R.A. Rodriges Zalipynis
Р.А. Родригес Залепинос

Computer Networks

Компьютерные сети

Course Syllabus
(Curriculum)
in English

Учебная программа
на английском языке

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии

Р.А. Родригес Залепинос

Computer Networks
Компьютерные сети

Учебная программа
на английском языке

Электронное издание
локального распространения

Санкт-Петербург
Наукоемкие технологии
2024

ISBN 978-5-907804-23-4

УДК 004.7
ББК 32.97
Р60

Рецензенты:

Миркин Борис Григорьевич, доктор технических наук, профессор департамента анализа данных и искусственного интеллекта, факультет компьютерных наук, НИУ ВШЭ;
Шилов Валерий Владимирович, кандидат технических наук, профессор департамента программной инженерии, факультет компьютерных наук, НИУ ВШЭ.

Р60 **Родригес Залепинос Р.А.** Computer Networks [Электронный ресурс] = Компьютерные сети: учебная программа на английском языке. – Электрон. текстовые дан. (0,9 Мбайт). – СПб.: Наукоемкие технологии, 2024. – 32 с. – 1 электрон. опт. диск (CD-ROM).

ISBN 978-5-907804-23-4

Компьютерные сети повсеместно распространены в современном ИТ-секторе. Компьютерные сети являются основой СУБД, веб- и мобильных приложений, распределенных систем и многих других программных продуктов. Хорошее понимание основополагающих принципов работы компьютерных сетей способствует созданию более надежных и эффективных программных продуктов. Следовательно, представленная учебная программа «Компьютерные сети» очень полезна для обучения будущих современных разработчиков программных продуктов.

Учебная программа «Компьютерные сети» предлагает получить теоретические знания, практические навыки и опыт программирования в области компьютерных сетей: каким образом функционируют компьютерные сети, работают протоколы компьютерных сетей, какое программное обеспечение необходимо для исследования компьютерных сетей, программные сетевые среды и библиотеки для разработки сетевых приложений.

Учебная программа предназначена для преподавателей ВУЗов, студентов программ бакалавриата (старшие курсы) и магистратуры (младшие курсы) в области компьютерных наук, а также лиц, заинтересованных в изучении компьютерных сетей.

Текстовое электронное издание

Минимальные системные требования:
- процессор: Intel x86, x64, AMD x86, x64 не менее 1 ГГц;
- оперативная память RAM ОЗУ: не менее 512 Мбайт;
- свободное место на жестком диске (HDD): не менее 120 Мбайт;
- операционная система: любая, при наличии Adobe Acrobat Reader;
- дисковод CD-ROM.

УДК 004.7
ББК 32.97

ISBN 978-5-907804-23-4

Учебное издание

Родригес Залепинос Рамон Антонио

Computer Networks
Компьютерные сети

Учебная программа
на английском языке

Электронное издание
локального распространения

Подписано к использованию 25.01.2024
Объем издания – 0,9 Мбайт.
Комплектация издания – 1 CD.
Тираж 100 CD.

Издательство «Наукоемкие технологии»
ООО «Корпорация «Интел Групп»
http://publishing.intelgr.com
E-mail: publishing@intelgr.com
Тел.: (812) 945-50-63

ISBN 978-5-907804-23-4

GOVERNMENT OF THE RUSSIAN FEDERATION
FEDERAL STATE AUTONOMOUS
EDUCATIONAL INSTITUTION OF HIGHER EDUCATION
NATIONAL RESEARCH UNIVERSITY
"HIGHER SCHOOL OF ECONOMICS"
(HSE UNIVERSITY)

Faculty of Computer Science
School of Software Engineering

Ramon Antonio Rodriges Zalipynis

Computer Networks

Course Syllabus (Curriculum)

Electronic Edition

Saint Petersburg
Naukoemkie Technologii
2024

ISBN 978-5-907804-23-4

Reviewers:

Boris G. Mirkin, Doctor of Sciences, Professor, School of Data Analysis and Artificial Intelligence, Faculty of Computer Science, HSE University.
Valery V. Shilov, PhD, Professor, School of Software Engineering, Faculty of Computer Science, HSE University.

R.A. Rodriges Zalipynis. Computer Networks [Electronic Resource] = Computer Networks: Course Syllabus (Curriculum). – Text electronic edition (0.9 MB). – St. Petersburg: Naukoemkie Technologii, 2024. – 32 P. – 1 electronic optical disk (CD-ROM).

ISBN 978-5-907804-23-4

Computer Networks are ubiquitous in the modern IT sector. Computer Networks are veins of DBMSs, Web & Mobile Applications, Distributed Systems, and many other software products. A good understanding of the underlying networking principles facilitates building more reliable and efficient software products. This makes this Computer Networks Course Syllabus (Curriculum) highly beneficial for educating future modern software engineers.

This Course Syllabus (Curriculum) proposes to acquire theoretical knowledge, practical skills, and programming experience in the area of Computer Networks: how Computer Networks function, how Computer Network protocols work, what software is needed to investigate Computer Networks, software networking frameworks and libraries to code networking applications.

University instructors, senior undergraduate and junior graduate computer science students, as well as anyone interested in studying Computer Networks are the target audience for this Course Syllabus (Curriculum).

Text electronic edition

Minimal system requirements:
- CPU: Intel x86, x64, AMD x86, x64 at least 1 GHz;
- RAM: at least 512 MB;
- Free disk space: at least 120 MB;
- Adobe Acrobat Reader or other PDF reader;
- CD-ROM.

ISBN 978-5-907804-23-4

Educational publication

Rodriges Zalipynis Ramon Antonio

Computer Networks

Course Syllabus (Curriculum)
in English

Electronic Edition

Signed up for use 25.01.2024
Publication volume – 0.9 MB.
Edition contents – 1 CD.
Edition circulation – 100 CD.

Publisher « Naukoemkie Technologii»
Intel Group Corporation, Ltd.
https://publishing.intelgr.com
E-mail: publishing@intelgr.com
Tel.: +7 (812) 945-50-63

ISBN 978-5-907804-23-4

TABLE OF CONTENTS

1. Course Description

1.1. ECTS

The Course Syllabus (Curriculum) is designed to meet the following ECTS parameters:

- Total ECTS credits: 5

With a total of 190 hours, of which
- Contact work: 64 hours (32 hours of lectures, 32 hours of seminars/practice)
- Self-study: 126 hours

One lecture per week is planned, which is accompanied by one seminar/practical lesson. The students are expected to devote about 8 hours per week to self-study.

The course has the following assessments: intermediate (one control work at the end of the 3rd quartile, two home assignments at both quartiles), and final (one exam at the end of the 4th quartile).

In general, this course has no mandatory pre-requisites if taught to Senior Bachelor Computer Science students or Junior Master Computer Science students. However, to successfully complete home assignments, students are required to have the following skills: Operating Systems, a Programming Language (Java, C/C++, not Python, as it is too high-level in terms of Computer Networks).

The course is given completely in English, so a good knowledge of English (at least listening and reading, intermediate speaking) is also required.

Experience in desktop, mobile, or web development before entering the course is not required but will provide more convenience during practical lessons and homework assignment selection.

1.2. Extended Abstract

Computer Networks are ubiquitous in the modern IT sector. Computer Networks are veins of DBMSs, Web & Mobile Applications, Distributed Systems, and many other software products. A good understanding of the underlying networking principles facilitates building more reliable and efficient software products. This makes the Computer Networks course highly beneficial for educating future modern software engineers.

This Course Syllabus (Curriculum) proposes to acquire theoretical knowledge, practical skills, and programming experience in the area of Computer Networks: how Computer Networks function, how Computer Network protocols work, what software is needed to investigate Computer Networks, software networking frameworks and libraries to code networking applications.

In this course, we explore the core principles of Computer Networks. We cover techniques for transmitting information efficiently and reliably over a variety of communication media. We look at the addressing and routing problems that arise and must be solved during the information transmission. We utilize different programming languages and popular technologies that make it possible to investigate the underlying principles of the networking protocols (for example, Java and the Netty framework). We explore the TCP/IP stack, including HTTP, FTP, and WebSocket.

Students are expected to meet weekly with the instructor (1 lecture and 1 seminar/practical lesson). A portion of the course is taught as hands-on training when lecture material is covered interactively with the course instructor. The course contains programming assignments as well as network exploration using the respective software tools.

This course includes 5 home assignments, 1 control work (in the middle of the course), and 1 final exam.

1.3. About this Course Syllabus (Curriculum)

This Syllabus (Curriculum) describes topics to be covered during the course and the knowledge and skills that a student will gain after successfully completing the course. It also regulates the assessment types and criteria for grading. The course embraces a wide range of modern technologies in the world of Computer Networks including widely used network protocols, tools, and software frameworks/libraries. A noticeable portion of the course is devoted to TCP/IP, which is one of the most popular networking protocol stacks nowadays, as well as gaining networking programming experience.

The Syllabus (Curriculum) is prepared for teachers responsible for the course (or closely related disciplines), teaching assistants, students enrolled within the course, as well as experts and statutory bodies carrying out assigned or regular accreditations in accordance with the Educational Standards of the National Research University "Higher School of Economics" (HSE University).

1.4. Courses that used this Syllabus (Curriculum)

The course "Computer Networks" was taught to students of the HSE University many times (2018–current). These courses used variants of this Syllabus (Curriculum) that has been continuously improved over the years. Information about each course appears on the official HSE University Portal. The following course meta-data are copied from the official HSE University Portal.

Computer Networks course in the year of study 2023/2024:

- Course name: Computer Networks
- Year of study: 2023/2024
- Type: Elective course (Software Engineering)
- Area of studies: Software Engineering
- Delivered by: School of Software Engineering
- Where: Faculty of Computer Science
- Degree: Bachelor
- When: 3 year, 3, 4 module (1 semester)
- Mode of studies: offline
- Open to: students of all HSE University campuses
- Instructors: Ramon Antonio Rodriges Zalipynis
- Language: English
- ECTS credits: 5
- Contact hours: 60

Link (official HSE University Portal):
https://www.hse.ru/en/edu/courses/836711350

Computer Networks course in the year of study 2022/2023:

- Course name: Computer Networks
- Year of study: 2022/2023
- Type: Elective course (Software Engineering)
- Area of studies: Software Engineering
- Delivered by: School of Software Engineering
- Where: Faculty of Computer Science
- Degree: Bachelor
- When: 3 year, 3, 4 module (1 semester)
- Mode of studies: offline
- Open to: students of all HSE University campuses
- Instructors: Ramon Antonio Rodriges Zalipynis
- Language: English

- ECTS credits: 5
- Contact hours: 60

Link (official HSE University Portal):
https://www.hse.ru/en/edu/courses/646484470

Computer Networks course in the year of study 2021/2022:

According to the students' votes, the course was selected as the best in 3 categories at once:

⋆ Category 'Best Course for Career Development'

⋆ Category 'Best Course for Broadening Horizons and Diversity of Knowledge and Skills'

⋆ Category 'Best Course for New Knowledge and Skills'

- Course name: Computer Networks
- Year of study: 2020/2021
- Type: Elective course (Software Engineering)
- Area of studies: Software Engineering
- Delivered by: School of Software Engineering
- Where: Faculty of Computer Science
- Degree: Bachelor
- When: 3 year, 3, 4 module (1 semester)
- Mode of studies: offline
- Open to: students of all HSE University campuses
- Instructors: Ramon Antonio Rodriges Zalipynis
- Language: English
- ECTS credits: 5
- Contact hours: 60

Link (official HSE University Portal):
https://www.hse.ru/en/edu/courses/339493635

Computer Networks course in the year of study 2020/2021:

According to the students' votes, the course was selected as the best in 3 categories at once:

⋆ Category 'Best Course for Career Development'

⋆ Category 'Best Course for Broadening Horizons and Diversity of Knowledge and Skills'

⋆ Category 'Best Course for New Knowledge and Skills'

- Course name: Computer Networks
- Year of study: 2020/2021
- Type: Elective course (Software Engineering)
- Area of studies: Software Engineering
- Delivered by: School of Software Engineering
- Where: Faculty of Computer Science
- Degree: Bachelor
- When: 3 year, 3, 4 module (1 semester)
- Mode of studies: offline
- Instructors: Ramon Antonio Rodriges Zalipynis
- Language: English
- ECTS credits: 5
- Contact hours: 60

Link (official HSE University Portal):
https://www.hse.ru/en/edu/courses/375278638

Computer Networks course in the year of study 2018/2019:

- Course name: Computer Networks
- Year of study: 2018/2019
- Type: Elective course (Software Engineering)
- Area of studies: Software Engineering
- Delivered by: School of Software Engineering
- Where: Faculty of Computer Science
- Degree: Bachelor
- When: 3 year, 3, 4 module (1 semester)
- Mode of studies: offline
- Instructors: Ramon Antonio Rodriges Zalipynis
- Language: English
- ECTS credits: 5
- Contact hours: 64

Link (official HSE University Portal):
https://www.hse.ru/en/edu/courses/219873311

2. Learning Objectives

Upon the completion of this course, a successful student should be able to understand basic networking principles, know the functioning of popular network protocols, and be able to programmatically develop a networking application that uses popular network protocols.

3. Learning Outcomes

After completing the “Computer Networks” course, a successful student will:

- Have a strong understanding of networking concepts and the functioning of computer networks;
- Be able to programmatically work with popular computer network protocols;
- Acquire skills and experience in modern technologies and tools related to computer networks.

A successful student should:

Know:

- Core concepts of computer networks;
- Message formats of the most common network protocols;
- Popular software tools that are able to monitor computer networks;
- Most widely used networking services;
- Modern networking frameworks and/or libraries.

Be able to:

- Interpret messages of the most common network protocols that were captured by popular software tools for network monitoring;
- Programmatically work with popular network protocols using modern networking frameworks/libraries;
- Design custom network protocols on top of existing network protocols;
- Build basic networks in a virtual network simulator.

Acquire experience in:

- Programmatic work with diverse computer network protocols;
- Use software libraries and tools to programmatically/manually manage computer networks.

4. Course Plan

4.1. Course Outline

Students should devote self-study hours to learning material for self-study (about 15–35% of the material from each topic), doing their homework, control work, and exam preparation.

№	Topic name		Course hours, total	Contact hours		Self-study
				Lectures	Practical lessons	
Module 3						
1	Introduction to Computer Networks		4	2	2	0
2	TCP/IP Stack	TCP/IP stack: overview, addressing mechanisms, WireShark tool	8	4	4	15
3		DNS (Domain Name Service)	4	2	2	10
4		IP (Internet Protocol)	8	4	4	15
5		TCP (Transmission Control Protocol)	8	4	4	20
Subtotal:			32	16	16	60
Module 4						
6	Routing Protocols		4	2	2	8
7	Netty Framework		8	4	4	18
8	Designing and Managing Computer Networks in Microsoft Azure Cloud		4	2	2	12
9	Networking Services		4	2	2	10
10	Computer Networks Security		8	4	4	15
11	Recent Trends in Computer Networks		4	2	2	5
Subtotal:			32	16	16	66
Total:			64	32	32	126

Lectures are given in the form of PowerPoint presentations. Lecture slides are usually made available within the Learning Management System (LMS) or at another course resource (the address is typically announced at the beginning of the course). The lecture slides contain more links to additional Web resources (readings, documentation, and tools). Students who experience difficulties watching the presentation from a laser projector may download the slides onto their own laptops or a class PC to be able to comfortably follow the lecture.

Information about the data required for practical lessons, software tools (links to web sites), and libraries (Maven dependencies to be added to the project's pom.xml) can also be published in the course resources. Students are encouraged to install software tools, resolve Maven dependencies for libraries, and download all necessary data before attending classes to save time. The lecturer reserves the right to notify students via e-mail about the material necessary for the next classes or changes in the schedule and/or material.

The source code in Java and other languages that is demonstrated during practical lessons is also available at the course resource for each lecture/seminar/practice.

4.2. Topic-Wise Course Content

Topic 1. Introduction to Computer Networks.

A large fraction of modern software is based on computer networks, the history of computer networks, nowadays networking, 100 GB/s network speed and beyond, literature.

RFC (Request For Comments), MAC addresses, and the structure of MAC address. Address assignment regulations, IEEE Registration Authority. Computer Network Simulators.

Topic 2. TCP/IP stack: overview, addressing mechanisms, WireShark tool.

The stack structure, the correspondence of the TCP/IP stack layers to the OSI Model layers. The review of the TCP/IP stack protocols (details on these protocols are given in the respective parts of the course): IP, ICMP, RIP, OSPF, TCP, UDP, FTP, Telnet, HTTP, SMTP, SNMP, TFTP. Data entity classification in the TCP/IP protocol stack: stream, datagram, segment, packet, frame.

Address types of the TCP/IP Protocol Stack: local (hardware) addresses, network IP addresses. An IP address is the address of a network interface, not a node (network card, router interfaces, etc.). Domain names, basic introduction to DNS.

IP Address Formats. Network address and host address. IP address masks. Classes of IP addresses. Destination address types: unidentified, unicast, multicast, limited broadcast, directed broadcast, broadcast, anycast (IPv6), loopback (127.*.*.*).

Centralized assignment of IP addresses: ICANN (Internet Corporation for Assigned Names and Numbers), the deficit of IPv4 addresses. CIDR (Classless Inter-Domain Routing).

ARP (Address Resolution Protocol), RFC 1122. ARP messages (requests, responses, gratuitous messages, targeted ARP requests), ARP message formats. ARP server, ARP tables, static and dynamic ARP records, time-to-live of dynamic records, and ARP cache. RARP (Reverse ARP). Proxy-ARP.

Practical lesson. Investigation of ARP activity over the network.

- Tools:
  - `arp` (command line, both Windows and Linux)
  - `ipconfig/ifconfig` (command line, Windows/Linux)
  - WireShark (GUI, both Windows and Linux, many other platforms are also supported)
- "Promiscuous mode" (the WireShark Wiki): https://wiki.wireshark.org/CaptureSetup/Ethernet

Topic 3. DNS (Domain Name Service).

Hierarchical name space, domains, and subdomains. Short, relative, and complete domain names. ICANN (Internet Corporation for Assigned Names and Numbers). ISO 3166, country domains. File hosts.txt. DNS protocol and DNS servers. DNS zones, authoritative and non-authoritative responses. DNS poisoning.

Types of DNS records: Address Mapping records (A), IP Version 6 Address records (AAAA), Canonical Name records (CNAME), Host Information records (HINFO), Integrated Services Digital Network records (ISDN), Mail exchanger record (MX), Name Server records (NS), Reverse-lookup Pointer records (PTR), Start of Authority records (SOA), Text records (TXT).

DNS name resolution: recursive and non-recursive procedures. Root DNS servers. The use of the anycast technique. Reverse DNS zones.

Topic 4. IP (Internet Protocol).

The purpose of IP. IP packet. The header of an IP packet. TTL (Time To Live). Checksum. IP routing scheme. Routing tables. Route types. Algorithms for traversing routing tables. IP routing with and without masks.

IP packet fragmentation. Fragmentation parameters. Fragmentation mechanism. ICMP (Internet Control Message Protocol). The format of ICMP messages. `traceroute`/`tracert` utility. `ping` utility.

IPv6. IPv6 addressing system. The structure of global aggregating unique IPv6 addresses. Decreasing the load on routers. Switching to IPv6.

Topic 5. TCP (Transmission Control Protocol).

Multiplexing and demultiplexing. TCP ports. TCP sockets. UDP (User Datagram Protocol). Stateful and stateless protocols. UDP datagrams. Dataflow scheme in TCP and UDP. TCP segments. ACK notion. The dataflow scheme of TCP segments. The format of the TCP header. TCP logical connections. TCP connection scheme. Automatic repeat request methods: source idle, sliding window, N-segments back. TCP buffer structure. Accumulated ACK principle. Dataflow control in TCP.

Topic 6. Routing Protocols.

Common properties and classifications of routing protocols. Source routing. Static and adaptive routing. Distance Vector Algorithms (DVA), and Link-State Advertisement (LSA).

RIP (Routing Information Protocol). Building a routing table. Router adaptation to network state dynamics. False routes.

OSPF (Open Shortest Path First). Two stages of building the routing table. Metrics. Routing in heterogenous networks. Using several routing protocols on the same network.

IGMP (Internet Group Management Protocol). The structure of an IGMP Message. Routing principles.

Topic 7. Netty Framework.

Challenges in creating a multi-user server that accepts and serves thousands of concurrent queries per second, maintains hundreds of thousands of connections. Networking peculiarities.

Architecture of Netty. Java InetAddress, blocking and non-blocking sockets, java.nio. Bootstrap or ServerBootstrap, EventLoop, EventLoopGroup, ChannelPipeline, Channel, Future or ChannelFuture, ChannelInitializer, ChannelHandler. Netty Framework threading model. Building a custom multiuser server. Creating TCP and HTTP servers. Customizing business logic. Designing custom network protocols on top of TCP and/or HTTP.

Developing a client-server application using the Netty Framework.

Topic 8. Designing and Managing Computer Networks in Microsoft Azure Cloud.

The description and peculiarities of the MS Azure Cloud. Java Azure API. Web GUI Azure console. Virtual machines, network interfaces, network security groups, network routing rules, and load balancers. API keys and access setup for Azure Cloud. Building a computer cluster in the MS Azure Cloud: a set of virtual machines connected to a single network. Choosing computer network size and selecting address masks. Creating and assigning IP addresses. Using a single IP address and different TCP ports to access different virtual machines. Monitoring network activity in the MS Azure Cloud.

Topic 9. Networking Services.

Web servers. HTTP (HyperText Transfer Protocol). URL (Uniform Resource Locator). Web client and web server. The format of HTTP messages and headers. Investigating HTTP using the Netty framework. WebSocket protocol. Binary and text modes. Creating a WebSocket client and a WebSocket server.

Mail servers. Electronic messages. Dedicated mail servers. Overview of POP3 (Post Office Protocol Version 3), SMTP (Simple Mail Transfer Protocol), and IMAP (Internet Message Access Protocol) mail protocols.

File servers. Overview of FTP (File Transmission Protocol) and Amazon S3 (Amazon Simple Storage Service). Caching, replication, and fault-tolerance. File service in the Cloud. Problems and solutions for the transmission of large files.

Topic 10. Computer Networks Security.

Traffic filtering and filtering rules. Firewalls. Software and hardware firewalls. Proxy-servers. NAT (Network Address Translation), NAPT (Network Address Port Translation), DMZ (Demilitarized Zone), and IDS (Intrusion Detection Systems). Overview of IPsec and VPN (Virtual Private Networks). Overview of SSL (Secure Sockets Layer).

Types and classifications of computer network threats and attacks. TCP attacks, ICMP attacks, UDP-attacks, and IP attacks. DoS (Denial of Service), DDoS (Distributed DoS), Man in the Middle, DNS attacks (DNS Poisoning, DNS Spoofing, Attacks on DNS Root Servers). Malware: trojans, worms, viruses, botnets. Methods of computer network protection.

Topic 11. Recent Trends in Computer Networks.

Large Modern Computer Networks. Grid systems. Large computer clusters. In-memory systems and networking issues. Networking hardware becomes smarter: solutions from Intel, Cisco, Mellanox and other companies for increasing the performance and value of networking hardware.

4.3. Topics for Control Work and Final Assessment

Any question targeted to check the understanding of any topic listed in the “Topic-Wise Course Content” section of this Syllabus (Curriculum) may be asked during the control work or the final exam. To answer the questions successfully, a student will need programming experience obtained during practical lessons in addition to lecture material.

Before the control work or exam, up to 20 minutes at a lecture or practical lesson are devoted to “Test Preparation”. This preparation is held in the form of joint (all students involved) answering (orally) sample questions similar to control work/exam questions.

Test preparation, control work, and exam formats are the same as standard Software Development certification programs, English tests, or similar certifications. It is highly recommended for a student to revise material using lecture slides and source code from practical lessons before they come to control work/exam.

It is also recommended to download all the lecture slides again before revision (from the course resource), even if a student has downloaded all of them previously (e.g., each time classes start). The lecturer may provide video recordings of the course, so it is also highly recommended to revise those before the exam or the control work.

4.4. Topics for Homework Assignments

4.4.1. Homework Assignment 1

This assignment consists of a series of micro-homework. It has several types of assignments. Students will receive a task on which they will start working in class and finish the task later, asynchronously. The tasks and requirements for them may be partially formulated in oral form by the instructor.

One type of assignment is targeted at the investigation of the protocol structure using popular networking tools. For example, students may use WireShark to capture network traffic and study the format of protocol headers.

The other type of assignment is the design of a virtual network using a network simulator (if it is freely available to the students and the lecturer). A student will construct a computer network consisting of routers, nodes, servers, and other components. This will enable students to have a comprehensive understanding of networking principles and designs.

4.4.2. Homework Assignment 2

In addition to the topics listed, at the beginning of some lectures, up to 5 minutes are devoted to “Networking Surveys”. They are held in the form of PowerPoint presentations by the course instructor. Their goal is to present the students with a wealth of diverse and interesting networking problems and tasks that could be solved using networking technologies.

The second homework is devoted to developing a software network application using a network framework, e.g., the Netty Framework. A student should develop an HTTP/TCP server, a custom protocol, or something related to a topic of our course.

Before a student starts doing their homework, they must arrive at a task to be solved with the application they aim to develop. Networking technologies, software, tools, and frameworks must be central to their work. However, since the area of computer networks is new to students, not all of them may be aware of the interesting challenges. The surveys are held to give examples of both problems and technologies.

Homework topics are not limited to the tasks to be described in the surveys. Students may also seek networking technologies and interesting tasks to solve for their homework assignments on their own.

Students are encouraged to form teams, choose to work on a startup, or just do interesting projects or research projects related to Computer Networks. Please see “Alternative Ways of Assessment”.

The topic for the home assignment may be proposed by the course instructor, a student (should be approved by the instructor) or worked out jointly with the instructor and a student (or a group of students in the case of teamwork).

In any case, before working on the selected topic, students should have their decision approved by the course instructor. The instructor will give a deadline for topic approval. The failure to approve the topic before the deadline results in no grade for the home assignment. The instructor may modify the task and/or data governed by some considerations, including if they feel it is unrealistic or too easy to implement. The final decision on which data and tasks are appropriate for the homework is left to the sole discretion of the instructor.

In this course, students are highly encouraged to develop a complete and quality-assured network application that can be useful to others, published on the Web, and/or demonstrated at programming competitions and/or personal portfolios.

4. Bibliography

This section contains the required and optional literature for this course.

4.2. Required

- Victor Olifer and Natalia Olifer, Computer Networks: Principles, Technologies and Protocols, Anniversary Edition, Piter, 1008 P., 2020.
- Victor Olifer and Natalia Olifer, Computer Networks: Principles, Technologies and Protocols, 5th edition, Piter, 992 P., 2017.
- James Kurose, Keith Ross, Computer Networking: A Top-Down Approach, Pearson, 7th edition, 864 P., 2016.
- Andrew S. Tanenbaum, David J. Wetherall, Computer Networks, 5th edition, Prentice Hall, 960 P., 2012.
- Christian Benvenuti, Understanding Linux Network Internals, O'Reilly Media, 1064 P., 2006.
- N. Maurer, Netty in Action, http://topconsulting.ru/wp-content/uploads/2011/03/Netty-In-Action-V5.pdf (this book is freely available on-line)

4.2. Optional

- Rodriges Zalipynis R. A. Towards Machine Learning in Distributed Array DBMS: Networking Considerations, in: *Machine Learning for Networking: Third International Conference, MLN 2020, Paris, France, November 24–26, 2020, Revised Selected Papers*. Springer, 2021. P. 284-304.
- Varghese G and Xu J, Network Algorithmics: an interdisciplinary approach to designing fast networked devices. Morgan Kaufmann, 2022.
- Lee RB, Security basics for computer architects. Springer Nature, 2022.
- Steven Noble, Building Modern Networks: Create and manage cutting-edge networks and services, Packt Publishing, 324 P., 2017.
- Jose Manuel Ortega, Mastering Python for Networking and Security: Leverage Python scripts and libraries to overcome networking and security issues, Packt Publishing, 426 P., 2018.
- James Bernstein, Networking Made Easy: Get Yourself Connected (Computers Made Easy), Independently published, 141 P., 2018.
- Andrew Crouthamel, Mastering Wireshark 2: Develop skills for network analysis and address a wide range of information security threats, Packt Publishing, 326 P., 2018.
- Netty short guide, http://docs.jboss.org/netty/3.2/guide/pdf/netty.pdf

5. Grading System

The 10-point scale is basic for all types of assessments. The contribution of each activity to the final grade is given in the table below.

Activity Contribution Table

Code	Description	Contribution, %
CW	Control work	15
HA1	Home Assignment 1	35
HA2	Home Assignment 2	30
EX	Exam	20
TT	Total contribution	

The final grade is calculated according to TT:

Final Grade Calculation Table

Total contribution, %	Final grade
<= 10	0
<=15	1
20	2
30	3
40	4
50	5
60	6
70	7
80	8
90	9
over 95	10

The HA1 percent of accomplishment is calculated as HA1PA = (*T* / *N*), where *N* is the total number of micro-assignments for HA1, and *T* is the sum of all assignments' grades that were evaluated higher than 5 by the instructor (10-point scale). HA1 = HA1PA * 3.5. The instructor gives 0 points for a micro-assignment that was not submitted before the deadline.

Please note that there are some homework assignments that are required to be defended during a practical lesson within a deadline defined by the instructor (please see the respective section of this syllabus for more details).

The Final Grade conversion rules from the 10-point grade to the 5-point grade are given in the following table below:

Conversion between the 10-point and the 5-point grading systems

10-points scale	5-points scale
1 – unsatisfactory 2 – very poor 3 – poor	unsatisfactory – 2
4 – satisfactory 5 – quite satisfactory	satisfactory – 3
6 – good 7 – very good	good – 4
8 – almost excellent 9 – excellent 10 – brilliant	excellent – 5

The Final Grade (Mark) is rounded upwards. Other grades are being kept precise throughout the whole course.

6. Guidelines for Knowledge Assessment

6.1. Assessment Types and Forms

This course has two types of assessment (intermediate and final) with four forms of assessment (one control work, two home assignments, and one final exam). All types and forms of assessment impact the final grade. The first and only control work is held at the end of the first course module (quartile). The home assignments assume that a student will complete the given task based on the knowledge and experience gained during this course. The first home assignment is for the first quartile, while the second home assignment is planned for the second module (quartile) and results in a software networking application (3rd and 4th quartiles if the course is given during the second semester, quartiles 3–4). The final exam is planned for the end of the last module (quartile). The table below summarizes assessment types and forms.

Assessment Types and Forms in Quartiles (Modules)

Assessment type	Assessment form	Quartiles (Modules)		Notes
		3	4	
Intermediate	Control Work (Test)	*		TCP/IP Stack Protocols
	Home Assignment 1 (some micro assignments require defense)	*		Understanding network protocols, principles, and functioning
	Home Assignment 2 (software development and defense)		*	Develop an application that uses software networking technologies/frameworks
Final	Exam (Test)		*	All course material

All results are evaluated according to the 10-point scale from 0 (failure) to 10 (excellent), inclusive.

6.2. Control Work and Exam Forms

The control work and exam are in the form of written tests (1 hour, 20 minutes each). They are designed using a state-of-the-art understanding of what a proper test should look like. The tests that are provided are somewhat similar in structure and types of problems to CCNA (Cisco Certified Network Associate), Oracle Java Certified Programmer, and other state-of-the-art exam/certification programs.

The control work and exam are held in written form. Printed or electronic forms with questions and answer versions are given to students. Questions and answers are in English. Students are not required to write any answers; the correct answer(s) should be circled with a pen or marked in electronic form. Students are also provided with specialized draft sheets, so a student should have a pen with them.

Cheating: use of mobile phones, the Internet, books, and notebooks is prohibited during the control work/exam. Communication with other students is prohibited. However, control works were sometimes held in the “open book” mode, which made little or no impact on the grade compared to the mode described above.

The evaluation criteria for questions are as follows. Correct answer to a question gives 1.0 points. Answers to questions with multiple choices are evaluated as follows:

- All correct items were checked without any incorrect ones — 1.0 (full score)
- At least one correct item was checked without any incorrect ones — 0.5 (half of a score)
- At least one incorrect item was checked — 0.0 (no points)

No total points (0 score) give 0% contribution to the final grade; all correct answers to questions give 100% contribution to the grade (see the corresponding section of this Syllabus for details).

6.3. Homework Assignment Forms

6.3.1. Homework Assignment 1

This kind of homework ensures regular practical experience for students at home. After the material for a topic is covered by the lecture, students receive a series of small tasks for the homework (micro assignments). Students should submit a micro-assignment in 1 or 2 weeks after it was assigned (exact deadline is set by the instructor for each assignment separately).

A delayed submission of the home assignment may be penalized by subtracting scores from the grade for the given assignment. A student may earn 0 – 10 points for their homework.

In general, the criteria for homework assignments are the understanding of network protocols, concepts, and network functioning. The teacher uses traditional techniques to check this: they will ask questions during homework defense to ensure understanding of the material by the student, source code written and source code authenticity (if any). Questions may also be based on the lecture material. The teacher grades the work according to the percent of questions answered, the amount of work done, the volume of work done, source code accuracy, the application as a whole (if any), the correctness of the application (if any), and other sound criteria that are applicable to this kind of work.

These homework assignments (not homework assignments № 2) should be done individually by each student. Cheating, including code borrowing or its other forms (plagiarism), will be penalized down to earning 0 points for the assignment.

6.3.2. Homework Assignment 2

The homework defense can be held in the form of a presentation of the developed application to the groupmates. Each student must submit a PowerPoint presentation, complete source code, and architecture description (Microsoft Word document) of their network application for review. Once all critical comments are fixed, a student submits their work for group review in a dedicated forum topic. Other group members should ask questions clarifying the application goals, its internal structure, and usage. The activity of asking questions can also contribute to the homework grade. The questions should be constructive; the answers to them may not be evident. The examined student must provide answers to all questions related to the home assignment. Based on the gathered questions, the student will have to refine their presentation and architecture description. After discussion and refinement are complete, the student submits the final version of their application. The developed application should be covered with unit tests, e.g., JUnit or another framework for quality assurance. A test-driven development (TDD) approach can be applied during the development of the application. The quality of tests and coverage percent can influence the homework grade. The grade for the application is given in accordance with the state-of-the-art evaluation criteria common to all courses that require application development during home assignments or other similar assessment activities. The grade for a homework assignment will also depend on the variety of network technologies utilized in the application. Tentative topics for home assignments are given in Section “Topics for Homework Assignments” of this Syllabus (Curriculum).

6.4. Alternative Ways of Assessment

In the course, a student may wish to substitute the Home Assignment and Exam (and possibly some practical lessons) with other types of activities.

A student (or a team of students) may propose an application project to work on, for example http://www.wikience.org/ru/темы_проектов/исследуй-климат-сам/ This will result in a sophisticated programming experience (teamwork, the development of a complete software product, etc.). The resulting project may be presented at corresponding competitions (e.g., IBM Smarter Planet, Microsoft Climate Initiative, Google Earth Engine Awards) or commercialized.

The other way is to tackle a research problem agreed upon with the course instructor in advance. This includes reading research papers, designing new methods and/or algorithms, and publishing journal or conference research papers.

6.5. Exam Retake Commission

The format of the exam retake is the same as during an ordinary exam format for this course; please refer to Section 6.2 for details. A student will not be asked any additional questions in oral, written, or any other form. The cumulative grade for a student (all items from Section 5 except "EX") may be considered by the exam retake commission. The commission may use the formula for calculating the grade, which can be found in Section 6. Item "EX" means the grade that was earned by the student during the exam retake (note that the exam retake grade may contribute to the final course grade with a weight of 0.2).

7. Methods of Instruction

There are several types of instruction: lectures, seminars, practice, hands-on training, and homework defense. Often, students are expected to try out things covered on the slides together with the instructor during a practice. Typically, 4 academic hours in class are planned for 1 lecture and 1 seminar/practical lesson.

The theoretical material within the slides is largely interleaved with practice. Lectures have the form of PowerPoint presentations. Practical lessons are provided as training: the instructor defines tasks to be accomplished using a programming language, networking monitoring tools, or other technologies (according to the current topic), or students are expected to follow the instructor during the practice.

The instructor helps students overcome difficulties that arise during the implementation of the given tasks by asking respective questions and pointing to corresponding Web resources and documentation. They also suggest improvements to the code or other results from the assignment.

8. Special Equipment and Software Support

All learning resources are freely available via the course resources. The respective links are given to students. Course instructors do not issue directly to students any software, books, documentation, or other material, even if they are freely available on the Web.

8.1. Software Support

Below is the main list of software that will be used during the course:

№	Name	Access type
1	Java Development Kit (JDK) at least 21 version	Free software
2	WireShark	Free software
3	Windows or Linux	Free software
4	arp, ping, tracert	Free software
5	IntelliJ IDEA or other IDE	Free software
6	Netty framework	Free software

8.2. Remote Support

The course portal is established with all lecture slides, source code, links to resources, and data that are available to students, e.g., the LMS (Learning Management System), which facilitates sending notifications to students, submitting homework, grading, and other activities. Students may also use the instructor's e-mail to contact them directly. It is also possible to arrange a Skype/Zoom meeting with the instructor (please contact them via e-mail to schedule a meeting).

8.3. Technical Resources

Students can prefer to take their personal laptops for each seminar/practical lesson. This should accelerate their learning and reduce stress since they will be working in a familiar and highly personalized environment.

The lecturer uses a laptop and laser projector for presentations and seminars/practical lessons.

Students use Java IDEs like IntelliJ IDEA or Eclipse. In addition, they can use various tools and technological tools for network monitoring, configuration, design, and other necessary activities. The specific software packages and tools that are required to accomplish the given practical task are listed before the respective practical lessons.

Some seminars/practical lessons require the Cloud or virtual machines. The instructor creates images for such virtual machines, deploys and runs them in the Cloud, e.g., Microsoft Azure, and provides access to the running virtual machines for free to students during seminars/practical lessons.

HOME ASSIGNMENT EXAMPLE

(one of the course assessment elements)

Task «Computer Network Characteristics»

Motivation.

If your software is going to be deployed on a new computer network and the performance (bandwidth, throughput, etc.) of this network is important to your application, then the first step to take is to measure the characteristics of this new communication medium.

Task Content.

Implement a client-server application in Java (both server and client must be implemented in pure Java without any additional network frameworks/libraries). Use the Socket class on the client side and the ServerSocket class on the server side.

The client has the following command line options: <IP> <port> <N> <M>. The client must connect to the server using the IP address <IP> and port <port>. The client must continuously send to the server an arbitrary array of length N*K bytes during a step number K=1, …, M. The client must wait for a response from the server before sending the next array. Thus, at step number 1, the client sends N bytes to the server; at minute 2, it sends 2*N bytes, ...; at minute M, it sends N*M bytes. Attention: the array must be sent with one call to the write method on the client socket; no need to use BufferedOutputStream or similar buffers. Alternatively, the client and the server can perform P iterations of sending and receiving messages (for example, the client makes a request 100 times and waits for a response).

The server in response should send the string YYYY.MM.DD HH:MM:SS (current date and time; you can add a time zone; you can use the ZonedDateTime class and form the response manually or DateTimeFormatter).

The client does not print responses from the server to the console. The client measures for each request the time that has elapsed from calling the write method to receiving a response from the server (System.currentTimeMillis). For each value of K, the client calculates an average time. At the end of the work, the client outputs a table (K, average time) to a file or to the console. Based on these data, it is necessary to build a graph in Excel (horizontal axis: K, vertical: average time). Please interpret the resulting graph.

The number of bytes sent from the client to the server can grow from 8 bytes to 16KB in 32-byte intervals. This is not the only possible configuration in which you can trace the patterns of interest to us; conduct experiments with different input parameters.

When building a graph, pay attention: "Legend items (series)" is the time, the vertical axis: you need to select a column with the measured time, and "Horizontal axis labels (categories)" is the number of bytes transferred, the horizontal axis.

Notes.

Be sure to conduct the server and client packet capture in WireShark (it's sufficient to find a few packets and make sure that they are generated by your client and server).

Please use on the client side:

```
Socket s = new Socket("127.0.0.1", server_port); // IP should be substituted
socket.setTcpNoDelay(true); // try both with true and false
s.getOutputStream();
```

On the server side:

```
ServerSocket ss = new ServerSocket(server_port);
Socket s = ss.accept();
s.getInputStream();
```

Deploying your Client and Server.

1. During development and basic testing, the server and client must be located on your working machine. In this case, the time is best measured in nanoseconds; in subsequent cases, you may need to measure the time in milliseconds.
2. During the experiment, the server must be located in the Cloud (access to the Cloud during the lesson will be provided; for the rest of the time, it is recommended to use free student credits), and the client must be located on your working machine.
3. If you wish, you can repeat the experiment when both the server and the client are located in the Cloud (you can select different geographical zones for the server and client machines).

Cloud Access Parameters: see the separate file (the link is provided by the instructor). If you are using Windows, you can use the WinSCP and Putty tools to access the virtual machines.

CONTROL WORK (EXAM) EXAMPLE

(one of the course assessment elements)

Surname, name:

Group: Date: :

A:

What is the number of layers in the OSI model?

1. 4
2. 5
3. 6
4. 7
5. 8
6. 9

B:

RFC stands for…

1. Resource for Consolidation
2. Reference for Consensus
3. Request for Comments
4. Response for Conclusion
5. Reason for Construction
6. Reply for Corollary

C:

The minimal size of an Ethernet DIX (II) Frame in bytes is

1. 46
2. 60
3. 64
4. 72
5. 80
6. 96

D:

TCP operates on…

1. Datagrams
2. Messages
3. Segments
4. Packets
5. Frames
6. Units

E:

Select all incorrect statements

1. The TCP/IP stack has fewer levels compared to the OSI model
2. The TCP/IP stack is independent of the underlying physical technology
3. The TCP/IP stack does not specify the hardware layer
4. The TCP/IP stack has a varying number of layers depending on the experts' views
5. The TCP/IP stack is one of the most popular protocol stacks
6. All the above are incorrect

F:

An ARP message is a “Gratuitous ARP request” when…

1. The target MAC address equals the sender MAC address
2. The sender MAC address equals 00:00:00:00:00:00
3. The sender IP address equals 0.0.0.0
4. Opcode = gratuitous (0x05)
5. The target IP address equals the sender IP address
6. The target hardware address equals ff:ff:ff:ff:ff:ff

G:

What is not possible to accomplish using the ARP protocol?

1. Detect duplicate IP addresses
2. Seamlessly reassign an IP address to a backup host when the primary host crashes
3. Notify hosts of a network about changing the physical host address
4. Identify the IP addresses of hosts that were disconnected from the network
5. Link an arbitrary MAC address with the IP address of another host on the network
6. Assign an arbitrary IP address to a host

H:

Select all incorrect statements about DNS and DNS zone files

1. DNS zone files can have only two record classes
2. DNS zone files can contain records about mail servers
3. The following is not a complete domain name: google.com
4. DNS root servers may give non-authoritative responses
5. DNS servers may use unicast ARP polling to detect a vacant IP address

I:

For a 150Mbit/s Ethernet DIX (II) network, what is the maximum throughput in frames per second for the payload equal to 500 bytes?

1. 23234 f/s
2. 23764 f/s
3. 34851 f/s
4. 35646 f/s
5. 36196 f/s

J:

For a 150Mbit/s Ethernet DIX (II) network, what is the maximum throughput Mbit/s for the payload equal to 500 bytes?

1. 100.9 Mbit/s
2. 113.5 Mbit/s
3. 128.6 Mbit/s
4. 139.4 Mbit/s
5. 144.8 Mbit/s

K:

Please select all correct statements about the Ethernet DIX (II) network.

1. The smaller the message volume in your network protocol, the more efficient your network protocol is for the Ethernet DIX (II) network in terms of performance.
2. The higher the message volume in your network protocol, the less efficient your network protocol is for the Ethernet DIX (II) network in terms of performance.
3. Ethernet DIX (II) technology is at the Data Link layer of the OSI model.
4. IEEE 802.11 WiFi frames have the same format/structure as the Ethernet DIX (II) frames.
5. It is possible to send more than 1500 bytes in a single Ethernet DIX (II) frame

L:

The output of the dig tool is shown below.

```
; <<>> DiG 9.9.4-RedHat-9.9.4-73.el7_6 <<>> ya.ru IN ANY +retry=2 +time=5
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 20619
;; flags: qr rd ra; QUERY: 1, ANSWER: 15, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;ya.ru.                         IN      ANY

;; ANSWER SECTION:
ya.ru.                  3227    IN      SOA     ns1.yandex.ru.
                                                sysadmin.yandex.ru.
                                                2023031600 900 600 2592000 900
ya.ru.                  6827    IN      MX      10 mx.yandex.ru.
ya.ru.                  227     IN      AAAA    2a02:6b8::2:242
ya.ru.                  227     IN      A       77.88.55.242
ya.ru.                  227     IN      A       5.255.255.242
ya.ru.                  6827    IN      NS      ns2.yandex.ru.
ya.ru.                  6827    IN      NS      ns1.yandex.ru.

;; Query time: 17 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Thu Mar 16 16:58:08 MSK 2023
```

Please select all correct statements.

1. Some DNS servers cannot cache the information on the ya.ru domain resource records as there are no appropriate service data.
2. If your e-mail client would like to send an e-mail to example@ya.ru, the e-mail client must negotiate with the mail server using the DNS address `mx.yandex.ru.`
3. All resource records listed above will expire in 2 hours.
4. As the answer above gives only 2 authoritative DNS servers: `ns1.yandex.ru.` and `ns2.yandex.ru.` for the ya.ru. domain, a query to a root DNS server for a resource record of `ya.ru.` will return an error.
5. The IP address `77.88.55.242` of `ya.ru.` will be different after 227 seconds starting from `Thu Mar 16 16:58:08 MSK 2023`.

M:

Consider the following scenario. You bought a domain called `company.org`.
You also bought a data bucket from a Cloud provider to put your static HTML web page into that bucket.
The bucket DNS address is `x-1484770602.eu-west-1.elb.amazonaws.com`
You also executed the `dig` command and discovered that
the IPv4 address for your bucket is 52.211.97.32
Finally, your goal is to publish your HTML web page under the address `fun.company.org`
(a user enters `fun.company.org` into the URL box in a Web browser and sees your HTML page; the user must NOT see the address `x-1484770602.eu-west-1.elb.amazonaws.com` in the URL box of a Web browser).

What is the correct DNS resource record to reach the goal described above?

1. fun.company.org. 4 294 967 297 IN CNAME x-1484770602.eu-west-1.elb.amazonaws.com
2. fun.company.org. 4 294 967 297 IN A 52.211.97.32
3. fun.company.org. 4 294 967 297 IN CNAME x-1484770602.eu-west-1.elb.amazonaws.com.
4. fun.company.org 4 294 967 297 IN A 52.211.97.32
5. fun.company.org. 0 IN CNAME x-1484770602.eu-west-1.elb.amazonaws.com
6. fun.company.org. 0 IN A 52.211.97.32
7. fun.company.org. 0 IN CNAME x-1484770602.eu-west-1.elb.amazonaws.com.
8. fun.company.org 0 IN A 52.211.97.32
9. All the above are incorrect

N:

Given the following ARP message:

```
ar$hrd = 0x1
ar$pro = 0x800
ar$hln = 6
ar$pln = 4
ar$op = 1
ar$sha = JuniperN_bc:9b:70
ar$spa = 172.16.10.22
ar$tha = 00:00:00:00:00:00
ar$tpa = 172.16.10.22
```

where

```
        16.bit: (ar$hrd) Hardware address space (e.g., Ethernet,
                         Packet Radio Net.)
        16.bit: (ar$pro) Protocol address space.  For Ethernet
                         hardware, this is from the set of type
                         fields ether_typ$<protocol>.
         8.bit: (ar$hln) byte length of each hardware address
         8.bit: (ar$pln) byte length of each protocol address
        16.bit: (ar$op)  opcode (ares_op$REQUEST | ares_op$REPLY)
        nbytes: (ar$sha) Hardware address of sender of this
                         packet, n from the ar$hln field.
        mbytes: (ar$spa) Protocol address of sender of this
                         packet, m from the ar$pln field.
        nbytes: (ar$tha) Hardware address of target of this
                         packet (if known).
        mbytes: (ar$tpa) Protocol address of target.
```

What is the purpose of the ARP message given above?

1. Find out someones' hardware address
2. Change of a MAC address for a Virtual IP address
3. Duplicate Address Detection
4. Notify about the change of the L2 address of the sender
5. Check whether a certain IP address if free

O:

Given the following ARP message:

```
ar$hrd = 0x1
ar$pro = 0x800
ar$hln = 6
ar$pln = 4
ar$op = 1
ar$sha = 32:1e:9a:58:7e:d0
ar$spa = 172.16.10.22
ar$tha = 32:1e:9a:58:7e:d0
ar$tpa = 172.16.10.22
```

where

```
        16.bit: (ar$hrd) Hardware address space (e.g., Ethernet,
                         Packet Radio Net.)
        16.bit: (ar$pro) Protocol address space.  For Ethernet
                         hardware, this is from the set of type
                         fields ether_typ$<protocol>.
         8.bit: (ar$hln) byte length of each hardware address
         8.bit: (ar$pln) byte length of each protocol address
        16.bit: (ar$op)  opcode (ares_op$REQUEST | ares_op$REPLY)
        nbytes: (ar$sha) Hardware address of sender of this
                         packet, n from the ar$hln field.
        mbytes: (ar$spa) Protocol address of sender of this
                         packet, m from the ar$pln field.
        nbytes: (ar$tha) Hardware address of target of this
                         packet (if known).
        mbytes: (ar$tpa) Protocol address of target.
```

What is the purpose of the ARP message given above?

1. Find out someone's hardware address
2. Change of a MAC address for a Virtual IP address
3. Duplicate Address Detection
4. Notify the sender of the change in the L2 address
5. Check whether a certain IP address is free

P:

Given the following ARP message:

```
ar$hrd = 0x1
ar$pro = 0x800
ar$hln = 6
ar$pln = 4
ar$op = 1
ar$sha = JuniperN_bc:9b:70
ar$spa = 172.16.05.01
ar$tha = 00:00:00:00:00:00
ar$tpa = 172.16.44.18
```

where

```
        16.bit: (ar$hrd) Hardware address space (e.g., Ethernet,
                         Packet Radio Net.)
        16.bit: (ar$pro) Protocol address space.  For Ethernet
                         hardware, this is from the set of type
                         fields ether_typ$<protocol>.
         8.bit: (ar$hln) byte length of each hardware address
         8.bit: (ar$pln) byte length of each protocol address
        16.bit: (ar$op)  opcode (ares_op$REQUEST | ares_op$REPLY)
        nbytes: (ar$sha) Hardware address of sender of this
                         packet, n from the ar$hln field.
        mbytes: (ar$spa) Protocol address of sender of this
                         packet, m from the ar$pln field.
        nbytes: (ar$tha) Hardware address of target of this
                         packet (if known).
        mbytes: (ar$tpa) Protocol address of target.
```

What is the purpose of the ARP message given above?

1. Find out someone's hardware address
2. Change of a MAC address for a Virtual IP address
3. Duplicate Address Detection
4. Notify the sender of the change in the L2 address
5. Check whether a certain IP address is free

Q:

Please select correct statements about the TCP (Transmission Control Protocol).

1. TCP guarantees data delivery.
2. A TCP module must implement all options that could be found in the TCP header.
3. It is possible to establish a reliable TCP connection in 2 steps.
4. You can find out the data size (in bytes) being transferred by analyzing the TCP header.
5. A custom software network protocol on top of TCP requires buffering.

R:

You are designing a computer network protocol. You would like to consider cases when the messages of your protocol are lost (not arriving at the destination within a given timeout). Please select the best way to react in this situation.

1. The destination host requests a possibly lost message from a host that is potentially on the route of this message.
2. The destination host breaks the connection unilaterally; the source host connects to the destination host again after some period of time.
3. The source host sends (repeats) the message with the same contents as the lost message to the destination host again.
4. The source host sends several copies of the same message via different routes to increase the reliability of the delivery.

S:

You are designing a TCP port scanning software tool. Your goal is to determine whether a certain TCP port on the given host is currently running a software application that listens to that TCP port. How to achieve this using the minimum possible number of messages?

Y = YOUR APPLICATION
H = GIVEN HOST
-> denotes a message

1. Y -> H [SYN seq=10]
2. Y -> H [SYN seq=0]
3. Y -> H [SYN seq=10], H -> Y [ACK ack=11]
4. Y -> H [SYN seq=10], H -> Y [SYN-ACK ack=10, seq=10]
5. Y -> H [SYN seq=10], H -> Y [SYN-ACK ack=11, seq=0]
6. Y -> H [SYN seq=10], H -> Y [SYN-ACK ack=11, seq=0], Y -> H [ACK ack=1, seq=11, data]
7. Y -> H [SYN seq=10], H -> Y [SYN-ACK ack=11, seq=0], Y -> H [ACK ack=0, seq=10, data]

T:

You are designing a high-availability (HA) cluster of servers (located in the data center, not geographically redundant/replicated). You would like (1) the cluster to be available publicly over the Internet, and (2) queries to go to one of the servers, but when a server crashes, subsequent queries can go to another server in the cluster. Please select the best way to implement your HA cluster.

1. With a single IPv4 or IPv6 address, your cluster will be reachable via that address.
2. With a single DNS address, your cluster will be reachable via that address.
3. With several IPv4 or IPv6 addresses, your cluster nodes will be reachable via each address.
4. With several DNS addresses, your cluster nodes will be reachable via each address.