appears.

2.8. Change parity control ODD on. Repeat the task 2.7.

---

[2] Osci2png program is specifically designed for the transfer of the oscillograms from the oscilloscope to a computer. The source code is available at http://tc.el.vgtu.lt/stud/Software/osci2png.

Content of Laboratory exercise report:

1. The oscillograms of chosen character. Explication of START and STOP bits positions.
2. The results in Table 1.2.
3. Conclusions.

Questions:

1. What is the reason for crossing of wires while connecting DTE-DTE?
2. By which voltage values the bits are being coded in RS232 standard?
3. Describe the port settings: speed, parity, data, stop bits.
4. Plot the RS232 signal when 9600 8O2 transmission scheme is used to transmit a word "VGTU" coded with ASCII.
5. What is the difference between ODD and EVEN parity schemes?
6. What do the notes 9600 8N1, 19200 8E2 mean?
7. What is the difference between hardware flow control and software flow control?
8. Suppose that the scheme 9600 8N2 is used for data transmission. How much time will it take exactly to transmit a 10 MB file? How many additional bits need to be transmitted in order to ensure synchronization between transmitter and receiver?
9. What is the use of RS232 DTR, DSR, DCD and RI signals?
10. Why RS232 signals cannot be connected to the TTL schemes directly?

2. ARP PROTOCOL

2.1. Types of network devices address

Nowadays network layer uses mainly only IP protocol. Although data link layer uses a significant range of various protocols, nevertheless, both local and backbone networks started increasingly use the Ethernet technology. Different protocols of data link layer have their own data link addresses, the so-called physical addresses. Such a name is given based on the fact that the address itself is usually implemented in hardware, for example, the physical address of Ethernet is 6 bytes and it is initially set by Ethernet interface network card manufacturer. Most often physical addresses are called MAC addresses. Physical addresses are used locally, namely, they are managed by network administrators, so it is possible that neighboring networks might use the same physical addresses. This is not a problem for data transmission, because the sender sends the data to the recipient's network layer address or IP address. The network layer address is global, so the situation when the network would have two nodes with the same logical addresses is practically impossible. Therefore only global logical addresses are used for data transmission. This in turn, makes it necessary to relate local physical and logical global addresses depending on the data link layer technology. This can be done in two ways. The first one is static address mapping, when the network administrator creates the table, which is filled with the physical and logical addresses of the network nodes, and then places the table into each network node. In order to transmit the data to another node with familiar IP address, the first network node refers to the static address table for the MAC address of the recipient. Such a solution has several drawbacks, primary among which is that when computer changes the network interface, this would change its MAC address; as a result the table would have to be updated in all network

nodes. In order to avoid this lack, the second way has been created: several newly developed protocols perform an automatic assembly of physical and logical address. In the IPv4 networks this is done by the ARP protocol, and in the IPv6 networks – by NDP protocol.

2.2. ARP protocol

The principle of ARP protocol is to send messages requesting MAC address. Messages are of two types: request and response. Request is always sent to the broadcast address, so all network nodes in the local network receive it. After receiving the ARP request each node performs the analysis in order to determine whether the request was sent personally for it. The aim of the analysis is to examine the Target Protocol IP address (Figure 2.1) in the ARP request packet, and to compare it with the node own IP address. If the IP addresses match, then the node sends a reply indicating its physical and logical address. This reply is transmitted only to the sender of request, and it is not sent to the broadcast address.

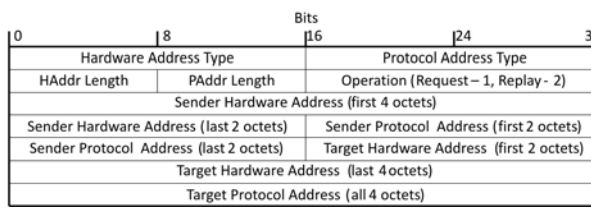


Figure 2.1. Structure of ARP packet

So that it would not be needed to repeat the same procedure of ARP protocol, the responses that were given are written into ARP cache. In this way, it is possible to increase the efficiency of the ARP protocol. Each entry of MAC address is present in an intermediate memory for a certain period, typically about 5 minutes, after which it is removed and the cycle repeats again.

The detailed illustration of ARP mechanism is given in Figure 2.2. Suppose, Node1 is ready for sending a data packet to Node4 and is already aware of Node4's IP address by using the DNS service. Before the data is sent from the network layer to the data link layer a check-up is made in order to find out if Node1 ARP table has an entry with Node4's MAC address. If the destination's MAC address is not in the table, then the ARP request packet is sent to the broadcast address (Figure 2.2), asking for MAC address of the node, which has an IP address of 10.0.0.4. All network nodes in the local network receive this request, but the reply is given solely by Node4, because it has the IP address of 10.0.0.4. Node4 sends a response to Node1, showing its MAC address. Node1 writes it into the ARP cache memory. As soon as the MAC address in the ARP table is found, the data packet is encapsulated into a data link layer protocol, which header has the information about Node1 and Node4 as the sender and the destination and their MAC addresses correspondingly.

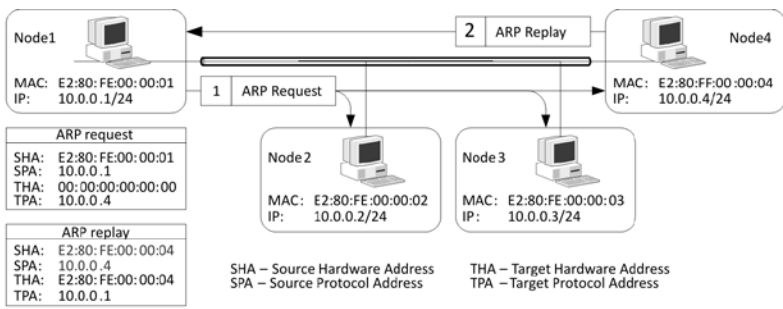


Figure 2.2. Principles of ARP protocol operations

When the sender and destination are on the same IP network subnet, the ARP protocol looks for the destination's MAC address. And when the sender and destination are on different IP subnets, ARP protocol searches the MAC address of that router, which IP

address is listed in the routing table. This IP address indicates an intermediate node, through which it is possible to reach the destination. If the node has only default gateway, then all the packets, which are supposed to be sent to the nodes in not local network, will be sent through the default gateway.

2.3. The structure of ARP packet

The structure of ARP packet is presented in Figure 2.1.

Hardware type: This 16-bit field defines the type of network, in which the ARP protocol is used. For Ethernet this field is equal to 1, for Fiber Channel its value is 18.

Protocol type: This field refers to which network layer protocol is used. In IPv4 case, its value is 0x0800.

Hardware length: It defines what a physical address length is. In Ethernet case it is 6.

Protocol length: Specifies what a logical address length is. In case of IPv4 protocol the value of the field is 4.

Operation: Indicates what the type of the packet is. Most commonly two types are used: 1 – request, 2 – replay.

Sender hardware address: This is a variable field; it contains the sender's physical address. In Ethernet network, the size of the field is 6 bytes.

Sender Protocol address: It describes the logical address of the sender. In case of IPv4 protocol, the size of the field is 4 bytes.

Target hardware address: It describes the physical target address. The value of the request packet for this field is equal to 0, because the transmitter is not aware of target's physical address.

Target protocol address: It describes the logical target address. Replay message is sent only by the station, whose address is in this field.

ARP packet is directly encapsulated into date link frame. In order for nodes in the network to identify the ARP packet, in the Type

field of data link frame the ARP packet type 0x0806 is indicated (Figure 2.3).

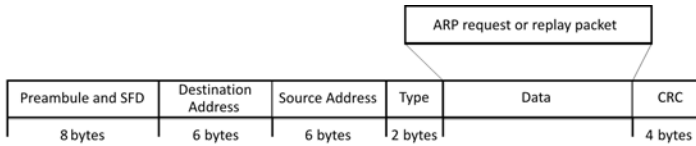


Figure 2.3. Encapsulation of ARP packet into Ethernet frame

2.4. Laboratory exercise. Analysis of ARP protocol

Goal of Exercise: examine the operation of the ARP protocol in Ethernet LAN network.

Tasks of Exercise: while using Linux and Cisco equipment learn to manage the ARP table, find out what is the ARP protocol algorithm, and analyze the structure of this protocol packet.

Equipment: three personal computers with Linux OS, Cisco 2621 router, any Ethernet switch.

Workflow:

1. Preparation:
   1.1. Examine the network diagram, given in Figure 2.4.
   1.2. Run terminal program and web browser on both PCs. Terminal program can be either gnome-terminal or xterm. In case of Fedora Core Gnome environment gnome-terminal program is set: Applications → System Tools → Terminal. Web browser might be Firefox or other. In case of Fedora Core Gnome environment Firefox program is set: Applications → Internet → Internet Browser.
   1.3. Run serial port terminal program GTKTerm on the arp-pc1 computer. In case of Fedora Core Gnome environment run Applications → Accessories → GTKTerm. Find Configuration

→ Port and set the following parameters: Port: ttyS0, Speed: 9600, Parity: None, Bits: 8, Stopbits: 1, Flow Control: None.

2. Analysis of ARP protocol. In Linux operating system ARP table can be created with two commands: arp and ip. Although arp command is founding in almost all operating systems, ip command is used in this laboratory exercise, because it is more flexible and has more control over the ARP table.

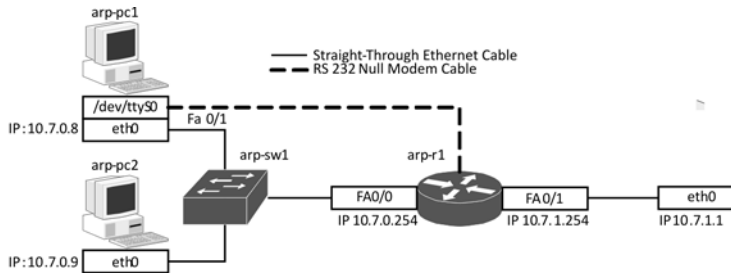


Figure 2.4. Network diagram of ARP protocol analysis laboratory exercise

2.1. Write down each computer's IP and MAC addresses and IP routing table. Write data into Table 2.1. In order to find out IP and MAC addresses, run the following command on both PCs:

```
[stud@arp-pcX ~]$ sudo ip addr show
```

The results explanations of this program are provided in Figure 2.5.

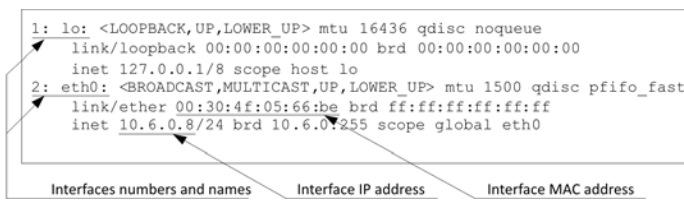


Figure 2.5. Results of ip addr show command

In order to dump the routing table of the PCs, use the following program:

```
[stud@arp-pcX ~]$ sudo ip route show
```

The results explanations of this program are provided in Figure 2.6.

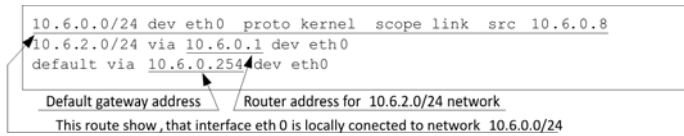


Figure 2.6. Results and explanations of ip route show command

2.2. Write down IP and MAC addresses of the router and IP routing table. Write data into Table 2.1. In order to find router's IP and MAC[3] addresses, type the following commands in GTKTerm window:

```
arp-r1>enable
arp-r1#show interfaces fastEthernet 0/0
arp-r1#show interfaces fastEthernet 0/1
```

The results explanations of these programs are provided in Figure 2.7.

Table 2.1. List of computers and routers IP, MAC addresses

Hostname	Interface	IP address	MAC address	Default gateway
arp-pc1	eth0			
arp-pc2	eth0			
arp-r1	fa0/0			
arp-r1	fa0/1			

[3] Please note that Linux OS MAC address and Cisco IOS MAC address are written differently.

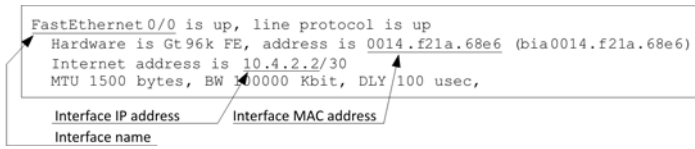


Figure 2.7. Results and explanations of show interfaces command

In order to dump the router's routing table, use the following command:

```
arp-r1#show ip route
```

The results explanations of this program are provided in Figure 2.8.

2.3. In order to check the connection between the PCs: arpk-pc1 and arp-pc2, run the following command on arp-pc1 PC:

```
[stud@arp-pc1 ~]$ ping -c 5 IP_address_of_arp_pc2
```

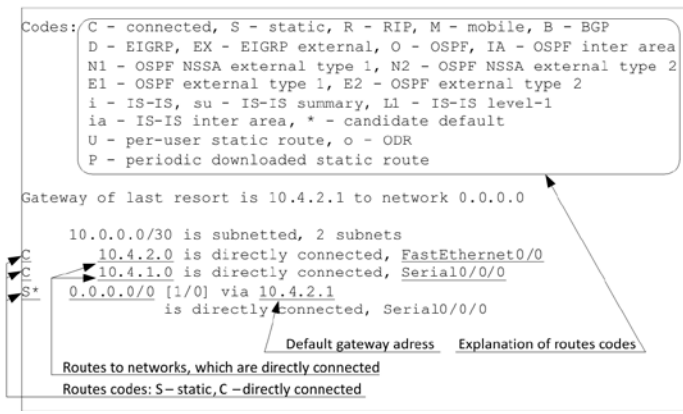


Figure 2.8. Results and explanations of show ip route command

Table 2.2. Results of ip neigh show command

Experiment #	Node	Neighbor IP	Neighbor MAC	State
I	arp-pc1			
	arp-pc2			
	arp-r1			
II	arp-pc1			
	arp-pc2			
	arp-r1			
III	arp-pc2			
	arp-r1			
	arp-r1			
IV	arp-pc1			
	arp-pc2			
	arp-r1			

In this case *IP_addresss_of_arp_pc2* corresponds to the IP address of arp-pc2.

2.4. Experiment I. In order to dump the ARP routing table, run the following command on both PCs:

```
[stud@arp-pcX ~]$sudo ip neigh show
```

The results explanations of the program are provided in Figure 2.9.

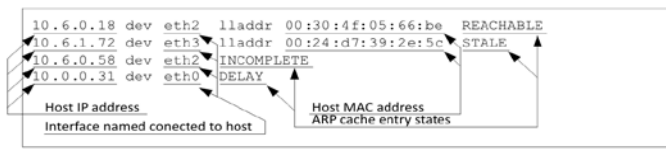


Figure 2.9. Results and explanations of ip neigh show command

Enter the obtained results into Table 2.2.

2.5. Experiment II. Go to the page www.vgtu.lt on the web browser. Check for new entries in the ARP table. Add the new data into Table 2.2. Answer the following question and give reasons for the answer:

Question 1. Does the new entry appear on the same computer, which has been tested for the connection?

2.6. Experiment III. In order to create a static entry of arp-pc2 ARP table in arp-pc1 computer, run the following commands:

```
[stud@arp-pc1 ~]$ sudo ip neigh flush dev eth0
```

```
[stud@arp-pc1 ~]$ sudo ip neigh add IP_address_of_
arp_pc2 lladdr aa:bb:cc:dd:ee:ff dev eth0
```

2.7. Check the connection between arp-pc1 and arp-pc2.

2.8. Dump the arp-pc1 ARP routing table. Add the new data into Table 2.2. Answer the following question and give reasons for the answer:

Question 2. Is there a connection between arp-pc1 and arp-pc2?

2.9. Experiment IV. In order to delete arp-pc1 permanent arp table entry, use the following command:

```
[stud@arp-pc1 ~]$ sudo ip neigh change IP_address_of_
arp_pc2 lladdr aa:bb:cc:dd:ee:ff dev eth0 nud failed
```

2.10. Check the connection between arpk-pc1 and arp-pc2 again.

2.11. Dump the arp-pc1 ARP routing table. Add the new data into Table 2.2. Answer the following question and give reasons for the answer:

Question 3. Is there a connection between arp-pc1 and arp-pc2?

In order to create a static arp cache entry in the router for arp-pc2 computer, type the following commands in GTKTerm window:

```
arp-r1#conf t
```

```
arp-r1(config)#arp IP_address_of_arp_pc2
0099.8877.6655
```

2.12. Go to any web page from arp-pc2 computer, for example, www.google.lt. Answer the following question and give reasons for the answer:

Question 4. Does arp-pc2 have a connection to the internet?

2.13. In order to delete the static arp cache entry in the router, use the following command:

```
arp-r1(config)# no arp IP_address_of_arp_pc2
0099.8877.6655

arp-r1(config)#exit
```

Repeat task 2.12.

2.14. Check for entry about arp-pc2 computer in the arp cache table of arp-pc1 computer. If there is none, run the command described in task 2.3. Change the eth0 interface MAC address in arp-pc2 computer using the following command:

```
[stud@arp-pc2 ~]$ sudo ip link set dev eth0
address 00:11:22:33:44:55
```

2.15. Check the connection between arp-pc1 and arp-pc2 again, only this time set the ping command to stop only after the Ctrl + C key combination is pressed. Run the following command for this:

```
[stud@arp-pc1 ~]$ ping IP_address_of_arp_pc2
```

Answer the following questions:

Question 5. Was the connection between arp-pc1 and arp-pc2 lost? After how long was it restored and why?

2.16. Stop arp-pc1 ping command.

3. Analysis of ARP protocol packets structure. For ARP packet analysis it is best to use the packets analysing program wireshark. This is software with an open source and a user friendly interface, which enables to analyse the packets content.

   3.1. Start Wireshark software. In case of Fedora Core Gnome environment this program is set: Applications → Internet → Wireshark Network Analyser.

Figure 2.10. Wireshark capture interfaces window

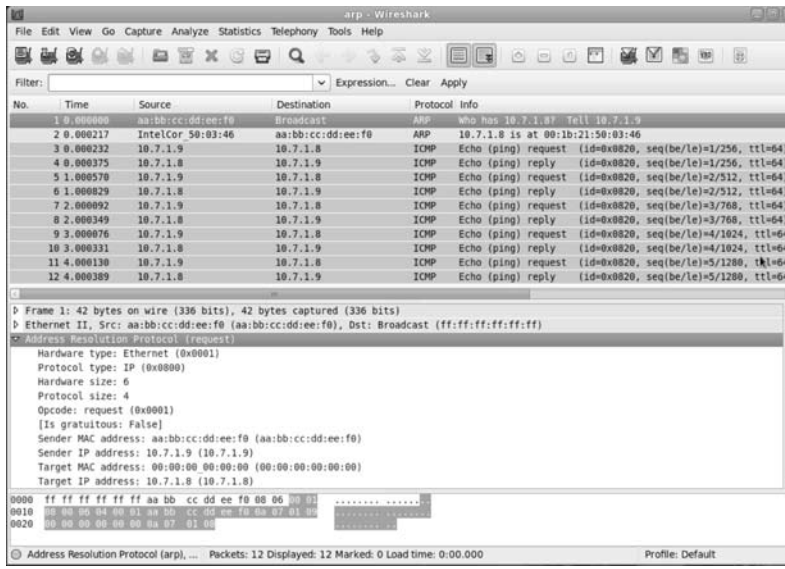


Figure 2.11. Wireshark packet display

3.2. In order to configure Wireshark for packet capturing, click: Capture → Interfaces. Then click Start button on eth0 interface line (Figure2.11).

3.3. In order to clear the arp tables on both PCs, run the following command:

```
[stud@arp-pcX ~]$sudo ip neigh flush dev eth0
```

3.4. Check the connection between arpk-pc1 and arp-pc2 as described in task 2.3.

3.5. In order to stop packets capture in Wireshark program, click Capture → Stop. Wireshark program window should look like, as shown in Figure 2.11. After examining the packets captured by Wireshark, answer the following questions:
Question 6. What type of ARP packet has been sent first?
Question 7. What type of ARP packet has been sent second?
3.6. Fill in Table 2.3 with the content of the first ARP packet.
3.7. Fill in Table 2.4 with the content of the second ARP packet.

Table 2.3. Data of the first ARP packet

Ethernet II frame	
Field	Value
Destination MAC address	
Source MAC address	
ARP packet	
Field	Value
Sender MAC address	
Sender IP address	
Target MAC address	
Target IP address	

Table 2.4. Data of the second ARP packet

Ethernet II frame	
Field	Value
Destination MAC address	
Source MAC address	
ARP packet	
Field	Value
Sender MAC address	
Sender IP address	
Target MAC address	
Target IP address	

Content of Laboratory exercise report:

1. The network diagram of experiments.
2. The results of Table 2.1, Table 2.2, Table 2.3, Table 2.4.
3. Answers to all questions.
4. Conclusions.

Questions:

1. What is ARP protocol used for?
2. How does an ARP spoofing type attack of computers network work?
3. When sending data to a non-local IP subnetwork router's MAC, but not the final computer's MAC address appears in ARP table. Why?
4. Why ARP protocol is not used in the network where the network nodes are connected by PPP protocol?
5. When computer's MAC address is changed, the connection failure occurs. Why?
6. What is the difference between the ARP protocol and NDP protocol?
7. When it is useful to create a static ARP entry in the PC?
8. Why the size of ARP packet is fixed?
9. Why the request of ARP is sent to the broadcast address?
10. Why the arp cache table entry is valid only for a limited period of time?

3. IPv4 PROTOCOL

3.1. Model of TCP/IP network

Currently, the TCP / IP protocol stack holds a dominant position in telecommunications networks. This stack has a number of various protocols, but most important one is the IP that performs logical addressing of network nodes. The Internet is namely based on this protocol. The Internet network can be seen as a set of small IP networks, connected with each other by routers (Figure 3.1). A router is a device with multiple network interfaces, which belong to different IP networks. When users of a single IP network want to transmit data to another IP network, the sender forwards its data to a router, according to which a routing table finds the way to the destination. It is worth noting that, if the destination is not connected locally, the router finds a way only to the next intermediate node. So we can say that the router performs two main functions: packet[4] forwarding and route finding.

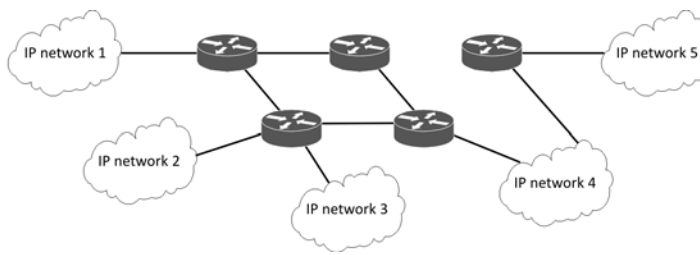


Figure 3.1. Network example, where IP is used as network layer protocol

[4] The term "packet" is used when it comes to data transmission in network layer. "Frame" is a term used when it comes to data transmission in data link layer.

The connection of network facilities requires a global addressing scheme, which is called logical addressing. To this purpose, the IPv4 is currently used in the Internet as it describes the node addressing in 32-bit address. This makes it possible to address 2^{32} network nodes. It is true though that all of the available IPv4 addresses have already been depleted, so there is a gradually transition to the IPv6 standard, which addresses nodes in 128-bit address. Yet, the IPv6 holds a small market share, so this work deals only with the IPv4 protocol.

All Internet addresses in the network are unique. There cannot be any two network nodes in the network at the same time with the same IP addresses. However, since the no address in the address field is available, this rule is partly circumvented by using network address translation (NAT) function, since only the public rather than private addresses must be unique, as they go forth to the Internet. The latter may recur many times in the local networks.

IPv4 address can be showed in several ways:

- 192.168.24.43 – the decimal system when dots are inserted to separate bytes;
- 11000000 10101000 00011000 00101011 – binary system;
- 3232241707 – the decimal system;
- C0A8182B – hexadecimal system.

In practice, the first way is used, as it is an easy to read and more compact form of an IP address.

As has already been mentioned, the global IP network consists of a set of smaller IP networks, which are combined with each other by a router. When a network node needs to transmit data to another network node, first, it checks on what IP network is the destination. If the destination is on the same IP network, then the sender forwards the data packet directly without a router support. If the destination is on another network, then the packet is transmitted to a router, which in turn transmits it to the next router and so on until the packet reaches its destination. It is therefore important to understand how to dis-

tinguish to which IP network the IP address belongs. Previously the classes have been used, i.e. the IP address belonged to a certain class. Currently, such a division is no longer used, but for understanding of the addressing principles it is worth of consideration.

Thus, in case of class addressing the IP address can belong to one of the 5-classes: A, B, C, D, E. These classes cover particular IP address ranges. In order to decide to which class IP address belongs, a value of the first byte is considered (Table 3.1). One part of bytes in class addressing is allocated to address the network, and the rest of bytes are used to address network nodes in the network.

Table 3.1. The classification of IP classes

Class	First byte	Bytes for computer addressing	Number of networks	Number of hosts
A	0–127	3	127	16777216
B	128–191	2	16384	65536
C	192–223	1	2097152	256
D	224–239	Multicast	-	268435456
E	240–255	Reserved	-	268435456

Each network has a set of IP address belonging to the network nodes and two specific IP addresses. The first is called network address, and the last is called network broadcast address. The network address is used to describe the network and broadcast address is used in cases when data needs to be transmitted to all nodes in the network. Therefore, if the network has N number in addresses, then there will always be $N - 2$ of network nodes in this network.

Let us examine the example where the address of network node is 192.168.2.3. This address is assigned to class C (Table 3.1). In this class, three bytes are assigned to addressing of network, and one byte for addressing a node. The smallest value of byte is 0, while the highest is 255. So in this case, the network address is 192.168.2.0 and the broadcast address is 192.168.2.255.

Let us analyse another address, for example, 172.16.4.5. This address is assigned to class B. Thus, for addressing a node two bytes, when the minimum value of a byte is 0.0 and the maximum 255.255. So in this case, the network address is 172.16.0.0, and the broadcast address is 172.16.255.255.

There are a number of addresses that are reserved, i.e., they do not belong to the classes:

- address 127.0.0.1 – loop back interface address;
- address 0.0.0.0 – default route;
- address 255.255.255.255 – broadcast address (for all nodes in the network).

From Table 3.1 it follows that the highest number of computer-assigned addresses is in class A, while in class C, the number is the least. In general these blocks of fixed number of addresses are inflexible and impractical. For example, depending on the needs, one may need only a few addresses, while in the other case, even thousands of addresses maybe needed. Therefore, in practice classless addressing is being used. It is when a block of certain class addresses is divided into smaller units, i.e. the bits that were allocated to node addressing, are assigned for network addressing. In this way the network is divided into subnetworks. When dividing into blocks it is necessary to comply with the following rules:

i) addresses in the blocks must be contiguous to one another;

ii) the number of addresses in the block must be a power of 2 (1, 2, 4, 8, 16, ...);

iii) The beginning address must be evenly divisible by the number of addresses. For example, when 4 subnets need to be created, then the first address of each block must be evenly divisible by the number of 4.

In order to determine what part in the IP address defines the network and what the host address, an additional parameter, called netmask, is being introduced. IPv4 netmask is a 4-byte length. This additional parameter can be recorded in two ways: dot-decimal no-

tation (as an IP address), for example: 255.255.255.0, or bit-length of the netmask separated by a slash (/), for example: /24. The notation of the IP address, when netmask is recorded according to the bit-length method, is called classless inter-domain routing (CIDR) notation. The conversion of netmask from one recording system into another is very simple. If the netmask is recorded as an IP address, then netmask should be expressed in binary form and the number consecutive "1" should be counted. The number of "1" should be equal to the bit-length of the netmask. As where the network mask is recorded in bit-length format, then it should be expressed in a binary number from left to right with the number of "1" equal to the bit-length, and then add "0", until the total number of symbols is equal to the 32. Then all that remains is to change this binary number into decimal form breaking it down by eight bits (Figure 3.2).

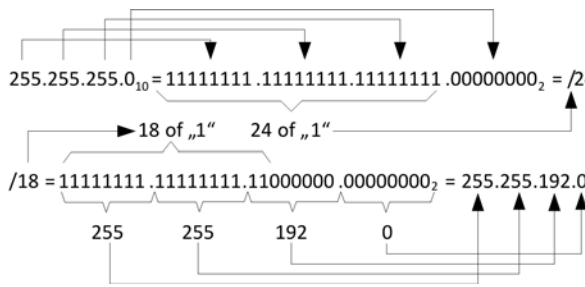


Figure 3.2. Methods of converting netmask between various written forms

In the classless case, the network address is calculated by using a bitwise AND operation between IP address and the netmask. Thus, each netmask bit is responsible for a particular IP address bit. If the netmask value is equal to "1", it means that the corresponding IP address bit is responsible for addressing of network, and if the netmask bit is equal to "0", then the IP address bit is responsible for addressing of node.

For example, the netmask "/27" shows that 27 of the most significant bits in IP address are addressing the network, and the remaining 5 (32 – 27 = 5) address the network nodes. Or in the case of netmask "/8", only the first 8 bits of the IP address are addressing the network and other bits address the node. To illustrate this, let us consider the example given in Figure 3.3. IP address is recorded in CIDR format 192.168.14.14/21. After the bitwise AND operation is accomplished, the network address of the IP address is 192.168.8.0/21.

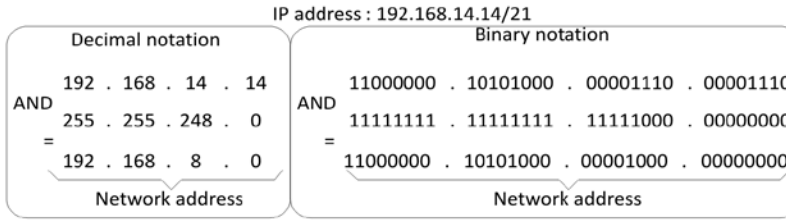


Figure 3.3. Example of network address calculation

Given the IP address and netmask it is easily possible to calculate how many network nodes there are in the network, and what is the network broadcast address. The number of network nodes is equal to: $N = 2^{(32 - NM)}$. N here is the number of network nodes, and NM is bit-length of netmask. Broadcast address is calculated as follows: $BA = NA + N - 1$. BA here is network broadcast address, and NA is subnetwork network address. Let's say, the IP address is 192.168.1.3/27. Then, there can be: $2^{(32 - 27)} = 32$ network nodes in this subnetwork. The address of this subnetwork network is 192.168.1.0, and the broadcast address is 192.168.1.0 + 32 – 1 = 192.168.1.31.

As already mentioned, when dividing the network into subnetworks, it is necessary to follow the three rules. Therefore below are given a few examples in order to explain their use.

Example 1. Let's say that a company receives the field of 256 addresses from the internet service provider, and the network address

is 172.16.10.0[5]/24. The company has two offices, and 100 working places in each office. The field of available addresses needs to be allocated, and split into separate subnetworks. Also it is needed to calculate the network address, broadcast address and the address range, in which network nodes could be addressed, for each subnetwork. First, the size of subnetwork must be found out. This procedure is regulated by ii rule, which defines that the block size must be the power of 2. "100" the number of work places required by the company is not equal to power of 2, so it is needed to find the size of such a block that satisfies the following condition: $N=2^n \geq NA + 2$, here n is an integer, and NA – the number of working places (2 more must be added, because one address is used for network addressing, and the other for broadcast addressing). It appears that if $n = 7$, then $2^7 = 128 > 100 + 2$. Thus, block size N used in both offices is equal to 128 addresses. In accordance with i rule, the first block starts with the following address: 172.16.10.0. The netmask for subnetwork is calculated by $NM = 32 – n$, $NM = 25$.

Thus, the network address of the first subnetwork is 172.16.10.0/25. The broadcast address is calculated by: BA=172.16.10.0 + 128 – 1 = 172.16.10.127. Finally it is possible to record the address field, which is used to address the network nodes. The first address is 172.16.10.1/25, and the last address is 172.16.10.126/25. Further, in accordance with i rule, the second block should start immediately after the first block, so network address of the second block is 172.16.10.128/25. The calculated broadcast address is equal to 172.16.10.255/25. Accordingly, the first and last addresses for the network nodes are 172.16.10.129/25 and 127.16.10.254/25. All results are provided in Table 3.2.

[5] All addresses that are used in this book come from a group of private addresses so that addresses belonging to third parties would not be used.

Table 3.2. Results of first example

Block number	Address	Value
1	Network	172.16.10.0/25
	Broadcast	172.16.10.127/25
	First for nodes	172.16.10.1/25
	Last for nodes	172.16.10.126/25
2	Network	172.16.10.128/25
	Broadcast	172.16.10.255/25
	First for nodes	172.16.10.129/25
	Last for nodes	172.16.10.254/25

Example 2. Suppose that the company has the block of addresses, and the network address of this block is 10.23.231.0/24. The company has to split the field of the addresses among the following four offices, where there are respectively 30, 100, 32 and 20 workplaces. Let's write down the network broadcast addresses of each of these blocks. However, in order to save the space in the book, the addresses used for the network nodes are excluded from calculation. To solve this task, it is important to remember iii rule, which states that the network address of the block must be evenly divisible by the number of addresses. To ensure that this condition is satisfied, it is advisable to start the division of addresses in the largest block and then go to the smallest. Therefore, let's begin by calculate the network size of the largest office: $N = 2^n \geq 100 + 2$, $n = 7$, N = 128. And the netmask: $NM = 32 - n = 25$. Then the network address is 10.23.231.0/25, and broadcast address is 10.23.231.127. Now let's calculate the other block, assigned for 32 workplaces. The network size then is: $N = 2^n \geq 32 + 2$, which means that $n = 6$ and $N = 64$. The netmask is: $NM = 32 - 6$, $NM = 26$. The conclusion is that the network address of the second block is 10.23.231.128/26 and broadcast address is 10.23.231.191.

Now let's make sure of the iii rule is fulfilled: $0 \div 128 = 0$ and $128 \div 64 = 2$, hence the rule is satisfied.

If at first the calculation would have been made for the last subnet, and then the subnet, which is assigned for 100 workplaces, the answer while verifying iii rule, would have been the following: $0 \div 64 = 0$, but $64 \div 128 = 0.5$, which in turn means that the iii rule is not met.

Let us continue by calculating a block, which is assigned for 30 workplaces. We find that for this subnetwork the network size is: $N = 2^n \geq 30 + 2$ and hence $n = 5$, $N = 32$, the network mask is: $NM = 32 - 5$, $NM = 27$. The network address is 10.23.231.192/27, and broadcast address is 10.23.231.223. Finally, let's calculate the last block of addresses, which is assigned for 20 workplaces' office. The network size of this subnetwork is: $N = 2^n \geq 20 + 2$ and hence $n = 5$, $N = 32$, the network mask: $NM = 32 - 5$, $NM = 27$. The network address is 10.23.231.224/27, broadcast address is 10.23.231.255. All results of this example are presented in Table 3.3.

Table 3.3. Results of second example

Block number, block size	Address	Value
1, 128	Network	10.23.231.0/25
	Broadcast	10.23.231.127/25
2, 64	Network	10.23.231.128/26
	Broadcast	10.23.231.191/26
3, 32	Network	10.23.231.192/27
	Broadcast	10.23.231.223/27
4, 32	Network	10.23.231.224/27
	Broadcast	1023.231.255/27

3.2. Routing tables

The most important work over IP networks is carried out by the routers. Unlike the Ethernet switches, the routers must be con-

figured before they start to operate. The main two parameters to be set are as follows: the IP address of router interface and the routing table. IP addresses should be classified as router has to distinguish between the networks which are available locally and which are not. Routing table contains information on how to achieve global networks. For making the routing tables in IP networks the next-hop method is used. In this method routing table has the information on how to achieve the next leap, but there is no information about the entire transmission path, i.e., no intermediate nodes. An example of the routing table created according to the next hop method is given in Figure 3.4. It shows that in the routing tables of Router1 and Router2 it is indicated that IP Network4 is reachable over Router4, and it does not matter that the IP Network4 is not connected to Router4. When the packets come to interfaces of Router1 and Router2 and if their destination is IP Network4, then these routers send the packet to Router4, which in turn transmits the data to Router3. It is also worth emphasizing that not only routers, but all the network nodes have routing tables. The example of PC routing table is presented in Figure 3.4.

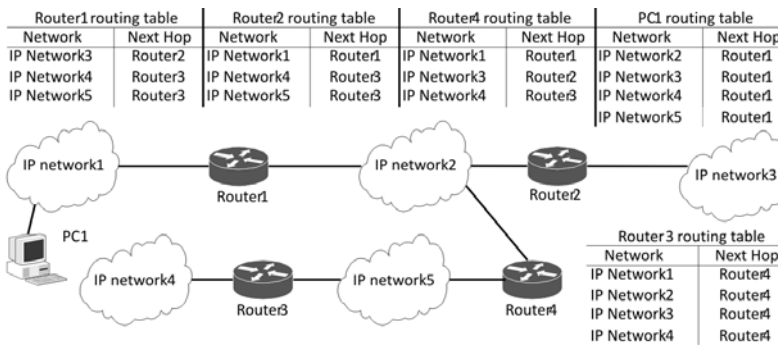


Figure 3.4. Routing tables of next-hop route method

Routing tables are filled in either manually or using routing protocols such as OSPF or BGP4. In this book the highlights only of manually filled routing tables are discussed. Routes that are entered manually are also known as static routes.

When configuring static routes it is necessary to specify the address of remote network and its netmask as well as next-hop router address. Instead of the interface address the output port of the configurable device also can be specified. This port is assigned for packets transmission. Such a choice is available for the serially connected point-to-point network type. These networks use HDLC or PPP protocol for transmission of packets to the port of the nearest device.

As an example, the following Cisco IOS and Linux commands for entering static addresses are shown:

Cisco IOS:

```
router (config) # ip route network_address network
        _mask ip_adress / output_interface
```

Linux OS:

```
[user@computer ~]$ sudo ip route add network_
address / network_mask_bit-lengh via ip_address
```

In here the *network_address* is the IP address of a remote network, which is added to routing table; *network _mask* is mask of the added network, *ip_address* is next-hop router IP address, and *network_mask_bit-length* is the length of netmask.

It is preferable to seek for as small as possible routing table in the network in order for their review while searching for appropriate route, would take the minimum possible time. One static route can include tens, hundreds or even thousands of other routes, so when configuring it is useful to summarize the routes. This could be achieved if:

• destination's network can be summarized into a single network;

• aggregated routes are described having the same output port or the address of the neighboring router's port, to which the packets are sent.

In order to understand the operation of summation the given addresses are written in bits (Figure 3.5). In order to find the aggregated mask (Figure 3.5), one should look from the left side and find, which bits out of all four addresses are identical as well as to determine their quantity. It appears that 22 bits match, so the aggregated netmask then is 255.255.252.0, one can also write with the prefix /22.

The aggregated network address can be found in two ways:

• write out matching bits from 4 addresses, and fill the spaces of other with "0";

• multiply all four addresses using logical IR operation.

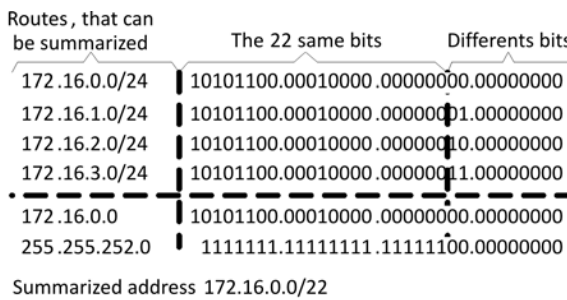
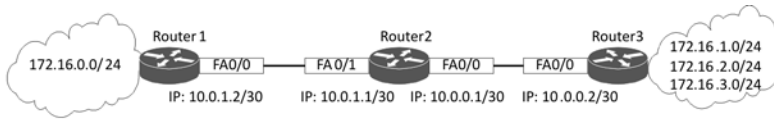


Figure 3.5. Summarized of routes

When filling in the routing table, the devices automatically sort the table according to the size of the netmask. The greater the bit-length, the higher the entry is in the table. This is necessary so that, when entries overlap each other, priority is given for the smaller size network. The example of such network is given in Figure 3.6.

Let's say, there are four networks. In order to lessen routing table of Router2 the networks, which are available through Router3, have been summarized (Figure 3.5) into the following network address: 172.16.0.0/22. An entry about network 172.16.0.0/24, which is available through Router1, is also made in the routing table of Router2.

In this case, the ambiguity arises because the network address 172.16.0.0/24 falls within the address field of the summarized network 172.16.0.0/22. However, after sorting the routing table the device of Router2 checks the route of 172.16.0.0/24 before the route 172.16.0.0/22. Therefore, if the packet is sent, for example, to the 172.16.0.3 address, then it is routed through Route1, and if it is sent to 172.16.1.3, then it is routed through Route3.



Router2 routing table

Network	Netmask	Next hop	Interface
10.0.0.1	255.255.255.252	-	FA 0/0
10.0.1.1	255.255.255.252	-	FA 0/1
172.16.0.0	255.255.255.0	10.0.1.2	FA 0/1
172.16.0.0	255.255.252.0	10.0.0.2	FA 0/0

Figure 3.6. Priority of routing table entries

It should be noted that if the IP network is connected to the router directly the entry about this also appears in the routing table. Only in this case the next-hop field is left blank, because the entry "interface" indicates to which router interface a specific network is connected to.

Currently there are many various IP networks, for example, the Internet consists of millions of networks, but there is no point for the router's routing table to have individual entries of all those networks, because this would definitely slow down the facility of the devices. The static or dynamic routing table entries about the networks, which are from the same organization, are most common-

ly used. The devices reach any other network through the specific routing table entry, called the default route. This entry states that networks, which have not been entered into the routing table, can be available via this route. This is especially obvious when the router is assigned for only one locally connected network (Figure 3.7). PC routing tables usually have only one route, i.e., the default route.

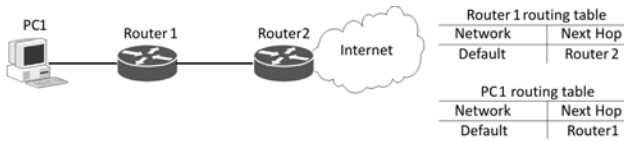


Figure 3.7. Example of default route

Cisco IOS default route is configured in the following way:

```
router (config) # ip route 0.0.0.00.0.0.0 ip_
adress / output_interface
```

Linux OS default route is configured in the following way:

```
[user@computer ~]$ sudo ip route add default via
ip_address
```

In the routing table the default route entry is always checked the last.

3.3. Laboratory exercise. Analysis of static IP routing

Goal of Exercise: learn how to create a simple IP network, which is used for static routing.

Tasks of Exercise: learn how to create simple IP networks, using Linux and Cisco router software. Learn to look for network errors and fix them.

Equipment: two PC with Linux OS, two Cisco routers with one Ethernet and one serial interfaces each.

Workflow:

1. Preparation:

1.1. Examine the network diagram of the laboratory exercise (Figure 3.8).

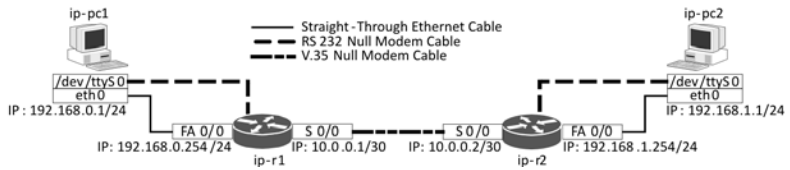


Figure 3.8. Network diagram

1.2. Run terminal program on each PC. Terminal program can be either gnome-terminal or xterm. In case of Fedora Core Gnome environment gnome-terminal program is set: Applications → System Tools → Terminal.

1.3. Run GTKTerm programs on each PC. In case of Fedora Core Gnome environment run Applications → Accessories → GTKTerm. Find Configuration → Port and set the following parameters on one copy of the program: Port: ttyS0, Speed: 9600, Parity: None, Bits: 8, Stopbits: 1, Flow Control: None. Note: with these programs ip-r1 and ip-r2 will be managed.

1.4. If the settings of serial port are correct, after pressing the Enter key, the following string must appear on GTKterm program window:

```
% Please answer 'yes' or 'no'.

Would you like to enter the initial configuration
dialog? [yes/no]:
```

The answer should be: no.

If the string does not appear, this means that the previous configuration of the router has not been deleted. In order to delete the previous configuration, run the following commands:

```
Router>en

Router#erase startup-config

Erasing the nvram filesystem will remove all
configuration files! Continue? [co]
```

Press Enter key:

```
Router#reload

Proceed with reload? [confirm]
```

Press Enter key.

2. Router configuration:

2.1. Perform the configuration of the main ip_r1 router parameters:
In order to switch into the privileged EXEC mode, run the following command:

```
Router>enable
```

In order to switch into the global configuration mode, run the following command:

```
Router #configure terminal
```

In order to set the name of the router, run the following command („*X*“ means the number of the router):

```
Router(config)#hostname ip-rX
```

In order to set the settings of management console, which prohibits the automatic logout when the user is not working with the router, run the following commands:

```
ip-r1 (config)#line con 0

ip-r1 (config-line)#logging synchronous

ip-r1 (config-line)#exec-timeout 0 0

ip-r1 (config-line)#end
```

2.2. Perform the configuration of the interface:
In order to switch into the serial standard interface configuration mode, run the following command:

```
ip-r1(config)#interface serial 0/0

ip-r1(config-if)#clock rate 2000000
```

In order to assign the IP address to the interface, run the following command:

```
ip-r1(config-if)#ip address 10.0.0.1
255.255.255.252
```

In order to connect the interface, run the following command:

```
ip-r1 (config-if)#no shutdown

ip-r1 (config-if)#exit
```

In order to configure another interface, run the following command:

```
ip-r1 (config-if)#interface fastEthernet 0/0

ip-r1 (config-if)#ip address 192.168.1.1
255.255.255.0

ip-r1 (config-if)#no shutdown
```

```
ip-r1 (config-if)#exit

ip-r1 (config)#exit
```

2.3. Perform the configuration of the main ip-r2 router parameters:
Exactly the same configurations are performed for ip-r2 router, only the IP addresses differ (Figure 3.8).
2.4. In order to make sure that there is a connection between the routers, run the following command in ip-r1 device:

```
ip-r1#ping 10.0.0.2
```

If all commands were entered correctly, the terminal should display the output of the ping command, similar to those presented in Figure 3.9:

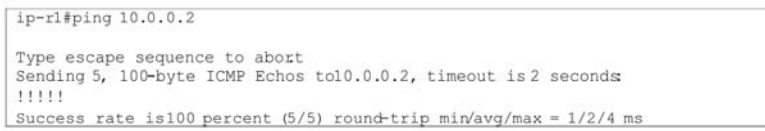


Figure 3.9. Connectivity testing between ip-r1 and ipr2 routers

Carrying out the laboratory exercises in order to time save do not verify the communication in the opposite direction, i.e., if a response is received from the ip-r2, when ICMP echo request (such is the operation of ping command) is sent from the ip-r1, there is no need to repeat the exercise switching the places of the sender and destination.

3. PC configuration:
3.1. Set the IP address and default gateway for ip-pc1 and ip-pc2. The nearest routers, through which they have a connection with other networks, are the network gateway for the PCs (when configuring indicates the IP addresses of their interfaces).

Hence, the network gateway for ip-pc1 is ip-r1 (interface FA0/0 with IP address 192.168.1.1), and the network gateway for ip-pc2 is ip-r2 (interface FA0/0 with IP address 192.168.2.1).

3.2. In order to assign IP address and netmask for ip-pc1 interface, run the following command in gnome-terminal window:

```
[stud@ip-pc1~]$sudo ip addr add 192.168.1.2/24
brd  + dev eth0
```

In order to assign network gateway for ip-pc1, run the following command:

```
[stud@ip-pc1 ~]$sudo ip route add default via
192.168.1.1
```

Repeat the same commands in order to set the IP address and netmask for ip-pc2 (assigning IP address 192.168.2.2, the netmask 192.168.2.2/24, and network gateway with IP address 192.168.2.1):

```
[stud@ip-pc2 ~]$sudo ip addr add 192.168.2.2/24
brd  + dev eth0
```

```
[stud@ip-pc2 ~]$sudo ip route add default via
192.168.2.1
```

3.3. In order to make sure if PCs have connections to their network gateways, run the following command:

```
[stud@ip-pc2 ~]$ping –c 5 192.168.1.1
```

If all commands were entered correctly, the terminal should display the output of the ping command, similar to those presented in Figure 3.10.

In the same manner check the connection between ip-pc2 and ip-r2.

4. Static route configuration:

4.1. For the routers to be able to forward packets to not local networks, they must have an entry in the routing table, what is the route to those networks. In this examination it is done by entering the static routes:
In case of ip-r1 router, the network has the following IP address 192.168.2.0 and netmask 255.255.255.0, and is accessible through the neighboring R2 router interface 10.0.0.2:

```
ip-r1(config)#ip route 192.168.2.0 255.255.255.0
10.0.0.2
```

For ip-r2 router use a different way of describing a static route, that is, when it is indicated that a not local network is accessible via the device interface. In this case the number and standard of the interface is recorded:

```
ip-r2(config)#ip route 192.168.1.0 255.255.255.0
serial 0/0
```

Check any selected device for connection with the indirectly connected subnet using the ping command.

5. Searching and fixing errors:

```
[root@bell ~]# ping -c 5 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=255 time=1.94 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=255 time=1.83 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=255 time=1.88 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=255 time=1.86 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=255 time=1.93 ms

--- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 1.837/1.894/1.945/0.041 ms
```

Figure 3.10. Connectivity testing between ip-pc1 and ip-r1

5.1. When it is noted that there is no connection between the computers and their associated network gateways (routers ip-r1 and ip-r2), or between the routers themselves, the search for errors is performed. This search is carried out following the multi-step plan.

5.2. Checking if the addresses of routers interfaces are set correctly. Attention should be also drawn, whether a "no shutdown" command have been omitted, the abstaining of which leave the interface in shutdown state (Figure 3.11). In order to check the address and state of the interface in the router, run the following command:

```
ip-r1#show ip interface brief
```

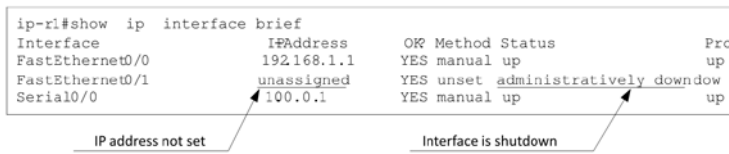


Figure 3.11. Results of show ip interface brief

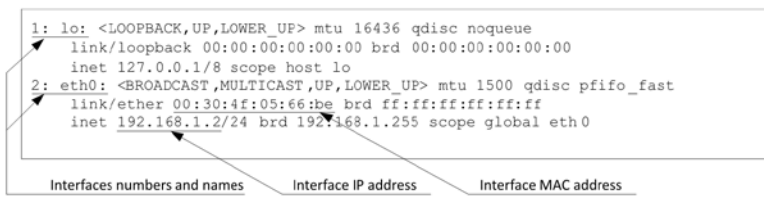


Figure 3.12. Results of ip addr show

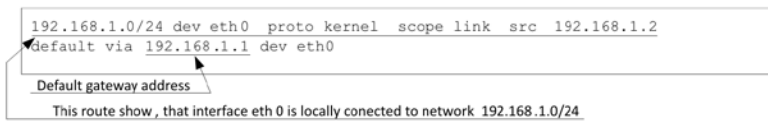


Figure 3.13. Results of ip route show

5.3. In order to check, if the interface addresses in users PC have been set correctly, run the following command (Figure 3.12):

```
[stud@ip-pc2 ~]$ ip addr show
```

5.4. In order to check, if default gateway addresses have been set correctly, run the following command (Figure 3.13):

```
[stud@ip-pc2 ~]$ ip route show
```

Content of Laboratory exercise report:

1. The schemes of the network.
2. Detailed configuration of the devices and connection test results.
3. Conclusions.

Questions:

1. How IPv4 is different from IPv6?
2. Why the class division of addresses is not used anymore?
3. Why it is beneficial to use the address summarization?
4. What is special about the default gateway?
5. What are the main fields of the routing table?
6. When an output interface of the device can be used instead of the next-hop IP address?
7. Divide the following network 172.16.2.0/23 into four equal blocks, write down each of their network and broadcast addresses.
8. Divide the network 172.16.2.0/24 into four blocks as follows: 6, 120, 40, and 21. Write down each of their network and broadcast addresses and the address field, which is used for the addressing network nodes.
9. What is the use of network and broadcast addresses?
10. Calculate the network and broadcast addresses of the following IP address: 10.3.4.5/12, 10.4.2.67/30, 172.16.12.43/28.

4. WLAN TECHNOLOGIES

4.1. IEEE 802.11 standards overview

Currently wireless area local network (WLAN), a network based on IEEE 802.11 family of standards, where information is transmitted by radio waves, is commonly used as access network. Radio waves are unguided media. It is therefore necessary to use additional technological measures so that the noise is separated from data transmission and the transmission of neighboring nodes. IEEE 802.11 architecture defines two types of networks: managed and ad-hoc.

Managed networks use two types of devices: Access Point (AP) and Mobile Station (MS) (Figure 4.1). AP carries the functions of switch and network management, therefore all data exchange between the mobile stations is made only through the AP.

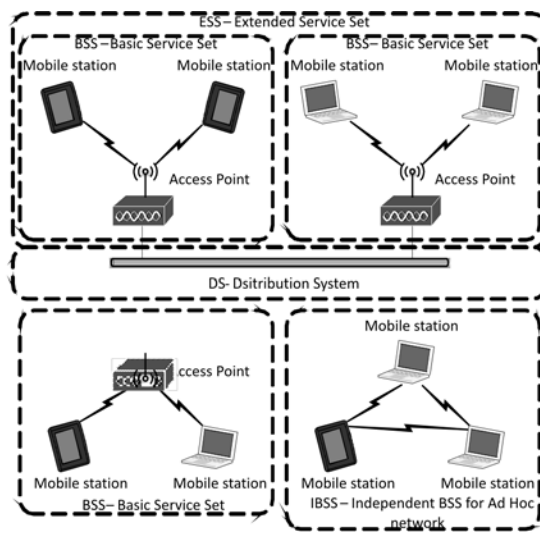


Figure 4.1. WLAN networks architecture

Devices operating in managed mode can be organized into a centralized wireless network. One access point and associated stations in its coverage area compose a basic service set (BSS), and several AP joined form an extended service set (ESA). The latter, connected by any wired or wireless media, is called WLAN core network or the distributed system (DS) (Figure 4.1). All ESS network stations belong to the same virtual MAC network. When one station from AP coverage area switches to the next, the APs transmit data to each other through the DS. When DS operates using the same radio channel that is used by the mobile station, then the system is called wireless distribution system (WDS).

Distributed WLAN networks commonly are referred to as ad-hoc. There are no base stations or access points in such a network. The essential difference between ad-hoc and managed network is that here not only the stations can send and receive the data, but they also broadcast the signals of other stations, when these cannot interact directly. Ad-hoc network topology has the following advantages: high reliability and the ability to organize stochastically.

The basic IEEE 802.11 standard describes the layer of media access control (MAC), its protocols, services and the physical layer (PHY). The basic standard described two physical layers of 2.4 GHz spread spectrum and one infrared physical layer. Later the basic standard was supplemented with new physical layers and some MAC layer enhancements. The main amendments are shown in the Table 4.1.

Table 4.1. IEEE 802.11 amendments

Standard	Description
IEEE 802.11a	New physical layer: 54 Mbit/s, 5 GHz
IEEE 802.11b	Enhancements: addition speed 5.5 and 11 Mbit/s
IEEE 802.11e	Enhancements: QoS features
IEEE 802.11g	New physical layer: 54 Mbit/s, 2.4 GHz
IEEE 802.11i	New security features: WPA
IEEE 802.11n	New physical layer: MIMO technology, up to 600 Mb/s, 2.4 GHz and 5 GHz
IEEE 802.11s	Mesh Networking

4.2. IEEE 802.11 physical layer

IEEE 802.11 standard covers the lower two layers of the network model (Figure 4.2). The physical layer connects the MAC layer to the radio channel. Physical layer performs three functions:

- exchange of frames between the MAC and PHY layers using physical layer convergence procedure (PLCP);
- transmission of data frames using physical medium dependent (PMD) layer;
- carrier registration and transfer of radio channel state to the MAC layer.

All PHY versions have individual PLCP and PMD layers.

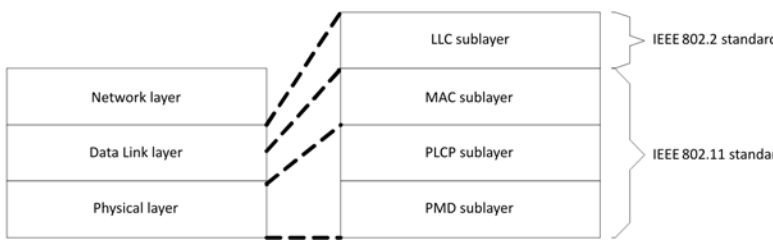


Figure 4.2. IEEE 802.11 reference model

As shown in Table 4.1, there are many physical IEEE 802.11 network standards. Nowadays most often the following versions of physical layer are used: IEEE 802.11a, IEEE 802.11g and IEEE 802.11n.

IEEE 802.11a physical layer uses orthogonal frequency division multiplexing (OFDM) transmission scheme and BPSK, QPSK, 16-QAM and 64 QAM modulations. The required channel width is 16.6 MHz. This PHY operates in the 5 GHz frequency band. In the EU 19 channels are designated for the standard, and their central frequencies are separated from each other by 20 MHz (Figure 4.3). No channels in the 5 GHz are overlapped. In other countries, it is allowed to use a different number of channels. Data transmission rate can vary from 6 to 54 Mb/s. PLCP preamble and header are always sent by 6 Mb/s rate using BPSK modulation. Such transmission principle is used so that there would be a maximum probability to process the received frame header, because the lower transmission rate is, the less signal power receiver requires in order to decode the frame. IEEE 802.11a physical layer frame structure is presented in Figure 4.4. Although the IEEE 802.11a standard allows to transfer 4095 data bytes from the MAC layer (PSDU), but the MAC layer frames themselves (MPDU) cannot be longer than 2346 bytes.

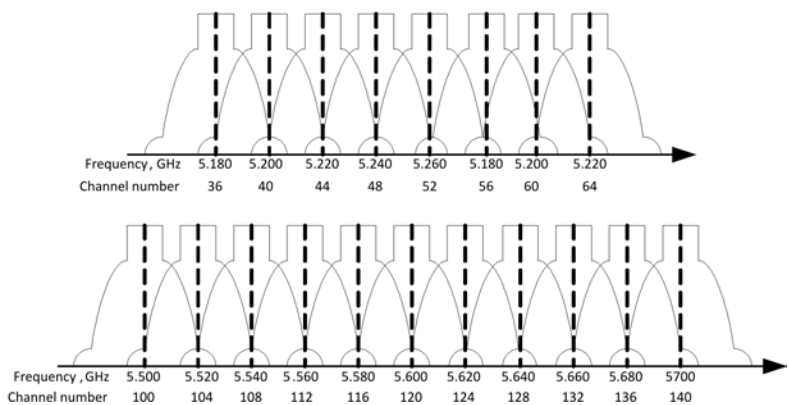


Figure 4.3. IEEE 802.11a channels in EU

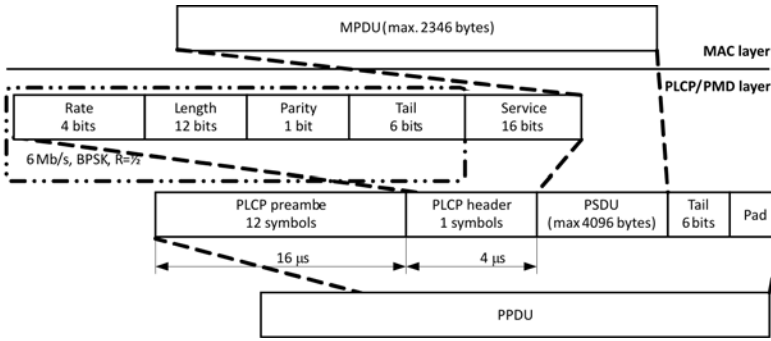


Figure 4.4. Structure of IEEE 802.11a PLCP frame

The physical layer of IEEE 802.11g is designed for 2.4 GHz frequency band. It enables to transfer the data in the rate range up to 54 Mb/s. There are four modulation schemes in this standard: the first two of which are mandatory, and the other two are additional (Figure 4.5).

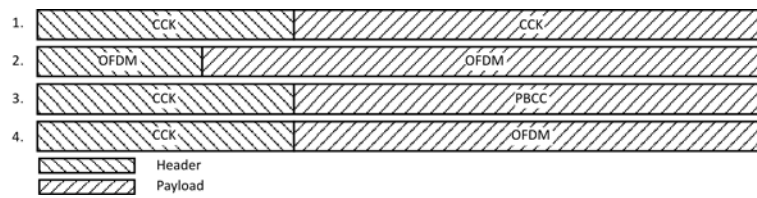


Figure 4.5. Types of IEEE 802.11g frame modulation

DSSS/CCK modulation is an accurate equivalent of the IEEE 802.11b PHY layer. This version of the standard allows IEEE 802.11g standard devices to interact with the IEEE 802.11b devices. The channel width of such modulation is 22 MHz.

OFDM modulation is an accurate equivalent of the IEEE 802.11a PHY layer. It works on 2.4 GHz frequency band and enables the data transmission by the rate up to 54 Mb/s. The channel width of such modulation is 16.25 MHz.

Packet binary convolutional coding (PBCC) modulation allows to additionally transmit the data by the rate of 22 and 33 Mb/s. Preamble, header and the format of the frame are identical to those in the IEEE 802.11b version. This version is not used in practice.

CCK-OFDM modulation is a hybrid that uses preamble and header of the DSSS and OFDM for data transfer. Possible data transfer rate is from 6 to 54 Mb/s. This transmission scheme is used when in the same area devices working on the IEEE 802.11b and IEEE 802.11g standards are used. IEEE 802.11g device, that transmits the header in CCK modulation, allows the devices, which operate only an IEEE 802.11b, to identify the transmission of data and to avoid collisions. The frame structure of such transfer scheme is presented in Figure 4.6.

In the EU ISM band, there are 13 channels in 2.4 GHz frequency band, between which the distance is 5 MHz. Meanwhile, using DSSS, PHY channel covers 22 MHz frequency band. When in the same area there are several independent WLANs, in order to avoid interference, it is recommended to use no more than three (namely, 1, 6, 11) channels, when using in CCK and CCK-OFDM modulation schemes, or not more than four (namely 1, 5, 9, 13) channels, when using the OFDM modulation scheme (Figure 4.7).

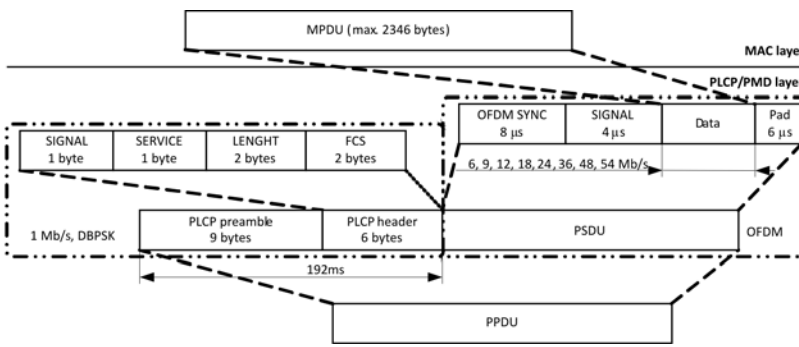


Figure 4.6. IEEE 802.11g CCK-OFDM frame structure

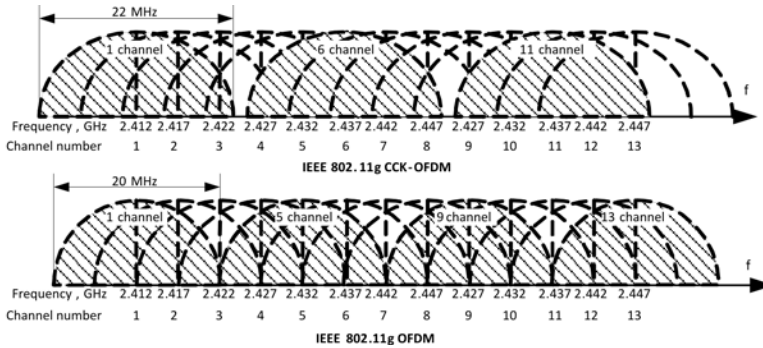


Figure 4.7. Channel width of IEEE 802.11b and IEEE 802.11g

4.3. IEEE 802.11 MAC layer

MAC protocols in WLAN networks are based on carrier sensing and collision avoidance principles. The station, before trying to transmit the data prepared for the transmission, listens to the channel status, i.e., measures the signal power from other stations. If the channel is busy at the time, then the station waits for a random period of time and repeatedly examines the state of the channel. If the channel is idle, then the node transmits its frame. All versions of IEEE 802.11 specification use the same MAC protocol.

The main objective of MAC layer is to control the transmission medium. Due to this layer, each node is able to transmit data and it can expect the data to be forwarded to it. The channel access mechanism is the backbone of all MAC layer. There are several basic access methods, of which the IEEE 802.11 MAC layer uses the following two: carrier sense multiple access/collision avoidance (CSMA/CA) and pooling.

CSMA/CA channel access mechanism is commonly used in wireless LANs operating in the industrial, scientific and medical (ISM) frequency band. The main principles of the access are “lis-

ten before talking" and contentions. CSMA/CA mechanism is an asynchronous message transmission, not oriented to interconnection, giving the best effort service, with no guarantees of conduction or delay.

The carrier senses multiple access / collision detection (CSMA/CD), implemented in the Ethernet standard networks, is the prototype of CSMA/CA. In the access devices of CSMA/CD the sender's receiver operates at the same time when the transmitter is turned on, so it can detect collisions, because the signal strength in the whole medium is approximately equal. Meanwhile, in the wireless connection the transmitter and receiver of one node can only work alternately, therefore is unable to detect collisions.

When the energy of the received signal is below the threshold, the devices of CSMA/CA standard determine that the channel is idle. Since WLAN devices cannot detect the collisions in media, the additional management tool, Network Allocation Vector (NAV), has been introduced. NAV shows how quickly the transmission media becomes idle. The sending node determines the vector of NAV, the receiving node analysis it and therefore does not transmit data for a time given by NAV, even if the energy measurements show that the channel is idle. This in turn helps to avoid collision. NAV is often referred to as a virtual carrier sensing mechanism. Thus, physical and virtual carrier sensing mechanisms are provided in the IEEE 802.11 MAC protocol in order to ensure reliable collision avoidance.

The second MAC task is the management of wireless media. This is done using the following functions:

- distribution coordination function (DCF). The independent basic service set (IBSS) is generated using this function. In this case, nodes exchange information directly, without any additional access points;
- point coordination function (PCF). The basic service set (BSS) is generated using this function, and the nodes exchange the information through access point.

In both DCF and PCF cases the transmitter and receiver in WLAN devices switches between each other according to certain time intervals. There are five time intervals in the IEEE 802.11 MAC standard. Physical layer defines two of them: short interframe space (SIFS) and the slot time t_{tl}. Three additional intervals, namely, distributed interframe space (DIFS), priority interframe space (PIFS) tPIFS ir extended interframe space (EIFS) t_{EIFS}, depend on the previous two. EIFS is used to synchronize NAV after the receipt of damaged frame. By using these time intervals, the data transmission process is controlled and therefore the dispatch of the frame is only possible at the beginning of the interval. The relations between time intervals are presented in Figure 4.8, and their numerical values are shown in Table 4.2.

Table 4.2. Values of time intervals

Standard	t_{SIFS}, μs	t_{DIFS}, μs	t_{tl}, μs	CW_{min}	CW_{max}
IEEE 802.11a	16	34	9	15	1023
IEEE 802.11g	10	50	20	15	1023
IEEE 802.11n	16	34	9	15	1023

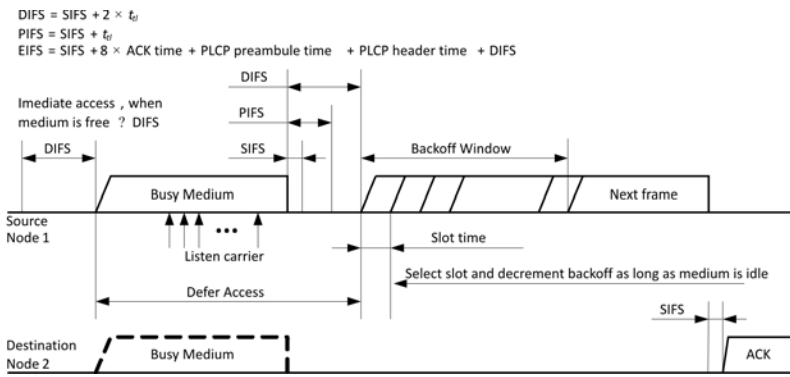


Figure 4.8. IEEE 802.11 interframe times

Coordination distribution function is a basic CSMA/CA function, which coordinates the transmission of any data. Before sending a data frame, the MAC layer checks the channel state by using the virtual and physical carrier sensing mechanisms. If both mechanisms show that the channel is idle for more than a DIFS interval (or EIFS if the received frame was damaged), then start the data transmission. Data transmission protocol consists of two parts:

- data frame, which is forwarded by the sender to the destination;
- acknowledge ACK frame sent by the destination when it receives the undamaged frame.

MAC protocol operates on "stop and wait" principle, i.e., it forwards the next frame for transmission only after the acknowledge frame is received, confirming that the previous frame has reached the destination. The exchange mechanisms of data frame and approval frame are an indivisible unit of the MAC protocol, so the other nodes cannot interrupt this process. When the sender does not receive the approval frame, it re-transmits the same data frame (Figure 4.9).

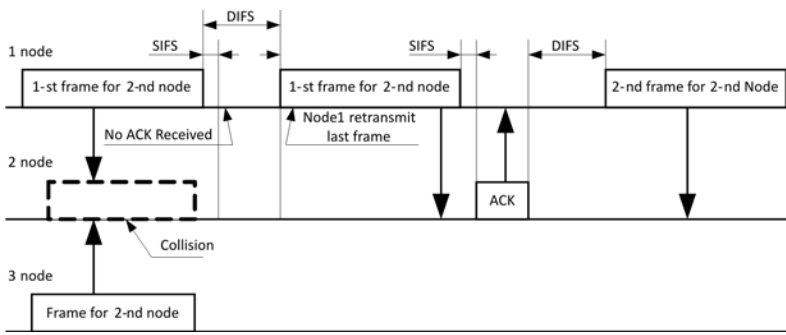


Figure 4.9. Principle of packet retransmission for IEEE 802.11 MAC protocol

If at least one channel sensing mechanism indicates that during DIFS interval the channel is used, then the node waits for the end of the transmission, and then, using the exponential back off algorithm, starts to compete for the media, waiting for a random period of time (Figure 4.10). During retransmission, a random number is chosen. This number identifies the time slot, after which, in the absence of data transmission, the transmission media must be idle (Figure 4.10). The re-sending timer duration t_{CW} is calculated according to the two parameters: the physical layer parameter called slot time t_{tl}, and the contention windows (CW), i.e., a set, from which a random number between 0 and CW is elected for the re-transmission timer. Contention window is doubled each time, if due to the collision the attempt to transmit the data fails. The initial value of CW is defined by the CW_{min} parameter of the standard. This parameter is increased until a maximum value of CW_{max}. Each physical layer of the standard is provided with the individual values of minimum CW_{min} and maximum CW_{max} (Table 4.2).

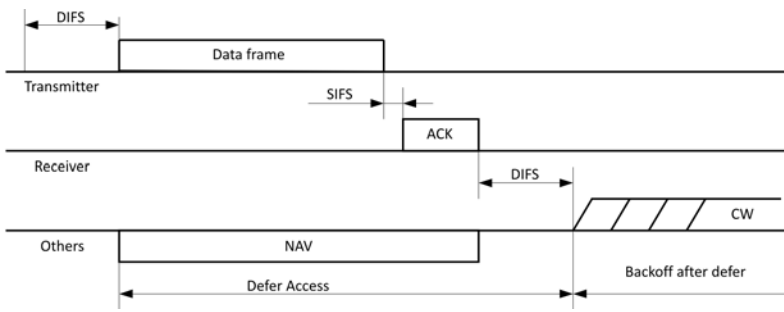


Figure 4.10. Principle of DCF function in the IEEE 802.11 MAC, when RTS/CTS function is switched off

When virtual and physical carrier sensing mechanism indicates that the channel is idle, the value of the re-transmission timer reduces by one unit (Figure 4.11). The node, which has the shortest delay

of contention, wins and has the right to transmit the frame. Other nodes must wait for the next contention period. Contention, as the process, is random and takes place after each transmitted frame, so each node has equal opportunities to transmit its data.

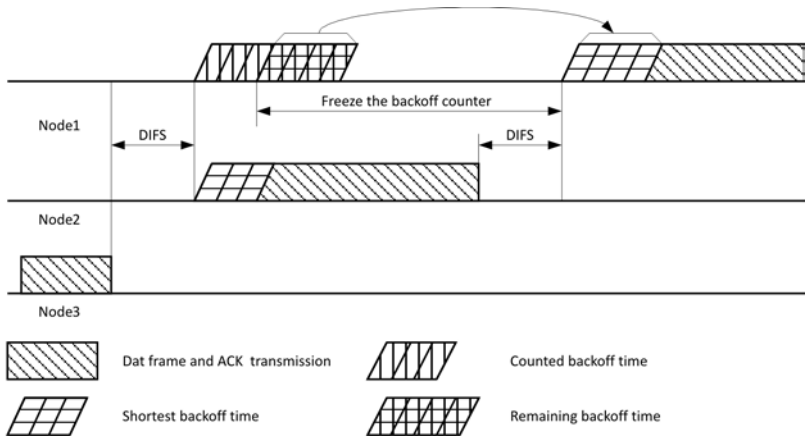


Figure 4.11. Principles of back off counter

When after sending a data frame, the approval frame ACK is not received, the node records the collision. Contention window doubles and the new back off re-transmission time is being elected, so the process is repeated until the frame is finally sent or until the duration of the frame ends.

When because of an excessive distance or signal attenuation not all the nodes can "listen" to each other, the problem of the hidden node appears. When nodes that "do not hear" one another transmit the data at the same time, and when the signal power of the nodes on the receiving side is similar, it results in a collision and the transmitted frames disappear (Figure 4.12). The reason for this problem is carrier sensing mechanism itself, whereby only the sender using local information determines whether the channel is

idle. Meanwhile, the situation in the destination's environment may be different.

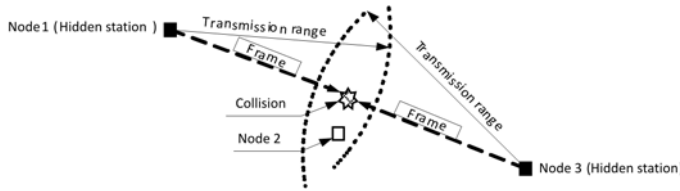


Figure 4.12. Example of hidden nodes

IEEE 802.11 standard has the solution for the problem of hidden node, namely it is the reservation of the channel. For this purpose, the node, which transmits the data, also sends an additional RTS frame. Destination responses to the RTS by a validation frame CTS. The receipt of such a frame indicates that the destination has been received RTS, also is called the "handshake" (Figure 4.13).

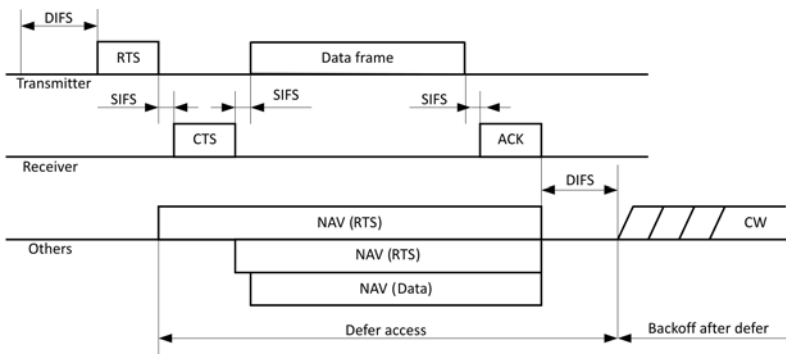


Figure 4.13. Principle of DCF function in the IEEE 802.11 MAC, when RTS/CTS function is switched on

After receiving RTS or CTS frame all other nodes stop sending data. The information in these frames determines the size of NAV vectors.

The collision occurring during the transmission of the RTC/CTS frames causes much less time interference than the collision, which occurs during the transmission of data because the size of RTC/CTS frames is smaller than the one of the data frame. However, the RTC / CTS frame still covers a certain amount of the channel resource; therefore, the data frame size limit has been introduced in IEEE 802.11 standard. RTS/CTS mechanism is used only when the data frame size exceeds the limit, thereby avoiding the rate downward of transmitting the small frames. The process of exchanging four frames (RTS, CTS, data and ACK) is also indivisible, and other nodes cannot stop it.

The pooling MAC protocol in the IEEE 802.11 PCF function is used as the channel access mechanism. It is an intermediate option between the CSMA/CA and TDMA as the access point has the full control of the channel, but the frame size is not fixed. Before sending the data through the AP, the node must register at the access point. The AP periodically sends a specific pool frame to each such node. With this frame, the node receives the data addressed to it as well as the permission to occupy the channel for data transmission. Due to the sharing the channel in time, PCF and DCF functions may operate on the same network, therefore the logon procedure is done during the contention period (CP) (Figure 4.14). After the CP phase, the access point in order to occupy the channel has to wait for time interval PIFS, which is shorter than DIFS, then send a beacon frame (B), which signals to the other nodes that the AP switches the channel into the contention free period.

Currently, only a small number of commercially available devices have this function. All data in the networks that support AP is provided by a DCF function.

The structure of IEEE 802.11a, IEEE 802.11g and IEEE 802.11n frames are provided in Figure 4.15. It demonstrates that the MPDU frame consists of a MAC header, MAC service data unit (MSDU) and frame check sequence (FSC). The following are the main types of frames:

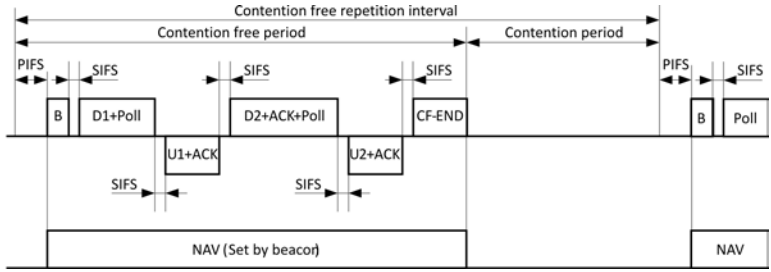


Figure 4.14. PCF function

1. data frame to transmit data from higher levels;
2. control frames for the transmission of the beacon frame to the nodes association with the AP, authentication, etc.;
3. official frames RTS, CTS and ACK.

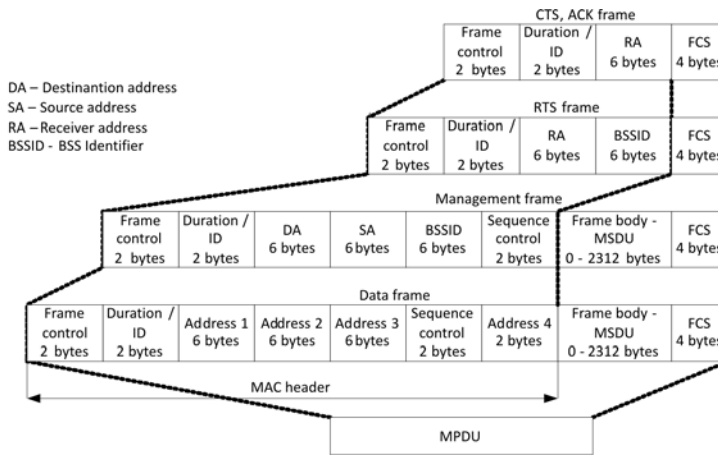


Figure 4.15. IEEE 802.11 MAC frame structure

Ethernet header always has only two addresses: the sender and the destination, meanwhile in the header the 802.11 frame in total four different addresses can be transmitted. The combinations of

addresses used in data frame are presented in Table 4.3. Depending on the situation, i.e., what is the type of wireless network and what sends the data, three or four addresses are used. In case of IBSS and BSS, the basic service set identification (BSSID) is specified next to the sender's and destination's addresses. In BSS network it is the AP's MAC address, in IBSS network it is a randomly generated MAC address of the local type and in WDS network all four addresses are used. The sending AP and Receiving AP MAC address are indicated next to the addresses of the sender and destination. The address fields of management, RTS, CTS and ACK frames are strictly defined, their values are given in Figure 4.15.

Table 4.3. IEEE 802.11 data frame address field contents

Situation	Address 1	Address 2	Address 3	Address 4
IBSS network	Destination	Source	BSSID	N/A
BSS network, from AP	Destination	BSSID	Source	N/A
BSS network, to AP	BSSID	Source	Destination	N/A
WDS network	Receiving AP	Sending AP	Destination	Source

In contrast to Ethernet, IEEE 802.11 standard uses IEEE 802.2 logical link control (LLC) protocol for the transmission of the higher layer protocols. An example of IP packet transmission in the IEEE 802.11 network using IEEE 802.2 LLC encapsulation mechanism is given in Figure 4.16, which shows that an 8-byte length header is added to the head of each IP packet before the transmission. Currently, there are two different methods for LLC encapsulation for data transmission in IEEE 802 networks. One of them is described in RFC 1042, and the other is the IEEE standard 802.1h. Both of these protocols are very similar because they are of equal

length, and originate from the subnetwork access protocol (SNAP), but for the transmission of IP packets the RFC 1042 protocol is used more often. The number of the higher layer protocol, which is used for the transmission of data packets, is being set from the internet engineering task force (IETF) list. This number is indicated in the "type" field of both protocols mention in the above.

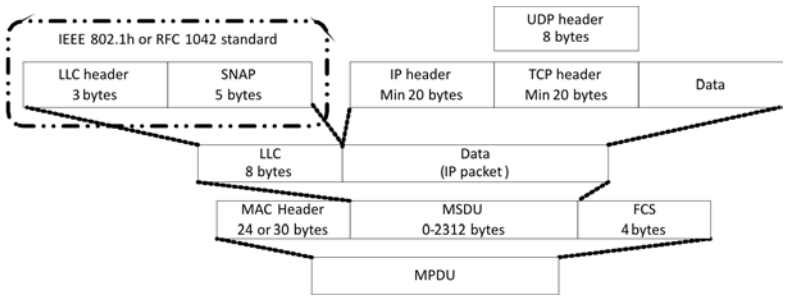


Figure 4.16. Incapsulation of IP packet into IEEE 802.11 frame

Basic DCF and PCF IEEE 802.11 MAC functions are unable to guarantee the quality of service (QoS). This problem is dealt with the IEEE 802.11e standard, which describes the additional hybrid coordination function (HCF), which provides two channel access methods: HCF controlled channel access (HCCA) and enhanced distributed channel access (EDCA). Both of these methods describe the prioritization of frames according to the stream category. EDCA is the most widely used method, also known as wireless multimedia extensions (WME) or Wi-Fi Multimedia (WMM). Four access categories are provided in this method, each of which has its own dispatch queue, which differ from one another by the back off settings and arbitration interframe space number (AIFSN).

Table 4.4. Default EDCA Parameters

Access Categories	CW_{min}	CW_{max}	AIFSN
Background (AC_BK)	31	1023	7
Best Effort (AC_BE)	31	1023	3
Video (AC_VI)	15	31	2
Voice (AC_VO)	7	15	2

EDCA replaces the DIFS time interval, which defines how long it is necessary to wait until it will be possible to compete for the channel, into the arbitration interframe space (AIFS) time (Table 4.4). AIFS time period depends on the AIFSN, SIFS and the physical standard t_{tl}. (Figure 4.17). When the stream is in the highest category, the AIFS value is identical to the DIFS period. When category declines, AIFS time interval increases, hence the nodes have to wait for a longer period, before moving on to the contention period. In addition, the parameters of the back off period in the higher categories are smaller, so the chance that the node wins the back off contention increases (Figure 4.17).

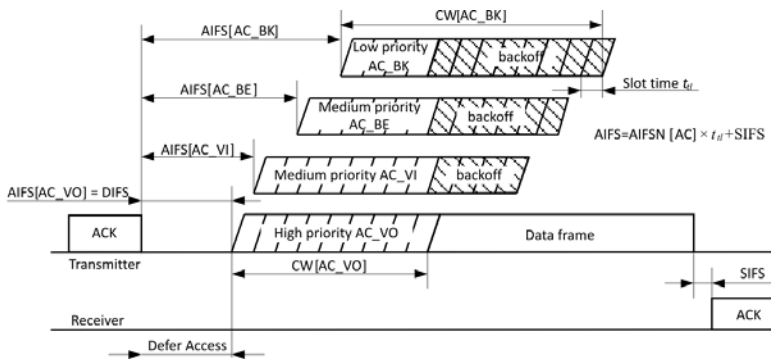


Figure 4.17. EDCA contention-based medium access

Table 4.5. DSCP, 802.1p and WMM access category mapping

DSCP priority range	802.1p Priority	WMM Access Category
8–31	1	Background (AC_BK)
0–7	0	Best Effort (AC_BE)
24–31	2	Best Effort (AC_BE)
32–39	3	Video (AC_VI)
40–47	4	Video (AC_VI)
48–55	5	Voice (AC_VO)
56–63	6	Voice (AC_VO)
	7	Voice (AC_VO)

IEEE 802.11 standard does not describe how a particular user's data stream can be assigned to a particular category; therefore currently it is done based on the content of IEEE 802.1p and DSCP IP protocol headers fields (Table 4.5).

4.4. Laboratory exercise. Analysis of WLAN technologies

Goal of Exercise: get acquainted with WLAN technology.

Tasks of Exercise: get acquainted with WLAN technology, measure its performance at various frequencies and frame sizes.

Equipment: two PC with Linux OS with madwifi WLAN driver, WLAN card with Atheros chipset, any IEEE 802.11 a/g band Access point with dd-wrt firmware.

Workflow:

1. Preparation:

   1.1. Run two terminal programs on the PC. Terminal program can be either gnome-terminal or xterm. In case of Fedora Core Gnome environment run Applications → Accessories → GTKTerm.

   1.2. In order to set the access point (AP) configuration as default, connect to the 192.168.1.1 address in the web browser

program. For that type username: admin, password: cisco. In the loaded page go to: Administration → Backup → Restore Configurations → Browse and choose a file: wlan_ap1_default.cfg, which is in the following catalogue: /home/stud. Press Restore. Once these steps are completed, the AP reboots automatically.

2. Examination of IEEE 802.11a standard data throughput. The survey is conducted in accordance with the network diagram shown in Figure 4.18. Computer wlan-PC1 is wirelessly connected to wlan-ap1 access point that runs dd-wrt firmware; and through this AP, using iperf software, a continuous data stream is transmitted to the computer pingsrv1.

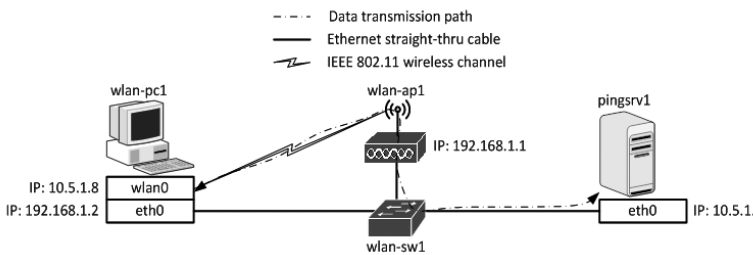


Figure 4.18. Network diagram

2.1. To figure out how many and at what corresponding frequency WLAN networks operate in the area operate and according to the investigated information choose IEEE 802.11a and IEEE 802.11g wireless channels in the explored WLAN. The selection needs to be done in such a way that the examined network interferes with networks already existing as little as possible. The surrounding WLAN networks could be observed easily using Kismet software and running the following commands:

```
[stud@wlan-pc1 ~]$ sudo wlanconfig wlan0 destroy

[stud@wlan-pc1 ~]$ sudo kismet
```

Choose Windows → Channel details in Kismet program. The program then shows the detected signal and noise levels in wireless card of each WLANs channel. It as well gives the information on how many frames per second are created by the surrounding WLAN networks. The example is given in Figure 4.19.

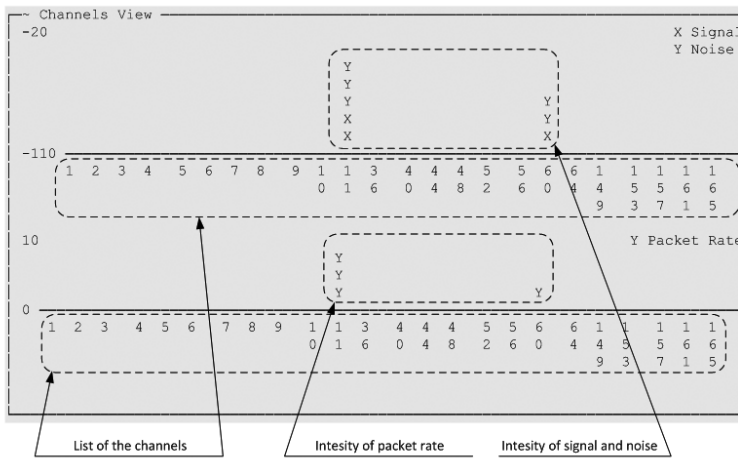


Figure 4.19. Explanation of Kismet channels view data

IEEE 802.11a standard uses 19 channels from number 36 to 165, and all of them are not overlapped. Therefore, for experimental purposes, an empty channel must be chosen, or in other words, the wireless card should not see any operating channel device.

IEEE 802.11g uses channels from number 1 to 13, but only 3 (1, 6, 11 or 13) channels are not overlapped, or 4 (1, 5, 9, 13) in

case of OFDM modulation (Figure 4.7). Therefore, the channel chosen for experimental purposes must not only be empty but also away from other operating channels (the nearest one could be 3 channels away). If it is not possible to find an empty channel or if it is too close to other operating channels, then a channel with the lowest intensity of packet rate should be chosen (Figure 4.19).

Enter the numbers of chosen channels into Table 4.6. Exit the Kismet program. In order to set WLAN card back to station mode, run the following command:

```
[stud@wlan-pc1 ~]$ sudo wlanconfig wlan0 create
wlandev wifi0 wlanmode sta
```

Table 4.6. Selected WLAN channels

WLAN standard	Channel
IEEE 802.11a	
IEEE 802.11g	

2.2. In order to set the AP physical standard into EEE 802.11a and to change the name of the network in service into wlan-a, go to the AP web page and choose: Wireless → Basic Settings. After that in "Wireless Mode" choose AP, in "Wireless Network Mode" choose "A Only", and in "Wireless Network name (SSID)" choose wlan-a. After this transfiguration, press "Apply Settings", and then "Save". In the "Wireless channel" field set an idle channel, which have been found in task 2.1. After this transfiguration yet again, do not forget to press "Apply Settings", and then "Save".

2.3. Configure wireless connection port wlan0 in wlan-ak1 for work with IEEE 802.11a standard's AP:

In order to assign the name of the wireless network, run the following command:

```
[stud@wlan-pc1 ~]$ iwconfig wlan0 essid wlan-a
```

In order to assign the IP address, run the following command:

```
[stud@wlan-pc1 ~]$ ip[6]addr add 10.5.1.8/24 brd +
dev wlan0
```

2.3.1. In order to activate the port, run the following command:

```
[stud@wlan-pc1 ~]$ ip link set wlan0 up
```

2.3.2. In order to make sure the wlan-PC1 is connected to the AP, run the following command:

```
[stud@wlan-pc1 ~]$ iwconfig wlan0
```

If the parameter of "Access Point" shows the MAC address, this means that wlan-pc1 is registered to access point of wlan-ap1. The functioning of iwconfig command is explained in Figure 4.20.

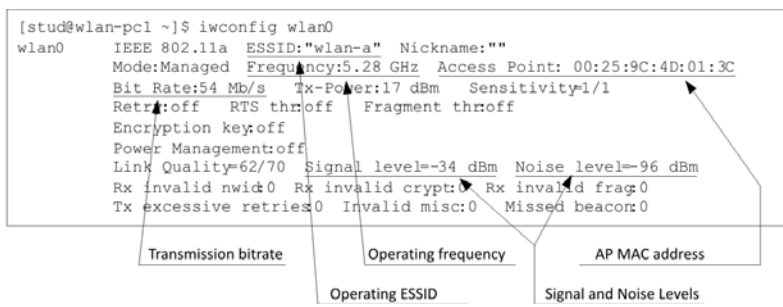


Figure 4.20. Results of iwconfig command

2.3.3. In order to make sure that wlan-ak1 has a connection with pingsrv1 PC, run the following ping command:

[6] IP address could also be assigned with ifconfig command.

```
[stud@wlan-pc1 ~]$ ping 10.5.1.1
```

2.3.4. Measure the network throughput between the wlan-ak1 and pingsrv1 and fill in Table 4.7.

Note: after measuring the physical data rate, measure the throughput with all MTU values and only then change the physical data rate into another value. This will help you save the time.

2.3.5. Change the AP physical data rate in the web page: "Wireless → Advanced Settings". The data rate is defined by "Transmission Fixed Rate" parameter. After this change do not forget to press "Apply Settings" and then "Save".

2.3.6. In order to set the physical data rate of wlan-pc1wlan0 interface, run the following command:

```
[stud@wlan-pc1 ~]$ iwconfig wlan0 rate XM
```

When *X* is the transfer rate in Mb/s (Table 4.7).

2.3.7. In order to measure the throughput, run the following command:

```
[stud@wlan-pc1 ~]$ iperf –t 30 –c 10.5.1.1 -mM
```

When *M* is the parameter from Table 4.7.

Fill in Table 4.7 with the results. The coefficient M, rather than MTU, is followed by because of the specificity of iperf command performance: MTU = M – 40. The results of iperf command are explained in Figure 4.21.

2.3.8. Repeat the measurements with all data transmission and MTU values taken from Table 4.7 (under 0–0 above).

3. Examination of IEEE 802.11g standard data throughput. The examination diagram is identical as in the IEEE 802.11a standard case.

   3.1. In order to set the AP physical standard into IEEE 802.11g and to change the name of the network in service into wlan-g, go to the AP web page and choose: Wireless → Basic Settings.

After that in “Wireless Mode” choose AP, in “Wireless Network Mode” choose “G Only”, and in “Wireless Network name (SSID)” choose wlan-g. After this transfiguration, press “Apply Settings”, and then “Save”. In the “Wireless channel” field set an idle channel, which have been found in task 2.1. After this transfiguration yet again, do not forget to press “Apply Settings”, and then “Save”.

Table 4.7. Results of throughput measurements of IEEE 802.11a standard

MTU size	M parameter for iperf	Transmission fixed rate, Mb/s		
		6	36	54
128	88			
256	216			
1024	984			
1500	1460			

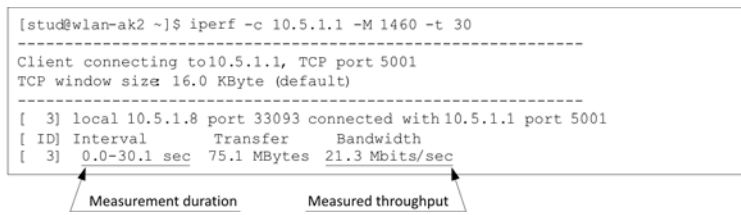


Figure 4.21. Results of iperf command

3.2. Configure wireless connection port wlan0 in wlan-ak1 for work with IEEE 802.11g standard’s AP:
In order to assign the name of the wireless network, run the following command:

```
[stud@wlan-pc1 ~]$ iwconfig wlan0 essid wlan-g
```

3.2.1. Check if wlan-PC1 PC is connected to the AP, according to the tasks 0 and 2.4.

3.3. Measure the network throughput between the wlan-ak1 and pingsrv1 and fill in Table 4.8. The measurement procedure is identical to the case of IEEE 802.11a standard (0–0 tasks).

Table 4.8. Results of throughput measurements of IEEE 802.11g standard

MTU size	M parameter for iperf	Transmission fixed rate, Mb/s			
		1	6	36	54
128	88				
256	216				
1024	984				
1500	1460				

Content of Laboratory exercise report:

1. Provide the computer network diagrams that were used during the examination.
2. Provide the measurement results.
3. Provide the figures drawn according to the measurement result.
4. Answer the question and provide the conclusions.

Questions:

1. What kind of data throughput is regulated by 802.11a and 802.11g standards?
2. What are the main functions of AP?
3. Why throughput regulated by standards differs from actually measured data?
4. What is the use of the IFS intervals?
5. What channel control algorithm is used in IEEE 802.11 networks?
6. What channel access algorithm is used in IEEE 802.11 networks?
7. What is the difference between the ESSID and BSSID?
8. How does EDCA algorithm work?
9. How does RTS/CTS mechanism work?
10. What is the difference between DCF and PCF functions?

5. SPANNING TREE PROTOCOL

5.1. Principles of Ethernet switch working algorithm

Currently Ethernet switches are used for frame switching in the local area networks (LAN). Previously Ethernet hubs carried out this function. Switches have many advantages against the hub: faster, safer and the absence of collision, etc. Ethernet switch is also called Layer 2 or L2 switch because of its operation on the second layer, i.e., Data Link Layer.

These devices perform switching in accordance with the destination's address in the Ethernet frame header. Upon the receipt of an Ethernet frame switch checks the address table to find out if it is aware of destination's address and to which switch port the destination is connected. If the destination is found in the address table, then the frame is transmitted only to that port, to which the destination is connected. If the destination's address is not found in the switching table, then the frame is sent to all switch ports except the one from which the frame was received. If the destination's address is broadcast or multicast address, the frame is also sent to all ports. Before forwarding the frame the switch makes an entry in the switching table. It enrolls the sender's address, which is in the frame header, and the number of port, through which it received the considered frame. Hereby, if the frame is addressed to the sender, the switch will transmit the frame only to the port to which it is connected. In this way the switching table is filled in by frame sender's address automatically.

Ethernet switch can operate correctly only when there is no transfer loops, i.e., when there is only one transmission path between the transmitter and the receiver. Loop is potentially dangerous for the switch network, as it creates broadcast (or multicast) storms and the address table no convergence. Loops in the network may

be caused by a variety of reasons, but mostly due to human factor, i.e., combining switches in the wrong way. Loops are also specially created when trying to create duplication, that is, when one switch fails, the action is taken over by another switch. Figure 5.1 shows a typical Enterprise network, and how the loop can be deliberately used to create redundancy.

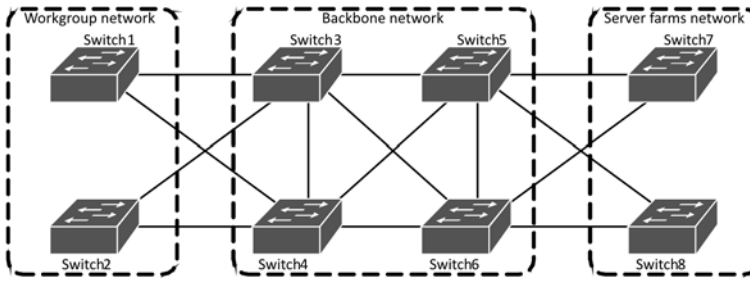


Figure 5.1. Example of redundancy network

5.2. Broadcast loop

Frame with broadcast as destination address and Ethernet switch loop can be a disastrous combination for the network. Let's consider the model given in Figure 5.2. In here the two network nodes, connected to different LAN segments, are combined with two Ethernet switches.

Suppose that Node1 sends a broadcast frame to the MAC address FF-FF-FF-FF-FF-FF. This action in Figure 5.2 is marked as step "1". Frame travels to both Switch1 and Switch2. This is the step number "2". The first switch, acting according to the switching algorithm, transmits the frame with a broadcast address to all its ports. This is the step number "3". Once again, the frame travels to all the nodes in the lower segment of the Ethernet, and to the second switch as well. This is the step number "4". During the 2nd step

the second switch accepted the frame, and passes it forward to all of its ports (3' step), so in the end the frame is placed in the lower segment of the network again. Thus, during the steps 3 and 3' there are already two identical frames in the lower segment. During the 4th step the second switch received a frame from the first switch in the lower segment, and must forwarded it to the upper segment, because, as already mentioned, the frames with the broadcast address in their destination's address field are forwarded to all ports. This is the step number "5". The first switch received the frame from the second switch in the upper segment, and sends it back to the lower segment, i.e., so cycle repeats. For the sake of simplicity, the Figure 5.2 does not show similar steps from 4th to 7th with the frame sent by the second switch to the lower segment. Hence such a network has two frame loops rotating in the opposite directions.

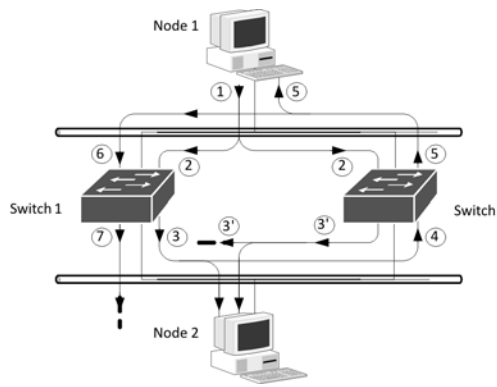


Figure 5.2. Example of broadcast loop

Loops create huge network congestion. This in turn makes it impossible for the switches to handle other frames sent by the network nodes and the computers that are in the network and that receive the frames created by broadcast loops, waste lots of resources for the processing; and since all the CPU resources are used for

processing of frames, it may be difficult to even move the computer mouse.

Loops resulting in the Ethernet networks are more dangerous than routing loops, which form in IP networks. This is because of the Ethernet frame structure (Figure 5.3). Ethernet frame consists of only source and destination fields, an identifier of higher layer type and the CRC. Meanwhile, the IP header contains a specific Time To Live (TTL) field, which is reduced each time the IP packet is processed by the router. When the TTL equals 0, the packet is discarded. If a loop occurs in the IP network, during the reduction of the TTL field, the packet circulates only for a certain period of time, then it is still eliminated, thus this enables to avoid the eternal cycle. Meanwhile, the Ethernet frame does not have the TTL field, so there are no tools that might protect against loops. Therefore, if a loop occurs in the Ethernet network, it continues without stopping until the switch is turned off, or until it is terminated physically, i.e., the cable, which formed a loop, is pulled out.

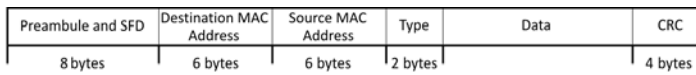


Figure 5.3. Structure of Ethernet frame

5.3. Bridge Table Corruption

In the event of the loop in the Ethernet network, even the targeted transmission frames can circulate forever (Figure 5.4).

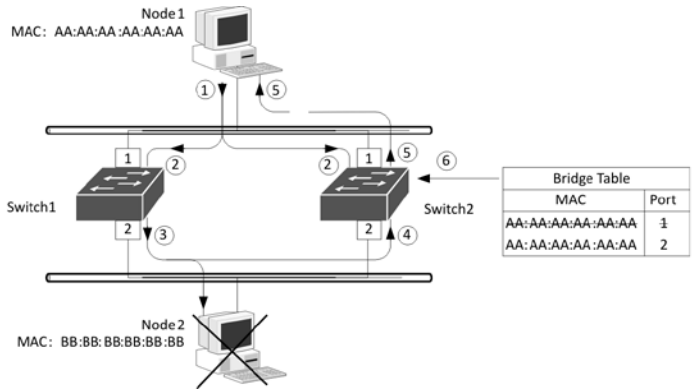


Figure 5.4. Example of corrupt bridging tables

Suppose, Node1 sends a unicast frame to Node2. However, Node2 is disabled and does not send data. Therefore, there is no information about Node2 in the switch bridge table. As in the previous example, frame travels to the first port of both switches (2nd step). To simplify the analysis, let's discuss the situation from the point of view of switch 1. Since Switch 1 is not aware of Node2 connection, it transmits the received frame to all ports, i.e., in this case to port2 (3rd step). In the 4th step Switch 2 receives the frame from port2. Two bad things happen then:

1. Switch 2 forwards the frame to port1, because it has no information about Node2 connection (5th step). This creates a feedback loop, and stops the network.
2. Switch 2 changes the number of Node1's port from 1 to 2, because it just received a frame sent by Node1. A new entry is incorrect, because Node1 is accessible through port1.

By the same principle the loop in the opposite direction is formed. Therefore, the entry about Node1's MAC address in the switch bridge table starts flipping between port1 and port2. Thus, when a loop and a unicast type of frame come together, not only forms a unicast type of frame storm, but it also makes switching tables unstable.

5.4. Spanning tree protocol

Since the loops in the network reduces its performance and reliability, a special protocol, called spanning tree protocol (STP), was designed. This protocol allows the switches[7] to find physical loops in the network and eliminate them. This is done by creating a structure of loop-free logical tree with the so-called tree trunk and branches. The structure of the tree is chosen, because the tree trunk and its branches as they grow never do loops, i.e., in the tree, there is only one path from one branch to another. Therefore, if the network would be constructed on the principle of a tree, it would not have loops. STP protocol is described in the IEEE 802.1D standard.

STP algorithm is based on Bridge Protocol Data Units (BPDU) management frames exchange between the switches. The logical tree topology is created according to these management frames. The tree is constructed in three main steps:

- Elect one root switch. Root switch is like a tree trunk, it can be the only one in the network, as the tree has only one trunk. From this switch a loop-free path to each of the remaining switches is being created.
- Elect root ports. Root ports are the ports, which are in the shortest distance to the root switch.
- Elect designated ports. These are the ports that serve a particular LAN segment. In the root switch all the ports are only designated.

After the elections, the redundant switch ports that form loops are disabled by software. BPDU frame structure and main fields are shown in Figure 5.5. By default BPDU are sent every 2 seconds.

[7] The definitions of bridge and switch are used in the text. Basically these are the devices that perform the identical functions, only bridge has less ports. In the description of STP protocol bridge is used because of the historical reasons.

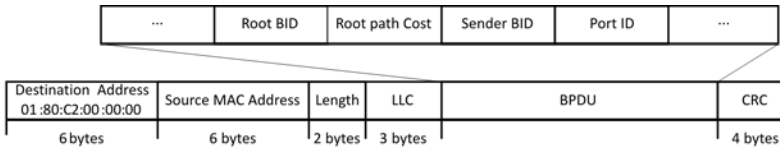


Figure 5.5. BPDU frame structure

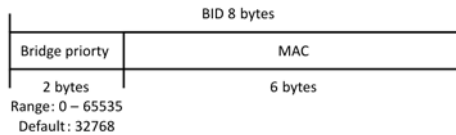


Figure 5.6. BID structure

In order for switches to unambiguously identify one another, BPDU frame uses Bridge ID (BID). BID is an 8-byte field, which consists of two parts: bridge priority and the bridge MAC address (Figure 5.6). Since the MAC address is always unique, therefore the BID value is always unique as well. Meanwhile bridge priority value is usually left to the default and is equivalent to 32768. The priority value is used when it is desired that one particular switch acquires a higher priority than other switches.

In the STP protocol switches use path cost in order to assess what kind of speed ports the switches are interconnected. The higher the speed, the lower is the number of path cost. This is necessary when forming a logical tree topology to take into account the speed of the ports and to use those ports, which have the maximum speed. The list of path cost used currently for Ethernet networks is given in Table 5.1.

Table 5.1. Path cost of STP protocol

Data rate	STP Path Cost
10 Mbit/s	100
100 Mbit/s	19
1 Gbit/s	4
10 Gbit/s	2

It must be underlined that the value of both the BID and path cost in the STP protocol is better, as it is lower, i.e., always wins the switch or port, which has lower values for these parameters.

As it was already mentioned, the STP tree is created by adopting three basic steps: the elections of root switch, root port and designated port. After receiving a BPDU frame switch carries out its analysis in a certain order, which is called a four-step rule:

- Lowest Root BID
- Lowest Path Cost to Root Bridge
- Lowest Sender BID
- Lowest Port ID

The root switch is elected by the first rule, and root port along with designated port is elected by all of the remaining.

During the election of root switch BID BPDU Root field is being analyzed, thus the rule is called the Lowest root BID. This field shows the switch, which now becomes the root. When the switches are turned on, they do not contain information which of them is the root. Therefore each switch assigns itself as the root by generating BPDU frames, in which in the Root BID field is their BID value. Upon receipt of BPDU from another switch the switch in question compares the Root BID field with its own. If the received Root BID is higher, then the switch, which received the BPDU, remains the root. So it discards the received BPDU frame, and instead sends the newly generated BPDU frame where in the Root BID field it enters its own BID. If the received Root BID is lower, then the switch cannot be the root switch, therefore it no longer generates the BPDU

and only transfers the incoming BPDU from other switches. Hereby only one switch remains in the network, the one that has the lowest Root BID and generates a BPDU frames, while all other switches just forward them.

Each switch that is not considered as the root has one root port through which it communicates with the root switch. Root port in the BPDU field is determined by the Root Path Cost field. Ordinary switches do not generate BPDU; they only forward the control frames of root switch. When the ordinary switch receives BPDU, it adds path cost of the port from which it received that particular BPDU to the value of the received BPDU's Root Path Cost field. Switch can receive BPDU from more than one port. The root port becomes the port from which the BPDU with the lowest Root Path Cost value after all aggregation was received. It is possible when different ports transmit BPDU with the same Root path cost value. In this case in order to elect the root port the lowest Sender BID field of BPDU is looked for. This field indicates BID of the switch, which sent / forwarded BPDU. And if this field is the same in several BPDU, then the Port ID field is analysed. This field indicates port ID, through which the BPDU was sent. The port ID values in the switches are different, so it will certainly be possible to elected root port then.

The last step, which completes the construction of STP tree, is the election of designated port. The designated port is a port that leads a way from the root. This type of port serves a particular LAN segment, but is not the root port. Since only one designated port can be in the LAN segment, therefore the election of a designated port is held. In these elections, the following BPDU fields are used: Root Path Cost, Sender ID and Port BID. LAN segment is served by the port, which sent a BPDU frame with the lowest Root Path cost field value. If values of several BPDU Root Path Cost fields are equal, then the value of Sender BID is considered; and if it is also the same, then the Port ID are compared. When the designated port is

elected the other ports, which lost the elections (in the same or other switch) become non designed ports. They are switched off by software to the blocked state. It must be noted that the designated port is elected for every LAN segment, and root port for each switch. Therefore, ordinary switch must always have only one root port, and it may have a few designated ports, or not even one. All ports for root switch are designated.

If the switch does not receive a BPDU frame through root port over some time (by default 20 seconds), it begins to generate BPDU frames itself; the other switches then recreate the STP tree anew. The same happens when a new switch is switched on in the network. Then at the start it also generates BPDU frames and consequently this may completely change the structure of STP logical tree. Therefore, if unwilling to change STP tree root switch after an entry of the new switch, the Priority value of BID field in the former is reduced.

Table 5.2. STP states

State	Action
Forwarding	Sending, receiving data, normal operation
Learning	Building bridging table
Listening	Building active topology
Blocking	Receiving and analysing BPDU
Disabled	Administratively down

After the switches have classified their ports into root designated and non-designated, it becomes rather easy to create a loop-free topology: the root port and designated ports forward message stream, and non-designated ports block it. Although the forwarding and blocking are the two most often seen STP states in a stable network, Table 5.2 shows that there are actually five states.

The diagram of switch's STP state transition is shown in Figure 5.8. Disabled state allows network administrators to manually dis-

able the port. After the initialization ports switch into the blocking state, in which they can only receive BPDU frames. Lots of events such as when the switch make the assumption that it is the root; immediately after the switch is turned on or non-receipt of BPDU frame over a period of time can cause the switch port to switch into listening state. In this state no user data is transmitted. The port sends and receives only BPDU, as it seeks to create a loop-free topology. For this a root switch, a root port and designated ports are elected. Ports that do not win become the non-designated ports and return to the blocking state. After fifteen seconds, that is a standard time value, ports, which become the designated port or root port, transfer into learning state. This is another period of fifteen seconds, during which the switch still does not transmit user data frames. Instead, the switch fills in its switching table. At the beginning of data transmission the learning state reduces the transmission of frames through all the ports. If the port remains the root port or designated root at the end of learning state period, it is switched into the forwarding state. In this state finally sending and receiving of data begins.

5.5. STP protocol operation example

Let us examine the network diagram shown in the Figure 5.8. The network is composed of four switches that are connected by 1 Gb/s or 100 Mb/s connection lines. Switches are connected in such a way that even two loops have been formed. Each switch has been appointed to the unreal MAC address. The priorities of all the switches are the same and they equal to 32,768.

When the switches are turned on, they look for the switch, which generates a BPDU with the lowest bridge identifier, and thus elect one root switch. As it was already discussed, BID is an 8-byte identifier that is composed of two subfields: the switch priority and MAC address. Examination of Figure 5.8 shows that the BID of

Switch1 is 32768.AA:AA:AA:AA:AA:AA, the BID of Switch2 is 32768.BB:BB:BB:BB:BB:BB, the BID of Switch3 is 32768. CC:CC:CC:CC:CC:CC, and the BID of Switch4 is 32768.DD:DD:DD:DD:DD:DD. Therefore, the Switch1 becomes the root switch, because its BID is the lowest.

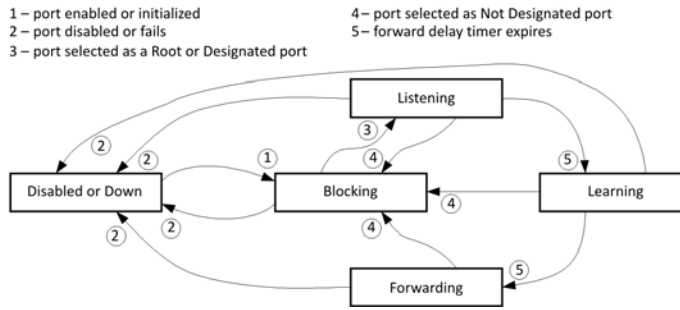


Figure 5.7. STP port states transition

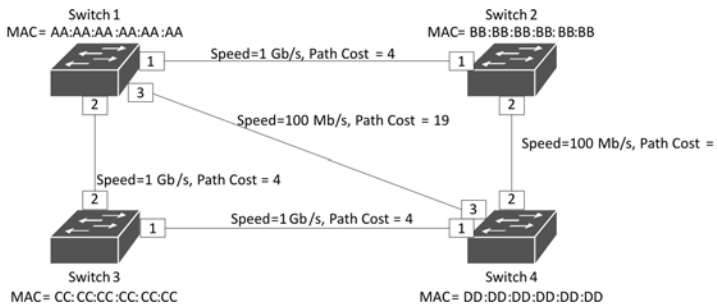


Figure 5.8. Network diagram of analyzed network

During the second step the root ports are being elected. The root port is the port that is closest to the root switch. Each ordinary switch must elect one root port. As discussed previously, switches use the concept of path cost. This value is calculated by summing up the cost from all the ports to the root switch. The calculation of

path cost to the root switch and the election of root port are shown in Figure 5.9.

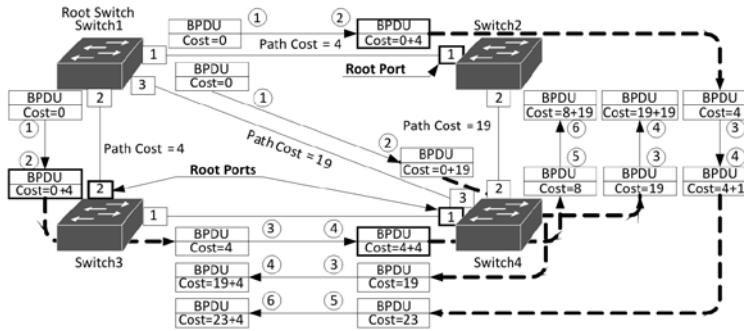


Figure 5.9. Root port election

Switch1 as a root switch sends BPDU in which the path cost to the root switch is equal to 0 (1st step). These BPDU are received by all the remaining switches, because they all have a direct channel of communication with the root switch. After receiving BPDU the switches adds the path cost of the port through which BPDU was received to the Root Path cost: Switch 2 and Switch3 add 4, and Switch4 adds 19 (2nd step). Then Switch2, Switch3 and Switch4 transmit the BPDU frame with the increased value of the Root Path cost to all the remaining ports, except for the one through which BPDU frame was received (3rd step). After receiving these frames, the switches once again add the path cost of the port through which BPDU was received to the Root Path cost (4th step). This is followed by another cycle of BPDU frame transmission and the aggregation of Root Path cost (5th, 6th steps). BPDU frames, which return to the root switch, are not shown in Figure 5.9, because they are ignored by root switch. At the end of the cycle each switch evaluates all the received BPDU frames and searches for the one which after the aggregation has the lowest Root Path Cost value. Therefore, root

port of Switch2 is port 1 (in Figure 5.9 all root ports are shown in bold line), since through it was received BPDU frame with the Root Path cost value equal to 4, others BPDU frames had a higher Root Path cost value (27 and 38 respectively from Switch3 and Switch4). The frame with the lowest Root Path Cost value in the Figure 5.9 is marked in bold. The root port of Switch3 is port 2, because through this port, the same as in Switch2 case, BPDU frame with the value of root path cost equal to 4 was received. Other possibilities were 23 and 27. Switch4 elects port1 as its root port, since it had received a proposal to reach the root switch by the path, which cost is 8. It is worth mentioning that Switch4 does not elect port3 as a root port, through which it is directly connected to the Root Switch at 100 Mb/s connection line, but it elects port1 and Switch3, because in this case, regardless of the need to perform more jumps the data link speed is 1 Gb/s. This example clearly shows that the use of cost path may help finding not the shortest path to the root switch, but the most efficient one.

For further actions it is necessary to clarify two definitions, i.e., path cost and root path cost. Path cost is a value that is assigned to each port. When the port receives a BPDU, its path cost value is added to the root path cost field's value. Root path cost is defined as an aggregated path value to the root switch, and this value is transmitted in BPDU frame's Root path cost field.

In the third step switches elect designated ports. Each segment of the network has one designated port. This port acts as a switch port, which sends and receives data traffic to and from the segment and the root switch. The idea is that if only one port handles the data traffic in each interface, all the loops should be terminated. As with the election of the root port the designated port is elected, by the calculation of the root path cost to the root switch (Figure 5.10).

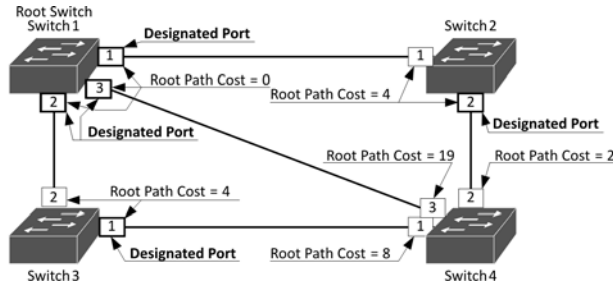


Figure 5.10. Designated port election

Designated port is assigned for each segment of the network. Let us consider the analysis of each network segment; in total there are five of them. In the segments of network between Switch1 and Switch2, Switch1 and Switch3, Switch1 and Switch4, the designated ports become ports in the Switch1; they are shown in bold line. This is because the Root Path Cost of these ports is equal to 0, as Switch1 is the root switch, and in the root switch all ports are designated. In the segment of network between Switch2 and Switch4 the port2 of Switch2 becomes the designated port as its Root path cost is equal to 4, while root path cost of port2 from Switch4, which also serves at the same network segment, is equal to 23. In the network segment between Switch3 and Switch4, port1 of Switch3 becomes the designated port.

Ports, which were elected as the root port and the designated port after listening and learning states, switch to the forwarding state, then they can send and receive user data. Ports that were not elected remain in the blocked state. When the network has loops and when these ports remain in blocked state, the logic tree topology is formed and the loops are terminated. In Figure 5.11 the state of ports after the elections, as well as after listening and learning states is shown. It can be seen that port2 and port3 of Switch4 are blocked and cannot send data, so the link between port3 of Switch1 port3 of Switch4, port2 of Switch2 and port2 of Switch4 is interrupted by software. Upon termination of these connections, the network

remains loop-free, because there is only one data transmission path between the switches.

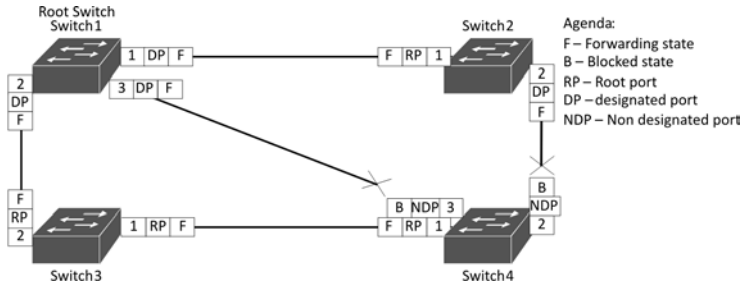


Figure 5.11. Ports states after election

5.6. Laboratory exercise. Analysis of STP protocol

Goal of Exercise: get acquainted with the STP technology.

Tasks of Exercise: get acquainted with the STP protocol and experimentally examine it by using the Cisco Ethernet switch.

Equipment: two PC with Linux OS, three Cisco Catalyst 2950 switches.

Workflow:

1. Preparation:

1.1. Examine the network diagram of the laboratory work (Figure 5.12).

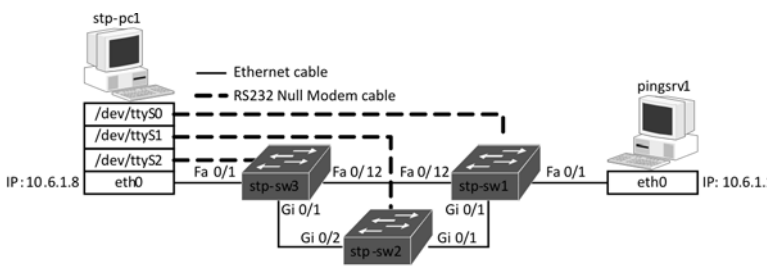


Figure 5.12. The network diagram of laboratory exercises

1.2. Run terminal program on the stp-pc1. Terminal program can be either gnome-terminal or xterm. In case of Fedora Core Gnome environment gnome-terminal program is set: Applications → System Tools → Terminal.

1.3. Run three GTKTerm programs on the stp-pc1. In case of Fedora Core Gnome environment run Applications → Accessories → GTKTerm. Find Configuration → Port and set the following parameters on one copy of the program: Port: ttyS0, Speed: 9600, Parity: None, Bits: 8, Stopbits: 1, Flow Control: None. Set ttyS1 port instead of ttyS0 on the second copy of the program. Set ttyS2 port instead of ttyS0 on the third copy of the program. Note: with these programs stp-sw1, stp-sw2 and stp-sw3 will be managed.

1.4. If the settings of serial port are correct, after pressing the Enter key, the following string must appear on GTKterm program window:

```
% Pleaseanswer 'yes' or 'no'.

Would you like to enter the initial configuration
dialog? [yes/no]:
```

The answer should be: no.

If the string does not appear, this means that the previous configuration of the switch has not been deleted. In order to delete the previous configuration, run the following commands:

```
Switch>en

Switch#erase startup-config

Erasing the nvram file system will remove all
configuration files! Continue? [co]
```

Press Enter key:

```
Switch#reload

Proceed with reload? [confirm]
```

Press Enter key.

1.5. Perform the initial configuration of the switches: set the name of the switch, the parameters of the ports and the terminal.

In order to switch into the configuration mode, run the following commands:

```
Switch>en

Switch#conf t

Enter configuration commands, one per line. End
with CNTL/Z.
```

In order to set the name of the switch, run the following command („*X*“ means the number of the switch):

```
Switch(config)#hostname stp-swX
```

In order to set the settings of management console, which prohibits the automatic logout when the user is not working with the switch, run the following commands:

```
stp-swX(config)#linecon 0

stp-swX(config-line)#logging synchronous

stp-swX(config-line)#exec-timeout 0 0

stp-swX(config-line)#end
```

2. Analysis of STP protocol. During the examination it would be needed to configure the switches. Note: analyse the results after about 50 s pause after the last time you performed the action.
   2.1. In order to turn off STP protocol, run the following command in each switch:

```
stp-swX(config)#no spanning-tree vlan 1
```

   2.2. In order to clean the ARP cache of the stp-pc1, run the following command on the stp-pc1:

```
[stud@stp-pc1 ~]$ ip neigh flush dev eth0
```

   2.3. Experiment I. Run the following command on stp-ak1:

```
[stud@stp-pc1 ~]$ ping 10.6.1.1
```

   Is there a regular connection with the pingsrv1? Enter the obtained results into Table 5.3.

Table 5.3. Results of the Experiment I

Experiment #	Is connection with pingsrv1	Additional Data	
I		None	
II		None	
III		Root switch	
IV		Blocked port	
V		Timeout	
VI		Blocked port	

   2.4. Experiment II. Disconnect the cable between any three of the switches. Is there a regular connection between the computers? Enter the obtained results into Table 5.3.
   2.5. Experiment III. Connect the cable that was disconnected during the Experiment II. In order to turn on STP protocol in all switches, run the following command:

```
stp-swX(config)#spanning-tree vlan 1

stp-swX(config)#exit
```

Is there a regular connection between the computers? Enter the obtained results into Table 5.3.

In each switch investigate the STP states of the ports, to which the cables are connected, and enter the obtained results into Table 5.4. In order to determine the port state, run the following command:

```
stp-swX#show spanning-tree
```

The explanation of command presented in Figure 5.13.

Table 5.4. STP port states

Switch	Port	Experiment #			
		III	IV	V	VI
1	Fa 0/1				
	Fa 0/12				
	Gi 0/1				
2	Gi 0/1				
	Gi 0/2				
3	Fa 0/1				
	Fa 0/12				
	Gi 0/1				

2.6. Experiment IV. Disconnect the cable that is connected to the blocked port. Does the connection with pingsrv1 still exist? Enter the obtained results into Table 5.3 and Table 5.4.

2.7. Experiment V. Connect back the port that was cut off during Experiment IV, and disconnect any other. Was the connection with the pingsrv1 lost? After what time has it recovered? Why? Determine the STP states of the ports that are used. Enter the obtained results into Table 5.3 and Table 5.4.

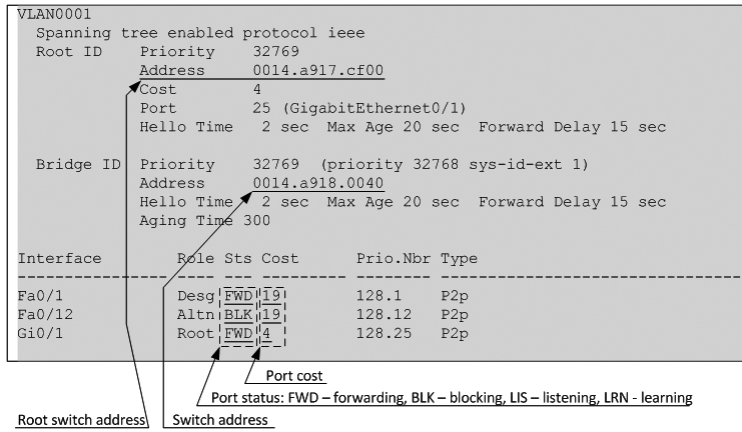


Figure 5.13. Results of show spanning-tree

2.8. Experiment VI. Connect back the port that was disconnected during Experiment V. In order to grant the ports Fa0/12 and Gi0/2 of stp-sw1 switch the higher priority, run the following command:

```
stp-sw1#conf t
stp-sw1(config)#interface fa0/12
stp-sw1(config-if)#spanning-tree cost 2
stp-sw1(config-if)#exit
stp-sw1(config)#interface gi0/2
stp-sw1(config-if)#spanning-tree cost 1
stp-sw1(config-if)#exit
stp-sw1(config)#exit
```

2.9. Run ping. Which ports transmit the packets now? Why? Determine the STP states of the ports that are used. Enter the obtained results into Table 5.3 and Table 5.4.

2.10. Show the obtained results to the tutor. With his permission in order to erase all the settings, run the following command:

```
stp-swX(config)#erase startup-config
```

Content of Laboratory exercise report:

1. The diagrams of the network.
2. The results in Table 5.3 and Table 5.4.
3. Explanation of each line of Table 5.3.
4. Answers to the questions.
5. Conclusions.

Questions:

1. What is STP protocol used for?
2. In what kind of network structures is it meaningful to turn STP on? Give examples.
3. Is there a situation when it is worth to have loops in the network? Give examples.
4. If all the ports have the same priority, which one is elected the root port?
5. Describe the principles of STP protocol.
6. What is the meaning of port priorities?
7. What is the meaning of switch priorities?
8. What is the average time spent when STP protocol reconfigures the network?
9. Does the reconfiguration time depend on the number of the switches in the network?
10. Can STP protocol completely perform its functions, if the equipment from different manufacturers is used in the network?

6. VIRTUAL LOCAL AREA NETWORK

6.1. Explanation of VLAN technology

Currently, all networks based on Ethernet technology use switches as switching devices. Compared with previously used hubs, the main advantages of the switches are greater safety and performance. This is achieved through the reduction or complete removal of the collision domain. In a large Ethernet network, there still are some gaps in security and performance loss, because the switch does not eliminate the broadcast domain. This problem becomes particularly acute when the network is large. It is therefore necessary to take measures in order to minimize the size of broadcast domain. This function is performed by the virtual local area network (VLAN). VLAN technology allows dividing a single physical LAN network into the multiple logical LANs. Nodes belonging to the same VLAN can send and receive a unicast, multicast and broadcast type of traffic without any restrictions. However, the devices belonging to different VLANs are isolated and cannot exchange information. Reducing the number of devices in the LAN increases security and reduces the network performance degradation generated by broadcast traffic. It is impossible to avoid broadcast traffic completely, because the such traffic is required for IP and MAC address mapping, file sharing services and etc. VLANs are developed by configuring the software of LAN switches. Switches that support VLAN technology, allow virtualizing their resources, i.e. instead of a single physical switch there are several logical switches with their ports and switching tables (Figure 6.1). An example of switch configuration is presented in this figure where five virtual LANs are created. Each VLAN has its own network number. By default, the switch port, which is not configured, belongs to the 1-st VLAN.

Another advantage of VLAN technology is that it allows separating the physical connection of LAN devices from the logical. This is especially relevant in the big enterprise networks. Let's say, employees of several different departments work in the same room. All of them can be connected to the same Ethernet switch, but to different VLAN networks. When the layout of their work places changes, there is no need to switch cables to the different Ethernet ports, it is enough only to re-configure the Ethernet switch (Figure 6.2). Thus, the VLAN technology allows flexible adaptation and change of network configurations, which in turn simplifies the administration.

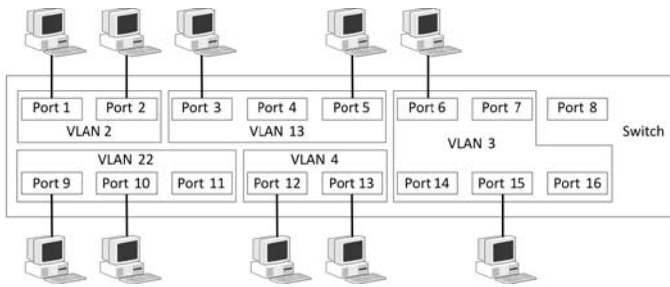


Figure 6.1. Concept of VLAN

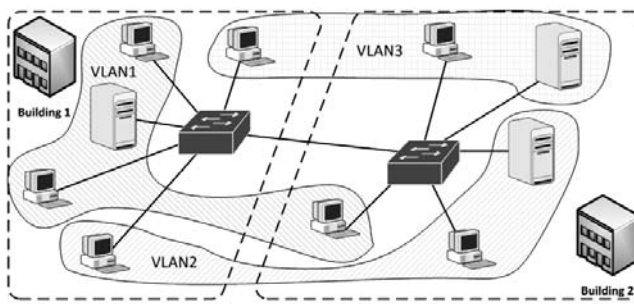


Figure 6.2. Enterprise network divided by VLANs

Previously, LAN segmentation was done using routers, but currently VLAN is used more widely because, compared with a router, VLAN switch has a much better performance. VLAN operates in the data link layer, so it is possible to use multiple IP subnetworks in one VLAN. However, almost always one-to-one relationship is used, that is when only a single IP subnetwork exists in one VLAN. Although the VLANs and IP subnetworks allow having independent Layer2 and Layer3 networks, but this rule simplifies the designing of networks.

If there is a need for separate VLAN networks to exchange data then these VLANs are connected by a router. There is an example in Figure 6.3 showing how to connect multiple VLAN networks. If more than two VLANs need to be connected, then there must be several routers or a router with more than two Ethernet interfaces.

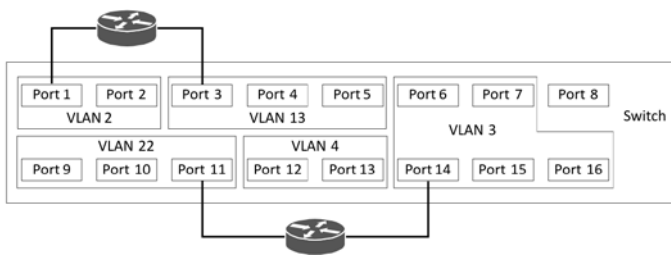


Figure 6.3. Routing between multiple VLAN

If it is necessary to connect several switches, containing the same VLAN, then each combined VLAN must have a single physical channel. Suppose there is a need to connect two switches, containing four VLANs. Then between these switches there should be four channels (cables). In addition, each switch for each VLAN should have an additional port for interconnecting cable. To solve this problem the trunk communications channel is used, as it allows the transmission of frames between multiple VLANs through

the same physical cable. An example in Figure 6.4 shows the connection of two switches by a trunk type line. Computers that are connected to different switches, but to the same VLAN, have a connection. In Figure 6.4 such computers are John and Peter and Mary and Steve. Meanwhile, the computers Adam and Bill will not have a connection, because Adam is connected to the VLAN13, which is not connected to any computer, and Bill is connected to VLAN 22, where this computer is the only node.

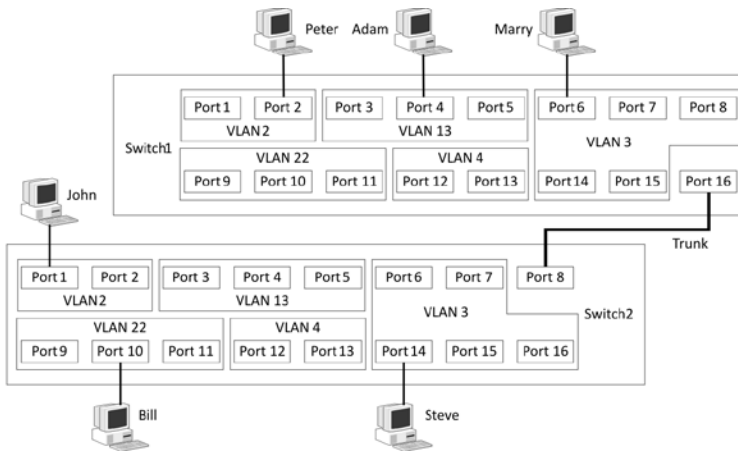


Figure 6.4. Trunk connection between switches

Since the trunk transmits data of the different VLANs, it is necessary for the switch to identify the VLAN number of the frame, which is received over the trunk port. Identification is necessary, because the received frame has to be transmitted only to that certain VLAN network, whose number is the same as VLAN network, which has sent that particular frame. Because VLAN is a Layer2 technology, so the VLAN number must be transmitted with data link layer protocol. Identification is performed by a tag, which is placed before sending a frame. There is an example of the tagging

given in Figure 6.5. Before transferring the frames through the trunk communication lines, the switch puts a tag in order to identify the VLAN, to which they belong to. At present IEEE 802.1Q protocol is commonly used for the tagging. The structure of this protocol frame is presented in Figure 6.6. The switch, which transmits the frame through the trunk type port, updates the outgoing Ethernet frame by inserting the VLAN Tag field between the source MAC address and Type field, and recalculates the CRC.

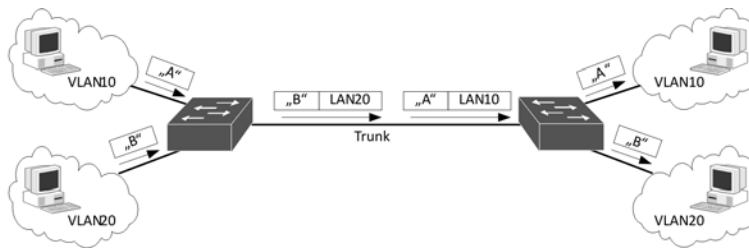


Figure 6.5. Frame tagging

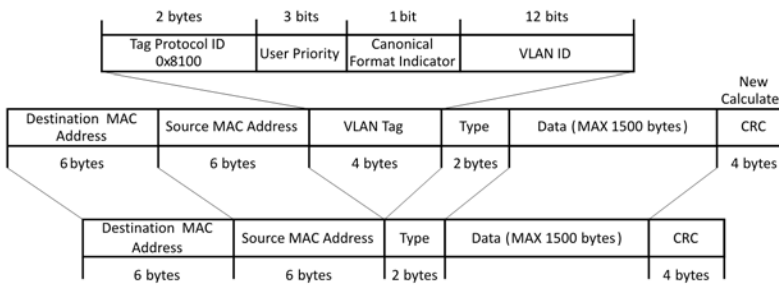


Figure 6.6. IEEE 802.1Q frame format

Table 6.1. IEEE 802.1p priorities

Priority Value	Network priority	Initials	Kind of traffic
1	0 (lowest)	BK	Background
0	1	BE	Best Effort
2	2	EE	Excellent Effort
3	3	CA	Critical Applications
4	4	VI	Video, < 100 ms la-tency
5	5	VO	Voice, < 10 ms latency
6	6	IC	Internetwork Control
7	7 (highest)	NC	Network Control

VLAN tag field of IEEE 802.1Q is 4 bytes long. Two bytes in it are intended for identification of tag protocol. This identifier uses a standard structure of Ethernet frame Type field. The code of VLAN tag frame is 0x8100. Further in the tag field is the frame priority. IEEE 802.1Q uses IEEE 802.1p standard to describe the priority. This field is used to ensure quality of service (QoS). Suppose that the same trunk line transmits two different traffics from two different VLANs: one is data traffic and the other is voice or video traffic. Then the intensive transmission of data can affect the voice and video transmission, because there can occur a frame delay, which in turn can degrade voice and video quality. Therefore in comparison with the priority of data traffic, the frames that are transmitted from voice or video VLAN receive a higher priority (VI or VO (Table 6.1)).

The canonical format indicator field (CFI) in VLAN tag indicates the format of the transmitted MAC address. If this field is equal to “0”, then the address is in canonical format, and if it is equal to “1”, then the address is in non-canonical format. In the Ethernet network the CFI is always a “0”.

The VLAN network number is transmitted in VLAN ID field. This field is 12 bits long, so the maximum possible number of different VLAN networks is 4094, as 0x000 and 0xFFF values are reserved. Since the IEEE 802.1Q standard modifies the standard IEEE 802.3 frame, by inserting additional 4 bytes, the maximum frame size in the networks that use VLAN tagging is increased from 1518 bytes to 1522 byte.

When configuring the VLAN in the switch, the ports must be assigned to other VLANs than the default VLAN, the ID of the latter is equal to 1. When assigning port to another VLAN its type is specified: untagged or tagged. Untagged type of port functions as a basic switch port. Frames sent through it are not tagged. Thus, the device, connected to that type of port, does not know whether it is connected to a switch that supports VLAN technology. Untagged ports mainly are used in access type of channels to which the end users nodes like workstations, printers and etc. are connected. The model of untagged type of switching is shown in Figure 6.1.

The tagged type of port tags the transmitted frames indicating, to which VLAN network they belong to. When the frames come through this type of the ports, the following combinations are possible: if the frame should be transmitted through the untagged type of port, then IEEE 802.1Q Tag field is cleared and the CRC is recalculated. If the frame should be transmitted through the tagged type of port, then the received frame is transmitted without any further modification. The tagged type of ports is connected only to those facilities, which also perform the tagging function to the dispatched frames: switches, routers, servers. The example of tagged type of ports is shown in Figure 6.5.

A case, when the router is connected to that type of port, should be discussed separately. It has already been mentioned that, if it is necessary for the network nodes of different VLANs to communicate, VLANs must be connected with a router. However, if the number of VLANs is high, it is then necessary to have a router with a

number of ports. However if the router is able to work with the IEEE 802.1Q tag, then a single port could be enough to consolidate all VLANs in the network. The model of such a connection is given in Figure 6.7. In this example all three computers can exchange information, it is important only that their IP settings, i.e., IP addresses and routing tables, are configured as follows: all three computers are on different IP subnets and router device is included in the computers' IP routing tables in order to reach nodes in other VLANs. The physical port of router device would create separate subinterfaces for each VLAN network (Figure 6.8). The same principle applies to connect the server through a trunk type of lines.

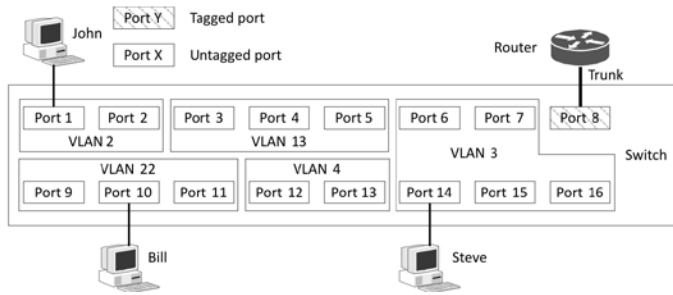


Figure 6.7. Connecting router with trunk

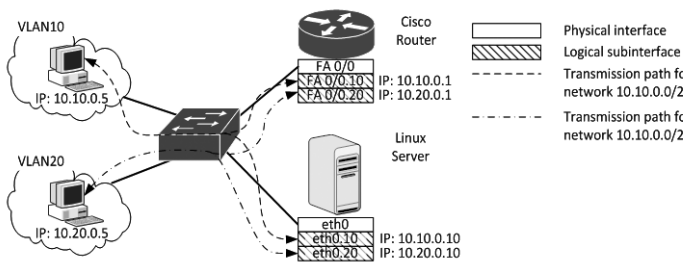


Figure 6.8. Connecting Router and Server over trunk lines

There are two methods of how ports are assigned to VLAN: static and dynamic. Network administrator makes the assignment of one static port to a particular VLAN. When a change in the user's work place occurs, the network administrator must manually reconfigure the switch.

The dynamic assignment is made according to the outgoing traffic attributes such as sender's MAC and IP addresses or by username. The identification of users in VLAN is performed by using the IEEE 802.1x protocol. This protocol allows creating a network, where even higher level of security is achieved, because when the username or password are entered incorrectly, the switch does not connect the computer to the network, or if it does connect then only to the guest VLAN, which has limited access.

If the network is small or home like then the VLAN configuration does not cause any serious trouble. However, under conditions of enterprise network it is easy to make mistakes in configuration. Therefore, there are several protocols that allow switching VLAN settings from one switch to another. Currently VLAN trunking protocol (VTP), created by Cisco, as well as IEEE 802.1ak, known as generic attribute registration protocol (GARP) are being used. VTP protocol operates only in Cisco switches and allows developing a central database for VLAN network, where all the switches in the network create VLAN. GARP operates in the majority of switches and allows creating a central database not only for VLAN, but also for multicast group.

6.2. Laboratory exercise. Analysis of VLAN technology

Goal of Exercise: get acquainted with the VLAN technology.

Tasks of Exercise: get acquainted with the specifics of VLAN networks and experimentally examine it by using the HP Ethernet switch.

Equipment: 2 units of HP ProCurve 2626 switches, 1 unit of Cisco 2620 router, 3 PCs with Linux OS.

Workflow:

1. Preparation:

1.1. Examine the network physical connection diagram of the laboratory work (Figure 6.9).

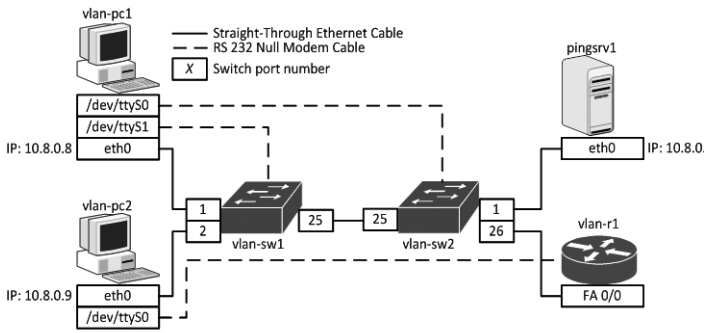


Figure 6.9. Laboratory exercise network diagram

1.2. Run terminal program on vlan-pc1 and vlan-pc2. Terminal program can be either gnome-terminal or xterm. In case of Fedora Core Gnome environment gnome-terminal program is set: Applications → System Tools → Terminal.

1.3. Run two GTKTerm programs on vlan-pc1. In case of Fedora Core Gnome environment run Applications → Accessories → GTKTerm. Find Configuration → Port and set the following parameters on one copy of the program: Port: ttyS0, Speed: 9600, Parity: None, Bits: 8, Stopbits: 1, Flow Control: None. Set ttyS1 port instead of ttyS0 on the second copy of the program. Note: with these programs the switches vlan-sw1 and vlan-sw2 will be managed. Run one GTKTerm program on vlan-pc2. Find Configuration → Port and set the following parameters: Port: ttyS0, Speed: 9600, Parity: None, Bits: 8, Stopbits: 1, Flow

Control: None. Note: with this program the router vlan-r1will be managed.

1.4. If the settings of serial port on vlan-pc1 are correct, after pressing the Enter key two times, the command string intended for switch control must appear on GTKterm program window. A particular entry depends on the status of the switch, so it is not displayed here. In order to delete the present configuration of the switch and reboot it, run the following commands:

```
Switch#erase startup-config

Configuration will be deleted and device rebooted,
continue [y/n]?
```

Press y key. Wait for the switch to reboot. When the following entry appears:

```
Waiting for Speed Sense.  Press <Enter> twice to
continue.
```

Press Enter key twice. When the following entry appears:

```
Connected at 9600 baud



ProCurve J4900C Switch 2626

Software revision H.10.83

...

Press any key to continue
```

Press Enter key one more time. The switch then is ready to accept configuration command.

1.5. Perform the initial configuration of the switches.
In order to switch into the configuration mode, run the following command:

```
ProCurve Switch 2626# conf t
```

In order to set the hostname of the switch, run the following command ("X" means the number of the switch):

```
ProCurve Switch 2626(config)# hostname vlan-swX
```

In order to logout from the configuration mode, run the following command:

```
vlan-swX(config)# exit
```

1.6. If the settings of serial port on vlan-pc2 are correct, after pressing the Enter key, the following string must appear on GTKterm program window:

```
% Please answer 'yes' or 'no'.

Would you like to enter the initial configuration
dialog? [yes/no]:
```

The answer should be: no.
If the string does not appear, this means that the previous configuration of the router has not been deleted. In order to delete the previous configuration, run the following commands:

```
Router>en

Router#erase startup-config

Erasing the nvram filesystem will remove all
configuration files! Continue? [co]
```

Press Enter key:

```
Router#reload

Proceed with reload? [confirm]
```

Press Enter key. The router then is ready to accept configuration command.

1.7. Perform the initial configuration of the router: set the hostname of the router and the parameters of the terminal.

In order to switch into the configuration mode, run the following command:

```
Router>en

Router#conf t

Enter configuration commands, one per line. End
with CNTL/Z.
```

In order to set the hostname of the router, run the following command:

```
Router(config)#hostname vlan-r1
```

In order to set the settings of management console, which prohibits the automatic logout when the user is not working with the router, run the following commands:

```
vlan-r1(config)#line con 0

vlan-r1 (config-line)#logging synchronous

vlan-r1 (config-line)#exec-timeout 0 0

vlan-r1 (config-line)#exit
```

Table 6.2. Checking of connection between computers

Experiment #	Is connection alive			Explanation
	vlan-pc1 – vlan-pc2	vlan-pc1 – pingsrv1	vlan-pc2 – pingsrv1	
I				
II				
III	-			
IV	-			
V	-			
VI	-			
VII	-			
VIII	-			
IX	-			
X				
XI				

2. Static VLAN network design and analysis.
   2.1. Experiment I. In this experiment the operation of switches is investigated, when VLAN is not configured. In order to check the connection between vlan-pc1 and vlan-pc2, between vlan-pc1 and pingsrv1, and then between vlan-pc2 and pingsrv1, run the following ping command on gnome-terminal program window:

```
[stud@vlan-pc1 ~]$ ping -c 5 10.8.0.9

[stud@vlan-pc1 ~]$ ping -c 5 10.8.0.1

[stud@vlan-pc2 ~]$ ping -c 5 10.8.0.1
```

   Enter the obtained results into Table 6.2.

2.2. In order to check what VLAN networks have been created in the switches, run the following command:

```
vlan-sw1# show vlans
```

Enter the obtained results into Table 6.3.

The explanation of this command results is given in Figure 6.10.

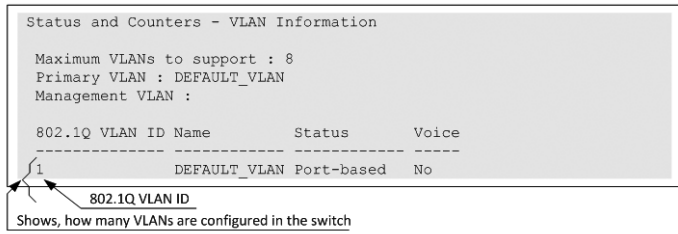


Figure 6.10. Explanation of show VLANs command

Table 6.3. List of default VLANs

Switch hostname	Number of VLANs
vlan-sw1	
vlan-sw2	

2.3. Experiment II. In this experiment several VLANs are created in vlan-sw1 switch and connectivity between computers is investigated. Connect pingsrv1 into the port 3 of vlan-sw1 switch. In vlan-sw1 create two VLAN networks, which ID would be 10 and 20. In order to assign the port, to which vlan-pc1 is connected, to VLAN 10, and then the port, to which vlan-pc2 is connected, to VLAN 20, run the following commands

2.4. In order to switch into the switches configuration mode, run the following command:

```
vlan-sw1# conf t
```

In order to create a new VLAN with ID 10, run the following command:

```
vlan-sw1(config)# vlan 10
```

In order to assign port 1 (which type is untagged) to VLAN ID 10, run the following command:

```
vlan-sw1(vlan-10)# untagged 1
```

In order to logout from VLAN 10 configuration mode, run the following command:

```
vlan-sw1(vlan-10)# end
```

Repeat the same actions for VLAN 20:

```
vlan-sw1(config)# vlan 20

vlan-sw1(vlan-20)# untagged 2

vlan-sw1(vlan-20)# exit

vlan-sw1(config)# exit
```

In order to check what VLANs have been created, run the following command:

```
vlan-sw1# show vlans
```

2.5. In order to check the connection between computers run the following ping command as explained in 2.1. Enter the obtained results into Table 6.2.

2.6. Experiment III. This experiment is a continuation of the Experiment II, in which vlan-sw1 switch is configured additionally. In order to verify, to which VLAN in vlan-sw2 a port 3 belongs, run the following command:

```
vlan-sw1# show vlans port 3
```

Assign this port to VLAN ID 10.
Check whether vlan-pc1 and vlan-pc2 have connection with pingsrv1. Enter the obtained results into Table 6.2.
2.7. Experiment IV. In this experiment the operation of two switches with configured VLANs are investigated. Connect pingsrv1 as it is shown in Figure 6.9. Check the connection between vlan-pc1 and pingsrv1 also between vlan-pc2 and pingsrv1. Enter the obtained results into Table 6.2.
2.8. Experiment V. This experiment is a continuation of the Experiment IV, in which vlan-sw1 is configured additionally. Assign port 25 of vlan-sw1 to VLAN ID 10. Check the connection between vlan-pc1 and pingsrv1 also between vlan-pc2 and pingsrv1. Enter the obtained results into Table 6.2.
2.9. Experiment VI. This experiment is a continuation of the Experiment V, in which vlan-sw2 is configured additionally. Create two VLAN networks in vlan-sw2, with accordingly ID 30 and ID 20. Assign port 1 to VLAN 30, and port 2 to VLAN 20. Check the connection between vlan-pc1 and pingsrv1 also between vlan-pc2 and pingsrv1. Enter the obtained results into Table 6.2.
2.10. Experiment VII. This experiment is a continuation of the Experiment VI, in which vlan-sw2 is configured additionally. Assign port 25 of vlan-sw2 to VLAN 30. Check the connection between vlan-pc1 and pingsrv1 also between vlan-pc2 and pingsrv1. Enter the obtained results into Table 6.2.
2.11. Experiment VIII. In this experiment the operation of tagged type ports is investigated. Set a tagged type for ports 25 and 26 in both switches and assign this port to all VLANs designed in the switch. In order to do so, run the following commands:

```
vlan-sw1# conf t
vlan-sw1(config)# vlan 1
vlan-sw1(vlan-1)# tagged 25-26
vlan-sw1(vlan-1)# exit
```

```
vlan-sw1(config)# vlan 10
vlan-sw1(vlan-10)# tagged 25-26
vlan-sw1(vlan-10)# exit
vlan-sw1(config)# vlan 20
vlan-sw1(vlan-20)# tagged 25-26
vlan-sw1(vlan-20)# end
vlan-sw2# conf t
vlan-sw2(config)# vlan 1
vlan-sw2(vlan-1)# tagged 25-26
vlan-sw2(vlan-1)# exit
vlan-sw2(config)# vlan 30
vlan-sw2(vlan-30)# tagged 25-26
vlan-sw2(vlan-30)# exit
vlan-sw2(config)# vlan 40
vlan-sw2(vlan-40)# tagged 25-26
vlan-sw1(vlan-40)# end
```

Check the connection between vlan-pc1 and pingsrv1, then between vlan-pc2 and pingsrv2. Enter the obtained results and explanations into Table 6.2.

2.12. Experiment IX. This experiment is a continuation of the Experiment VIII. Connect pingsrv1 into port 2 of vlan-sw2. Check the connection between vlan-pc1 and pingsrv1, then between vlan-pc2 and pingsrv2. Enter the obtained results and explanations into Table 6.2.

2.13. Experiment X. This experiment explains, how it is possible to forward data from one VLAN to another using a router. In order to assign any other IP address to vlan-pc2 and set the routing table, run the following commands:

```
[stud@vlan-pc2 ~]$ ip addr flush dev eth0

[stud@vlan-pc2 ~]$ ip addr add 10.8.1.9/24 brd +
dev eth0
```

```
[stud@vlan-pc2 ~]$ ip route add default via
10.8.1.253
```

In order to create a route to 10.8.1.0/24 network from vlan-pc1, run the following commands:

```
[stud@vlan-pc1 ~]$ sudo ip route add 10.8.1.0/24
via 10.8.0.253
```

Configure two virtual subinterfaces in vlan-r1 router, then assign them to corresponding VLANs in accordance with Figure 5.12:
In order to switch to the configuration mode, run the following commands:

```
vlan-r1#conf t

Enter configuration commands, one per line. End
with CNTL/Z.
```

In order to activate interface fastEthernet 0/0, run the following commands:

```
vlan-r1(config)#interface fastEthernet 0/0

vlan-r1(config-if)#no shutdown
```

In order to create subinterface fastEthernet 0/0.10, run the following command:

```
vlan-r1(config-if)#interface fastEthernet 0/0.10
```

In order to assign VLAN Tag 10 to this subinterface, run the following command:

```
vlan-r1(config-subif)#encapsulation dot1Q 10
```

In order to assign IP address to this subinterface, run the following commands:

```
vlan-r1(config-subif)#ip address 10.8.0.253
255.255.255.0

vlan-r1(config-subif)#exit

vlan-r1(config)#interface fastEthernet 0/0.20

vlan-r1(config-subif)#encapsulation dot1Q 20

vlan-r1(config-subif)#ip address 10.8.1.253
255.255.255.0

vlan-r1(config-subif)#exit
```

Create one more VLAN with ID 10 in vlan-sw2 switch. Assign this VLAN for port 25 and 26 as tagged type and for port 1 untagged type. Connect pingsrv1 to port 1 of vlan-sw2. Check the connection between all computers. While doing so, do not forget that vlan-pc2 IP address has been changed. Enter the obtained results and explanations into Table 6.2.

2.14. Experiment XI. This experiment explains, how to connect computer to a switch using tagged connection. Connect vlan-pc1 to port 26 of vlan-sw1. Delete all eth0 interface IP addresses in vlan-pc1, create two subintrafaces and assign them accordingly to VLAN 10 and VLAN 20. In order to do so, run the following commands:

In order to create eth0.10 subinterface and assign VLAN ID 10 tag to it, run the following command:

```
[stud@vlan-pc1 ~]$ sudo vconfig add eth0 10
```

In order to create eth0.20 subinterface and assign VLAN ID 20 tag to it, run the following command:

```
[stud@vlan-pc1 ~]$ sudo vconfig add eth0 20
```

In order to assign IP addresses for interfaces: eth0.10 and eth0.20, run the following commands:

```
[stud@vlan-pc1 ~]$ ip addr add 10.8.0.8/24 brd + dev eth0.10

[stud@vlan-pc1 ~]$ ip addr add 10.8.1.8/24 brd + dev eth0.20
```

In order to activate interfaces eth0.10 and eth0.20, run the following commands:

```
[stud@vlan-pc1 ~]$ ip link set eth0.10 up

[stud@vlan-pc1 ~]$ ip link set eth0.20 up
```

Check the connection between all computers. While doing so, do not forget that vlan-pc2 IP address has been changed. Enter the obtained results and explanations into Table 6.2.

Content of Laboratory exercise report:

1. Laboratory network diagram.
2. Provide fully explained Table 6.2 and VLAN connection diagrams that were used during each experiment. Use diagram shown in Figure 6.1 as a prototype.
3. Provide the conclusions.

Questions:

1. In which layer does VLAN technology operate?
2. What is the difference between tagged and untagged types of port?
3. Why it is recommended to have a separate IP subnetwork in each VLAN?
4. For what purpose subinterfaces are created in the servers and routers?

5. What is the purpose of priority field in VLAN tag?
6. What is the maximum possible VLAN number in the network?
7. What are the benefits of VLAN technology?
8. Why untagged type of port could be assign only to one VLAN?
9. Explain how to create a connection between different VLANs.
10. What type of VLAN ID is created as default?

7. REFERENCES

1. Behrouz, F. 2006. *Data Communications and Networking*. McGraw-Hill. 1168 p.
2. Charles, S. 2000. *Ethernet: The Definitive Guide* (Definitive Guides). O'Reilly. 522 p.
3. Donahue, A. G. 2011. *Network Warrior*. O'Reilly Media. 786 p.
4. Forouzan, F. 2009. *TCP/IP Protocol Suite* (Mcgraw-Hill Forouzan Networking). McGraw-Hill Science. 928 p.
5. Gast, M. 2005. *802.11 Wireless Networks: The Definitive Guide.* Second Edition. O'Reilly. 656 p.
6. Kozierok, C. M. 2005. *The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference*. No Starch Press. 1616 p.
7. Peterson, L. L.; Davie, B. S. 2011. *Computer Networks.* 5th Edition: A Systems Approach. Morgan Kaufmann. 920 p.
8. Rich, S. 2008. *The Switch Book: The Complete Guide to LAN Switching Technology*. Wiley. 816 p.
9. Tanenbaum, A. S.; Wetheral, D. J. 2010. *Computer Networks.* 5th Edition. Prentice Hall. 960 p.
10. William, S. 2010. *Data and Computer Communications.* 9th Edition. Pearson Education International. 888 p.