Large-Scale Distributed Systems and Networks

Programme course

13 credits

Storskaliga distribuerade system och nätverk

TDDD93

Valid from: 2017 Spring semester

Determined by
Board of Studies for Computer Science and Media Technology

Date determined
2017-01-25

Main field of study
Computer Science

Course level
First cycle

Advancement level
G1X

Course offered for
- Computer Science and Software Engineering, M Sc in Engineering

Specific information
This course is not available for exchange students

Entry requirements
Note: Admission requirements for non-programme students usually also include admission requirements for the programme and threshold requirements for progression within the programme, or corresponding.

Prerequisites
Calculus, statistics, and programming knowledge (preferably both in java and C).

Intended learning outcomes
Computer networks have become an indispensable part of the infrastructure of our modern society. With billions of people and devices being connected and using critical distributed services implemented over the Internet, for example, it is becoming very important to understand how these networks, as well as the distributed systems and services operating over these networks, are designed to scale to large number of machines and users. Also, at the level of individual machines, it is important to know how to build applications and services that effectively scale with the resources (e.g., the number of cores, processors, etc.). In this course, we will use a combination of theory and practice (including exploration of real data) to gain a deeper understanding of modern large-scale systems and services.

Within the area of computer networks, participants successfully completing the course are expected to be able to:

- Explain, describe, and analyze a typical network architecture, including arguing regarding the importance of network layers and encapsulation
- Explain the different basic types of protocols, communication channels, and

network types
- Design, implement, verify, and test your own protocols
- Explain fundamental performance tradeoffs, including showing an understanding of where delays can occur in a network, what different types of delay that exist, the impact of packet losses and jitter on various protocols
- Using concrete examples, and in detail, describe the interaction between the different protocols in the network architecture, and the protocols associated with the different layers
- Describe and analyze the most common application architectures in the Internet, how the most important application-layer protocols work, the service they provide, as well as have the ability to design and implement their own application-layer protocols
- Analyze and explain important design considerations at the transport layer, including hands-on knowledge of how flow control and congestion control works, and how reliable data transfer is implemented
- Motivate and explain how routing and forwarding is implemented on the Internet, including basic design and implementation principles of network-layer protocols used to ensure scalability
- Describe and explain different link-layer technologies and how they work
- Exemplify how different types of security services can be implemented in different layers with the help of different standards
- Analyze and exemplify some of the unique challenges as we are moving towards increasingly mobile users
- Explain and discuss the fundamentals of how multimedia services are provided over the Internet

The students are also expected to understand how distributed systems can be built on-top of the network architecture to provide scalable services, as well as how multicore systems and embedded systems can be used to further enhance services. More specifically, after successfully completing the course the student is expected to be able to:

- Define what a distributed system is and its most important goals
- Explain the relationship between architectures, processes and communication
- Exemplify different types of transparency, scaling techniques
- Analyze and explain some of the fundamental differences in different system architectures
- Describe and explain how to achieve synchronization, consistency and replication
- Implement, motivate, and explain the design of various types if distributed system architectures, including object-based distributed systems (e.g., using Corba) and Web-based distributed systems (including how a proxy cache works)
- Understand fundamental homogeneous and heterogeneous multicore architecture concepts and their performance implications; basic techniques for multicore programming with threads and tasks; and some techniques to design parallel algorithms and analyze their complexity, including parallel scalability

- Understand system-level methods and tools for the design of real-time embedded systems; understand basic tradeoffs and design implications that need to be taken into consideration when making system-level design decisions; and place the design in a bigger context (including in the context of the hardware architecture and software implementation).

By introducing general design concepts, some basic scientific methodologies (such as basic systems performance modeling), exploration of real-world data, and a general systems thinking, with scale and performance as important aspects, used throughout the course, we also expect the student to be able to:

- Explain using concrete examples fundamental network design principles and scalability tradeoffs
- Design and perform targeted experiments to critically evaluate network and distributed systems technologies
- Apply basic system models and analysis methods to analyze systems and networks
- As a team, plan and conduct a project study of an identified problem within a selected technology area, including experiments using real data sources (in some cases collected by the students themselves)
- Based on a project study, present and explain (both written and orally) findings within a selected area of technology, to an audience with similar general knowledge of computer networks
- Give/receive constructive feedback to/from other students

Course content

Computer networking: The fundamental design principles of computer networks, their protocols, and the Internet stack. Application layer protocols (e.g., HTTP), transport layer protocols (e.g. TCP), network layer protocols (e.g., IP and BGP), link layer protocols (e.g., Ethernet, WiFi, and Bluetooth). Introduction to multimedia applications, wireless networking, and network security for each of the layers.

Distributed systems, multicore systems, and embedded systems: Basic distributed architectures and their processes and communication. Synchronization, replication, and consistency issues and tradeoffs. Object-based and Web-based systems. Multicore architectures, their opportunities, and basic challenges they present. Embedded systems and their integration into a wide range of modern systems.

Project: The precise topics for the project will vary slightly from year to year, to keep the projects exciting and up-to-date with developments in the areas. Recurring topics include: Fundamental properties of large-scale systems (e.g., power laws, rich-gets-richer); Scalable systems and designs (e.g., hierarchical vs. flat designs; layered designs); Measurement, modeling and analysis methods using real network data; Important modern distributed systems such as cloud-based services (e.g., EC2), CDNs, the Internet routing architecture itself, and social networks.

Teaching and working methods

The course consists of both theory (lectures and assignments) and practical hands-on training and exploration (lab assignments and project). The course has two written exams. The first on networking and the second on introductory material about distributed systems, multicore systems, embedded systems, and basic system science methodologies. The project should result in a written report, should be presented in a seminar during which the students will act as both presenters and opponents (evaluating and providing each other with feedback, such as to improve the reports and projects). Assignments should be done in pairs. The projects should be done in groups of thee-to-four students. The course runs over the entire spring semester.

Examination

PRA1	Project work	3 credits	U, G
LAB1	Laboratory work	4 credits	U, G
TEN2	Written examination	3 credits	U, 3, 4, 5
TEN1	Written examination	3 credits	U, 3, 4, 5

For a pass grade in the course, at least a pass grade is needed for all components. The final grade will be calculated using the average from the individual exam grades.

Grades

Four-grade scale, LiU, U, 3, 4, 5

Other information

Supplementary courses:
Distributed Systems; Advanced Networking; Individual projects

Department

Institutionen för datavetenskap

Director of Studies or equivalent

Patrick Lambrix

Examiner

Niklas Carlsson

Education components

Preliminary scheduled hours: 80 h
Recommended self-study hours: 267 h

Course literature

J. F. Kurose and K. W. Ross (2012), Computer networking: A top-down approach. Sixth Edition. Pearson. TBD [top-two candidates]: G. Coulouris, J. Dollimore, T. Kindberg, and G. Blair (2011), Distributed Systems: Concepts and Design, Fifth Edition, Pearson. Andrew S. Tanenbaum and Maarten van Steen (2007), Distributed Systems: Principles and Paradigms, Second Edition, Pearson Education.

Common rules

Regulations (apply to LiU in its entirety)

The university is a government agency whose operations are regulated by legislation and ordinances, which include the Higher Education Act and the Higher Education Ordinance. In addition to legislation and ordinances, operations are subject to several policy documents. The Linköping University rule book collects currently valid decisions of a regulatory nature taken by the university board, the vice-chancellor and faculty/department boards.

LiU's rule book for education at first-cycle and second-cycle levels is available at http://styrdokument.liu.se/Regelsamling/Innehall/Utbildning_pa_grund-_och_avancerad_niva.