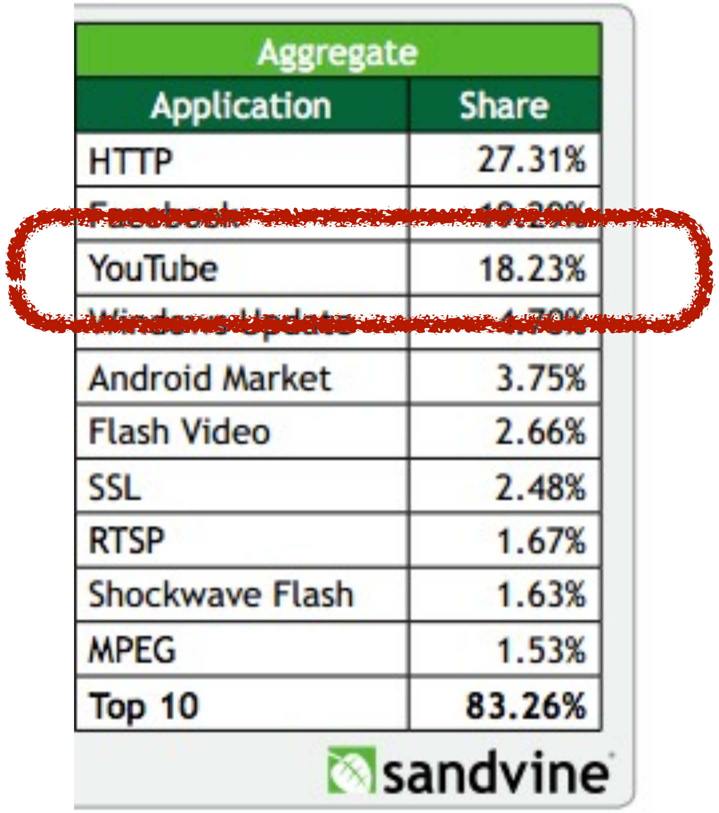
Trickle: Rate Limiting YouTube™ Video Streaming

Monia Ghobadi
<monia@cs.toronto.edu>

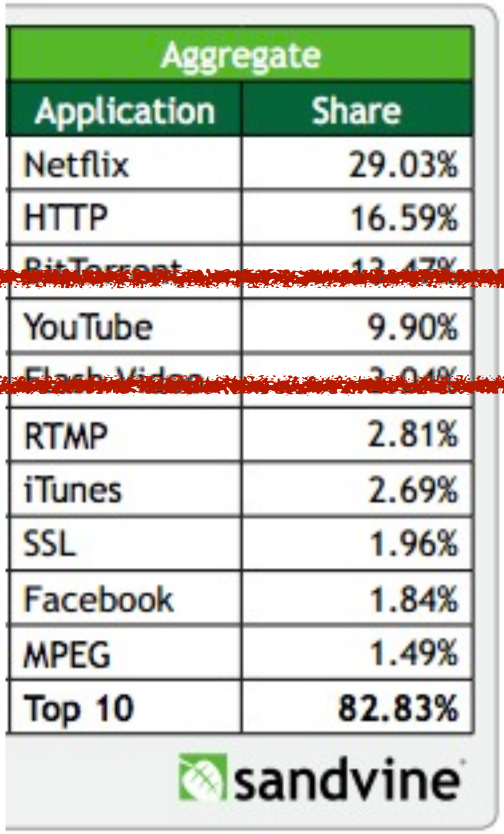
Yuchung Cheng, Ankur Jain, Matt Mathis
<ycheng, jankur, mattmathis@google.com>

YouTube Video Streaming



Aggregate	
Application	Share
HTTP	27.31%
Facebook	10.20%
YouTube	18.23%
Windows Update	4.70%
Android Market	3.75%
Flash Video	2.66%
SSL	2.48%
RTSP	1.67%
Shockwave Flash	1.63%
MPEG	1.53%
Top 10	83.26%

sandvine

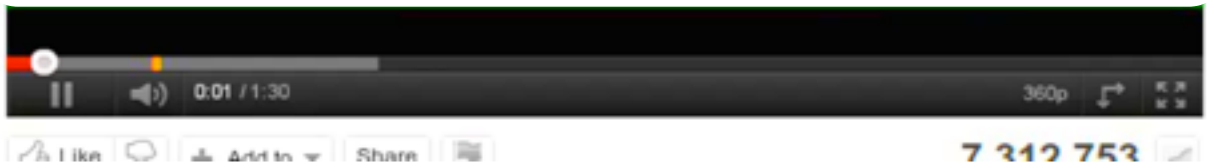


Aggregate	
Application	Share
Netflix	29.03%
HTTP	16.59%
BitTorrent	13.47%
YouTube	9.90%
Flash Video	3.04%
RTMP	2.81%
iTunes	2.69%
SSL	1.96%
Facebook	1.84%
MPEG	1.49%
Top 10	82.83%

sandvine

TCP

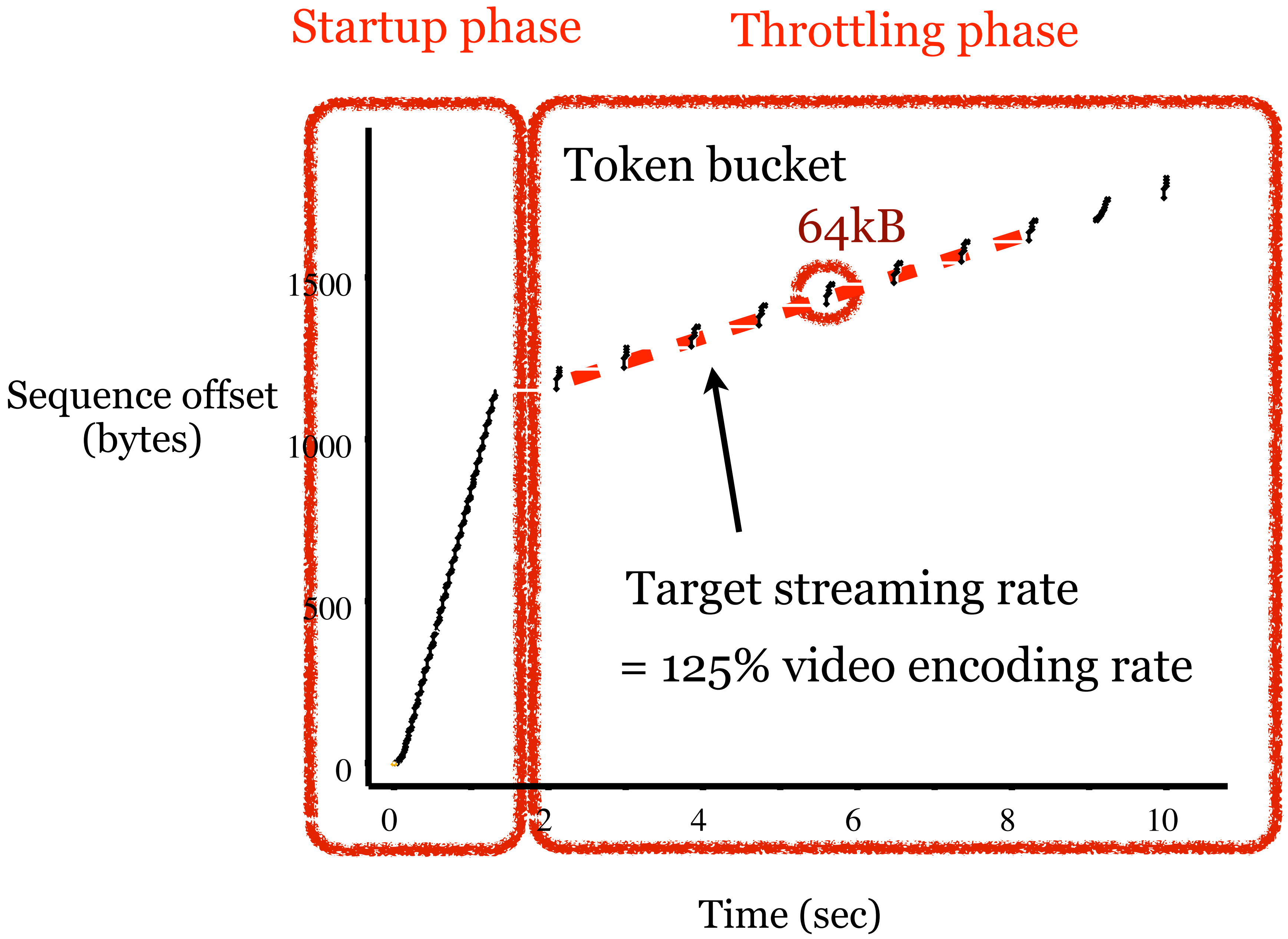
Just-in-time video delivery



Ustreamer

Application pacing

YouTube Video Streaming



The Problem: Burstiness is Bad for TCP

Not specific to YouTube videos.

Netflix sends bursts as large as 2MB.

Main contribution:

*A **simple** and **generic** technique to implement just-in-time video delivery by smoothly rate-limiting TCP transfers.*

Trickle To Rate Limit TCP

- Dynamic upper bound on TCP's congestion window.
- Periodically computed based on RTT and target rate (R).

R = 50 pkts/sec (600Kbps) RTT = 200 ms

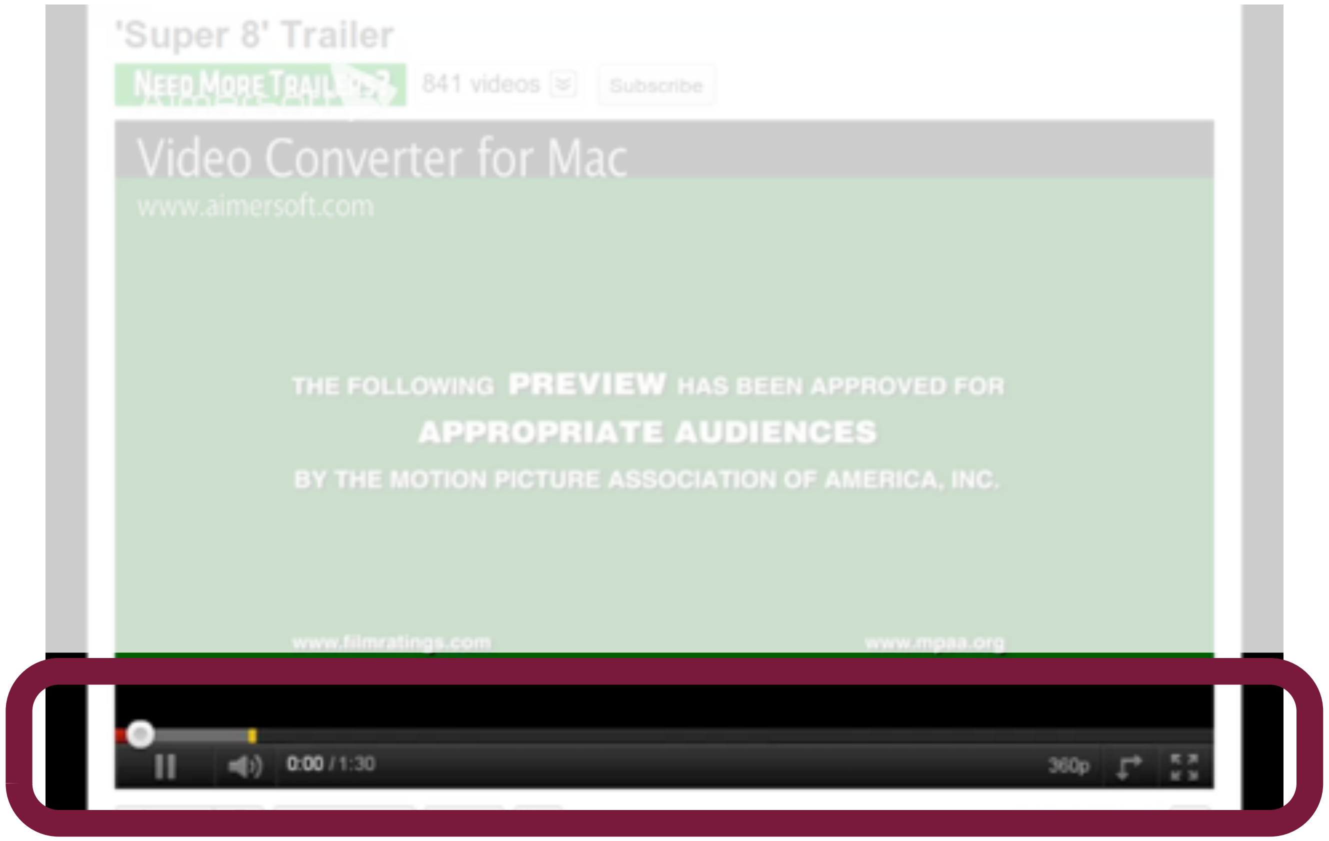
max_cwnd = 50 (pkts/sec) x 0.2 (sec) = 10 pkts

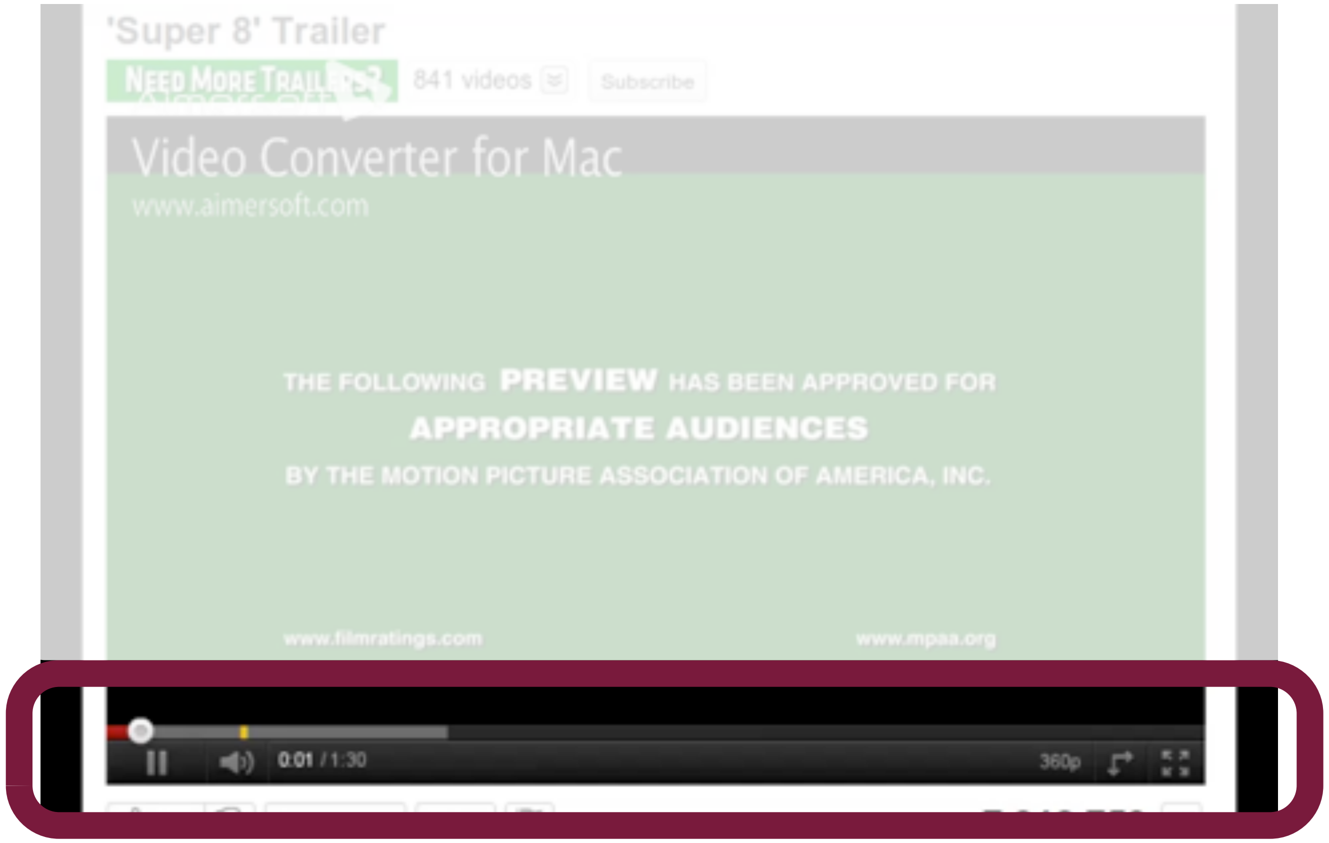
- Only server side changes for easy deployment.
- Not a special mechanism tailored only for YouTube.

Demo*

Smooth

Bursty





Trickle

Ustreamer

* http://www.cs.utoronto.ca/~monia/tcptrickle.html

Experiments

Two data centers: India and Europe.
15 days in Fall 2011, total of 23 million videos.

4-way experiment:
(1) Baseline1: application pacing with 64kB blocks,
(2) Baseline2: application pacing with 64kB blocks,
(3) Trickle,
(4) shrunk-block: application pacing with 16kB blocks.

Experiments Methodology

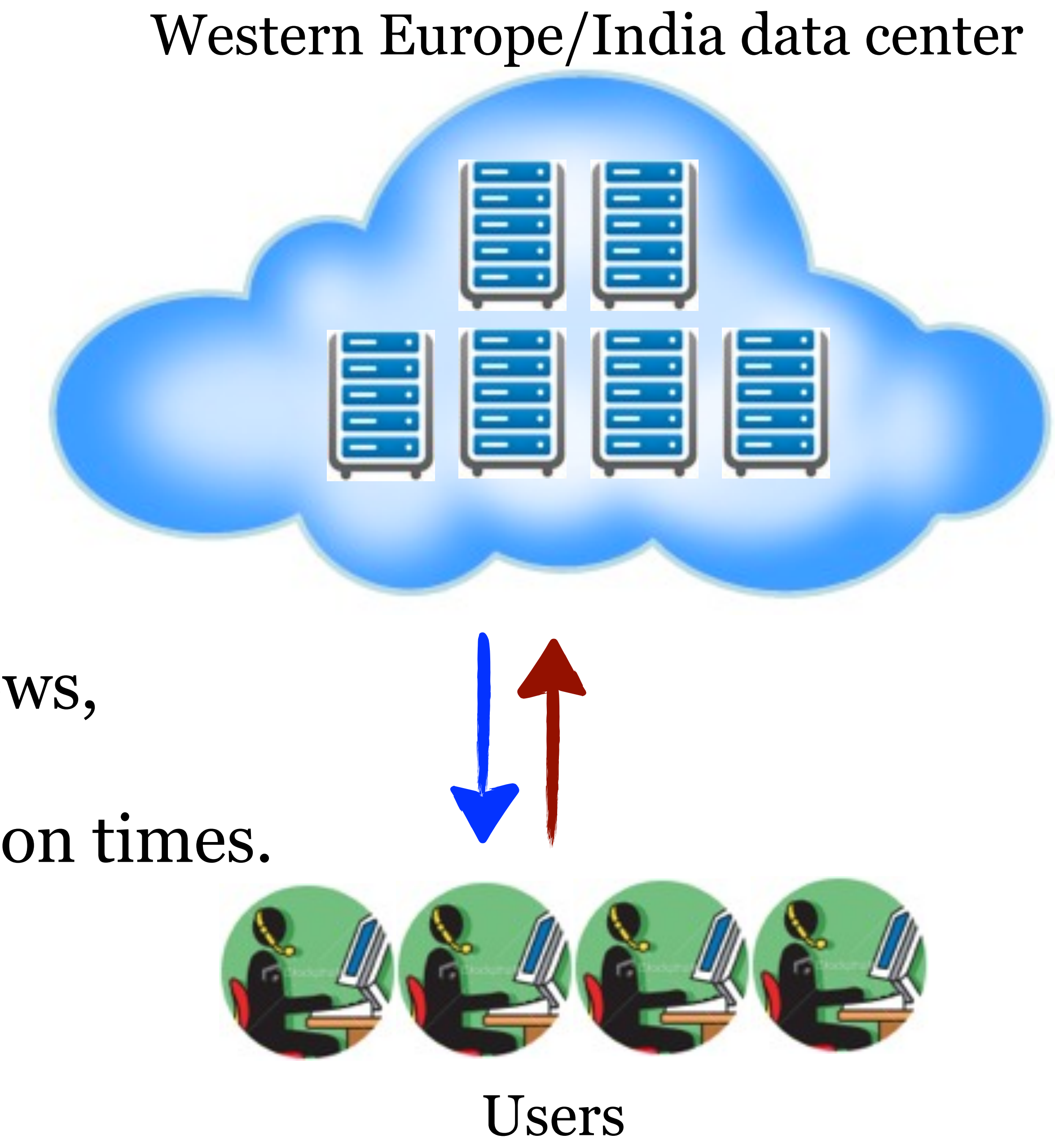
(1) Baseline1

(2) Baseline2

(3) Trickle

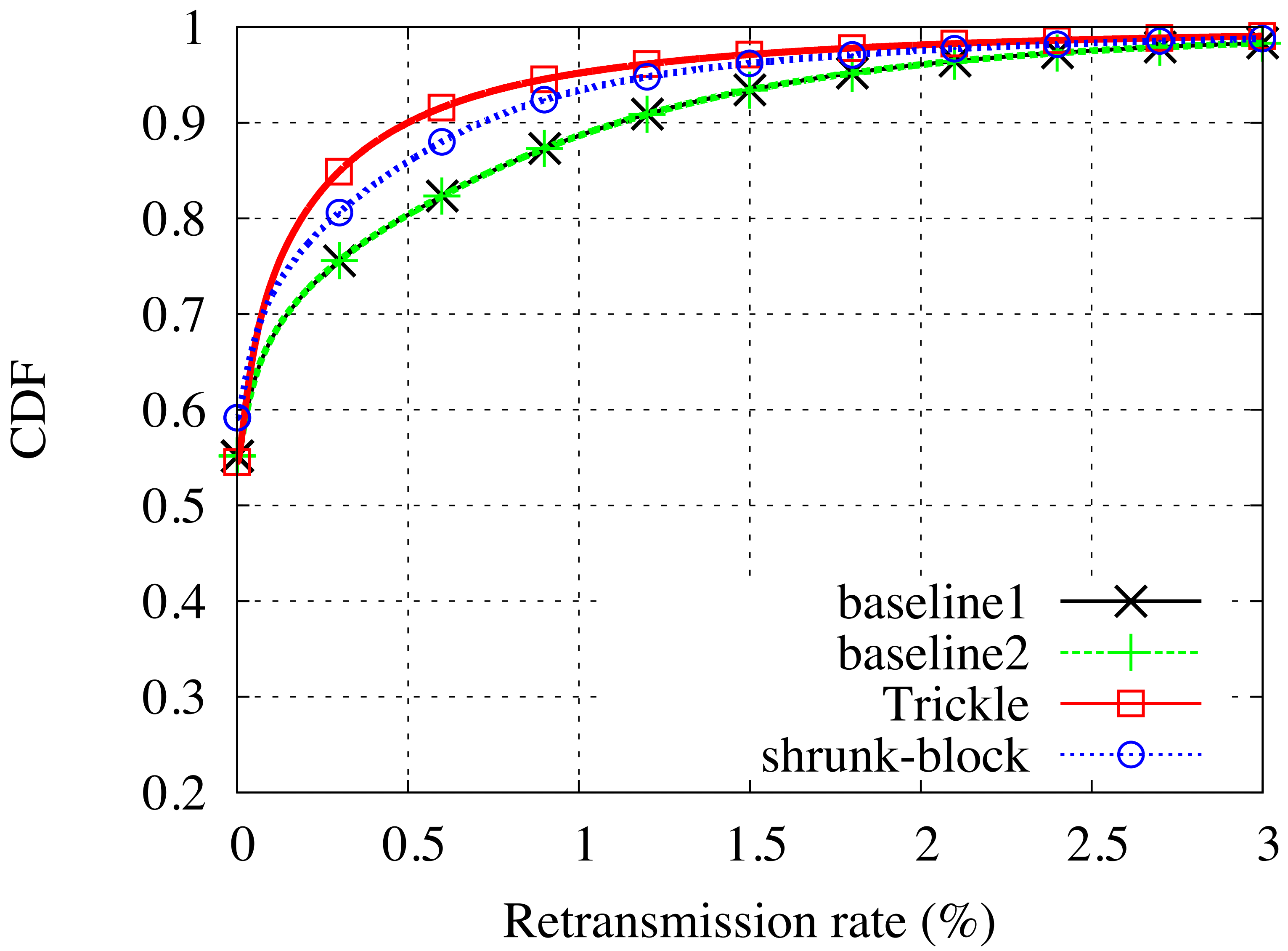
(4) shrunk-block

Same number of flows,
flow sizes,
flow completion times.



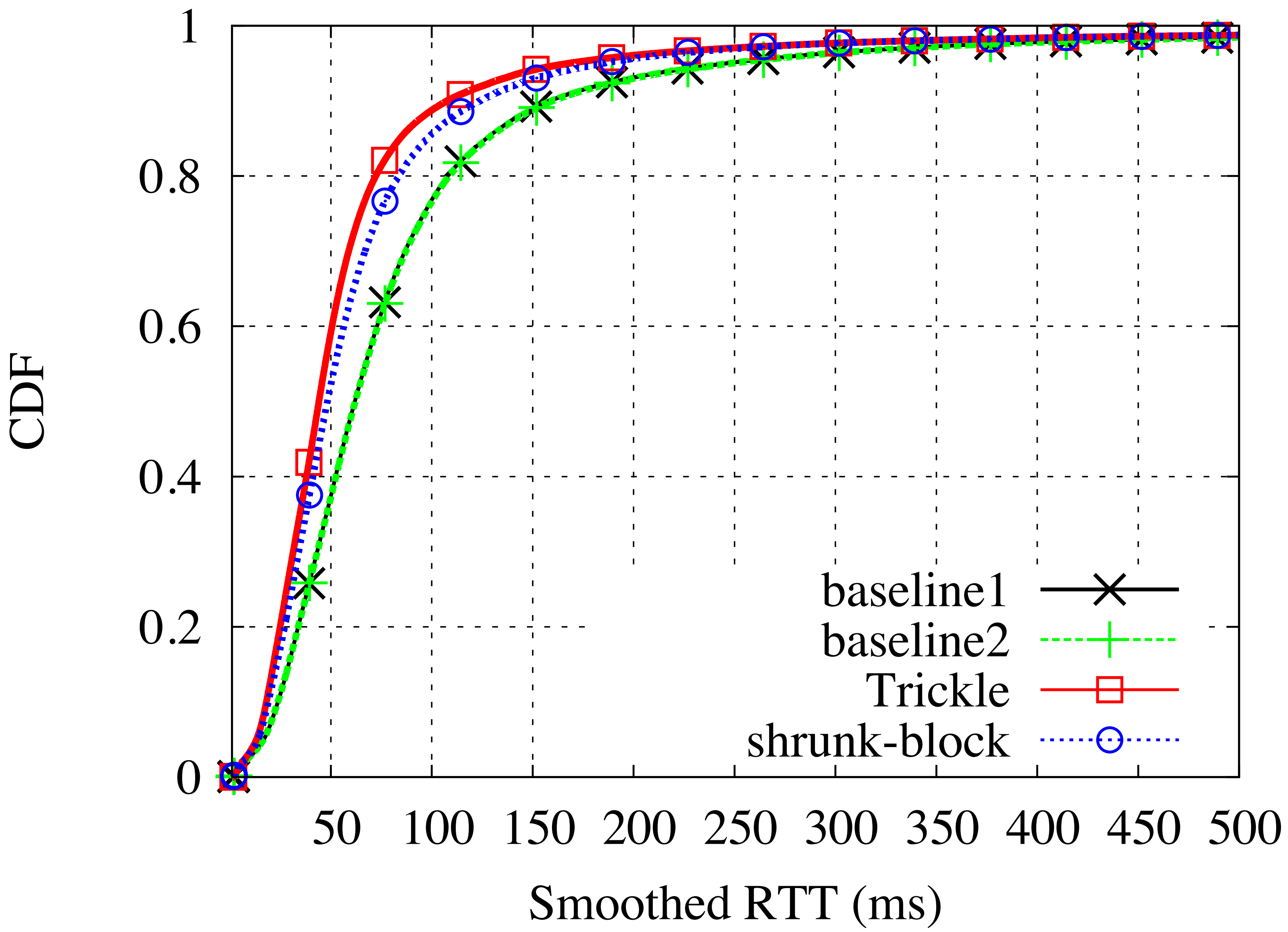
Video ID
IP/Port
Bytes sent
Retransmission rate
RTT
Transmission time
Goodput
Target rate

Experiments: Packet Losses



Trickle reduces the average retransmission rate by 43%.

Experiments: Queueing Delay



Trickle reduces the average RTT by 28%.

Conclusions

- Trickle rate limits TCP by dynamically setting the maximum congestion window size.
- Minimal sender-side changes, fast deployment.
- Generally applicable to rate limit other kinds of streaming.