TCPTokens: Introducing Currency into Data Center Congestion Control

Anand Jayarajan
University of British Columbia
anandj@cs.ubc.ca

Michael Przystupa
University of British Columbia
michael.przystupa@gmail.com

Robert Reiss
University of British Columbia
rreiss@cs.ubc.ca

Fabian Ruffy
University of British Columbia
fruffy@cs.ubc.ca

ABSTRACT

KEYWORDS

TCP, Congestion Control, SDN, Data center

1 INTRODUCTION

In the early years, research in network congestion has been dominated by the traditional assumption of a decentralised, autonomous network. End-hosts only have control over the amount of traffic they send, and are unaware of the intentions or traffic rate of their peers. Similarly, switches and routers are unaware of global traffic patterns and only forward based on their local notion of optimality.

In line of these assumptions, TCP has been designed to optimize traffic globally on a simple fairness principle. Nodes react to packet loss and assume that others will behave accordingly. An equilibrium is reached when all end-hosts achieved a traffic rate that, in sum, conforms to the maximum available bandwidth of the congested link. TCP works excellently in scenarios where many different distrustful participants compete for limited bandwidth. Still, TCP is a reactive protocol. The fact that packet loss and latency increase occur in the network, already indicates a problem. Packet loss is largely created by an overflow of queues in forwarding elements, implying that traffic has not been optimally distributed.

Ideally, a network should always be "zero-queue", i.e., latency will merely be induced by propagation, and not queuing delay. Queueing has generally not been a dominant issue in wide-area and enterprise networks, as traffic is sufficiently distributed and diverse, with only few "hot" target hosts. Optimal traffic optimization is a substantial challenge, if not impossible, as network operators have no control over the individual network elements and its participants. Under these conditions TCP and its extensions can be considered a best-effort solution.

However, new developments in the past decade have changed the general networking environment. Datacenters have emerged as a guiding force of research, posing new design challenges as well as opportunities. Driven by minimization of costs and maximization of compute power they are intended to run at maximum utilization to achieve an optimal compute/cost ratio. In such a scenario, inefficient routing can quickly lead to bufferbloat and the eventual collapse of a high-load network, requiring more sophisticated approaches to solve congestion control. On the other hand, operators now have the ability to freely control and adapt their network architecture. leading highly customized systems and fine-grained control.

Much in line with the trend of datacenters, Software-Defined Networking (SDN) emerged as a new networking paradigm. Moving away from the principle of distributed communication and routing, SDN introduced the notion of "centralized management". A single controller with global knowledge is able to automatically modify and adapt the forwarding tables of all switches in the network, as well as notify end hosts of changes in the network. These two new trends in systems facilitated impactful new innovation opportunities in the space of TCP congestion research. Traffic can now be managed centralised based on global knowledge of the entire topology and traffic patterns. In recent years, a new line of centralised schedulers has emerged, which make use of these advantages and achieve close to optimal bandwidth optimization.CITATIONS However, these schedulers are still reactive in their nature. The central controller responds to changes in the network or requests by applications, which may cost valuable roundtrip latency. Often, short-term flows or bursts are unaccounted for, which causes undesirable packet loss and backpropagating congestion.

A much more desirable solution is a global, centralised arbiter which is able to predict and fairly distribute flows in the network before bursts or congestion occurs. By treating the networkâĂŹs compute and forwarding power as a single finite resource, a controller could act like the OS scheduler distributing CPU time slices to processes. This design approach also follows SDNâĂŹs aspiration of introducing operating systems abstractions to the networking domain space.

In this project, we plan to explore the possibilities of a centralised, proactive flow scheduler. We ask ourselves the following research questions:

(1) Is it possible to design a centralised token-based scheduling network?
(2) Is it possible to predict traffic and preemptively schedule flows and token distribution in a datacenter context?
(3) Using this approach, are we able to achieve better performance and utilization than existing solutions?

In the scope of this course, we attempt to answer question 1 and design a simple token-based scheduler in Mininet. If we succeed, we will benchmark our results and evaluate the level of utilization compared to contemporary scheduling systems.

2 RELATED WORK

3 DESIGN

4 IMPLEMENTATION

5 EVALUATION