Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Prof. Laurent Vanbever
Networked Systems Group

# Ensuring Transport Fairness with Smart Networks

Semester thesis proposal

One of the fundamental challenges of communication networks is fairness. More precisely: the fair allocation of network resources to communication flows. Currently, the network itself is mostly treated as a dumb medium which provides all its resources first-come-first-serve, and it is up to the hosts to ensure fairness. Consequently, hosts must not pick more than their fair share of the available network resources, which is challenging. Below, we briefly argue why we believe that the network should be ensuring fairness, not the hosts, and how we can leverage programmable network device for this.

Classical transport protocols like TCP try to determine their fair share indirectly, which often fails in complex network environments. The protocols do not have any knowledge of other flows in the network, and instead listen to network signals like delay and packet loss to determine how much and how fast they can send. For example, TCP increases the sending rate of flows over time until packets are dropped (indicating resource exhaustion), at which point it reduces the sending rate. The concrete changes in sending rate are carefully designed such that multiple concurrent flows converge to a fair steady state. If all hosts in the network were to use the same protocol, this would be feasible. But modern networks carry traffic from many different variants of TCP alone, among other protocols.

We argue that this is a fundamental design flaw that unnecessarily slows down the design of new and improved transport protocols. On the one hand, having to consider 'being fair' to other protocols puts an unnecessary burden on protocol designers, and this will only get worse in the future, with more and more protocols emerging. On the other hand, it makes adoption of new protocols difficult, as they might not get their fair share from existing protocols.

Instead of hosts trying to figure out their fair share, we envision *smart* networks to take over this task: Like a scheduler in an operating system, the network can supervise resource allocation, and with that, ensure fairness. In network devices, resource allocation revolves nearly exclusively around the device buffer, which is typically a set of queues. The device can control the in- and output to these queues, e.g. how many packets per second to enqueue and dequeue, as well as manage the queues itself, e.g. using some queues with higher priority, or dropping packets when the queue grows to long. Consequently, ensuring a fair resource allocation means designing an algorithm that intelligently manages the device buffer. Programmable network devices offer the required level of control to monitor and control the buffer.

Our goal is to propose a buffer management algorithm that ensures fairness between different transport protocols. The required work may be separated into the following milestones:

1. Review literature for transport protocols and fair resource sharing.
   - Understand different classes of congestion control algorithms, e.g. different TCP variants [4, 6] or modern alternatives such as BBR [2]; in particular which signals they use, and how they react to them.
   - Choose a suitable fairness metric and understand which scenarios result in unfairness between different classes of congestion control algorithms [1, 3, 5].
2. Setting up a network simulation using ns-3 to reliable reproduce an unfair scenario.
3. Investigate how the observed unfairness is reflected in the device buffer.
4. Propose a buffer management algorithm to ensure fairness.
5. Test the proposed algorithm, potentially on real hardware.

#### Requirements

- Some familiarity with communication networks, in particular with the basics of congestion control.
- Background in game theory and/or control.
- Skills in programming, simulation and data analysis.

#### Contact


- Alexander Dietmüller, adietmue@ethz.ch
- Maria Apostolaki, apmaria@ethz.ch
- Prof. Dr. Laurent Vanbever, lvanbever@ethz.ch


#### References


[1] M. Bachl, J. Fabini, and T. Zseby. Cocoa: Congestion control aware queuing. *arXiv preprint arXiv:1910.10604*, 2019.

[2] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and Van Jacobson. BBR: congestion-based congestion control. *Communications of the ACM*, 60(2):58–66, Jan. 2017.

[3] F. Fejes, G. Gombos, S. Laki, and S. Nádas. Who will save the internet from the congestion control revolution?

[4] S. Ha, I. Rhee, and L. Xu. CUBIC: a new TCP-friendly high-speed TCP variant. *ACM SIGOPS Operating Systems Review*, 42(5):64–74, July 2008.

[5] T. Lynn, N. Hanford, and D. Ghosal. Impact of buffer size on a congestion control algorithm based on model predictive control.

[6] J. Mo, R. La, V. Anantharam, and J. Walrand. Analysis and comparison of TCP Reno and Vegas. In *IEEE INFOCOM '99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No.99CH36320)*, volume 3, pages 1556–1563 vol.3, Mar. 1999. ISSN: 0743-166X.