Delay Optimization for Multi-source Multi-channel Overlay Live Streaming

Jie Dai
Department of Computer Science and Engineering, HKUST

Part of this work is published in Proc. IEEE ICC, London, UK, pp. 8587-92, 8-12 June 2015

Outline

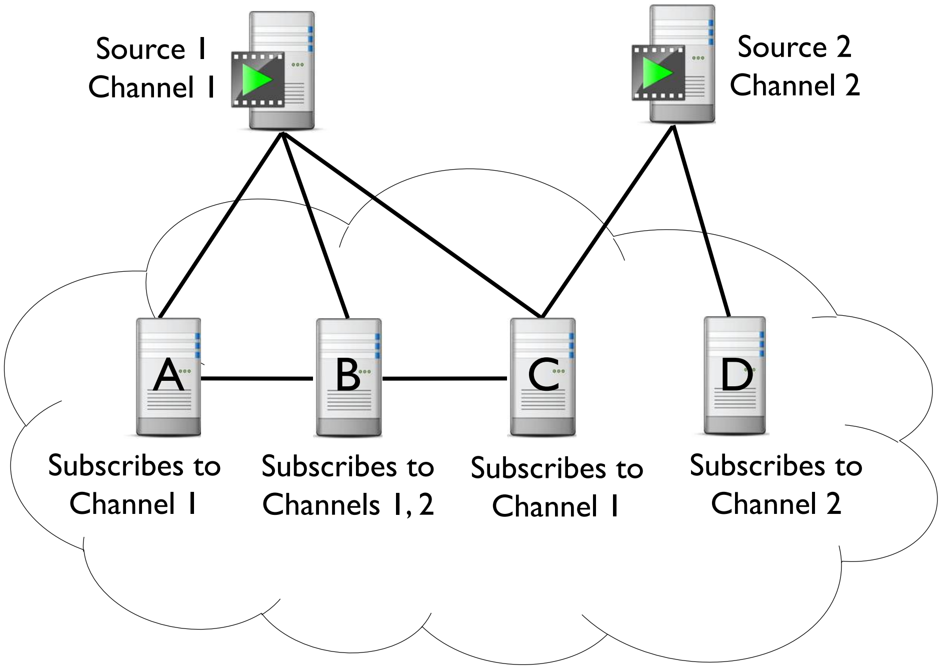
- Introduction
- Problem Formulation
- COMMOS: A heuristic for collaborative multi-source multi-channel overlay streaming
- Simulation Results
- Q & A

A Live Streaming Cloud

- Live streaming services
  - 11.4 % of the network traffic contributes to live streaming services in 2014
  - Live streaming traffic grew 47% in 2014 and will continue to grow at a rapid pace, increasing fourfold by 2019
  - It is expected to reach 14% in 2019 [Cisco et al. The Zettabyte Era—Trends and Analysis'15]
- To serve **geo-dispersed users**, a content provider often deploys a content distribution network (CDN)

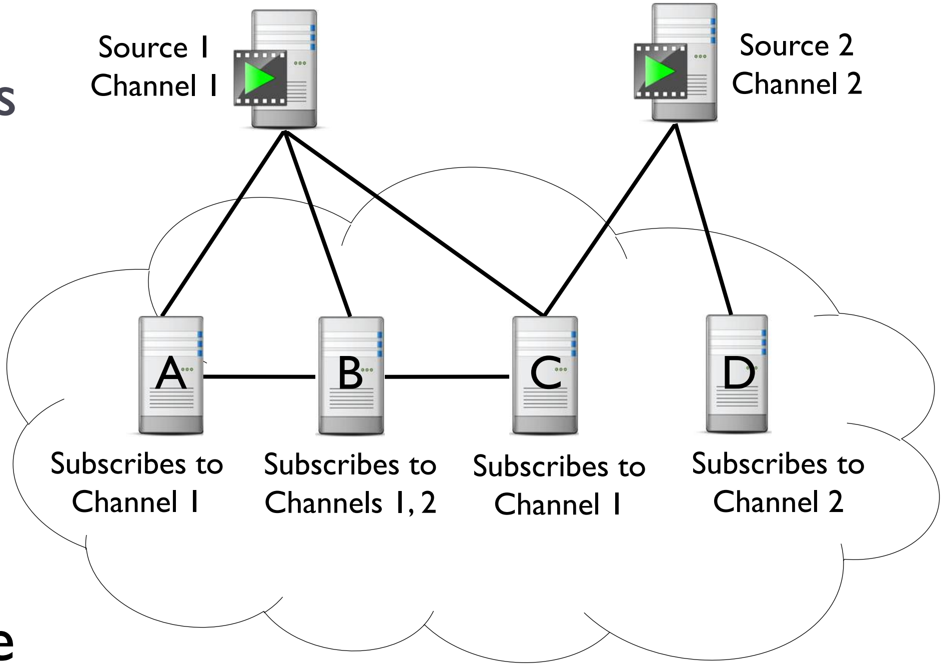
CDN for Live Streaming

- Multiple sources
  - Each source originates one or more video channels
- Multiple video channels
  - Heterogeneous streaming rate
  - Each channel stream is divided into multiple substreams of a certain bit-rate
    - Better utilization of network bandwidth as compared with the single-stream approach



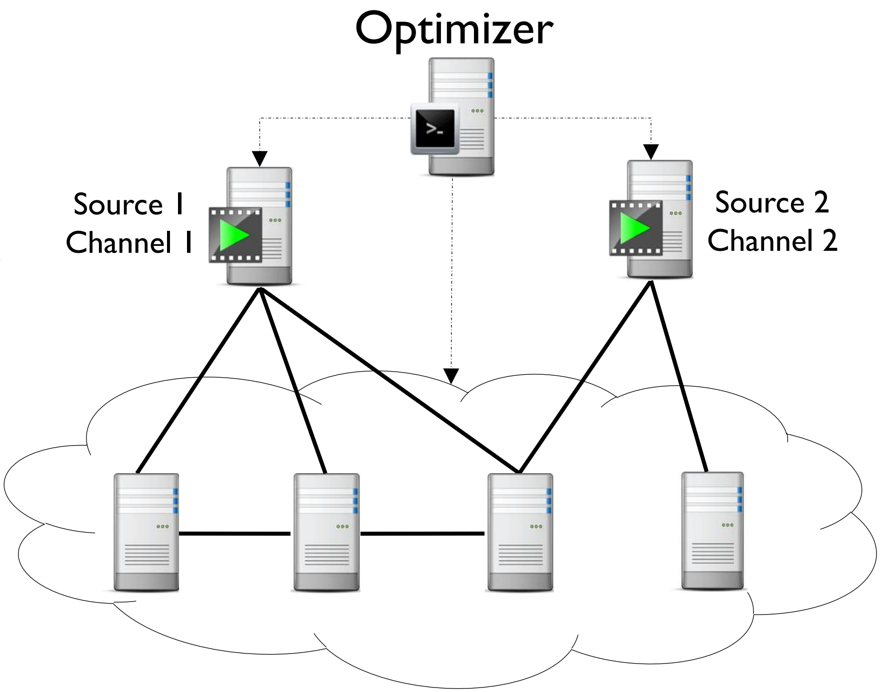
CDN for Live Streaming

- Multiple servers
  - A server subscribes to a subset of channels for its local user pool
  - It has to collect all the substreams of its subscribed channels
  - A server may help *relay* un-subscribed channel substreams
    - Increase the availability of network bandwidth
    - Provide rich path diversity
- Each substream is pushed from the source to the servers via a delivery tree



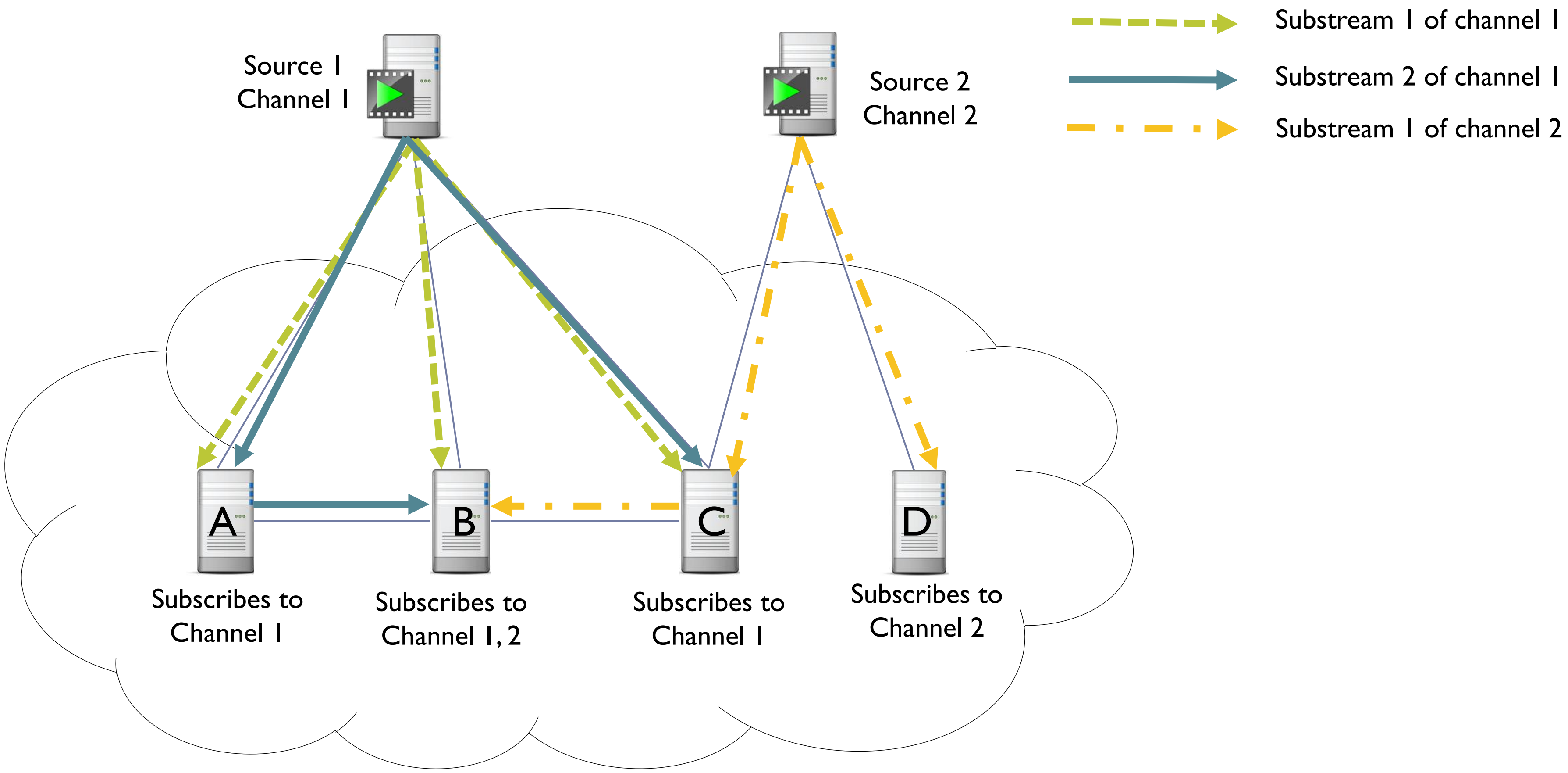
Optimizer in the CDN

- Servers are connected by subscribed links
  - Forming a mesh network
- Bandwidth is limited
  - Each source or server has limited uploading capacity
  - End-to-end link capacity between any pair of nodes is limited
  - Bandwidth is shared among all the channels
- An optimizer computes streaming topology
  - Sources and servers send their network information to the optimizer
  - The optimizer computes substream trees and informs the servers of the optimized topology



An illustration on substream trees

- Channel 1 is of 2 substreams, Channel 2 is of 1 substream



Source-to-End Delay

- The delay from the source to the subscribing servers is an important QoS measure in live streaming

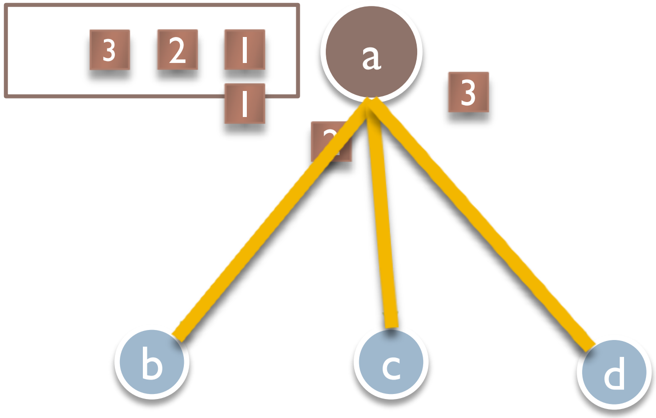
- The construction of substream trees affects the delay from the source to the subscribing servers
  - Given by the longest path from the source to the subscribing servers via different substream trees

Packet Delay Components

- Propagation delay
  - The time that a packet travels from one server to the next over a physical connection
  - Usually reflected by the round-trip time (RTT)
- Scheduling delay at a parent to dispatch packets to its children
  - Due to round-robin scheduling of packets at the parent to its children
  - The delay between the instant the parent fully receives a packet to the instant that the packet fully departs from the node for its child
- Total source-to-end delay
  - The summation of propagation delay and scheduling delay from the source to the end nodes

An illustration of scheduling delay

Scheduling Delay		
Packet 1	**Packet 2**	**Packet 3**
d_{ab}	$d_{ab} + d_{ac}$	$d_{ab} + d_{ac} + d_{ad}$



- Some packets arrive at the node sequentially initially
- Before the packet is sent out, it has to wait for the other packets departing from the node

Research Objective and Methodology

- Construct trees for all the substreams to minimize the maximum delay of all the channels
  - Min max source-to-end delay
- Constraints:
  - The server uploading capacity
  - The end-to-end link capacity between pairs of nodes

Contributions

- Problem formulation and NP-hardness proof
  - Capturing all the essential network parameters and delay components
  - Travelling salesman problem is reducible to the problem in polynomial time
- A novel algorithm: COMMOS
  - Collaborative multi-source multi-channel overlay streaming
  - Minimize the maximum channel delay
- Extensive simulation results
  - Based on real Internet topologies
  - Our algorithm significantly reduces the total delay

Related Works

- Single-source single-channel [Azarpira et al. IST'12, Zhuang et al. ISPA'11, Magharei et al. TON'09, Ren et al. TMM'09]
  - Cannot be extended to multi-source multi-channel live streaming due to bandwidth sharing

- Multi-source multi-channel work
  - VoD based approaches cannot be extended to live streaming due to delay consideration [Zhao et al. INFOCOM'14, Tan et al. NET'13, Wu et al. PDS'12, Li et al. ICACT'10]
  - No consideration on the use of helpers [Kondo et al. NETWORKS'14, Liu et al. ICCCN'13, Meskill et al. LCN'11, Wu et al. INFOCOM'07]
  - Other live streaming works mainly focus on different objectives, such as maximizing bandwidth or minimizing source load [Wu et al. NET'11, Wang et al. INFOCOM'10, Wu et al. INFOCOM'09, Wu et al. INFOCOM'08, Wang et al. GLOBECOM'10, Wu et al. PDS'08]

Outline

- Introduction
- **Problem Formulation**
- COMMOS:A heuristic for collaborative multi-source multi-channel overlay streaming
- Simulation Results
- Q & A

Modeling Delay Components

- The propagation delay between servers *i* and *j*: d_{ij}^{Prp}
- The worst-case scheduling delay at parent *i*: d_i^{Sch}
  - Simply the round-robin cycle time, which is the sum of the packet transmission times from the parent to its children for all trees
  - Affected by the number of trees a link participates, link capacity between i and each of its children, *i*'s uploading capacity and the out degree of server *i*
  - $$d_i^{\mathrm{Sch}} = \sum_{k \in \boxed{C(i)}} \frac{\boxed{L} \cdot \boxed{t_{ik}}}{\min(\boxed{b_{ik}}, \boxed{U_i})}$$

Packet size ⇓

⇐ Number of trees that edge $\langle i, k \rangle$ participates

⇐ Uploading capacity of node i

⇑ The set of all children of node i in all trees

⇑ Link capacity of edge $\langle i, k \rangle$

Maximum Channel Delay in the Network

- Given channel *m* and its substream *k*, the delay of node *j* from the source is the sum of:
  - The delay of its parent node *i*: $D_i^{(mk)}$
  - The worst-case scheduling delay of its parent node: d_i^{Sch}
  - The propagation delay: d_{ij}^{Prp}

$$D_j^{(mk)} = D_i^{(mk)} + d_i^{\mathrm{Sch}} + d_{ij}^{\mathrm{Prp}}$$

- The delay of node *i* subscribing to channel *m* is the maximum delay of all substream trees of the channel:

$$D_i^{(m)} = \max_k D_i^{(mk)}$$

- The **channel delay** of channel *m* is the maximum delay of the nodes subscribing the channel

$$D^{(m)} = \max_i D_i^{(m)}$$

Minimum-Delay Streaming with Server Collaboration (**MDSSC**)

- Objective: minimize the maximum channel delay (also defined as the streaming diameter)

$$\min \boxed{D} = \min \max_{m\in\mathbf{M}} \boxed{D^{(m)}}$$

Streaming diameter

Channel delay of channel m

Subject to:

- Link bandwidth constraint: $0 \le t_{ij} \le \min\left(b_{ij}, U_i\right), \forall\langle i,j\rangle \in \mathbf{E}$
- Uploading capacity constraint: $\sum_{j\in C(i)} t_{ij} \le U_i \ , \forall\langle i,j\rangle \in \mathbf{E}$
- Each node should receive all the substreams of the channels that it subscribes to: $\boxed{a_i^{(mk)}} = 1, \forall k \in \mathbf{K}^{(m)}, \forall \mathrm{m} \in \mathbf{M}, \forall i \in \mathbf{P}$

An indicator variable indicates whether node i is in tree $T^{(mk)}$

- For continuity and stability, the inter-arrival time of packets cannot exceed the worst-case scheduling delay: $\frac{L}{\boxed{\tau}} \ge d_i^{\mathrm{Sch}}, \forall i \in \mathbf{V}$

Streaming rate of each substream

MDSSC is NP-hard

- The Travelling Salesman Problem (TSP) is reducible to MDSSC problem in polynomial time
  - TSP is a well-known NP-hard problem

- Considering:
  - Only one source
  - Only one channel with 1 unit streaming rate
  - The uploading capacity of each node is 1 unit and the bandwidth capacity of each link is 1 unit
  - Add a node with zero uploading capacity and edges from all nodes to it.
  - The TSP instance is thus reduced to a MDSSC instance
  - The delay at the last node is the maximum delay of the network. Such delay is minimum if and only if the delay in the Hamiltonian cycle is minimum.

Outline

- Introduction
- Problem Formulation
- **COMMOS: A heuristic for collaborative multi-source multi-channel overlay streaming**
- Simulation Results
- Q & A

COMMOS Overview

- Initialization
  - Each delivery tree is initialized to contain only the source
  - The streaming diameter is initialized as zero

- Add one server into one delivery tree in a way that such connection greedily minimizes the diameter increment of the overlay
  - For each node not included into the trees, compute the diameter of the network if it were to join
  - Choose the node and the corresponding delivery tree so that the increment of streaming diameter is the minimum
  - Use helper to relay the substream whenever necessary

- Until each server has joined all the delivery trees that it subscribes to

Adding a node into a delivery tree

- We consider two ways:
  - 1. Directly connect an existing node in the delivery tree
  - 2. Indirectly connect to an existing node through a helper
- Choose the connection such that the increase in streaming diameter is minimum

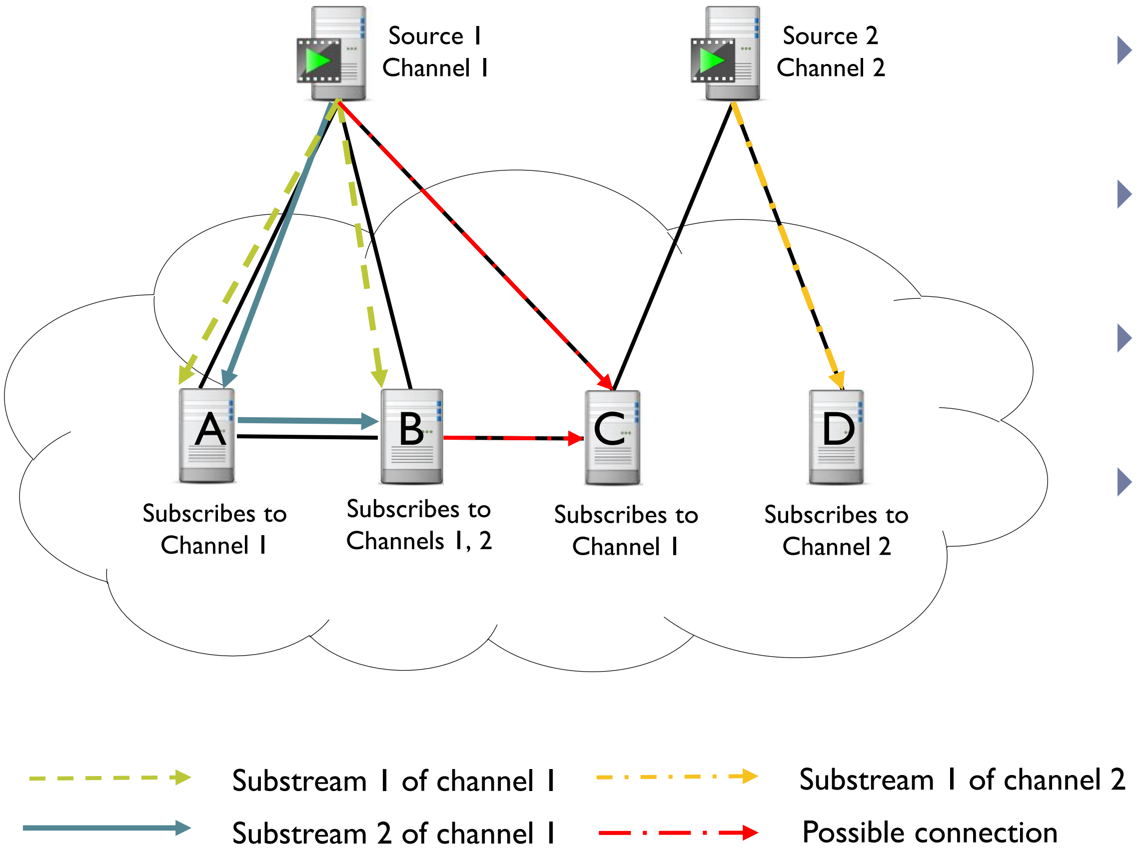
1. Directly connect a parent in the delivery tree

- Calculate the delay of the new node in the tree
  - The connection incurs an extra scheduling-delay to its parent
- Recalculate the delays of all the descendents of the parent
- Calculate the increase in streaming diameter
- Iterate through all the nodes in the delivery tree and select the parent such that the increment delay is the minimum

2. Use a helper to relay the substream from the parent

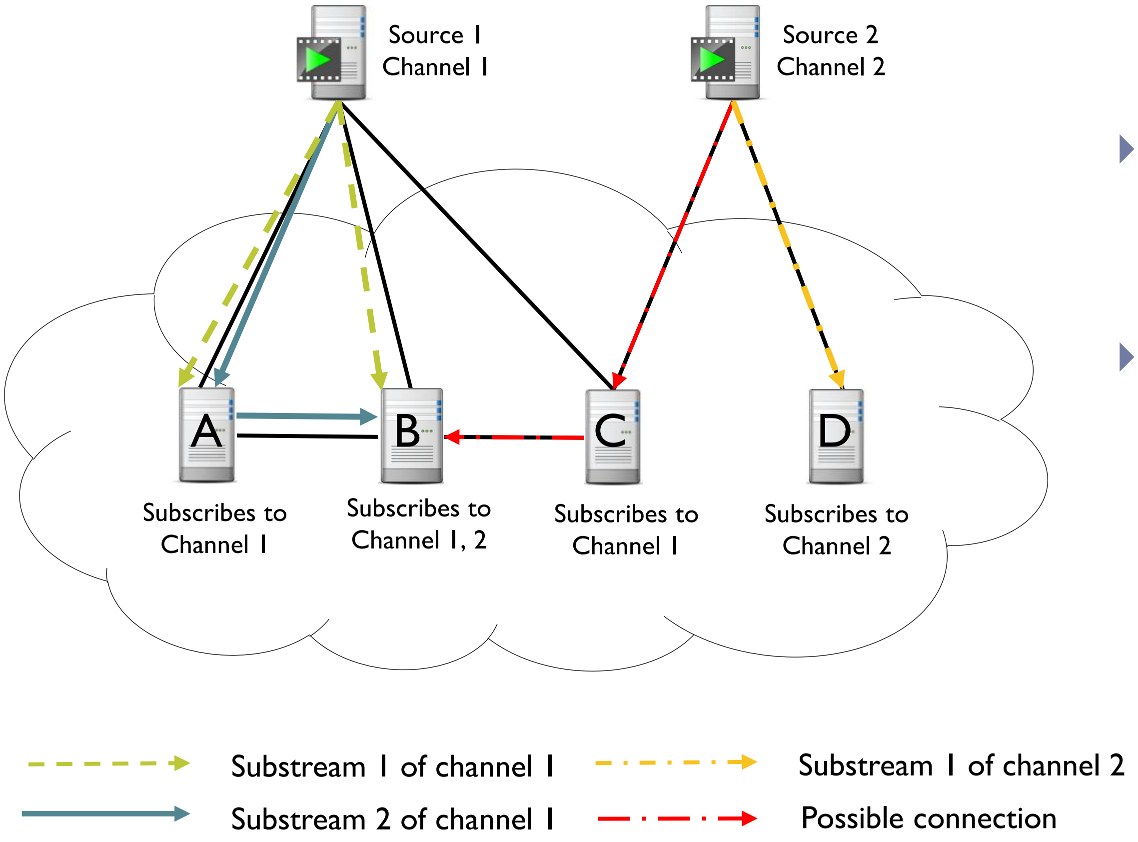
- Iterate through all the nodes that are not included into the delivery tree
  - Let the node be the helper
  - The helper relays the substream from the parent to this new node
- Calculate the source-to-end delay of the helper and the new node
  - The connection incurs an extra scheduling-delay to both the parent and the helper
- Recalculate the delays of all the descendants of the parent and the helper
- Calculate the increase in streaming diameter
- Iterate through all the parent nodes in the delivery tree and select the parent-helper pair such that the increment delay is the minimum

1. Directly connect to an existing node in the delivery tree



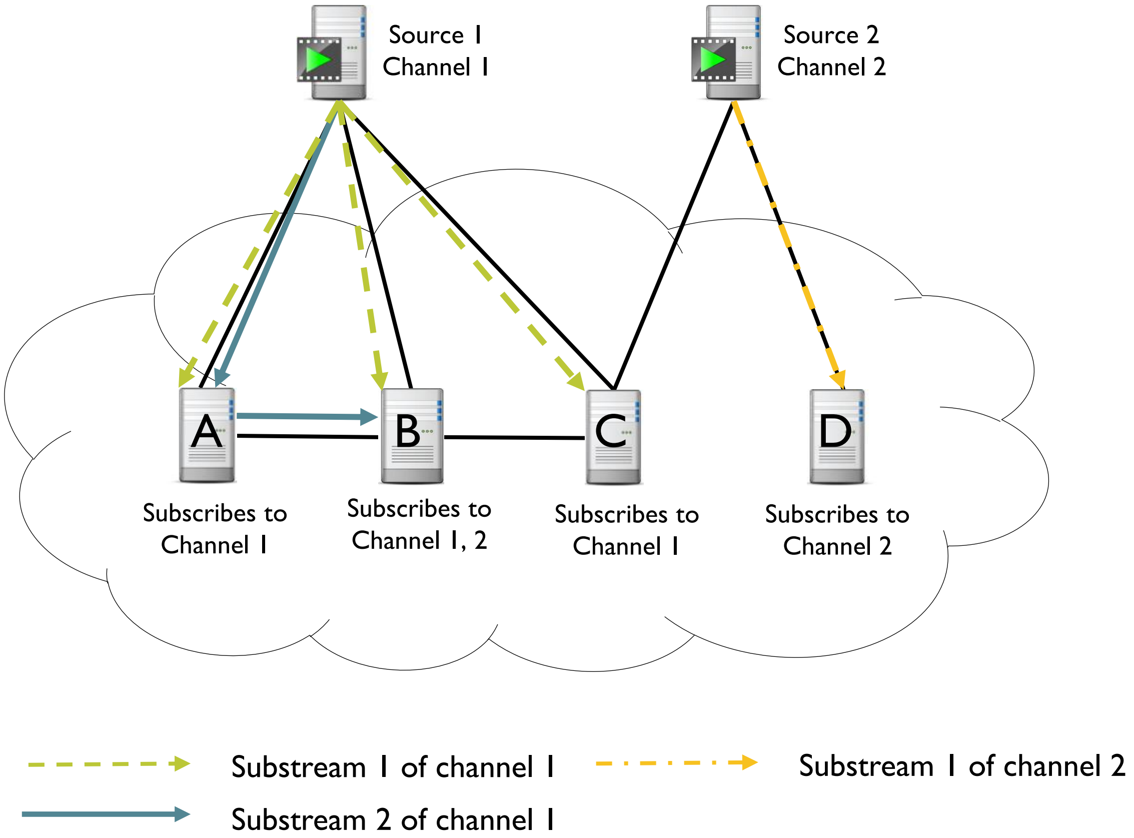
- Currently there are three partial delivery trees
- One server is to be added into the one delivery tree
- Iterate through all possible **direct** connections
- Calculate the increase of the streaming diameter correspondingly

2. Considering using a helper to relay the stream



- Iterate through all possible connections when a helper is used to relay the substream
- Calculate the increase of streaming diameter of all such connections

Choose the server and substream with minimum increment in diameter



- Add the server into the corresponding delivery tree
- Continue adding servers until all the streams are fulfilled

Complexity Analysis

- The overall complexity is $O(|\mathbf{M}|^2|\mathbf{P}|^3|\mathbf{V}|)$
  - $|\mathbf{M}|$ is the number of channels
  - $|\mathbf{P}|$ is the number of servers
  - $|\mathbf{V}|$ is the number of servers and sources

- One node is added into one delivery tree at each round: $O(|\mathbf{M}||\mathbf{P}|^2|\mathbf{V}|)$:
  - Selection of the node and the corresponding delivery tree costs
  - Checking and update of the network resources costs
- There are totally $O(|\mathbf{M}||\mathbf{P}|)$ rounds

Outline

- Introduction
- Problem Formulation
- COMMOS: A heuristic for collaborative multi-source multi-channel overlay streaming
- **Simulation Results**
- Q & A

Simulation Setting

- Real Internet topology provided by CAIDA
- Sources and servers are randomly attached to the routers
- Link capacity is generated by normal distribution
  - Accepting only the positive values
- Each server subscribes a channel based on the Zipf's law

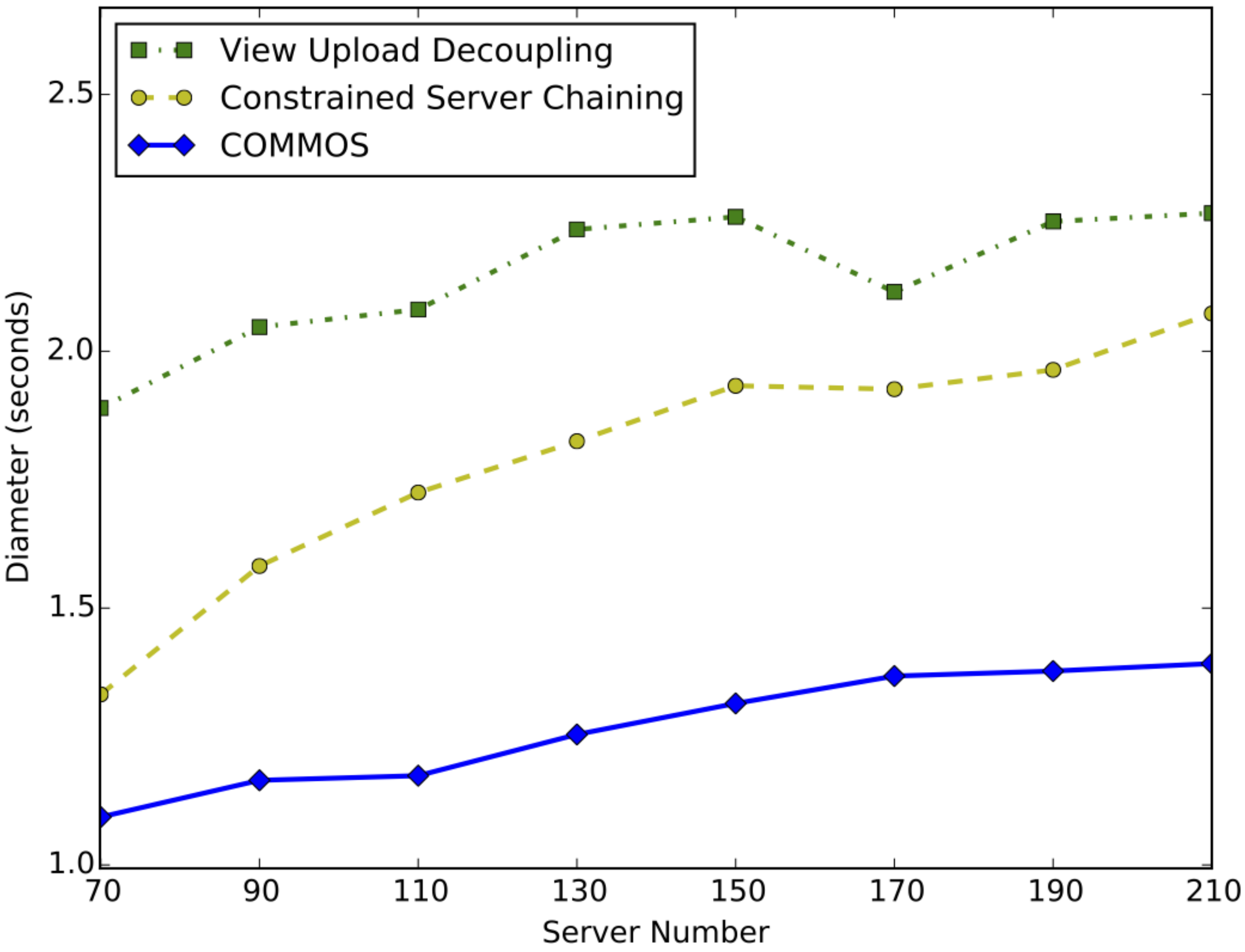
Parameters	Baseline Value
Number of servers	150
Number of channels	20
Streaming rate	1.2 Mbps
Substream rate	400 kbps
Segment size	100 Kbits
Server upload capacity	Mean = 6.5 Mbps Standard deviation = 2 Mbps
Link capacity	Mean = 4 Mbps Standard deviation = 2 Mbps
Zipf Parameter	0.4

Comparison Schemes

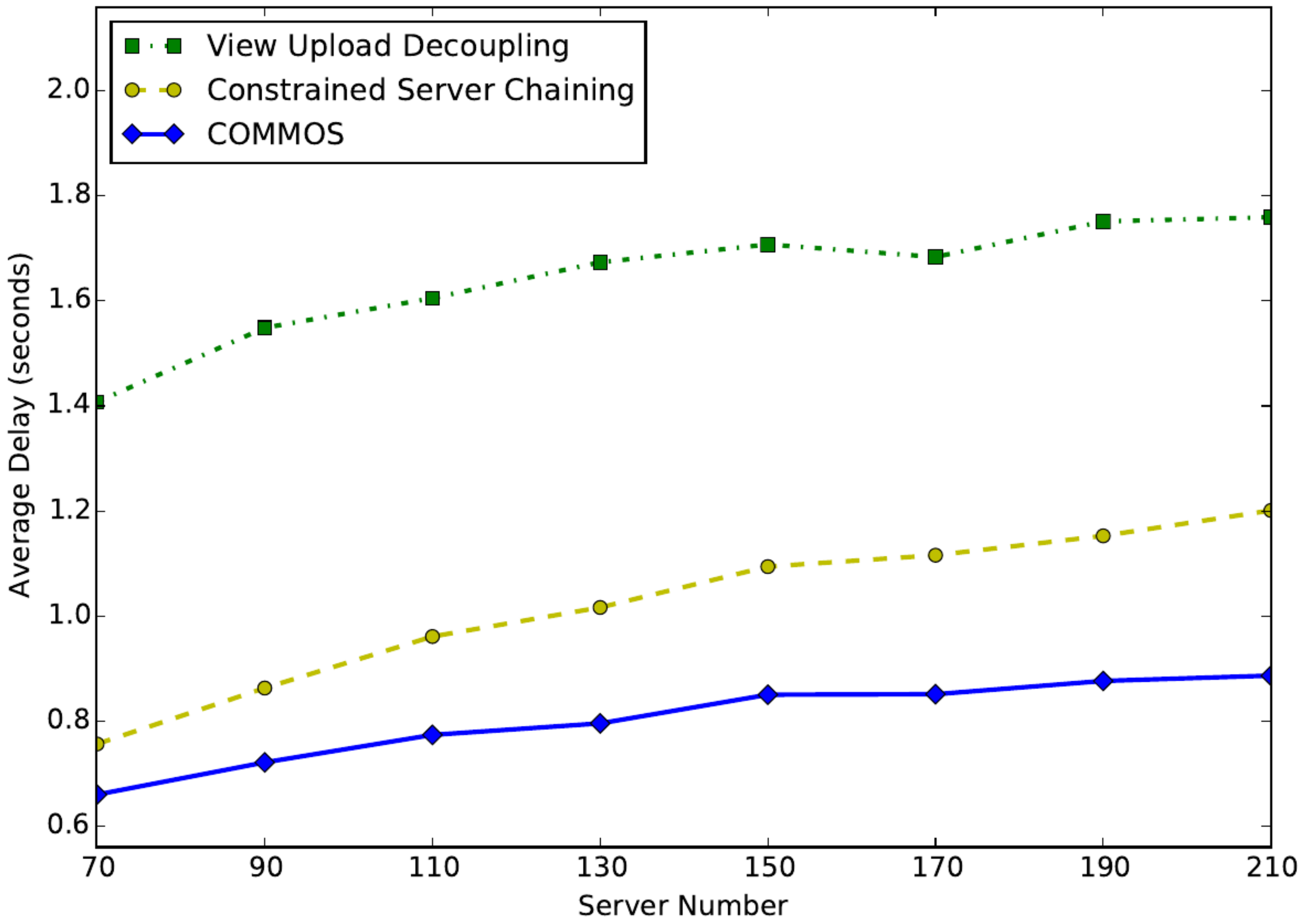
- Constrainted Server Chaining (CSC) [Zhuang et al. ISPA'11]
  - Minimize the source-to-end delay
  - No helper is involved in this scheme

- Video-Upload Decoupling (VUD) [Wu et al. INFOCOM' 09]
  - Focus on P2P multi-channel overlay
  - Each node subscribes to channels independent of the requests of other nodes
  - In COMMOS, each node subscribes to the channels with the objective of minimizing the delays of the network

COMMOS achieves the lowest network diameter

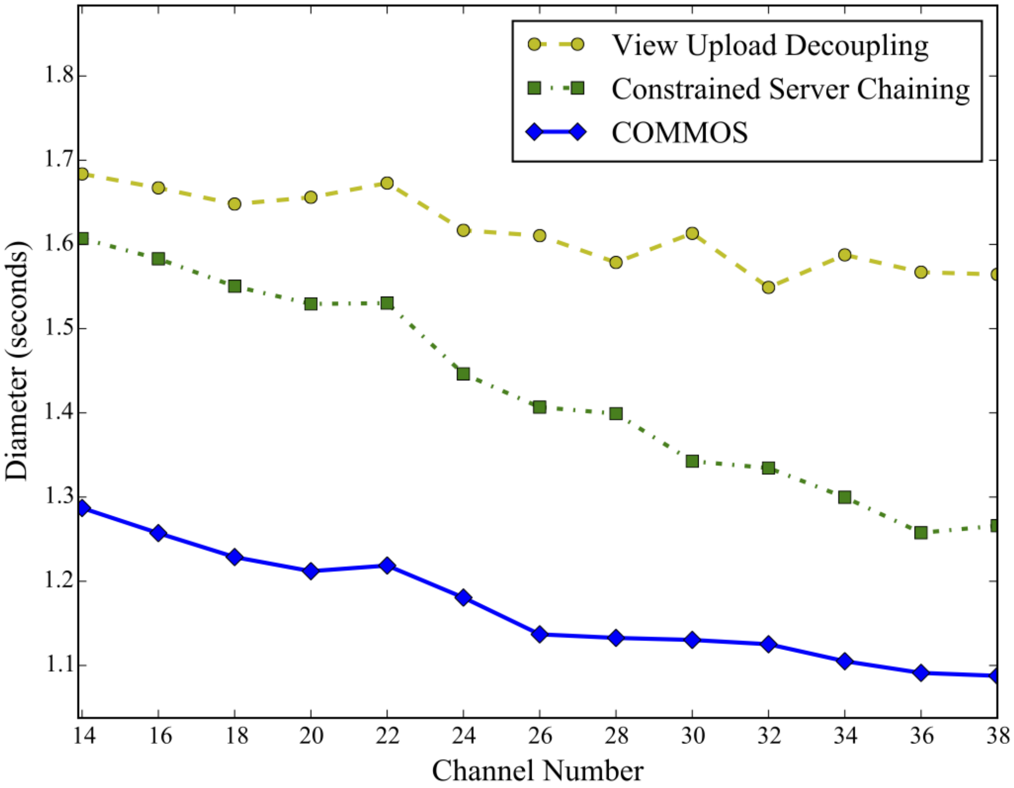


COMMOS achieves the lowest average delay

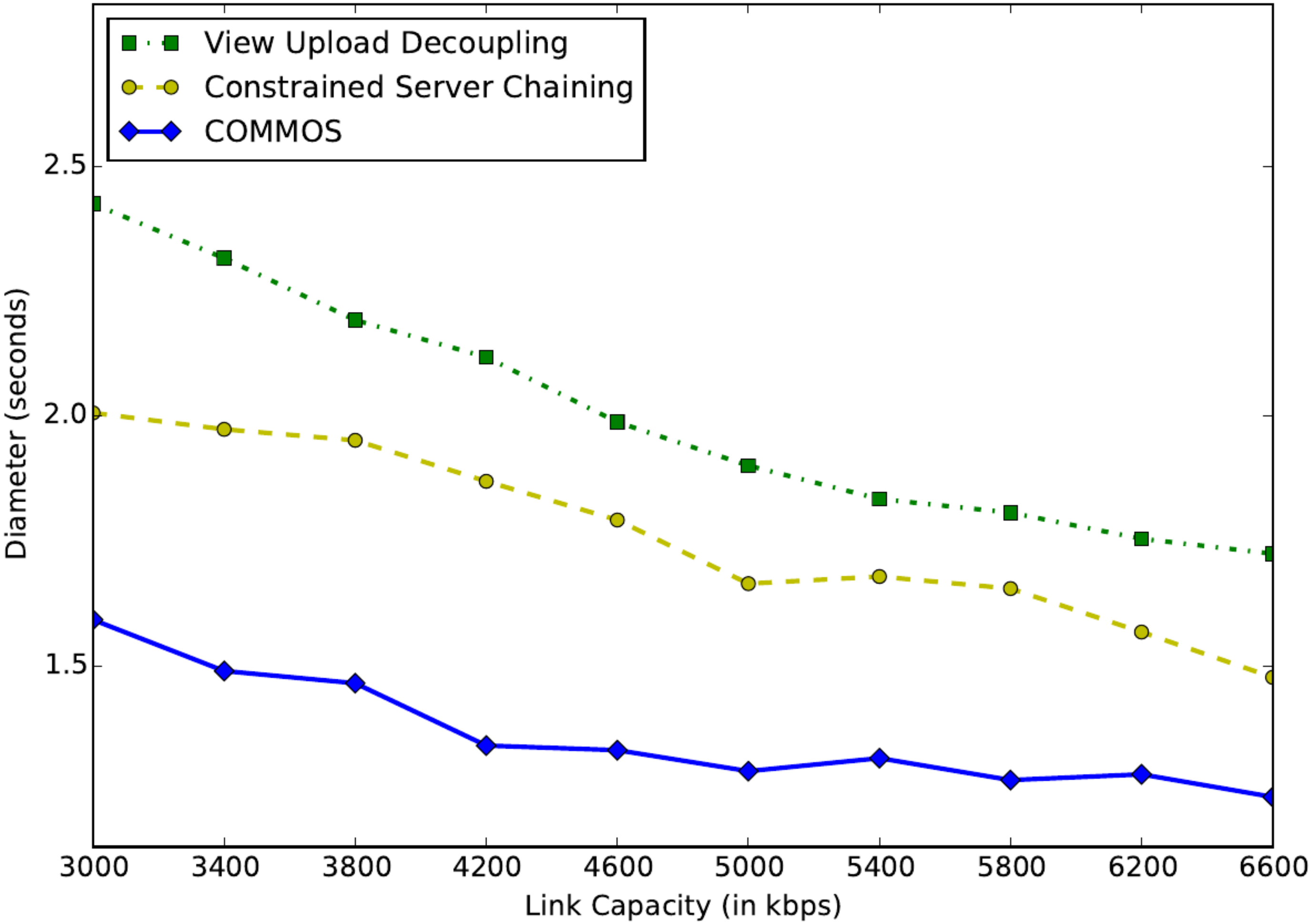


Diameter versus channel number

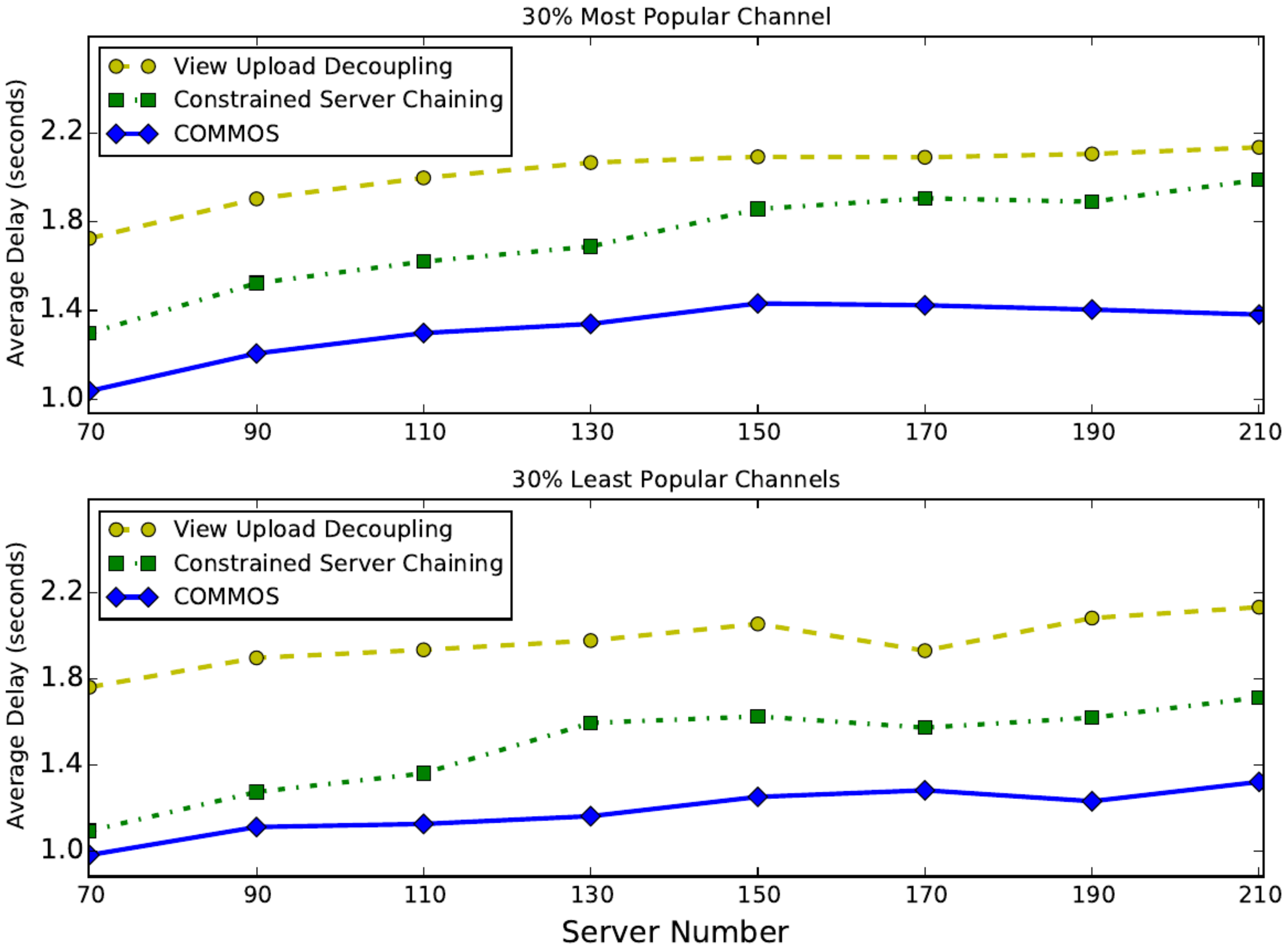
- The number of subscription to each channel declines with rise of channel number
  - The total subscription to all the channels is fixed
  - The depth of each tree and the streaming diameter also decreases
- COMMOS enjoys more performance improvement with fewer channels



Diameter decreases with link capacity



COMMOS performs the best irrespective of channel popularity



Conclusions

- Delay optimization for multi-source multi-channel live overlay streaming
  - Minimizing the streaming diameter of the network
- Novel problem formulation and complexity analysis
- COMMOS: a simple and efficient heuristic
- Simulation results validate the performance of COMMOS

Reference List

- [Cisco et al. The Zettabyte Era—Trends and Analysis'15] Cisco,. 'The Zettabyte Era—Trends And Analysis'. N.p., 2015.
- [Azarpira et al. IST'12] Hamed Azarpira and Saleh Yousefi. On optimal topology in hierarchical P2P live video streaming networks. In *Sixth International Symposium on Telecommunications (IST)*, pages 644–649. IEEE, 2012.
- [Zhuang et al. ISPA'11] Zhenyun Zhuang and Chun Guo. Optimizing CDN infrastructure for live streaming with constrained server chaining. In *9th International Symposium on Parallel and Distributed Processing with Applications (ISPA)*, pages 183–188. IEEE, 2011.
- [Magharei et al. TON'09] Nazanin Magharei and Reza Rejaie. Prime: Peer-to-peer receiver-driven mesh-based streaming. *IEEE/ACM Transactions on Networking (TON)*, 17(4):1052–1065, 2009.
- [Ren et al. TMM'09] Dongni Ren, Y.-T. Hillman Li, and S.-H. Gary Chan. Fast-mesh: A low-delay highbandwidth mesh for peer-to-peer live streaming. *IEEE Transactions on Multimedia*, 11(8):1446–1456, December 2009.
- [Zhao et al. INFOCOM'14] Yuhong Zhao, Hong Jiang, Ke Zhou, Zhijie Huang, and Ping Huang. Meeting service level agreement cost-effectively for video-on-demand applications in the cloud. In *INFOCOM, 2014 Proceedings IEEE*, pages 298–306, April 2014.
- [Tan et al. NET'13] Bo Tan and Laurent Massouli´e. Optimal content placement for peer-to-peer videoon-demand systems. *IEEE/ACM Trans. Netw.*, 21(2):566–579, April 2013.
- [Wu et al. PDS'12] Weijie Wu and John Lui. Exploring the optimal replication strategy in p2p-vod systems: Characterization and evaluation. *Transactions on Parallel and Distributed Systems*, 23(8):1492–1503, 2012.
- [Li et al. ICACT'10] Xia Li, Rua Zou, Xinchao Zhao, and Fangchun Yang. A grouping algorithm of helpers in peer-to-peer video-an-demand systems. In *Proceedings of the 12th international conference on Advanced communication technology*, ICACT'10, pages 497–501, Piscataway, NJ, USA, 2010. IEEE Press.
- [Kondo et al. NETWORKS'14] D. Kondo, Y. Hirota, A. Fujimoto, H. Tode, and K. Murakami. P2p live streaming system for multi-view video with fast switching. In *Telecommunications Network Strategy and Planning Symposium (Networks), 2014 16th International*, pages 1–7, Sept 2014.
- [Liu et al. ICCCN'13] Jiayi Liu and Gwendal Simon. Fast near-optimal algorithm for delivering multiple live video channels in CDNs. In *22nd International Conference on Computer Communications and Networks (ICCCN)*, pages 1–7. IEEE, 2013.
- [Meskill et al. LCN'11] B. Meskill, A. Davy, and B. Jennings. Server selection and admission control for ip-based video on demand using available bandwidth estimation. In *Local Computer Networks (LCN), 2011 IEEE 36th Conference on*, pages 255–258, Oct 2011.
- [Wu et al. INFOCOM'07] Chuan Wu and Baochun Li. Strategies of conflict in coexisting streaming overlays. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 481–489, May 2007.
- [Wu et al. NET'11] Chuan Wu, Baochun Li, and Shuqiao Zhao. On dynamic server provisioning in multichannel P2P live streaming. *IEEE/ACM Trans. Netw.*, 19(5):1317–1330, Oct 2011.
- [Wang et al. INFOCOM'10] Miao Wang, Lisong Xu, and B. Ramamurthy. Linear programming models for multichannel p2p streaming systems. In *INFOCOM, 2010 Proceedings IEEE*, pages 15, March 2010.
- [Wu et al. INFOCOM'09] D. Wu, Y. Liu, and K. W. Ross. Queuing network models for multi-channel live streaming systems. In *INFOCOM*, 2009.
- [Wu et al. INFOCOM'08] Chuan Wu, Baochun Li, and Shuqiao Zhao. Multi-channel live P2P streaming: Refocusing on servers. In *The 27th Conference on Computer Communications (INFOCOM)*. IEEE, 2008.
- [Wang et al. GLOBECOM'10] Zhi Wang, Chuan Wu, Lifeng Sun, and Shiqiang Yang. Strategies of collaboration in multi-channel P2P VoD streaming. In *IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1–5. IEEE, 2010.
- [Wu et al. PDS'08] Chuan Wu, Baochun Li, and Zongpeng Li. Dynamic bandwidth auctions in multioverlay P2P streaming with network coding. *IEEE Transactions on Parallel and Distributed Systems*, 19(6):806–820, June 2008.

Thanks

Q & A