

Computer Networks

WS20/21

Exercise 10

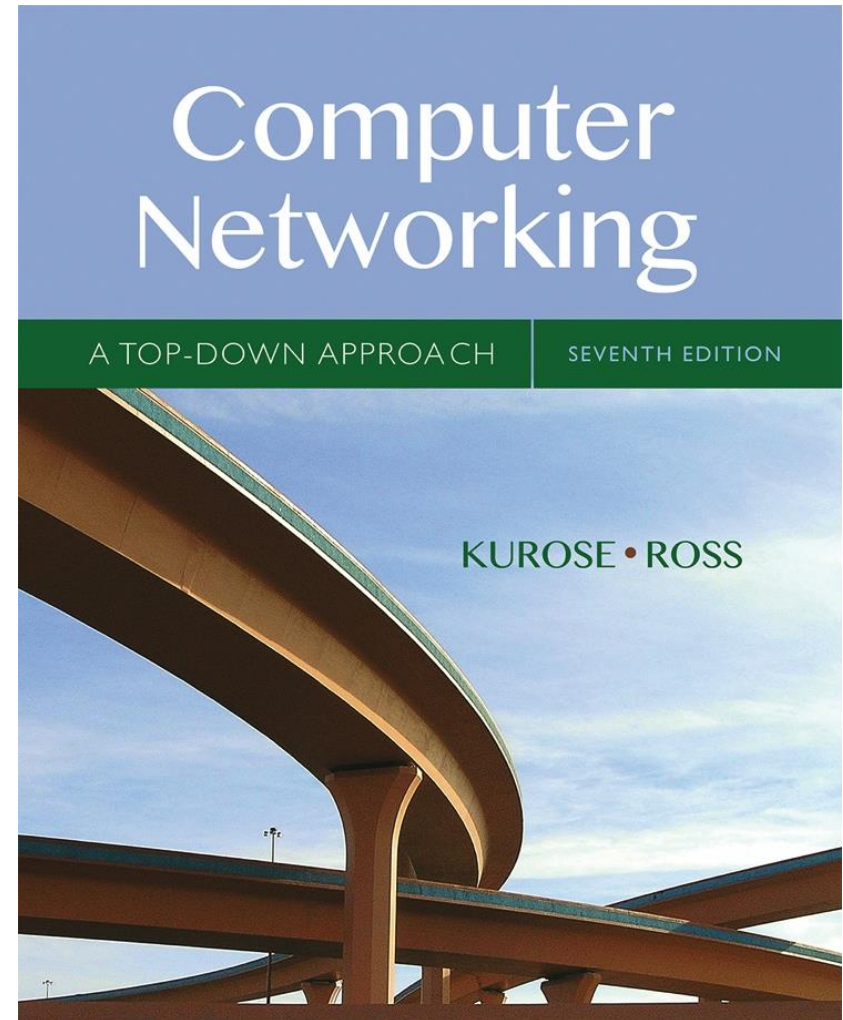
Recommendation

Try to borrow (or buy) this book:

Computer Networking: A Top Down Approach

7th edition. Jim Kurose, Keith Ross,
Pearson, 2019.

It is very good to understand!



Quality of Service Building Blocks

- Q1: Name three building blocks of Quality of Service.
- Classification: distinguish one flow/class from another
- Policing: ensure the traffic is in conform to the contracted parameters
- Scheduling: ensure the QoS guarantees (e.g., bandwidth, delay)
- Admission control and resource reservation: check if resource requirements can be met and admit or deny flows accordingly

Scheduling Policies

- Q2: Name and characterize four scheduling policies that were introduced in the lecture.
- First In First Out (FIFO)
 - Packets are sent in order of arrival
 - If queue is full, discard either
 - Newly arriving packets
 - Packets based on priority
 - Random packets

Scheduling Policies

- Q2: Name and characterize four scheduling policies that were introduced in the lecture.
- Priority Scheduling
 - Packets are sent in order of priority
- Round Robin
 - Packets are sent alternating between different classes
- Weighted Fair Queuing
 - Combination of Priority Scheduling and Round Robin
 - Weighted amount of packets sent from each class every cycle

Policing

- Q3: What are the criteria that policing mechanisms can use to control a data stream?
- (Long term) Average Rate
 - E.g. 6,000 packets per min (ppm)
- Peak Rate
 - E.g. 6,000 ppm on avg but limited to 1,500 packets per sec (pps) peak rate
- (Max.) Burst Size
 - E.g. 500 packets

Packet Scheduling

- Q4: Illustrate in what sequence packets from three different queues (a, b, c) can be sent from a sending buffer that implements a WFQ scheduling policy. Suppose the weights for the queues are 0.25, 0.25, and 0.5. How could the sequence look like if there are no packets in queue 'c'?
- abcc abcc abcc -or- acbc acbc acbc
- abab abab abab

Token Bucket

- Q5: Consider a token bucket that is filled with a rate of 1,000 tokens/min and has a size of 500 tokens. Each bit of data consumes one token. Does a flow of 20bps conform to this bucket?
- 20 bit per second = 1,200 bit per minute \rightarrow excess of 200 bit per minute
- \Rightarrow The flow does not conform to this bucket
- However: If the bucket is full, flow has 2.5 min to adjust (500 tokens / 200 tokens/min)

Delay Guarantees

- Q6: Assume a router maintains QoS guarantees for two data flows. Flow a is policed by a token bucket with a capacity of 100 tokens a fill rate of 20 token/sec. Flow b is policed by a token bucket with a capacity of 200 tokens and a fill rate of 12 token/sec. The router uses weighted fair queuing to schedule the packets with a weight of 3 for flow a and a weight of 2 for flow b. It's sending rate is 40 packets/sec. What is the upper bound delay that the router can guarantee for both flows?

Delay Guarantees

- Flow A
 - 20 tokens/sec plus 100 tokens if filled
 - Worst case: 100 tokens worth of traffic instantaneously => Queue for A fills up with 100 packets
 - Outgoing rate is 40 packets/sec of which A is allocated at least $\frac{3}{5}$ => 24 packets/sec
 - Last packet in the queue will be delivered after $100/24 = 4.17$ sec

Delay Guarantees

- Flow B
 - 12 tokens/sec plus 200 tokens if filled
 - Worst case: 200 tokens worth of traffic instantaneously => Queue for B fills up with 200 packets
 - Outgoing rate is 40 packets/sec of which A is allocated at least $\frac{2}{5}$ => 16 packets/sec
 - Last packet in the queue will be delivered after $200/16 = 12.5$ sec

IntServ and DiffServ

- Q7: Briefly compare the IntServ to the DiffServ architecture.

Criteria	DiffServ	IntServ
Based on	Aggregated service classes	Single flow
Level of control	Coarse-grained	Fine-grained
Basic principle	Traffic classification	Resource reservation
States	Edge: per flow Core: none	Every node: per flow
Setup	Per class (long-term)	Per flow
Scalability	Average	Bad

Any Questions?

Mail us:

Yachao Shao: yachao.shao@cs.uni-goettingen.de

Fabian Wölk: fabian.woelk@cs.uni-goettingen.de