Architecting Massive Content Delivery

Joe Wenjie Jiang
Jennifer Rexford, Mung Chiang

Princeton EDGE Lab
A Unified Formulation for Wide-Area Content Distribution

Global 1Mbps Video Streaming: Theory & Practice
Network Content Distribution

Network as a graph

\[ G = (V, L) \]

Content distribution problem as \( <S, C, \{M_c\}> \)

- \( S \): servers
- \( C \): clients
- \( M_c \): demand (traffic rate)
Design Variable 1: Server Selection

Contents replicated over multiple servers

Map to one server or split traffic over multiple servers

Variable: $R_{cs}$ amount of client $c$ traffic mapped to server $s$

Goal: performance, load balancing
Design Variable 2: Path Selection

Multiple paths exist between any two nodes

Path selection as a network traffic engineering problem

Variable: $R_{csp}$ amount of traffic on path $p$ that connects $(c,s)$

Goal: min network congestion, performance
Design Variable 3: Content Placement

A large content catalog, but server has limited storage size

Deploy contents to meet content popularity and spatial constraints

Variable: $Z_{sf} = 1$ if content $f$ is stored on server $s$

Goal: performance, demand satisfaction
Design Variable 4: Server Placement

Multiple server/client instances, e.g., virtual machines (VMs), share the same physical node.

Each VM requires certain resources, e.g., CPU, memory.

**Variable:** $X(s) = v$, if VM $s$ is placed on node $v$

**Goal:** locality, max bi-section bandwidth
A Unified Optimization Framework

\[
\begin{align*}
\text{minimize} & \quad \Phi(R, X, Z) \quad \text{Performance cost} \\
\text{subject to} & \quad \sum_{s \in S(c)} R_{cs} = M_c \quad \text{Demand satisfaction} \\
& \quad \sum_{u=X(c), v=X(c), p \in P(u,v)} R_{csp} = R_{cs} \quad \text{Multi-path routing} \\
& \quad \sum_{i : X(i) = v} e_i \leq e_v \quad \text{Physical resource} \\
& \quad \sum_{f \in F} Z_{sf} \leq H_s \quad \text{Storage limit} \\
\text{variable} & \quad R_{csp} \geq 0, R_{cs} \geq 0, \\
& \quad X(i) \in V, Z_{sf} \in \{0,1\} \quad \text{Integral}
\end{align*}
\]
Case Studies
1. Decentralized Server Selection for Cloud Services

“DONAR: Decentralized Server Selection for Cloud Services,” Sigcomm 2010
1. Decentralized Server Selection for Cloud Services

minimize \[ \sum_{n \in N} \sum_{c \in C(n)} \sum_{s \in S(c)} R_{cns} \cdot \text{cost}(s, c) \]  \\
subject to \[ \sum_{s \in S(c)} R_{cns} = M_c \]  \\
\[ R_s = \sum_{n \in N} \sum_{c \in C(n)} R_{cns} \]  \\
\[ R_s \leq B_s \]  \\
\[ \frac{R_s}{B} - w_s \leq \epsilon_s \]  \\
variable \[ R_{cns} \geq 0 \]  \\

Weighted network cost
Demand satisfaction
Bandwidth cap
Server load split within tolerance
Serve selection via mapping node
2. Cooperative CDN and Traffic Engineering

ISP performs traffic engineering to reduce network congestion

Content Distribution Network (CDN) selects close server replica

Both server selection & path selection affect user latencies

An opportunity for collaboration between two separate entities

“Cooperative Content Distribution and Traffic Engineering in an ISP network,” Sigmetrics 2009
2. Cooperative CDN and Traffic Engineering

minimize \[ \sum_{l \in L} \Phi(R_l / C_l) + \alpha \cdot \sum_{(s,c)} R_{cs} \cdot \text{cost}(s,c) \] ... Congestion cost + performance cost

subject to \[ \sum_{s \in S(c)} R_{cs} = M_c \] ... Demand satisfaction

\[ \sum_{p \in P(s,c)} A_{scp} = 1 \] ... Multi-path routing

\[ R_l = \sum_{(c,s) \in l} \sum_{p \in P(s,c)} R_{cs} \cdot A_{scp} \]

\[ R_l \leq C_l \] ... Link capacity

variable \[ R_{cs} \geq 0, A_{scp} \geq 0 \] ... Server selection & path selection
3. Collaborative Caching for Nano Data Centers

User request:
local cache -> remote box -> CDN server
3. Collaborative Caching for Nano Data Centers

minimize \[ \sum_{(c,c')} \sum_{f \in F} R_{cc'}^f \cdot \text{cost}(c, c') + \sum_{f \in F} R_{cs}^f \cdot \text{cost}(c,s) \]  

subject to \[ \sum_{f \in F} p_c^f = H_c \]  
\[ \sum_{c'} R_{cc'} + R_{cs} = \lambda_c^f (1 - p_c^f) \]  
\[ \sum_{f \in F} \sum_{c} R_{cc'}^f \leq N_{c'}B_{c'} \]  
\[ \sum_{c} R_{cc'}^f \leq N_{c'}B_{c'}p_c^f \]  

variable \[ R_{cc'}^f \geq 0, R_{cs}^f \geq 0, p_c^f \geq 0 \]  

User performance cost
Cache size limit
Demand satisfaction
Upload capacity
Content capacity
Content placement & server selection
4. Joint Server Placement and Routing in Data Center Networks

- Layer-2 switch
- Storage
- Tenant VM
  - Red
  - Green
  - Yellow
- Host machine

WAN
4. Joint Server Placement and Routing in Data Center Networks

\[
\text{minimize} \quad \sum_{l \in L} \Phi\left(\frac{R_l}{C_l}\right) \quad \text{Network congestion cost}
\]

subject to
\[
\sum_{i: X(i) = v} e_i \leq e_v \quad \text{Host physical resource}
\]

\[
\sum_{p \in P(i, j)} A_{ijp} = 1 \quad \text{Multi-path routing}
\]

\[
R_l = \sum_{(i, j) \in p: p \in P(i, j)} \sum M_{ij} \cdot A_{ijp}
\]

\[
R_l \leq C_l \quad \text{Link capacity}
\]

variable \( X(i) \in V, A_{ijp} \geq 0 \quad \text{Server placement & path selection} \)
Theory inspires architectural choices:

Centralized vs distributed

Coupling vs functionality separation

Coarse vs fine control granularity

Large vs small timescale
Global 1Mbps Video Streaming: Theory & Practice
P2P content delivery a huge success in the last decade

High definition video presents new challenges for peer-to-peer streaming: 250K-1M streaming rate upgrade

Bottlenecks: uplink bandwidth and stringent latency requirements

FastMesh-SIM: a revisit of architectural consideration, theory-practice proximity
A Hybrid Proxy-P2P Approach

FastMesh-SIM
Gary Chan, Hong Kong University of Science & Technology

FastMesh:
- Packing multiple streaming trees
- **Throughput** and delay optimization
- Distributed and adaptive protocol

SIM:
- **Resilient** streaming tree
- Integrate IP-multicast with overlay multicast
Demo of Web-based Prototype

Streamphony
A Proxy-based P2P Internet Streaming Network

LIVE Channels

Gospel Channel
Bit-rate 312 kbps

CSE Department Introduction
Bit-rate 361 kbps

CSE Department Introduction (Low Quality)
Bit-rate 273 kbps

HKUST Demo
Bit-rate 1119 kbps

Princeton Demo (for testing ONLY)
Bit-rate 1 Mbps

Princeton Demo Low Quality (for testing ONLY)
Bit-rate 273 Kbps

GNCI's channel
Bit-rate 940 Kbps
Connecting Theory and Practice

**P2P Streaming Capacity**
- Viewed as packing of multiple spanning trees
- Assign a bit rate to each spanning tree
- Streaming capacity as sum of rates over all trees

*What is the maximum streaming rate*, given
- each peer’s uplink bandwidth
- node degree constraint
- tree depth constraint

Proximity of FastMesh-SIM and theoretical bounds
- “P2P streaming capacity under node degree bound,” ICDCS’10
- 1Mbps peer-assisted streaming: a global experiment
Architectural Comparison

Theory suggests

We implemented
Inter-Cluster Peering Algorithm

Theory suggests

- Each cluster elects a cluster head that communicates with each other
- Cluster heads form multiple degree bounded trees

We implemented

- Proxies form a mesh as packing spanning trees
- No explicit degree bounded, but considers delay optimization
Inter-Cluster Peering Algorithm

Theory (Bubble Alg.)

Implementation (FastMesh)
Intra-Cluster Peering Algorithm

Theory suggests

- **MutualCast** algorithm: generate multiple one or two hops trees
- Achieves **optimal** streaming rate

We implemented

- Local peers form **resilient** streaming trees: **Scalable Island Multicast** (SIM)
- Integrated with **IP multicast**: one-hop IP-multicast tree
- Achieve **near-optimal** rate
# Performance Comparison

\(N\): total # of peers  
\(K\): total # of clusters  
\(M\): node out-degree bound

<table>
<thead>
<tr>
<th></th>
<th>Theory</th>
<th>FastMesh-SIM</th>
</tr>
</thead>
</table>
| **Streaming rate**  | Inter-cluster: \(\frac{1}{2}\)-approx.  
Intra-cluster: **optimal** | Achieves 85% of upper bound  |
| **Playback delay**  | \(O(N/K+M)\log_M K\)          | Delay bounds less tighter due to protocol overheads. |
| **Node degree**     | \(O(\log N)\)                 | Not guaranteed, \(\leq 7\) in practice |
| **Robustness upon churns** | Split and merge clusters.  
Need re-optimization. | Assume stable proxies.  
Highly **adaptive** tree re-construction. |
| **Computation**     | Centralized                   | Distributed                   |
A Global Deployment

42 proxies, ~100 peers
Dedicated nodes contributed by our collaborators
IP-multicast capitalized on
P2P Streaming Capacity (15 Proxies)

<table>
<thead>
<tr>
<th></th>
<th>Streaming Rate (Kbps)</th>
<th>Max Node Out-degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal</td>
<td>1506</td>
<td>15</td>
</tr>
<tr>
<td>Bubble Alg.</td>
<td>1199</td>
<td>4</td>
</tr>
<tr>
<td>FastMesh</td>
<td>1000</td>
<td>7</td>
</tr>
</tbody>
</table>
Multi-tree Substream Rate

![Graph showing substream rate comparison between Bubble Alg. and FastMesh. The graph illustrates the streaming rate (Kbps) for different substreams, with Bubble Alg. depicted in blue and FastMesh in green. The rate varies across substreams, with Bubble Alg. showing a higher peak in the first substream and FastMesh maintaining a more consistent rate across substreams.]
Rate-Delay Tradeoff

![Rate-Delay Tradeoff Graph]

- Bubble Alg.
- FastMesh

**Graph Details:**
- **Y-axis:** Max Node Delay (ms)
- **X-axis:** Substream Rate (Kbps)
- Data points indicate the relationship between substream rate and max node delay, with indications for Bubble Alg. and FastMesh.
Summary

High-level architecture inspired by theoretical results: proxy-based peer-assisted streaming

Bridging theory-practice gap: rate vs delay tradeoff by metric definition and joint optimization
Acknowledgement

Hong Kong University of Science and Technology

Technicolor Paris Lab

HP Labs

Collaborators:

Gary Chan, Tony D. Ren, Patrick Wendell, Mike Freedman, Stratis Ioannidis, Laurent Massoulie, Mike Schlansker, Yoshio Turner, Jean Tourrilhes, TianLan, Sangtae Ha, Minghua Chen