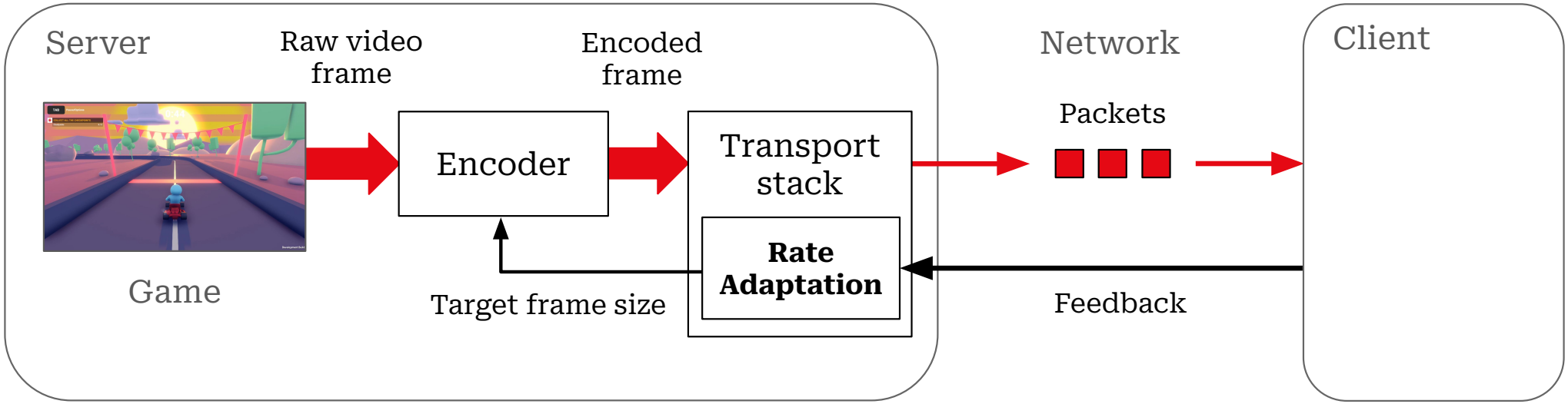
# Network Delivery Time Control: a Novel Approach to Rate Adaptation for Low-Latency Video Flows

Paul-Louis Ageneau  Grenville Armitage
Netflix

# Context: Rate Adaptation for Cloud Gaming




The rate adaptation algorithm uses feedback from client to drive the encoder target bitrate in real time and pace outgoing packets.

# Existing Congestion Control Algorithms

Existing algorithms for real-time flows (*e.g.* GCC, SCReAM) typically rely on classic congestion control strategies:

- Probe capacity by slowly increasing encoder target bitrate
- Back off on early congestion signal (*e.g.* RTT increase) or loss

It has multiple issues for high-bitrate ultra-low-latency cloud gaming:

- Early congestion signals are error-prone
- Ramp-up to the maximum bitrate is slow
- Encoder must stick to the target bitrate
- It induces delays or packet losses by design

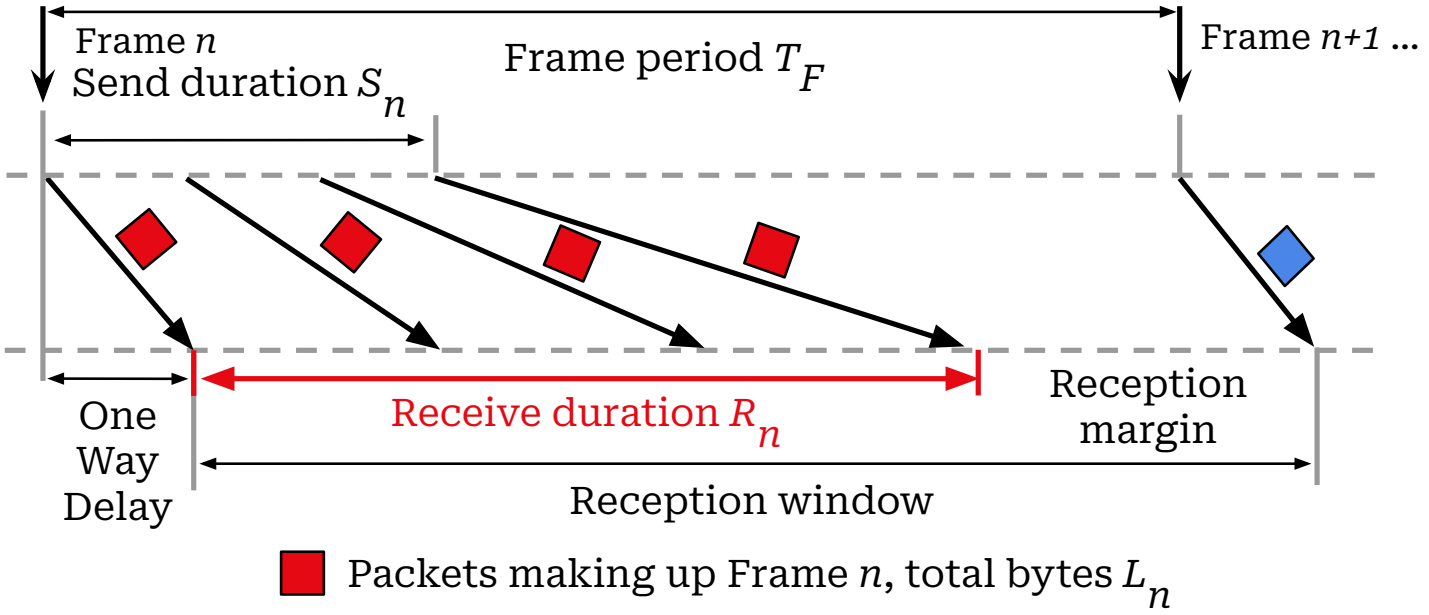
Other algorithms (e.g. Salsify, BurstRTC, and SQP) have attempted to leverage packet trains to measure capacity, however:

- In practice, packet bursts lead to losses and reduced QoE
- They fail to properly account for cross-traffic

# Network Delivery Time Control

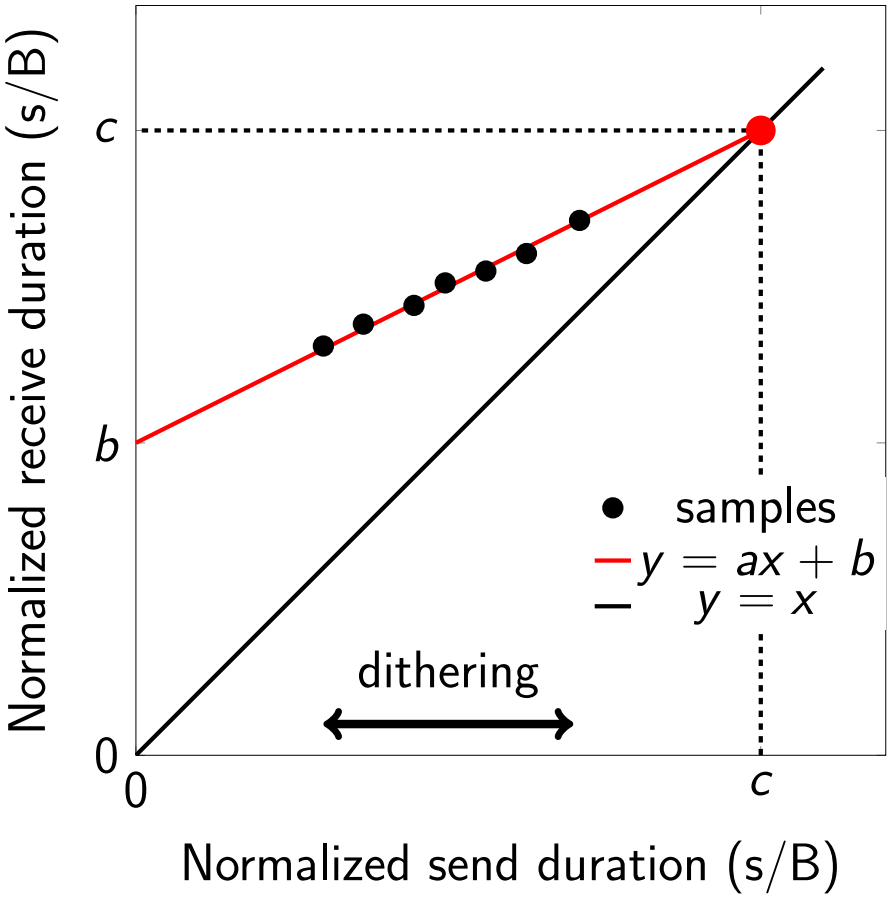
Estimate available capacity to proactively minimize packet loss and queue delays

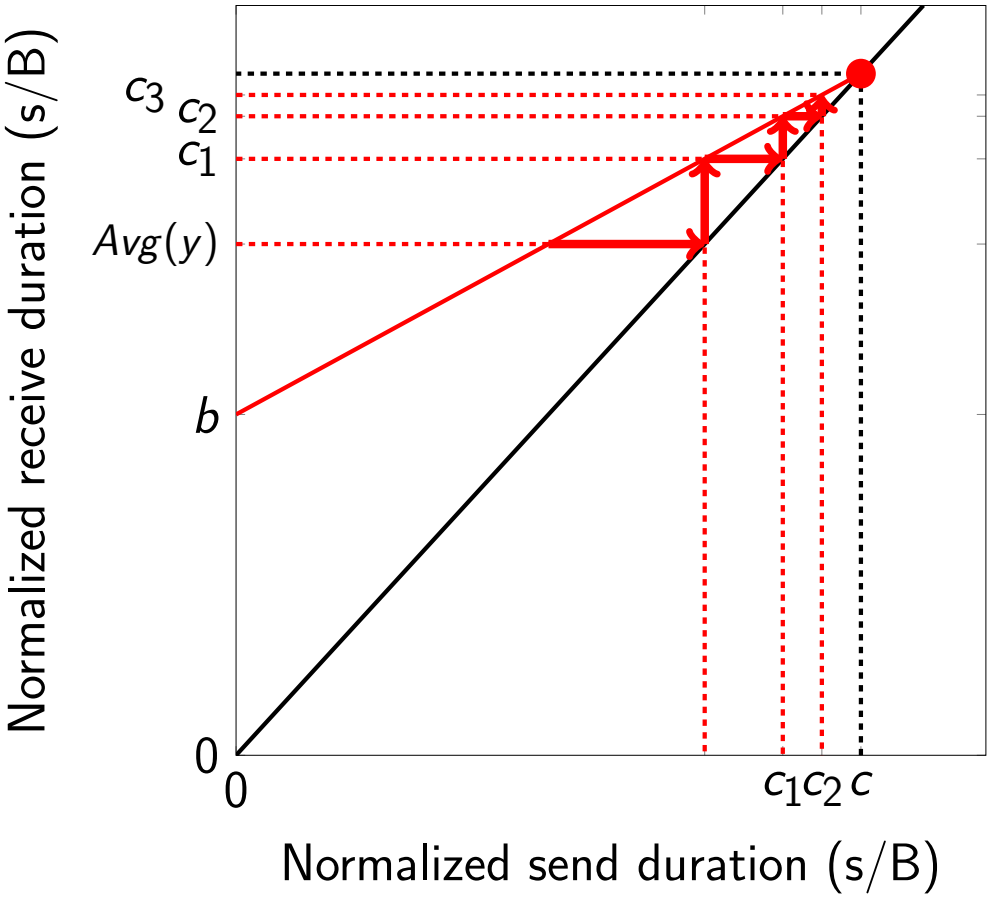
- Ensure frames arrive on time instead of probing capacity
- Allow for encoded frame size variability
- Send no supplemental traffic




FDACE leverages paced frames as probes to estimate available capacity from the relationship between frame send duration $S_n$ and frame receive duration $R_n$. NDTC adapts target frame size so frames are received within target duration.

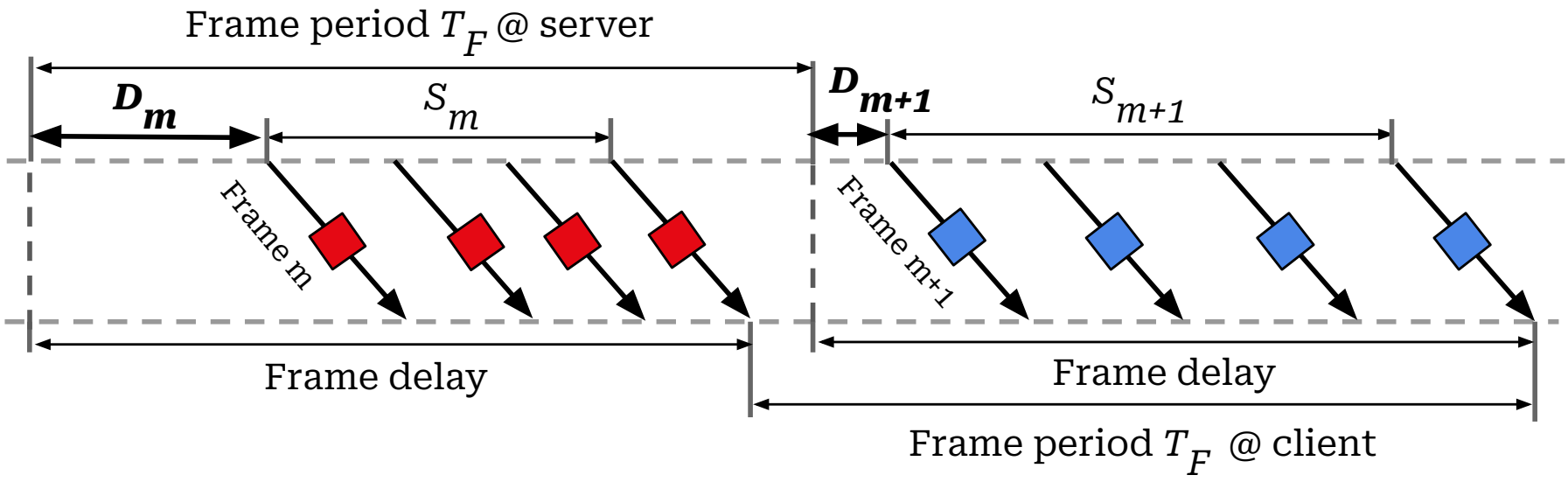
# Frame Dithering Available Capacity Estimation



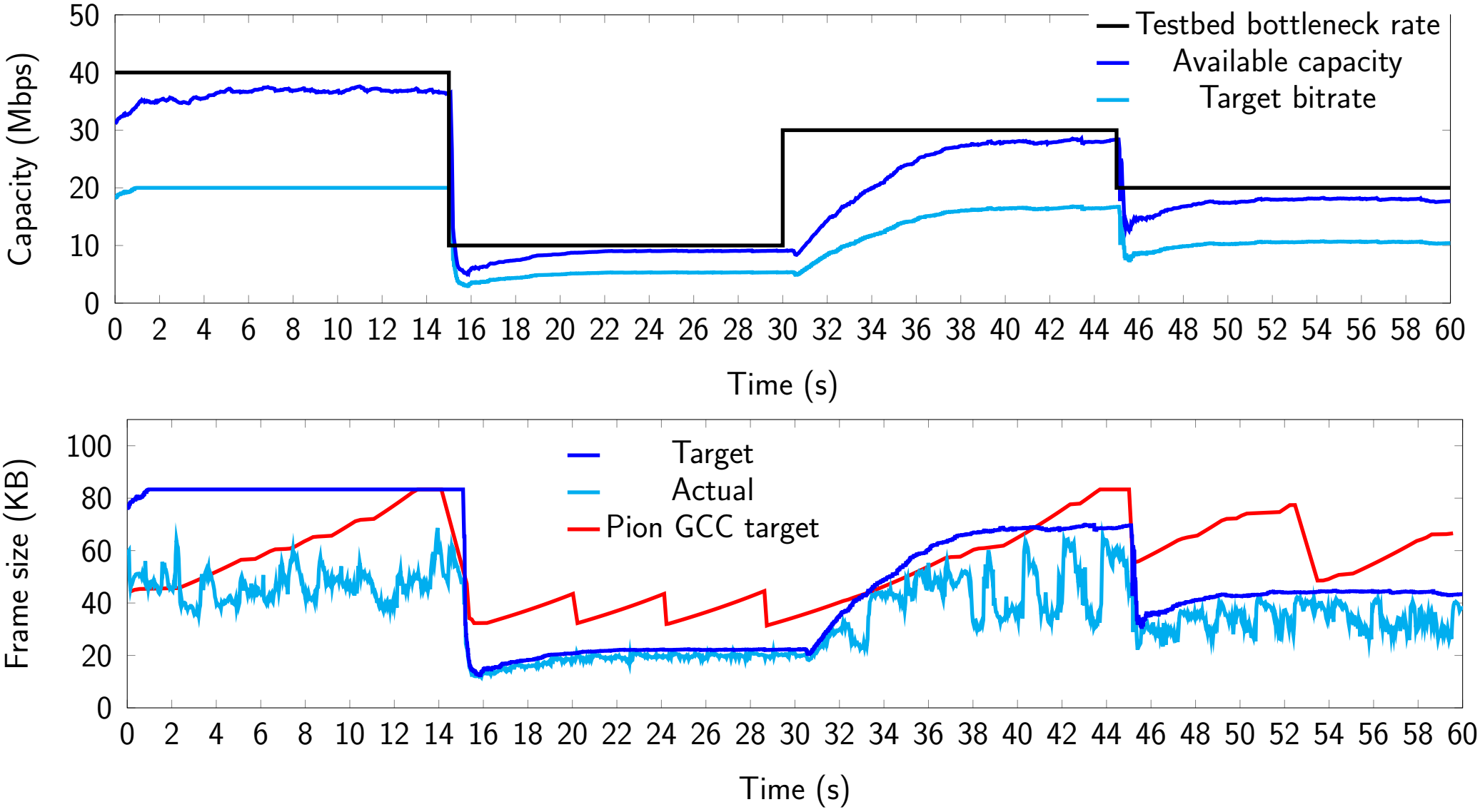

- At y-intercept (bursting a frame), $b$ corresponds to total capacity.
- At intersection with $y = x$ (sending at receive rate), $c$ corresponds to available capacity.

# Time-aligned delivery



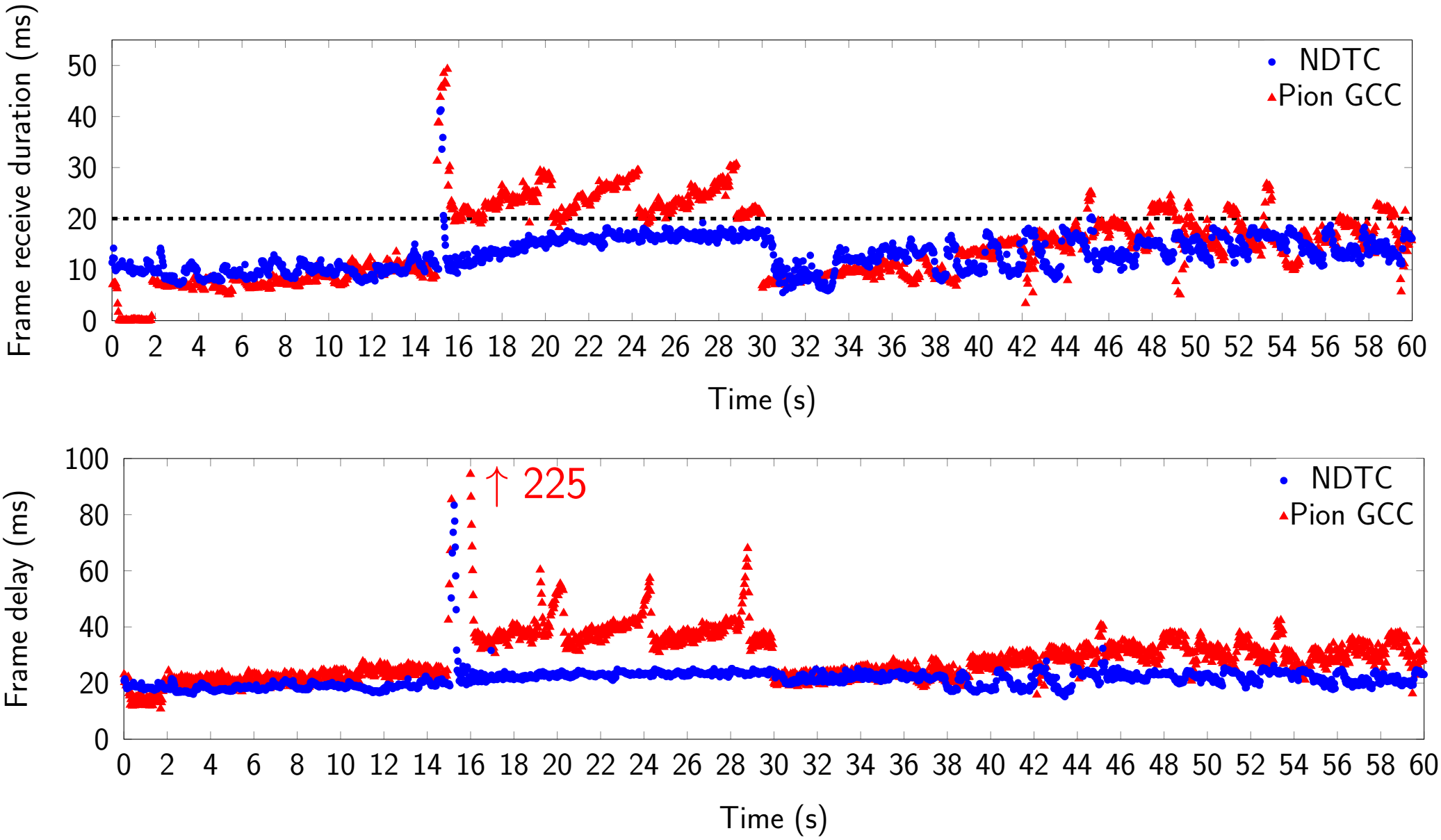

NDTC introduces a send delay $D_m$ before pacing each frame to stabilize delay and make the cadence more consistent at reception, which reduces frame jitter.

# Tracking Time-varying Available Capacity



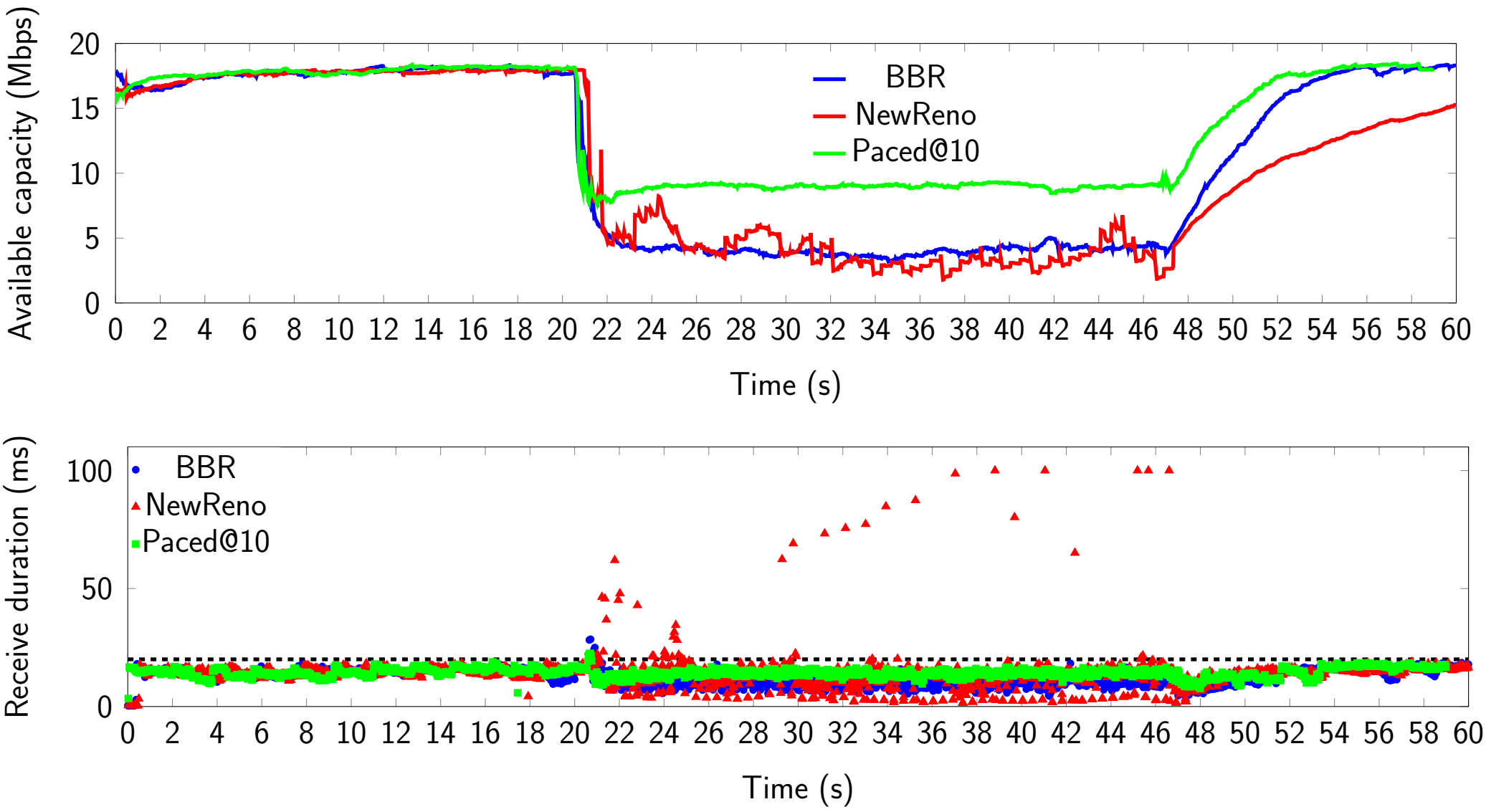

FDACE tracks changes closely and target is decoupled from actual frame size.

# Controlling Receive Duration and Frame Delay



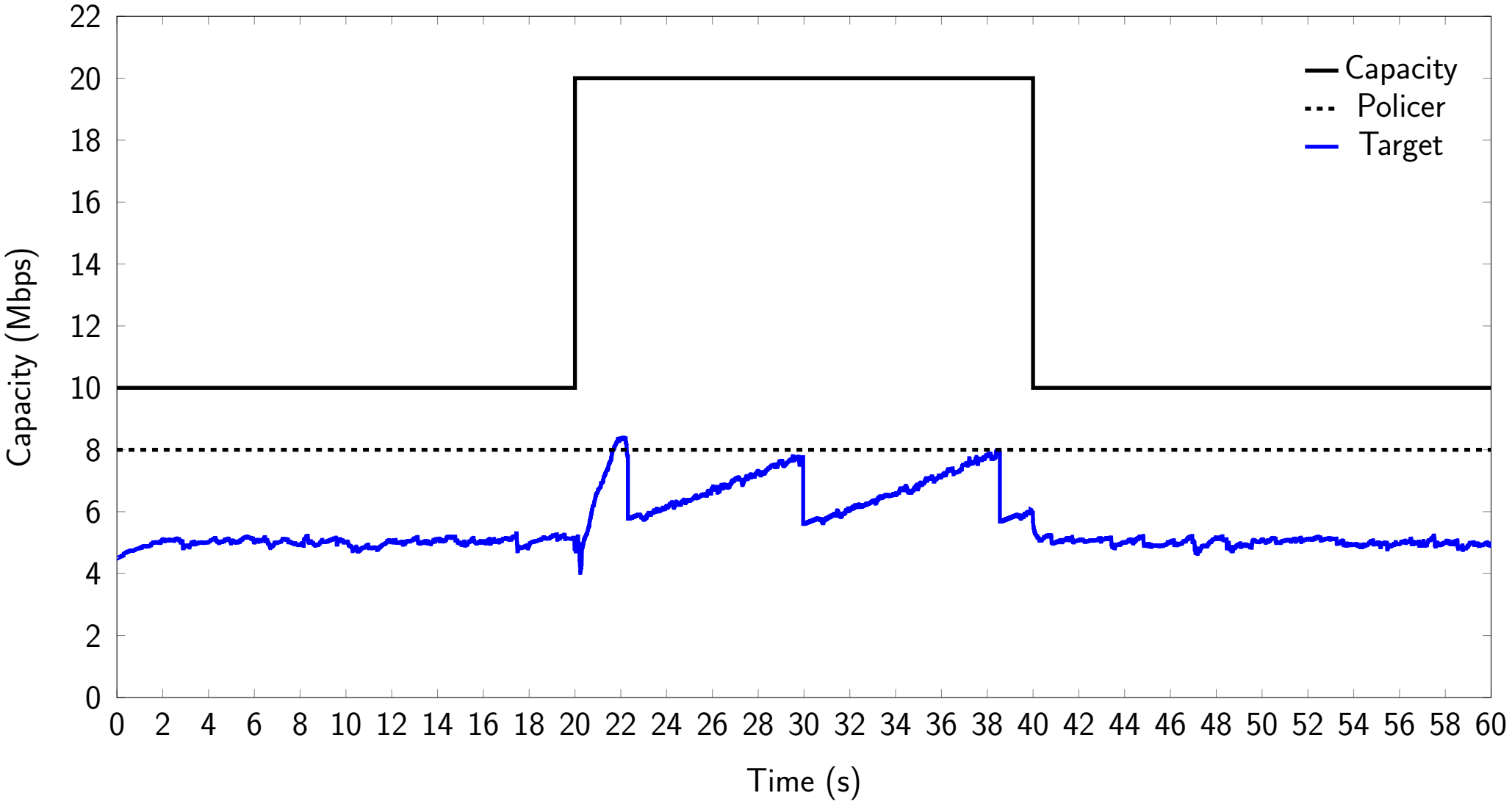

NDTC keeps frame receive duration and frame delay under control.

# Coexisting with TCP-like cross-traffic




Available capacity is measured correctly even with cross-traffic so NDTC coexists with other flows, but it favors timely delivery over fairness.

# Coupled AIMD congestion control for loss signal




NDTC seamlessly transitions between capacity estimation and packet loss.