

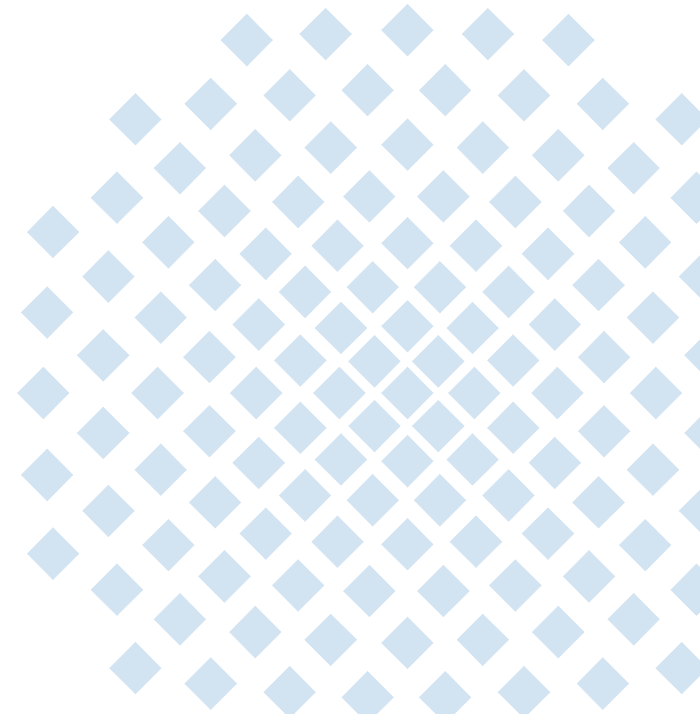
ConEx Implementation & Policing

ConEx – 87. IETF Berlin – July 29, 2013

David Wagner <david.wagner@ikr.uni-stuttgart.de>

Mirja Kühlewind <mirja.kuehlewind@ikr.uni-stuttgart.de>

Sebastian Dörner



Overview

ConEx Implementation & Usage

Use case: ConEx Policing of Reno-Traffic

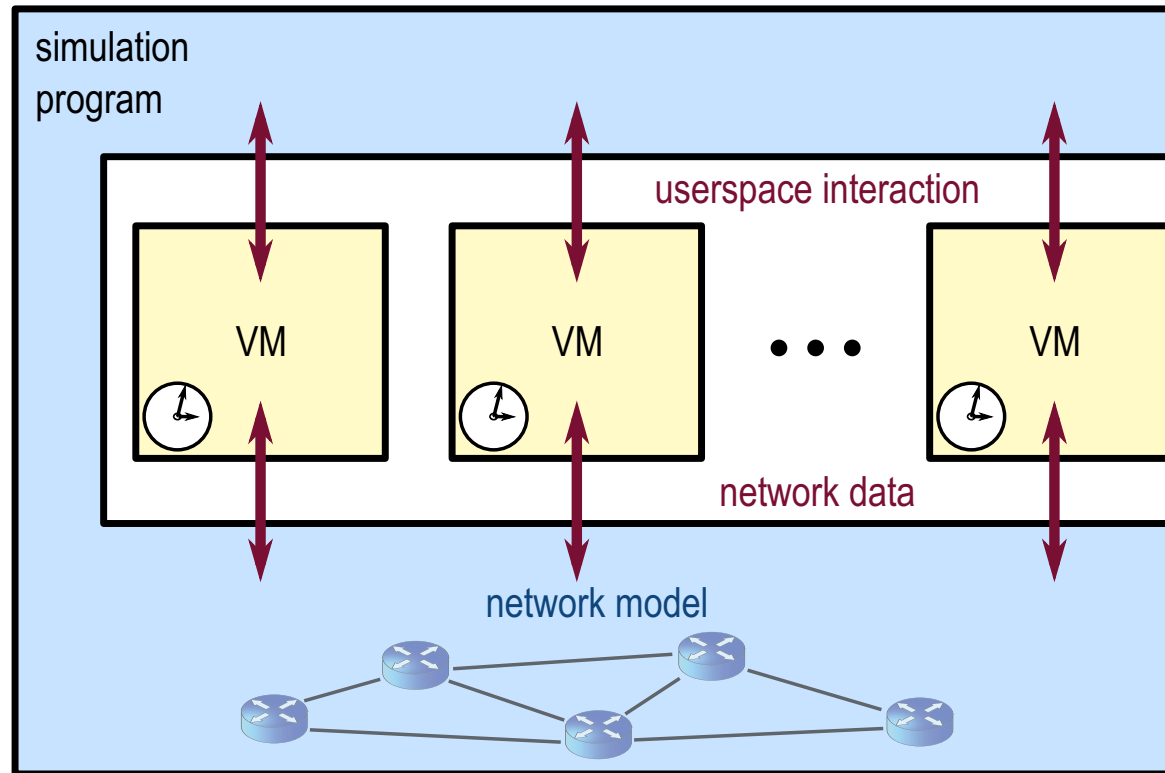
Implementation

Patched Linux 3.5.4

- Implemented draft-ietf-conex-tcp-modifications-04 & draft-ietf-conex-destopt-04
 - Different sending of credits: send credit for max. cwnd
 - No detection of audit false positives
- Added "accurate 2-state ECN-echo mechanism" from DCTCP to support accurate non-loss congestion signals (assuming no loss on return path)

How we use it

IKR Simlib: event-based simulations integrating real kernels



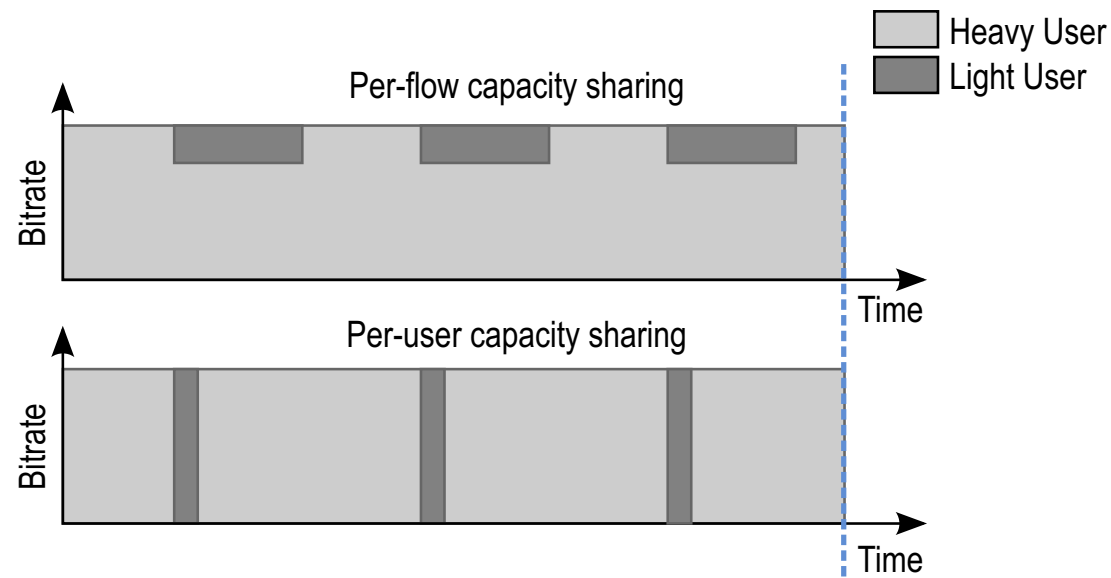
→ for more details see IKR SimLib-QEMU presentation in ICCRG on Wednesday

Overview on Policing Use Case

Use Case

Long-term per-user capacity sharing in the Internet

→ Initial evaluation on ConEx policing in extreme simple scenario with two users

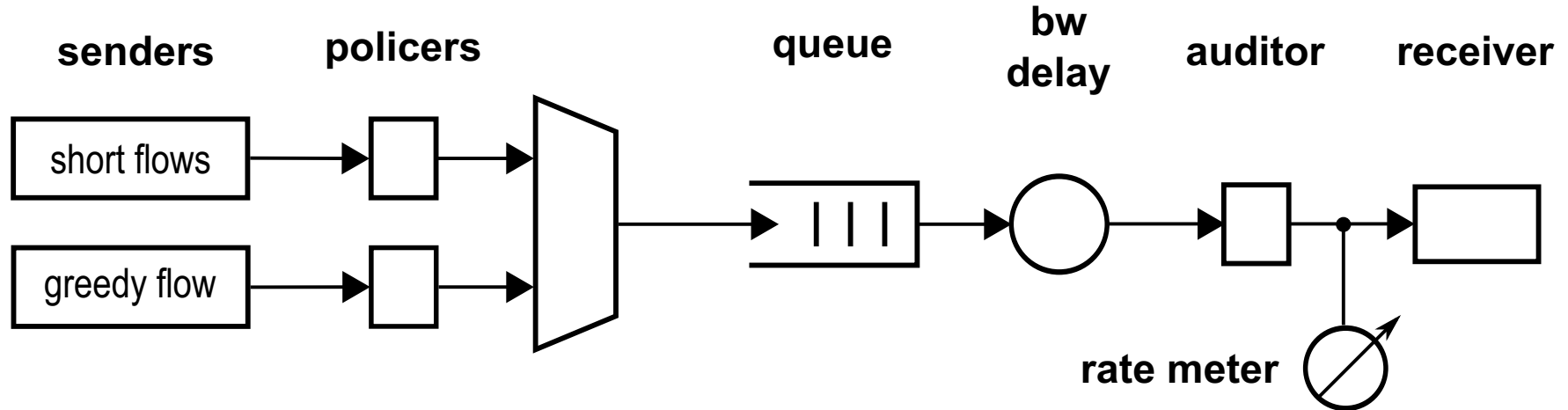


→ not limiting bursts, no shallow bucket needed

Goals for ConEx Policer

- Detect congestion increase due to competition for bandwidth fast
- Throttle heavy user by dropping packets
- Detect end of competition phase during throttling

Simple Scenario

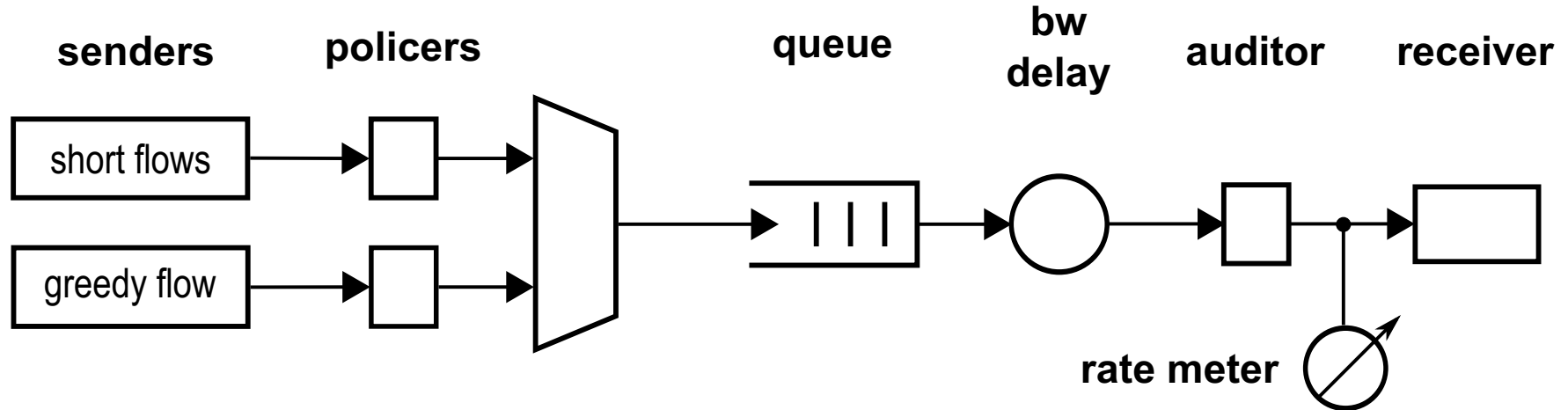


Network Parameters

- One-Way Delay (OWD) = 50 ms
- Bandwidth (BW) = 10 Mbit/s

DropTail queue size = 125000 Byte (Bandwidth-Delay-Product) = 83.3 full packets

Traffic Model



Greedy flow

- Always data to send available
- Starts at the beginning of simulation

Short flows

- Constant flow size of 50MB
- Constant inter-arrival time of 50 seconds

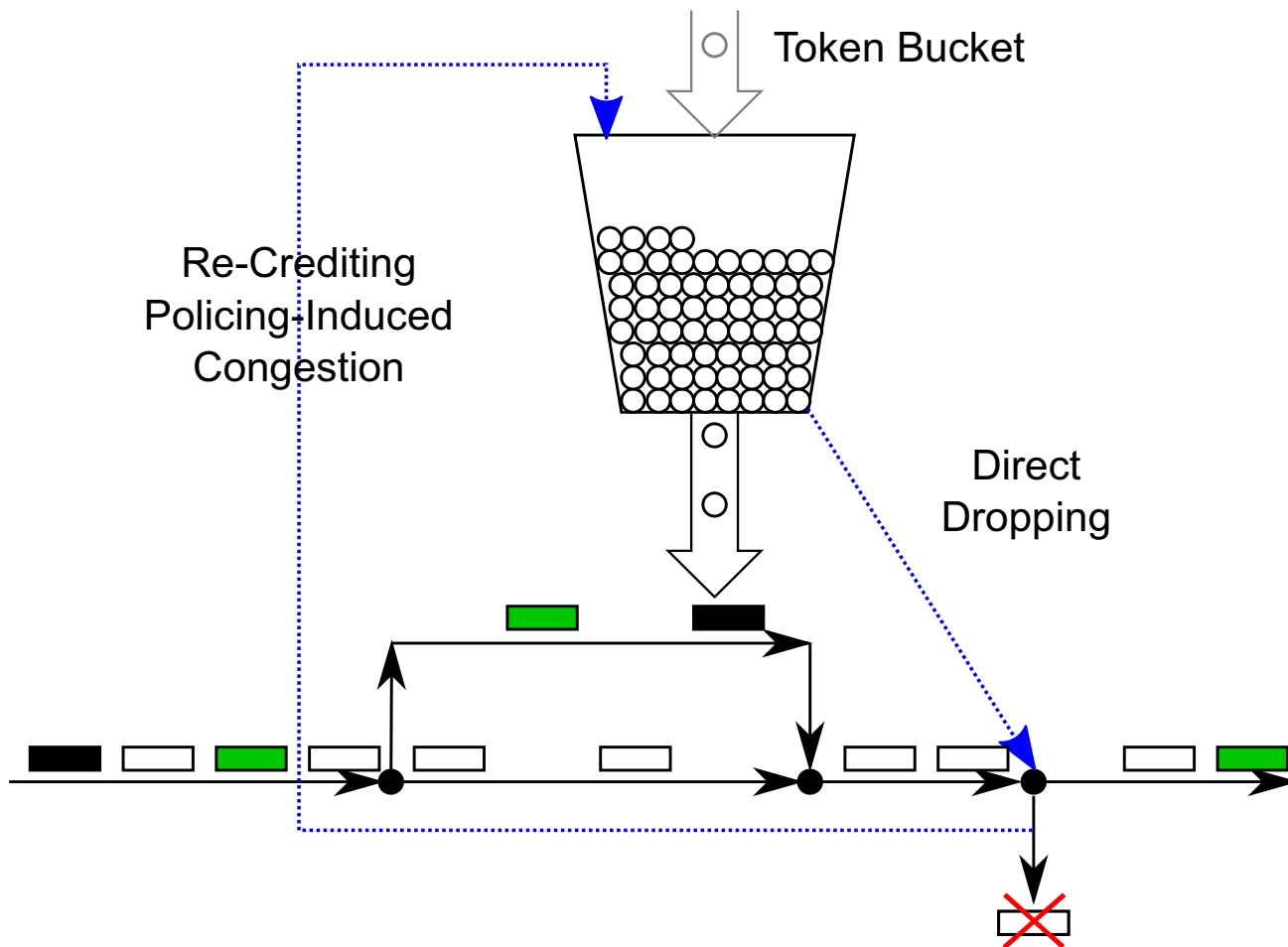
Endsystem Parameter

- Congestion control: TCP Reno

Policing Approaches

Simple Token-Bucket Policer

- One per-tenant token bucket
- Only marked packets cost tokens
- Police when bucket is empty
- Police by discarding packets
- Instantly credit for induced congestion signals → impact on bucket fill

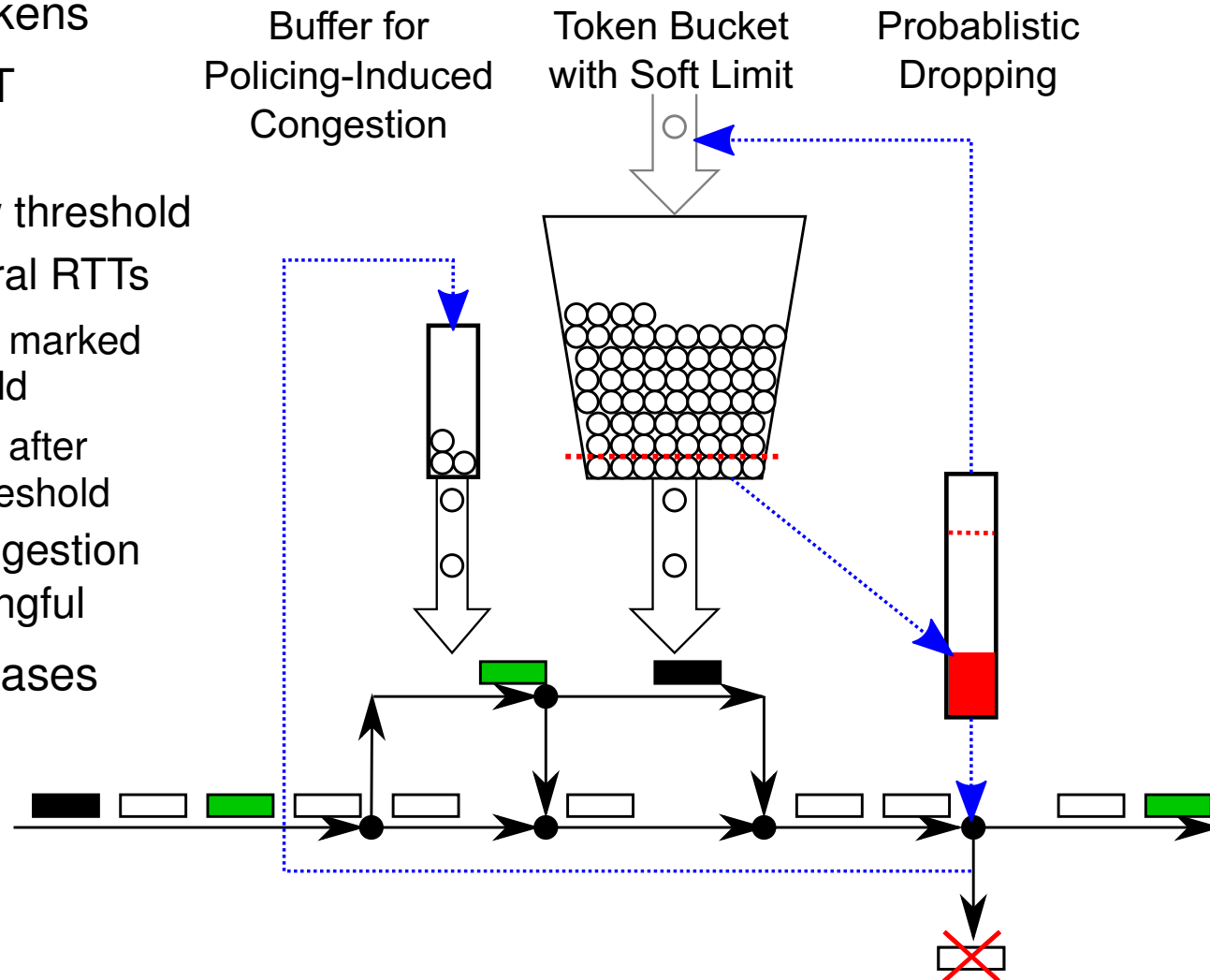


Policing Approaches

Drop-Rate Token-Bucket Policer

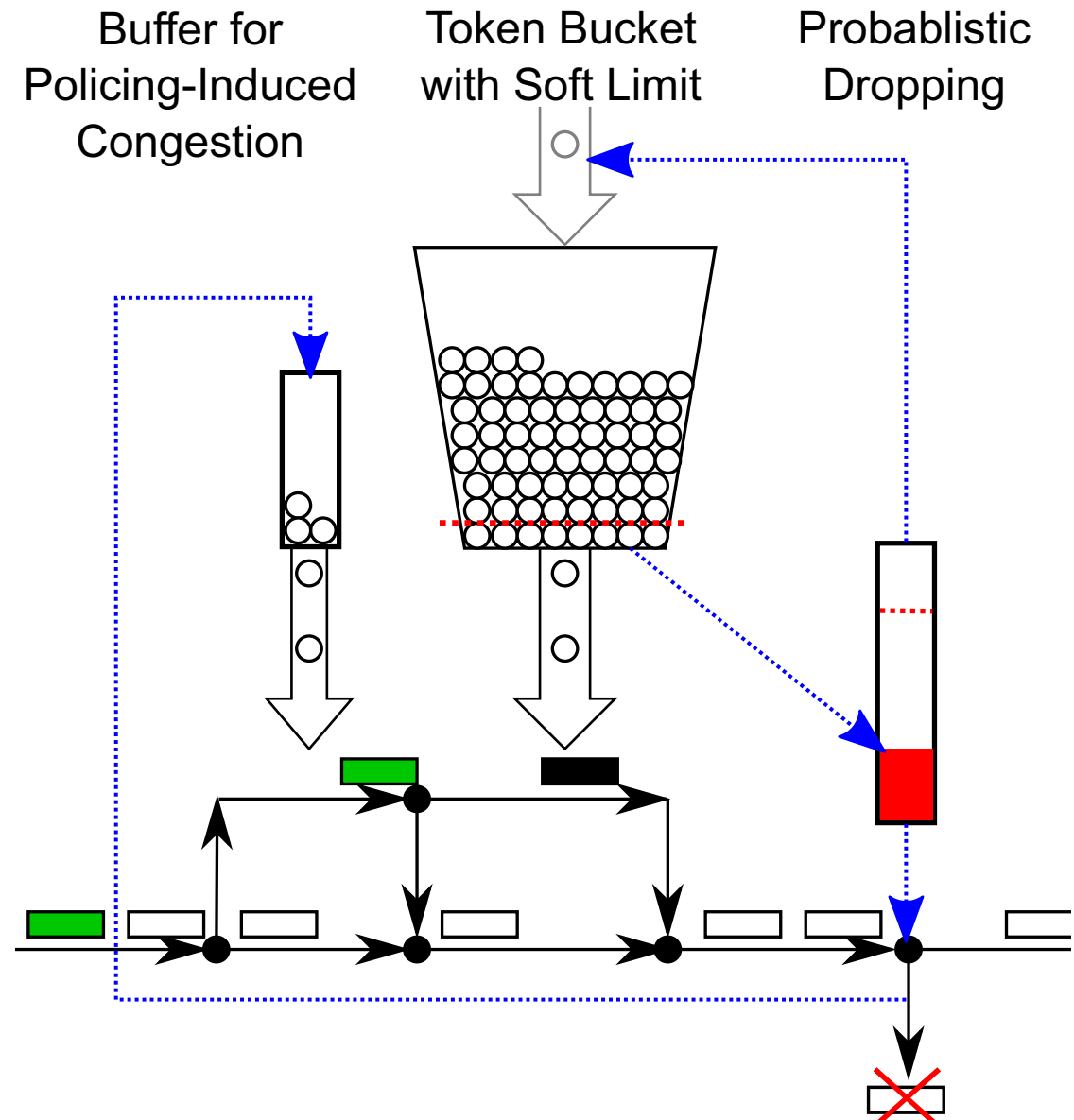
- Per-tenant token bucket
- Only marked packets cost tokens
- Police for more than one RTT to achieve stronger impact
 - Police when bucket is below threshold
 - Police by drop-rate for several RTTs
 - Increase drop rate for each marked packet while below threshold
 - Decrease drop rate linearly after bucket fill is higher than threshold
 - Buffer credit for induced congestion → bucket fill remains meaningful
- Do not reward for policing phases

Suspend bucket filling while drop probability > 0



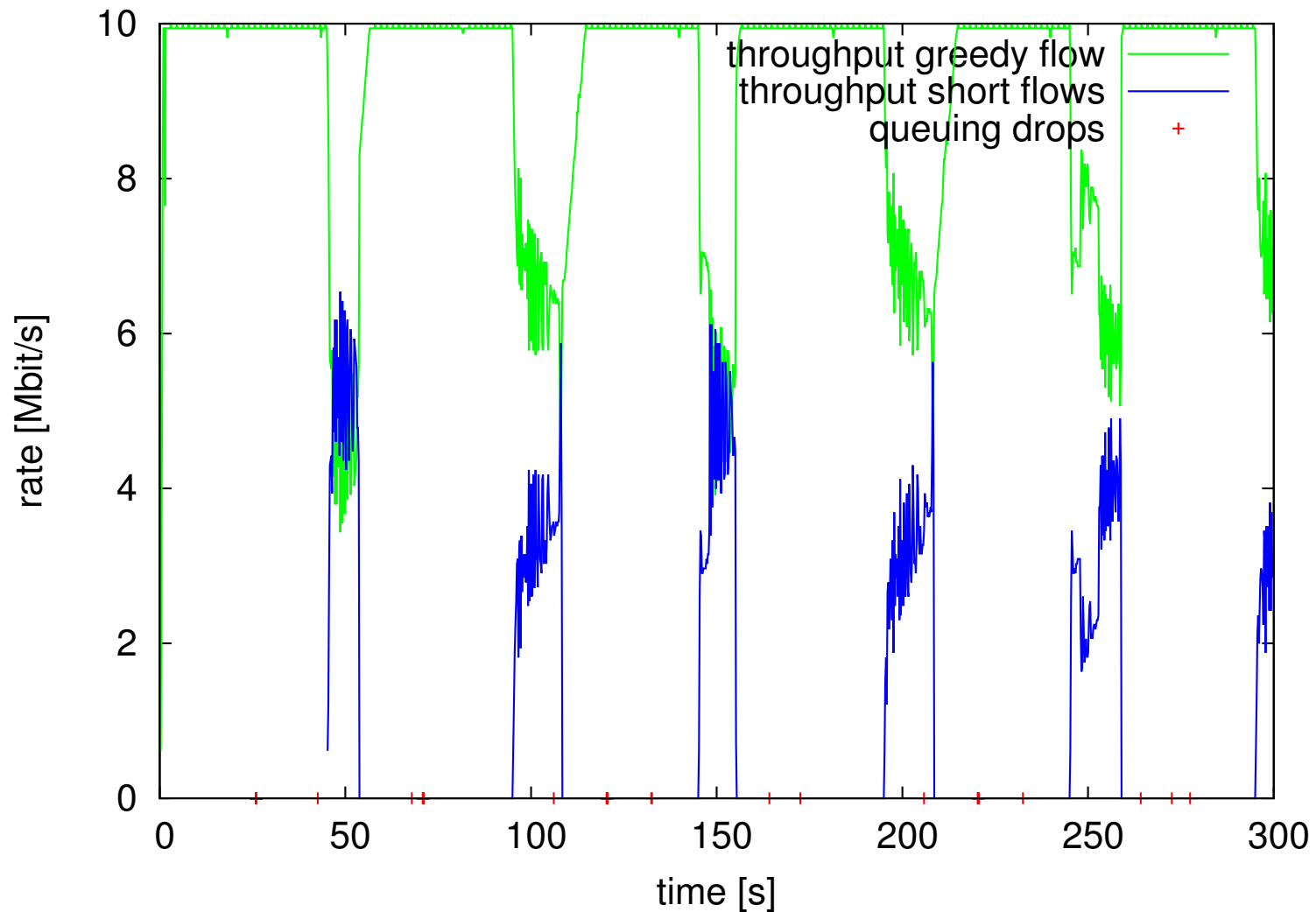
Policing Approaches

Drop-Rate Token-Bucket Policer



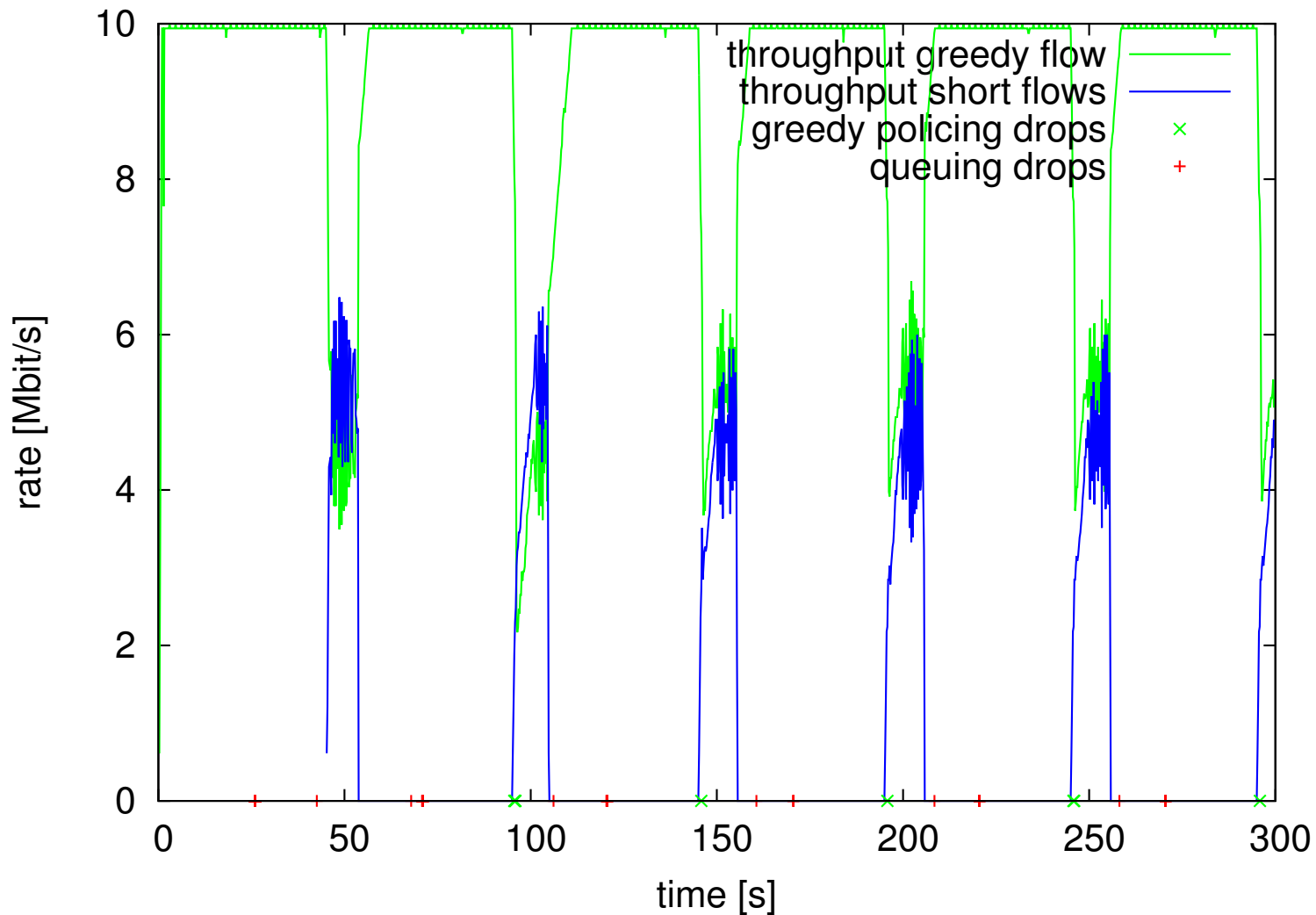
Simulation Results

Scenario without Policing



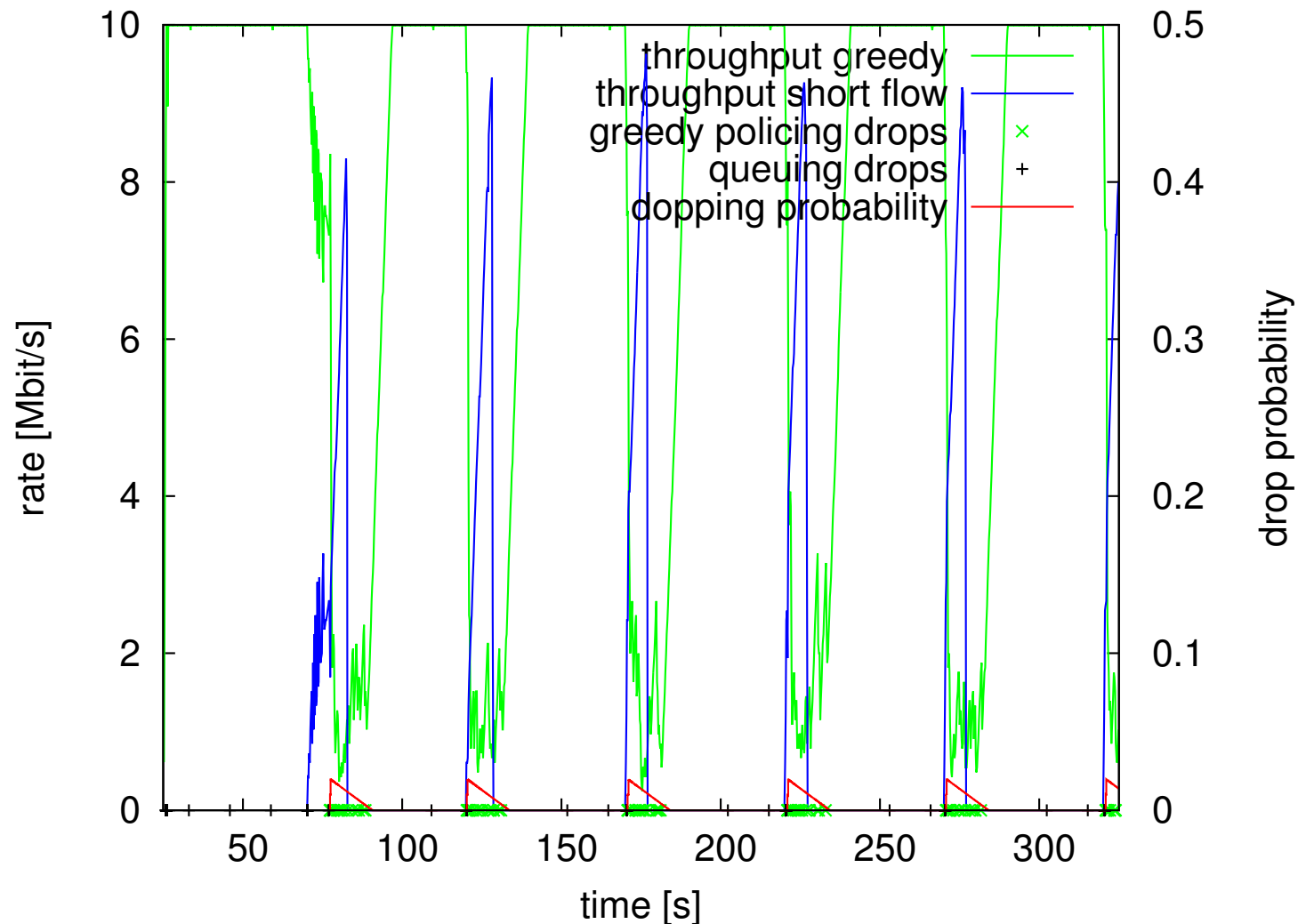
→ Light user gets less capacity in active periods but should get most capacity

Scenario with Simple Token-Bucket Policer



Short flows get (sometimes, slightly) more capacity due to policing drops of long-living flow

Scenario with Drop-Rate Token-Bucket Policer



Quite reliable & effective intervention. Bursts completed much faster. Slow retreat.

Conclusion

Working ConEx implementation

ConEx policing works also for today's limited-scalable congestion controls

- Parameterization critical
- More research needed

Future work

- Non-deterministic mechanisms, i.e. RED-queues
- Other congestion controls, e.g. cubic