Nerijus PAULAUSKAS
Eimantas GARŠVA

COMPUTER NETWORKS

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

Nerijus PAULAUSKAS
Eimantas GARŠVA

COMPUTER NETWORKS

A Laboratory Manual

Vilnius "Technika" 2012

N. Paulauskas, E. Garšva. Computer Networks: A Laboratory Manual. Vilnius: Technika, 2012. 166 p. [6,16 author's sheets. 2012 06 04]

The *Computer Networks* laboratory works and methodical guidelines are designed to help students to acquire knowledge about computer network technologies, network equipment, network design principles, configuration and troubleshooting tasks. Twelve laboratory works are presented in the book. Each laboratory work consists of theoretical material that introduces the object of investigation, laboratory work assignments and methodical guidelines. At the end of each laboratory work the references and review questions and problems are presented.
This book is intended for students studying Computer Engineering and Information Systems Engineering courses, also for students studying the subject of computer networks.

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

Reviewed by:
Dr Vaidotas Barzdėnas, VGTU Department of Computer Engineering,
Assoc. Prof Dr Vladislavas Daškevičius, VGTU Department of Computer 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 1341-S, issued by VGTU Press TECHNIKA *http://leidykla.vgtu.lt*

Language editor Dalia Blažinskaitė
Typesetter *Donaldas Petrauskas*

eISBN 978-609-457-161-9
doi:10.3846/1341-S

Contents

Preface

The rising importance of the computer systems to everyday life makes the role of the computer networks vital. Computer networks are used to transfer data between the communicating systems. Computer networks need to be designed using appropriate topology and network technologies in order to be fast, reliable and easy expandable. The transmitted data is divided into the appropriate Protocol Data Units. Ability to analyse the data is required to understand the network protocols and to troubleshoot the network problems. Local Area Networks interconnect to one another and compose Wide Area Networks. Routing is critical in WANs.

This book provides knowledge about the network topologies and the cables used, LAN design principles and Internet Protocol addressing practises. Introduction to Windows OS network diagnostic tools and the instructions how to analyse the network traffic are provided. Routing and Virtual LAN concepts are described. Instructions how to use Graphical Network Simulator GNS3 for computer network simulation in order to understand how the network technologies work and to test the network design are provided. Laboratory works provide initial knowledge on these network protocols: IP, TCP, Ethernet, ARP, DNS, HTTP, SMTP, POP3, RIP, OSPF and STP.

For laboratory works, specially created network designs and general purpose applications for Windows OS are used. Program names and important terms are highlighted in *italics* in the text book, program menu items or window element names are written in **bold**.

Before coming to the laboratory, students must read the theoretical material and prepare to answer the control questions.

Laboratory work 1

Computer Networks, Communication Technologies and Topologies

Objectives

The aim of the laboratory work is to get acquainted with computer network topologies, cables, connectors and wireless technologies. Analyse their main characteristics, advantages and disadvantages.

Basic knowledge and theory

Computer network – interconnected computers using the appropriate hardware and software capable of exchanging the information contained therein.

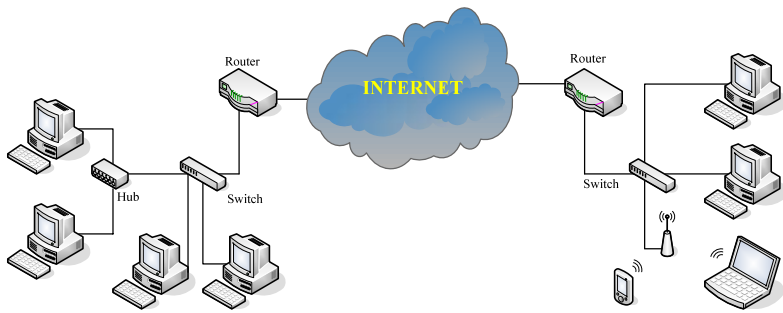


Fig. 1.1. Computer network

According to the size, computer networks can be divided into:

- *Local Area Network* – LAN;
- *Metropolitian Area Network* – MAN, or regional;
- *Wide Area Network* – WAN.

Local Area Network – closed network, serving users of one organisation in a small area (up to several kilometers) connected via telephone, cable, optical or wireless communication lines.

Metropolitian Area Network – connecting computer users in a large area (region, city) via various communication lines.

Wide Area Network – a set of smaller networks connected via communication lines positioned in a large geographic location.

Networks are one-level (*peer-to-peer*) or with a distinguished management server (*client-server*).

One-level network has no central computer. Some of the hardware equipment (hard disks, CD-ROMs, printers), connected to individual computers can be used together. Each network user can specify access rights to resources of his/her computer to other users.

In a network with a managing server stands out a central network computer – the server connected to user computers – clients. Such network is also called a client-server network. Users can use server resources. Network is managed and peripheral devices are monitored by network software – networking operating system.

Computer networking method is called topology. The term topology in the context of networks defines a way in which the hosts are interconnected in a network. Topology is described as the layout of lines and switching elements and defines the data transmission paths, which can be used between any pair of hosts.

There are physical and logical topologies. The physical topology describes the ways of physical connections between network hosts, while a logical topology describes the data flow between network hosts. For example, a logical ring topology is realized in a physical star topology. In many cases, the physical and logical topologies coincide. We are going to discuss the physical computer network topologies.

Local Area Network topologies

Bus topology is shown in Figure 1.2. Until the year 2000 it was a widely applied local network topology type.

Bus topology advantages:

- low-cost cable system;
- hosts can communicate without additional switching devices.

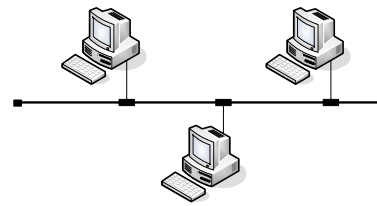


Fig. 1.2. Bus topology

Bus topology disadvantages:

- in case of cable failure, the whole network is out of service;
- low performance – only one host at a time can send information;
- when connecting a new host to a network, it is necessary to stop operation of the network.

In a network with a *Star topology* switch equipment (hub or switch) is in the centre of the network (Fig. 1.3). The purpose of switching devices is to transmit information to one (switch) or all (hub) network hosts.

Star topology advantages:

- in case of cable failure only one host is out of service and it does not affect the others;
- host connection to a network is simple, because the connection is performed only with the switching device;
- more advanced switching devices can filter out the transmitted data packets.

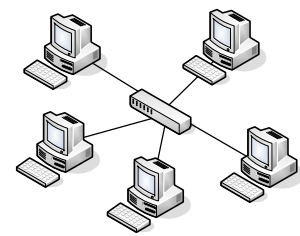


Fig. 1.3. Star topology

Star topology disadvantages:

- the network price is higher than the bus topology, because the switching device must be used;

- if the switching device is not geographically in the centre of the network, a host connection can be expensive and difficult;
- network performance and scalability depend on the switching device performance and switching port numbers;
- in case of the switching device failure the network becomes unavailable.

Existing networks are usually designed by the star topology, the hierarchical connected hubs or switches, located in network centres. The combination of several star topology networks to one makes a tree-like network topology.

Network topology in which the central switching unit (upper level of the hierarchy) is connected to one or more second-level switching devices and the latter with a third-level devices etc. is called a *Tree topology* (Fig. 1.4). Advantages and disadvantages of the tree topology are adequate to the star topology's advantages and disadvantages.

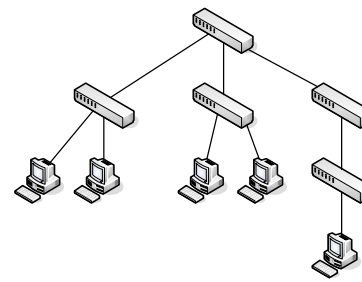


Fig. 1.4. Tree topology

When all the hosts on the network are connected to the ring, the network topology is called the *Ring topology* (Fig. 1.5). Data is transmitted sequentially from one host to another, usually in one direction. If the host detects its data, it copies them into its buffer.

Ring topology advantages:

- high data transfer reliability – the sender can control the data acquisition because the data must come back to him;

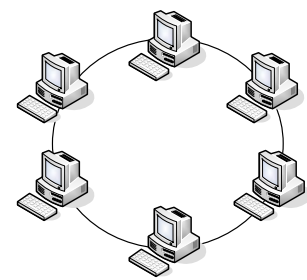


Fig. 1.5. Ring topology

- not restrictions for the size of the ring, there are just the distance restrictions between the hosts;
- greater reliability in comparison with the star and bus topology, in case of ring disruption in one place the connection remains.

Ring topology disadvantages:

- data transfer time increases in proportion to the number of stations;
- special measures are necessary to ensure that the ring works in case of the cable or host failure or when connecting a new host to the ring.

When all the host on the network are connected with each other in separate communication lines, the network topology is called the *Mesh topology* (Fig. 1.6). In practice, usually only partial mesh network topology is applied, where not all but a few particularly relevant to the network hosts are connected by separate lines.

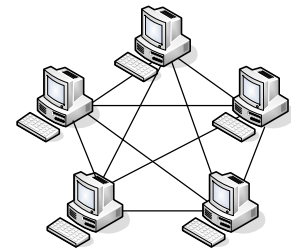


Fig. 1.6. Mesh topology

Mesh topology advantage:

- ensures a reliable and fast data transfer – upon failure of one communication line the data can be transmitted through others communication lines.

Mesh topology disadvantages:

- it is not cost-effective, because it requires a large amount of connections on each host;
- applied only to a small networks.

Cables used at Local Area Networks

Cables that are used for computer networks are standardized. Standards describe and evaluate a number of parameters, such as signal suppression, active resistance, impedances, the capacity of the electromagnetic field surrounding the wire strength and so on.

The following cable standards are presently used:

- American EIA/TIA-568A;
- International ISO/EIC11801;
- European EN50173.

There are three main groups of cables:

a) Coaxial cable:
   - thin;
   - thick.

b) Twisted pair cable:
   - *Unshielded Twisted Pair* – UTP;
   - *Shielded Twisted Pair* – STP;
   - *Foiled Twisted Pair* – FTP.

c) Fiber optic cable:
   - *Single Mode Fiber* – SMF;
   - *Multi Mode Fiber* – MMF.

Coaxial cable. The thicker the cable and the better the shielding, the less attenuated the signal. Cables with reduced attenuation are more suitable for large transmission speeds with low class equipment, and under the same conditions can transmit a signal over a greater distance.

Base of the cable is a copper wire in the center of the cable, and a metal mesh separated by a dielectric insulator – the shield (Fig. 1.7).

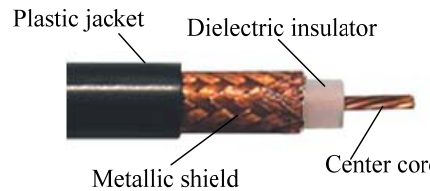


Fig. 1.7. Coaxial cable

Twisted pair cables. The simplest twisted pair is two copper wires twisted with each other and separated with a dielectric. This allows reducing electromagnetic interaction of several twisted nearby pairs. Twisted pair can be used for both analog and digital data transmission.

Unshielded twisted pair cable consists of pairs of insulated conductors twisted together. One conductor of the pair is called a

Ring (in coloured marking – one-coloured), the other a *Tip* (two-coloured). All pairs are also numbered Ring1, Tip1, Ring2, Tip2, etc. Pairs have their numbers in accordance with colour marking: Blue/White – 1 pair, Orange/White – 2 pair, Green/White – 3 pair, Brown/White – 4 pair.

In the environment with very strong electromagnetic fields it is recommended to use a shielded twisted-pair cable.

This type of cable can have two types of shields: foil and metal mesh. Foil is used more frequently due to smaller weight and price. Shield of this cable must be grounded.

Shielded twisted pair (STP) cable each twisted pair is shielded (Fig. 1.8), while in *Foiled Twisted Pair* (FTP) type of cable the shield is the same to all twisted pairs (Fig. 1.9).

For laying the cable outside a twisted pair cable with special double insulation is used. If the cable is installed between two buildings (runs above the ground), it is convenient to use a special cable with steel messenger (Fig. 1.10).

Fig. 1.8. STP cable

Fig. 1.9. FTP cable

Fig. 1.10. Twisted pair outdoor cable with steel messenger

For final connection (e.g. between the wall socket and the computer) more flexible patch cable with stranded wires is used (Fig. 1.11).

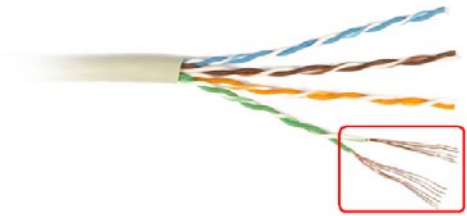


Fig. 1.11. Flexible patch cable

Fiber optic cable is different from copper cables, because the signal is transmitted using light pulses. Single mode fiber optic cables are composed of a core surrounded by a cladding. The cladding is surrounded by a coating, dielectric strengthening material, and finally an outer jacket (sheath) (Fig. 1.12). The cladding provides a lower refractive index to cause reflection within the core so that light waves can be transmitted through the fiber.

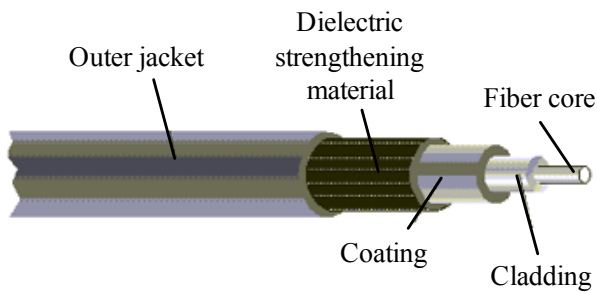


Fig. 1.12. Single mode fiber optic cable

Fiber optic cables are used for high-speed networks. It is common that light pulsation is logic one and absence of light is logic zero.

Signal, which travels via a cable, is reflected from the cladding. According to the refractive index and the core width, the cables are divided into:

a) *Single Mode Fiber*;
b) *Multi Mode Fiber* cable with a rapidly changing refractive index;
c) *Multi Mode Fiber* cable with smoothly changing refractive index.

Several fiber optic cable connections are given below:

- ST (*Straight Tip*) – circular connection;

- SC (*Subscriber Connector* or *Square Connector*) – rectangular connection;

- FC (*Ferrule Connector* or *Fiber Channel*) – circular connection.

Twisted pair cable categories

Twisted pair cables are divided into the following categories:

- *Category 1* – a telephone cable, which transmits voice, not suitable for data transmission. Maximum transmitted signal frequency 1 MHz;
- *Category 2* – a cable that can transmit data up to 4 Mbps rate and is composed of four twisted pairs;
- *Category 3* – a cable capable of transmitting a signal up to 10 Mbps. Used in networks, operating in accordance with an Ethernet 10Base-T technology standard. Maximum transmitted signal frequency – 16 MHz;
- *Category 4* – a cable that can transmit data up to 16 Mbps and consists of four twisted pairs. Used in Token Ring networks. Maximum transmitted signal frequency – 20 MHz;
- *Category 5* – a cable that can transmit data up to 100 Mbps and consists of four twisted pairs. Used in networks, operating in Ethernet 100Base-TX standard technology, as well as other network technologies such as ATM, Token Ring, 100Base-T, 10Base-T. Maximum transmitted signal frequency – 100 MHz. Cables in this category are: UTP, FTP, STP types;
- *Category 5e* (the letter "e" means 'enhanced') – this cable is suitable for 1000Base-T. Can be shielded or unshielded.

The twisting degree of twisted pairs varies depending on the category. The higher the category, the higher twisting degree.

Twisted pair cable categories are defined in EIA/TIA 568 A and EIA/TIA 568 B standards.

Category 5e, UTP cable. Its cross-section is shown in Figure 1.13.

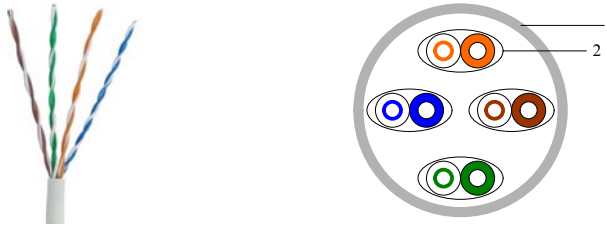


Fig. 1.13. Category 5e, UTP cable and its cross-section: 1 – jacket, 2 – solid twisted pair

Category 5e, FTP cable. Its cross-section is shown in Figure 1.14.

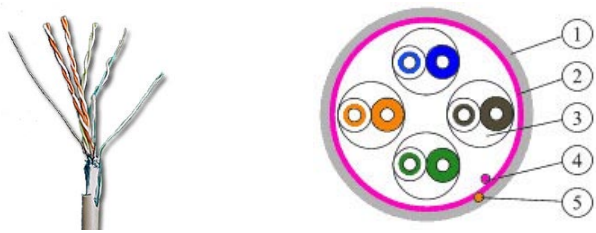


Fig. 1.14. Category 5e, FTP cable and its cross-section: 1 – jacket, 2 – shield foil, 3 – solid twisted pair, 4 – drain wire, 5 – rip-cord

Category 5e, S/FTP cable. Its cross-section is shown in Figure 1.15.

Fig. 1.15. Category 5e, S/FTP cable and its cross-section: 1 – jacket, 2 – shield-braid, 3 – drain wire, 4 – shield foil, 5 – stranded twisted pair

- *Category 6* – a cable that can transmit data up to 600 Mbps. Used in networks, operating in Ethernet 1000Base-T standard technology, as well as other network technologies such as 10BaseT Ethernet, 100BaseTX Fast Ethernet, 1000BaseTX, 155 MBit ATM, 622 MBit ATM, 1.2 GBit ATM. Category 6, UTP cable and its cross-section is shown in Figure 1.16.

Fig. 1.16. Category 6, UTP cable and its cross-section: 1 – jacket, 2 – solid twisted pair, 3 – spacer

- *Category 7* – Transmitted signal frequency – 600 MHz. Category 7 cable is different from other categories because it has be fully shielded, so it is thicker and less flexible. Used in networks, operating in Ethernet 1000Base-TX and 10GBase-T standard technologies. Category 7, SSTP cable and its cross-section is shown in Figure 1.17.

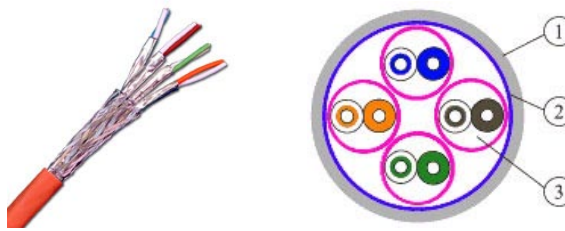


Fig. 1.17. Category 7, SSTP cable and its cross-section: 1 – jacket, 2 – shield-braid 3 – solid twisted pair with shield foil

Twisted-pair cables are connected to network devices using various types of connectors. Modular connectors *Modular Jacks* (sockets) and *Modular Plugs* are the most common in connections of 1, 2, 3 and 4 pairs of category 3–6 cables. Plugs are better known as RJ-11 (4 wires) or RJ-45 (8 wires). The correct name of this type of network sockets is Jack Modular 8P8C, of plugs – Modular Plug 8P8C, here 8P indicates a link connector (8 positions) and 8C the number of contacts used (in this case 8). For telephone cables it is used in 6P4C (6 positions, 4-pin) configuration. Other markings, e.g., P-6-4 – six-positions 4-pin plug, PS-8-8 – eight positions 8-pin shielded plug are also known. 6-position plugs can be plugged into sockets of 8 positions but not vice versa.

Structure of sockets of the fifth and higher categories and method of wiring connection differ from the sockets of category 3. Here the socket is mounted on a printed board on which S110 or Krone type contacts are attached. In addition, reactive elements are printed on the board combining reactive impedances. These elements help to reduce signal reflections from contacts in high-speed (100 Mbps or more) networks. Various types of sockets are shown in Figure 1.18.

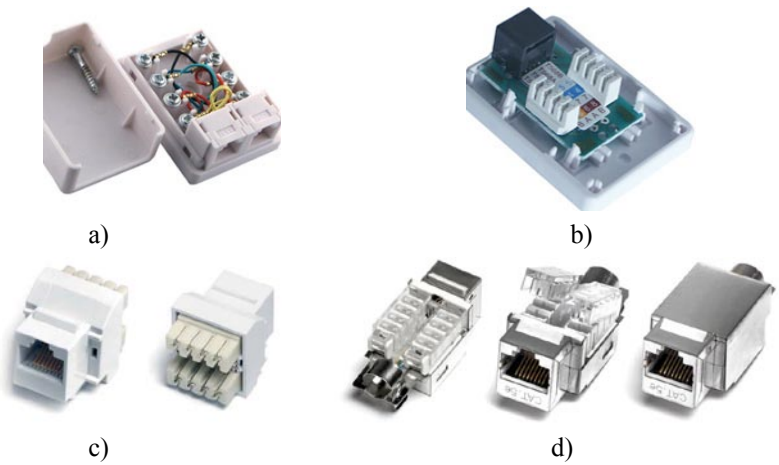


Fig. 1.18. Various types of sockets: a) phone line socket; b) category 6 socket; c) category 5e not shielded socket; d) category 5e shielded socket

Figure 1.19 shows the RJ-45 connection for a twisted pair 8-wire cable connected to a computer network, while the similar RJ-11 is used to connect phone lines to modem and telephone sets and has only 4 wires.

a) b)

Fig. 1.19. Twisted pair cable's plugs: a) RJ-45, b) RJ-11

Network switching patch panels are used for convenience. They are mounted in the racks or cabinets. On the one side of the patch panel there is an RJ-45 connection block, and on the other side there is an S110 or Krone type contacts to which network cables from different locations are connected (Fig. 1.20).

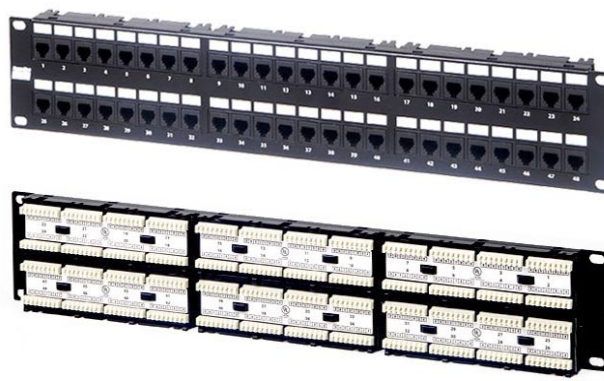


Fig. 1.20. Patch panel

Twisted-pair cable wires are of different colours and connected according to a diagram provided in the TIA/EIA-568A or TIA/EIA-568B standards (Table 1.1).

Table 1.1. Cable wiring schemes

RJ-45 pin number	EIA/TIA-568A	EIA/TIA-568B
1	White/Green	White/Orange
2	Green	Orange
3	White/Orange	White/Green
4	Blue	Blue
5	White/Blue	White/Blue
6	Orange	Green
7	White/Brown	White/Brown
8	Brown	Brown

Connection also depends on the purpose of network devices connected with a cable. Direct cable connection is used for connecting a computer to the hub or switch when connection diagrams of RJ-45 plugs at both ends of the cable are the same, i.e. only TIA/EIA 568B or TIA/EIA 568A cable wiring scheme is used.

For connection of two computers or switches a crossover cable is used when at one end a plug is placed in accordance with TIA/EIA 568B diagram, and at the other end in accordance with TIA/EIA 568A. It should be noted that modern network equipment has the ability to automatically recognize what connection cable is connected and can work with both direct connection and crossover cables.

Examples of the cable connection are listed below.

Work assignment and methodical guidelines

1. Review the material on networks and their topologies provided at the beginning of the laboratory work and answer the following questions:

1.1. Explain what a computer network is?
1.2. How are networks divided according to their size? Define them.
1.3. Explain what a network topology is?
1.4. Which topologies are used in forming local area networks?
1.5. Explain what a physical and logic topology is?
1.6. Which topologies are critical to the breakdown of cable or switch and which topologies are not critical?
1.7. What is the main difference between the bus and star network topologies?
1.8. Name advantages and disadvantages of each topology.
1.9. What is the difference between one rank topology and topology based on the managing station?

2. Type in 192.168.0.200 address into web browser. After the page has displayed, select **Computer Networks → Laboratory work 1 methodical materials → Ethernet 10Base, 100Base, 1000Base networks**. Review the material on Ethernet standards and fill the following tables.

Standard 10Base-5. Main characteristics:

Speed:	
Topology:	
Cable type:	
Maximum segment length:	
Advantages:	
Disadvantages:	

Standard 10Base-2. Main characteristics:

Speed:	
Topology:	
Cable type:	
Maximum segment length:	
Advantages:	
Disadvantages:	

Standard 10Base-T. Main characteristics:

Speed:	
Topology:	
Cable type:	
Maximum segment length:	
Advantages:	
Disadvantages:	

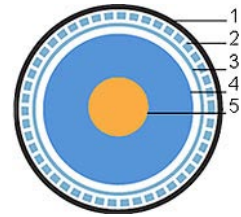
Standard 10Base-F. Main characteristics:

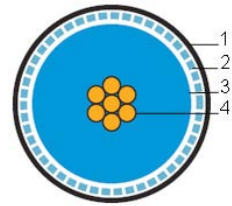
Speed:	
Topology:	
Cable type:	
Maximum segment length:	
Advantages:	
Disadvantages:	

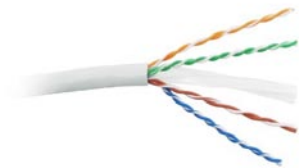
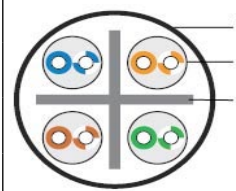
Main characteristics:

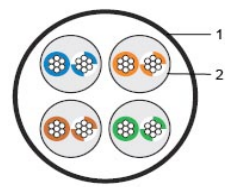
	Speed	Topology	Cable type	Maximum segment length
100Base-T4				
100Base-TX				
100Base-FX				
1000Base-T				
1000Base-SX				

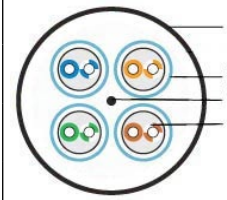
3. Review the material on cable and their categories used in Local Area Networks and do the exercises given below:
   3.1. Name the main cable standards.
   3.2. Name the types of cables used in computer networks.
   3.3. Name the categories of twisted pair cables.
   3.4. Review the types of cables provided at this address: (*http://www.hyperline.com/catalog/cable/*). Identify what types of cables are shown in tables provided below and describe them.

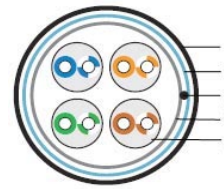
Cable type: ?		Designations:
		1. 2. 3. 4. 5.

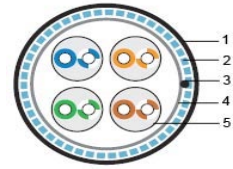
Cable type: ?		
		1. 2. 3. 4.

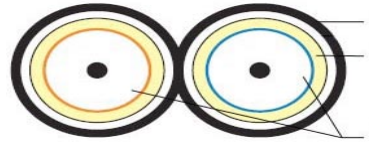
Cable type: ?		Designations:
		1. 2. 3. 4.

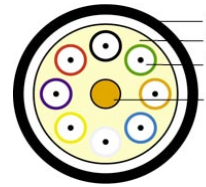
Cable type: ?		Designations:
		1. 2.

Cable type: ?		Designations:
		1. 2. 3. 4.

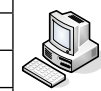
Cable type: ?		Designations:
		1. 2. 3. 4. 5.

Cable type: ?		Designations:
		1. 2. 3. 4. 5.

Cable type: ?		Designations:
		1. 2. 3.

Cable type: ?		Designations:
		1. 2. 3. 4.

3.5. Connect two computers using twisted pair cable. (Write wire colours in the table)

Com-puter 1	RJ-45			RJ-45	Com-puter 2
	1		1		
	2		2		
	3		3		
	4		4		
	5		5		
	6		6		
	7		7		
	8		8		

3.6. Connect a computer and a switch using twisted pair cable. (Write wire colours in the table)

Com-puter	RJ-45			RJ-45	Switch
	1		1		
	2		2		
	3		3		
	4		4		
	5		5		
	6		6		
	7		7		
	8		8		

Content of report

1. Work objective.
2. The results of Tasks 1–4.
3. General conclusions of the work.

Review questions and problems

1. How computer networks are divided according to the size?
2. Explain what a network topology is.
3. Name and explain all network topologies. What are advantages and disadvantages of each network topologies?
4. What type of cables is used in local area networks?
5. What are the main twisted pair cable categories?

Literature

Barnett, D., Groth, D., McBee, J. 2004. *Cabling: The Complete Guide to Network Wiring*. 3rd Edition. Sybex. 720 p. ISBN-13: 978-0782143317.

Components of structured cabling systems. http://www.hyperline.com/catalog/cable/.

Laboratory work 2

Design of Local Area Computer Network

Objectives

The aim of the laboratory work is to get acquainted with design principles of local networks and to design local area computer network. Base the decision by which specific technologies and equipment were selected.

Basic knowledge and theory

Active computer network equipment is powered from the main power grid and transmits information via network cables in the form of electrical signals. Active network equipment varies according to their roles and the layer of the OSI model (more details about OSI model is provided in Laboratory work 3, theory section), which data units it processes.

A Hub (also known as *concentrator*, *repeater*) is the simplest form of communication equipment used in star topology networks. A hub functions at the physical (1st) layer of the OSI model. Every computer is connected to the hub by a separate cable. A signal that came to one port is also transmitted (repeated) to all other ports. Hubs have different number of ports, for a twisted pair RJ-45 connectors are used and for optical cable – ST connectors. 10 Mbps or 100 Mbps speed hubs are used in Ethernet networks.

Up to four hubs can be connected with each other (this logic is called rule 3-4-5). Since all hubs are interconnected and are repeating the received signal, all devices connected to them are on the same collision domain – a single physical network segment where the packets can "collide". The more computers are in the same collision domain, the less the network efficiency is, because while a single computer is sending data all the other computers must wait.

For the same reasons, the switches can not be connected using a ring topology.

Passive hubs also exist, which are realized using precise resistors or diodes. They allow up to three computers to be connected and do not use electricity.

A *Bridge* is used to amplify network signals and to combine two computer network segments. These segments can be implemented using different topologies and technologies. The bridge operates at the data link layer and transmits a signal only if the recipient is on a network segment that is on the other side of the bridge. The bridge has two ports and network segments that are connected to them are in different collision domains. The bridge can be viewed as a two-port switch.

A *Switch* is the main network device in a star topology which may have different number of ports. Modern switches often have 5, 8, 16, 24 or 48 ports. When sending data, a switch toggles specific ports which depend on the sender and receiver, and does this in full duplex mode. This means that one switch can be used to independently transmit different data flows in a duplex mode. A switch operates in OSI model data link layer (2nd) and determines which computer is connected to a specific port by its MAC (Media Access Control) address.

Switches can be implemented as independent devices or be modular, which can be installed in a router or a special chassis. It is appropriate to install switches in racks, so they usually have certain width and height. Modern switches often have several slots to insert GBIC (*Gigabit Interface Converter*), and special ports for connecting together a number of switches. Switches can be connected with each other using Ethernet ports, but the use of specific ports allows higher transfer rates and lets you control the interconnected switches as one.

Some switches support VLAN (*Virtual Local Area Network*), which allows a single physical switch to be split into multiple logical switches, thus isolating separate network parts.

Switches can not be connected in a ring topology. That means there can be no more than one route, unless the switch supports STP (*Spanning Tree Protocol*), which selects the best routes and uses them, and if their termination occurs, it selects other paths thus increasing the reliability of the network.

Some of the switches are able to process packets header information transmitted by the network layer. Those types of switches are called third level. They add flexibility in managing network traffic and have full or partial routing capabilities.

Switches, according to their use in a network, can be divided into access (end users are connected to them, they have a lot of ports, user filtering capabilities), transport (they are involved in transmitting large data streams and should be very fast) and core (used in Internet supplier network core and data centers, must be very fast and reliable).

A *Router* is a device that connects computer networks and performs data routing function, i.e. formation of network route maps and tables. A router is a device that connects networks or subnets to a local area network. So if the network is continuous and there are no subnets, the router is a device after which the network administrator's control and responsibility ends. In this case, the router is a gateway through which all local network computers indicated by the default route reach other networks (mostly internet). Router routes, indicating how to access other networks may be set by the administrator (static routing table entries) or by a dynamic router protocol (then the routing table entries are formed automatically according to the routing protocol algorithm). The router is usually implemented as a separate unit, although a personal computer with two network cards can also do its functions. The router shows its functionality when used in global networks.

The router is an OSI model network (3rd) layer device that directs IP packets to a designated route according to the IP address. Network firewall functions can be realized inside a router. In home

networks routers with integrated switches (Small Office/Home Office – SOHO) are widely used.

Converter allows connecting two different transmission media. Recently optical transmitters became widespread and popular, which allow attaching an optical fiber to a copper twisted pair. Converters can be autonomous, realized in a separate enclosure (Fig. 2.1a) or modular, realized as, for example, a *Gigabit Interface Converter* – GBIC (Fig. 2.1b).

a)

b)

Fig. 2.1. Media converters: a) autonomous; b) GBIC

A *Wireless Access Point* is a device that allows other wireless devices to communicate with him and, if the setting permits it, with each other. Local networks mostly use connections that use the Wi-Fi standards set. The device allows authorized devices to connect, ensures access and data security by using dedicated security protocols (for example WEP, WPA, WPA2).

Wireless networks devices for home users or offices often have integrated routers and switches.

A *Network Interface Card* (NIC) is a physical interface between the computer and data transmission media, which, along with network drivers, performs protocol functions. Usually a network card is inserted into the computer's PCI (*Peripheral Component Interconnect*) slots, integrated into the motherboard or connected to the USB port (usually wireless). The speed of the network card

depends on the type of interface (for example, 100 Mbps). Simple network cards use CPU resources, while more advanced may have additional features (for example, hardware encryption).

Network card has an assigned manufacturer's hardware MAC address, which can be changed by the card driver.

Local area computer network design principles

Determine how many computers and other network devices need to be connected with each other and leave an opportunity for the network to grow.

Select the network topology. It depends on the amount of network equipment, minimum network speed. The most common local area network topology is a star and tree.

Set up user's computers. Leave an opportunity to connect shared printers, scanners and so on.

Select and setup the active network equipment. Network equipment must have an appropriate number of ports and the required speed. It is appropriate to place the network equipment in the geometric center of the user's computer collections. Foresee how the active network equipment will be connected to one another, since linking them also takes up ports. Although when a local computer network is being setup, cabling and other passive network equipment require most of the work and resources, the amount of active network equipment must also be optimal. Not all network devices can be placed at the desired location, sometimes the location is determined by prior conditions (for example, network input place).

Plan the cable-laying places. Consider the maximum cable length. The cable from the switch to the computer consists of three parts:

1. Cable from the switch to the commutation panel. A commutation cable is used, with the length no more than 1 m;
2. Cable from the commutation panel to the wall socket length should not exceed 90 m;

3. Commutation cable from the wall socket to the computer should be no longer than 10 m.

Such cable dividing provides a network with the necessary flexibility when being used and if rearrangement is required, and allows using more expensive flexible cables with stranded wires for commutation cables and cheaper solid wires for the long parts of the network. It is convenient when drilling is not required, for example, laying out the cable above a suspended ceiling. Network wall socket should be 30 cm from the ground. Cable, if necessary, can be laid out on the floor using special plastic or aluminum channels.

Let's discuss other important practical nuances when designing a local area network, which will not be taken into consideration while doing this laboratory work. Distance must be maintained from the power cables. If voltage of the power cables is up to 2.5 KV, the distance should be at least 12 cm, if the voltage is higher – more than 25 cm. The network cable should not be bent at an acute angle; even for copper cables the recommended radius is more than 5 cm, optical cable are even more susceptible to bending. Shielded twisted pair at one end must be grounded. The most convenient way to do it is by grounding the commutation rack. Sockets and commutation panel ports must be marked otherwise the use of the network becomes particularly difficult.

Work assignment and methodical guidelines

1. Read attentively the theory section of laboratory work. Before starting the work discuss all obscure questions with the lecturer.
2. Type the 192.168.0.200 IP address in the web browser and click the *Computer Networks* link on the displayed page. When select *Diagram_editor_CADE* at the second laboratory work paragraph to review a brief description of the 2D vector editor CADE.
3. From the same web page, download the building plan indicated in the task version list (Table 2.1). The task version corresponds to the number of your computer.

4. Activate the 2D vector editor CADE.

4.1. Create a new document: **File → New**.

4.2. Insert the building plan as a different layer:

Create a new layer: **Tools → Layers**, *New*, name the layer *plan*, then select it and click *Set Current*, then *OK*. Insert a picture of the plan: click a button *Insert picture* (see CADE description), then *Load from file*, locate your plan, select it and click *Open*. Stretch the loaded plan across the whole page. Lock the plan layer: **Tools → Layers**, select layer 0, then *Set current*; select layer *plan* and deselect the second and third boxes (from *Enable* and *Select*).

4.3. Arrange the users on the building plan (user number is given in the task version list). Mark user computers using *Office layout equipment/PC*, for printers use *Office layout equipment/Printer*. Reduce the icons so that they comfortably fit into the plan.

4.4. Plan and arrange the active network equipment on the building plan. Use the icons from *Network Logical Symbols* to mark the network equipment. Mark the switches and note the number of ports next to them by using the tool *Insert text*.

4.5. Draw the network connections with the *Line* or *Polyline*. Write down the amount of cables next to it, as shown in Figure 2.2.

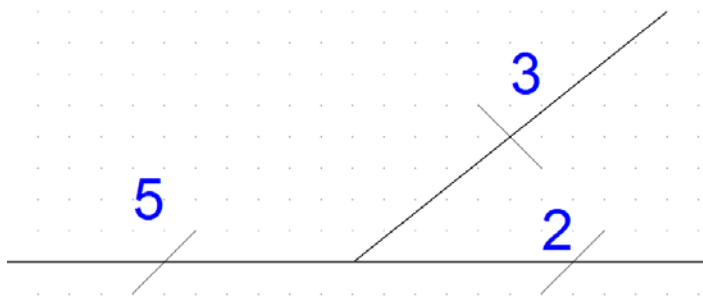


Fig. 2.2. Designation of cables amount in the project

4.6. Save the designed local computer network, name the file *surname_taskversion*.

4.7. Check the local area network settings: the longest distance of the cable, cable length used for the network, the most loaded switch.

5. Suggest suitable network devices for your design project. Find particular network devices models on the internet and analyze their options.

Table 2.1. Task version list

Version Nr.	No. of work places / Expandable to	Maximum employee number in a room	Shared printer No.	Wireless connection	Suspended ceiling / Height (m)
1.	30/40	4	2	Lobby	–/2.7
2.	20/30	4	3	Lobby	–/2.8
3.	40/45	3	3	Hallway	–/3
4.	20/45	3	4	Hallway	–/2.4
5.	30/35	2	4	Yard	–/2.5
6.	40/47	2	4	Yard	+/2.6
7.	40/50	–	6	–	–/2.9
8.	40/50	–	6	–	+/3.1
9.	40/45	2	6	–	+/3.2
10.	40/45	–	3	Common space	–/3
11.	30/35	2	6	Hallways	–/2.5
12.	30/40	3	4	Big hallway	–/2.9
13.	40/50	2	6	–	–/2.8
14.	40/60	2	6	Lobby	+/2.6
15.	40/60	2	3	Lobby	+/2.7
16.	45/55	2	2	Lobby	+/2.6

Content of report

1. Work objective.
2. Local computer network project.
3. Local are network settings, the designed network pros and cons.

4. List the network equipment that was used in the project.
5. General conclusions of the work and your thoughts and considerations about the design network.

Review questions and problems

1. What are the main differences between a switch and a hub?
2. At what OSI model layer does the mentioned active network equipment operate?
3. What are the differences between a bridge and a converter?
4. What is the maximum length of a UTP cable at an Ethernet 100 Mbps network segment?
5. What are the pros and cons of using Wi-Fi in a local area network?
6. What are the main local area computer network design principles?

Literature

Olifer, N., Olifer, V. 2006. *Computer Networks: Principles, Technologies and Protocols for Network Design*. Wiley. 1000 p. ISBN-13: 978-0470869826.

Oppenheimer, P. 2010. *Top-Down Network Design*. 3rd Edition. Cisco Press. 600 p. ISBN-13: 978-1587202834.

Laboratory work 3

Investigation of Internet Protocol (IP) Addressing

Objectives

To become acquainted with the *Open System Interconnection* (OSI) model and *Transfer Control Protocol* and *Internet Protocol* (TCP/IP) stack. Understand the main principles of IP addressing and division of a network into subnets.

Basic knowledge and theory

A computer network is an intricate complex of hardware and software components. The entire software-hardware complex of the network can be described by a multilayer model. Such description of the complex enables standardisation of the characteristics of its various layers. Standardisation allows interconnection of networks employing various technologies, using hardware and software of different manufacturers.

Considering this, *International Organization for Standardization* (ISO) has developed an *Open System Interconnection* (OSI) model in the early 80-ties.

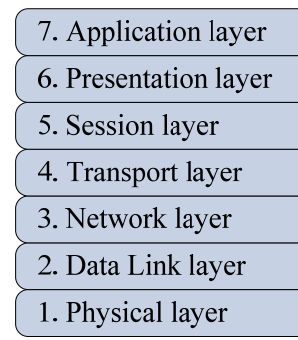


Fig. 3.1. OSI model layers

OSI model provides communication layers between systems, their names and functions. The model consists of seven layers (Fig. 3.1). The layers were established according to the following basic principles:

1. a layer is established when a new layer of abstraction is necessary;
2. each layer executes its closely specified functions;

3. functions of each layer have to be chosen considering the international protocol standards;
4. information flow between the interfaces of various layers has to be minimised;
5. number of layers has to be sufficient to prevent unnecessary instances of different functions occurring at the same layer, and small enough that the entire architecture does not become cumbersome.

OSI model does not specify exact services and protocols which have to be used at each layer. It only describes what each layer should do.

OSI model layers

Physical layer describes the physical medium of data transfer (e.g., coaxial cable, twisted pair, optical cable, wireless communications). This layer ensures interaction between the network host and the medium of data transfer. Series of bits are transferred through the network channels at the physical layer. Control at the physical layer means identification of the start and end of the frame, which is carrying the data, also forming and accepting of signals of a certain physical nature. Standards of the physical layer describe mechanical, electrical, functional and procedural characteristics necessary for establishment, operation and termination of physical communication.

The objective of the *Data Link layer* – checking of accessibility of the transmission medium, error detection and correction. Bits are grouped into frames at the data link layer, checksum of the frame is calculated and added to the frame. If checksums of the sent and received frame do no match, an error is produced. The data exchange between the two objects at the data link layer can be executed in three ways.

The directions of data exchange can be: *simplex*, *half-duplex* and *full-duplex*.

Data can travel in a single direction in case of simplex transfer. In half-duplex data transfer data is transferred in one direction or

the other at a certain time. Full-duplex enables sending data in both directions at the same time.

Network layer ensures transfer of network packets between the network nodes and is responsible for data addressing. The following basic functions are implemented in this layer: packet routing, fragmentation and identification of transport layer protocol.

The function of the *Transport layer* is to accept data from the session layer, divide into smaller parts if necessary, transfer to the network layer and ensure that all parts reach the destination correctly. All this must be done efficiently in a way to protect the upper layers from the inevitable change of hardware.

The most popular connection of the transport layer is error-free point-to-point channel which delivers messages or bytes in such order in which they were sent. However one of the types of transport services can be transmission of individual messages without acknowledgement of delivery.

Several connections could be established simultaneously, therefore it is necessary to indicate which message belongs to which connection. This is identified by a socket which is a pair of the port and an IP address. A port is information of the transport layer, related to the service supplied by the application layer.

The Session layer allows users of different computers to establish communication sessions with each other. This layer establishes the beginning and the end of the communication session (normal or emergency), the duration and mode of a session. Rights of the object to contact another object are controlled when establishing communication sessions.

The Presentation layer describes methods of presentation of transferred data (coding, compression, conversion) without changing the contents of information. The formats of data presentation can differ by these characteristics:

- the ordering of bit sequence and the number of bits for encoding a symbol;

- ordering of byte sequence;
- presentation and coding of symbols;
- file structure and syntax.

The Application layer describes protocols which are used by the applications. The application layer is a collection of protocols, designed for user access to distributed resources – files, printers, websites, etc. The most popular protocols of the application layer are HTTP, FTP, DNS, SMTP, etc.

Every layer receives data from the upper layer and adds their header thus creating a new data unit. The newly created data unit is sent to the lower level. The data units of the application, presentation and session layers are called messages, units of the transport layer – segments, network layer – packets, data link layer – frames, and the physical layer – bits.

The data path from computer A to computer B is indicated in Figure 3.2.

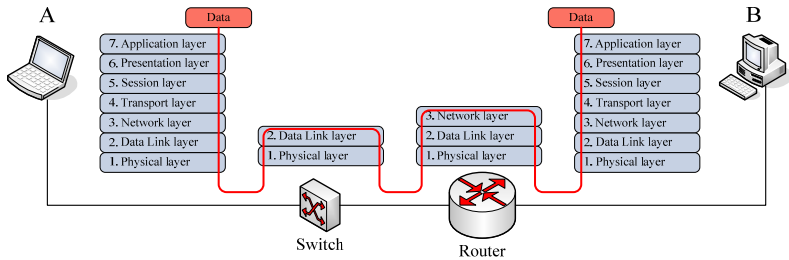


Fig. 3.2. The data path from computer A to computer B

The data sent by computer A have to pass all layers to reach the lowest physical layer and will be sent to the switch. The switch reads the data link layer header which indicates to which network device the data is addressed. Further on, the data are transferred to the router which transfers data to the recipient, in this case computer B, according to the address (IP) in the network layer header. Before reaching a computer B the user data pass the reverse path, i.e. from physical to the application layer in which headers of each layer are removed.

OSI seven layer model describes general principles of data transmission in a network. Protocols and interfaces are used to describe the interconnection of software and hardware elements. *Protocol* – a set of rules of interconnection between objects of a single layer, describing formats of data transferred between objects.

The most frequently used protocol stack in computer networks is TCP/IP.

The TCP/IP model is based on the TCP/IP stack. The comparison of this model and OSI model is presented in Figure 3.3. The main difference – number of layers. The TCP/IP model has 4 layers. The application layer corresponds to the 3 upper layers of the OSI model, and the data link – two lower layers of the OSI model, other layers are the same. ***Note:*** *different number of TCP/IP stack layers is given in various sources: 4 or 5 layers. The literature which indicates that the TCP/IP stack has 4 layers, the data link and the physical layers of the OSI model are usually merged into one, and in case of 5 layers, the data link and the physical layer are separated.*

ARP (*Address Resolution Protocol*) is designed to link the address of the network adapter (*Media Access Control* – MAC) with the IP address.

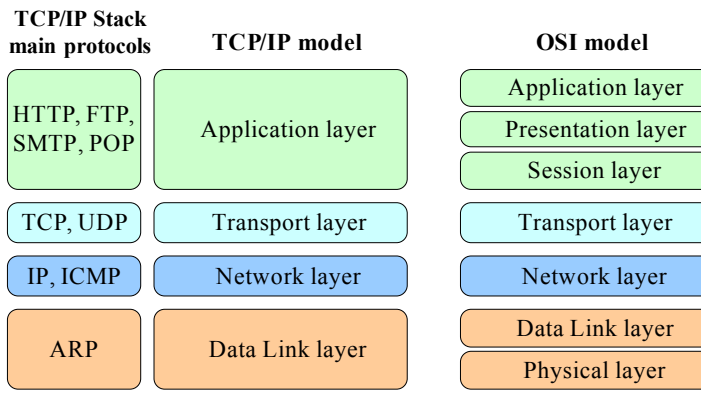


Fig. 3.3. TCP/IP stack and OSI mode comparison

IP (*Internet Protocol*) is designed for sending packets and routing them between networks.

ICMP (*Internet Control Message Protocol*) is designed to send packet transmission error reports.

TCP (*Transmission Control Protocol*) is connection oriented. This protocol is used by applications which need acknowledgement of data reception.

UDP (*User Datagram Protocol*) does not guarantee transmission of packets. This protocol is widely applied for multimedia processes, such as IP telephony, real time video conferencing, etc.

HTTP, FTP, SMTP, POP protocols ensure user access to network resources.

IP addressing

Computers and routers on the network are identified by IP addresses. An IP address consists of two logical parts – *network* number and a *host* number. The first is the same for all host of the network. The second is unique (inside a given network) and is designated to a specific network host.

IP protocol version 4 (IPv4) IP address consists of 32 bits, therefore a total of $2^{32} = 4\ 294\ 967\ 296$ IP addresses is possible. As the popularity of the Internet has grown it turned out that 32 bit addresses are not sufficient, thus the new internet protocol version 6 (IPv6) assigns 128 bits to an IP address, that is $2^{128} = 3.403 \cdot 10^{38}$ addresses. Since IPv6 protocol version is not yet widely implemented we will examine the IPv4 version protocol. In this protocol the 32 bit IP address is separated into four 8-bit long fields – octets (1 byte), where each of them can have a decimal number from 0 to 255. IP addresses can be written in several ways:

Dotted decimal:	88.	203.	5.	43
Binary:	01011000	11001011	00000101	00101011
Hexadecimal:	58CB052B			
Decimal:	1489700139			

The first method was chosen for simplicity – the bytes of the IP address are written by four decimal numbers, separated by the dots.

IP address classes

All IP addresses are divided into several categories called the address classes. An IP address class indicates which part of the address is designated to the network number, and which – to the hostnumber. Five IP address classes are distinguished: A, B, C, D, E. IP address class is indicated by the leading bits of the first byte (Fig. 3.4).

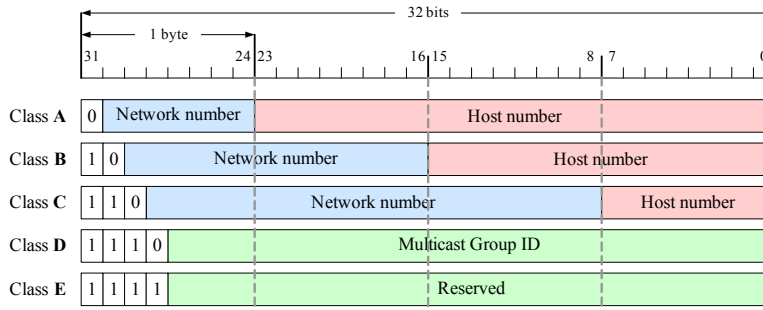


Fig. 3.4. IP address classes

Class A addresses are assigned to very large networks. The leading bite of the address always equals 0. The next seven bits indicate the network number, and the rest 24 bits (3 bytes) indicate the host number. This allows $2^7 = 128 - 2 = 126$ of class A networks. It is additionally subtracted by two because the highest and the lowest addresses of the A class network have special purpose and cannot be assigned to a specific network (Table 3.2). Class A networks can have $(2^{24} - 2) = 16\,777\,214$ host addresses. When calculating the number of nodes the total number must be subtracted by two, because a host address with all zero bits corresponds to the address of the network itself and an address consisting of 1's – a broadcast address for the entire network (Table 3.2).

Class B addresses are assigned to large and medium size networks. Binary number 10 is written in the two leading bits of the IP address. The following 14 bits indicate the network number, while the remaining 16 bits (2 bytes) indicate the host number. This permits $2^{14} = 16\ 384$ networks each consisting of $2^{16} - 2 = 65\ 534$ hosts.

Class C addresses are used in small networks. Binary number 110 is written in the three leading bits of the IP address. The following 21 bits indicate the network number, while the remaining 8 bits (1 byte) indicate the host number. This way more than 2 million ($2^{21} = 2\ 097\ 152$) networks can exist, each with 254 ($2^8 - 2 = 254$) hosts.

Class D addresses are used for group addressing. Binary number 1110 is written in the four leading bits of the IP address. D class addresses are from 224.0.0.0 to 239.255.255.255. Such IP addresses make up logical groups of computers, their nodes can belong to separate networks. D class has several special purpose addresses: 224.0.0.1 means all systems in a given subnet, 225.0.0.2 means all routers in a given subnet. Group addresses are usually used when simultaneously sending audio or video data for many users.

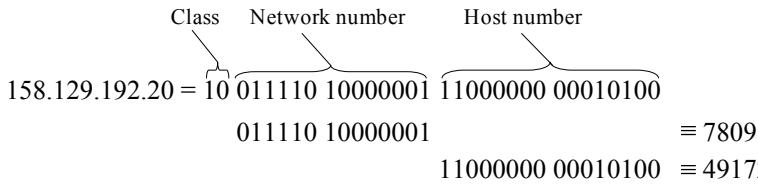
Class E addresses are reserved and used for experimental needs. In this case four leading bits of the IP address equal 1111_2.

Internet users are assigned A, B and C class IP addresses.

Table 3.1. Number of networks and hosts according to IP class

	Class		
	A	**B**	**C**
Number of bits used to identify network/host	8/24	16/16	24/8
Leading address bits	0	10	110
Number of networks	$2^{8-1} - 2 = 126$	$2^{16-2} = 16\ 384$	$2^{24-3} = 2\ 097\ 152$
Number of addresses per network	$2^{24} - 2 = 16\ 777\ 214$	$2^{16} - 2 = 65\ 534$	$2^8 - 2 = 254$
Network number range	1–126	128–191	192–223

The address of the web server of a Vilnius Gediminas Technical University website is this: 158.129.192.20. This address belongs to Class B, it is the 49 172 computer on the network 7809:



Some IP addresses have special purpose and cannot be assigned to a specific network host. These addresses are listed in Table 3.2.

Table 3.2. Special purpose IP addresses

IP address	Description
255.255.255.255	Broadcast address on a given network
<network>.0.0.0	Class A network IP address
<network>.255.255.255	Broadcast address on a class A network
<network>.0.0	Class B network IP address
<network>.255.255	Broadcast address on a class B network
<network>.0	Class C network IP address
<network>.255	Broadcast address on a class C network
127.<any>/8	Internal host loopback address
0.0.0.0	Refers to the default network

The number of computers, service stations, routers and other network devices is constantly increasing and currently it is significantly larger than the number of possible IP addresses. Thus a question arises: how can all these devices be connected to the Internet if the number of IP addresses is limited? For this purpose each of the IP address classes allocate a certain range of IP addresses, assigned for private networks (Table 3.3). These addresses can be assigned at administrator's own discretion. The same private IP addresses can be used on different private networks. Thus it becomes possible to

create a large number of private networks that are connected into a single network using IP addresses allocated to public networks.

Table 3.3. The range of private IP addresses according to IP class

Class	IP addresses range
A	10.0.0.0–10.255.255.255
B	172.16.0.0–172.31.255.255
C	192.168.0.0–192.168.255.255

IP addresses used in local networks can be static or dynamic. Static IP addresses are permanent IP addresses, which are set by the administrator who configures the parameters of the network. Static IP addresses are always given to routers. Dynamic IP addresses are temporary IP addresses which are assigned to devices automatically by using the DHCP (*Dynamic Host Configuration Protocol*) protocol. Dynamic allocation of IP addresses is usually used in wireless networks.

Methods of IP address allocation

There are no specific rules for assigning IP addresses. An address can be given in a sequence or values which are easy to remember can be chosen:

- IP addresses can be assigned grouping nodes by type – servers, workstations;
- only certain IP addresses can be assigned to routers.

Such methods minimise conflicts due to IP address collision.

Figure 3.5 illustrates an example of three connected IP networks. Each network has its unique network number (network 1 – 23.x.x.x; network 2 – 188.96.x.x; network 3 – 192.168.1.0). The hosts on these networks are assigned IP addresses consisting of the network number (same for all hosts on the network) and a unique (in that network) host number. Host numbers in different networks can coincide.

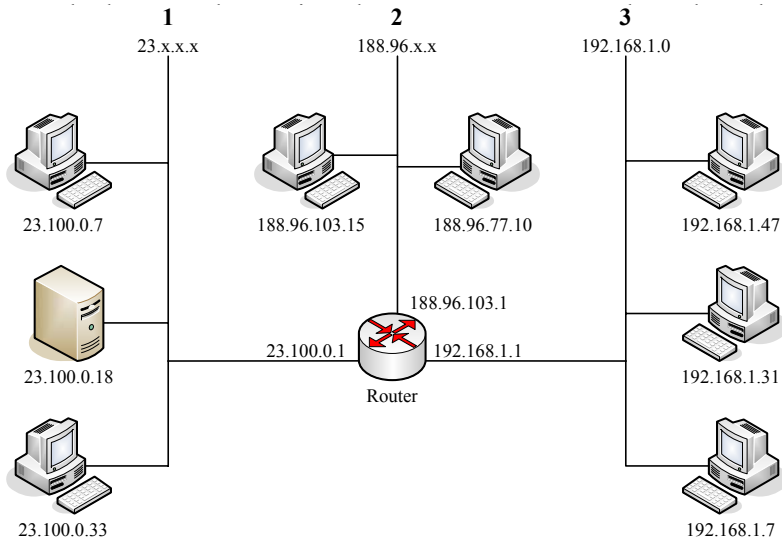


Fig. 3.5. An example of three connected IP networks

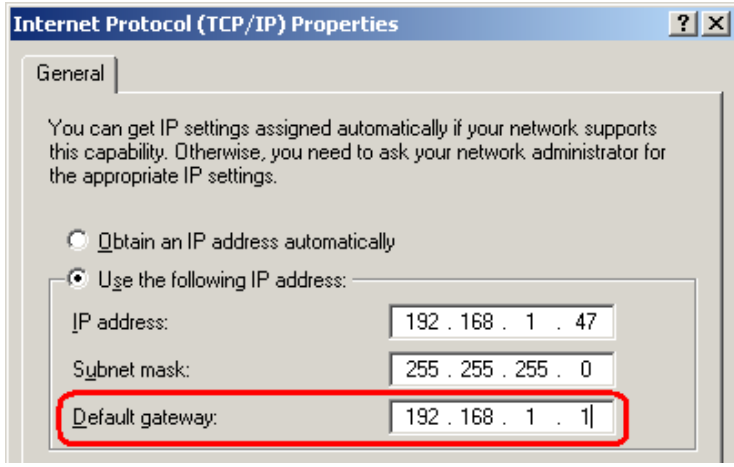


Fig. 3.6. *Windows XP* network configuration window

The usual separation of the IP address into the network number and the node number is related to the network class, which is deter-

mined according to the values of the leading bits. However, when using the A, B and C classes according to their description, the allocation of IP addresses is ineffective, because there are no real networks which would require A (more than 16 million node addresses) or B (over 65 thousand) class addresses. Therefore another method is used, which permits identification of which part of the IP address is used for network identification, and which – for host identification. A subnet mask is used for this purpose.

A subnet mask is a 32 bit number, used in pair with an IP address and is necessary to identify the network and host numbers. In binary form bits which designate the network number will equal 1, and bits which designate the host number will equal 0. Typical masks corresponding to A, B and C class networks are presented in Table 3.4.

Table 3.4. Default subnet masks

Class	Binary form	Dotted decimal notation
Class A	11111111 00000000 00000000 00000000	255.0.0.0
Class B	11111111 11111111 00000000 00000000	255.255.0.0
Class C	11111111 11111111 11111111 00000000	255.255.255.0

Logic operation AND is used to determine the network and host numbers. First the IP address and the subnet mask are written in binary form, and after that they are logically multiplied. The result of logical multiplication shows the network number:

IP address:	88.134.61.99	01011000 10000110 00111101 01100011
		&
Mask:	255.255.0.0	11111111 11111111 00000000 00000000
Network number:	88.134.0.0	01011000 10000110 00000000 00000000

Subnet mask is also used to determine if a network host to which a packet is sent is in the same or a different network. Before

sending a packet the IP addresses of the recipient and the sender are logically multiplied by the subnet mask. If the result corresponds, then the sender and the recipient are on the same network. Otherwise the packet is sent to the router of the network on which the sender is located.

Benefits of using masks:

- restructuring of the local network without changing the outer network configuration. The network can be divided into smaller parts using the same external IP address;
- minimisation of network load. Usage of masks allows minimisation of network load by limiting sending of broadcast packets;
- simpler administration. A network divided into smaller parts is easier to maintain;
- better security. Usage of masks allows the organisation to separate local networks to which connection from external networks could be forbidden.

Division of a network into subnets

A subnet is a segment of a network, in which IP addresses with a common network number are used. Division of a network into subnets allows division of one large network into smaller logic networks.

As we know, an IP address consists of a network number and a number of a host on that network. A subnet is established by borrowing bits that are allocated for host numbering. The subnet mask shows how many bits were borrowed: all bits, allocated for numbering networks and subnets equal 1, and the rest equal 0. Division of a network into subnets increases the number of possible networks, but the number of hosts in these networks decreases. Each new subnet established has all attributes characteristic to a network: subnet IP address, IP addresses allocated for hosts and a broadcast address (Fig. 3.7).

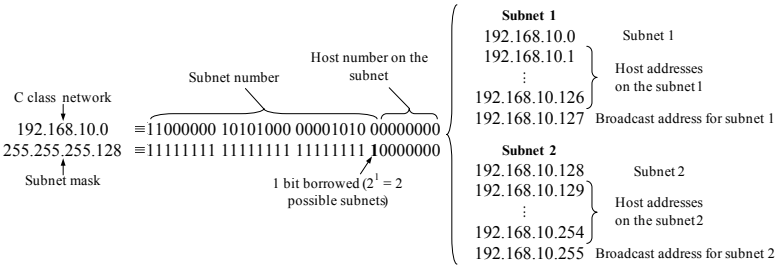


Fig. 3.7. C class network 192.168.10.0 division into two subnets

Before dividing a network into subnets it is necessary to determine:

- the number of planned network segments;
- the number of host addresses in each of the segments. Considering the possibilities for network expansion it is recommended to leave unused addresses in each of the segments.

When the number of networks and hosts that they contain is known a subnet mask is determined, as well as numbers of each of the subnets and the subnet host address range.

The subnet mask can be found according to the algorithm below:

1. The number of necessary segments N is subtracted by one (N – 1) and the received value is changed into binary format.
2. The number of bits of this binary value is calculated.
3. These bits are replaced by 1's and the value is supplemented by 0's to the right side to receive 1 byte. The resulting binary number is changed into a decimal value.

Example for Class C:

Number of necessary subnets N	7
N – 1	6
Binary value	110 (3 bits)
Change into 1's and additions to form a byte	11100000
Mask expression	111111111 11111111 11111111 11100000
Decimal value	255 . 255 . 255 . 224

Irrelevant of whether the network will have to be divided into 5, 6, 7 or 8 subnets, a network mask dividing a network into 8 subnets will have to be used anyway, because the number of possible subnets equals 2^n ($2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 16$, …, where n is the number of bits allocated for a subnet). When dividing a network into 10 subnets a network mask dividing a network into 16 subnets will have to be used.

Below is a change table for dividing class A, B and C networks into subnets when the network mask uses 1 to 8 bits. More than 8 bits can be used for the subnet mask of classes A and B.

Table 3.5. Dividing class A, B and C networks into subnets

Class A			
Number of subnets	Borrowed bits	Subnet Mask	Number of hosts per subnet
2	1	255.128.0.0	8 388 606
4	2	255.192.0.0	4 194 302
8	3	255.224.0.0	2 097 150
16	4	255.240.0.0	1 048 574
32	5	255.248.0.0	254 286
64	6	255.252.0.0	262 142
128	7	255.254.0.0	131 070
256	8	255.255.0.0	65 534
Class B			
Number of subnets	Borrowed bits	Subnet Mask	Number of hosts per subnet
2	1	255.255.128.0	32 766
4	2	255.255.192.0	16 382
8	3	255.255.224.0	8190
16	4	255.255.240.0	4094
32	5	255.255.248.0	2046
64	6	255.255.252.0	1022
128	7	255.255.254.0	510
256	8	255.255.255.0	254

Class C			
Number of subnets	Borrowed bits	Subnet Mask	Number of hosts per subnet
2	1	255.255.255.128	126
4	2	255.255.255.192	62
8	3	255.255.255.224	30
16	4	255.255.255.240	14
32	5	255.255.255.248	6
64	6	255.255.255.252	2
128	7	255.255.255.254	2*
–	8	–	–

* may be used only for point-to-point link between two hosts.

When a subnet mask is determined, every subnet number and subnet host address ranges are found. This can be done by using the method below.

1. Subnet increment P is found according to formula (3.1)

$$P = 2^{8-z}, \tag{3.1}$$

here z is the number of bits borrowed to make the subnet mask. If more than 8 bits are used for the subnet mask of classes A and B, then the number of bits equal to 1 of the trailing value byte is taken.

2. Possible subnet numbers are sequentially written starting with the first subnet number and adding the subnet increment (the subnet increment is added until it exceeds 255).
3. By increasing every following subnet number value by 1 the beginning of the host address range of this subnet is found, and by subtracting the value of the next subnet number by two the end of the host address range of that subnet is found.

Example: We have a class C network address 192.168.1.0 and a subnet mask, to which 2 bits are allocated, i.e. the subnet mask equals 255.255.255.192.

1. We use formula (3.1) to find the network increment P: $P = 2^{8-2} = 2^6 = 64$;
2. Beginning with the first subnet number and by adding the subnet increment, possible subnet numbers are sequentially written:

0	192.168.1.0
+64	
=64	192.168.1.64
+64	
=128	192.168.1.128
+64	
=192	192.168.1.192

By subtracting 2 from the increment we will find the number of possible nodes in each subnet. Two is subtracted because the first subnet address indicates the subnet number and the last address indicates the broadcasting address in the subnet.

Since 2 bits are allocated to the subnet mask in this case, the number of possible subnets equals 4 ($2^2 = 4$). The number of host addresses in each subnet equals 62 (64 – 2).

192.168.1.0	Subnet 1
192.168.1.1	First host address in subnet 1
…	
192.168.1.62	Last host address in subnet 1
192.168.1.63	Broadcast address for subnet 1
192.168.1.64	Subnet 2
192.168.1.65	First host address in subnet 2
…	
192.168.1.126	Last host address in subnet 2
192.168.1.127	Broadcast address for subnet 2
192.168.1.128	Subnet 3
192.168.1.129	First host address in subnet 3
…	
192.168.1.190	Last host address in subnet 3
192.168.1.191	Broadcast address for subnet 3

192.168.1.192	Subnet 4
192.168.1.193	First host address in subnet 4
…	
192.168.1.254	Last host address in subnet 4
192.168.1.255	Broadcast address for subnet 4

The subnet mask is often provided as a prefix written after the IP address, i.e. after a slash, e.g. 192.168.15.77/28. This prefix indicates how many bits of the mask are allocated for network identification. Since the subnet mask is 32 bits long, in this case we see that 28 bits are allocated to the network number, and the remaining 4 are allocated to the host number. Thus we could get the network mask by writing a 28 digit long series of 1's and by supplementing it with four zeros up to 32 bits:

$$11111111\ 11111111\ 11111111\ 11110000_2 = 255.255.255.240_{10}$$

The prefixes of class C subnets are shown in Table 3.6.

Table 3.6. Prefixes of class C subnets

Prefix	Subnet mask	Number of subnets	Number of hosts per subnet	Total number of hosts
/24	255.255.255.0	1	254	254
/25	255.255.255.128	2	126	252
/26	255.255.255.192	4	62	248
/27	255.255.255.224	8	30	240
/28	255.255.255.240	16	14	224
/29	255.255.255.248	32	6	192
/30	255.255.255.252	64	2	128
/31	255.255.255.254	128	2*	256

* may be used only for point-to-point link between two hosts.

Work assignment and methodical guidelines

1. Read attentively the theory section of laboratory work. Before starting the work discuss all obscure questions with the lecturer.

2. Convert numbers from binary format to decimal:

Binary value	Decimal value
10101000	
10000111	
11011100 10010111	
00100111 01010001	
10110110 00110010 01101101 00001111	
00000000 10010110 01011000 11110101	

3. Convert numbers from decimal format to binary:

Decimal value	Binary value
168	
202	
131.187	
88.103	
45.197.70.234	
80.243.175.55	

4. Determine to which IP address classes the following addresses belong:

IP address	Class
201.225.14.180	
135.67.39.232	
77.39.81.52	
190.117.92.3	
223.24.182.85	
22.47.18.167	

5. Determine the network and host numbers of the IP address:

IP address	Network number	Host number
e.g. 61.137.203.178	61.	137.203.178
129.207.77.137		
196.143.149.210		
1.46.196.39		
173.233.252.206		
59.13.194.33		

6. Write:
   - class A IP address of the 59th network, 901665th computer;
   - class B IP address of the 9963rd network, 46118th computer;
   - class C IP address of the 43018th network, 10th computer.

7. According to the provided IP addresses determine the address class to which it belongs, the number of network in that class and the number of the host in that network.

IP address	Class	Network in the class	Host in the network
e.g. 61.137.203.178	A	61	9030578
184.37.137.75			
96.33.154.222			
202.108.179.78			

8. Explain why the below IP addresses are false.

IP address	Explanation
192.168.256.10	
225.214.215.40	
127.66.27.239	
283.236.198.190	

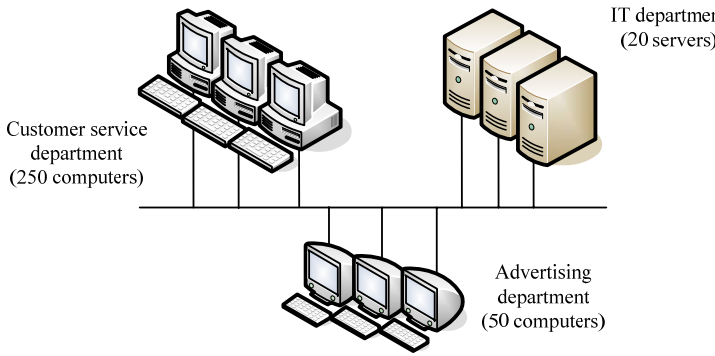
9. Determine the location of the recipient of information – is it on the same or a different network, if:

Source IP address	11001101 11000110 00100111 00001111
Subnet mask	11111111 11111111 11111111 00000000
Destination IP address	11001101 11000110 01100111 00001111

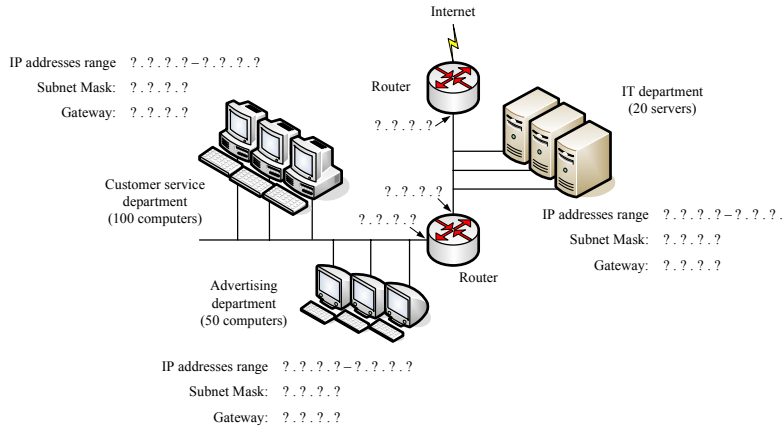
Source IP address	11000000 10101000 00001010 01100111
Subnet mask	11111111 11111111 11111111 00000000
Destination IP address	11000000 10101000 00001010 00001111

Source IP address	10110100 00101000 01110100 01001001
Subnet mask	11111111 11111111 00000000 00000000
Destination IP address	10110100 00101000 00001010 01100111

10. Determine the IP address class needed for the network of an organisation illustrated in the figure below. Provide IP address ranges for each department of the organisation.



11. Provide the following information on the network in the figure below: IP address ranges for each department, subnet masks, IP addresses of network gateways and IP addresses of routers.



12. Determine how many bits are allocated to establish subnets based on the subnet masks provided:

Class	Subnet mask	Number of bits
B	255.255.192.0	
C	255.255.255.240	
B	255.255.255.240	
A	255.128.0.0	

13. Based on Table 3.5, find the subnet mask if:
    - class C network needs to be divided into 3 subnets with 50 host addresses in each;
    - each subnet of a Class C network must have 8 to 12 host addresses;
    - class B network needs to be divided into 60 subnets, each containing 1000 host addresses.

14. Based on the provided IP address and the subnet mask find the subnet IP addresses and host address ranges in the subnets.

Network address: 192.168.73.0
Subnet mask: 255.255.255.192

Subnet	Subnet address	Host addresses range
1	192.168.73.0	
2		
3		
…	…	…

Network address: 172.23.0.0
Subnet mask: 255.255.224.0

Subnet	Subnet address	Host addresses range
1	172.23.0.0	
2		
3		
…	…	…

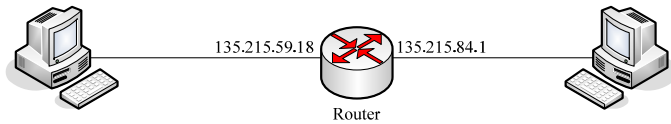
15. Find the subnet mask according to the provided range of IP addresses.

IP address range		Subnet mask
From	To	
97.8.0.1	97.15.255.254	
137.93.128.1	137.93.159.254	
211.45.61.97	211.45.61.126	
28.12.192.1	28.12.255.254	
152.59.10.1	152.59.10.254	

16. What is the subnet number to which a host the IP address 193.219.146.19/26 belongs? How many hosts can there be on such a subnet and what IP addresses can be assigned to these hosts? What is the broadcast address of the subnet?

17. Find 7th subnet number of class B 172.16.0.0/22 network, its broadcasting address and the subnet host IP address range.
18. Explain what is wrong in the following figures and why it is wrong. What influences the network connections? What should be the correct settings of network configuration parameters?

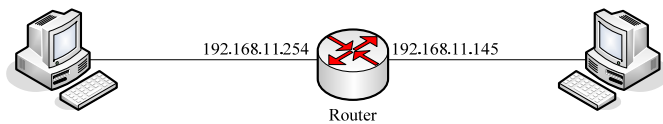
A



IP address: 135.215.59.50
Subnet mask: 255.255.0.0
Default gateway: 135.215.59.18

IP address: 135.215.84.50
Subnet mask: 255.255.0.0
Default gateway: 135.215.84.1

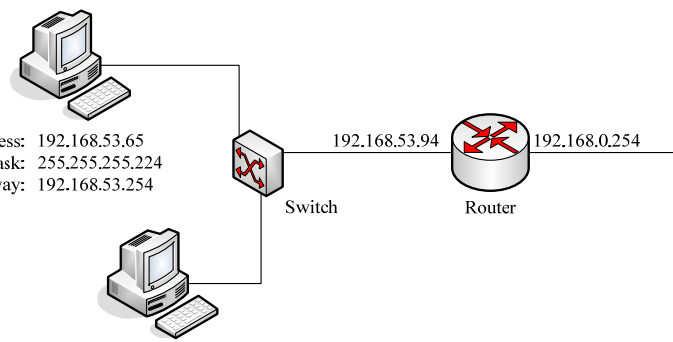
B



IP address: 192.168.11.15
Subnet mask: 255.255.255.240
Default gateway: 192.168.11.254

IP address: 192.168.11.143
Subnet mask: 255.255.255.240
Default gateway: 192.168.11.145

C



IP address: 192.168.53.65
Subnet mask: 255.255.255.224
Default gateway: 192.168.53.254

IP address: 192.168.11.97
Subnet mask: 255.255.255.240
Default gateway: 192.168.0.254

19. Assign IP addresses to the local computer network designed in the laboratory work 2.

Content of report

1. Work objective.
2. The results of Tasks 2–19.
3. General conclusions of the work.

Review questions and problems

1. Name the layers of the OSI model and explain their purpose.
2. Which layers of the OSI model does the TCP/IP protocol stack occupy?
3. What parts do constitute the IP address? What are their uses?
4. What are IP address classes? What types of such classes are there and how do they differ?
5. Explain what a subnet mask is.
6. How is the location of the recipient of an IP packet determined on a computer network?
7. Explain what a subnet is.
8. What are the benefits of dividing a computer network into subnets?
9. How is a subnet mask determined?
10. How are the subnet numbers determined?
11. How are the subnet address ranges determined?
12. How are the subnet broadcast addresses determined?
13. Is a router necessary to connect subnets?
14. What mask is necessary to divide class A network into 32 subnets?

Literature

Casad, J. 2008. *Sams Teach Yourself TCP/IP in 24 Hours*. 4th edition. Sams. 456 p. ISBN-13: 978-0672329968.

Held, G. 2000. *TCP/IP Professional Reference Guide*. Auerbach Publications. 256 p. ISBN-13: 978-0849308246.

Subnet Mask Calculator. http://www.subnetmask.info/.

Laboratory work 4

Application of Windows OS Built-in Networks Diagnostic Tools

Objectives

Get acquainted with Windows OS command-line network diagnostic, monitoring and management tools and their application for network troubleshooting.

Basic knowledge and theory

When solving problems in computer networks, command-line tools can be very useful. These tools allow you to quickly find and resolve the problem. Next, we will look at the basic and most commonly used Windows operating system (OS) command-line tools for computer network diagnostics.

Windows OS command-line tools

Windows OS command-line can be accessed in two ways. First, the command-line can be called from the program list in **Start → All Programs → Accessories → Command Prompt**. The second method is faster and is suitable if the command button *Run...* is present in the *Start* menu. When you press the *Run...* command button, a window will open where you must enter *cmd*, and press the *OK* button.

If you want to adjust the command that was recently executed, it is not necessary to re-enter it. It is enough to press the up arrow keyboard key and the command that was entered before will be shown again. *cls* command clears the screen.

All command-line tools syntax is unified. For example, when the parameter */?* is used, it will always show support for a specific tool. More detailed information about the tool can be found through

the Windows OS support system (**Start → Help and Support**), which will give you all the additional parameter explanations and tool examples.

hostname – displays the computer name. Have no additional parameters.

ipconfig – provides information about the network interface(s) configuration, updates the settings, assigned by the *Dynamic Host Configuration Protocol* (DHCP), and the *Domain Name System* (DNS) server list.

If *ipconfig* command is entered with no parameters, the basic network configuration is shown i.e. computer's IP address, subnet mask and gateway IP address. *Ipconfig* command entered with the */all* parameter will show all the information about the network settings.

getmac – displays the computer's network adapters physical (MAC) addresses. The MAC addresses can also be found by using the command *ipconfig /all*. This address consists of two parts, each containing 3 bytes. The first three bytes refer to the manufacturer, and the remaining 3 bytes are assigned by the manufacturer.

arp – displays and allows modification of the *Address Resolution Protocol* (ARP) entries table. This protocol is designed to identify the physical network adapter address by using the computer IP address. In local area networks, data transmission to the recipient is not based on IP addresses, but on the MAC addresses.

ping TargetName – the main command, which is used to check whether the network host in the local network can be accessed. First at all, the *Internet Control Message Protocol* (ICMP) *Echo Request* message is sent. If the recipient, for whom the message was sent, can be reached, it responds with an *Echo Reply* message. ICMP can be used in *Denial of Service* (DoS) attacks. For security reasons, in many cases, ICMP *Echo Request* messages that are sent via the *ping* command are blocked by the firewalls, so checking if a computer that is in another network is reachable becomes impossible. In addition, a computer that is on your net-

work also can not respond to ICMP *Echo Request* message if the firewall prohibits the ICMP *Echo Reply* messages.

netstat – displays the active TCP connections, open ports, sent and received packets statistics, the IP routing table. When *netstat* command is executed with no parameters, the active TCP connections are shown.

tracert – displays the packet forwarding path to the recipient and the duration of the trip.

nslookup – allows the user to send a query to the DNS name server and by using the sites name, find out its IP address and vice versa.

nbtstat – displays the computer's NetBIOS names, active connections, and records stored in the cache.

route – used to view your computer's routing table, insert a new route or remove an existing one.

Work assignment and methodical guidelines

1. Use command line tools named in basic knowledge and theory section to complete the following tasks. To view help information for particular command line tool, at the command prompt, type the following *CommandName* /?. Additional information can be found in the *Windows Help and Support Center* (**Start → Help and Support**). Illustrate the results of the work by examples of the carried out commands.
2. The use of *ipconfig* command.
   2.1. Determine your computer's TCP / IP network configuration parameters, and answer the following questions:
      2.1.1. Windows IP Configuration:

Host Name:	
IP address:	
Subnet Mask:	
Default Gateway:	
Physical Address:	
DNS Servers:	

2.1.2. What IP address class does your computer IP address belongs? Specify this IP address network number and host number.

2.1.3. Indicate to which class of IP addresses the DNS server's IP address belongs.

2.1.4. Does the computer use a static IP address or DHCP server assigns him one?

2.1.5. From what parts is the network adapter's physical address (MAC) made off?

2.2. Compare your computer's TCP/IP network configuration parameters to another computer, which is on the same network, parameters. What are the similarities and difference between them?

2.3. Visit several web sites in foreign servers. Review the entries in the DNS server names cache and give three record examples.

3. The use of *ping* command.

3.1. Using the *ping* command send an *Echo Request* query to an adjacent computer. Determine the *Echo Reply* response delay time.

3.2. Determine the *Echo Reply* responses delay time dependence on the amount of data that was sent by *Echo Request* query and draw a graph.

The amount of data sent, KB	**Delay time, ms**
1	
5	
10	
20	
30	
40	
50	
60	

3.3. Send *Echo Request* query with *-i* (packet lifetime in the network) parameter set to 1, to a computer that is in another subnet. Make conclusions based on the obtained results.

3.4. Using the *ping www.vgtu.lt* command, determine when the intermediate nodes of the network received the *Echo Request* and *Echo Reply* messages.

4. Viewing and modification of the ARP table.

4.1. Send an *Echo Request* query to an adjacent computer and view the ARP table. Does the table contain a record linking the adjacent computer's IP address with its MAC address?

4.2. Delete the adjacent computer record.

4.3. Add an ARP entry linking the adjacent computer's IP address with a fictional MAC address, and try to send Echo Request query again. Draw conclusions.

4.4. Delete entries in the ARP table.

5. What command allows you to determine how much and what kind of network router does the network packet that you sent passes from your computer until it reaches *www.vgtu.lt* web site server. What is the packet delay time?

6. What command allows the user to send a query to the DNS name server and by using the name of the web site find out its IP address and vice versa? By using the command locate *www2.el.vgtu.lt*, *www.google.lt*, *www.litnet.lt* sites IP addresses. What is the domain registered on this IP address: 72.163.4.161?

7. Determine your public network IP address which is visible for others when you browse on the Internet.

8. The use of *netstat* command.

8.1. Determine all active TCP connections and open TCP and UDP ports. What kinds of applications are related to the open TCP ports?

8.2. View the general sent and received packet statistics.

8.3. View the ICMP protocol sent and received packets statistics. Send 5 *Echo Request* queries to a default gateway ad-

dress and then review the ICMP protocol sent and received packets statistics. Draw conclusions.

9. Viewing and modifying Windows IP routing table:
   9.1. Review your computer's routing table and analyze the records.
   9.2. Add a path to the network 10.100.15.0/24 via an adjacent computer.
   9.3. Check did you succeeded in adding the route.
10. Using the *net use* command connect the shared folder and read the content of the text file in it. Check how the communications table has changed.

Login data:
Server IP address: *192.168.10.254;*
Folder name: *shared_folder*
User name: *user*
Password: *4lab*

Content of report

1. Work objective.
2. The results of the *ipconfig* command use (Task 2).
3. The results of the *ping* command use (Task 3).
4. ARP table inspection and modification results (Task 4).
5. The answers to the Tasks 5–7.
6. The results of the *netstat* command use (Task 8).
7. Windows IP routing table inspection and modification results (Task 9).
8. The results of the *net use* command use (Task 10).
9. General conclusions of the work.

Review questions and problems

1. What command in the Windows OS is used when the computer TCP/IP network configuration parameters need to be identified?

2. What does the command *ipconfig /all* mean and what kind of information does it provide?
3. What do the first three hexadecimal pairs of the network physical address (MAC) mean?
4. What does the *ping* command do?
5. Why, when the *ping* command is being executed to a remote computer, we receive the response "*timed out*"?
6. Which protocol functions does the *ping* command use?
7. What does the *netstat* command display?
8. For what purpose is the *tracert* command used?
9. Which command in the Windows OS displays the computer routing table?
10. What is the purpose of the *arp* table?
11. What command is used to find the IP address of web server?

Literature

Mueller, J. P. 2010. *Windows Command Line Administration Instant Reference*. Sybex. 576 p. ISBN-13: 978-0470650462.

Laboratory work 5

Network Packet Monitoring and Analysis Tools

Objectives

Learn how to use network packet monitoring and analysis tools. Analyse the operation principles of the main network layer (IP) and transport layer (TCP and UDP) protocols.

Basic knowledge and theory

Network analysis (also known as traffic analysis, protocol analysis, sniffing, packet analysis, eavesdropping) is the process of capturing network traffic and inspecting it closely to determine what is happening on the network. A network analyzer captures all the traffic that is going across the network and displays it in readable format.

Network analyzers typically are used to:

- troubleshoot problems on the network,
- detect and identify malicious software,
- capture and decode data on a network,
- analyze the operations of applications,
- generate and display statistics about the network activity.

Network analyzers capture network data passively. It observes messages being sent and received over the network, but never sends packets itself. Instead, a network protocol analyzer receives a copy of packets being sent and received over the network. Packets are captured by placing the network interface in promiscuous mode. In promiscuous mode, all packets are captured regardless of their destination address.

Wireshark is rich of features, open source, commercial-quality network analyzer and will be used in this and following laboratory works to help analyse and understand the basic operation principal of main network protocols.

The *Wireshark* graphic user interface has three panes where information about captured packets are displayed (Fig. 5.1):

Packet List pane displays a one line summary for each packet captured, including the frame number within the capture, the relative time the packet was captured, the source and destination of the packet, the highest level protocol that was decoded, and protocol-specific information contained in the packet.

Packet Details pane provides details (in a treelike structure) of each layer contained in the captured packet selected in the *Packet List* pane. The amount of detail displayed can be collapsed and expanded to show all of the information collected about an individual packet.

Packet bytes pane displays the entire contents of the captured frame, in both ASCII and hexadecimal format.

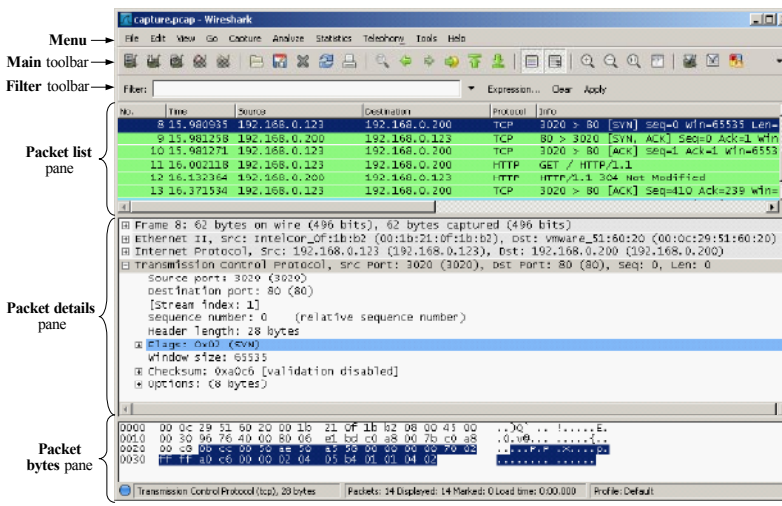


Fig. 5.1. *Wireshark* main window

Wireshark has two main types of filters: *Capture filters* and *Display filters*. Capture filters are used to capture only packets that are relevant to the particular problem. This allows to reduce the

amount of traffic captured. Display filters provide a powerful syntax to sort traffic that is already captured. Only packets that match the display filter string are displayed in the packet list pane.

In addition to *Wireshark* there are some other tools that are useful for packet analysis and will be used in this laboratory work: *Windump*, *Netcat* and *Nemesis*. A brief description of these tools is provided below. How to use these tools are described in practical section of this laboratory work.

WinDump is the Windows version of *Tcpdump* which is the oldest and most commonly used command line network packet analyzer. It was originally developed to analyze TCP/IP performance problems. *Tcpdump* is used to analyze network behaviour, performance and applications that generate or receive network traffic. Capture files created by *Tcpdump* can be read and analysed by *Wireshark*.

Netcat is a command line, distributed freely, feature rich network debugging and analysis tool. Its functionality is helpful as both a standalone program and a back end tool in a wide range of applications. Some of *Netcat's* major features include outbound or inbound connections, TCP or UDP, to or from any ports, port redirection, transferring files, tunneling mode, port scanning and others.

Nemesis is a command line custom packets construction and injection tool. *Nemesis* is well suited for testing software configuration and learning network protocols. *Nemesis* allows to generate packets and packet payload for ARP, DNS, Ethernet, ICMP, IGMP, IP, RIP, TCP, and UDP. Using the IP and the Ethernet injection modes, almost any custom packet can be crafted and injected.

Internet Protocol

Internet Protocol (IP) is the network layer protocol that provides logical addressing and routing capabilities. The IP is a protocol that enables the connection of individual networks into a worldwide Internet. IP is a connectionless, unreliable packet delivery service. Connectionless means that there is no continuing connection be-

tween the end points that are communicating. Each packet that travels through the Internet is treated as an independent of any other packet. IP is unreliable because there is no guarantee that a packet gets delivered. Any required reliability must be provided by the higher layer protocols, such as the Transmission Control Protocol (TCP).

The most widely used version of IP today is version 4 (IPv4). However, IP Version 6 (IPv6) is also beginning to be supported. Each network interface has at least one unique IP address, which for IPv4 is 4 bytes and for IPv6 is 16 bytes. The basic unit of transferred data in the IP is called an IP packet. The IP packet may be divided into two pieces: the *header* and the *payload*. The header contains addressing and control fields, while the payload carries the actual data. The minimum size for an IP header is 20 bytes. However, a header can also contain optional entries that can make it longer. The IPv4 datagram structure is shown in Figure 5.2. The most significant bit is numbered 0 at the left, and the least significant bit of a 32 bit value is numbered 31 on the right.

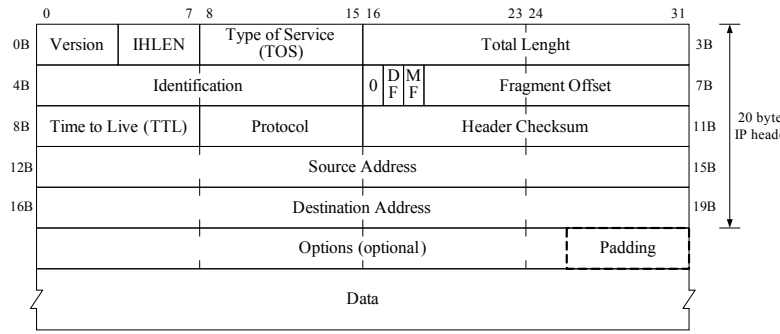


Fig. 5.2. IP header

The *Version* is the first entry in the IP packet header. This 4 bit field specifies the IP protocol version of the datagram. The purpose of this field is to ensure compatibility between devices that may be running different versions of IP.

The *Internet Header Length* (IHLEN) is 4 bit field which indicates length of the IP header in 32 bit words. The minimum header length is five 32 bit words.

The *Type of Service* (TOS) field for a long time did not have a practical use. Therefore its meaning has been redefined for use by a technique called Differentiated Services (DS). The importance of this entry increased with the requirement to guarantee bandwidth, especially by applications requiring sound and video transmission.

The *Total Length* field specifies the total length of the IP packet, in bytes.

The *Identification* field uniquely identifies each datagram sent by a host. This field, together with the DF (*Don't Fragment*) and MF (*More Fragments*) flags and *Fragment Offset* field, is used by the datagram fragmentation mechanism. If the DF bit is set to 1, fragmentation is not allowed. If the MF bit is set to 1, it specifies that this is not the last fragment. *Fragment Offset* field is 13 bits long and specifies the offset of a particular fragment relative to the beginning of the original unfragmented IP packet.

The *Time to Live* field specifies how long the datagram is allowed to wander on the network, in terms of router hops. This field is decremented by one each time the datagram is processed by a router. If the TTL field reaches 0, the datagram must be discarded.

The *Protocol* field contains the identification number of the transport layer protocol that is encapsulated in the IP packet. Some common protocol values are: ICMP – 1; TCP – 6; UDP – 17.

The *Header Checksum* field is used to ensure the integrity of the IP header. The data portion of the packet is not included in the packet checksum.

Both the *Source Address* and *Destination Address* fields are 32 bits in length under IPv4. The source address represents the originator of the datagram, while the destination address represents the recipient.

The *Options* field supports a number of optional header settings primarily used for testing, debugging, and security. The IP options field may vary in length. The Padding field provides additional zero bits so that the total header length is an exact multiple of 32 bits.

The Internet Protocol is defined in the Request For Comment (RFC) document number 791.

Transmission Control Protocol

The Transmission Control Protocol (TCP) is connection-oriented, reliable transport layer protocol. Connection-oriented means that the two applications using TCP must establish a TCP connection with each other before they can exchange data. Reliability is provided by checksums, data sequence numbers and acknowledgements. If the acknowledgement is not received within a timeout interval, the data is retransmitted. At the receiver end, the sequence numbers are used to correctly order TCP data units (called segments) that may be received out of order and to eliminate duplicates. All TCP segments carry a checksum, which is used by the receiver to detect errors with either the TCP header or data. Additionally TCP manages transmission of data in a diverse quality network through the use of flow control. Flow control stops buffer overruns and network congestion.

TCP uses a "*three-way handshake*" mechanism to establish the connection. The TCP handshake occurs in three separate steps, as shown in Figure 5.3.

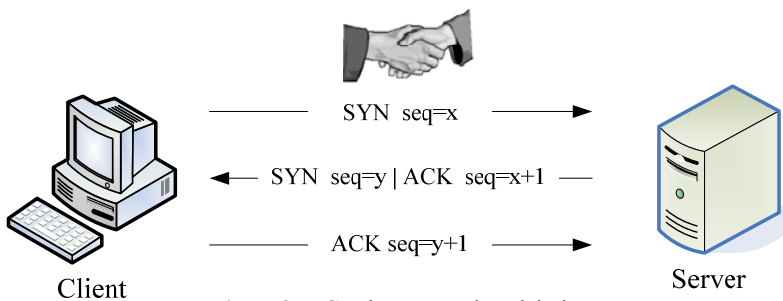


Fig. 5.3. TCP three-way handshake

Suppose that the client wants to establish the connection to the server. First the client sends a TCP segment to the server with SYN flag set to 1 (known as TCP SYN request), and some initial sequence number (in this case “x”) in the sequence field. This initial packet contains no application layer data. The server responds to this request by sending a similar segment with the SYN flag set and its own initial sequence number (in this case “y”) along with the ACK flag set which indicates that the next expected byte from client should contain data starting with sequence number x+1. Finally, the client finishes the connection establishment process by sending an acknowledgement segment with ACK flag set and acknowledgement number y+1 in the acknowledgement field. Once this process is completed, both devices may exchange data.

The TCP segment structure is shown in Figure 5.4. The TCP segment consists of header fields and a data field.

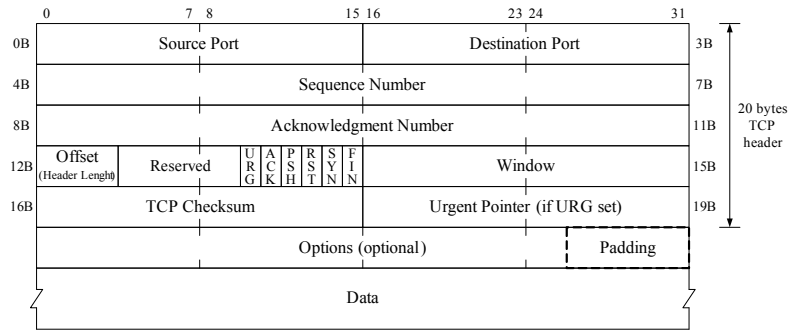


Fig. 5.4. TCP header

The *Source Port* and *Destination Port* fields are each 16 bits in length. Most applications use the destination port number to denote a particular process or application, and either set the source port field value to a random number greater than 1024 or to zero. These two fields plus the source and destination IP addresses, unambiguously identify the particular connection on the Internet at any given time.

The *Sequence number* is the sequence number of the first byte in this particular segment. That field is 32 bits in length and provides the mechanism for ensuring that missing or miss ordered packets are noted or identified.

The *Acknowledgment number* expresses the number of the next byte that the destination is ready to accept. This is therefore the sequence number plus 1 of the last successfully received byte of data.

The data *Offset* specifies the size of the TCP header in 32-bit words. The minimum size header is 5 words and the maximum is 15 words thus giving the minimum size of 20 bytes and maximum of 60 bytes.

The *Flags* field contains six flags. URG flag is used to indicate whether the urgent pointer field is significant. ACK flag is used to indicate that the value carried in the acknowledgment field is valid. PSH flag indicates that the receiver should pass the data to the receiving application immediately. RST is used to reset the connection. SYN is used within the connection start up phase, and FIN is used to close the connection in an orderly fashion.

The *Window* field provides TCP with the ability to regulate the flow of data between source and destination. The Window field indicates the maximum number of bytes that the receiving device can accept.

The TCP *Checksum* is used to check the integrity of the header and data.

The *Urgent Pointer* is valid only if the URG flag is set. The value in this field acts as a pointer to the sequence number of the byte following the urgent data.

The purpose of *Options* field is to enable TCP to support various options, e.g. *Maximum segment size*, *Window scale*, *Timestamp* and others. The TCP header *Padding* is used to ensure that the TCP header ends and data begins on a 32 bit boundary.

The Transmission Control Protocol is defined in the RFC 793.

User Datagram Protocol

User Datagram Protocol (UDP) is another transport layer protocol commonly used in TCP/IP networks. UDP is simpler and faster alternative to TCP. Unlike TCP, UDP is connectionless protocol. This means that an application using UDP can send its data in the form of IP packets without establishing a connection to the destination. UDP does not guarantee reliable communication. There is no guarantee that the datagram will be delivered to the destination, or arrive in a different order from the one in which they were sent. It's up to the application layer to process any errors and verify correct delivery. UDP is primarily used by applications that transmit relatively short segments (e.g. *Domain Name Service* – DNS, *Simple Network Management Protocol* – SNMP) or require low-latency (e.g. *Voice over IP* – VoIP).

As shown in Figure 5.5 a UDP datagram header consists of only four fields of two bytes each: source port number, destination port number, datagram size and checksum.

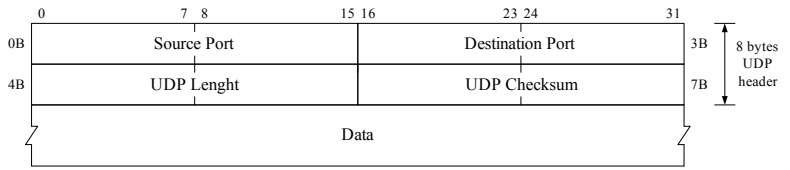


Fig. 5.5. UDP header

The *Source Port* specifies the port used to transmit the packet.

The *Destination Port* specifies the port to which the packet will be transmitted.

The UDP *Length* field specifies the length in bytes of the entire datagram, including header and data.

The use of *Checksum* field is optional and its value is set to 0 if the application does not require a checksum.

The User Datagram Protocol is defined in the RFC 768.

Work assignment and methodical guidelines

1. Read attentively the theory section of laboratory work. Before starting the work discuss all obscure questions with the lecturer.
2. The use of *Wireshark version 1.6* network protocol analyzer.
   2.1. Start up the *Wireshark* network protocol analyzer. To begin packet capture click **Capture → Options…** in *Wireshark* menu, then choose an appropriate interface that corresponds to the LAN and click *Start*.
   2.2. Enter the following URL *www.vgtu.lt* into your favourite browser. After the page has displayed, stop *Wireshark* packet capture. All captured packets are displayed in the *Wireshark* top (*Packet List*) pane. The middle and bottom panes display packet details and contents.
   2.3. Type in *http* into the display filter specification toolbar and click *Apply*. Only the HTTP messages will be displayed in the *Packet List* pane. Note: use fragments of screenshots to show your work for this and the following tasks. Use *http. request* filter to see only HTTP request messages.
   2.4. Use *Wireshark* to find the IP address of *www.vgtu.lt* web site.
   2.5. Type in *tcp* into the display filter to display only TCP packets. Are there any other protocol packets into *Packet List* pane except TCP? Explain why?
   2.6. Use *ip.src* == <*IP address*> filter to display packets sent only from your computer. What filter should be used to show packets with destination IP address of your computer?
   2.7. Display only the packets sent to TCP port 80. To complete this task, type in *tcp.* into the display filter and choose an appropriate filter expression from drop down menu. Write down which expression you chose.
   2.8. To display the packets with specified sequence of characters use additional operator *contains*. For example, to display TCP packets which contain string “public” use the *tcp contains public* filter.

2.9. Multiple filters are combined using standard C language operators (==, <, >, !=, &&, || et al.). For example: *!(ip.src ==192.168.0.100) && ip.dst == 192.168.0.99 && tcp*. Also English-like term can be used to describe these operators, e.g. *eq* (==), *ne* (!=), *and* (&&), *not* (!). In this case the previous example can be rewritten as: *not (ip.src eq 192.168.0.100) and ip.dst eq 192.168.0.99 and tcp*

Type in *arp || dns* (*arp or dns*) to display only ARP and DNS protocol packets. What expression should be used to display only TCP protocol (without HTTP)?

3. Using the *Wireshark* statistics tool answer the following questions:
   3.1. What is the total number of packets captured in previous task? What was the average bandwidth used? Click **Statistics → Summary** in *Wireshark* menu.
   3.2. What number of the ARP, UDP and TCP protocol packets was captured? Click **Statistics → Protocol Hierarchy**.
   3.3. What is the number of packets sent from *www.vgtu.lt* web server to your computer and vice versa? Click **Statistics → Conversations** and choose the IPv4 tab.
   3.4. What was the most common TCP packet length? What was the least common TCP packet length? Click **Statistics → Packet Lenghts**. Type in *tcp* and click *Create Stat*.
4. Analysis of TCP three-way handshake. You will need the virtual machine and three tools (*Wireshark*, *Netcat*, and *Telnet* client) for this task. Remember to use fragments of screenshots to show your work.
   4.1. Open *VMware Workstation* desktop virtualization software. Then click **File → Open**, select the *vm_WinXP_net.vmx* file in the *c:\Documents and Settings\mikro\My Documents\!Networks* directory and click *Open*. After the *vm_WinXP_net* tab has opened click *Start this virtual machine* button. This will cause to start *Windows XP* virtual machine which

we will name *Guest*. The main computer on which the virtual machine is running we will name *Host*.

4.2. Open a Windows terminal on the Host computer. There are two ways to do it: 1) click **Start → All Programs → Accessories → Command Prompt**; 2) click **Start → Run**, type *cmd* and click *OK*.

4.3. Use *cd* command to browse to the *Netcat* location directory *c:\tools\nc*. Type in *nc -h* into Windows terminal to see all *Netcat* command line options.

4.4. Make the *Netcat* run in listen mode on 23 (*Telnet*) port and answer to *Telnet* negotiation. Type in *nc –l –p 23 –t –v* into terminal window. Option *–v* (verbose) tells *Netcat* to give a feedback on the connection process.

4.5. What MS-DOS command can be used to verify that *Netcat* is listening on the 23 port? Show the command and the output.

4.6. Before you start *Wireshark* on the Guest computer verify if the packet capture library *WinPcap* version 4.x is installed. Click **Start → Control Panel** and open *Add or Remove Programs* window. If there is no any or older version of *WinPcap*, browse to *c:\tools* directory there you will find a *WinPcap 4.x* and install it.

4.7. Start up the *Wireshark* on the Guest computer. Type in *eth. addr == MACaddressofGuest && tcp* into display filter toolbar and start packet capture.

4.8. Open a Windows terminal in the Guest computer ant type in *telnet HostIPaddress*. Then stop packet capture. You should see three TCP packets (three-way handshake) captured in the *Wireshark Packet List* pane. Analyse them and fill the following table:

Field	**Packet**		
	1st	**2nd**	**3rd**
Source IP address			
Destination IP address			
Source port number			

Destination port number			
Flags			
Relative Sequence Number			
Absolute Sequence Number*			
Relative Acknowledgment Number			
Absolute Acknowledgment Number*			

* – to view absolute sequence and acknowledgment numbers, select *Transmission Control Protocol* in the packet details (middle) pane and press right mouse button. Then choose *Protocol Preferences* and de-select *Relative sequence numbers*.

4.9. If the initial TCP sequence value from Guest is 0, why did Host respond with an acknowledgement of 1? How did the Host determine that value?

5. Analysis of TCP packets used to transfer data.

5.1. Start up the *Wireshark* packet capture on the Guest computer with the same display filter as in Task 4.

5.2. Your *Telnet* connection should be active from the previous task. If so, press Ctrl+]. Then the text *Welcome to Microsoft Telnet Client…* will appear on the terminal window. Type *help* to see supported commands.

5.3. Issue the *send* command to transfer your name. Then use the same command to transfer the string of only 4 characters, e.g. "vgtu". After that issue *quit* command to terminate *Telnet* communication and stop packet capture.

5.4. Analyse captured packets and answer following questions:

5.4.1. What is the sequence number of the TCP segment that is used to transfer your name? What is the value of the acknowledgement field in the following ACK segment?

5.4.2. What is the sequence number of the TCP segment that is used to transfer the string of only 4 characters? What is the value of the acknowledgement field in the following ACK segment?

5.4.3. What will be the acknowledgement field in the following ACK segment if you will transfer your surname?

6. Analysis of the TCP connection termination process.
   6.1. The last four captured packets (starting from FIN ACK segment) represent the TCP connection termination process. Analyse those packets and fill the following table:

Field	Packet			
	1st	2nd	3rd	4th
Source IP address				
Destination IP address				
Source port number				
Destination port number				
TCP Flags				
Relative Sequence Number				
Relative Acknowledgment Number				

7. Analyse the attempt to initiate TCP connection to the closed port.
   7.1. Start up the *Wireshark* packet capture on the Guest computer with the *eth.addr == MACaddressofGuest && tcp* display filter.
   7.2. Type in *telnet HostIPaddress RandomPortNumber* into Windows terminal. Choose the *RandomPortNumber* in the range from 20000 to 30000.
   7.3. Stop packet capture. Analyse captured packets and answer the following questions:
      7.3.1. What TCP segment was sent by Host as the response to the SYN segment?
      7.3.2. How many times Telnet client tries to initiate TCP connection?
8. Analysis of the UDP protocol.
   8.1. Open a Windows terminal in the Host computer and browse to the *Netcat* location directory *c:\tools\nc*.
   8.2. Make the *Netcat* run in listen mode on 5555 port and accept UDP packets. Type in *nc –l –u –p 5555 –v –n* into terminal

window. Option *–n* (numeric only) tells *Netcat* do not resolve DNS names for IP addresses.

8.3. Start up the *Wireshark* packet capture on the Guest computer with the *eth.addr == MACaddressofGuest && udp* display filter.

8.4. Open a Windows terminal in the Guest computer and browse to the *Netcat* location directory *c:\tools\nc*. Type in *echo YourName > file.txt* into terminal window to create a file. Then issue the *nc –u HostIPaddress 5555 < file.txt* to send created file using UDP protocol and stop packet capture.

8.5. Analyse captured packet and fill the following table and answer the questions:

<table>
<tr><td rowspan="5">IP Header</td><td>IP version</td><td>Header length</td><td>Type of service</td><td colspan="2">Total length (in bytes)</td></tr>
<tr><td colspan="3">Identification</td><td>Flags</td><td>Fragment offset</td></tr>
<tr><td colspan="2">Time to live</td><td>Protocol</td><td colspan="2">Header checksum</td></tr>
<tr><td colspan="5">Source IP address</td></tr>
<tr><td colspan="5">Destination IP address</td></tr>
<tr><td rowspan="3">UDP</td><td colspan="3">Source port number</td><td colspan="2">Destination port number</td></tr>
<tr><td colspan="3">UDP length</td><td colspan="2">UDP checksum</td></tr>
<tr><td colspan="5">Data</td></tr>
</table>

8.5.1. What field specifies the next (after IP) encapsulated protocol? What value of this field specifies UDP protocol? What value of this field specifies TCP protocol?

8.5.2. Was there any response from the Host computer to the received UDP packet? Explain why?

8.6. Stop *Netcat* on the Host by pressing *Crtl+C* buttons. Do the same on the Guest.

8.7. Send the UDP packet to the closed port. Start up the *Wireshark* packet capture on the Guest computer with the same display filter as in previous task. Then issue the *nc –u HostIPaddress 5555 < file.txt* on the Guest computer and stop packet capture.

8.8. What packet was sent in response to UDP packet? If no response would be sent, does this mean that UDP packet reach the destination successfully? Explain why?

9. The usage of the command line tools for network packet analysis and generation: *Windump* – network packet analyzer (the Windows version of *tcpdump*) and *Nemesis* – network packet crafting and injection utility.

9.1. Before you start using *Nemesis* the *WinPcap 3.0* should already be installed on the Guest computer. Browse to the *c:\tools* directory there you find *WinPcap 3.0* and install it.

9.2. Open a Windows terminal on the Guest computer and browse to the *Nemesis* location directory *c:\tools\nemesis*. Then type in *nemesis* without any options to display basic usage information. To display more options issue *nemesis [mode] help*, there mode is a particular protocol, e.g. ARP, TCP, UDP and others.

9.3. Create a simple TCP SYN segment and send it on the network. Issue *nemesis tcp -S GuestIPaddress -D HostIPaddress –fS*. If the text "*TCP Packet Injected*" will display, then it means that packet was sent successfully.

9.4. Create three different packets which will be sent to the network:

	1st packet	**2nd packet**	**3rd packet**
Protocol	**TCP**	**UDP**	**ICMP**
Source IP	*Guest IP*	*Guest IP*	*Guest IP*
Destination IP	*Host IP*	*Host IP*	*Host IP*
Source port	*Choose randomly*	*Choose randomly*	–
Destination port	*Choose randomly*	*Choose randomly*	–
Flags	*SYN and ACK*	–	–
Sequence number	*Choose randomly*	–	*Choose randomly*
Acknowledgment number	*Choose randomly*	–	–

Data to send	–	*Your name*	*Your name*
ICMP type	–	–	8
ICMP code	–	–	0

9.5. Open a Windows terminal on the Host computer and browse to the *Windump* location directory *c:\tools\windump*. Issue *windump –D* to display the list of the network interfaces available on the system. The option *–i* is used to specify an interface on which to capture (e.g. *–i 3*).

9.6. Prepare capture filters for *Windump* to capture packets sent by *Nemesis*.

9.7. Use *Nemesis* to send created packet and *Windump* to capture it once at the time. Show all options used to create packets, to capture packets and the results of capture.

Some common *Windump* options and capture filter primitives are listed in following table:

Options						
–D	List available interfaces					
–i <iface>	Specifies the capture interface					
–n	Don't convert addresses to names					
–w <file>	Write captured packets to file					
–X	Print frame payload in hex and ASCII					
-c <count>	Exit after capturing count packets					
Filter Primitives		**Protocols**			**Combine Operators**	
[src|dst] host		tcp	udp	ip	! (not)	|| (or)
[tcp|udp] [src|dst] port		icmp	arp	ether	&& (and)	
TCP Flags			**ICMP Types**			
tcp-syn	tcp-ack	tcp-rst	icmp-echo		icmp-echoreply	
tcp-fin	tcp-psh	tcp-urg	icmp-unreach		icmp-redirect	
Examples						
windump –i 2 –X –w capturefile tcp and dst port 23						
windump –i 3 –X –n tcp and src host 192.168.100.1						
windump –i 1 –X –c 100 "tcp[tcpflags] & (tcp-rst|tcp-fin) !=0"						
windump –i 1 –X "icmp[icmptype] == icmp-redirect"						

Content of report

1. Work objective.
2. The use of the *Wireshark* network protocol analyzer (Tasks 2.3–2.9 and 3).
3. The analysis results of the TCP protocol (Tasks 4.5, 4.8–4.9, 5.4, 6.1 and 7.3).
4. The analysis results of the UDP protocol (Tasks 8.5 and 8.8).
5. The use of the *Windump* and *Nemesis* (Tasks 9.4 and 9.7).
6. General conclusions of the work.

Review questions and problems

1. What is a network analysis?
2. What is a network analyzer? Who usually uses a network protocol analyzer and for what purposes?
3. What are IP, TCP and UDP?
4. Explain the TCP three-way handshake mechanism to establish a connection.
5. Explain the TCP connection termination process.
6. Write the following *Wireshark* display filter expressions to display: a) any traffic except with a source or destination IP address of 192.168.10.25; b) only packets from a specific network interface card manufacturer, e.g. Intel; c) only unicast traffic; d) only the packets less than 256 bytes in length; e) any traffic except from/to *www.vgtu.lt*.
7. Write the following *Windump* capture filter expressions to capture traffic: a) from a range of IP addresses ports; b) TCP communication packets between two particular hosts; c) only UDP traffic and ICMP port unreachable messages; d) only TCP packets with RST and FIN flags set.

Literature

Casad, J. 2008. *Sams Teach Yourself TCP/IP in 24 Hours*. 4th edition. Sams. 456 p. ISBN-13: 978-0672329968.

Kurose, J. F.; Ross, K. W. 2009. *Computer Networking: A Top-Down Approach*. Addison Wesley, 5th edition. 864 p. ISBN-13: 978-0136079675.

Orebaugh, A., Ramirez, G., Beale, J., Wright, J. 2007. *Wireshark & Ethereal Network Protocol Analyzer Toolkit*. Syngress. 448 p. ISBN-13: 978-1597490733.

Sloan, J. D. 2001. *Network Troubleshooting Tools*. O'Reilly Media. 364 p. ISBN-13: 978-0596001865.

Laboratory work 6

Analysis of the Data Link Layer Protocols (Ethernet, ARP)

Objectives

Investigate the Ethernet protocol and the functions of *Address Resolution Protocol* (ARP). Analyse the ARP packet format and explore the *arp* command features.

Basic knowledge and theory

Each network adapter, in the Ethernet networks, is assigned a unique 48-bit (6 byte) hardware address also known as MAC (*Media Access Control*). This address consists of two parts, each containing 3 bytes. The first three bytes refer to the manufacturer, and the remaining 3 bytes are assigned by the manufacturer.

Two computers connected to a single physical network can interact with each other only when both are aware of each other's hardware address. Consequently the network hub or router before sending a package must link the device, to which packet is being sent, IP address to its hardware address. For this purpose the Address Resolution Protocol (ARP) is used.

ARP protocol operation scheme is shown in Figure 6.1. When A computer wants to know the hardware address of computer B, it sends the ARP request with an IP address of the computer, which hardware address it wants to know. All computers who are in the same psychical network receive the request, including computer B. However, only computer B sends back the response. When computer A receives the response, it uses the acquired hardware address to send direct packets to computer B.

It may seem strange that if computer A wants to send a packet to computer B, first it must send a query for all computers on a network

including computer B. Why the computer A can not send a broadcast packet to computer B immediately? It is important to understand, that broadcasting takes a lot of network resources, that's why it can't be used for simple exchange of packets between two computers. In this mode all computers on the network receive the sent packet and each of them (except one) have to process it uselessly.

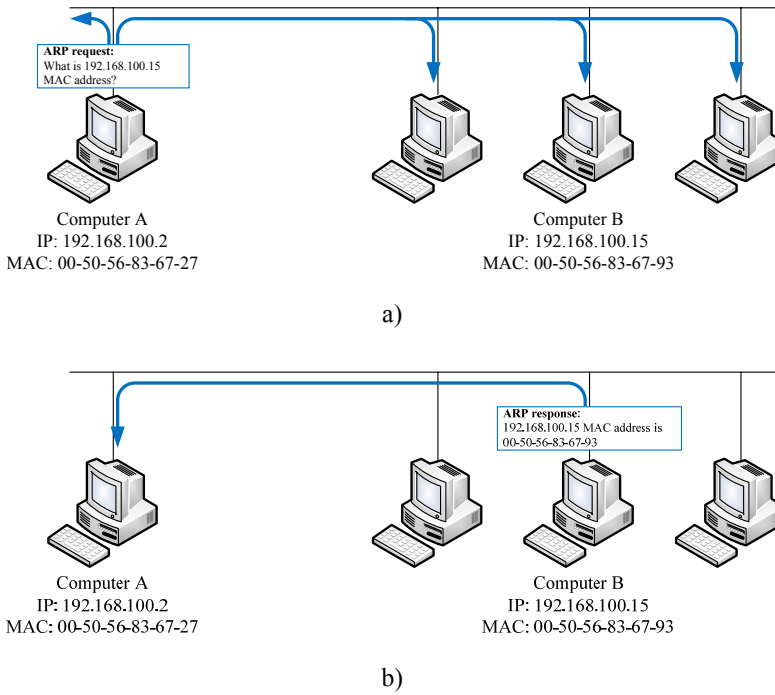


Fig. 6.1. ARP protocol operation scheme

Applications, using the ARP protocol, must retain the last few replies to ARP requests in the ARP cache in order to reduce network load. ARP cache is a table which contains information about IP addresses and their corresponding hardware (MAC) addresses. In other words, once the program receives a response to the forwarded

ARP request, it saves the IP address and a corresponding hardware address for later use. Before sending a new ARP request the program always checks the cache first. If the needed addresses are found, then the ARP request is not sent to the network.

It is possible that some information about the IP and MAC addresses stored in the computer's cache memory gets old, and the client does not know anything about it. Suppose computer A sends an ARP request to the computer B, to which it responds and then crashes. Meanwhile computer A does not know about the computer B current state, because it did not receive a message informing about the changes in computer B status. Furthermore, since computer A already has a cache entry linking computer B IP and MAC addresses, computer A will continually try to send packets directly to computer B. Ethernet hardware does not provide means to monitor the status of computer B, because Ethernet technology does not include a mechanism guaranteeing packet delivery to the recipient. Therefore, computer A does not have any means to find out when the information stored in the cache memory about the ARP requests becomes obsolete. Computer A needs to take additional steps to update records stored in the cache. In practice, this is done quite simply. When an application receives a response to the ARP request it activates a timer, which sets a certain time interval, during which the stored ARP entries linking the IP and MAC addresses is considered correct. After the time interval, regardless of whether the entry is correct or not, it is deleted from the cache. The only drawback of this mechanism is delay. For example, if the time interval after which information about the ARP requests is refreshed is equal to X seconds, the sender will know that the recipient is not available only after X seconds.

ARP message, before sending, must be encapsulated inside the Ethernet frame. As shown in Figure 6.2, the ARP packet takes up the frame data field.

Frames, in which the ARP packet is placed, are identified by specifying an appropriate value into the frame header field *Type*. In

Ethernet networks these types of frames are identified by writing into the frame type field a value 0x0806.

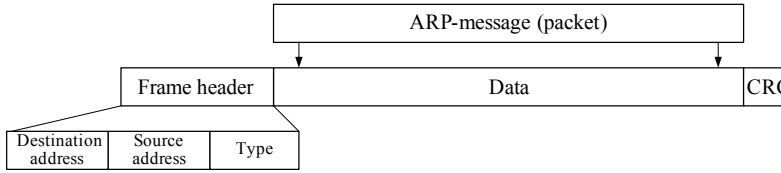


Fig. 6.2. ARP message encapsulation into the Ethernet frame

ARP packet format that is used in Ethernet networks is shown in Figure 6.3.

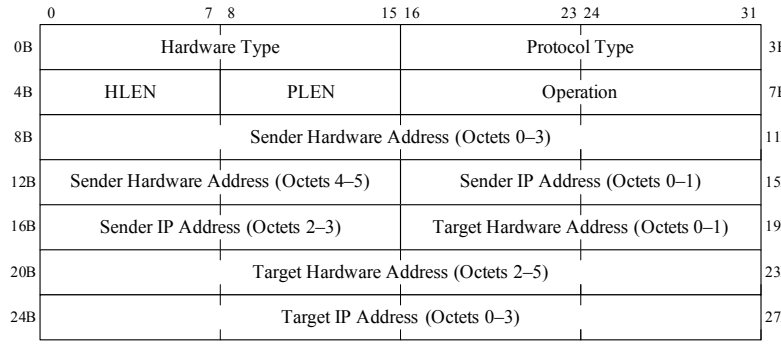


Fig. 6.3. ARP message format

As you may recall, an Ethernet network hardware address length is 48 bits or 6 octets (1 octet = 1 byte), and the logical address (IP address) length – 4 octets. *Hardware type* field indicates the type of hardware used for the local network transmitting the ARP message. This value is equal to 1 in Ethernet networks. The *Protocol Type* field indicates a higher layer protocol type. When using IP protocol, this value is set to 0x0800. The *Operation* field specifies which operation is running: 1 for ARP request and 2 for ARP response (In case of RARP protocol: 3 for RARP request and 4 for RARP response). The *Hardware Address Length* (HLEN) field specifies the length (in octets) of a link layer address and the *Protocol address length* (PLEN)

field specifies the length (in octets) of a network layer address in the message. For Ethernet HLEN=6 and PLEN=4.

When a sender composes an ARP query, in the relevant fields of the package it indicates its hardware address, IP address and IP address of the computer whose hardware address whishes to know. When the computer receives the query, it fills the missing fields in the ARP packet, swaps the source and destination addresses, specifies the *operation type* field value and sends a response.

Work assignment and methodical guidelines

1. Read attentively the theory section of laboratory work. Before starting the work discuss all obscure questions with the lecturer.
2. The use of *arp* command. The *arp* command (the same in both MSDOS and Linux/Unix) is used to view and modify the ARP cache table entries (the Ethernet MAC to IP address link) on the local computer.
   2.1. Open a Windows terminal. There are two ways to do it: 1) click **Start → All Programs → Accessories → Command Prompt**; 2) click **Start → Run**, type *cmd* and click *OK*.
   2.2. The *arp* command with no options will display help information. Type the *arp* command and read the output.
   2.3. Use *arp –a* command to view ARP table and answer the following questions:
      2.3.1. What entries if any are in the ARP table?
      2.3.2. Why are there entries or no entries?
   2.4. Use *arp –d* * command to clear all entries in the ARP cache. The wildcard * is used to delete all entries. Note that addresses can be deleted individually by specifying the IP address. Verify that ARP table is empty.
3. Use the *ping* command to dynamically add entries in the ARP table about the neighbour computer.
   3.1. To find out the IP address of the neighbour computer first of all use *ipconfig* command to find the IP address of your

computer and then increment it by 1 (e.g. if your IP address is 192.168.10.100, then the neighbour computer IP address would be 192.168.10.101).

3.2. Issue the *ping IPofyourneighbour* command and view the ARP table entries. Fill the following table.

Internet Address (IP)	Physical Address (MAC)	Type

3.3. Issue *arp –d* * command to clear all entries in the ARP cache and ping the neighbour computer again. Analyse ICMP echo request packets reply time. Why might the first ping take longer than the rest?

4. Find the physical address of the default gateway.

4.1. Issue the *ping* command to the gateway. To find out the IP address of the gateway use *ipconfig /all* command.

4.2. View the ARP table entries and fill the table below.

Gateway IP Address	Gateway Physical Address	Type

5. Capture and analyse the Ethernet frame and ARP packet format.

5.1. Start the *Wireshark* network protocol analyzer. Then click **Capture → Interfaces…**, choose the interface that corresponds to the LAN and click *Start*.

5.2. Use *arp –d* * to clear all entries in the ARP cache. It will require ARP to remap IP addresses to physical addresses.

5.3. Send one ping request to the neighbour computer, using the command *ping IPofyourneighbour –n 1*. (option *–n* is used to specify the number of echo request to sent).

5.4. Stop *Wireshark* packet capture by pressing *Stop live capture*  button. All captured packets are displayed in the *Wireshark* top (*Packet List*) pane. The middle and bottom

panes display packet details and contents. To filter out various broadcast packets from other computers and to leave only packets with your computer IP address apply the display filter *eth.addr == YourMACaddress*.

5.5. Examine the *Packet List* pane. There should be two ARP packets. Fill in the following table with information about the first ARP packet:

ARP packet type:	?	
Frame Header		
Destination MAC address	Source MAC address	Type
ARP Packet		
Hardware Type		Protocol Type
Hardware address length (HLEN)	Protocol address length (PLEN)	Operation Code
Sender MAC Address		
Sender IP Address		
Target MAC Address		
Target IP Address		

Fill in the following table with information about the second ARP packet:

ARP packet type:	?	
Frame Header		
Destination MAC address	Source MAC address	Type
ARP Packet		
Hardware Type		Protocol Type
Hardware address length (HLEN)	Protocol address length (PLEN)	Operation Code
Sender MAC Address		
Sender IP Address		
Target MAC Address		
Target IP Address		

5.6. Compare these two ARP packets. List and explain the differences.

5.7. Fill in the following table with information about the frame header of first ICMP echo request packet:

Frame header of ICMP echo request packet

Destination MAC address	Source MAC address	Type

5.8. Compare the frame header *Type* field of ICMP packet with the frame header *Type* field of ARP packet. Explain the differences. What does the *Type* field indicate?

6. Examine ARP exchanges when the ping request is sent to a host which location is outside the LAN.
   6.1. Start the *Wireshark* packet capture mode.
   6.2. Delete all entries in ARP table and issue the *ping* command to a host outside of your local network where the ping must be forwarded by the default gateway.
   6.3. Stop *Wireshark* packet capture and analyse captured ARP packets. Explain why the ARP request was for the default gateway and not the IP address of the ping.
   6.4. Fill in the following table with information about the frame header of first ICMP echo request packet:

Frame header of ICMP echo request packet

Destination MAC address	Source MAC address	Type

   6.5. Does the destination MAC address belongs to the IP address of the ping? Explain why.
7. Manually add entries to the ARP cache. The *arp –s IP_address MAC_address* command allows to manually add static entry to the ARP cache that resolves IP address to the physical address (*MAC_address*).
   7.1. Delete all entries in ARP cache.
   7.2. Manually add two static ARP entries: 1) for the neighbour computer with correct IP address, and correct physical address; 2) for the default gateway with correct IP address, but the wrong physical address.

7.3. Issue the *ping* command for the neighbour computer, then for default gateway and finally for the host outside of your local network. Analyse and explain the results.

8. Examine if the host that issues the ARP reply, stores the ARP request sender's IP address and MAC address in its ARP cache.
   8.1. You need two computers for this task, therefore choose the partner to work in group of two students.
   8.2. Delete all entries in ARP tables on both computers.
   8.3. Issue the *ping* command only from one computer and then view the ARP caches on both computers. Explain the results.
   8.4. How can you determine if the host that issues the ARP reply, stores the ARP request sender's IP address and MAC address in its ARP cache using the network protocol analyzer?

Content of report

1. Objectives of the work.
2. The results of the *arp* command use (Tasks 2.3, 3.2–3.3 and 4.2).
3. The analysis results of the Ethernet frame and ARP packet format (Tasks 5.5–5.8).
4. The analysis results of ARP communication (Tasks 6.3–6.5, 7.3, and 8.3–8.4).
5. General conclusions of the work.

Review questions and problems

1. What is the purpose of Address Resolution Protocol (ARP)?
2. What are the advantages and disadvantages of ARP?
3. What is the length of physical address? What specifies the first and the second three bytes of physical address?
4. What command is used: a) to display all entries in ARP cache, b) to delete entry for 192.168.10.15, c) to delete all ARP cache

entries, d) to create a static ARP entry for 10.10.10.1? Show the commands and the ARP table outputs.

5. What is difference between dynamic and static ARP entry? What is the advantage of a static ARP entry?
6. Explain the reason why the ARP request is a broadcast and the ARP reply is a unicast.
7. When sending a request to a host outside of your local network what IP address is used? What physical address is used? Why?

Literature

Comer, D. E. 2005. *Internetworking with TCP/IP Vol.1: Principles, Protocols, and Architecture*. 5th Edition. Prentice Hall. 688 p. ISBN-13: 978-0131876712.

Kurose, J. F.; Ross, K. W. 2009. *Computer Networking: A Top-Down Approach*. 5th edition. Addison Wesley. 864 p. ISBN-13: 978-0136079675.

Laboratory work 7

Analysis of the Web Protocols (DNS, HTTP)

Objectives

Investigate the name resolution process to translate host addresses into IP addresses. Analyse the *Hypertext Transfer Protocol* (HTTP) designed for communications between clients and servers.

Basic knowledge and theory

When you type *www.vgtu.lt* into your browser to see the web page of Vilnius Gediminas Technical University (VGTU), before your browser will display the requested page it has to establish a TCP connection to the web server where the page is located. To do that, it first has to obtain the IP address of the web server. This is done using *Domain Name System* (DNS) resolution mechanism, which translates domain names to IP addresses. DNS implements a distributed database that contains the names and addresses of all reachable hosts on a TCP/IP network. DNS additionally provides other services such as caching request, host and mail server aliasing, load distribution.

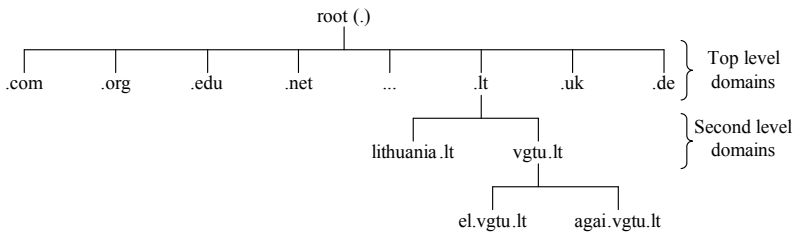


Fig. 7.1. Hierarchical structure of the DNS

The DNS database resides on a large number of servers organized in a hierarchical (tree) structure (Fig. 7.1), where each server has a small portion of mappings. At the top of the DNS tree is a *root*

node followed by the *Top Level Domains* (TLDs). The root node is represented as a dot (.). In the Internet there are 13 root DNS servers with the names based on letter (from *a.root-servers.net* to *m.root-servers.net*) and distributed around the globe [www.root-servers.org]. Top Level Domains are divided into two types: 1) *Generic Top Level Domains* (gTLDs) (e.g. .com, .org, .edu etc.) and 2) *Country Code Top Level Domains* (ccTLDs) (e.g. .lt, .uk, .de etc.).

When an application (e.g. web browser) wants to connect to a host by its hostname, it invokes the resolver residing on a local computer and passes to it the name, for example *www.vgtu.lt*. Then the resolver sends a request to the local DNS server defined in the network card TCP/IP configuration. If the local DNS server doesn't know the IP address for the requested name, it sends a request to one of the root servers, which returns the names and addresses of the top level domain servers for the ".lt" domain. Then the local DNS server queries one of these TLD servers, asking what name server is responsible for *vgtu.lt* domain. The TLD server checks its database, finds an entry for *vgtu.lt* domain and returns the IP address of authoritative server for vgtu.lt. Finally, local DNS server queries the authoritative server for *vgtu.lt*, which returns the IP address for the host *www.vgtu.lt*. The local DNS name server returns this IP address to the resolver, which returns the result to the application.

As described above, DNS is implemented as distributed database which stores information in the form of *Resource Records* (RRs). The resource record structure (in the form in which RRs are propagated through the network) is shown below:

Name	*Type*	*Class*	*TTL*	*RDLength*	*RData*

where: *Name* – the meaning of this field varies according to the *Type* of the resource record; *Type* – record type; *Class* – record class, the normal value is IN (0x0001) for internet protocols; TTL – time to live specifies the time interval that the record may be cashed and

kept valid; *RDLength* – specifies the length of the *RData* field; *RData* – the format of this field varies according to the *Type* and *Class* of the resource record.

Some common types of resource records in IPv4 networks are:

- *Type*=A (*Address*) – record maps a DNS name to an IP address. *Name* is a hostname and *RData* is the IP address for this hostname;
- *Type*=NS (*Name server*) – specifies the name of a DNS name server that is authoritative for the domain. *Name* is a domain and *RData* is the hostname of an authoritative DNS server;
- *Type*=CNAME (*Canonical name*) – enables an alias to be linked to the canonical (real) name of the node. This is used to associate a number of hostnames with one IP address. *Name* is an alias hostname and *RData* is a canonical name;
- *Type*=MX (*Mail eXchange*) – specifies the mail server address for the domain name. *Name* is the alias hostname of a mail server and *RData* is the canonical name of a mail server.

All DNS query and response messages are sent within UDP datagrams to port 53. Both query and response messages have the same format (Fig. 7.2).

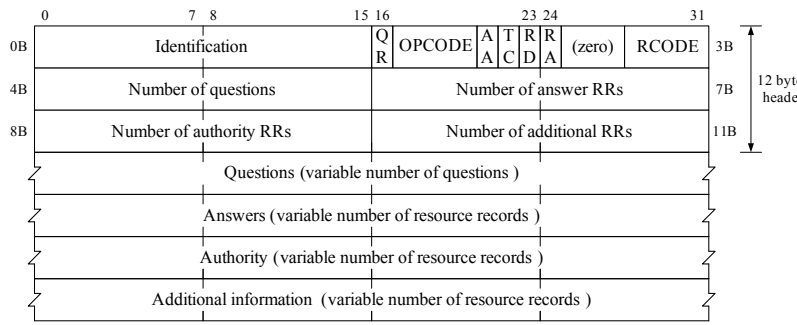


Fig. 7.2. DNS message format

The DNS message format has a fixed 12-byte header followed by four variable-length fields. The *Identification* field is set by a

client and copied into the answer by a server, allowing the client to match a query with response. The next two bytes of a header contain a number of flags. A 1 bit query-response (QR) flag specifies whether this message is a query (0), or a response (1). The OPCODE field identifies the request operation type: 0 – standard query, 1 – inverse query, 2 – status query. The authoritative answer (AA) flag is set in a response message indicating that the name server is authoritative for the domain in the question section. The truncated (TC) flag, if set, indicates that only the first 512 bytes of the response was returned. The recursion desired (RD) flag may be set in a query and is copied into the response (if recursion supported by the server). This flag tells the name server to perform recursion itself. The recursion available (RA) flag is set to 1 in the response if the server supports recursion. A 3 bit field (zero) is reserved for future use and must be set to 0 in all queries and responses. A 4 bit response code (RCODE) field is set as part of responses. The common values are 0 (no error) and 3 (name error). The next four 16-bit fields specify the number of entries in the four sections following the header.

Every DNS query message usually contains only one question. Nonetheless whether the message is a query or a response, it always includes one question in the question section. Each question has a name, type and class associated with it. The answer section contains the resource records corresponding directly to the asked question. The authority section contains the resource records of other authoritative servers. The additional information section usually contains IP addresses of authoritative name servers.

Hypertext Transfer Protocol

The Hypertext Transfer Protocol (HTTP) is the application layer protocol that applications use to communicate over the World Wide Web. It is defined in RFC 1945 (HTTP/1.0) and RFC 2616 (HTTP/1.1). The basic elements of the World Wide Web were cre-

ated by Tim Berners-Lee in 1990 at the CERN research institute in Geneva, Switzerland.

The HTTP protocol is based on a request-response model. The client (web browser) sends a request message and the web server replies with a response message. HTTP defines the structure of these request and response messages. The communication generally takes place over a TCP/IP connection on the Internet. The client establishes a TCP connection with server, port 80. This is the default port number of HTTP. Then the client (browser) requests a web page (document) from the server. The server answers with the requested document within the same TCP connection. Finally, the browser displays the response to the user.

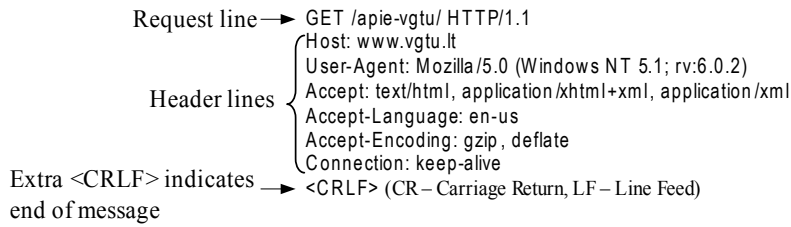
A web page usually consists of several objects: a base HTML file, JPEG images, mp3 files, video clips and others. If, for example, a web page contains HTML text and two images, the web page has three objects: the base HTML file plus two images. Every object must be downloaded by a separate HTTP request from the web server. In the older versions of the HTTP protocol, a new TCP connection was always established for each request, resulting in many consecutive short-lived TCP connections. Only the basic text of the web page is downloaded by the first request. HTTP protocol version 1.1, by default, assumes that only one TCP connection will be established between the client and server for the entire web page. HTTP/1.1 specifies that connections should remain open until explicitly closed, by either party.

The base HTML file usually contains many references to other objects, with object's Uniform Resource Locator (URL). Each URL has two components: the hostname and path. The hostname is the fully qualified domain name of the server on which the path is accessible. For example, the URL *http://www.vgtu.lt/images/favicon.png* consists of hostname *www.vgtu.lt* and path */images/favicon.png*. An URL is defined in RFC 1738.

HTTP is a stateless protocol. This means that HTTP protocol is unable to retain information or status about each client that connects to a web site and therefore treats each request as a unique and independent connection. Usually it's not a problem, however for applications such as shopping carts which accumulate information about items added to the cart it is. The common solutions are the use of server side sessions and HTTP cookies.

As mentioned before, all communications using the Hypertext Transfer Protocol takes place via HTTP messages, of which there are only two types: requests and responses. HTTP messages are text-based.

An example of an HTTP request message is shown below:



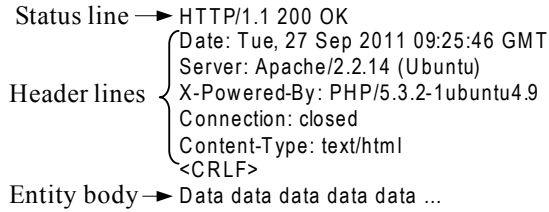
A request line has three fields, separated by spaces: the method field, the URL field of the requested object, and the version of protocol used. The method field indicates the method to be performed on the object identified by the URL field. HTTP version 1.1 supports the following methods: GET, POST, HEAD, OPTIONS, TRACE, CONNECT, PUT, and DELETE. The most common methods used are GET, POST and HEAD. GET method is used to retrieve objects identified by the URL. The POST method is commonly used to submit HTML form data. A HEAD method is identical to the GET method, except it tells the server to return the response headers only, without the requested object. It is useful for application debugging.

A header is a series of lines, with each line containing a name followed by a colon and a space, and then a value. The order of

header lines is not significant. Several options in the value field are separated using a comma. The header lines allow the client to pass additional information about the request, and about the client itself, to the server. The *Host:* header line indicates the host on which the requested object is located. The second *User-Agent:* header line identifies the client as Mozilla (version 5.0) running on Windows OS. The *Accept:* header line indicates what types of content the browser will accept. The *Accept-Language:* header line indicates that the client prefers American English content. The *Accept-Encoding:* header line indicates that the browser can handle *gzip* or *deflate* compressed content. The last header line *Connection:* indicates that the client is requesting the use of persistent TCP connections.

All HTTP header lines, including request line, should end with CRLF, where CR and LF stand for Carriage Return and Line Feed, respectively. A line with nothing preceding the CRLF indicates the end of the header fields, and possibly message-body.

After receiving and interpreting a request message, a server responds with an HTTP response message. The response message to the example request message above could be as follow:



The response message begins with a status line and series of header lines giving information about document and the server itself. The status line consists of the protocol version followed by a numeric status code and its associated textual phrase. In this case, the status line indicates that the server is using HTTP/1.1 and that the request has been accomplished correctly (status code: 200, message: OK).

Status codes consist of three digits. The first digit specifies the general class of the status code: the 1xx are informational (e.g. 100 Continue); the 2xx indicate success (e.g. 202 Accepted); the 3xx specify redirection (e.g. 301 Moved Permanently); the 4xx show a client error (e.g. 404 Not Found); and the 5xx specify a server error (e.g. 500 Internal Server Error).

Each header line is composed of a name describing the header type, followed by a colon and the value of the header. The *Date:* header line indicates the date and time that the message was sent by the server. The header line *Server:* indicates the web server application. The *X-Powered-By:* header line is non standard header field as it is marked with prefix *X-*. This line specifies the technology supporting by the web server application. The *Connection:* header line indicates that the server will close the connection after sending the response. The *Content-Type:* header line tells the browser what type of resource the server is supplying in the entity-body. A blank line with CRLF separates the header from the message body.

Work assignment and methodical guidelines

1. Read attentively the theory section of laboratory work. Before starting the work discuss all obscure questions with the lecturer.
2. Analyse the DNS protocol.
   2.1. Start up the *Wireshark* network protocol analyzer. Type in *ip.addr == YourIPaddress && dns || http* into display filter toolbar to filter for DNS and HTTP packets initiated to and from your own IP address. Then start packet capture.
   2.2. Enter the following URL *http://itsauga.lt/networks/nlab7a.html* into your favourite browser. After the page has displayed, stop *Wireshark* packet capture. The captured DNS and HTTP packets will be displayed in the *Wireshark Packet List* pane. You should notice that DNS packets are sent before HTTP. This is because the browser does not know the IP address for *itsauga.lt*. To resolve this, DNS protocol is used.

2.3. Analyse captured DNS packets and answer following questions:
   2.3.1. What is the IP address of DNS server?
   2.3.2. What is the Transport layer protocol used for DNS packets? What is the destination port number of DNS standard query?
   2.3.3. What is DNS question *Name*, *Type* and *Class*?
   2.3.4. What is the IP address of *itsauga.lt*?
   2.3.5. What is the time to live value of *itsauga.lt*?
   2.3.6. What additional information was sent in DNS response?
2.4. Restart the *Wireshark* packet capture with the *ip.addr == YourIPaddress && dns* display filter.
2.5. Open a Windows terminal and type in *nslookup* command.
2.6. Find an IP address of a randomly picked website (note that google, yahoo, microsoft and other usual websites are not random!). Type in *randomly_picked_website* into already activated *Nslookup* tool. After response was displayed, stop packet capture.
2.7. You should see four captured DNS packets. First DNS query was initiated by *Nslookup* tool to resolve the domain name of your DNS server. This is a reverse DNS lookup. A reverse DNS lookup is the inverse process to determine the domain name associated with an IP address. As we know, usually the DNS is used to determine the IP address associated with a domain name. Reverse DNS lookups for IP addresses use a reverse *in-addr* entry in the special domain *in-addr.arpa*. In this domain, an IP address is represented in reverse order to the usual textual writing form of the IP addresses, to which is appended the second level domain suffix *.in-addr.arpa*. The process of reverse resolving an IP address uses the PTR record type.
2.8. Analyse captured DNS packets and answer following questions:
   2.8.1. What is the domain name of your DNS server?

2.8.2. What is the IP address and domain name of one of the authoritative name servers?

2.8.3. What website did you randomly pick? What is its IP address?

2.8.4. How many authoritative name servers have records about picked website? List one.

2.9. Restart the *Wireshark* packet capture with the same display filter as in previous task.

2.10. Try to find an IP address of none existing website. Use the same *Nslookup* tool for this task.

2.11. Stop the *Wireshark* and analyse captured packets. Answer the following questions:

2.11.1. How many DNS queries were sent?

2.11.2. What responses were received?

3. Analyse the simple HTTP client-server interaction.

3.1. Start up the *Wireshark*. Type in *ip.addr == YourIPaddress && http* into display filter toolbar and start packet capture.

3.2. Clear your browser's cache and enter the following URL *http://itsauga.lt/networks/nlab7a.html*. After the page has displayed, click the *Refresh* button on your browser (or press F5 button on your keyboard) and stop *Wireshark* packet capture. You should see two HTTP GET requests and two responses. If you see more then two request and two responses, this may be the result of requests for *favicon.ico* (a website's small icon which is displayed in the address bar) or other web client which may be running on your computer. Ignore them.

3.3. Highlight the first captured HTTP GET request message and examine its content. Fill the following table.

Request HTTP Protocol Version:	
Request Method:	
Request URI:	
User Agent:	
Accept Language:	

3.4. Examine the content of first HTTP response message and answer the following questions:
   3.4.1. What is the status code returned from the server to the browser? What does that status code mean?
   3.4.2. When was the HTML file received by browser last modified at the server?
   3.4.3. How many bytes of content are sent in the response message?
   3.4.4. What is the value of page entity tag (ETAG)? What is it used for?
3.5. Examine the content of second HTTP GET request when the *Refresh* button was pressed. Compare this GET request with the first one. What are differences?
3.6. Examine the content of response message to the second GET request and answer the following questions:
   3.6.1. What is the status code and phrase returned from the server? What does that status code mean?
   3.6.2. Was the requested page returned from the server? Explain why?

4. Analyse the HTTP protocol used to retrieve large document.
   4.1. Click **Edit → Preferences** on the *Wireshark* menu. Expand protocols list and select HTTP protocol. Then disable *Reassemble HTTP headers spanning multiple TCP segments*, *Reassemble HTTP bodies spanning multiple TCP segments* and *Reassemble chunked transfer-coded bodies*. This tells *Wireshark* to display separate packets used to transfer large file instead of one HTTP response which represents reassembled entire requested document. Finally click *Apply* and *Ok*.
   4.2. Start up the *Wireshark* packet capture with the *ip.addr == YourIPaddress && http* display filter.
   4.3. Clear your browser's cache and enter the following URL *http://itsauga.lt/networks/rfc792.txt*. After the page has displayed stop *Wireshark* packet capture.

4.4. Analyse captured packets and answer following questions:
    4.4.1. How many HTTP GET request messages were sent to retrieve requested document?
    4.4.2. How many data-containing packets were needed to transfer the single HTTP response?
    4.4.3. What is the maximum data length sent by one packet?
    4.4.4. Sum the data in bytes sent by each packet and compare this value with the received file size. (Note: to find the file size, save it on a local disk and view its properties.) Are they equal? Explain why?

5. Analyse the HTTP protocol used to retrieve HTML document with embedded objects.
    5.1. Click **Edit → Preferences** on the *Wireshark* menu. Expand protocols list and select HTTP protocol. Then enable *Reassemble HTTP headers spanning multiple TCP segments*, *Reassemble HTTP bodies spanning multiple TCP segments* and *Reassemble chunked transfer-coded bodies.* Click *Apply* and *Ok*.
    5.2. Start up the *Wireshark* with the *ip.addr == YourIPaddress && http* display filter.
    5.3. Clear your browser's cache and enter the following URL *http://itsauga.lt/networks/nlab7b.html*. After the page has displayed stop *Wireshark* packet capture.
    5.4. Analyse captured packets and answer following questions:
        5.4.1. How many HTTP GET request messages were sent to retrieve embedded objects? What are IP addresses of the web servers to which these GET requests were sent?
        5.4.2. The downloaded HTML page has two duplicated objects. Does your browser sent GET request to these objects?
        5.4.3. Did your browser download the objects serially, or in parallel? Explain. (Hint: check TCP ports.)

6. Analyse how data is transferred using basic HTML form.
   6.1. Start up the *Wireshark* with the same display filter as in previous task.
   6.2. Clear your browser's cache and enter the following URL *http://itsauga.lt/networks/nlab7c1.php*.
   6.3. Fill the submission form. Use your name for *Username* and surname for *Password*. Then press *Submit* and stop *Wireshark* packet capture.
   6.4. Analyse captured packets and answer following questions:
      6.4.1. What request was sent by browser then you pressed *Submit* button? How is it different from the first request sent by browser?
      6.4.2. Does the data you typed into submission form can be readily found by analysing captured packets? Is it secure to use this method for access to valuable resource on the Internet?
      6.4.3. What would be differences in URL if GET method would be used instead of POST? To answer this question, repeat 6.1–6.3 steps using this URL *http://itsauga.lt/networks/nlab7c2.php*.
      6.4.4. Which method, GET or POST, is more secure (only in case of peeking over the shoulder) to use? Explain.
7. Analyse the basic authentication method.
   7.1. Start up the *Wireshark* with the same display filter as in previous task.
   7.2. Clear your browser's cache and enter the following URL *http://itsauga.lt/networks/authentication/nlab7d.html*.
   7.3. Enter a username and password into authentication required window. Use the username *ShortPassword* and the password *BigProblems*.
   7.4. After the page has displayed, stop *Wireshark* packet capture. Examine captured packets and answer the following questions:

7.4.1. What is the status code of the first response message? Does the content of this message exactly correspond to the authentication required window displayed by your browser? Explain.

7.4.2. What is the *Authorization* header value of the second request message sent by your browser? What does this value mean?

7.4.3. Use any online Base64 decoder to decode this value. What do you find? Is the basic authentication method sufficient to protect username and password?

Content of report

1. Objectives of the work.
2. The analysis results of the DNS protocol (Tasks 2.3, 2.8 and 2.11).
3. The analysis results of the simple HTTP client-server interaction (Tasks 3.3–3.6).
4. The analysis results of the HTTP protocol used to retrieve large document (Task 4.4).
5. The analysis results of the HTTP protocol used to retrieve HTML document with embedded objects (Task 5.4).
6. The analysis results of the data transfer using basic HTML form (Task 6.4).
7. The analysis results of the basic authentication method (Task 7.4).
8. General conclusions of the work.

Review questions and problems

1. What is DNS? Explain the hierarchical structure of the DNS.
2. What is DNS resource records (RRs)? Explain the structure of RRs and give an example.

3. What is HTTP? What lower layer protocol is commonly used to provide the connection management and reliable delivery of the HTTP messages?
4. What HTTP request methods are most commonly used? Describe them.
5. Explain the general format of an HTTP request and HTTP response messages. Give examples.
6. Is the HTTP protocol secure enough to protect username and password? Explain.

Literature

Dostálek, L., Kabelová, A. 2006. *Understanding TCP/IP: A clear and comprehensive guide to TCP/IP protocols*. Packt Publishing. 480 p. ISBN-13: 978-1904811718.

Kurose, J. F.; Ross, K. W. 2009. *Computer Networking: A Top-Down Approach*. 5th edition. Addison Wesley. 864 p. ISBN-13: 978-0136079675.

Laboratory work 8

Analysis of the Email Protocols (SMTP, POP3)

Objectives

Analyse the main email protocols: *Simple Mail Transfer Protocol* (SMTP) used to transfer messages from one host to another and *Post Office Protocol version 3* (POP3) used by local email clients to retrieve email from a remote server.

Basic knowledge and theory

Electronic mail, commonly called email or e-mail, is one of the oldest and also one of the most widely used network applications. Email is delivered using the client-server architecture. The process of email creation and delivery is shown in Figure 8.1.

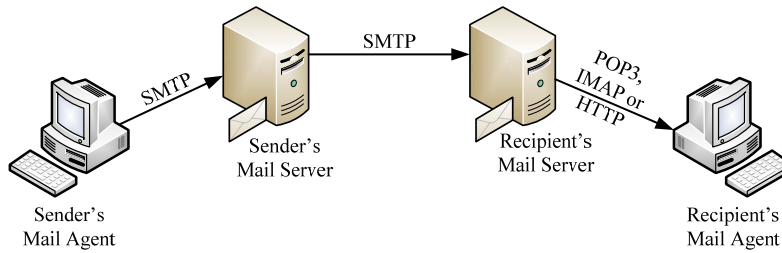


Fig. 8.1. The process of email creation and delivery

The email sender uses a mail client program (*Mail User Agent*) to create an email message. This program then sends the message to the sender's mail server (*Mail Transfer Agent*), which then forwards the message to the recipient's mail server. Finally the recipient uses its own mail client program to receive a message from the mail server. To enable this process, a variety of standard protocols are used. Email message delivery from a client application to the server, and from it directly or via intermediate servers to the destination server,

is handled by the *Simple Mail Transfer Protocol* (SMTP). The SMTP protocol can only be used to send emails, not to receive them. To retrieve email from mail servers the three main protocols are used *Post Office Protocol – Version 3* (POP3), *Internet Mail Access Protocol – version 4* (IMAP4) and *HyperText Transfer Protocol* (HTTP).

The Simple Mail Transfer Protocol defined in RFC 5321 is a client-server protocol. SMTP uses TCP as a transport protocol to transmit messages from one host to another. Mail can be transmitted by a client to his mail server, or from sender's mail server to recipient's mail server. SMTP client initiates a TCP connection to SMTP server port 25. Once the connection is established the SMTP client initiates a mail transaction. Only five SMTP commands are enough to send the mail: HELO, MAIL, RCPT, DATA, and QUIT. The server responds to each command with a reply which consists of three digit numeric code and an optional comment. The replies are used to indicate that the command was accepted, that additional commands are needed, or that an error condition occurs. This guarantees that the SMTP client always knows the state of the SMTP server. A simple example of such transaction is shown below:

S: 220 mailserver.lt ESMTP
C: HELO e-mail.lt
S: 250 Hello.
C: MAIL FROM: sender@e-mail.lt
S: 250 OK
C: RCPT TO: recipient@mailserver.lt
S: 250 OK
C: DATA
S: 354 OK, send.
C: E-mail server test
C: .
S: 250 Queued (55.093 seconds)
C: QUIT
S: 221 goodbye

The first line indicates the status code (220 *Service ready*) sent by server (S) after TCP connection has been established. It means that the mail server is up and running. The client (C) introduces itself to the server by sending the HELO command. The domain name of the client is included as an argument. Then, the client begins a mail transaction with a MAIL command. The argument of this command must be FROM: followed by the email address of the sender. The recipient of the mail message is identified by RCPT command. To specify more than one recipient, a series of RCPT commands are issued. Each recipients email address is specified as an argument to the RCPT command after the word TO:. Then, the client issues a DATA command to inform a server that the client is ready to start sending the content of the message. The mail data is terminated with a period (".") character on a new line. The client issues the QUIT command to close the connection. If the SMTP client has another message to send, it can issue a new MAIL command before closing the connection.

The mail message format is defined in RFC 5322. Each email message consists of a message header followed, optionally, by a message body. Both parts are represented in ASCII text. The message header is formed from individual header fields (a sequence of lines). Each header field begins with a field name followed by a colon (":"), followed by a value and terminated by two characters CRLF: carriage-return (CR) and line-feed (LF). As shown in example below, message header must have From: header field and To: header field. The Subject: header field as well as other header fields are optional.

```
From: sender@e-mail.lt
To: recipient@mailserver.lt
Subject: A test message.
```

Note that the header fields From: and To: are not the same as the SMTP commands used to send a message. These fields are part

of the mail message and are not related to SMTP protocol. The message body is a sequence of characters that follows the header section and is separated from it by an empty line, i.e., a line with nothing preceding the CRLF.

To retrieve emails from mail servers the other protocols are used. POP3 (defined in RFC 1393) and IMAP (defined in RFC 3501) are the two most commonly used mail protocols for retrieving emails. POP3 is the simplest one. It is used by user agent (client) to retrieve messages from the server to the local computer. This allows to read downloaded message even when you are offline. The client initiates a TCP connection to the POP3 server application on the mail server, port 110. When the connection is established, the POP3 server sends a greeting. A POP3 session progresses through three states: *authorization*, *transaction* and *update*. In the authorization state, the client must identify itself to the POP3 server by sending a username and password. After successful authentication the session enters the transaction state where the client can list and retrieve messages, obtain mail statistics or mark retrieved messages for deletion. When the client issues the QUIT command, the session enters the third, update, state which concludes POP3 session. In the update state the server deletes the messages marked for deletion in the transaction state.

An example of the POP3 session is shown in the Figure 8.2 (S stand for server, C stands for client). The client uses the USER and PASS commands to enter the user's name and password. POP3 commands are not case sensitive and consist of four characters, possibly followed by one or more arguments. The server always responds to the each command with a two possible responses: +OK response indicates that the command was executed successfully, -ERR response indicates errors. Response usually is followed by additional information. The LIST command returns the list of messages in the mailbox. The RETR command, followed by a space and an integer, is used to retrieve specified message to local computer. The DELE

command is used to mark for deletion the specified message from the server. The QUIT command terminates the POP3 session and tells the server to delete all the messages that have been marked for deletion by using the DELE command.

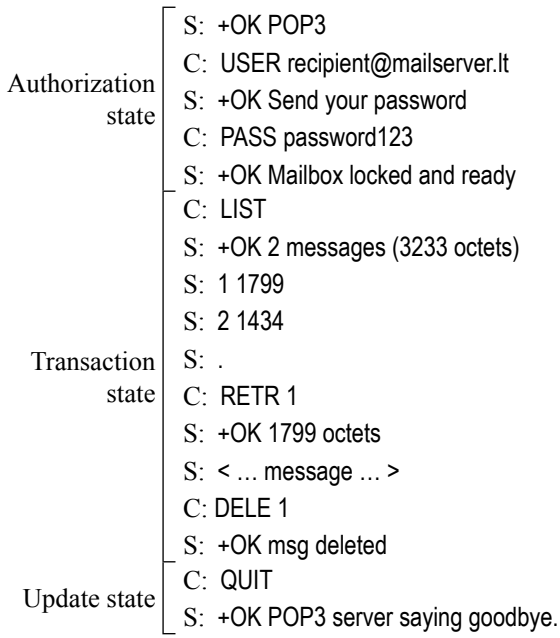


Fig. 8.2. An example of POP3 session

The IMAP4 is an alternative protocol to POP3. It provides more functionality than is available through POP3, but it is much more complex protocol. IMAP4 protocol allows not only retrieve messages from a server, but also keep them on the server and manipulate the remote message folders (or mailboxes) in which these messages are stored. IMAP4 protocol also includes operations for creating, deleting, and renaming mailboxes, checking for new messages, permanently removing messages etc. In addition, IMAP protocol allows multiple users access the same mailbox simultaneously.

Work assignment and methodical guidelines

1. Read attentively the theory section of laboratory work. Before starting the work discuss all obscure questions with the lecturer.
2. Configure the e-mail client application. The *Outlook Express version 6* e-mail client will be used in this task.
   2.1. Click **Start → All Programs → Outlook Express**. If this is a first time you start the *Outlook Express*, then the *Internet Connection Wizard* window will appear. Click *Cancel* and answer *Yes* to the next dialog box. If the *Outlook Express* was opened before, then the *Internet Connection Wizard* will not be displayed.
   2.2. Click **File → Identities → Manage Identities…** in the *Outlook Express* menu. A *Manage Identities* window will appear. Then click *New* and type your name into the text field. *Require a password* check box leave unchecked and press *OK*. Answer *Yes* to the question "*Do you want to switch to 'Your name' now?*".
   2.3. New *Internet Connection Wizard* will open. Type in your name and click *Next*.
   2.4. Type in *pc**X**@e-mail.lt* email address. Here **X** is the number of your computer. If your computer number is 5, then type *pc5@e-mail.lt*. Click *Next*.
   2.5. Select incoming mail server POP3 and type in incoming and outgoing server addresses. In this case, preconfigured mail server is located inside the LAN and its domain name is not registered on the DNS. Therefore, to access email server you should use IP address instead of domain name. In general the domain name is used instead of IP address. Type in the same 192.168.0.199 IP address into incoming and outgoing server text fields. Click *Next*.
   2.6. Type in full *pc**X**@e-mail.lt* email address as account name and *pc**X*** as password. Recall that X is the number of your computer. Click *Next* and *Finish*.

2.7. You should see one unread message in the *Inbox* folder. This is a welcome message from *Microsoft Outlook Express* team. Select it and press *Delete* button.

2.8. Click **Tools → Options** in the *Outlook Express* menu. Select *Send* tab in the displayed *Options* window. Then select *Mail Sending Format* as *Plain Text*.

2.9. Now try to create and send a new test message to yourself. Click *Create Mail* button, type *pcX@e-mail.lt* email address into *To* text field and type *Test* into *Subject* field. Write some message and click *Send* button.

2.10. Click *Send/Recv* button (or F5 button on the keyboard) to receive new messages from the server. If you have not received the message sent in previous task, it means that some *Outlook Express* configurations are incorrect. Check the previous configuration steps.

3. Analyse SMTP protocol used to send email message.

3.1. Start up the *Wireshark* network protocol analyzer. Click **Edit → Preferences** on the *Wireshark* menu. Expand protocols list and select SMTP protocol. Then deselect *Reassemble SMTP command and response lines spanning multiple TCP segments* and *Reassemble SMTP DATA commands spanning multiple TCP segments.* Finally click *Apply* and *Ok*.

3.2. Type in *smtp* into display filter toolbar and start packet capture.

3.3. Create and send the same message as in previous (2.8) task. Then stop *Wireshark* packet capture.

3.4. Analyse captured packets and fill the following table with client–server communication information:

No.	Server / Client	Response code / Command	Response parameter / Request parameter
1.	Server	220 Service ready	E-MAIL_SERVER ESMTP
2.	Client		

3.	Server		
4.	Client		
…	…	…	…

4. Analyse POP3 protocol used to retrieve email message from the server.
   4.1. Type in *pop* into *Wireshark* display filter toolbar and start packet capture.
   4.2. Click *Send/Recv* button (or F5 button on the keyboard) in the *Outlook Express* toolbar to receive a message from the server. Stop packet capture when the *Outlook Express* will receive a message.
   4.3. Examine captured packets and fill the following table with client–server communication information:

No.	**Server / Client**	**Response indicator / Request command**	**Response description / Request parameter**
1.	Server	+OK	POP3
2.	Client		
3.	Server		
4.	Client		
…	…	…	…

5. Use the Windows command line *Telnet* tool to test SMTP server operation. *Telnet* is useful tool for troubleshooting issues related to SMTP and mail flow. Using *Telnet* you can see the success or failure of each step in the connection and message submission process.
   5.1. Click **Start → All Programs → Accessories → Command Prompt** to open Windows terminal.
   5.2. Use *telnet* command to open *Telnet* session. Type *set logfile* <*filename*>. This optional command enables logging of the *Telnet* session to the specified log file. The location of the

log file is the current working directory. To view all available *Telnet* set options use *set ?* command.

5.3. Type *open 192.168.0.199 25* to connect to SMTP server. Now you can use the same SMTP commands, which you found by analyzing SMTP protocol (Task 3.4).

5.4. Issue the following SMTP command sequence:

Note 1: The commands in *Telnet* client are not case-sensitive. Here the SMTP commands are capitalized for clarity.

Note 2: *Telnet* does not have a full-featured text editor. If you make a mistake and then backspace to correct the mistake, the command may not be recognized. In most cases you will receive a command error. You must press *Enter* and then type the command again.

HELO YourName

250 Hello.

MAIL FROM: pcX@e-mail.lt

250 OK

RCPT TO: pcX@e-mail.lt

250 OK

DATA

354 OK, send.

<Here press *Enter* to leave one blank line>

This is a SMTP server test message.

. <Note: Here a dot character '.' is used to indicate the end of the message body>

250 Queued (40.047 seconds)

QUIT

Response should be as follows: 221 goodbye

5.5. Type *quit* command to terminate *Telnet* session. Use the content of *Telnet* logfile for the report.

5.6. Before you receive this message using *Outlook Express*, click **Tools → Accounts…** in the *Outlook Express* menu.

Select your account and click *Properties*. Then select *Advanced* tab and check the "*Leave a copy of messages on server*" and "*Remove from server when deleted from 'Delete Items'*" boxes. This tells *Outlook Express* to leave a copy of message on server until you delete it from the *Delete Items* folder.

5.7. Now click *Send/Recv* button to receive a message from the server. Does the *Outlook Express* display the sender and the subject of the message?

5.8. Use the *Telnet* tool to send another message, but in this case specify the sender, recipient and the subject in the message body as shown below:

…
DATA
354 OK, send.
From: pcX@e-mail.lt
To: pcX@e-mail.lt
Subject: Test message
<Here press *Enter* to leave one blank line>
This is a SMTP server test message.
.
250 Queued (40.047 seconds)
QUIT
…

5.9. Click *Send/Recv* button (or F5 button on the keyboard) in the *Outlook Express* toolbar to receive a message from the server. Does, in this case, the *Outlook Express* display the sender and the subject of the message?

6. Use the *Telnet* tool to test POP3 server operation.

6.1. Type *telnet* command into Windows terminal window to open *Telnet* session. Then type *set logfile <filename>* to define a log file to capture the text of the session.

6.2. Type *open 192.168.0.199 110* to connect to POP3 server. Now you can use the same POP3 request commands, which you found by analyzing POP3 protocol (Task 4.3).

6.3. Issue the following POP3 command sequence:

USER pcX@e-mail.lt
+OK Send your password
PASS pcX
+OK Mailbox locked and ready
LIST
+OK 2 messages (344 octets)
1 182
2 162
.
RETR 1
+OK 182 octets

<Here is a message body >

DELE 1
+OK msg deleted
QUIT
+OK POP3 server saying goodbye...

6.4. Use the content of *Telnet* logfile for the report.

7. Analyse secured SMTP and POP3 communication by using *Secure Sockets Layer* protocol.

7.1. Click **Tools → Accounts…** in the *Outlook Express* menu. Select your account and click *Properties*. Then select *Advanced* tab and check both "*This server requires a secure connection (SSL)*" boxes for SMTP and POP3. In addition, type 465 port number for SMTP server and 995 port number for POP3 server. These are the default port numbers for SMTP and POP3 over SSL. Then click *Apply* and *OK*.

7.2. Click *Send/Recv* button in the *Outlook Express* toolbar to verify that email client can make a connection to the server. The *Internet Security Warning* message should appear which tells that the server you are connected to is using a security certificate that could not be verified. This is because the server is using a self-signed certificate for test environments which is not signed by a trusted Certificate Authority. Answer *Yes* to this warning message.

7.3. Type in *ssl* into *Wireshark* display filter toolbar and start packet capture.

7.4. Create and send a test message to yourself using *Outlook Express*.

7.5. Click *Send/Recv* button to receive a message from the server and stop *Wireshark* packet capture.

7.6. Examine capture packets and answer the following questions:

7.6.1. What protocol is used to secure communications between email client and server?

7.6.2. What information about the email sender, recipient or message content can be found by analysing captured packets?

7.7. Remove created email account from the *Outlook Express*. Click **File → Identities → Switch Identity…** in the *Outlook Express* menu. Select *Main Identity* and click *OK*. Then Click **File → Identities → Manage Identities…**, select your account and click *Remove*. Close *Outlook Express*.

Content of report

1. Objectives of the work.
2. The analysis results of the SMTP protocol (Task 3.4).
3. The analysis results of the POP3 protocol (Task 4.3).
4. The analysis results of the SMTP server operation using *Telnet* (Tasks 5.4–5.5, 5.7–5.9).

5. The analysis results of the POP3 server operation using *Telnet* (Tasks 6.3–6.4).
6. The analysis results of the secured SMTP and POP3 communication (Task 7.6).
7. General conclusions of the work.

Review questions and problems

1. What are SMTP, POP3 and IMAP4? What are differences between POP3 and IMAP4?
2. Explain the process of email creation and delivery.
3. Explain a mail message format. Give an example.
4. Name and explain the SMTP commands which are used to send a mail.
5. Name and explain the POP3 commands which are used to retrive a mail.
6. Name and explain the POP3 session states.
7. Are the SMTP and POP3 protocols secure enough to use in nowadays networks? Explain.

Literature

Dostálek, L., Kabelová, A. 2006. *Understanding TCP/IP: A clear and comprehensive guide to TCP/IP protocols*. Packt Publishing. 480 p. ISBN-13: 978-1904811718.

Kurose, J. F.; Ross, K. W. 2009. *Computer Networking: A Top-Down Approach*. 5th edition. Addison Wesley. 864 p. ISBN-13: 978-0136079675.

Laboratory work 9

Design of Local Area Computer Network Using GNS3

Objectives

Becoming acquainted with Graphical Network Simulator GNS3 and using it to simulate computer networks.

Basic knowledge and theory

Router

Router is the network interconnecting device that is able to handle data routing. Routing is a process of moving data along the entire network from the source to the destination. Router is usually a specialised computers with a specific operating system (e.g.: Cisco IOS, Juniper Networks JUNOS or other), RAM used in routing operations, NVRAM used to store startup configuration, flash used to store image of the operating system, one or more CPUs and different network interfaces.

Routers interconnect two or more networks, in order to do that they poses, accumulate and exchange routing information. They store that information in the routing tables and send the network packets through the route to destination.

Routing consists of two functions: optimal route selection and data packet transmission from source to destination (packet switching). Packet switching is simpler process than route selection.

Routing is performed in OSI model layer three (network layer).

Performing network interconnection router can get flooded with the packets exceeding its capacity. Decision how to act in such situation is important and there are three of them used: *Tail Drop* – last packet in the queue is dropped, *Random Early Detection* – the probability that the packet will exceed the buffer is calculated using statistical algorithms before the packet arrives and *Weighted Random*

Early Detection – uses random early detection algorithm extended to evaluate the priority of the packet.

The first device which had similar functionality was Interface Message Processor designed for the ARPANET. Device could interconnect different physical medias and perform the switching without delivery assurance. The first IP router was created by BBN Technologies in 1976.

Cisco

Cisco Systems was established in 1984 at Stanford University by the Len Bosack and Sandy Lerner family. Company name arose from the city name – San Francisco.

Cisco was the first company selling multiprotocol routers. Company grew fast, mostly by acquiring other companies. It was the most expensive company during the .com boom in 2000 (500 billion USD). It stays to be the biggest network equipment producer.

Cisco started using modular systems in its equipment in 1990 and in order to upgrade the equipment, only module upgrade was needed. PCMCIA slots and GBICs (*GigaBit Interface Converter*) were started to be used in later years.

Cisco 3640 router

This router belongs to the industry's first multifunction platform. Support of the different network modules makes this router universal. Among other supported modules there are the switch modules, one of them NM-16ESW will be used during the switching laboratory work. Cisco 3640 router is presented in Figure 9.1.

Cisco Internetwork Operating System IOS

Cisco IOS is used on Cisco routers, switches and other network equipment. OSes of other network devices look similar or even align to IOS. IOS has command line interface (CLI) allowing inputting multiple-word commands.

a) b)

Fig. 9.1. Cisco 3640 router: a) front view, b) rear view

Cisco IOS as other OSes has different versions, but because of different network device types and functionality needed versioning is more complex. General form of the version is *a.b(c.d)e*, where: *a* is the major version number; *b* is the minor version number; *c* is the release number; *d* is the interim build number; *e* is the release train (set of features) identifier.

Cisco IOS configuration is usually performed from Command Line Interface, graphical wizards are also available. IOS has different configuration modes.

User executive (also known as unprivileged) mode commands allow you to connect to remote devices, change terminal settings on a temporary basis, perform basic tests, and list system information. The EXEC commands available at the user level are a subset of those available at the privileged level.

Privileged executive commands set operating parameters. The privileged command set includes those commands contained in user EXEC mode, and also the configure command through which you can access the remaining command modes. Privileged EXEC mode also includes high-level testing commands, such as debug.

Global configuration commands apply to features that affect the system as a whole.

Interface configuration commands modify the operation of an interface such as an Ethernet or serial port. Many features are enabled on a per-interface basis. Interface configuration commands always follow an interface global configuration command, which defines the interface type.

ROM monitor commands are used to perform low-level diagnostics. You can also use the ROM monitor commands to recover from a system failure and stop the boot process in a specific operating environment

Graphical view of the different modes and how to get from one mode to another with the CLI line is presented in Figure 9.2. To negate a command (delete line from configuration) add *no* to the beginning of the command.

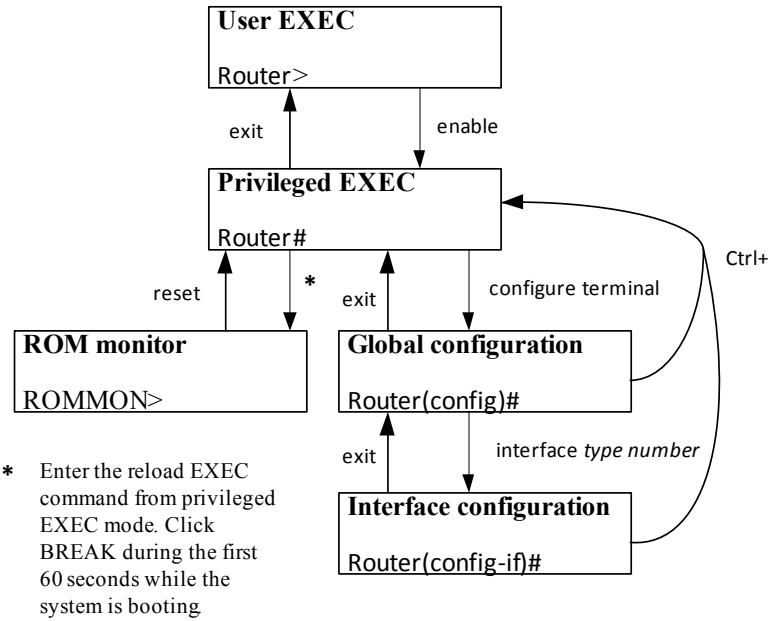


Fig. 9.2 Cisco IOS configuration modes

As an example let's review a common task – configuring network interface: first you need to go to the privileged mode, then to global configuration mode, then to interface configuration mode, input the IP addressing information, to enable the device (state of the interface after configuration is disabled), then return to privileged mode, review the configuration done (this can be done only from the

privileged mode, some command from unprivileged too) and write the configuration to starting configuration file (one of the ways to do this is using *write* command).

R1>enable
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface FastEthernet 0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)# exit
R1#show interface f0/0
FastEthernet0/0 is up, line protocol is up
Hardware is AmdFE, address is cc00.067c.0000 (bia cc00.067c.0000)
Internet address is 192.168.1.1/24
…… (the rest of the output is cut)
R1#write
Building configuration...
[OK]
R1#

To see the possible options of the command type: *command ?*. Let's see available options for *show* command:

R1#show ?
aaa Show AAA values
aal2 Show commands for AAL2
access-expression List access expression
access-lists List access lists
…… (the rest of the output is cut)

Pressing the Tab will end the command if it is unambiguous. Command can be entered by typing only the beginnings of the commands and interface names in shortened forms (e.g.: Ethernet 0/0 – e0/0, FastEtherner 0/1 – f0/1):

R1>
R1>en
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int f0/0
R1(config-if)#ip add 192.168.1.1 255.255.255.0
R1(config-if)#no shut
R1(config-if)#ex
R1(config)#int f0/0
R1(config-if)#^Z
R1#sh int f0/0
FastEthernet0/0 is up, line protocol is up
Hardware is AmdFE, address is cc00.067c.0000 (bia cc00.067c.0000)
Internet address is 192.168.1.1/24
…… (the rest of the output is cut)
R1#wr
Building configuration...
[OK]
R1#

Graphical network simulator GNS3

GNS3 is a free, open source, graphical network simulator capable simulating complex computer networks. The tools alike the computer virtualization programs allow routers to function in the virtual environment, to connect desired topology and test its performance.

GNS3 is graphical interface of *Dynagen* application. *Dynagen* is text network configuration generator for *Dynamips*. *Dynagen*

uses initialization files which contain information about router port number and type; it allows device access via command line, turns on and off the devices, allows capturing network traffic and etc. Cisco router simulation itself is performed by *Dynamips*. GNS3 is capable to simulate different Cisco routers and firewalls, Juniper routers, to use basic network switches or complex ones realized as a Cisco router with high number of ports. GNS3 emulates routers using IOS as the physical router but is not able to replace the actual router in the production environment because of Cisco licensing and small 1000 pps (packets per second) throughput.

Main GNS3 window (Fig. 9.3) consists of:

1. Main menu, containing links to the options of the simulated project and the GNS3 package itself;
2. Tools menu, containing management and editing options of the simulated project;
3. Nodes types which can be used in the project;
4. Network topology window;

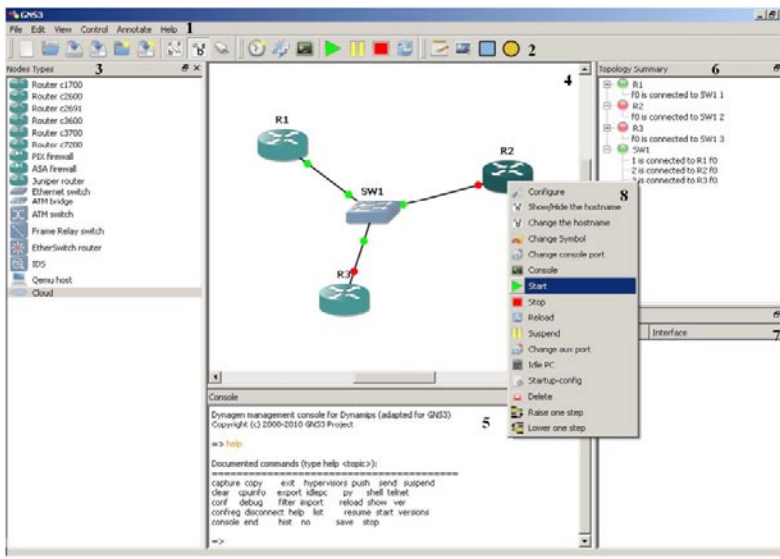


Fig. 9.3. GNS3 main window

5. *Dynagen* management window;
6. Topology summary window which reflects states and interconnections of the topology elements;
7. Network capture window;
8. Topology element contextual menu.

Let's review Tools menu buttons from left to right: new blank topology, open project or topology file, save project or topology file, save topology file as another name or directory, new blank project, save project as another name or directory, show interface labels, show hostnames, add a link (adding a link you need to choose link type; to leave link adding mode click the icon again), take a snapshot, import or export startup configuration, connect to AUX (auxiliary) consoles of all devices, connect to all devices consoles, start or resume all devices, suspend all devices, stop all devices, reload all devices, add a note, add a picture, draw a rectangle, draw an ellipse.

Let's review Main menu options from left to right:

File menu (Fig. 9.4a) allows to start a new blank topology, open and save GNS3 file, save GNS3 topology as another name or directory, start a new blank project, save project as another name or directory (project name, directory and saving router NVRAM as file and saving router startup configuration (that is needed for transferring configuration to other routers, starting configuration needs to be imported, process is not automatic)), import of exported configurations of all the routers, make a screenshot, make a snapshot of the moment (there might be few snapshots and choosing among them is possible) and quit a program.

Edit menu (Fig. 9.4b) allows undoing and redoing changes, selecting all, deselecting all, configuring IOS images and hypervisors, editing symbols and setting the options of the program.

View menu (Fig. 9.4c) allows to zoom in and out, go back to the original size, choose if hostnames and interface labels are shown, define the visible main window elements.

Control menu (Fig. 9.4d) options affect all the devices in the topology, they can be turned on and off, paused, reloaded or connected to using the console.

Annotate menu options allows creating symbolic labels, help provide help on using the program.

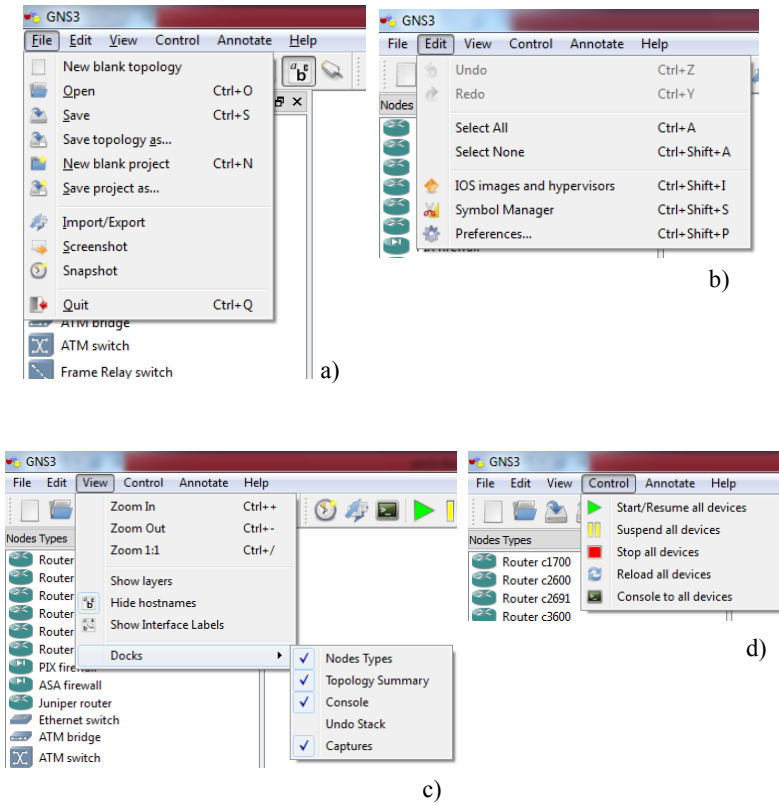


Fig. 9.4. GNS3 Main menu options: a) File, b) Edit, c) View, d) Control

Work assignment and methodical guidelines

1. Read attentively the theory section of laboratory work. Before starting the work discuss all obscure questions with the lecturer.

2. Prepare the GNS3 simulator.
   2.1. Create a directory and rename it to *N.Surname* in *My Documents* directory. Start GNS3.
   2.2. Name the project as the number of your variant, as project directory choose the directory you created. Check both ticks as the first will save the configuration in routers memory, the second will save the configuration in the file, which can be used to import it to other routers if needed (Fig. 9.5).

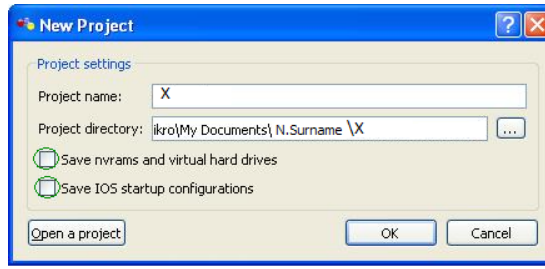


Fig. 9.5. New Project window

   2.3. Router images are compressed, in order the boot of the routers to be faster it was decompressed using the tool unpack.exe. Add the IOS image of the router: **Edit → Images and hypervisors**, then choose the image named *c3640-io3s56i-mz.120-7.T.bin.unpacked* (Fig. 9.6). Delete *Base Config* value as such file is not created and then press *Save*.
3. Configure the GNS3 package itself. Choose **Edit → Preferences**, then *Dynamips* (Fig. 9.7). Check the tick *Enable ghost IOS support*, that will allow to use the same IOS image in different routers and that will reduce RAM consumption. Check the state of virtualization component *Dynamips* by pressing the *Test* button, the response indicating success should be indicated.

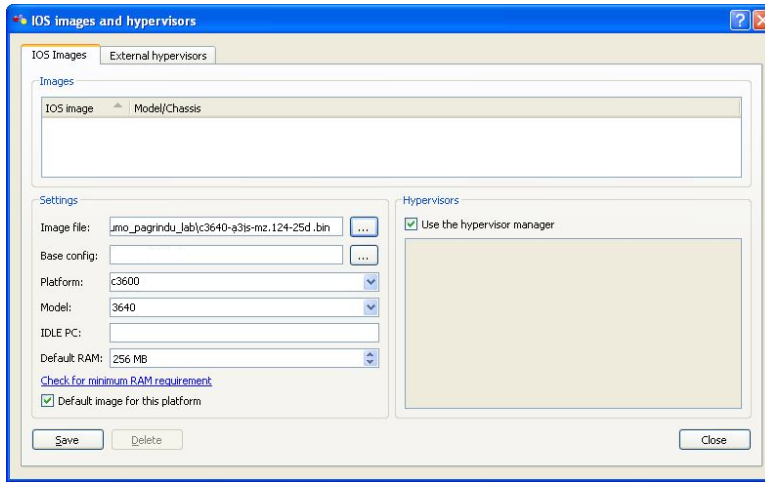


Fig. 9.6. IOS image and hypervisors configuration window

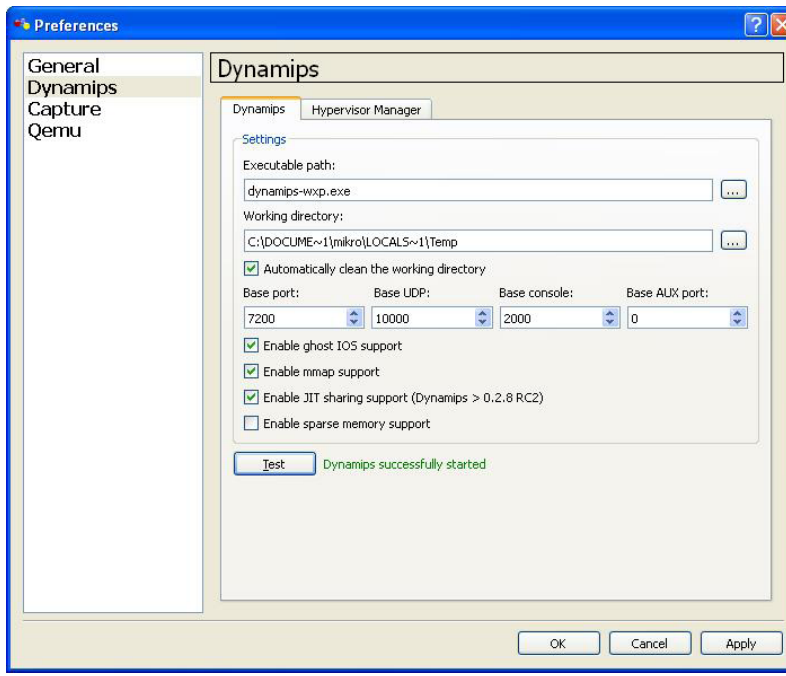


Fig. 9.7. *Dynamips* configuration window

Capture tab allows setting the capture directory to which all the intercepted network packets go. *Qemu* tab can be used to configure network devices other than routers. Set the path to quemu image in *Quemu Host* tab (*C:\Documents and Settings\...\linux-microcore-3.8.2.img*), choose name PC and save the configuration. Test *Qemu* preparedness by pressing *Test* in *General Settings*. Close the window by pressing OK.

4. Add the router to the network by dragging the corresponding icon. Power it on by pressing the green triangle. Check the amount of used CPU and RAM on the Laboratory PC (**Windows Task Manager → Performance**), what process related to GNS3 is using it. Press left mouse button on the router object and select Idle PC. GNS3 will count most probable Idle-PC value and mark it with a star, select it. Idle-PC value depends on router IOS and configuration used. It identifies router process on the physical computer. Check the CPU load again; the load should be reduced drastically.
5. Change the name of the router: stop it at first by pressing red rectangle icon in the toolbar and then click the left mouse button on router object and choose *Change the hostname*, name the router with your variant number word.
6. Add *Ethernet switch* a component. Connect routers FastEthernet 0/0 interface to switch port 1 by pressing *Add Link* button on the toolbar. Check the configuration of the switch, what port types are available.
7. Start the router by pressing the green triangle and connect to the router console by pressing the black button on the toolbar. New putty window will open, click with the mouse in it. If the configuration wizard will be suggested, refuse it by entering *No* and pressing *Enter*.
8. Enable privileged mode with the command *enable* and the configuration mode with: *configure terminal*.
   (variant name)>enable

(variant name)#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
(variant name) (config)#

8.1. If the hostname of the router is not set by configuring set hostname in the Task 6, change the hostname using command: *hostname (variant name)* from the configuration mode.

9. Configure network interface by choosing it, configuring IP address and activating the interface (variant number X is used in your IP address):

(variant name) (config)#interface FastEthernet 0/0
(variant name) (config-if)#ip address 192.168.X.1 255.255.255.0
(variant name) (config-if)#no shutdown

10. Leave the interface configuration mode and return to configuration mode with: *exit*. Router can show some informational announcement and not return a carriage, in that case press *Enter*.

(variant name) (config-if)#exit
(variant name) (config)#exit
(variant name)#

11. Check the router configuration.
    11.1. Check how network interface is configured: show ip interface brief. Analyze the presented information. Check the more detail interface information: show interfaces.
    11.2. Check the running configuration: show running-config. To scroll one line down press Enter, to scroll down a whole page – Space.
12. Secure the device by configuring the passwords.
    12.1. Configure activation password – *cisco*, console (*line console 0*) – (*variant name*) as a password and telnet (*line vty 0-4*) password – *lab*. Configuration must be performed from the configuration mode. Activation password can be saved in open or ciphered form. All other passwords are in

open form, addition services must be turned on in order to cipher them. Passwords can be enabled with this sequence of the commands:

(variant name) >enable
(variant name) #configure terminal
(variant name) (config)#enable secret cisco
(variant name) (config)#line console 0
(variant name) (config-line)#login
(variant name) (config-line)#password (variant name)
(variant name) (config-line)#line vty 0 4
(variant name) (config-line)#login
(variant name) (config-line)#password lab
(variant name) (config-line)#exit

12.2. Disconnect from privileged mode and connect with entering the passwords.

13. Check the *running configuration*, then *starting configuration*. Save the *running configuration* to the *starting configuration*. Analyse the *running configuration*.

(variant name) # show running-config
(variant name) # show startup-config
(variant name) # copy running-config startup-config
(variant name) # show startup-config

14. Add and configure the *Quemu* host.
    14.1. Add the *Quemu* host and connect it to the switch port 2 using FastEthernet link. Use e0 network interface on the *Quemu* host. Power on the *Quemu* host and wait till it loads. Progress will be seen in newly opened window. Loading can take some time.
    14.2. Clicking on the *Quemu* host window allow you to configure the host. To regain the control of your mouse press *Ctrl*

and *Alt* on the right at one time. For numeric values use the number row on the top of the keyboard.

14.3. Configure network interface eth0, it corresponds to e0 used to connect to the router:

sudo ifconfig eth0 192.168.X.5 netmask 255.255.255.0.

14.4. Check the configuration: *ifconfig eth0*.

14.5. Send ICMP packets using ping command: *ping 192.168.(variant name).1*, to stop the pinging press: *Ctrl+C*.

14.6. Connect to the router using telnet command: *telnet 192.168.(variant name).1* input the configured password.

15. If the pinging and telneting succeeded exit the GNS3 saving the topology and configuration. Check if the laboratory work was saved by loading it and checking the configuration on the router. To save the router configuration use the command *write*, saving of the *Qemu* settings is complicated, so we will not save them.

Content of report

1. Objectives of the work.
2. Review the abilities of the GNS3 package.
3. Describe the process of computer resource optimization (Task 4).
4. Describe GNS3 switch configuration options (Task 6).
5. What is the sequence of commands used to configure network interface (Tasks 8, 9 and 10)?
6. How the status of the network interfaces can be checked (Task 11)?
7. What types of router access passwords can be configured (Task 12)?
8. What files contain router configuration, how they differ, how to save a configuration (Task 13)?

9. How the network configuration of the *Qemu* host is performed (Task 14)?
10. General conclusions of the work.

Review questions and problems

1. What applications are incorporated in to GNS3 package, what is their purpose?
2. What are the shortcomings of GNS3 package?
3. What are router configuration modes, what are they used for?
4. How the router can be accessed for configuration?

Literature

Cisco 3640 router data sheet. http://www.cisco.com/en/US/products/hw/routers/ps274/products_data_sheet09186a0080091f6f.html.

GNS3 Tutorial. http://downloads.sourceforge.net/gns-3/GNS3-0.5-tutorial.pdf?download.

Olifer, N., Olifer, V. 2006. *Computer Networks: Principles, Technologies and Protocols for Network Design*. Wiley. 1000 p. ISBN-13: 978-0470869826.

Singh, A. K. 2005. *Computer Network*. Laxmi Publications. 181 p. ISBN-13: 978-8170087021.

Laboratory work 10

Computer Network Routing Using Statical Routes and RIP Protocol

Objectives

Design a computer network which uses static routes and then enhance routing using RIP.

Basic knowledge and theory

Routing

Routing is a process of moving data along the entire network from the source to the destination.

Routing table is constructed in order to determine the best route of the packet. To construct the routing table differently obtained information is used: local interfaces (networks which are directly connected to the router), static routes (set by network administrator) and routing information received from other routers (using dynamical routing protocols).

Static routes are added by administrator of the computer network, they don't change without administrator intervention. When something changes in the network, administrator needs to reconfigure the routes according to these changes. If the network is complex then this manual task takes time and the probability of the mistake gets higher.

Dynamic routing protocols use metrics to determine the best route. Metric is a standard measure. Examples of metrics are throughput and the hop count. What metric is used is determined by the routing protocol used.

Routing protocols must know about the networks the router is connected to a priori, then communicating with their neighbours it will build the routing table after the convergence time which cor-

responds to the network. Convergence times differ because of the protocol and the network complexity.

Networks according have administration borders, one networks are administered by one organization, the others by the other, logic, technologies and protocols may differ. Networks of the Internet Service Providers (ISP) fall into to the Autonomous Systems (AS). Other Internet provider knowing that some networks belong to the first ISP AS routes all the designated traffic to these networks to that AS. If the routing is between ASes it is called exterior gateway routing. If the routing is within a single AS it is called interior gateway routing.

There are three major classes of the routing protocols in IP networks:

- Distance vector protocols for interior gateway routing, such as RIP, Interior Gateway Routing Protocol and Enhanced Interior Gateway Routing Protocol.
- Distance vector protocols for exterior gateway routing. Border Gateway Protocol is the routing protocol of the Internet.
- Link-state routing protocols for interior gateway routing, such as OSPF and Intermediate System to Intermediate System (IS-IS). These are the only ones link-state routing protocols.

Distance vector routing protocols constructs and manipulates vectors of distances to other hosts in the network. Routers do not have knowledge of the entire route to a destination. In order to construct the routing table it needs the direction (IP address of the next hop) in which or interface to which a packet should be forwarded and the distance from its destination (hop count). Router informs its neighbors of topology changes periodically.

Link State protocols monitor the status (reachable or not) and connection type (available throughput) of each connection and calculate metric based on these and some other factors. Link State protocols will use the more distant route having more hops if it uses a

faster medium. Link State protocols are aware of all the network topology.

Routing Information Protocol

Routing Information Protocol (RIP) is a standards-based (RFC 1058, RFC 1388 and others), distance vector, interior gateway protocol. RIP uses hop count as a metric in order to determine the best route between two networks. Hop count is the number of routers the packet goes through until it reaches the destination network. The maximum number of hops in RIP is 15. Hop count 16 means that the network is unreachable. RIP can be used on a small networks, but nowadays is rarely met because of the lack of functionality and security issues. Nonetheless RIP is the simplest routing protocol perfect for the introduction into routing.

Each router sends its entire routing table to its neighbouring routers every 30 seconds. Router receives a neighbour's routing table and uses this information to update its own routing table and then when time to send comes it sends the updated table to its neighbours.

There are two version of rip for IPv4: RIPv1 and RIPv2.

RIPv1 is a classful (does not support network mask, default network class mask is used) protocol, broadcasts updates every 30 seconds, hold-down period 180 seconds.

RIPv2 is a classless protocol (network masks can be used), uses multicasts instead of broadcasts used RIPv1, supports triggered updates (issued when a change occurs), also supports authentication between routers.

Work assignment and methodical guidelines

1. Variants. For this and other laboratory works will be used information from the Table 10.1.

Table 10.1. Task variants

Variant (X)	City
1	Vilnius
2	Kaunas
3	Klaipeda
4	Siauliai
5	Panevezys
6	Alytus
7	Marijampole
8	Mazeikiai
9	Jonava
10	Utena
11	Kedainiai
12	Telsiai
13	Taurage
14	Visaginas
15	Ukmerge
16	Plunge

2. Read attentively the theory section of laboratory work. Before starting the work discuss all obscure questions with the lecturer.
3. You are administering the network of your city in a global organisation. Network is presented on Figure 10.1. Your network is interconnected with other cities administered by your colleagues via WAN. User network is *172.17.X.0/24*. This network layout will be used in this and the following laboratory work. Addressing information is presented in Table 10.2. CIDR network mask is presented in the picture and the table.

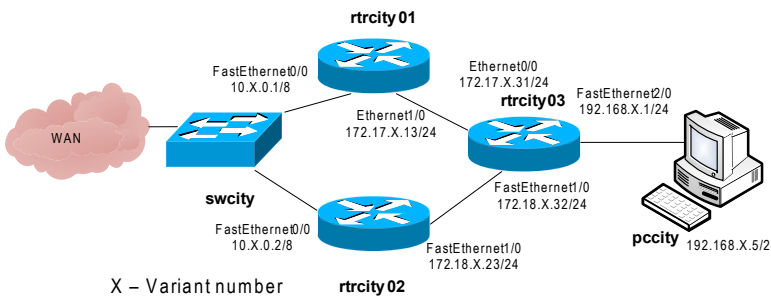


Fig. 10.1. Network diagram

Table 10.2. Addressing information

Device name	Interface	IP addressing
swcity	1, 2, 3	–
rtrcity01	Fe 0/0	10.X.0.1/8
	E 1/0	172.17.X.13/24
rtrcity2	Fe 0/0	10.X.0.2/8
	Fe 1/0	172.18.X.23/24
rtrcity03	E 0/0	172.17.X.31/24
	Fe 1/0	172.18.X.32/24
	Fe 2/0	192.168.X.1/24
pccity	E0 (eth0)	192.168.X.5/24

4. Construct a simulated network. Start the GNS3 package and choose to create a new project named *N.Surname_lab10*.

4.1. Perform a construction of the network based on the presented topology. Add the cloud element and configure it by pressing the right mouse button and choosing *Configure* (Fig. 10.2).

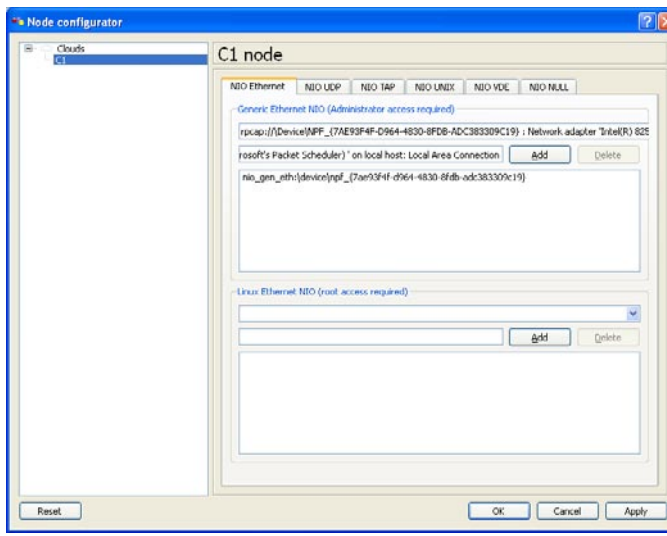


Fig. 10.2. Configuring cloud object

To WAN interface assign the existing network card on the physical PC (press ADD when you select it). Change the name of the cloud to WAN. Change the names of the other network objects with the name of the city defined by your variant.

Use the connection tool to interconnect the network elements (Fig. 10.3), choose the link types corresponding to the interface types (FastEthernet or Ethernet). Routers have more than one interface, please mind where they are connected.

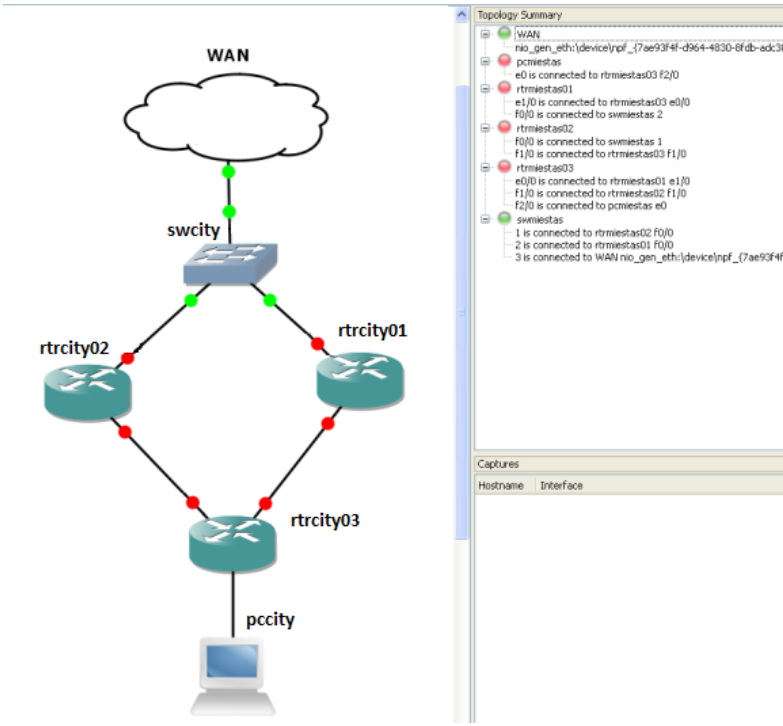
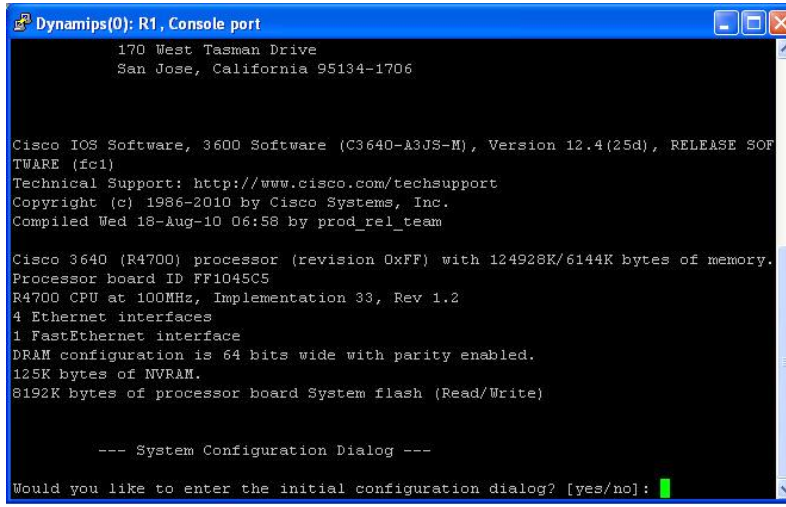


Fig. 10.3. Network view in GNS3

5. Start network simulation. Turn on the routers one by one, the colour of the router network interface will change from red to green. Activate router console by choosing console from the router drop down menu or turning on the console screens of all the routers by

choosing the appropriate button at the toolbar menu. This connection corresponds to COM connection to the router. First start of the router will bring the wizard screen, enter *no* at this screen followed by *Enter*. Following boots of the router will not bring this screen.



5.1. Review the networking information:

rtrcity01>show interfaces

Enter will shift the window one line and the *Space* whole page. To stop the review press Ctrl+c. What interfaces are present on the router, what equipment is used and what is their status?

5.2. Configure network interface: choose the particular interface, configure IP address and activate it. X in the address must be changed to your variant number:

rtrcity01 (config)#interface FastEthernet 0/0
rtrcity01 (config-if)#ip address 172.17.X.1 255.255.255.0
rtrcity01 (config-if)#no shutdown

5.3. Configure the other interface FastEthernet 1/0 with IP 10.X.0.1 and network mask 255.0.0.0.

5.4. Use *exit* to leave the interface configuration mode and to return to configuration mode:

rtrcity01 (config-if)#exit
rtrcity01 (config)#exit
rtrcity01 #

5.5. Check the configuration of network interfaces using: *show ip interface brief*. Ping all the configured interfaces.

5.6. Review the running configuration: *show running-config*.

6. Configure other routers and perform same checks (Task 5). Try pinging interfaces on routers *rtrcity01* and *rtrcity03* from the router *rtrcity03*. Which interfaces are reachable? Why some interfaces are unreachable?

7. Check the settings of Cisco Discovery Protocol CDP on the router rtrcity01: *show cdp*.

Find the configured neighbouring devices: *show cdp neighbors*. Analyse the presented information.

8. Configure the PC object pccity, set the IP address and the default route. What will be the default gateway?

8.1. Configure the IP address:

pc@box:~$ sudo ifconfig eth0 192.168.X.5 netmask 255.255.255.0

8.2. Check the IP configuration:

pc@box:~$ sudo ifconfig eth0
eth0 Link encap:Ethernet HWaddr 00:AA:00:3C:74:00
inet addr:192.168.X.5 Bcast:192.168.X.255 Mask:255.255.255.0

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:98 errors:0 dropped:0 overruns:0 frame:0
TX packets:60 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:32928 (32.1 KiB) TX bytes:19620 (19.1 KiB)

8.3. Configure the default route:

tc@box:~$sudo route add default gw 192.168.X.1

8.4. Check the routing table:

tc@box:~$ sudo route -n
Kernel IP routing table

Destination	*Gateway*	*Genmask*	*Flags*	*Metric*	*Ref*	*Use Iface*
127.0.0.1	*0.0.0.0*	*255.255.255.255*	*UH*	*0*	*0*	*0 lo*
192.168.X.0	*0.0.0.0*	*255.255.255.0*	*U*	*0*	*0*	*0 eth0*
0.0.0.0	*192.168.X.1*	*0.0.0.0*	*UG*	*0*	*0*	*0 eth0*

9. Configure routers in such a way, that PC pccity WAN network was accessing via router rtrcity01, and if it fails via rtrcity02. Routers rtrcity01 and rtrcity02 needs to know how to reach network 192.168.X.0/24.

9.1. Configure two default routes with different priorities on router rtrcity3, configuration must be performed from the configuration mode:

rtrcity03 (config)# ip route 0.0.0.0 0.0.0.0 172.17.X.13
rtrcity03 (config)# ip route 0.0.0.0 0.0.0.0 172.18.X.23 100

9.2. Review routing table (from the privileged mode) *show ip route*. Notice that it is shown only the default route which is in use. What do symbols C, S and * represent?

9.3. Add routes to the PC network on routers rtrcity01 and rtrcity02:

rtrcity01 (config)# ip route 192.168.X.0 255.255.255.0 172.17.X.31
rtrcity02 (config)# ip route 192.168.X.0 255.255.255.0 172.18.X.32

9.4. Send ICMP packets from the PC pccity to WAN interface (Fe0/0) of the router rtrcity01using *ping* command and check the route to it using *traceroute* command. Ping replies must return and the route must go via rtrcity01.

10. Shutdown interface connecting router rtrcity03 to router rtrcity01 (E0/0) and check how this impacted rtrcity03 routing table.

rtrcity03(config)# interface Ethernet 0/0
rtrcity03(config-if)#shutdown

10.1. Press Ctrl+z to return to privileged mode:

rtrcity03 (config-if)#^Z
rtrcity03#

10.2. Check the states of the interfaces:

rtrcity03#show ip interface brief.

10.3. How the state of the interface Ethernet0/0 changed? Check the routing table using *show ip route* command.

11. Send ICMP packets from the PC pccity to WAN interface (Fe0/0) of the router rtrcity02 using *ping* command and check the route using *traceroute* command. Test must be successful.

11.1. Send ICMP packets from the PC pccity to WAN interface (Fe0/0) of the router rtrcity01 using *ping* command. Is a reply coming back? What route leads to that interface (which devices and interfaces, what networks are in-between)? The problem is that router rtrcity01 has the information, that network 192.168. X.0 is reachable via router interface with IP address 172.17. X.31, and this route is non-functional because interface E0/0 is turned off.

11.2. Turn off the misleading route and add the correct one:

rtrcity01 (config)# no ip route 192.168.X.0 255.255.255.0 172.17.X.31
rtrcity01 (config)# ip route 192.168.X.0 255.255.255.0 10.X.0.2

11.3. Check the accessibility of rtrcity01 WAN interface repeating the steps described in 11.1.

11.4. Check if WAN interface (Fe0/0) of router rtrcity01 is accessible from *rtrcity03*, why?

12. Add the route to the neighbouring city on router rtrcity02:

rtrcity02 (config)# ip route 192.168.(X+1).0 255.255.255.0 10.(X+1).0.1

When your colleague will configure neighbouring cities route send ICMP packets from pccity to neighbouring cities PC.

13. Restore initial network configuration.

13.1. Remove all static routes on router rtrcity01. Find them reviewing running configuration with a command *show running-config*. Mind the configuration modes.

rtrcity01 (config)#no ip route 172.18.X.0 255.255.255.0 10.X.0.2
rtrcity01 (config)#no ip route 192.168.X.0 255.255.255.0 172.17.X.31
rtrcity01 (config)#no ip route 192.168.X.0 255.255.255.0 10.X.0.2

13.2. Enable Ethernet 0/0 interface on router rtrcity03 with: *no shutdown*.
13.3. Static routes from the remaining routers will be removed later using the same command.
14. Enable dynamic routing protocol – RIP (Routing Information Protocol) on routers rtrcity01 and rtrcity03; define all the networks which are connected to the router directly:

rtrcity (config)#router rip
rtrcity (config-router)#network XXX.YYY.ZZZ.QQQ

E.g.:
rtrcity01#
rtrcity01#conf t
rtrcity01 (config)#router rip
rtrcity01 (config-router)#network 10.X.0.0
rtrcity01 (config-router)#network 172.17.X.0

rtrcity03 (config)#router rip
rtrcity03 (config-router)#network 192.168.X.0
rtrcity03 (config-router)#network 172.17.X.0
rtrcity03 (config-router)#network 172.18.X.0

15. Analyse the routing tables on both routers: *show ip route*. Notice how new routes appear in the routing table after some time.
16. Check how the routes from the PC pccity to routers rtrcity01 and rtrcity02 WAN interfaces (Fe0/0) changed using *traceroute*. How did the route change and why (take in mind the enabled interface and route administrative distances)?
17. Disable the static routes on rtrcity02 and rtrcity03 and configure RIP on router rtrcity02.
18. Review the routes (*show ip route*) and RIP status on all the routers, to review the status use *show ip protocols*. What networks

were added by you and what networks are advertised by RIP, why?

18.1. Check how the routes from the PC pccity to routers *rtrcity01* and *rtrcity02* WAN interfaces (Fe0/0) changed using *traceroute.*

18.2. Try reaching neighbour city router WAN interface (Fe0/0) using *traceroute*. Are they accessible, why? To stop traceroute you can press the combination *Ctrl+C*.

19. Enable RIP debugging on router rtrcity01: *debug ip rip*. Which RIP version is used by default? What address is used to send RIP information, which addresses are used to receive it? What metric is used and how frequently the updates are sent? Stop the debugging by running the command: *no debug all*.

20. Configure RIP version 2 to be used on all the routers with no automatic network summarization:

```
rtrcity01#conf t
rtrcity01(config)#router rip
rtrcity01 (config-router)#version 2
rtrcity01 (config-router)#no auto-summary
```

21. Review the routes (*show ip route*) and RIP status (*show ip protocols*) on all the routers. Notice that network information in protocol status and routing table differ.

21.1. Enable RIP debugging on router rtrcity01: *debug ip rip*. Which RIP version is used now? What address is used to send RIP information, which addresses are used to receive it; how this differs from version 1?

21.2. Stop the debugging by running the command: *no debug all*.

21.3. Try pinging the PCs of the other cities.

Content of report

1. Objectives of the work.
2. The analysis of information received using CDP (Task 7).
3. The analysis of the running configurations of the routers when static routing is used and when RIP is used. What is a major difference (Tasks 13–14)?
4. The analysis of the routing table when static routes are used and when RIP is used (Tasks 10 and 18).
5. The analysis of RIP routing steps and the difference between the versions (Tasks 19–21).
6. General conclusions of the work.

Review questions and problems

1. What is the difference between broadcast and multicast?
2. How two gateways can be configured on the router?
3. What is transferred during RIP communication?
4. What is more convenient for network administrator: static routes or dynamic routing protocol, why?

Literature

Configuring Routing Information Protocol. http://www.cisco.com/en/ US/ docs/ios/iproute_rip/configuration/guide/irr_cfg_rip_ps6350_TSD_ Products_Configuration_Guide_Chapter.html.

Kurose, J. F.; Ross, K. W. 2009. *Computer Networking: A Top-Down Approach*. 5th edition. Addison Wesley. 864 p. ISBN-13: 978-0136079675.

Laboratory work 11

Computer Network Routing by Using Open Shortest Path First (OSPF) Dynamic Routing Protocol

Objectives

Design a computer network which uses OSPF protocol, research the principles of OSPF protocol.

Basic knowledge and theory

Open Shortest Path First (OSPF) is a link state interior gateway protocol. It uses network masks to fully define a network. OSPF detects changes in the network, such as link failures and converges on a new routing structure within seconds. It computes the shortest path tree for each route using Dijkstra's algorithm.

Router first finds its ID – the highest IP address of all router interfaces is used, or if loopback interface is enabled it is used instead of the physical interface. Then neighbour relationship between the routers is set.

OSPF then selects one Designated Router and one Backup Designated Router for a network, other network will have other DR and BDR. DR becomes a router with a highest priority and ID.

Then by neighbour communication DR gathers link state information from available routers and constructs a topology map of the whole network.

The routing table is constructed based on the link costs associated with each interface. Cost may be influenced by the distance, network throughput, link availability and reliability. Traffic load is balanced between the routes if the cost is equal.

Cost equals to 100 divided by the bandwidth in MBps. That was set in the time when Fast Ethernet was not implemented and higher bandwidths were hard to imagine. Metric cannot have a fraction, so for

the bandwidths equal or higher than Fast Ethernet metric equals to 1. To tune the cost to represent nowadays situation some default value must be changed.

OSPF maintains three tables: neighbour, topology and routing. All routers in an area have a same topology table.

An OSPF network may be divided into routing areas to simplify administration and optimize traffic and resource utilization.

Area 0 or 0.0.0.0 represents the backbone of the network. In this laboratory work single area network will be used. In complex networks different areas are used for different segments of the network.

OSPF is a transport layer protocol. It uses multicast addressing to distribute routing information. The OSPF protocol supports a variety of authentication methods to allow only trusted routers to participate in routing.

Work assignment and methodical guidelines

1. Variants: use the information provided in laboratory work 9.
2. Read attentively the theory section of laboratory work. Before starting the work discuss all obscure questions with the lecturer.
3. Remove all static routes or RIP configuration if there is left any of it. View the working configuration by running: *show running-config*, then from the configuration mode run: *no ip route (a.a.a.a) (b.b.b.b) (c.c.c.c)* for all the static routes, to disable RIP (if this peace of configuration is saved) run: *no router rip*. Check if the configuration is changed by reviewing startup-config. Check if all the interfaces are up, that can be done using: *show ip interfaces brief* command. Perform this operation on all the routers. Configure virtual PC pccity using instructions provided in lab 9.
4. Configure loopback interfaces on all the routers with a such logic: *X.(router no) .(router no) .(router no)*, with network mask

255.255.255.0; e.g.: for the router rtrcity01 and 15 variant it would be 15.1.1.1.

rtrcity01#configure terminal
rtrcity01 (config)#interface loopback 0
rtrcity01 (config-if)#ip address X.1.1.1 255.255.255.0
rtrcity01 (config-if)#no shut

Loopback is a virtual interface used for management purposes, not to send data.

5. Configure single area OSPF in the simulated network. Configure ospf with process ID 1: *router ospf 1*. Add all the networks (including loopbacks) providing the network masks to area 0: *network (a.a.a.a) (b.b.b.b) area 0*, eg.:

rtrcity02#configure terminal
rtrcity02(config)#router ospf 1
rtrcity02(config-router)#network 10.0.0.0 255.0.0.0 area 0
rtrcity02(config-router)#network 172.18.X.0 255.255.255.0 area 0

Configure all the remaining routers.

6. Check the connectivity between the routers by pinging their interfaces. Analyse routing tables of the routers: what is the administrative distance of the route added by OSPF protocol, what is represented by the cost value?
7. Analyse the parameters of OSPF: *show ip protocol* (what is router ID, what is reference bandwidth and administrative distances?), *show ip ospf* (analyse the timers, what they represent? Use online references). Check the neighbours of all the routers using: *show ip ospf neighbor*. What routers became designated routers and what backup designated routers? Why? What is the remaining type of the router?
8. Analyse the OSPF database using: *show ip ospf database*, check router link states using*: show ip ospf database router* (what net-

works types routers are connected to?). Run these commands on the other router: does the information differ? Why?

9. Check the routing tables on the routes using: *show ip route* command. Run a traceroute from PC to rtrcity01 Fe0/0 interface; is it reachable? Why? What is the default reference bandwidth used (*show ip protocol*)? Change the default reference bandwidth to 1 Gbps with: *auto-cost reference-bandwidth 10000* command in ospf configuration mode.
10. Analyse how the adjacencies are built, on rtrcity03 run: *debug ip ospf adj* command, then reset ospf process with*: clear ip ospf process* answer yes to the question asked. How the states are changing? Which routers are selected as DRs, why? Check how ospf communication between the routers commences by running on rtrcity03: *debug ip ospf packet*. What is the update interval? Disable debugging with: *no debug all*.
11. Check costs on the interfaces on router rtrcity03: *show ip ospf interface*. Let's change the cost on interface Ethernet 0/0 to 50. In order to do this you need to be in Ethernet 0/0 configuration mode and run the command: *ip ospf cost 50*. Check how the routing table changed*: show ip route* and run *traceroute* from PC to *rtrcity01* Fe0/0 interface.

Content of report

1. Objectives of the work.
2. How to config loopback, what is it used for (Task 4)?
3. What are the steps to configure OSPF routing protocol? (Tasks 5, 6, 7 and 8). How they differ from RIP configuration?
4. Analyse the important aspects of OSPF routing: router types, tables, values to control the protocol. How they can be configured? (Tasks 9–11).
5. General conclusions of the work.

Review questions and problems

1. How OSPF cost is calculated?
2. To what area every router must have communication to in OSPF?
3. How the routing information message is called in OSPF, what is being transmitted?

Literature

Configuring OSPF. http://www.cisco.com/en/US/docs/ios/12_0/np1/configuration/guide/1cospf.html.

Parkhurst, W. R. 2002. *Cisco OSPF Command and Configuration Handbook.* Cisco Press. 528 p. ISBN-13: 978-1587055409.

Laboratory work 12

Virtual Local Area Networks

Objectives

Getting acquainted with switching environment: VLANs and *Spanning Tree Protocol*.

Basic knowledge and theory

Switches

Switches improve network performance by stopping unneeded traffic from crossing the network segments; allowing multiple communication paths between the segments and not introducing performance degradation.

Switches perform at OSI Layer 2. They examine the packets traveling through it and build a forwarding (MAC) table based on what they hear. Every port has associated MAC table which is constructed over time from source MAC addresses from the packets which arrive to that port. When MAC tables are consistent switch connects only the two ports: source port and destination ports based on MAC address where the packet is being sent to. Switch forwards packet to all the ports if he is not aware of the destination MAC.

Switches can be OSI Layer 3 if they support IP addresses and routing functions.

Virtual Local Area Network

VLAN is a broadcast domain created by switches. Such broadcast domains usually are created by OSI layer 3 devices, e.g. routers. Switch ports are assigned to VLANs. All ports in the same VLAN are in the single broadcast domain. Default VLAN is marked by number 1. Putting ports to other VLANs is like dividing a physical switch to a few virtual ones and without additional configuration,

they would not be able to communicate with any other devices from the different VLAN.

VLANs are needed when there are more than 200 devices on LAN, there is a lot of broadcast traffic on the LAN, users need more security, is a need to separate voice and data traffic, or there is a need to divide a single switch into multiple virtual switches.

Each VLAN should be in its own subnet. The benefit is that devices in different physical locations can be on the same network.

Spanning Tree Protocol

STP (IEEE 802.1D) is used to establish redundancy and a loop-free path through the network. STP forces certain ports into a standby state and only one path to each network segment is active at any time. If there is a problem with connectivity, the other ports are activated and connectivity is restored. STP communication is done by transmitting Bridge Protocol Data Units (BPDU).

STP at first elects one root bridge. On the root bridge, all ports act as designated ports sending and receiving traffic. Then the protocol establishes one root port on each non-root bridge. Then STP establishes one designated port on the bridge that has the lowest path cost to the root bridge on each network segment.

STP port types: root port (switch port with the best path to the root bridge; forwards data traffic towards the root bridge), designated port (receives and forwards data frames towards the root bridge), nondesignated port (is blocking data frames), disabled port (switch port that is shut down).

Each Layer 2 port on a switch running STP can be in one of these states port states: blocking, listening (sending and receiving BPDU, no data traffic yet), forwarding, disabled (port does not participate in spanning tree).

To determine its root port (best port towards the root bridge), each switch uses a cost value: 10 Gbps – 1, 1 Gbps – 4, 100 Mbps – 19, 10 Mbps – 100.

Work assignment and methodical guidelines

1. Variants. Use the values from Table 9.1.
2. Read attentively the theory section of laboratory work. Before starting the work discuss all obscure questions with the lecturer.
3. Diagram of the network is presented in Figure 12.1.

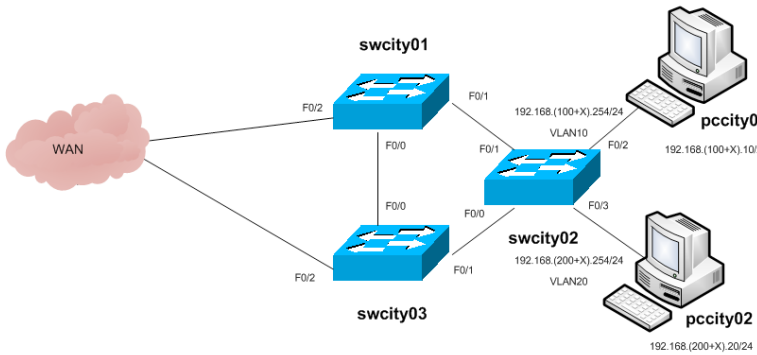


Fig. 12.1. Network diagram

4. Connect the simulated network as shown on Figure 12.2. Add switch modules (*NM-16ESW*) to swcity switching routers: right click on the object and select configure, then click on the node name and select NM-16ESW module on the Slots tab (Fig. 12.3.). You will be warned that you must use "manual mode" to connect a link with a NM-16ESW module, remember to use such type of a link when connecting the devices. Change hostnames, hostname of the PC object must be changed prior the connecting the object. If intelligent physical switches are used in the laboratory, BPDU guard on the lab switches must be disabled.

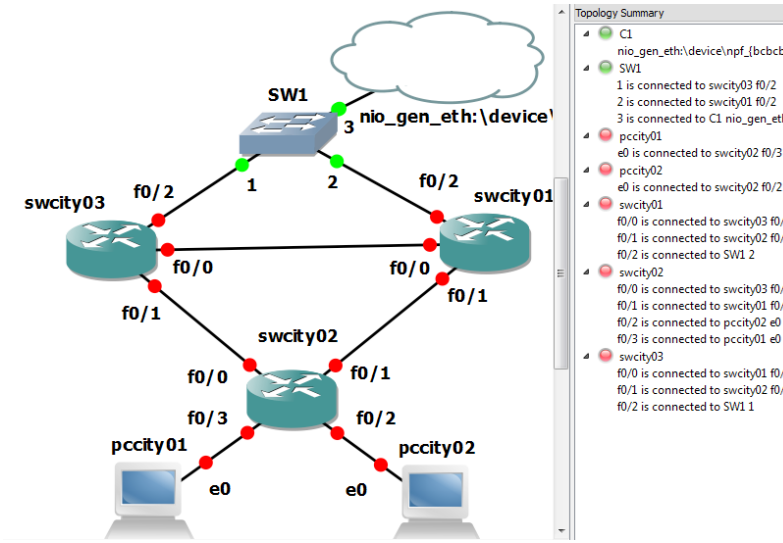


Fig. 12.2. GNS3 network layout for VLAN simulation

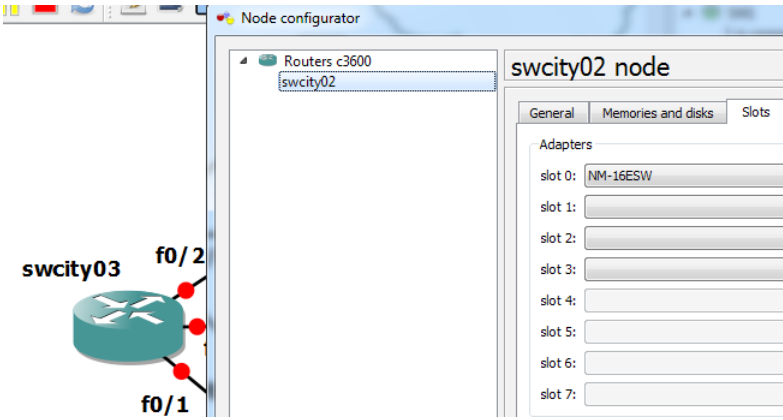


Fig. 12.3. Adding the switch module

5. Power on the devices and open their consoles. Start the routers first; choose their *idle-pc* values. CPU load near 100 % is not acceptable. Start the PC objects one by one. PC object can be

accessed using console too, login name to be used then is: **tc**. To enable root rights for all operations use the command: *sudo su*.

6. Configure *Qemu* object.
   6.1. Change hostname on PC console using: *hostname pccity*.
   6.2. Configure IP addressing on PC objects, 192.168.(100+X).10/24 on pccity01 and 192.168.(200+X).20/24 on pccity02, for this use a such command: *ifconfig eth0 192.168.A.B netmask 255.255.255.0*. IP addressing can be reviewed using: *ifconfig eth0*. Try pinging pccity02 from pccity01, no replies are coming back.
   6.3. Add routes from pccity01 to pccity02 and vice versa: *route add -net 192.168.(200+X).0 netmask 255.255.255.0 dev eth0* on pccity01 and: *route add -net 192.168.(100+X).0 netmask 255.255.255.0 dev eth0* on pccity02. Routing table can be reviewed by: *route –n*. Try pinging again, ping should work.
7. By default all switch interfaces operate on VLAN 1. Configure different VLANs on swcity02: VLAN 10 with name *left* for interface FastEthernet 0/3 and network 192.168.(100+X).0/24 and VLAN 20 with name *right* for interface FastEthernet 0/2 and network 192.168.(200+X).0/24. Ports modes for FastEthernet 0/2 and FastEthernet 0/3 should be access and for FastEthernet 0/0 and FastEthernet 0/1 – trunk:

```
swcity02#vlan database
swcity02 (vlan)#vlan 10 name left
VLAN 10 added:
   Name: left
swcity02 (vlan)#vlan 20 name right
VLAN 20 added:
   Name: right
swcity02 (vlan)#exit
APPLY completed.
Exiting....
```

swcity02#conf t
swcity02(config)#int f0/2
swcity02(config-if)#switchport mode access
swcity02(config-if)#switchport access vlan 20
swcity02(config-if)#exit
swcity02(config)#int f0/3
swcity02(config-if)#switchport mode access
swcity02(config-if)#switchport access vlan 10
swcity02(config-if)#exit
swcity02(config)#int f0/0
swcity02(config-if)#switchport mode trunk
swcity02(config-if)#exit
swcity02(config)#int f0/1
swcity02(config-if)#switchport mode trunk
swcity02(config-if)#exit

Try pinging pccity02 from pccity01, no replies are coming back. Check interface status and descriptions: show interface status and show interface description. Continue configuring the switch.

8. Configure IP information on switch2, what OSI layer device does this make?

swcity02(config)#int vlan 10
swcity02(config-if)#ip address 192.168.(1X).254 255.255.255.0
swcity02(config-if)#exit
swcity02(config)#int vlan 20
swcity02(config-if)#ip address 192.168.(2X).254 255.255.255.0
swcity02(config-if)#exit
swcity02(config)# interface Loopback0
swcity02(config-if)# ip address X.1.1.1 255.255.255.0

9. Configure remaining switches swcity01 and swcity03 as L2 (OSI model Layer 2) devices with trunk ports using: *switchport mode trunk* command in port configuration mode.

10. Add default route on the PC objects using: *route add default gw 192.168.X.254*. Ping the gateway and check routing table.
11. Configure RIP routing protocol to distribute VLAN 1 (default) information between routers with configured networks on *swcity01*. First configure RIP protocol itself, stating the networks which are connected to the router directly:

    swcity02(config)# router rip
    swcity02(config-router)# network a.a.a.a

    and then stating that it must use VLAN 1 for communicating RIP version 1 information:

    swcity02(config)# interface Vlan1
    swcity02(config-if)# ip rip send version 1
    swcity02(config-if)# ip rip receive version 1

12. Run *debug ip routing* command on swcity02 to see how this router gets informed of new routers from another cities. Check the routing table of the router: *show ip route*. Try pinging the neighbour cities PCs. Notice that PCs are pingable despite the VLAN. Communication between VLANs can be forbidden using access lists.
13. Check what router is primary root for the VLAN 1 by running: *show spanning-tree root*, which ports are used for communication: *show spanning-tree vlan 1*. Configure Spaning Tree Protocol to use swcity03 as default one:

    swcity03(config)#spanning-tree vlan 1 root primary.

    Check the status of ports on the router swcity01: *show spanning-tree vlan 1*.
14. Simulate the break of swciry03. Ping your neighbour city PC and disable all ports on swcity03 using *shutdown* command in the port configuration mode. Check how ports on swcity01 got enabled. How long the ping to your neighbour city PC was disrupted?

Content of report

1. Objectives of the work.
2. The steps and options to configure switching ports on Cisco router 3640 (Task 7).
3. What is the difference between layer 2 and layer 3 switch configuration (Tasks 8–12).
4. The analysis of STP, switch types, algorithm and redundancy (Tasks 13–14).
5. General conclusions of the work.

Review questions and problems

1. What is the number of the default VLAN?
2. What is a trunk port?
3. What type of ports STP has, what are they used for?

Literature

Froom, R., Sivasubramanian, B., Frahim, E. 2010. *Implementing Cisco IP Switched Networks (SWITCH) Foundation Learning Guide: Foundation Learning for SWITCH 642-813*. Cisco Press. 560 p. ISBN-13: 978-1587058844.

Kurose, J. F.; Ross, K. W. 2009. *Computer Networking: A Top-Down Approach*. 5th edition. Addison Wesley. 864 p. ISBN-13: 978-0136079675.

McQuerry, S., Jansen, D., Hucaby, D. 2009. *Cisco LAN Switching Configuration Handbook*. 2nd edition. Cisco Press. 360 p. ISBN-13: 978-1587056109.

Nottingham, H., Odom, S. 2002. *Cisco Switching Black Book*. Paraglyph Press. 656 p. ISBN-13: 978-1932111330.

Seifert, R., Edwards, J. 2008. *The All-New Switch Book: The Complete Guide to LAN Switching Technology*. Wiley. 816 p. ISBN-13: 978-0470287156.