Minimizing Server Throughput for Low-Delay Live Streaming in Content Delivery Networks

F. Zhou, S. Ahmad, E. Buyukkaya, R. Hamzaoui and G. Simon
Live Stream Delivery

Content Provider

encoders

ingest server

origin server

CDN

dge servers

Clients

Recent large-scale live video streaming failed Superbowl Korean Telecom and smart TVs

Minimizing Server Throughput in CDN
Recent large-scale live video streaming failed

- Superbowl
- Korean Telecom and smart TVs
Motivations

Where is the bottleneck in today’s CDN?

[Diagram showing CDN architecture with nodes labeled as origin servers, reflectors, and edge servers, connected through ISPs 1, 2, and 3.]
Motivations

Where is the bottleneck in today’s CDN?

Origin servers
Reflectors
Edge servers
Last-mile?
Motivations

Where is the bottleneck in today’s CDN?

- Origin servers
- Reflectors
- Edge servers
- Last-mile?
- Adaptive streaming → Last-mile
Motivations

Where is the bottleneck in today’s CDN?

- **origin servers**
- **reflectors**
- **edge servers**
- **peering link?**

adaptive streaming → last-mile
Where is the bottleneck in today’s CDN?

- Origin servers
- Reflectors
- Edge servers
- ISP 1
- ISP 2
- ISP 3

Adaptive streaming → last-mile
Edge servers in ISP → peering link?
Where is the bottleneck in today’s CDN?

- Origin servers
- Reflectors
- Edge servers

ISP 1  ISP 2  ISP 3

Adaptive streaming → last-mile
Edge servers in ISP → peering link
The upload capacity equipment bottleneck

CPU

NIC

4 3 5

Gwendal Simon

Minimizing Server Throughput in CDN
The upload capacity equipment bottleneck

Minimizing Server Throughput in CDN
Our proposal

- no cooperation

Objectives:
- Minimizing source throughput
- Maintaining a low transmission delay
Our proposal

Objectives:
- Minimizing source throughput
- Maintaining a low transmission delay

no cooperation

cooperation between nodes
Our proposal

Objectives:

- minimizing source throughput
- maintaining a low transmission delay
Rateless codes

Main advantages:
- Adaptive: no fixed code rate
- Low complexity
- Suitable for multi-source systems (Wu'2008)
Rateless codes

Main advantages:
- adaptive: no fixed code rate
- low complexity
- suitable for multi-source systems (Wu’2008)
Multi-tree delivery (1/2)

Main objective for the delivery of one video chunk:
- minimize the number of trees (packets)

Main constraint in the trees:
- each node should be in $K$ trees
- upload capacity constraint $c$ on nodes
**Main objective** for the delivery of one video chunk:
- minimize the number of trees (packets)

**Main constraint** in the trees:
- each node should be in $K$ trees
- upload capacity constraint $c$ on nodes

\[ K = 3, c = \{2, 2, 3, 1\} \]
Multi-tree delivery (2/2)

Additional constraints

- Do not introduce much delay
  - sum of delays over all paths in any tree below $D$

- Do not introduce much packet loss
  - overall probability of all paths in any tree below $P$
Our contributions

Two algorithms:

1. without last constraints:
   an optimal $O(Kn^2)$ algorithm

2. with limited tree height:
   an efficient $O(Kn^3)$ heuristic
Our contributions

Two algorithms:

1. without last constraints:
   an optimal $O(Kn^2)$ algorithm

2. with limited tree height:
   an efficient $O(Kn^3)$ heuristic

Algorithm performances depend on the context:

- over-provisioned system
  - “source rate = video rate” is achievable

- under-provisioned system
  - source has to compensate missing resources
Simulations

Video and rateless code settings:

- H.264 with bitrates from 320 kbps to 3.2 Mbps
- One chunk is one GOP: 0.5 seconds
- One UDP packet is 1,000 bytes
- Raptor code with redundancy 5%

Network and node settings:

- From 5 to 25 nodes
- Upload capacity follows log-normal distribution
  - Mean capacity is \{512, 1,024, 2,048\} kbps
- Homogeneous packet loss probability and RTT
Scalability

![Graph showing scalability analysis](image)

- Transmission rate (in kbps) vs. number of nodes
- **512 kbps** line indicates the scalability pattern

Minimizing Server Throughput in CDN
Scalability

transmission rate (in kbps)

5,000
4,000
3,000
2,000
1,000
0

number of nodes

5
10
15
20
25

--- 512 kbps
--- 1024 kbps

number of nodes vs. transmission rate
Scalability

transmission rate (in kbps)

number of nodes

512 kbps
1024 kbps
2048 kbps

Minimizing Server Throughput in CDN
Decoding lag

![Graph showing decoding lag with number of nodes and different bitrates](image)

- **512 kbps**
- **1024 kbps**
- **2048 kbps**

**Y-axis**: playback lag (in ms)
**X-axis**: number of nodes

**Title**: Minimizing Server Throughput in CDN
Future works

**Real implementation.** We currently have:
- a fully-developed program that just works
- some contacts with a small CDN company

**Academic work:**
- resource management for multiple flows
- more dynamic algorithms