Multipath Parallel Reverse Segment Download (MPRD) for Peer-to-Peer Content Delivery

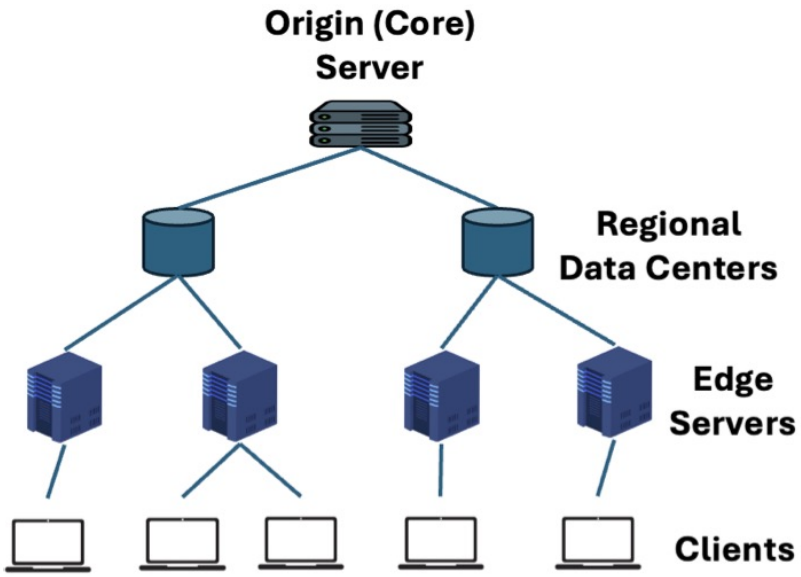
Suresh Srinivasan and **Ehsan Aryafar**

Computer Science Department, Portland State University, Portland, OR, USA

July 21st, 2025

IEEE ICDCS Conference, Glasgow, UK

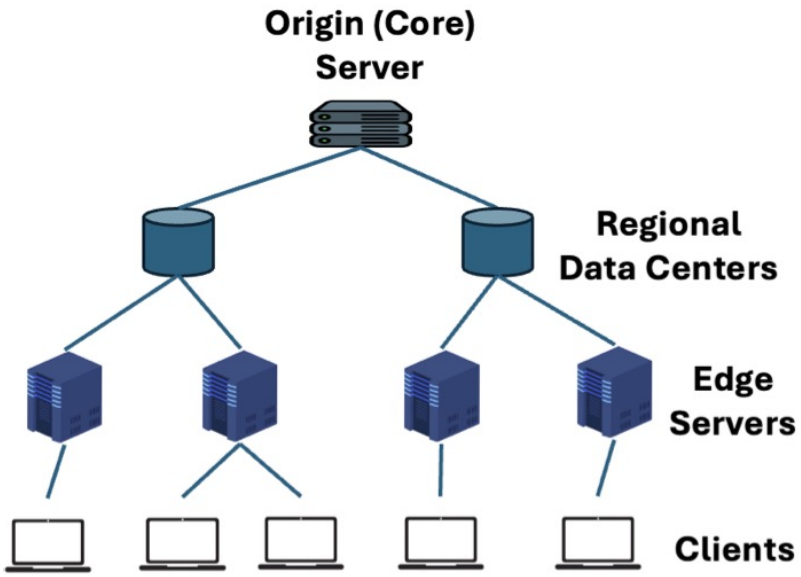
Background: CDN and PCDN

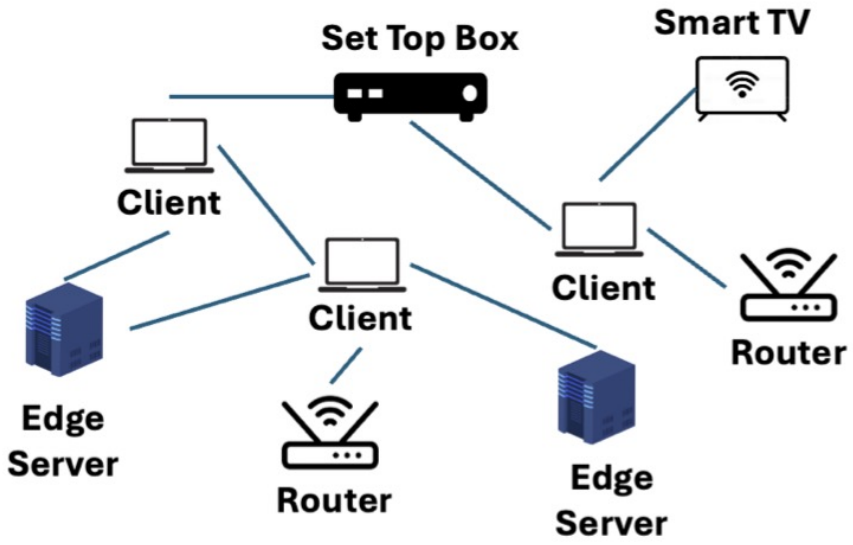


CDN (Content Delivery Network)

- Hierarchical architecture with original, regional, and edge servers
- Deliver content quickly and reliably
- High cost to deploy and maintain

Background: CDN and PCDN





CDN (Content Delivery Network)

- Hierarchical architecture with original, regional, and edge servers
- Deliver content quickly and reliably
- High cost to deploy and maintain

PCDN (Peer-to-Peer CDN)

- Use edge devices (smart TVs, routers) to store and send data
- Use to augment or replace CDNs
- Storage and connectivity issues

PCDN History and Evolution

- P2P File sharing (late 1990s)
  - End devices can also help store and send, not just consume
  - Napster (1999) and BitTorrent (2001)

PCDN History and Evolution

- P2P File sharing (late 1990s)
  - End devices can also help store and send, not just consume
  - Napster (1999) and BitTorrent (2001)
- First gen P2P to support video streaming and content (2005-2012)
  - PPLive / PPStream (China): P2P live streaming for TV
  - Octoshape: P2P streaming for broadcasters like CNN

PCDN History and Evolution

- P2P File sharing (late 1990s)
  - End devices can also help store and send, not just consume
  - Napster (1999) and BitTorrent (2001)
- First gen P2P to support video streaming and content (2005-2012)
  - PPLive / PPStream (China): P2P live streaming for TV
  - Octoshape: P2P streaming for broadcasters like CNN
- Second generation P2P CDN (2012-2018)
  - WebRTC (real-time browser-based P2P) opened new doors
  - Netflix and others experimented for on-demand video

PCDN History and Evolution

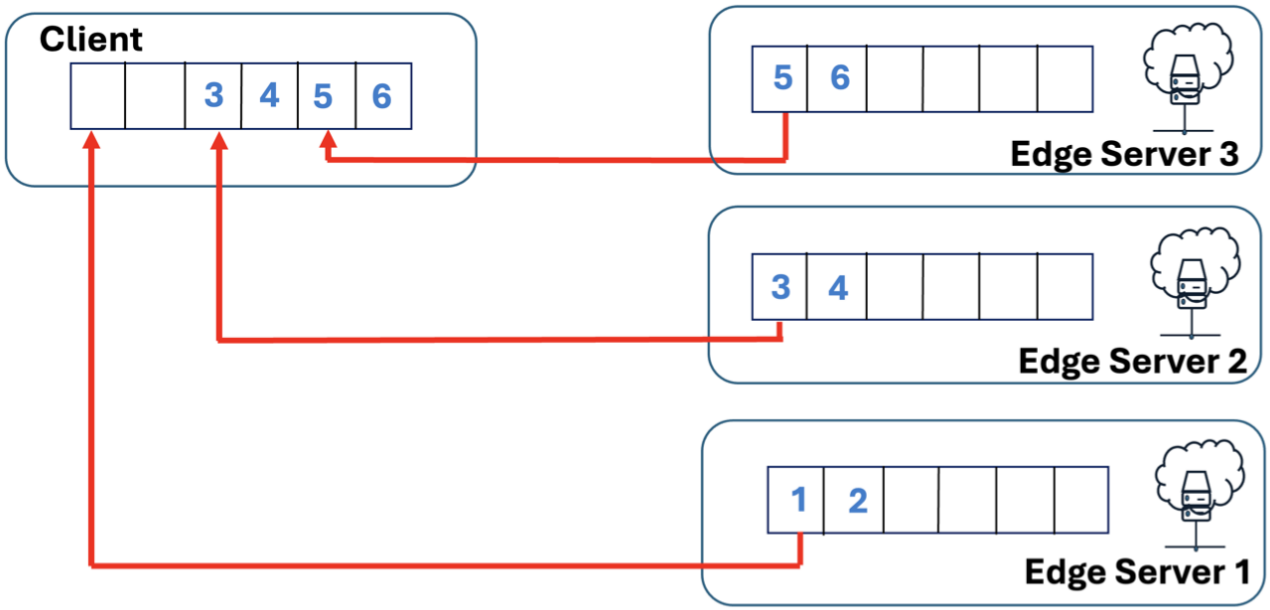
- P2P File sharing (late 1990s)
  - End devices can also help store and send, not just consume
  - Napster (1999) and BitTorrent (2001)
- First gen P2P to support video streaming and content (2005-2012)
  - PPLive / PPStream (China): P2P live streaming for TV
  - Octoshape: P2P streaming for broadcasters like CNN
- Second generation P2P CDN (2012-2018)
  - WebRTC (real-time browser-based P2P) opened new doors
  - Netflix and others experimented for on-demand video
- Recent renewed interest in P2P CDN
  - ByteDance using PCDNs to replace CDNs (short videos)
  - USENIX ATC 2024 and P-Scheduler INFOCOM 2024

PCDN Architecture

- How to handle device unreliability in PCDNs?

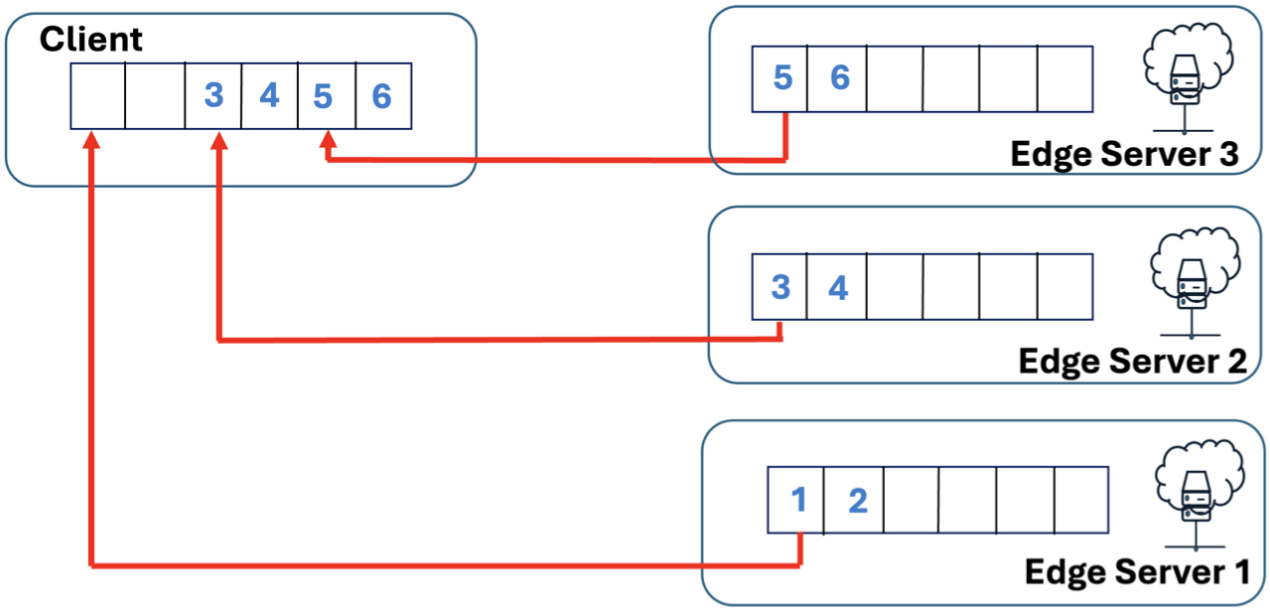
PCDN Architecture

- How to handle device unreliability in PCDNs?
  - Use multiple servers in parallel to stream data to each user



PCDN Architecture

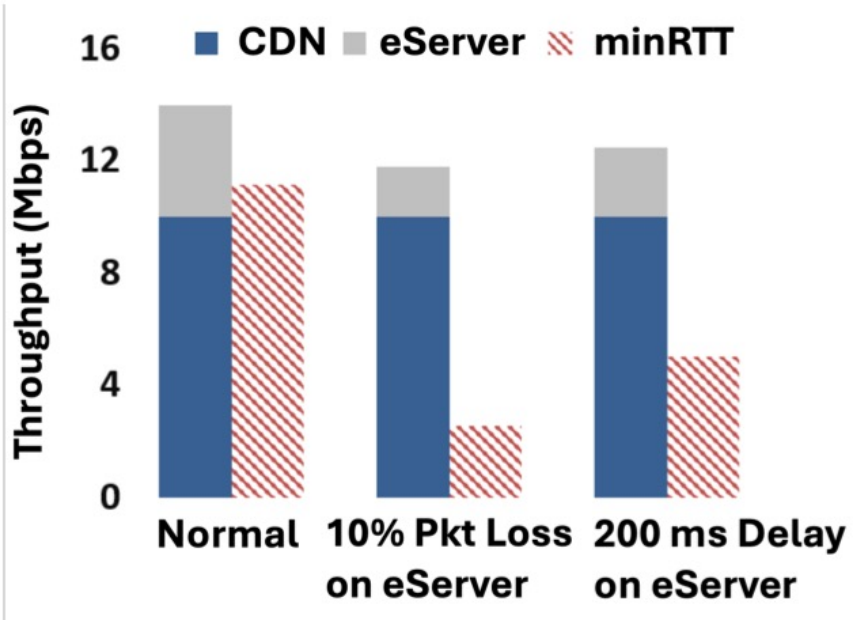
- How to handle device unreliability in PCDNs?
  - Use multiple servers in parallel to stream data to each user



- However, Head-of-Line Blocking (HoL) and Out-Of-Order (OOO) delivery can significantly drop throughput

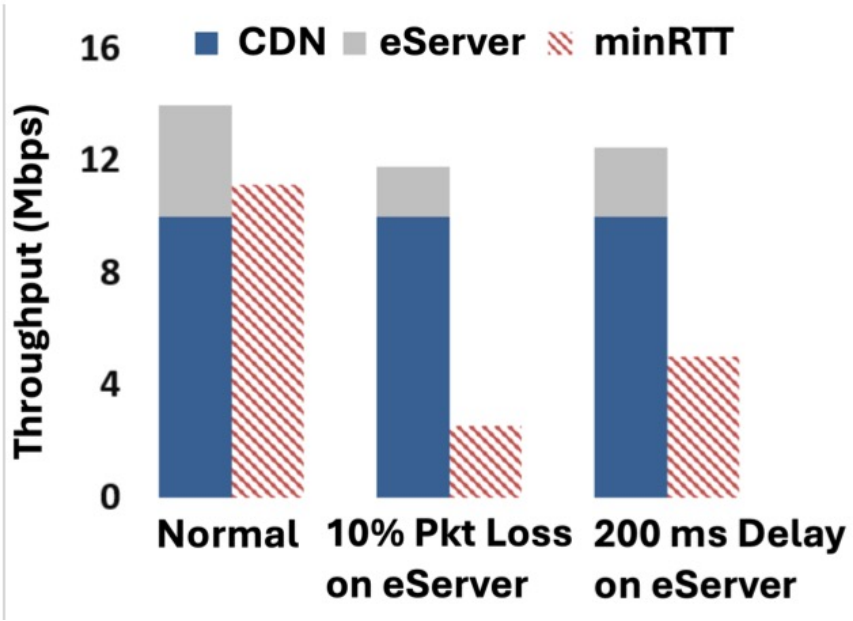
Simple Quantification of HoL Blocking Impact

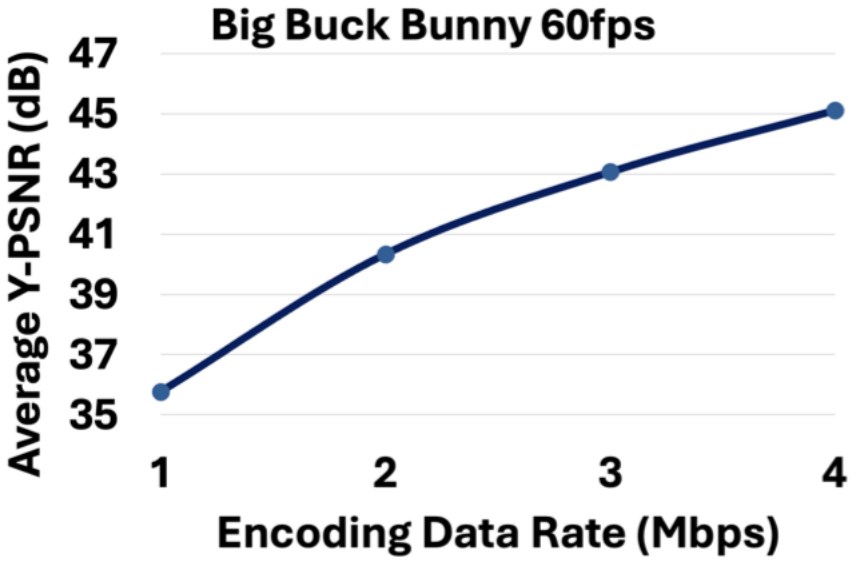
- Setup: CDN and edge server (eServer) instantiated on a single Linux machine, client connected to both on another Linux machine
- minRTT: Multi-Path TCP Scheduler updated for PCDN



Simple Quantification of HoL Blocking Impact

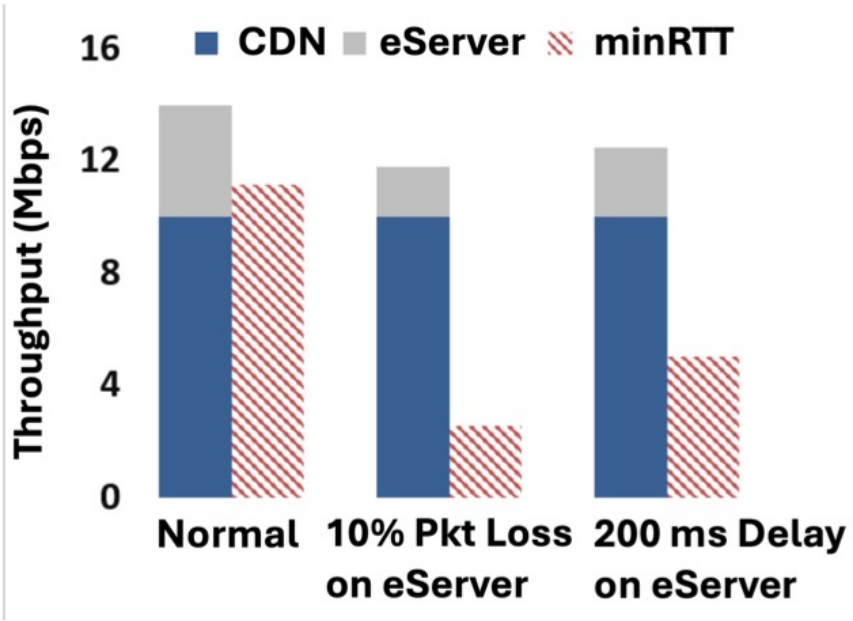
- Setup: CDN and edge server (eServer) instantiated on a single Linux machine, client connected to both on another Linux machine
- minRTT: Multi-Path TCP Scheduler updated for PCDN

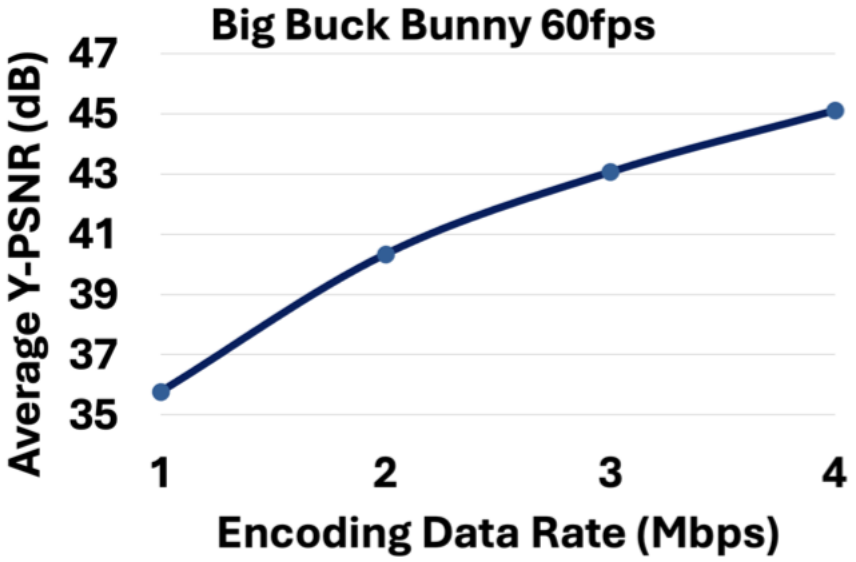




Simple Quantification of HoL Blocking Impact

- Setup: CDN and edge server (eServer) instantiated on a single Linux machine, client connected to both on another Linux machine
- minRTT: Multi-Path TCP Scheduler updated for PCDN





We introduce a multi-path transport solution to address this issue in PCDNs

Outline

- Background and Motivation
- MPRD Architecture
- Hardware and Software Implementation
- Performance Evaluation
  - Through Aggregation (Data Rate)
  - Video Delivery (Stalls and PSNR)

MPRD: Multipath Parallel Reverse Segment Download

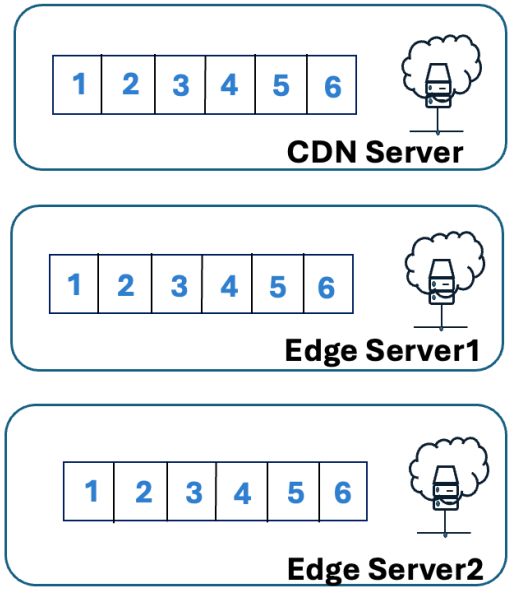
MPRD Architecture

- Does not explicitly use a scheduler or assign packets to paths

MPRD: Multipath Parallel Reverse Segment Download

MPRD Architecture

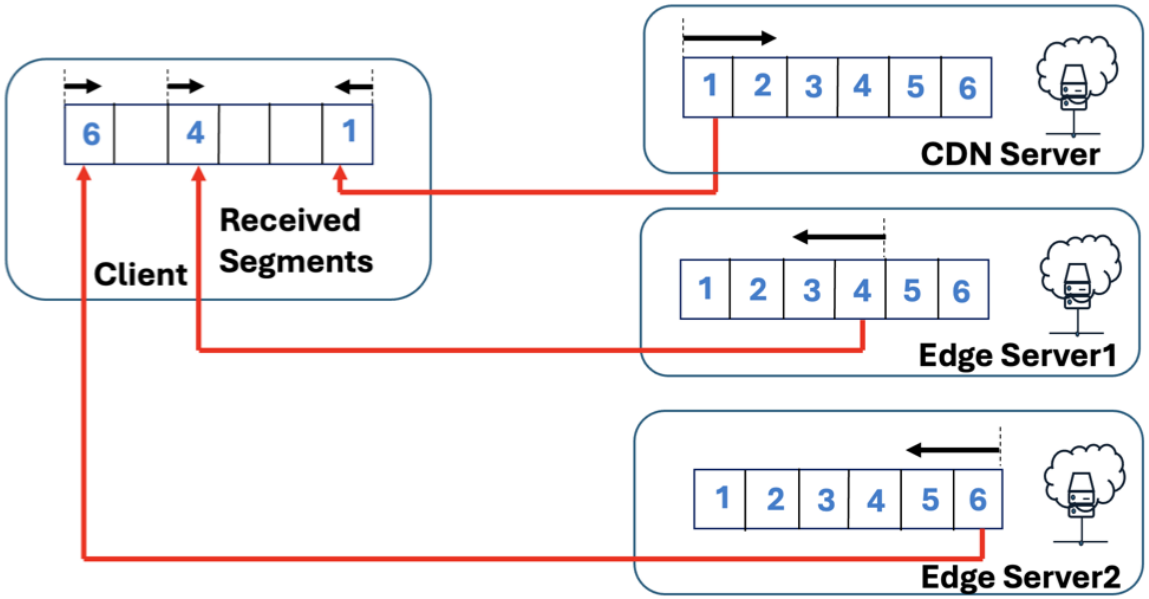
- Does not explicitly use a scheduler or assign packets to paths
- Assume content (e.g., video) broken into fixed segments
- All servers have a horizontal queue loaded with the same segment



MPRD: Multipath Parallel Reverse Segment Download

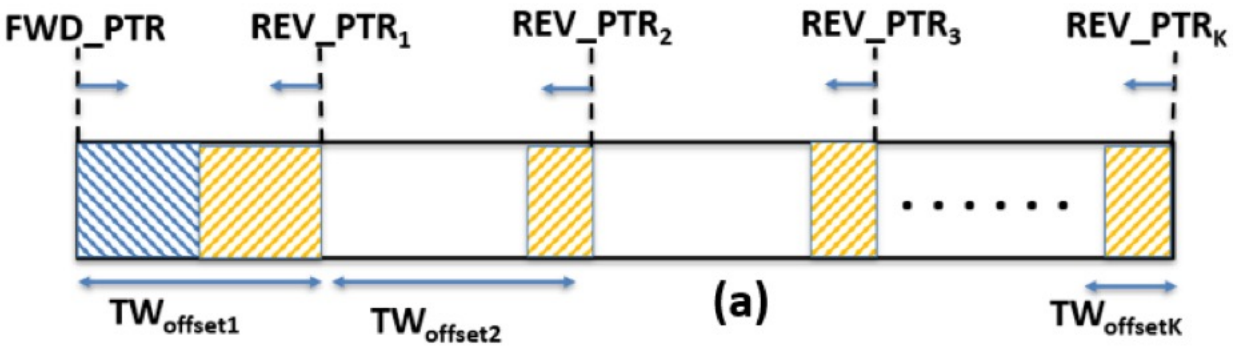
MPRD Architecture

- Does not explicitly use a scheduler or assign packets to paths
- Assume content (e.g., video) broken into fixed segments
- All servers have a horizontal queue loaded with the same segment
- Multiple single path TCP with different starting points and directions



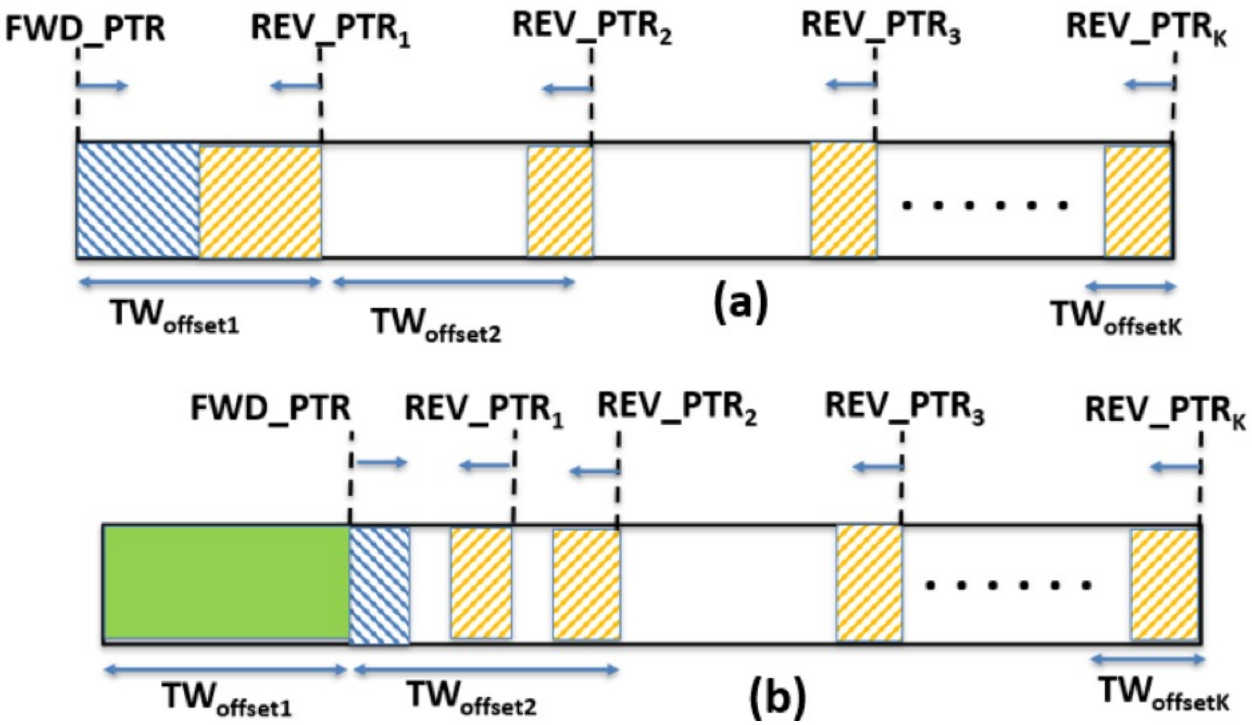
MPRD Detailed Architecture

- Assume K+1 servers, most reliable server assigned to forward direction



MPRD Detailed Architecture

- Assume K+1 servers, most reliable server assigned to forward direction
- Once two pointers start to overlap, they get reassigned
- Duplicated received packets are dropped at the transport layer

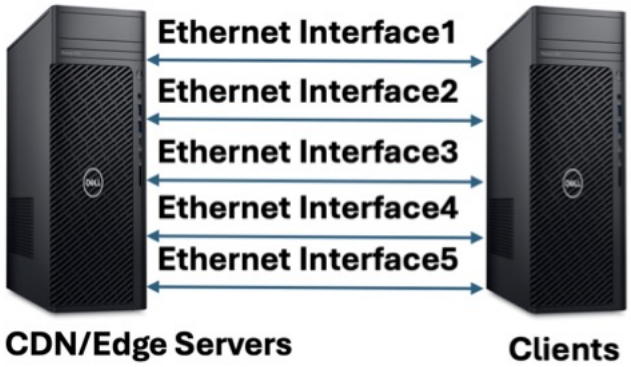


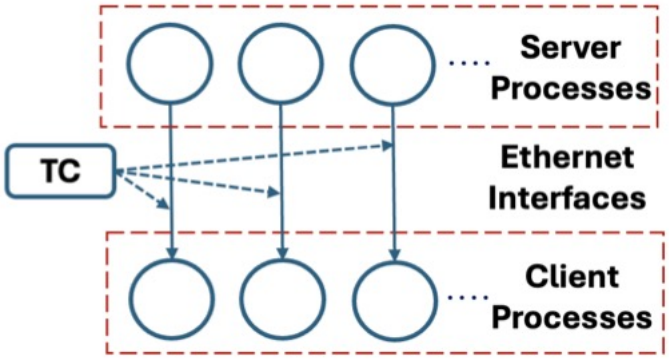
MPRD Properties

- Avoids HoL blocking issue
  - If one path is blocked, another path would take over
- Supports both coupled and uncoupled congestion control
  - Current default is uncoupled congestion control
- Total throughput is close to summation of individual path/server throughput values in isolation
  - Can slightly increase the number of redundant packet transmissions

Hardware Implementation

- Servers and clients initiated on two separate machines
- Up to five paths between servers and clients
- Linux TC controls path bandwidth, delay, and packet loss





Software Implementation

- Generic software for PCDNs using C++ in Linux environment
- Each client process establishes a bidirectional socket connection with the server
  - One socket for downloading the requested content
  - One handles metadata communication such as the content (e.g., video), ID, start sequence number offset, …

Software Implementation

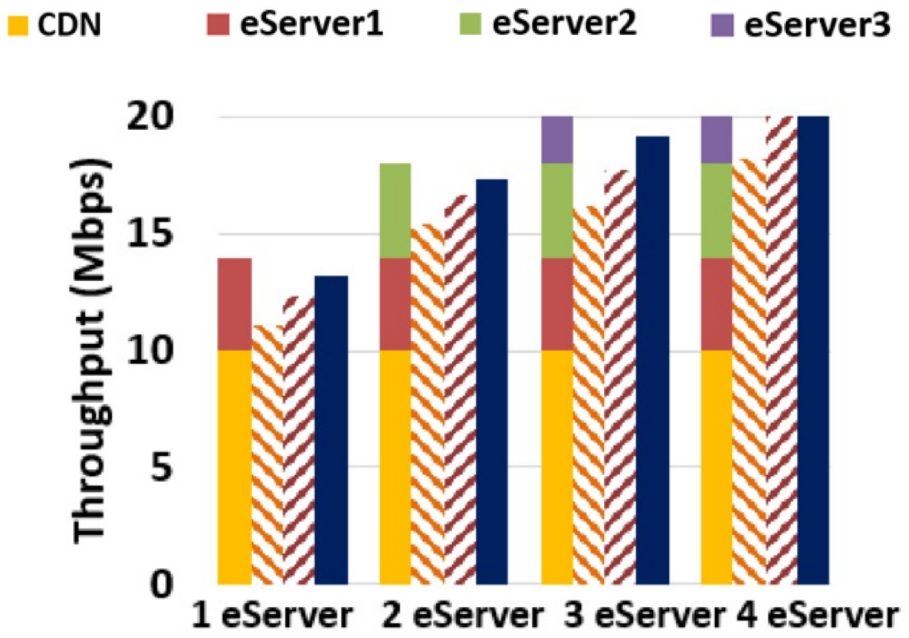
- Generic software for PCDNs using C++ in Linux environment
- Each client process establishes a bidirectional socket connection with the server
  - One socket for downloading the requested content
  - One handles metadata communication such as the content (e.g., video), ID, start sequence number offset, …
- Three protocols and schedulers implemented
  - minRTT: Default MPTCP scheduler in Linux extended to PCDN
  - P-Scheduler (Infocom 2024, ByteDance)
    - P-scheduler estimates path quality using a congestion-control-decoupled algorithm and distributes data by a proposed path-pick-packet method
  - MPRD

Performance Evaluation: Throughput Measurement Setup

- Scenario: Hybrid PCDN+CDN Architecture
  - One or more edge servers augment CDN
  - CDN server has highest data rate to clients
  - Rates from 1 Mbps to 10 Mbps

- The rates were selected to mimic a scenario with lower download speeds. We also experimented with higher speeds in the paper.

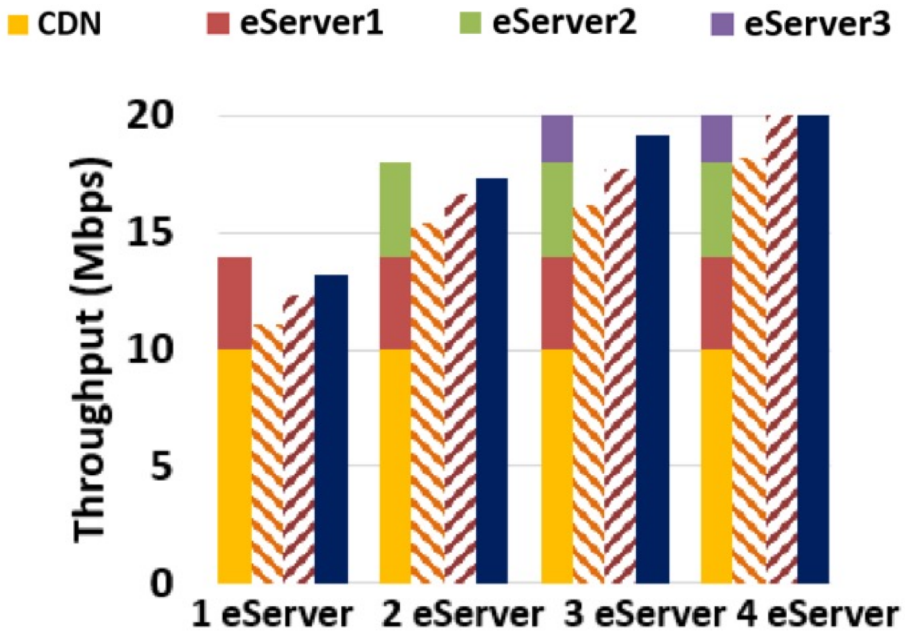
Performance Evaluation: Throughput Aggregation



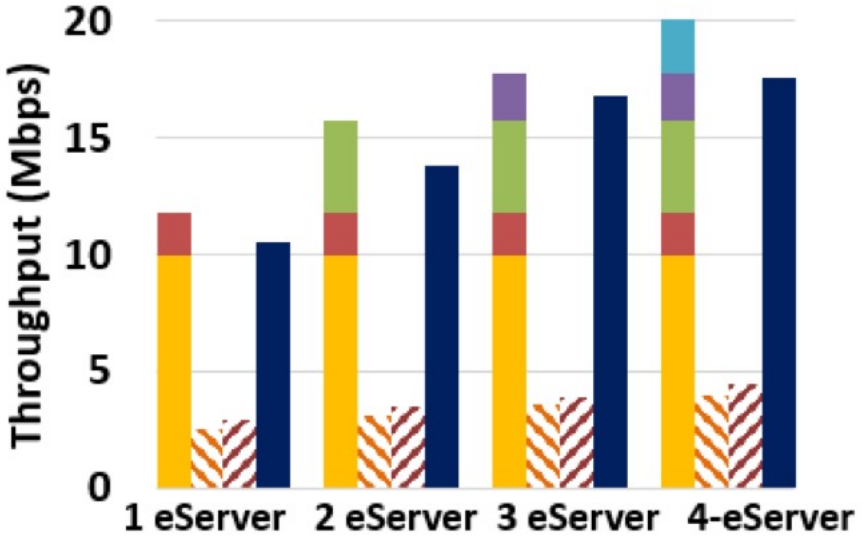
No packet loss

- Without packet loss, minRTT achieves 80%, P-Scheduler 85%, and MPRD 87-90% of summation of individual throughput values

Performance Evaluation: Throughput Aggregation



No packet loss



10% packet loss on edge server one

- Without packet loss, minRTT achieves 80%, P-Scheduler 85%, and MPRD 87-90% of summation of individual throughput values
- With 10% packet loss, minRTT and P-Scheduler collapse, whereas MPRD maintain 70-85% of summation of individual throughput values

Performance Evaluation: Video Preparation Setup

- One CDN server with 10 Mbps rate and one edge server with 4 Mbps rate and 10% packet loss rate
- We used a 5-minute clip from the short film Big Buck Bunny

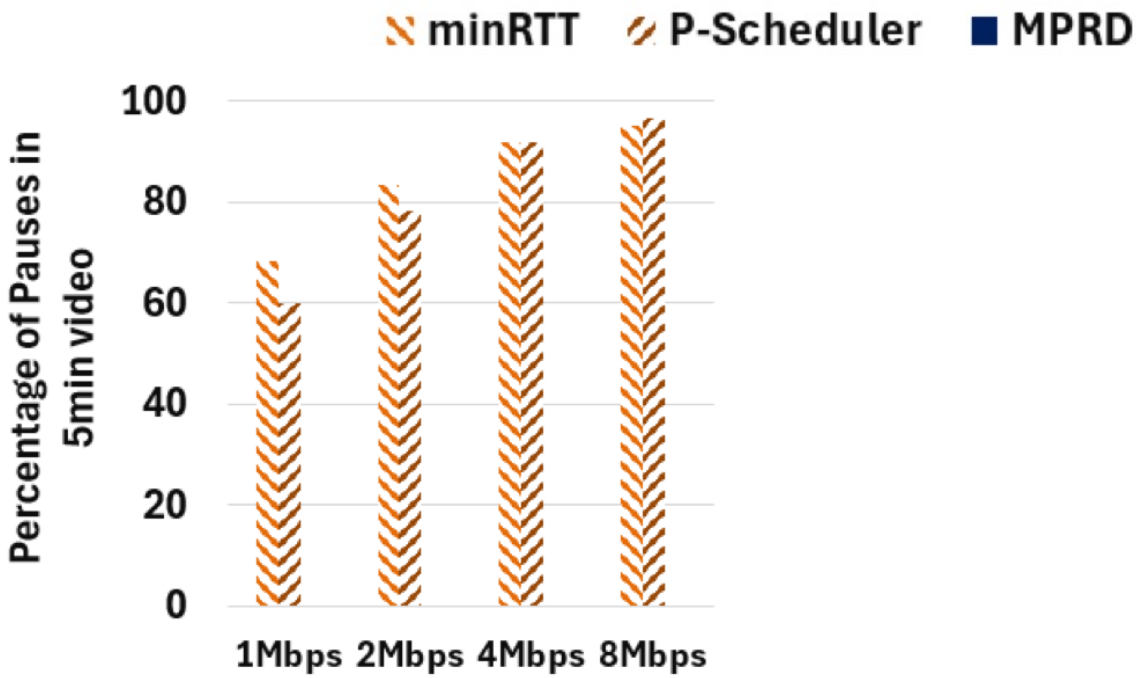
Performance Evaluation: Video Preparation Setup

- One CDN server with 10 Mbps rate and one edge server with 4 Mbps rate and 10% packet loss rate
- We used a 5-minute clip from the short film Big Buck Bunny
- Encoded the video into four representations with bitrates of 1 Mbps, 2 Mbps, 4 Mbps, and 8 Mbps, with resolutions of 360p, 480p, and 1080p (for the two highest bitrates)
- Frame rate set to 60 fps across all representation with a uniform GoP (Group of Pictures) size of 60 frames

Performance Evaluation: Video Preparation Setup

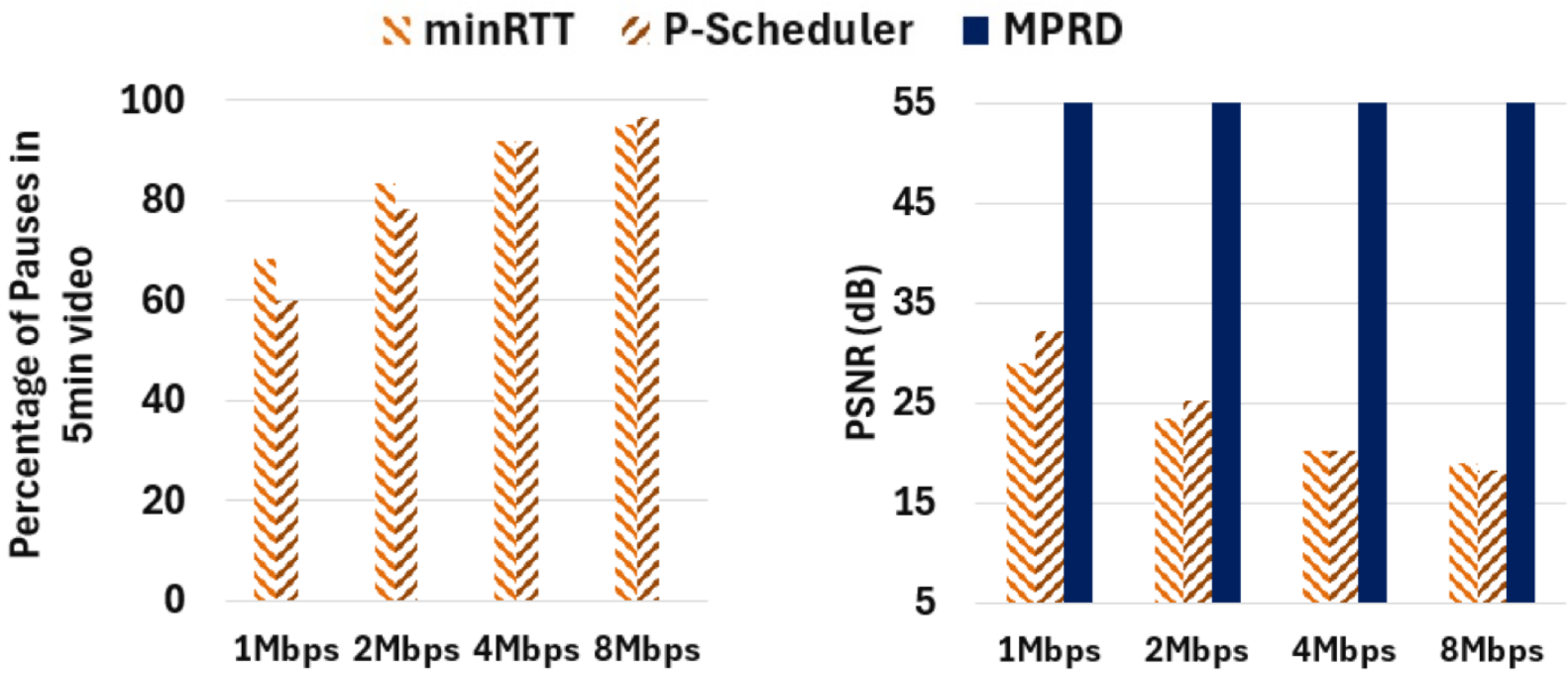
- One CDN server with 10 Mbps rate and one edge server with 4 Mbps rate and 10% packet loss rate
- We used a 5-minute clip from the short film Big Buck Bunny
- Encoded the video into four representations with bitrates of 1 Mbps, 2 Mbps, 4 Mbps, and 8 Mbps, with resolutions of 360p, 480p, and 1080p (for the two highest bitrates)
- Frame rate set to 60 fps across all representation with a uniform GoP (Group of Pictures) size of 60 frames
- Client logged bytes received at one-second intervals and recorded any pauses when the received data fell short of the required amount for a single GOP. We then calculated the percentage of pauses over the entire video duration and assessed video quality by computing PSNR values relative to the original video.

Performance Evaluation: Video Performance



- Significant rebuffering (stalls) with minRTT and P-Scheduler
  - 60-90% of pauses in the 5-minute video

Performance Evaluation: Video Performance



- Significant rebuffering (stalls) with minRTT and P-Scheduler
  - 60-90% of pauses in the 5-minute video
  - Large drop in PSNR when streamed through minRTT/P-Scheduler

In Summary

- MPRD is built for multipath transport from multiple servers to a single client
  - Built on top of single path TCP
- Can be used in a pure PCDN system or when CDN architecture is augmented with low-cost edge devices in PCDN
- MPRD is implemented in Linux userspace
- Significant increase in throughput and video quality delivery when servers have large and fluctuating packet losses/delays