Advanced Computer Networking (ACN)

IN2097 – WiSe 2025–2026

Prof. Dr.-Ing. Georg Carle, Sebastian Gallenmüller

Christian Dietze, Marcel Kempf, Lorenz Lehle

Chair of Network Architectures and Services
School of Computation, Information and Technology
Technical University of Munich

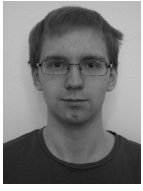
Introduction & Organization

Lecturers:

Prof. Dr.-Ing. Georg Carle

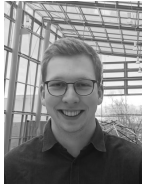
Sebastian Gallenmüller

Teaching Assistants:



Christian Dietze

Marcel Kempf



Lorenz Lehle

Introduction
Georg Carle

Professional career:

1985	–	1992	Studies of Electrical Engineering, University of Stuttgart, Germany
1988	–	1989	Master of Science, Brunel University, London, UK
		1990	Ecole nationale Supérieure des Télécommunications (ENST), Paris, France
1992	–	1996	PhD in Computer Science at University of Karlsruhe, Germany
		1997	Postdoc at Institut Eurecom, Sophia Antipolis, France
1997	–	2002	Fraunhofer FOKUS, Berlin, Germany
			Head of Competence Center Global Networking
2003	–	2008	Professor, University of Tübingen, Germany
Since 2008			Professor, Technical University of Munich, Germany

Further positions:

- Since 1997 co-PI in many national and international projects
- Since 2022 Deputy head, Department of Computer Engineering
- 2013-2022 Information Officer of Department of Informatics at TUM (previously Managing Director)
- Secretary of IFIP Working Group 6.2 Network and Internetwork Architecture
- Steering Committee member of the TUM-IMT German-French Academy for the Industry of the Future
- Scientific Institution Representative of the Interim Supervisory Board of the Scientific Large-scale Infrastructure for Computing/Communication Experimental Studies (SLICES-RI) European Research Infrastructure

Who studies what?

- Master in Informatics?
- Master in Informatics: Games Engineering?
- Master in Information Systems (*Wirtschaftsinformatik*)?
- Other master courses?
- Bachelor in Informatics?
- Bachelor in Informatics: Games Engineering?
- Bachelor in Information Systems?
- Other bachelor courses?

Previous relevant courses?

- Grundlagen Rechnernetze und Verteilte Systeme (GRNVS)?
- iLab (Internet Lab)?
- Network Security?
- Peer-to-Peer Communications and Security?
- Network Coding?
- Other courses in Computer Networks?

Goals of the course:

- Learn to take responsibility for yourself
- Think about the topics
  - Do not aim just being able to repeat content of these slides without deeper understanding
- Learn to *reflect* on technical problems
- Learn to apply your knowledge
  - Use Moodle forum of this course for technical discussions and for answering questions
- Understand the principles
  - What is the essence to be remembered in some years?
  - What would you consider suitable questions in an exam?
- Learn from practical project performed during the course

General learning outcomes - Bloom's taxonomy

1. Knowledge
   - ⇒ Being able to reproduce facts
2. Understanding
   - ⇒ Being able to explain properties with own words
3. Applying
   - ⇒ Apply known methods to solve questions
4. Analyzing
   - ⇒ Identifying the inherent structure of a complex system
5. Synthesis
   - ⇒ Creating new solutions — from known elements
6. Assessment
   - ⇒ Identifying suitable criteria and perform assessment

General learning outcomes of this course

Knowledge, Understanding, Applying

- Protocols: data link layer, network layer, transport layer, application layer
- Concepts: measurements, signaling, QoS, resilience

⇒ Lectures, exercise questions, final exam

Analyzing, Synthesis, Assessment

- Special context: network properties
- Tools: git, measurement tools, DPDK, ...
- Methods: plan solution, program, administer experiment setup, measure, reflect, document

⇒ Course project

Course overview (*to be modified . . .*)

Part 1: Internet protocols – an overview on computer networks link layer

- Overview on computer networks
- Link layer
- Software-Defined Networking
- Internet structure
- Transport layer
- Application layer

Part 2: Advanced concepts

- Measurements
- Quality of Service
- Network Calculus
- Node architectures and mechanisms
- Design principles

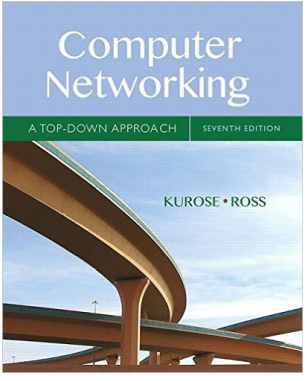
Acknowledgements:

Parts of the course are based on this book:

- J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach, 7th ed. Addison Wesley, 2016

Jim Kurose, University of Massachusetts, Amherst, USA

Keith Ross, New York University, USA

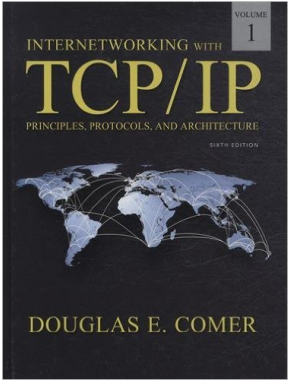


Acknowledgements

Additional book relevant for the course:

- D. E. Comer, Internetworking With TCP/IP, Principles Protocols, and Architecture, 5th ed. Prentice Hall, Englewood Cliffs, 2006, vol. 1

Douglas Comer, Purdue University, West Lafayette, USA



Course organization

Times and addresses

Time slots

- Tuesday, 16:15 - 17:45, Interims HS 1
- Thursday, 14:15 - 15:45, Interims HS 2
- Exercises are done on certain days (about each 2–3 weeks)
- Exercise timeslot: Thursday, 14:15 - 15:45
- You can ask questions during the exercise sessions
- Online stream via live.rbg.tum.de (best effort; we do not guarantee availability of stream)

TUMonline

- Registration is required for access to course infrastructure
- Exam registration is required

Course material

- Slides are available online (may be updated during the course)
- Additional supporting material (exercise sheets, exams of previous semesters)
- Web address: https://acn.net.cit.tum.de
- Git access will be provided next week

Course organization

Questions and answers

- Lecturers:
  - Prof. Dr.-Ing. Georg Carle
  - Sebastian Gallenmüller
- Teaching assistants:
  - Christian Dietze
  - Marcel Kempf
  - Lorenz Lehle
- For all matters regarding ACN only use: `acn@net.in.tum.de`
- All lecturers and teaching assistants will get the messages of this mailing list

Course organization

User forum

- ACN course page on moodle.tum.de
- Tool for collaboration
- You can ask questions and other students / teaching assistants answer them

Course organization

Exam

- Exam date (preliminary; date/time may change): Wed, Feb. 18, 2026 14:00–16:15 (CET)
- Written exam at the end of the semester (75 min, 75 credits)
- Official date to be announced via TUMonline

Bonus

- Exercise (up to 60 credits)
- Project (up to 10 credits)
- Teamwork is not allowed
- bonusCredits = min(15, (creditsExercise / 6 + creditsProject))
- finalGrade = grade(creditsFinalExam + bonusCredits)
- Bonus is only added IFF the final exam is passed without bonus, i.e., 4.0 or better

Course organization

People caught cheating in any submission are excluded from the entire bonus system.
Adhere to the official guidelines of our department/school:

- EN, http://go.tum.de/103707
- DE, http://go.tum.de/750259

What about AI tools?

- Official TUM citation guidelines, https://mediatum.ub.tum.de/1723332?show_id=1236069
- Use of AI must be disclosed
- Rule of thumb:
  - Using AI for idea collection, formatting, translating, or styling is fine
  - Using AI for replacing your own thought process is typically not allowed and may lead to disqualifications from the bonus system

Course organization

Retake exam

- Retake exam date (preliminary, date/time may change): Tue, Apr. 07, 2026 08:00–09:15 (CEST)
- Retake exam will be a written exam
- The bonus will also be valid for the retake exam
- You need to register for the retake exam separately (usually starting in mid-March)
- You do not need to be registered for the endterm exam to participate in the retake exam

Exercise and project

Approach to exercises

- Self-correction
- Gain insight by reviewing your own mistakes

Regular 2-week exercise process

1. New problem is released on a Thursday
2. Submission via Git as an electronic notebook on a Thursday one week later
3. Discussion of solution during the Thursday lecture slot
4. Submission of self-corrected solution until Tuesday of the following week

Exercise and project

Self-correction methodology

- Learn from your mistakes
- Improve your solution
- **Do not copy the presented sample solution, adapt your own solution!**
- Correct mistakes in first submission
- Submit via Git

Exercise and project

Submission process

- Everyone gets an individual Git repository (hosted at the LRZ GitLab)
- **Note:** There are different Git repositories for downloading lecture slides or the project code
- Access with personal SSH public key
- Put the submission in the correct folder
- Commit **and push** before the deadline
- More details on how to access the infrastructure is provided in the first exercise sheet

Exercise and project

Grading

- After final submission we will grade your initial solution and the correction
- Grades will be published in your individual Git repository
- Solution will be released after grading is finished

Jupyter Notebook

- Will be used for the exercises
- Think of it as an interactive worksheet
- Write python code and plot graphs directly in your answers
- Accessible via your browser
- Hosted on a VM (no configuration required)

Exercise and project

jupyter Welcome to Python (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Markdown

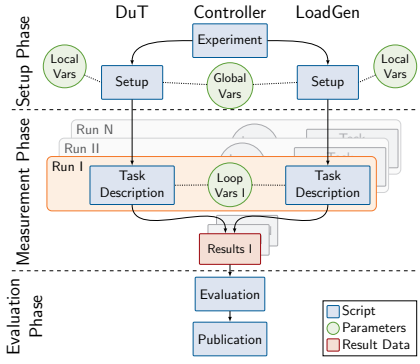
```
import pandas as pd
import numpy as np
import matplotlib

from matplotlib import pyplot as plt
import seaborn as sns

ts = pd.Series(np.random.randn(1000), index=pd.date_range('1/1/2000', periods=1000))
ts = ts.cumsum()

df = pd.DataFrame(np.random.randn(1000, 4), index=ts.index,
                  columns=['A', 'B', 'C', 'D'])
df = df.cumsum()
df.plot(); plt.legend(loc='best')
```

Figure 1

Exercise and project

Infrastructure for winter term 25/26

- Our research group develops a scientific testbed:
  - plain orchestrating service (pos)[3][4]
  - pos framework makes experiments reproducible
  - Primarily used for our own research
- You will get access to the testbed for exercise and project
- Testbed is remotely accessible via ssh
- Benefits of the new infrastructure:
  - Access to server-grade hardware
  - Reproducible experiments guaranteed by pos framework
  - First-hand experience with up-to-date research facilities



pos' experiment workflow

[3] S. Gallenmüller et al., "The pos framework: A methodology and toolchain for reproducible network experiments", in CoNEXT '21, Virtual Event, Munich, Germany, December 7 - 10, 2021, ACM, 2021, pp. 259–266. DOI: 10.1145/3485983.3494841

[4] https://www.youtube.com/watch?v=qtYifgkmUSI

Exercise and project

Project

This term we offer the project:

- Software Router on a Research Infrastructure
- The maximum number of bonus credits from the project is limited to 10

Exercise and project

Project software router

- Implement a software router
- Using the packet processing framework DPDK
- Programming language: C or C++
- You get access to a scientific testbed for developing, testing and optimizing your router

- Submissions using Git repository
- Project deliverables are graded

Exercise and project

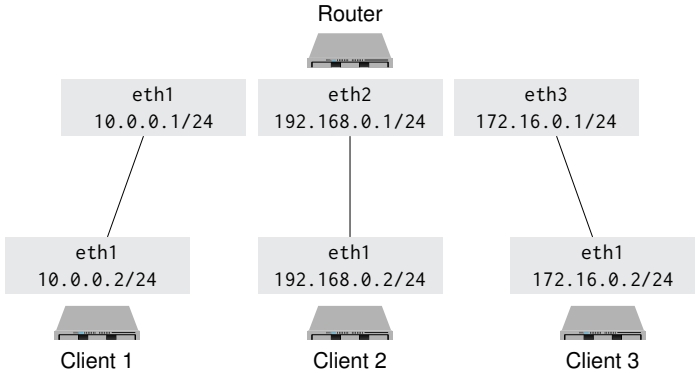
- Implementation of a high-performance software router
- High-performance packet processing framework DPDK
- Programming language: C or C++
- You get access to a scientific testbed for developing, testing and optimizing your router

- Submissions using Git repository
- Project deliverables are graded

Exercise and project

Step 1

- Get to know the pos testbed
- Configure your virtual machines (boot OS)
- Configure the VM setup (network interfaces)
- Compile & configure DPDK
- Test your setup with a simple DPDK forwarding example
- Submission: scripts configuring router and clients



Exercise and project

Step 2

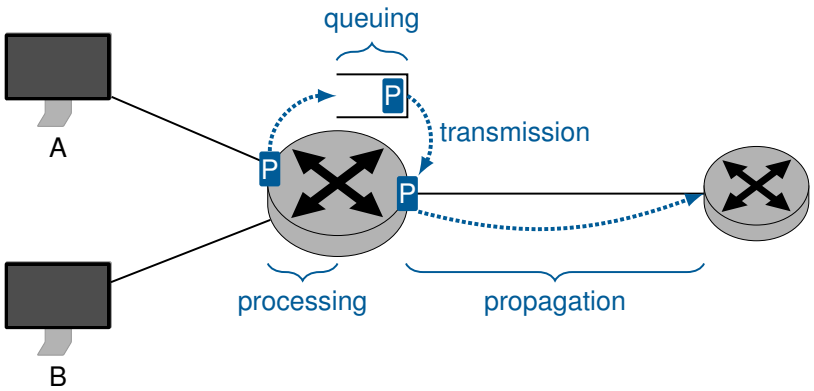
- Command line interface
- Router should answer the clients' ARP requests
- Sanity checks on IP packets
- Do routing decision and forward packets accordingly

Step 3

- Implement a routing table
- Algorithm of choice: DIR-24-8
- Integrate routing table into your software router

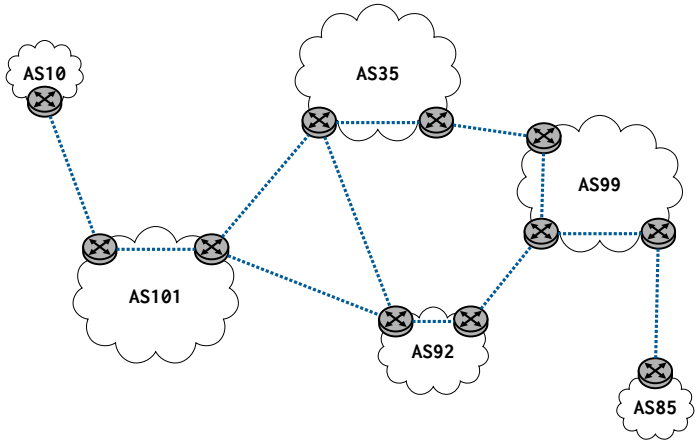
Step 4

- Implement a performance benchmark
- Use pos to automate the benchmark and its evaluation
- Create an automated test report of your findings

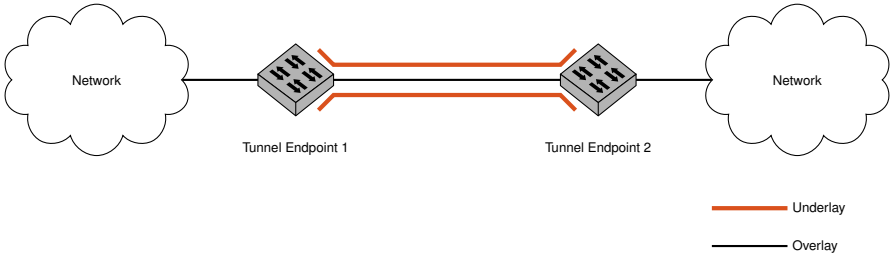

queuing
P
transmission
A
P
P
processing
propagation
B

Lecture overview
Internet structure

- Autonomous systems (AS level structure)
- Routers and hosts (IP level structure)



Lecture overview
Tunneling



- Tunneling is the art of encapsulating datagrams inside other datagrams
- Most widely known examples are VPNs

Lecture overview
Network layer - routing

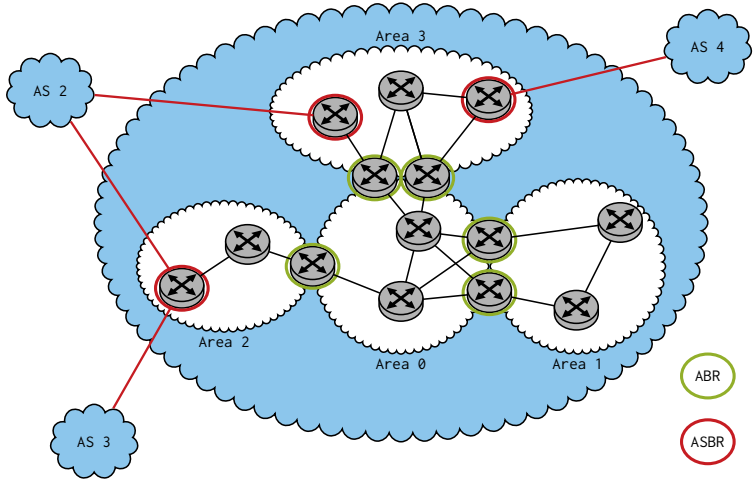
Routing algorithms

- Link state
- Distance Vector
- Hierarchical routing

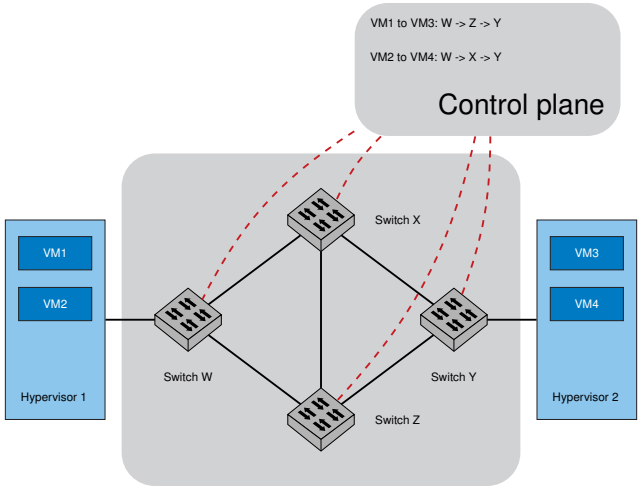
Routing in the Internet

- RIP
- OSPF
- BGP

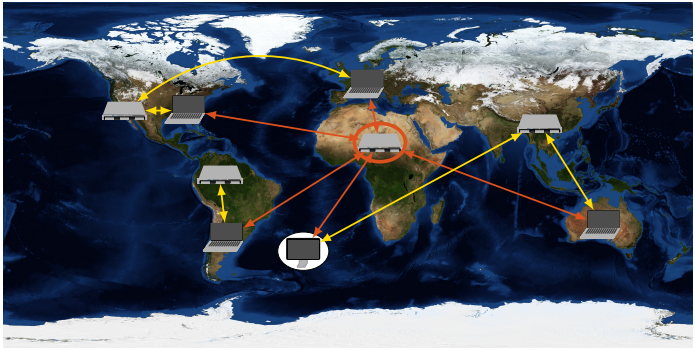
Broadcast and multicast routing



Example OSPF network


VM1 to VM3: W -> Z -> Y
VM2 to VM4: W -> X -> Y
Control plane
Switch X
VM1
VM2
VM3
VM4
Switch W
Switch Y
Hypervisor 1
Hypervisor 2
Switch Z
Data plane

Lecture overview
CDN

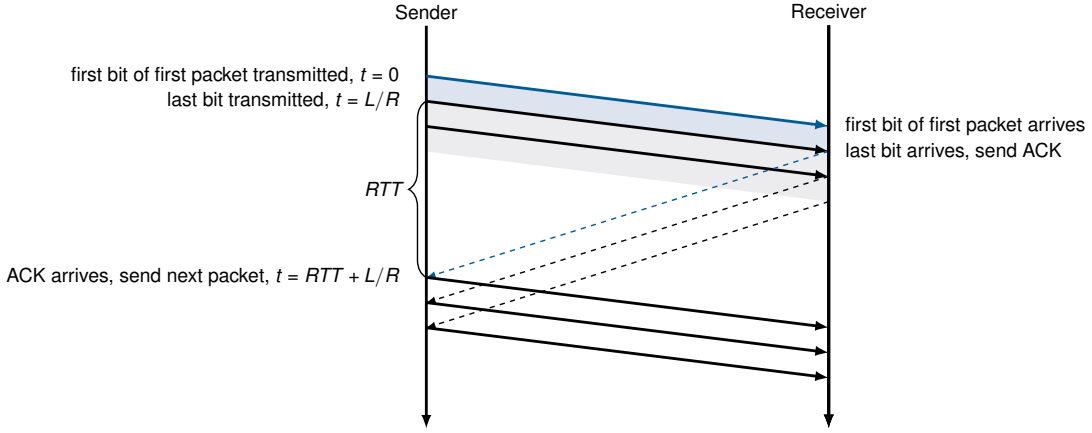


- Network traffic is constantly growing
- Growth/Scaling can be achieved using CDNs

Lecture overview
Transport layer services

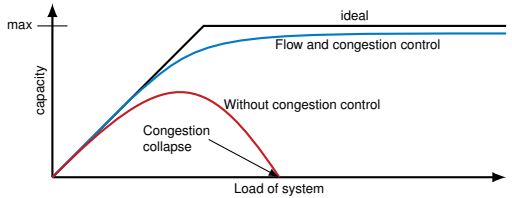
- Transport-layer services
- Multiplexing and demultiplexing
- Connectionless transport: UDP
- Connection-oriented transport: TCP
  - Segment structure
  - Reliable data transfer
  - Flow control
  - Connection management
- TCP congestion control
- QUIC

Pipelining for increased utilization



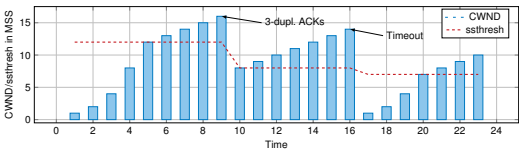
$$U_{sender} = \frac{3 \cdot L/R}{RTT + L/R}$$

Lecture overview
TCP Congestion Control

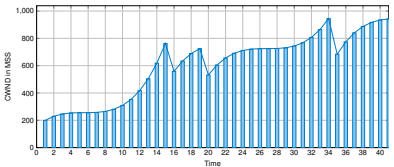


Congestion is bad...

- So... How exactly do we control it? ⟶ Ongoing research effort



TCP Reno

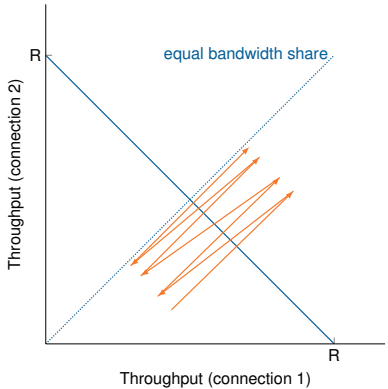


TCP Cubic

Lecture overview
Why is TCP fair?

Two competing sessions:

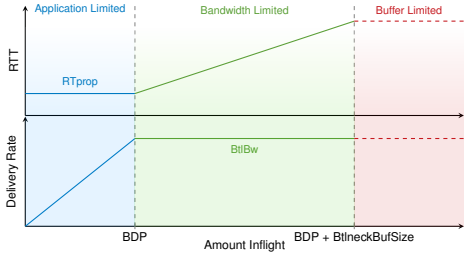
- Additive increase gives slope of 1, as throughput increases
- Multiplicative decrease decreases throughput proportionally



Lecture overview
Does 36 year old TCP even have a place here?

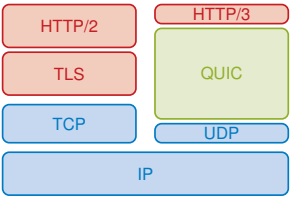
Let's squeeze all out of it

- TCP BBR
- Newest Congestion control algorithm from Google
- Achieves high throughput ...
- ... while maintaining low latency
- No need to adapt applications



Newer alternative: QUIC

- Way faster development cycle
- Built-in encryption support
- 0-RTT handshake (with a bit of luck...)
- No head-of-line blocking
- IP mobility proof
- Shiny new toy the cool kids play with :)



Lecture overview
Network measurements

- Introduction
- Architecture & mechanisms
- Protocols
  - IPFIX (netflow accounting)
  - PSAMP (packet sampling)
- Scenarios

[1] J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach, 7th ed. Addison Wesley, 2016.

[2] D. E. Comer, Internetworking With TCP/IP, Principles Protocols, and Architecture, 5th ed. Prentice Hall, Englewood Cliffs, 2006, vol. 1.

[3] S. Gallenmüller, D. Scholz, H. Stubbe, and G. Carle, "The pos framework: A methodology and toolchain for reproducible network experiments," in CoNEXT '21, Virtual Event, Munich, Germany, December 7 - 10, 2021, ACM, 2021, pp. 259–266. DOI: 10.1145/3485983.3494841.