FS 2019 Prof. R. Wattenhofer

Computer Engineering II

Exercise Sheet Chapter 3

We categorize questions into four different categories:

Quiz Short questions which we will solve rather interactively at the start of the exercise sessions.

Basic Improve the basic understanding of the lecture material.

Advanced Test your ability to work with the lecture content. This is the typical style of questions which appear in the exam.

Mastery Beyond the essentials, more interesting, but also more challenging. These questions are **optional**, and we do not expect you to solve such exercises during the exam.

Questions marked with $^{(g)}$ may need some research on Google.

Quiz

1 Quiz Questions

a) Can the rate $F(e)$ of a flow F on some edge e be larger than its rate F? Which additional condition for the flow F would change this assessment?

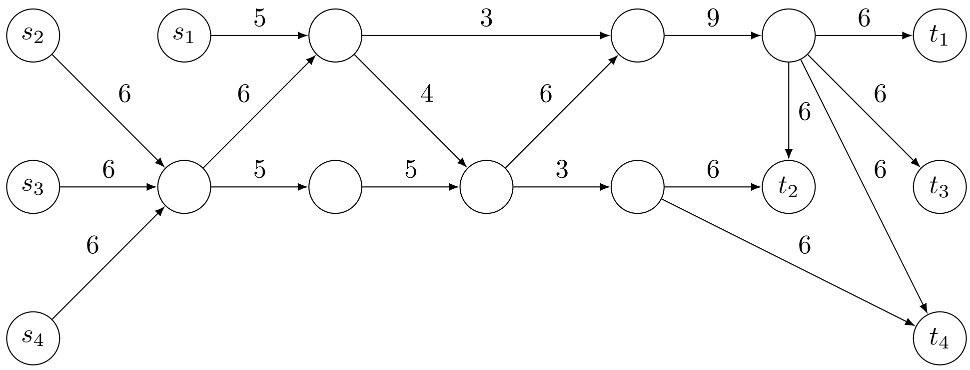
b) Is it true that in a max-min-fair allocation, the different bandwidths allocated on an edge differ at most by a factor of 2?

c) If you wanted to implement Voice over IP, would you use UDP or TCP? Why?

d) In slow-start, how long does it take to double the size of the congestion window?

2 Flows and Allocations

Consider the following graph:



a) Can you fit an unsplittable multi-commodity flow $\mathcal{F} = (F_1, F_2, F_3, F_4)$ into the given graph such that the F_i start in s_i and end in t_i and $F_1 = 1$, $F_2 = 2$, $F_3 = 3$ and $F_4 = 4$?

Assume in the following that the four flows follow the four paths found in a) but are not restricted by any demands.

b) Determine the max-min-fair allocation. What is its throughput, i.e., the sum of the rates of the four flows?

c) Can the throughput be increased if no fairness is required? What is the maximum throughput in this case?

Advanced

3 UDP and TCP

a) Imagine that both UDP and TCP packets arrive at a router, but its buffers cannot accomodate all of them since they are already pretty full. Should the router rather drop UDP packets or TCP packets? What arguments can you find for either approach?

b) In TCP, how does a sender establish if some router dropped one of its packets? What would you gain if a router dropping packets informed the affected senders? Why, do you think, is this not done in practice?

Let's have a look at what may happen at a router in TCP. Assume that the data of three clients C_1, C_2 and C_3 is routed through the router R. From C_1, a packet arrives every 2 ms at R, from C_2 every 3 ms and from C_3 every 4 ms. For simplicity, assume that the rates stay the same until a client notices that one of its packets was dropped upon which he will decrease its rate by a factor of 1/2. At time $t = 0$, the first packet from each of the three clients arrives at R.

Our router forwards the packets over its only outgoing link with a rate of 1 packet/ms. Assume that a packet arriving at R at time t can also leave the router at time t. In particular, the first packet is forwarded at time $t = 0$. The buffers of R can accomodate 10 packets. If the buffers are full and a single packet arrives at time t, then the packet will be buffered since another packet leaves the buffer at time t. Any packets that cannot be buffered will be dropped.

If several packets arrive at the same time and R has to drop some of them, it prefers dropping packets from C_1 over dropping packets from C_2 over dropping packets from C_3 (but it will only drop arriving packets, not buffered ones).

c) At what point in time does R drop the first packet? Which client is affected?

d) Since the client affected in c) and our router R happen to be on different continents, it takes a considerable time until the client realizes that it lost a packet. How fast does the client affected in c) have to decrease its rate so that the other two clients get lucky and do not lose any packets due to congestion?

e) In general, should a router rather drop packets from a far-away client or from a close-by client? And should it rather drop packets from a client sending with a large rate or from one sending with a small rate?

4 LPs

$^{(g)}$**a)** Consider the following LP:

Maximize $f(\mathbf{x}) = x_1 + 2x_2$
subject to

(a) $x_1 \geq 0$

(b) $x_1 \leq 2$

(c) $x_2 \geq 0$

(d) $x_2 \leq 2$

(e) $x_1 + x_2 \leq 3$

Simulate the simplex algorithm starting at vertex $(0, 0)$. What are the vertices of the polytope? How many steps does the simplex algorithm take? What is the solution to the LP?

b) Design an LP for maximizing a multi-commodity flow (for given commodities), i.e., for maximizing the sum of the rates of the contained flows.

c) Assume you are given a weighted directed graph (G, c) and demands $d_i, 1 \leq i \leq k$ and you want to find out if you could fit a multi-commodity flow $\mathcal{F} = (F_1, ..., F_k)$ into your graph without violating any capacity constraints and such that the F_i have rate d_i. Design an LP that determines if such a multi-commodity flow exists. How do you infer a YES or NO answer to your question from a solution to the LP?

5 Fairness and Congestion Control

The figure below shows a network where four source nodes (S1, S2, S3, S4) are sending four flows of data. The capacity of the edges are given in packets per milisecond, that is, the maximum number of packets that can be sent through a given edge per milisecond.

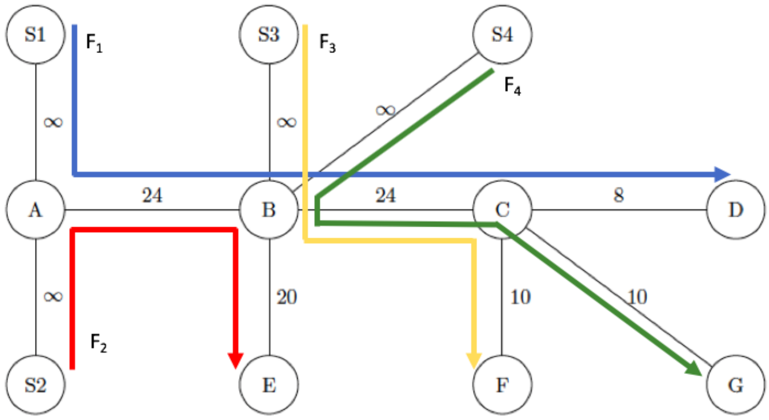


Figure 1: Network with four flows.

a) Compute the max-min-fair allocation of the four flows, we only allow integer values for the flows.

Proportional fairness is another strategy for fair flow allocation. In proportional fairness we want to maximize the sum of the logarithm of the flows, i.e:
$\max_{\{F_i \forall i\}} \sum_{i=1}^{N} log(F_i)$.

b) Compute the proportionally fair flow allocation for the four flows in the figure, we only allow integer values for the flows. Hint: although the function to maximize is not linear, Linear Programming (LP) can be used to solve a broader type of problems. In particular, this problem can be formulated and solved in an LP-manner, where the constraints are linear and the target function is not.

Now, S3 and S4 stop transmitting and so, only the flows F_1 and F_2 are active. At $t = 0$ the value of each flow is $F_1 = 3$ packets/ms and $F_2 = 19$ packets/ms. From this point, the system evolves using the congestion control mechanism AIMD (additive increase/multiplicative decrease). The round-trip-time (RTT), i.e., the time between the moment when a packet is sent and an ACK is received is 1 ms for both flows. In our setting, when the capacity of a flow is exceeded, only the exceeding packets are dropped. If an even number of packets have to be dropped, the same amount of packets from each flow are dropped. If the number is odd, one more packet from F_1 than from F_2 is dropped. The timeout for both flows is also 1 ms; thus, if a packet is dropped the source of the flow will notice after 1 ms. Only integer values are allowed for the flows, so if multiplicate decrease occurs, the ceil of the half is taken.

c) Draw and explain the evolution of the flows and the traffic in the edge A-B between $t = 0$ and $t = 10ms$: the horizontal axis should represent the time and the vertical axis the number of packets. You should draw three plots, one for each flow and one for the edge A-B.

d) Assume that we change the AIMD mechanism and instead of additive increase, we double the number of packets sent each time an ACK is received. Give the pros and cons of this modification.

Mastery

6 Fairness vs. Efficiency

Let $\mathcal{F} = (F_1, ..., F_k)$ be a set of unsplittable flows specifying a bandwidth allocation. Define the *throughput* $T(\mathcal{F})$ of $\mathcal{F}$ as the sum of the rates of the flows in $\mathcal{F}$. Moreover, define the *efficiency* of the allocation $\mathcal{F}$ as the ratio of $T(\mathcal{F})$ and the maximal throughput achievable by changing the rates of the flows (but not their paths) without violating any capacity constraints. Assume that the rates of the flows are not restricted by demands.

How inefficient can a max-min-fair allocation be? Prove a lower bound (the larger the better) for the efficiency of a max-min-fair allocation, i.e., find a weighted directed graph and flows in the graph such that the efficiency of the max-min-fair allocation is as small as possible.