

Advanced Networking

Course Introduction

Antonio Carzaniga

Faculty of Informatics
Università della Svizzera italiana

February 22, 2021

- General course information
- Program
- Preliminary schedule
- A preview of *Advanced Networking*

- On-line course information

- ▶ on iCorsi

- ▶ and on my web page: <https://www.inf.usi.ch/carzaniga/edu/adv-ntw/>

- On-line course information

- ▶ on iCorsi

- ▶ and on my web page: <https://www.inf.usi.ch/carzaniga/edu/adv-ntw/>

- Announcements (***you are responsible for reading them!***)

- ▶ through iCorsi

- ▶ and <https://www.inf.usi.ch/carzaniga/edu/adv-ntw/news.html>

■ On-line course information

- ▶ on iCorsi
- ▶ and on my web page: <https://www.inf.usi.ch/carzaniga/edu/adv-ntw/>

■ Announcements (*you are responsible for reading them!*)

- ▶ through iCorsi
- ▶ and <https://www.inf.usi.ch/carzaniga/edu/adv-ntw/news.html>

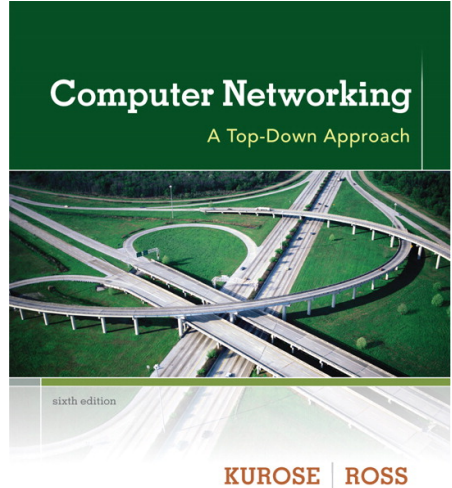
■ Office hours

- ▶ Antonio Carzaniga: *by appointment*
- ▶ Ali Fattaholmanan: *by appointment*

Computer Networking A Top-Down Approach

James F. Kurose
Keith W. Ross

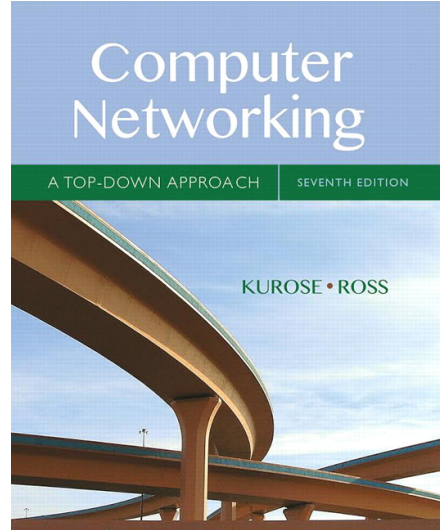
Addison-Wesley



Computer Networking A Top-Down Approach

James F. Kurose
Keith W. Ross

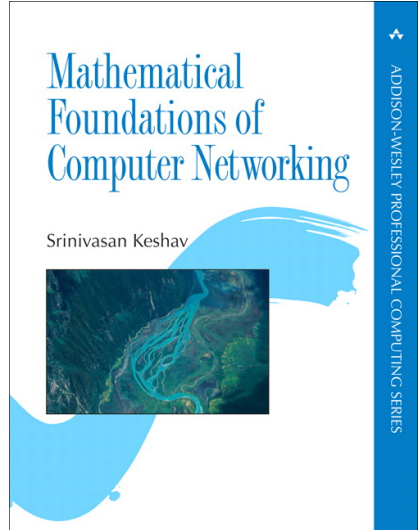
Addison-Wesley



Mathematical Foundations of Computer Networking

Srinivasan Keshav

Addison-Wesley Professional



- +70% homework assignments and projects
- +30% paper presentations
- $\pm 10\%$ instructor's discretionary evaluation
 - ▶ participation
 - ▶ extra credits
 - ▶ trajectory
 - ▶ ...

*A student should never take someone else's material and present it as his or her own.
Doing so means committing plagiarism.*

A student should never take someone else's material and present it as his or her own. Doing so means committing plagiarism.

**Do NOT take someone else's material
and present it as your own!**

*A student should never take someone else's material and present it as his or her own.
Doing so means committing plagiarism.*

**Do NOT take someone else's material
and present it as your own!**

ALWAYS properly acknowledge your sources!

*A student should never take someone else's material and present it as his or her own.
Doing so means committing plagiarism.*

*A student should never take someone else's material and present it as his or her own.
Doing so means committing plagiarism.*

material means: ideas, words, code, suggestions,
corrections on one's work, etc.

A student should never take someone else's material and present it as his or her own. Doing so means committing plagiarism.

material means: ideas, words, code, suggestions, corrections on one's work, etc.

- Using someone else's material may be appropriate
 - ▶ e.g., software libraries
 - ▶ ***always clearly identify the external material, and acknowledge its source; failing to do so means committing plagiarism.***
 - ▶ the work will be evaluated based on its *added value*

- Committing plagiarism on an assignment or an exam will result in ***failing that assignment or that exam***
- Penalties may be escalated in accordance with the regulations of the Faculty of Informatics

Deadlines are firm.

Deadlines are firm.

Deadlines Are Firm!

Deadlines are firm.

Deadlines Are Firm!

- Exceptions may be granted
 - ▶ at the instructors' discretion
 - ▶ only for documented medical conditions or other documented emergencies

Deadlines are firm.

Deadlines Are Firm!

- Exceptions may be granted
 - ▶ at the instructors' discretion
 - ▶ only for documented medical conditions or other documented emergencies
- Each late day will reduce the assignment's grade by *one third of the total value* of that assignment

Deadlines are firm.

Deadlines Are Firm!

- Exceptions may be granted
 - ▶ at the instructors' discretion
 - ▶ only for documented medical conditions or other documented emergencies
- Each late day will reduce the assignment's grade by *one third of the total value* of that assignment
 - ▶ corollary: the grade of an assignment turned in more than two days late is 0

What this course is about

You thought computer networking is simple?!

■ Problem-solving

1. I give you a problem, which we discuss together
2. You solve it on your own without any directions
3. We discuss your solutions
4. I present my solution
5. We generalize and study the theory

■ Problem-solving

1. I give you a problem, which we discuss together
2. You solve it on your own without any directions
3. We discuss your solutions
4. I present my solution
5. We generalize and study the theory

■ The Feynman Technique (sort-of)

- ▶ The best way to learn a concept is to *teach it!*
- ▶ Seminars on topics of your choice, possibly including the topics of the course
- ▶ We all discuss, but the point is that *you* are the teacher!

- Recap on Basic Networking Concepts: Network architecture, application protocols, TCP, datagram network service, router architecture, forwarding, routing, and in particular link-state routing.

- Recap on Basic Networking Concepts: Network architecture, application protocols, TCP, datagram network service, router architecture, forwarding, routing, and in particular link-state routing.
- Sampling from the nodes in a network
 - ▶ Graph model; Monte-Carlo simulation; analytic solutions; design and optimization

- Recap on Basic Networking Concepts: Network architecture, application protocols, TCP, datagram network service, router architecture, forwarding, routing, and in particular link-state routing.
- Sampling from the nodes in a network
 - ▶ Graph model; Monte-Carlo simulation; analytic solutions; design and optimization
- Traffic engineering
 - ▶ Flow problems; linear programming; whole-flow optimization; integer linear programming; randomized routing schemes.

- Recap on Basic Networking Concepts: Network architecture, application protocols, TCP, datagram network service, router architecture, forwarding, routing, and in particular link-state routing.
- Sampling from the nodes in a network
 - ▶ Graph model; Monte-Carlo simulation; analytic solutions; design and optimization
- Traffic engineering
 - ▶ Flow problems; linear programming; whole-flow optimization; integer linear programming; randomized routing schemes.
- Network modeling and simulation
 - ▶ Packet-level modeling and simulation

- A network as a queuing system
 - ▶ Basics of queueing models; basic results in queueing theory; Little's theorem and applications; Poisson processes; analysis of an M/M/1 queue and applications; statistical multiplexing

- A network as a queuing system
 - ▶ Basics of queueing models; basic results in queueing theory; Little's theorem and applications; Poisson processes; analysis of an M/M/1 queue and applications; statistical multiplexing
- Network and Communication Security
 - ▶ Basics of communication security; modern cryptography and provable security; basics of symmetric cryptography; basics of public-key cryptography; concrete protocols and systems: IPSec

■ A network as a queuing system

- ▶ Basics of queueing models; basic results in queueing theory; Little's theorem and applications; Poisson processes; analysis of an M/M/1 queue and applications; statistical multiplexing

■ Network and Communication Security

- ▶ Basics of communication security; modern cryptography and provable security; basics of symmetric cryptography; basics of public-key cryptography; concrete protocols and systems: IPsec

■ Advanced Architectures and Protocols

- ▶ The modern Web: HTTP/2; the future Web: HTTP/3? Data-center networking: architectures and protocols; DCTCP; Timely

■ A network as a queuing system

- ▶ Basics of queueing models; basic results in queueing theory; Little's theorem and applications; Poisson processes; analysis of an M/M/1 queue and applications; statistical multiplexing

■ Network and Communication Security

- ▶ Basics of communication security; modern cryptography and provable security; basics of symmetric cryptography; basics of public-key cryptography; concrete protocols and systems: IPsec

■ Advanced Architectures and Protocols

- ▶ The modern Web: HTTP/2; the future Web: HTTP/3? Data-center networking: architectures and protocols; DCTCP; Timely

■ Programmable Networks

- ▶ SDN: programming the control plane: the OpenFlow interface. Programmable data plane: P4.