More on Principles of Congestion Control

Let's look at a couple of more key principles that congestion control algorithms adhere to!

What we know so far is that congestion control schemes **must avoid congestion**. In practice, this means that the bottleneck link (the link with the lowest bandwidth) cannot be overloaded. This means on average, the sum of the transmission rate allocated to all hosts at any given time should be less than or equal to the bottleneck link's bandwidth.

Additionally, the congestion control scheme must be **efficient**. The bottleneck link is usually both a shared and an expensive resource. Usually, bottleneck links are wide-area links that are much more expensive to upgrade than the local area networks. The congestion control scheme should, therefore, ensure that such links are **efficiently used**. Mathematically, the control scheme should ensure that the sum of the transmission rate allocated to all hosts at any given time should be approximately equal to the bottleneck link's bandwidth.

Max-min Fairness

Furthermore, the congestion control scheme should be **fair**. Most congestion schemes aim at achieving **max-min fairness**. An allocation of transmission rates to sources is said to be **max-min fair** if:

1. No link in the network is congested
2. The rate allocated to a source j cannot be increased without decreasing the rate allocated to another source i, whose allocation is smaller than the rate allocated to the source j,

In other words, this principle postulates that **increasing the transmission rate of one end-system necessarily decreases the transmission rate allocated to another end-system with an equal or smaller allocation**.

To visualize the different rate allocations, it's useful to consider the graph shown below. Consider hosts A and B that share a bottleneck link.

Possible allocated transmission rates
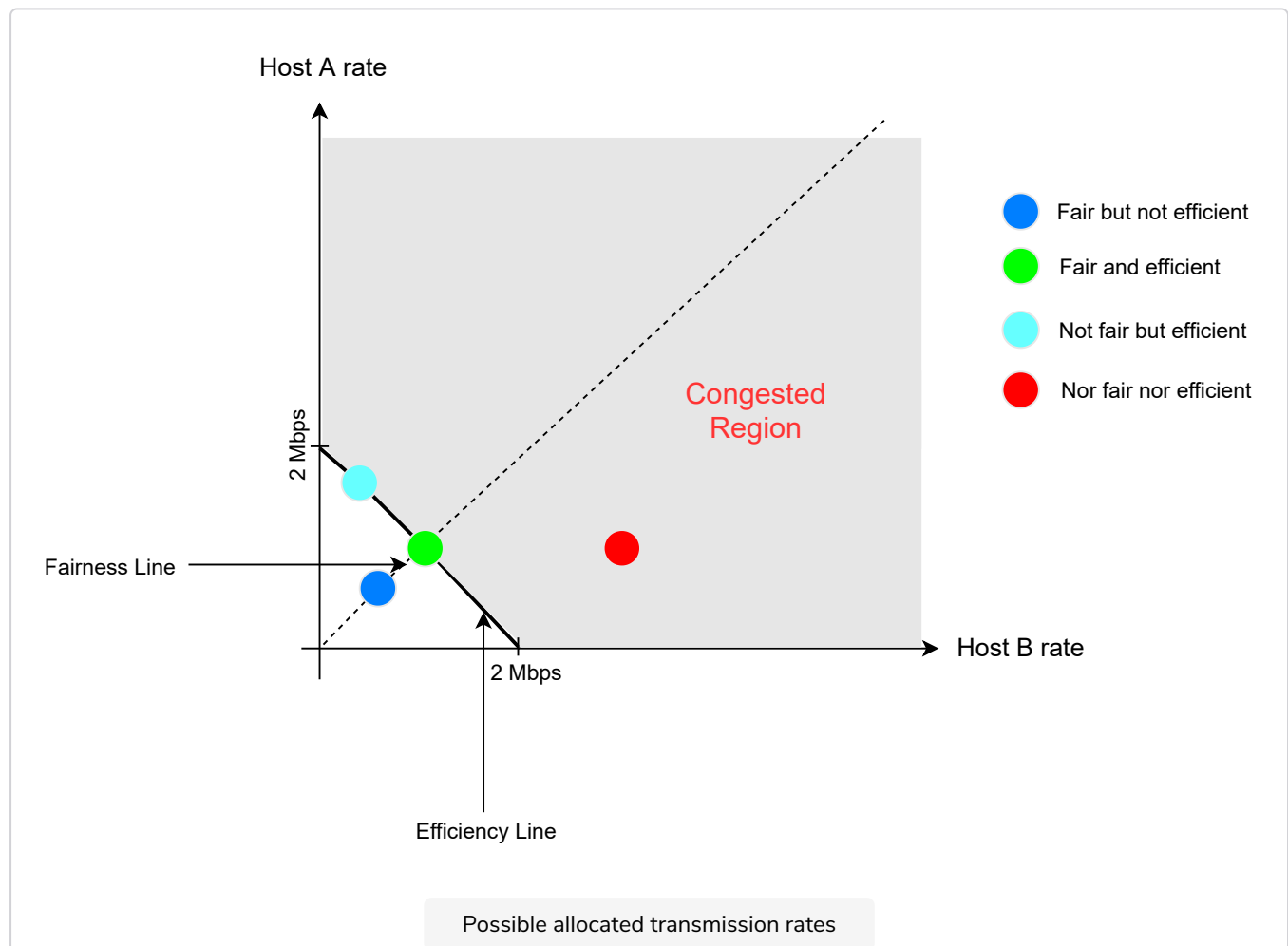
In this graph, we plot the **rate allocated to host B** on the x-axis and we plot the **rate allocated to host A** on the **y-axis** . A point in the graph (r_B, r_A) corresponds to a possible allocation of the transmission rates. Since there is a 2 Mbps bottleneck link in this network, the graph can be **divided into regions**:

1. The white-colored part of the graph contains all allocations (r_B, r_A) such that the bottleneck link is not congested $(r_A + r_B < 2)$.

2. The right border of this region is the efficiency line or the set of allocations that completely utilize the bottleneck link $(r_A + r_B = 2)$.

3. Finally, the fairness line is the set of fair allocations.

Depending on the network, a max-min fair allocation may not always exist. In practice, max-min fairness is an ideal objective that cannot necessarily be achieved. When there is a single bottleneck link as in the example above, max-min fairness implies that each source should be allocated the same transmission rate.

Ideally, the allocation should be both fair and efficient. Unfortunately, maintaining such an allocation with fluctuations in the number of flows that use the network is a challenging problem. Furthermore, there might be several thousands of TCP connections or more that pass through the same link.

Convergence

Additionally, bandwidth should be allocated such that it converges to a fair (a host does not hog all of it), and efficient value. This means it should not oscillate and most of it will be used. Furthermore, it should also change in response to changes in the demands of the network over time. Here are some slides that demonstrate this convergence.
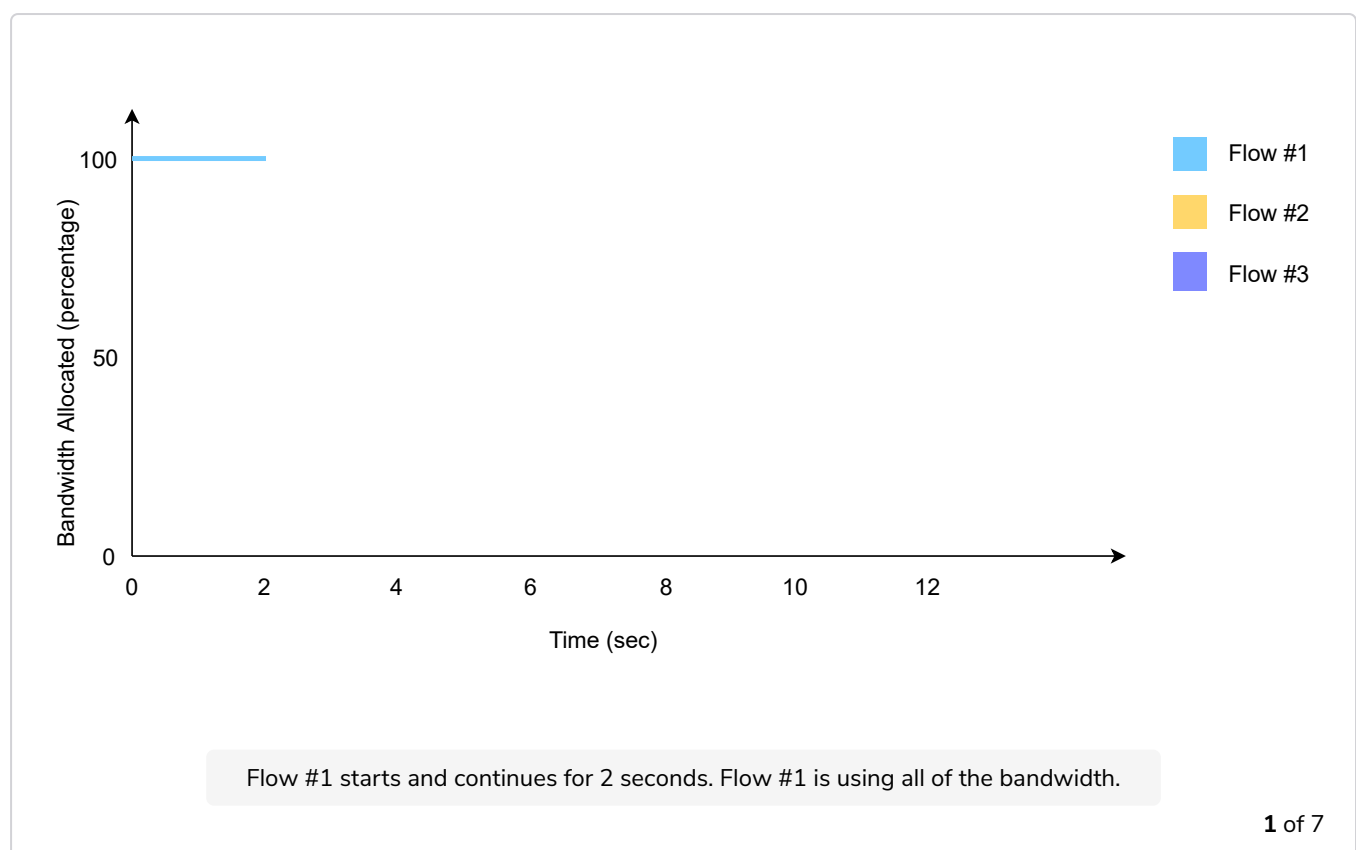
Flow #1 starts and continues for 2 seconds. Flow #1 is using all of the bandwidth.

Quick Quiz!

1	In the context of congestion control, an efficient rate allocation is desirable because _____.

○ A) Regulators mandate it

○ B) It uses network resources efficiently

○ C) It's also fair

COMPLETED 0%   1 of 3   ‹   ›

We now understand some basics of congestion control and bandwidth allocation. Now that we have an in-depth understanding of how UDP works, let's move on to perhaps one of the most important protocols of this entire course: TCP.