



COURSE DESCRIPTION CARD - SYLLABUS

Course name

Network programming [S1Bioinf1>PSIEC]

Course

Field of study

Bioinformatics

Year/Semester

3/5

Area of study (specialization)

–

Profile of study

general academic

Level of study

first-cycle

Course offered in

Polish

Form of study

full-time

Requirements

elective

Number of hours

Lecture

30

Laboratory classes

30

Other

0

Tutorials

0

Projects/seminars

0

Number of credit points

4,00

Coordinators

dr inż. Michał Sajkowski

michal.sajkowski@put.poznan.pl

dr inż. Andrzej Stroiński

andrzej.stroinski@put.poznan.pl

Lecturers

Prerequisites

Knowledge: Student starting this module should have basic knowledge regarding computer systems organization, algorithms and data structures, and operating systems. Skills: He/she should have skills allowing formulation of algorithms and their programming with the use of at least one widely used software tool. He/she should have skills that are necessary to acquire information from given sources of information. Student should understand the need to extend his/her competences and should express cooperativeness in a team. Social competencies: In addition, in respect to the social skills the student should show attitudes as honesty, responsibility, perseverance, curiosity, creativity, manners, and respect for other people.

Course objective

1. Provide students" knowledge regarding computer networks, within the scope of using, configuration, design and programming of local area and wide area networks, and cognition of technical solutions applied in these networks. 2. Develop students" skills in solving simple problems related to the use and configuration of computer networks as well as network programming. 3. Develop students" skills in team work, especially in configuration, design, and programming of technical solutions applied in computer networks.

Course-related learning outcomes

Knowledge:

1. knows the selected topics considering networking technologies - [K1_W10]
2. has the basic knowledge dealing with life cycle of computing systems - [K1_W12]

Skills:

1. designs and creates computer software according to given specification, using appropriate methods, technics and tools - [K1_U07]
2. is able to prepare, in Polish or English, well documented report and oral presentation dealing with topics from the area of computer networks - [K1_U09]
3. is able to perform the functionality analysis and requirements analysis of computing systems - [K1_U12]

Social competences:

1. understands the need of self-learning all the life and improving the competences - [K1_K01]
2. is able to cooperate and work in a team, accepting various roles - [K1_K02]
3. is able to define the priorities used for the implementation of the task defined by himself or others - [K1_K03]

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Learning outcomes presented above are verified as follows:

Formative assessment:

a) lectures:

based on answers to questions on previous lectures,

b) laboratory classes:

evaluation of doing correctly assigned tasks,

Total assessment:

a) In terms of lectures

- assessment of knowledge and skills in the form of test.

b) In the field of laboratories in the form of average grade based on:

- assessment of knowledge and skills acquired in the laboratory through a test

- entry quizzes in selected laboratories

Programme content

As part of the lecture component, students are introduced to the fundamentals of computer networks, beginning with a historical overview, motivation behind network development, and essential characteristics that modern networks must fulfill. The course covers network architecture models such as OSI and TCP/IP, network topologies, types of networks, and commonly used networking devices. Particular attention is given to the lower layers of the ISO/OSI model and to local area network technologies, including Ethernet (CSMA/CD) and wireless networks (CSMA/CA). Topics such as packet switching, routing principles, path selection algorithms, and routing protocols like RIP and OSPF are also addressed. Additionally, the course covers essential communication protocols, as well as an introduction to network programming—its core concepts, architectures, and the structure of client-server applications, along with different approaches to implementing network-based software.

The laboratory sessions are designed to provide hands-on experience and practical skills required to work with modern computer networks. Students explore key areas such as basic and advanced IPv4 addressing,

network architecture, and essential aspects of the layered network model. They gain experience in configuring Ethernet-based networking devices and setting up Linux systems for IP networking. The labs also include practical exercises in static routing using Linux systems and Cisco routers. A significant portion of the lab work is devoted to network programming—covering core concepts such as client-server architecture, socket programming, and the use of TCP and UDP protocols. Students also engage with the technologies and challenges involved in implementing both client-side and server-side network applications.

Course topics

As part of the lectures, students will learn the following topics:

1. Fundamentals (historical background, motivation, required network features, network architecture – OSI and TCP/IP, network topologies, types of networks, network devices).
2. Lower layers of the ISO/OSI model and local area networks (CSMA/CD – Ethernet, CSMA/CA – wireless networks).
3. Packet switching (switching and routing, route selection – routing algorithms, RIP and OSPF protocols, switching hardware).
4. Basic communication protocols (structure, purpose, standards).
5. Fundamentals of network programming.
6. Network application architecture.
7. Basic concepts such as client, server, P2P.
8. Possible approaches to implementing network applications using TCP / UDP.

As part of the laboratory exercises, students will learn the following topics:

1. Practical aspects of IP addressing and protocols.
2. Automatic assignment of IP addresses.
3. Switching and routing: role, operation, and selected aspects configuration.
4. Configuring Linux systems for network operation.
5. Application layer: protocols, diagnostic tools.
6. Concepts of sockets and client-server architecture.
7. System APIs for handling network sockets.
8. Creating TCP and UDP protocol clients and servers.
9. What TCP and UDP protocols guarantee and what must be handled by the programmer.
10. Domain Name System (DNS) and querying it from the command line and program code.
11. Programming methods for handling multiple simultaneous network connections.
12. Firewall and NAT as mechanisms of server access control.
13. Wireless networks and their security.

Teaching methods

Lectures: multimedia presentation, presentation illustrated with examples presented on blackboard.

Labs: solving tasks, practical exercises with use of network devices, discussion, teamwork, multimedia showcase, programming project.

Bibliography

Basic

1. Unix - Network Services Programming. Networking APIs: Sockets and XTI, W.R. Stevens, Prentice-Hall 1998
2. TCP/IP Protocol Suite, 4th edition, B.A. Forouzan, McGraw-Hill Education, New York, 2009
3. Computer Networks, 5th edition, A.S. Tanenbaum, D.J. Wetherall, Pearson, Boston, 2011
4. Computer Networking: A Top-Down Approach, 7th edition, J.F. Kurose, K.W. Ross, Pearson Education, Boston, 2016
5. Computer Networks: A Systems Approach, L.L. Peterson, B.S. Davie, 5th edition, Morgan Kaufmann, San Francisco, 2012

Additional

1. Network Analysis and Troubleshooting, J. Scott Haugdahl, Addison-Wesley, 1999

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,00
Classes requiring direct contact with the teacher	60	2,50
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	40	1,50