

Improving TCP performance with
bufferless token bucket policing:
A TCP friendly policer



Ronald van Haalen

Richa Malhotra

Alcatel-Lucent, Bell Labs Europe

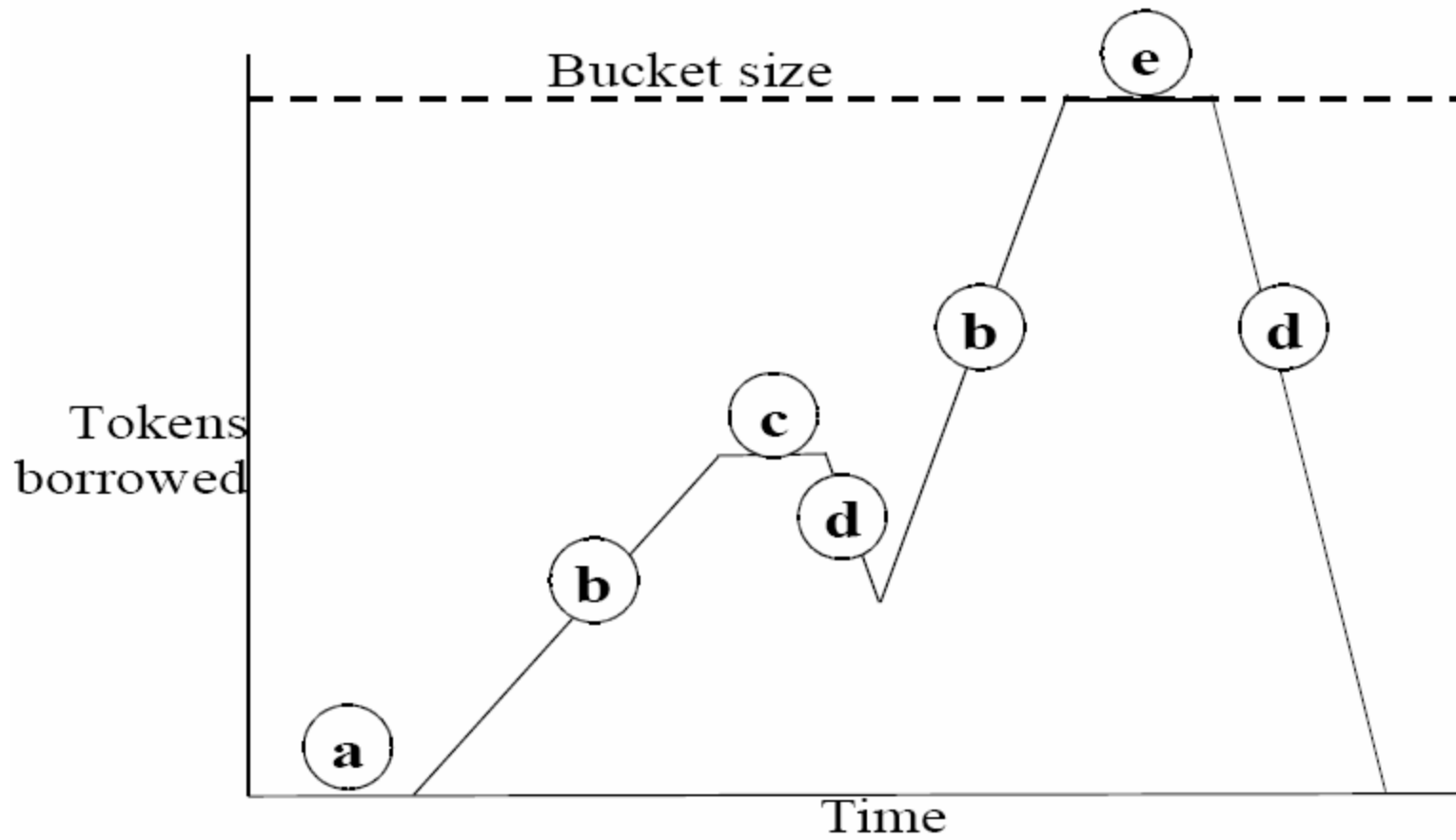
Overview

- Introduction
- Token bucket policer
- Considered scenarios
- TCP performance with a token bucket
- Large bucket size
- Dynamic bucket size policer
- Simulation results
- Conclusions

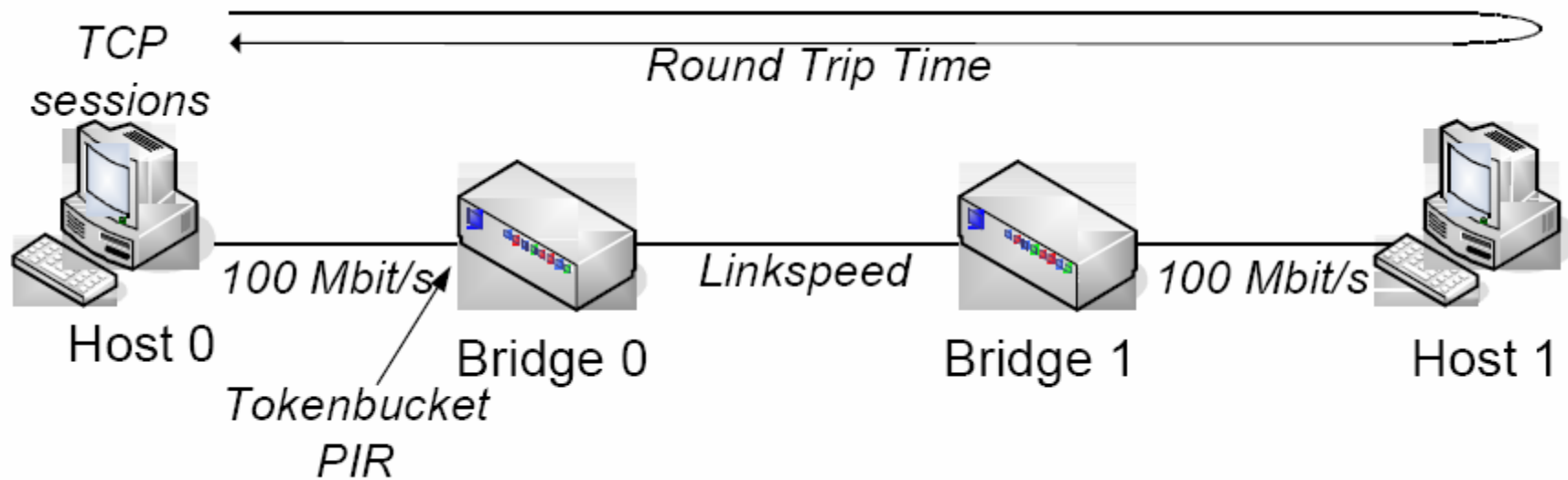
Introduction

- Policing used to enforce a traffic contract
- A buffer-less token bucket is most commonly used, but can lead to low traffic rates when interacting with TCP
- Solutions exists, but:
 - Are strongly dependent on the traffic offered, e.g. level of aggregation, Round Trip Times (RTTs), or
 - Proposes changes to TCP
- We propose a *dynamic* bucket size scheme, which monitors TCP reaction to drops and determines minimum optimal bucket size

Token bucket policer

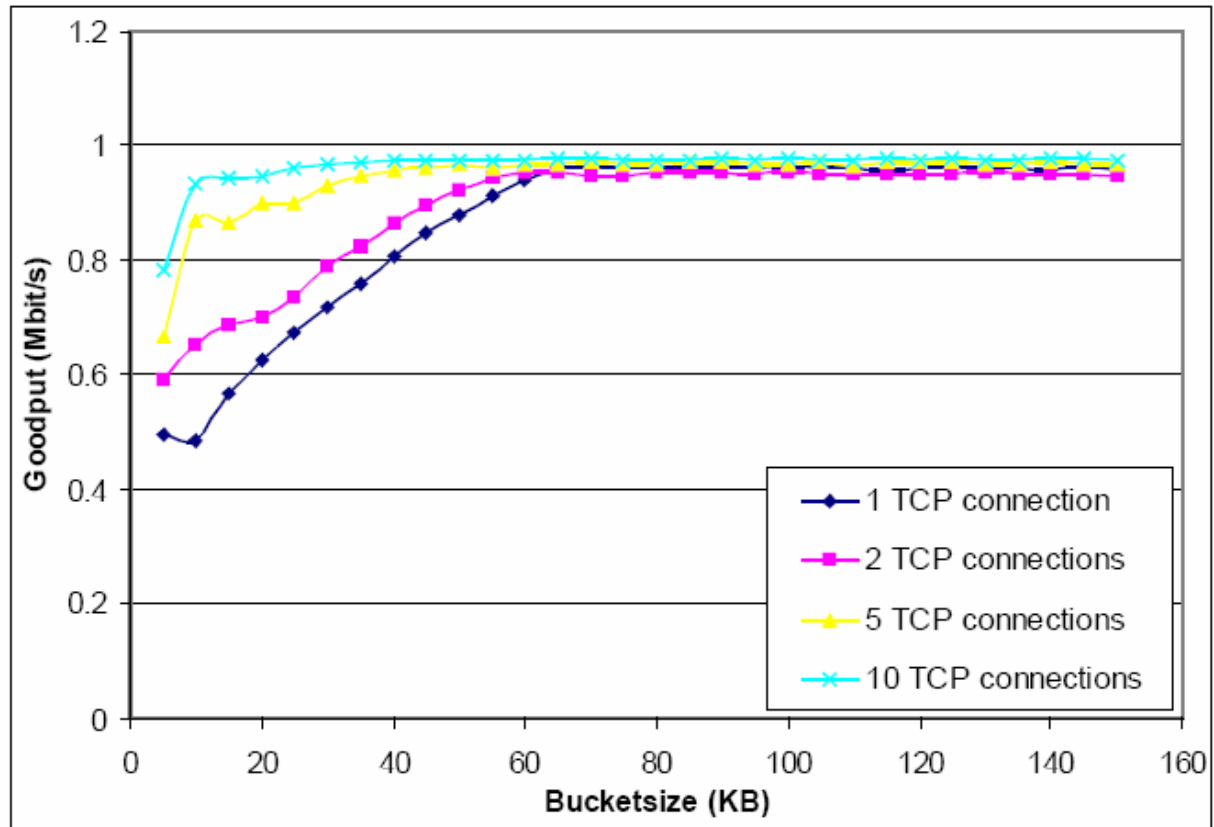


Considered scenarios



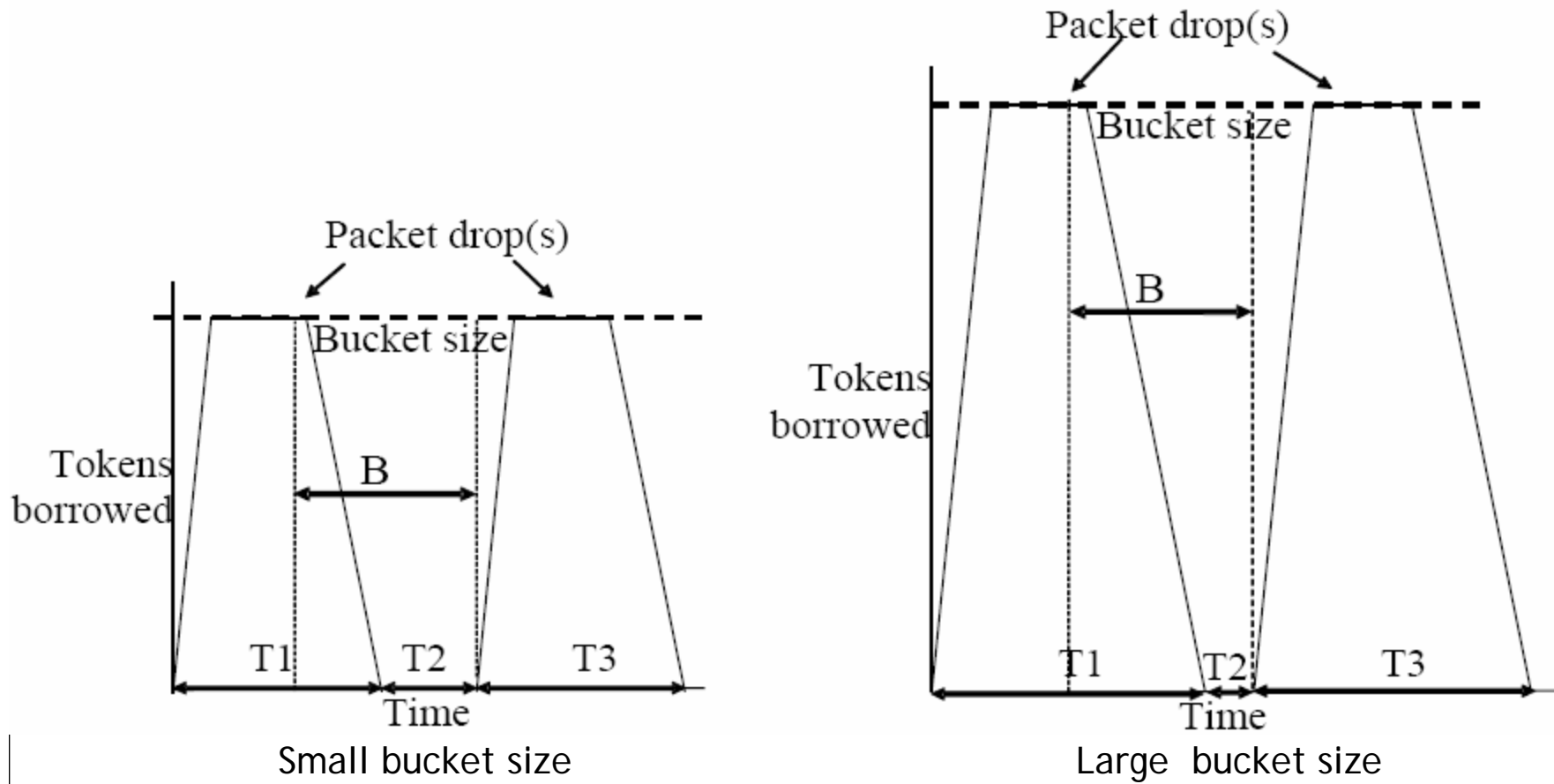
- RTT: 10, 50 and 100 ms
- WAN linkspeed: 2, 10 and 50 Mbit/s
- PIR values: 0.5, 1, 2, 5 and 10 Mbit/s
- Number of TCP connection: 1, 2, 5 and 10

TCP performance with a token bucket



Bucket size versus TCP goodput. Link speed: 10 Mbit/s, RTT: 50 ms, PIR 1 Mbit/s

TCP performance with a token bucket

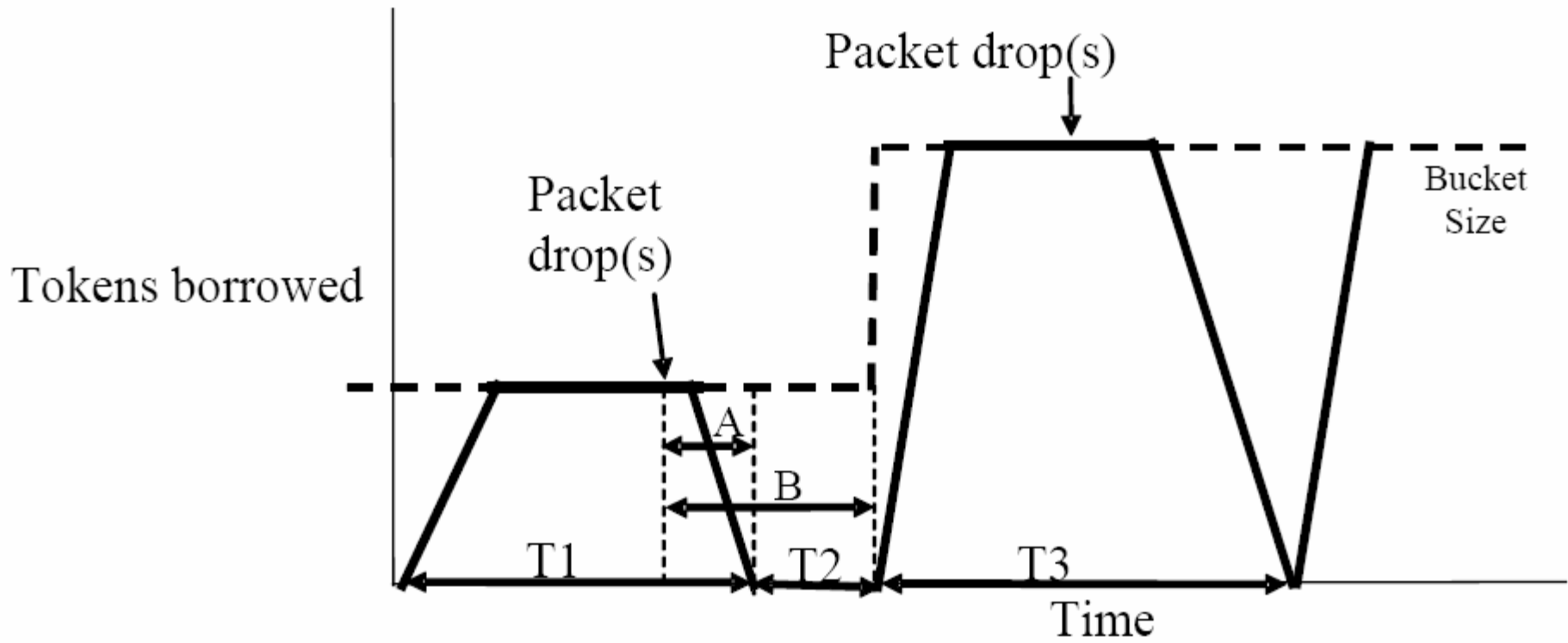


Large bucket size

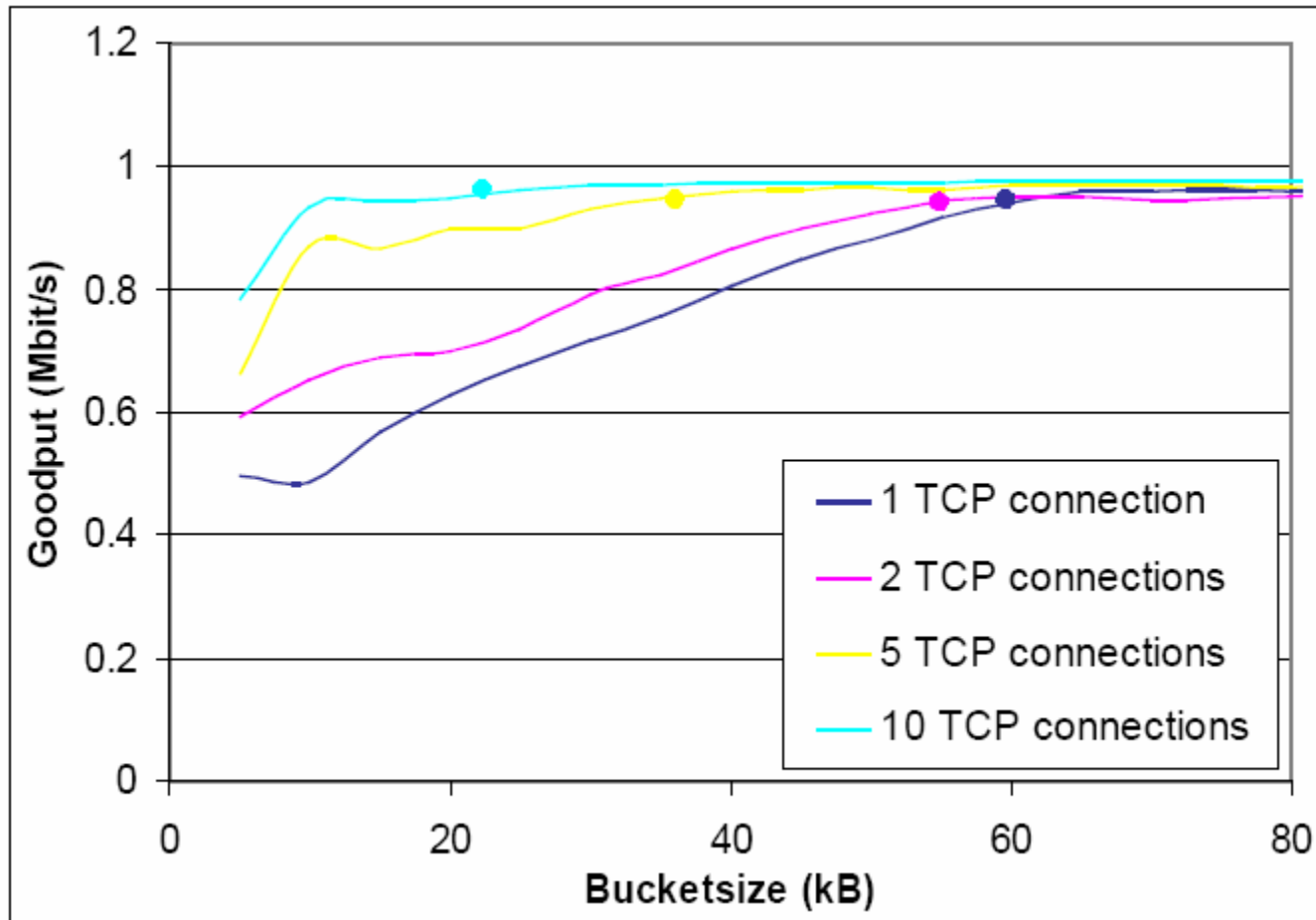
A static, large bucket size will solve the problem, but causes the following problems:

- Traffic Engineering
- Temporary congestion
- Increased drops/latency

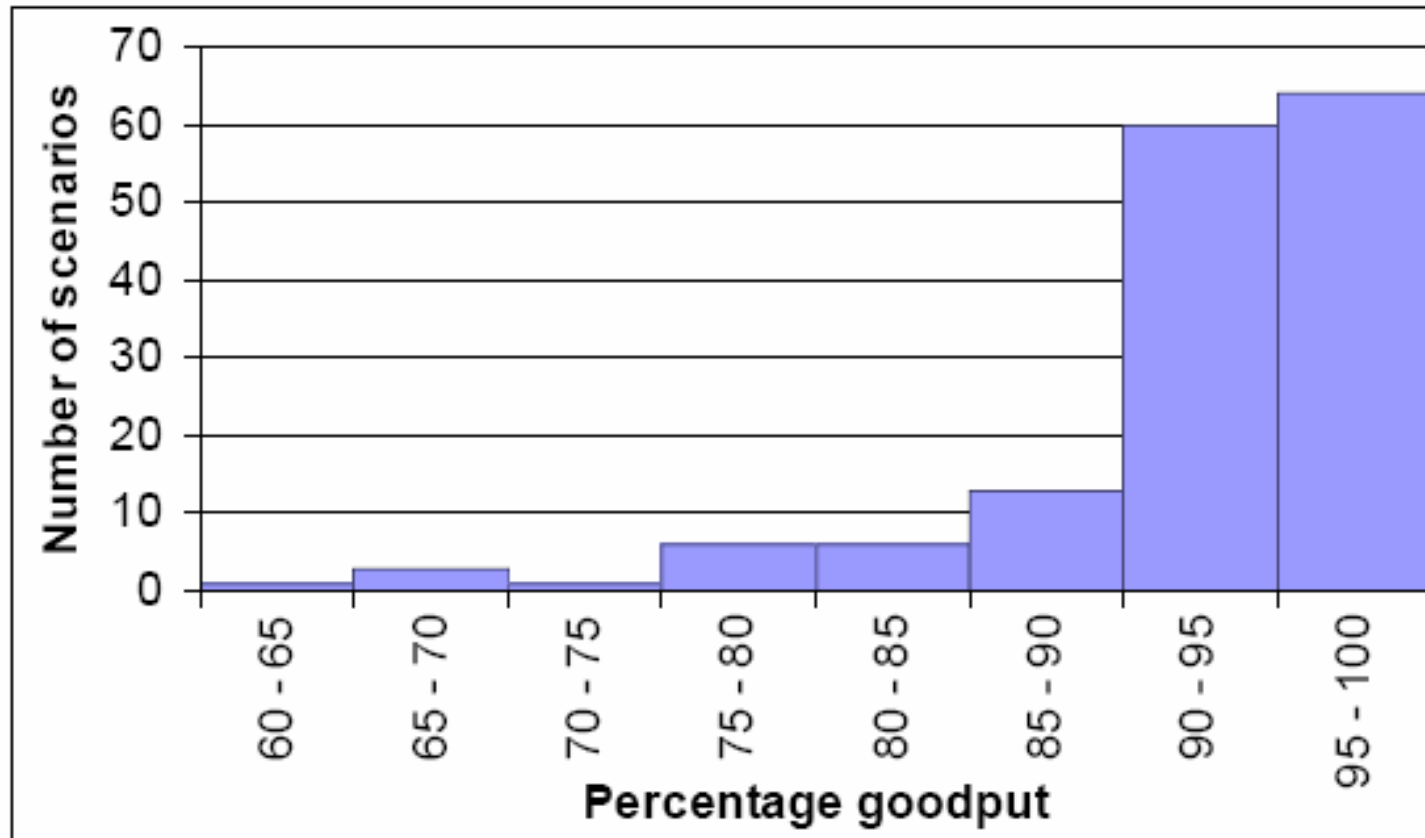
Dyamic bucket size policer



Dynamic bucket size policer results



Simulation results



Average good put: 92%

Conclusions and future work

Conclusions:

- TCP performance can degrade significantly when policed
- Very large bucket size prevents the problem, but causes new problems (e.g. loss, latency)
- Dynamic bucket size solution gives 92% average good put, while minimizing large bucket size problems

Future work:

- A loose coupling between monitoring time and RTT will improve results
- More dynamic traffic profiles and aggregation

Questions?

Questions?