Advanced Computer Networking (ACN)

IN2097 – WiSe 2023–2024

Prof. Dr.-Ing. Georg Carle

Sebastian Gallenmüller, Max Helm, Benedikt Jaeger,
Marcel Kempf, Patrick Sattler, Johannes Zirngibl

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

Introduction & Organization

Introduction

Course organization

Exercise and project

Lecture overview

Bibliography

Introduction & Organization

Introduction

Course organization

Exercise and project

Lecture overview

Bibliography

Introduction
People

Lecturer: Prof. Dr.-Ing. Georg Carle

Teaching Assistants:

Sebastian Gallenmüller

Max Helm

Benedikt Jaeger

Marcel Kempf

Patrick Sattler

Johannes Zirngibl

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 Télécommunitaions (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

Introduction

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?

Introduction

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?

Introduction

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

Introduction

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

Introduction

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

Introduction

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

Introduction

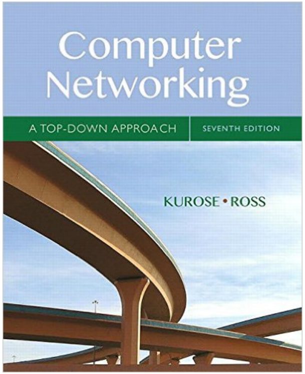
Acknowledgements:

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



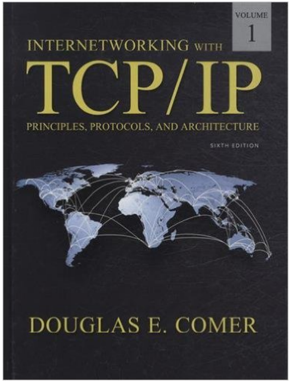
Introduction

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



Introduction & Organization

Introduction

Course organization

Exercise and project

Lecture overview

Bibliography

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 to 3 weeks)
- Exercise timeslot: Thursday, 14:15 - 15:45
- You can ask questions during the exercise sessions

TUMonline

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

Course material

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

Course organization

Questions and answers

- Prof. Dr.-Ing. Georg Carle
- Teaching assistants
  - Sebastian Gallenmüller
  - Max Helm
  - Benedikt Jaeger
  - Marcel Kempf
  - Patrick Sattler
  - Johannes Zirngibl

  - Coordination of exercises and project
  - Contact: `acn@net.in.tum.de`

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): Mo, Feb. 19, 2024 13:30-14:45 (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)
- No teamwork 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

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

Course organization

Retake Exam

- Retake exam date (preliminary, date/time may change): Wed, Apr. 03, 2024 8:00-9:15 (CEST)
- There will be a written on-site retake exam
- The bonus will also be valid for the retake exam
- You need to register for the retake exam separately (usually starting at mid of March)
- You do not need to be registered for the main exam to participate in the retake exam

Introduction & Organization

Introduction

Course organization

Exercise and project

Lecture overview

Bibliography

Exercise and project

Approach to exercises

- Self correction
- Gain insight by reviewing 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 how to access the infrastructure is provided in the first exercise sheet

Exercise and project

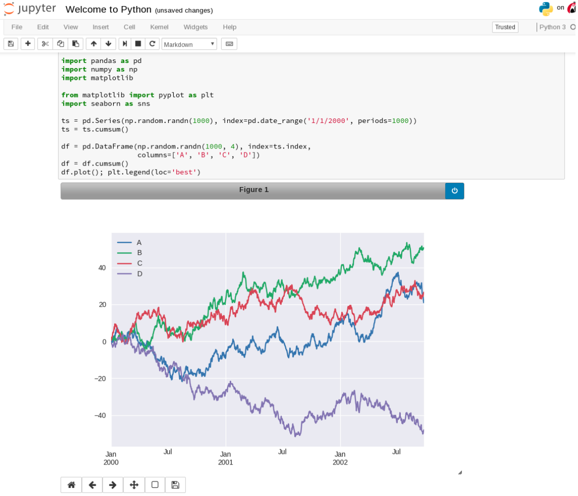
Grading

- 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



```
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')
```

Exercise and project

Projects

This term we offer two different projects

- QUIC Interoperability Runner
- 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

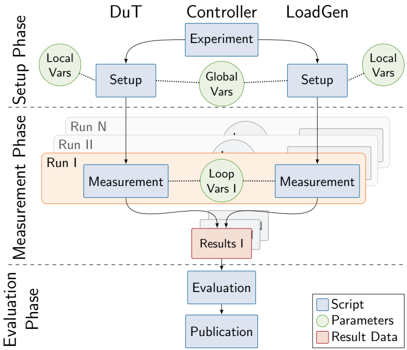
- Implement a software router
- Using the packet processing framework DPDK
- Programming language: C or C++
- You get access to a scientific testbed for developing, testing and optimizing your router

- Submissions using git repository
- Project deliverables are graded

Exercise and project

New Infrastructure for winter term 23/24

- 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 development and testing your implemenation
- 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] [3] S. Gallenmüller, D. Scholz, H. Stubbe, 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

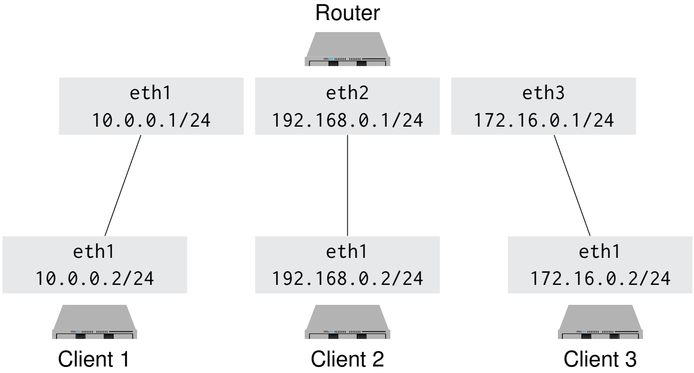
- Implementation of a high-performance software router
- High-performance packet processing framework DPDK
- Programming language: C or C++
- You get access to a scientific testbed for developing, testing and optimizing your router

- 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



Router

eth1
10.0.0.1/24

eth2
192.168.0.1/24

eth3
172.16.0.1/24

eth1
10.0.0.2/24

eth1
192.168.0.2/24

eth1
172.16.0.2/24

Client 1

Client 2

Client 3

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

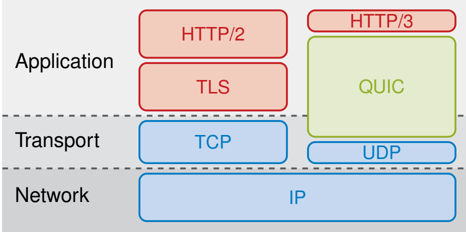
- Implement a performance benchmark
- Use pos to automate the benchmark and its evaluation
- Create an automated test report of your findings

Exercise and project

Project QUIC Interoperability Runner

- New transport protocol, originally developed by Google to replace the TCP / TLS stack
- Recently (May 2021) standardized by the IETF as RFC 9000
- Implemented on top of UDP in the user space
- More about QUIC in subsequent lectures

- There exist multiple different implementations: `https://github.com/quicwg/base-drafts/wiki/Implementations`
- QUIC Interop Runner: checks different client and server implementations for compatibility, e.g. support of certain QUIC features
- Utilizes docker as wrapper functions for each implementation



Goal of this project:

- Similar setup as with Interop Runner, but without docker
- Compare different implementations from a performance point of view (bandwidth, CPU utilization, . . .)
- Run measurements on 10G links and find performance bottlenecks

Exercise and project

Project QUIC Interoperability Runner

Step 1

- Select one implementation and survey code
- Answer some questions regarding this implementation and the RFC

Step 2

- Setup client and server application for first functionality tests
- Tests should be runnable locally

Step 3

- Implement wrapper so implementation runs in our testbed (we provide a Gitlab CI runner for compatibility testing)
- Implement support for given test cases (e.g. 0-RTT handshake, version negotiation, . . .)

Step 4

- Optimize your implementation to get the highest throughput on a 10G link

Introduction & Organization

Introduction

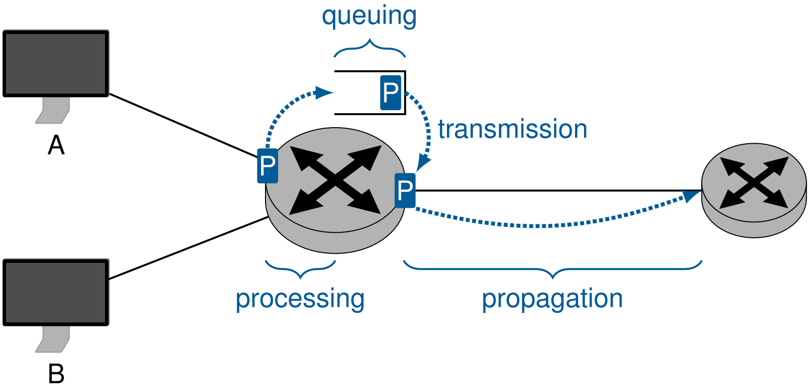
Course organization

Exercise and project

Lecture overview

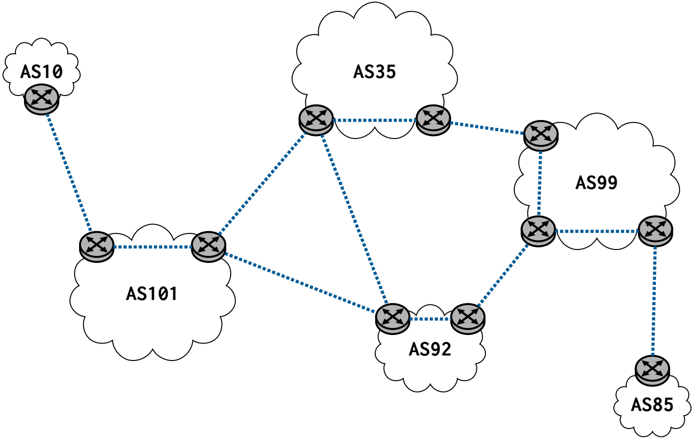
Bibliography

Lecture overview
Sources of delay

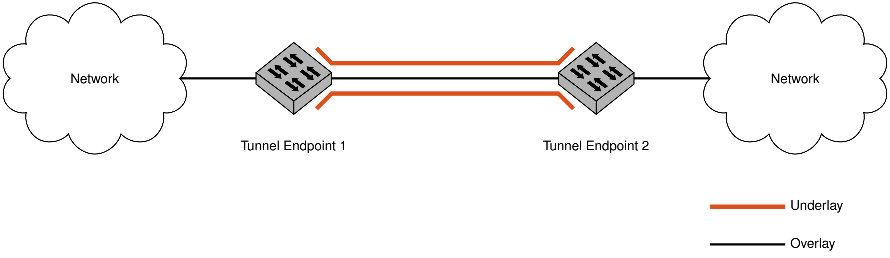


Lecture overview
Internet structure

- 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

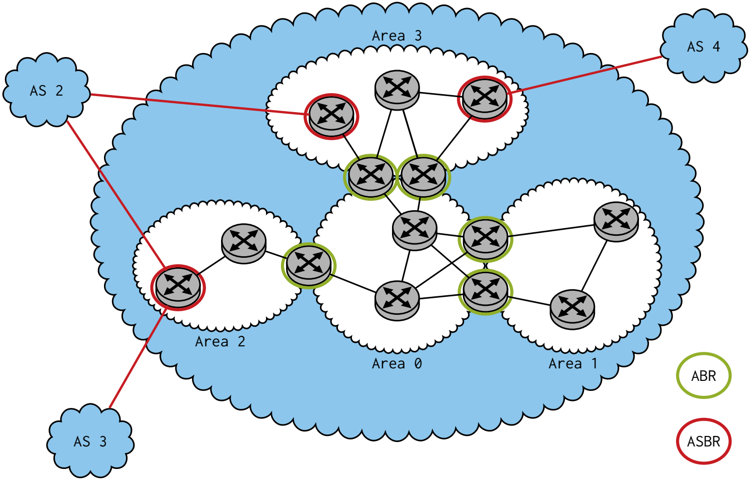
Routing algorithms

- Link state
- Distance Vector
- Hierarchical routing

Routing in the Internet

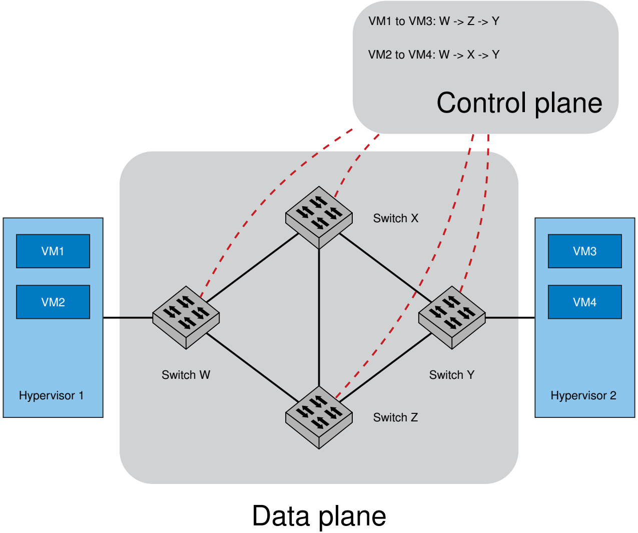
- RIP
- OSPF
- BGP

Broadcast and multicast routing

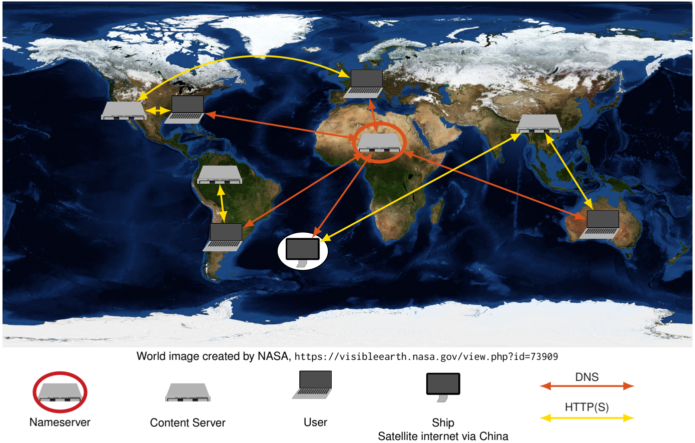


Example OSPF network

Lecture overview
SDN



Lecture overview
CDN



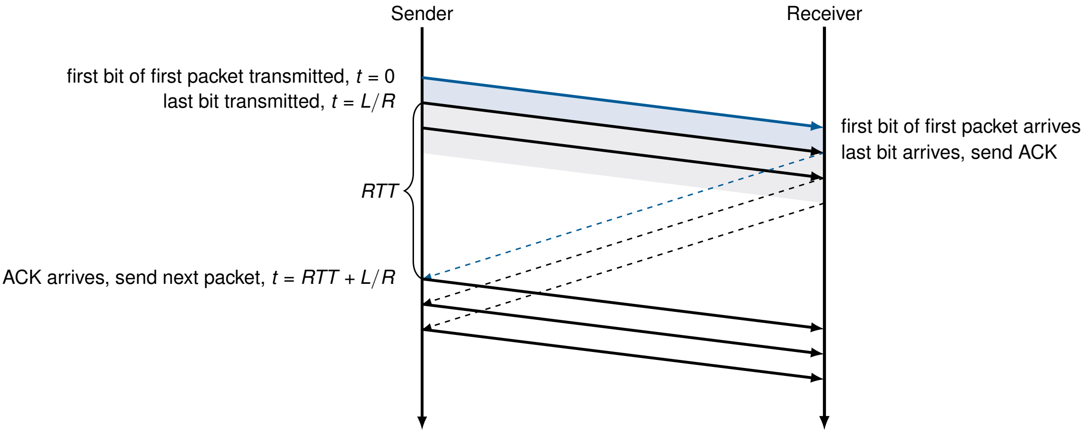
World image created by NASA, `https://visibleearth.nasa.gov/view.php?id=73909`

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

Lecture overview
Transport layer services

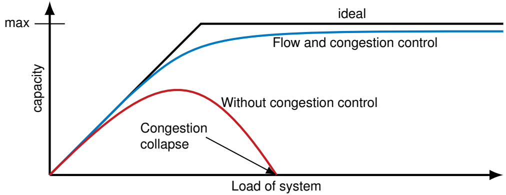
- Transport-layer services
- Multiplexing and demultiplexing
- Connectionless transport: UDP
- Connection-oriented transport: TCP
  - Segment structure
  - Reliable data transfer
  - Flow control
  - Connection management
- TCP congestion control
- QUIC

Lecture overview
Pipelining for increased utilization


Sender
Receiver
first bit of first packet transmitted, $t = 0$
last bit transmitted, $t = L/R$
first bit of first packet arrives
last bit arrives, send ACK
RTT
ACK arrives, send next packet, $t = RTT + L/R$

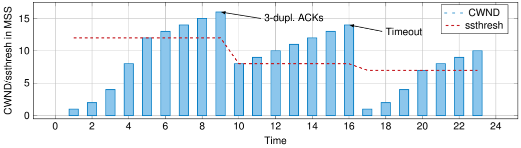
$$U_{sender} = \frac{3 \cdot L/R}{RTT + L/R}$$

Lecture overview
TCP Congestion Control

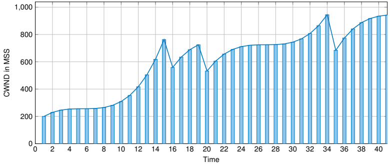


Congestion is bad...

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



TCP Reno

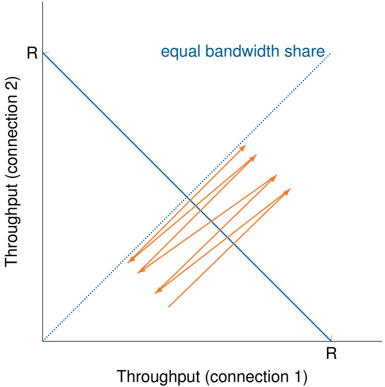


TCP Cubic

Lecture overview
Why is TCP fair?

Two competing sessions:

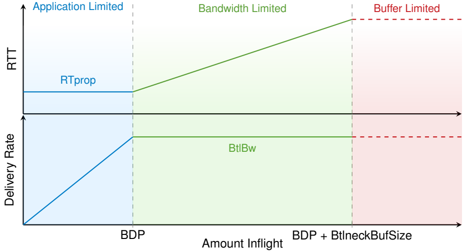
- Additive increase gives slope of 1, as throughput increases
- Multiplicative decrease decreases throughput proportionally



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

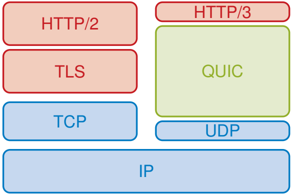
Let’s squeeze all out of it

- TCP BBR
- Newest Congestion algorithm from Google
- Gets 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

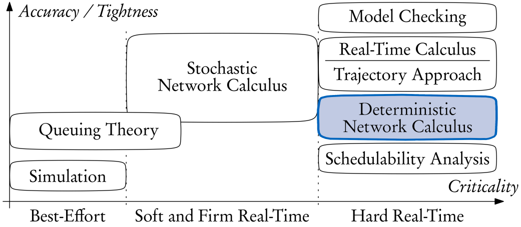
- Introduction
- Architecture & mechanisms
- Protocols
  - IPFIX (netflow accounting)
  - PSAMP (packet sampling)
- Scenarios

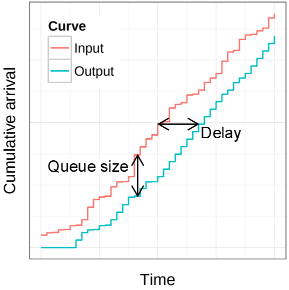
Lecture overview
Quality-of-Service and Network Calculus

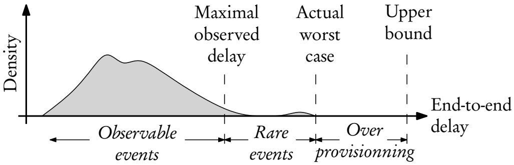
Performance guarantees can be provided at different:

- Levels (e.g. packet-level, flow-level, application-level)
- Granularities (e.g. best-effort, soft real-time, hard real-time)

⇒ Different frameworks to obtain performance guarantees. Focus on packet-level guarantees with network calculus.







Introduction & Organization

Introduction

Course organization

Exercise and project

Lecture overview

Bibliography

Introduction & Organization

[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.