Bi-criteria Approximation for a Multi-Origin Multi-Channel Auto-Scaling Live Streaming Cloud

Chang, Zhangyu

Supervised by Prof. Gary Chan

19 April 2025

Contents

Live Video Broadcasting Trends

Bandwidth dedicated to **video traffic** is expected to jump to 82% of the total traffic by 2022.

Live Internet video will account for 17% of Internet video traffic by 2022. Live video will grow 15-fold from 2017 to 2022.

Surveillance: 2%; Live Video: 17%; Long-form VoD: 62%; Short-Form VoD: 18%

(Cisco's Annual Internet Report, Updated: March 9, 2020)

Sports Games

News Broadcasting

Online Education
Seminar/Lecture

Live Video Based on Auto-scaling Cloud

Live video streaming

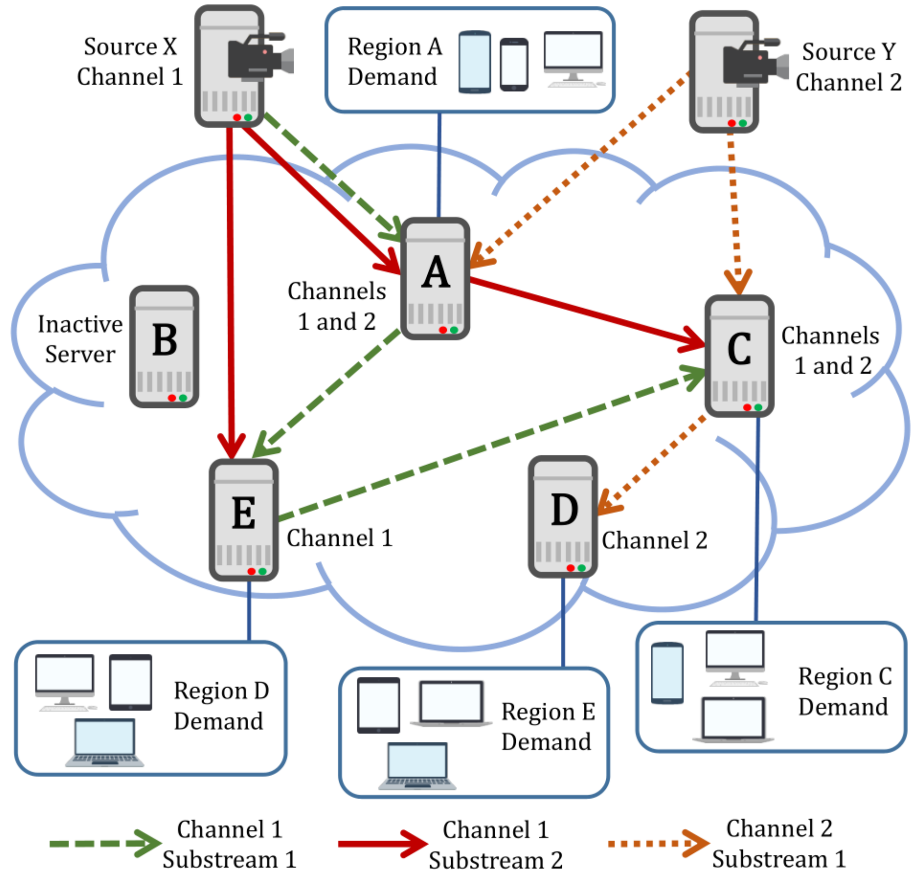
- Sports Games, News Broadcasting, Online Education etc.
- A significant component of Internet traffic
- Traffic varies significantly over a day
- Require considerable and flexible network resource
- Geo-dispersed users with heterogeneous access pattern

Auto-scaling cloud

- Infrastructure as a Service (IaaS): No permanent investment for content providers (CP).
- Globally geo-distributed auto-scaling data centers: Easy to deploy a global service.
- Pay-as-you-go feature: Auto-scaling servers (e.g., virtual servers) and link capacities (dedicated link between servers) on demand with minimal standby cost.
- On-the-fly resource allocation: Avoid cost of over-provisioning due to user dynamics.

Multi-Origin Multi-Channel Live Video Cloud

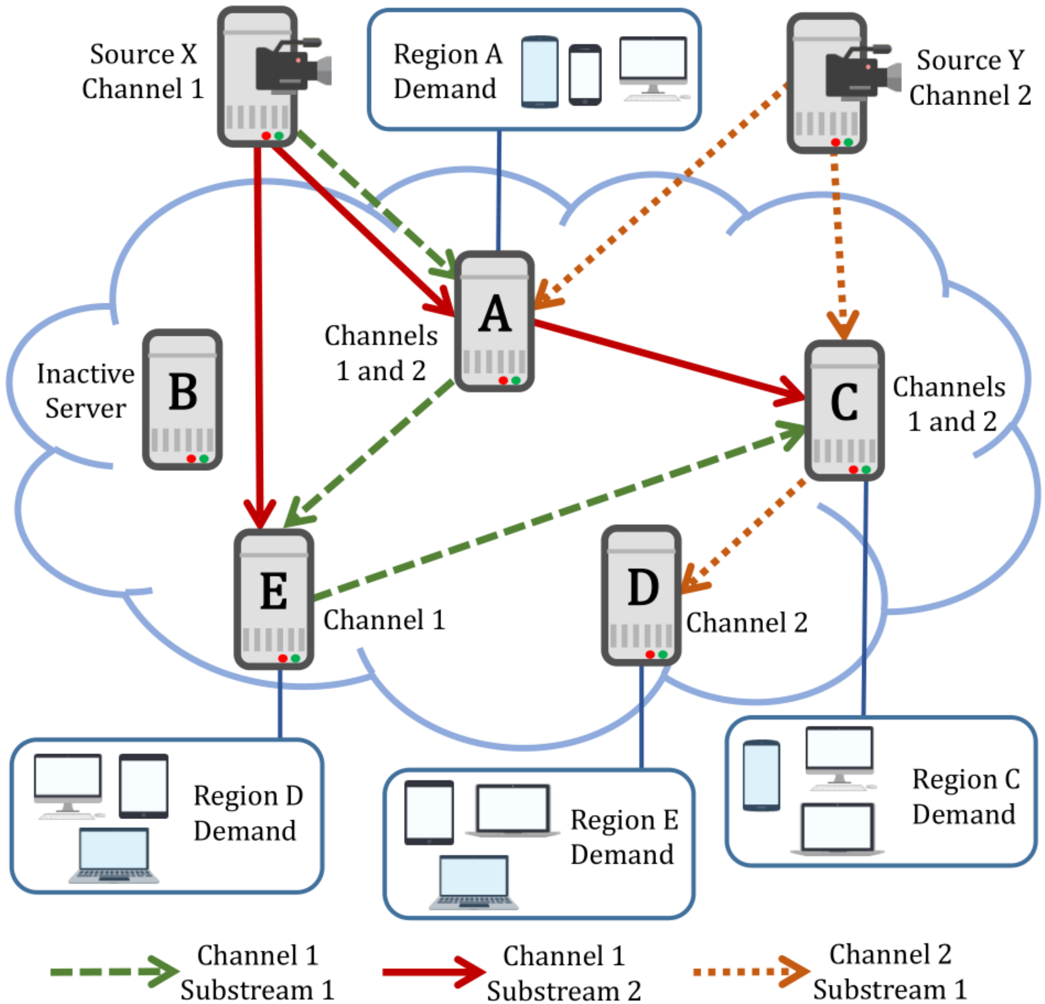
Major Components in a Live Video Cloud	
Origin Server	Origin of live video channels.
End Server	Stream the live content to its associated local users. (e.g., a CDN node, server farm, data center, etc.)
Channel	Multiple streaming rates; a channel may be split into n substreams.
Substream	Same bitrate; each substream is delivered to the servers through a delivery tree.



- **Server** *requests* channels based on local demand and *serves* its local users.
- **Servers** (with user demands) help each other in streaming contents.
- **Substreams** are *pushed* from the origins to the end servers.
- A server needs *all* the **substreams** to recover the original video.

An Example of Streaming Delivery

Origin	X & Y originate Channel 1 & 2 respectively.
Channel	Channel 1 has 2 substreams and Channel 2 has only 1 substream.
End Server	Servers receive streams either from origin or other end servers.
Server A	Receives both Channel 1 and Channel 2 from the origins.
Server C	Receives Channel 2 from source but Channel 1 from A and E.
Server D	Receives Channel 2 from Server C.
Server E	Receives one substream of Channel 1 from source and the other from A.
Server B & Some Links	Not used, therefore no cost is from them.



Bi-criteria Objectives: Deployment Cost and Source-to-end Delay

Minimize Source-to-end Delay

Source-to-end delay, an important Quality-of-Experience (QoE) measure, consists of **Propagation delay** and **Scheduling delay.**

- **Propagation delay**: The time for a signal to travel from one server to the next over the overlay, given by half of the round-trip time (RTT)
- **Scheduling delay**: The worst-case elapsed time from a server having fully received a video segment to the instant that the segment fully departs the server for the next server.

Minimize Deployment Cost

Deployment cost consists of **Server Cost** and **Link Cost.**

- **Server Cost**: Due to the servers allocating its uploading capacity to the other servers.
- **Link Cost**: Depends on the pairwise link capacity allocated between servers

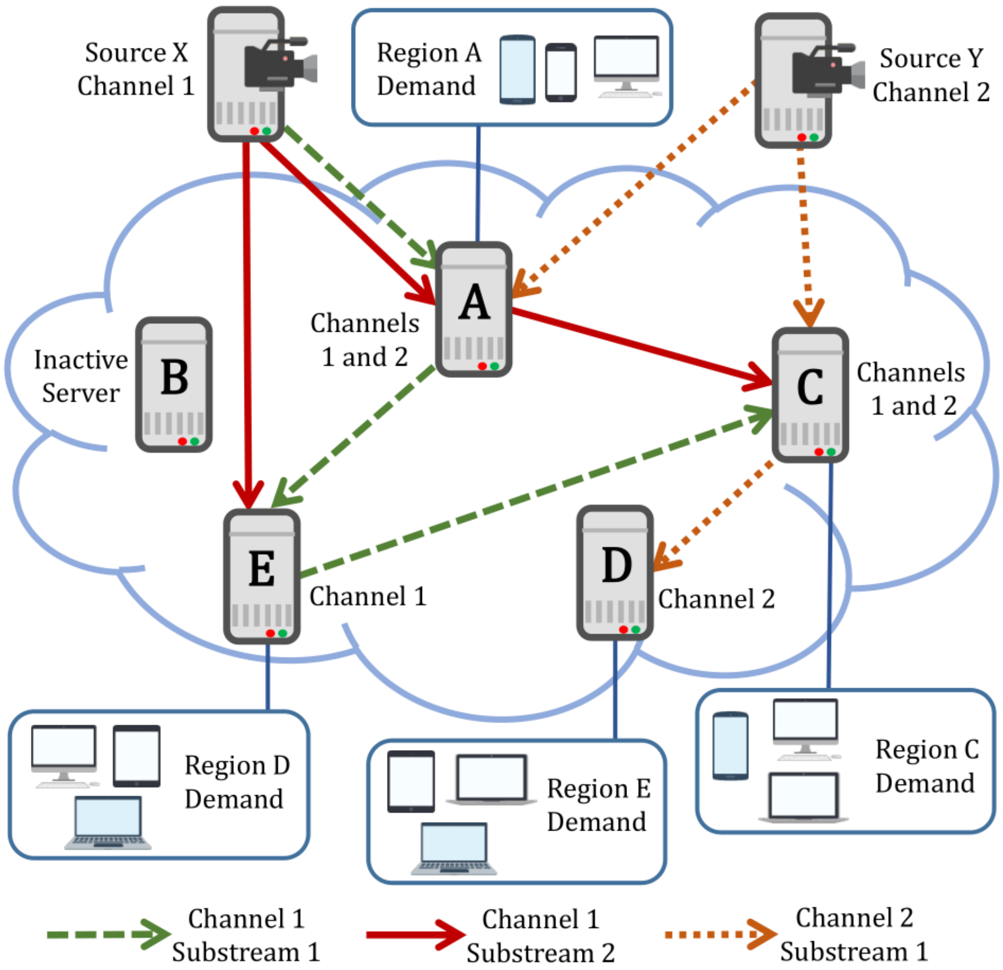
Both server and link capacities are shared with all the channels.

The capacity to serve the local demand is fixed given the local user demand (not an optimization parameter).

Cost to Deploy a Live Video Service on an Auto-scaling Cloud

Cost to deploy the service on an auto-scaling cloud

- **Server Cost**: CP allocates *auto-scaling servers* from cloud service providers at geo-dispersed locations.
- **Link Cost**: CP allocates *link capacities* between the data centers.
- **Pay-as-you-go**: Cost only based on the resource *allocated* for the service
- Usually billed on certain interval (e.g., hourly basis)
- CP can **reallocate** the resource based on *actual* and *predicted* user demand



Optimizing the Bi-criteria Problem

The bi-criteria problem is equivalently to minimizing *deployment cost* subject to a certain maximum *source-to-end delay* constraint of each substream.

The decision variables for this bi-criteria optimization are:

- **Overlay construction**: The construction of the delivery tree of each substream, namely, *to what servers* and *what substreams* an auto-scaling server should forward.
- **Resource allocation**: The allocation (purchase) of server and link capacities.

Overlay construction and resource allocation are inter-dependent decisions , have the comparable time scale for readjustment, and hence they need to be jointly optimized.

Contributions

Bi-criteria problem formulation and complexity analysis

- We give comprehensive and realistic model:
  1. Multi-source multi-channel overlay live streaming
  2. Captures various cost and streaming delay components
  3. Minimizes the total deployment cost while meeting QoE constraints
- We prove the NP-hardness of this problem

COCOS: A novel bi-criteria approximation algorithm

- **C**ost-**O**ptimized Multi-**S**ource Multi-**C**hannel **O**verlay **S**treaming
- Linear-programming based solution
- Proven approximation ratio by manipulating the LP solution
- Strictly meets the QoE constraints

Extensive trace-driven experimental results

- Trace-driven experimental studies using real-world data (from a leading Chinese video service provider)
- Substantially outperform the state-of-the-art schemes
- Cutting the cost significantly (in general by more than 50%)

Related Work

Single-source single-channel

- Cannot be extended to multi-source multi-channel live streaming due to bandwidth sharing [Tran et al. ICISA' 17, Ha et al. MTA'16, Wichtlhuber et al. IFIP'14, Roverso et al. MMSys'15]

Multi-source multi-channel

- VoD based approaches cannot be extended to live streaming due to delay consideration [Ma et al. NOSSDAV '16, Hu et al. IEEE JSAC'17, Chen et al. IEEE JSAC'17, Liu et al. IEEE TMM'16]
- [Kondo et al. NETWORKS'14, Liu et al. ICCCN'13, Meskill et al. LCN'11, Liu et al. GLOBECOM'15, H. K. Yarnagula et al. IEEE TMM '19, I. Irondi et al. IEEE TMM '19]

Live streaming works mainly focus on different objectives

- Effective delivery of live contents from a CDN server to its local audience [R.-X. Zhang et al. MM'20, R.-X. Zhang et al. NOSSDAV'19, F. Haouari et al. ICC'19]
- Maximizing bandwidth or minimizing source load [Detti et al. Computer Networks'15, Wang et al. INFOCOM'10, Wu et al. INFOCOM'09, Chen et al. Peer-to-Peer Netw. Appl.'17, Wang et al. GLOBECOM'10, Kuo et al. Peer-to-Peer Netw. Appl'15, S. Budhkar et al. Peer-to-Peer Netw. Appl'19]

None above has proven approximation ratio

Contents

Major Symbol Used in Formulation

S	The set of sources	$\boldsymbol{b_{ij}}$	Link capacity of edge $\langle i, j\rangle$ (bits/s) **(RA)**
R	The set of auto-scaling servers	$T(m)$	The delivery tree of channel m
V	The set of all sources and servers ($V = S \cup R$)	$\boldsymbol{x_{ij}\ (m)}$	Binary variable indicating whether link$\langle i, j\rangle$ is in $T(m)$ **(OC)**
$\langle i, j\rangle$	The directed edge from node i to j	L_{ij}	Server-to-Server (S2S) delay of link $\langle i, j\rangle$
E	The set of all edges	$D_i(m)$	Origin-to-End (O2E) delay of channel m at server i (in second)
M	The set of all channels	$\mathbb{D}_i(m)$	O2E delay upper bound of channel m at server i (in second)
$R(m)$	The set of servers that demand channel m	Θ_i	Server cost of server i (per second)
M_i	The set of channels that server i demands	θ_i	Unit price of uploading streaming at server i (per bit)
$\tau(m)$	Streaming rate of channel m (bits/s)	Φ_{ij}	Link cost due to traffic through link $\langle i, j\rangle$ (per second)
$s(m)$	The live source of channel m	ϕ_{ij}	Unit price of data transmission through link $\langle i, j\rangle$ (per bit)
$\boldsymbol{u_i}$	Uploading capacity of node i (bits/s) **(RA)**	C	Total deployment cost (per second)

Comprehensive Model as an NP-Hard Problem: Minimum Cost Streaming with Delay Constraints

Minimize the total deployment cost :

$$C = \sum_{\langle i,j \rangle \in E} \Phi_{ij} + \sum_{i \in V} \Theta_i$$

Cost function
• Server Cost $\Theta_i = \theta_i u_i, \forall i \in V.$
• Network Cost $\Phi_{ij} = \phi_{ij} b_{ij}, \langle i,j \rangle \in E.$

NP-complete Restricted Shortest Path Problem (RSP) is reducible to our problem. It is **NP-Hard**!

Delay Constraints
• Source-to-end Delay of Node j: $D_j(m) = D_i(m) + L_{ij}$
• Source-to-end Delay Constraint: $D_i(m) \leq \mathbb{D}_i(m), \forall i \in R(m), m \in M.$

Fulfill Demand of Channels

$$x_{ij}(m) = \begin{cases} 1, & \text{if } \langle i,j \rangle \in T(m); \\ 0, & \text{otherwise.} \end{cases}$$

$$\sum_{\langle i,j \rangle \in E} x_{ij}(m) \geq 1,$$

$$\forall j \in R(m), m \in M.$$

NP-Hardness

- **NP-hard** *Restricted Shortest Path Problem* **(RSP)** is reducible to our **MCSDC** problem.
- **RSP problem**: In a graph $G(V,E)$, each link $\langle i,j\rangle \in E$ has an associated positive cost Φ_{ij} and a positive delay L_{ij}.
  - Find a spanning tree;
  - Minimum total deployment cost;
  - Delays satisfy the given limits.
- RSP is a special case of **MCSDC** with one source, one channel, and no server cost.

Contents

Major Symbol Used in COCOS Algorithm

$\alpha + \delta$	Approximation ratio of the deployment cost	$f_{ij}^{l}(m)$	Flow fraction of channel m to server l through link $\langle i, j \rangle$
β	Approximation ratio of the O2E delay	$u_i\ (m)$	Uploading rate of channel m at server i in LP (bit/s)
k	Number of substreams for a channel	$b_{ij}\ (m)$	Transmission rate of channel m through link $\langle i, j \rangle$ in LP (bit/s)
C_{SO}	Deployment cost of LP super optimum solution (per second)	$C(m)$	Deployment cost due to channel m in LP (per second)
C_{EO}	Deployment cost of exact optimum solution (per second)	$\Psi(m)$	The set of substreams of channel m
C_{ID}	Deployment cost of COCOS given $k \to \infty$ (per second)	$\Gamma(\psi)$	The delivery tree of substream ψ
C_{CC}	Deployment cost of COCOS (per second)	$n_{ij}\ (m)$	Number of substreams allowed on link $\langle i, j \rangle$ for channel m
$z_{ij}\ (m)$	Fractional stream of channel m through link $\langle i, j \rangle$		

Overlay Construction and Resource Allocation

Relaxed to a LP problem

We transform the original problem through the following relaxation:

- LP solution can have arbitrary number of fractional substream paths ($0 \leq x_{ij}(\psi) \leq 1$)
- Constraints on source-to-end delay: Use average substream delay

Topology construction

- Assign a link with edges proportional to its traffic (αk edges for full stream)
- Construct trees then pick up the tree that has minimum cost and satisfies the delay constraints
- **Cost approximation ratio**: $\alpha + \delta$
- δ goes to 0 as k goes to infinity
- **Delay approximation ratio**: β
- $1/\alpha + 1/\beta \leq 1$
- Complexity $O(|V|^9|M|k)$

Relax the Formulation to a Linear Program

We transform the original problem through the following relaxation:

- Reformulate the server fulfillment constraints by using the property of flow conservation:

$$\sum_{\langle i,j\rangle\in E} f_{ij}^{l}(m) + \sum_{\langle j,k\rangle\in E} f_{jk}^{l}(m) = \begin{cases} 1, & \text{if } j = s(m); \\ -1, & \text{if } j = l; \\ 0, & \text{otherwise.} \end{cases}$$

- Construct the traffic of the channel on the edge:

$$0 \le f_{ij}^{l}(m) \le z_{ij}(m) \le 1, \forall l \in R(m), \langle i,j\rangle \in E.$$

- Use average delay to fulfill the delay upper bound:

$$\sum_{\langle i,j\rangle\in E} f_{ij}^{l}(m) L_{ij} \le \frac{1}{\beta}\mathbb{D}_l(m), \forall l \in R(m), m \in M.$$

Approximation Ratio and Algorithmic Complexity

Markov's Inequality:

- If *X* is a nonnegative random variable and *a* > 0, then the probability that *X* is at least *a* is at most the expectation of *X* divided by *a*

$$P(X \geq a) \leq \frac{E(X)}{a}$$

- E.g., if the average score of a group of students is 10, the fraction of student who has the score 100 is no more than 1/10.

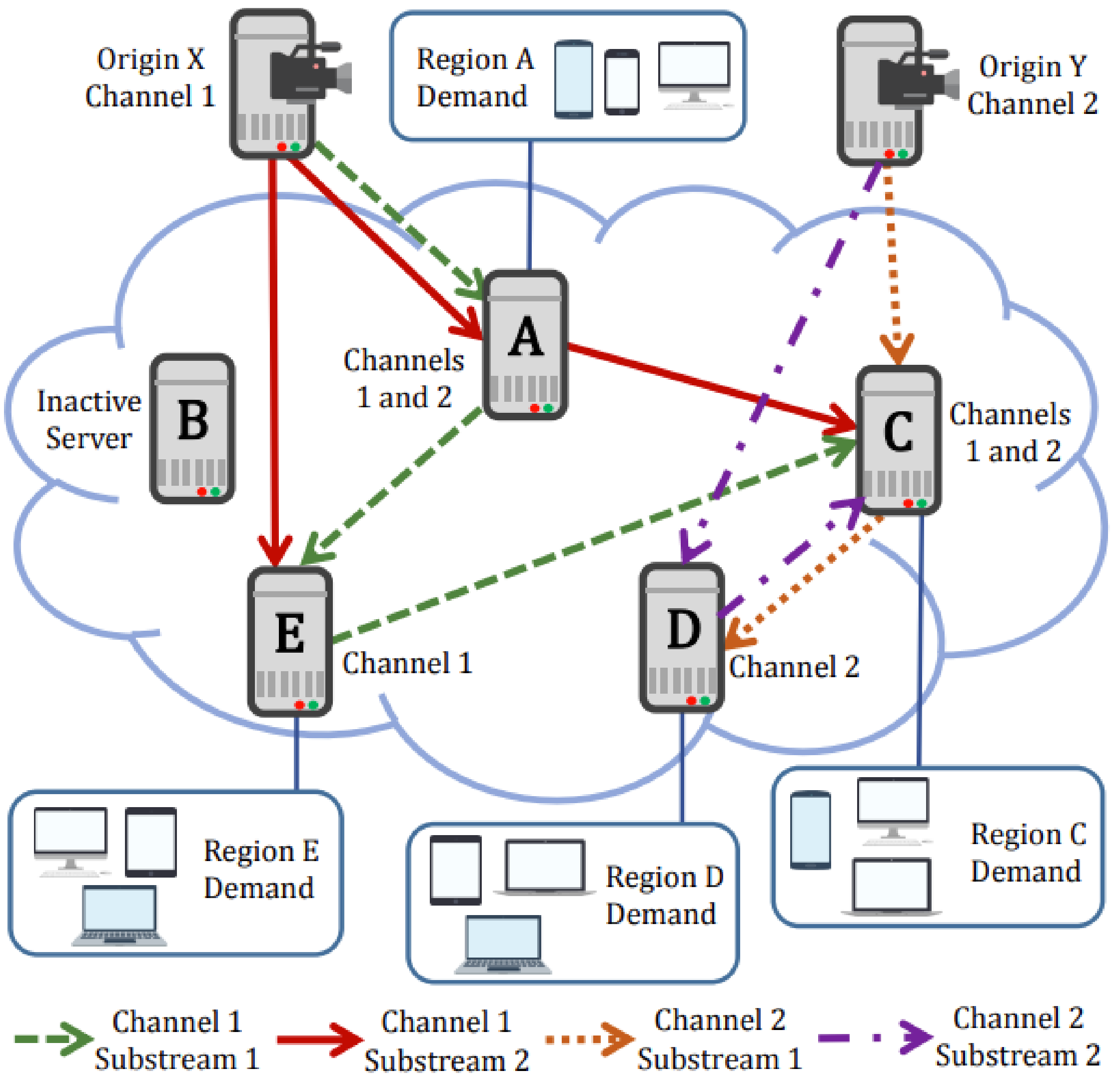
Therefore, in the LP solution, there must exist trees such that

- The cost is at most α times the optimal solution;
- The delay is at most β times the optimal solution.

As the major component of time complexity is solving the linear program, the algorithmic complexity is

$$O(|V|^9|M| + k|V|^2|M|)$$

Example of substream solution for $k = 2$



Contents

Experimental Setup

Performance Metrics

Deployment cost

- Server cost
- Link cost

Delay

Comparison Schemes

Nearest Peer [27]

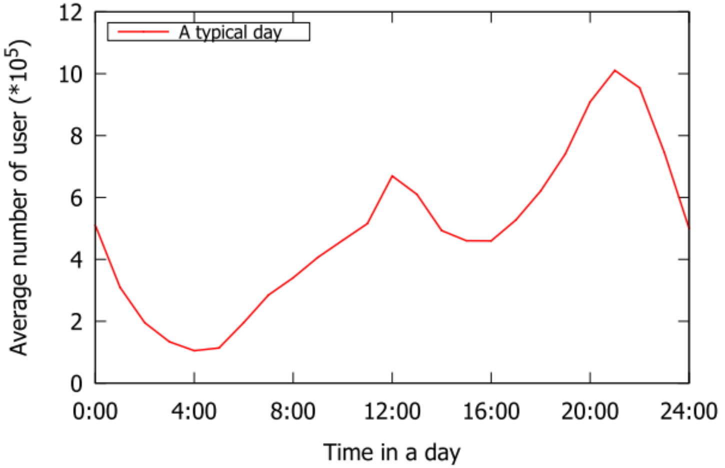
- Consider local popularity
- No cooperative replication

Prim [28]

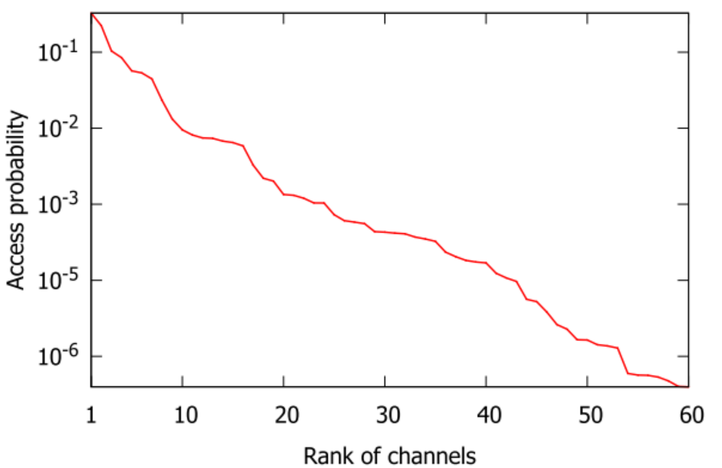
- Minimum cost tree
- Modified to meet delay constraints

Super-optimal

- Relaxed LP solution



Average user number over a typical day.



Access probability of the channels.

Good Tradeoff for humble k and β

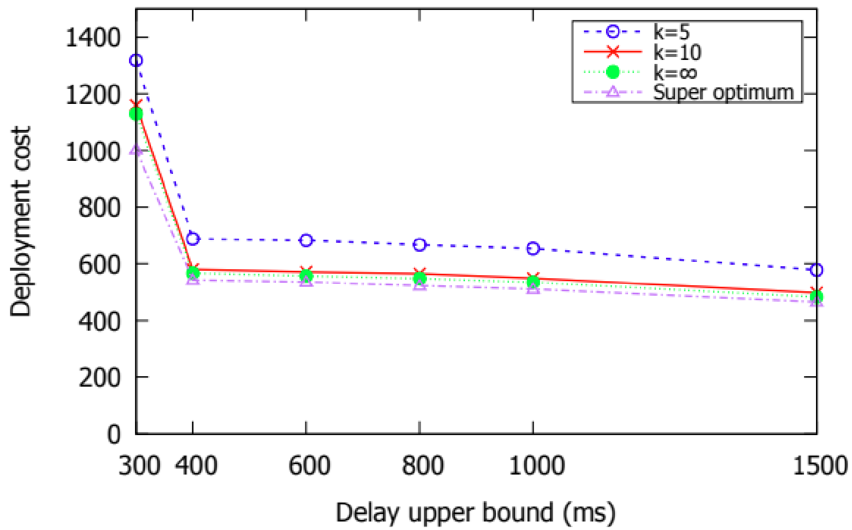


Figure 5.6. Deployment cost versus delay upper bound given different number of substreams.

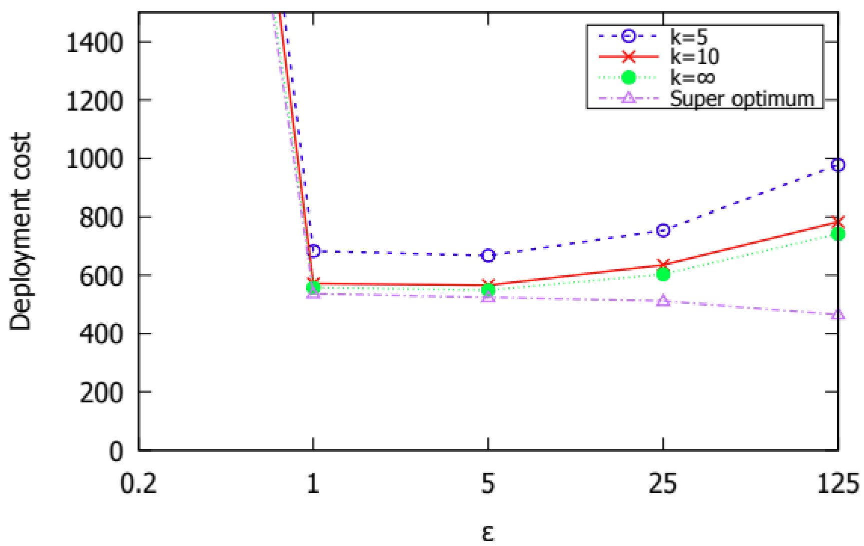
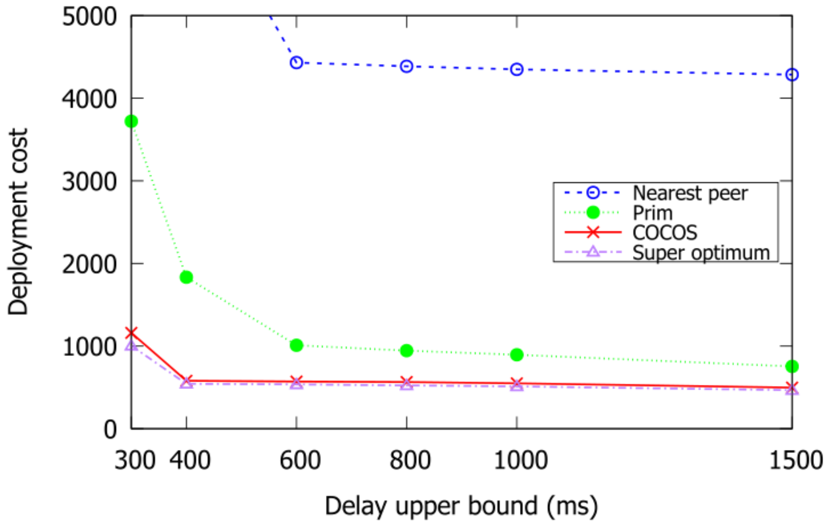


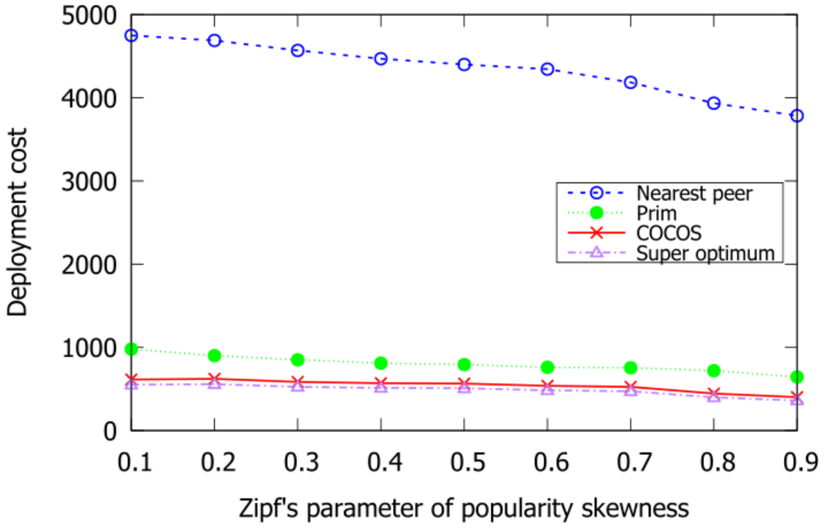
Figure 5.5. Deployment cost versus approximation ratio tradeoff parameter.

$$\alpha = 1 + \varepsilon$$

$$\beta = 1 + 1/\varepsilon$$

Near-Optimal Performance





Real-world data trace:

- From a leading video service website in China (Tencent) over 2 weeks

Near-optimal performance:

- Outperform state-of-the-art schemes

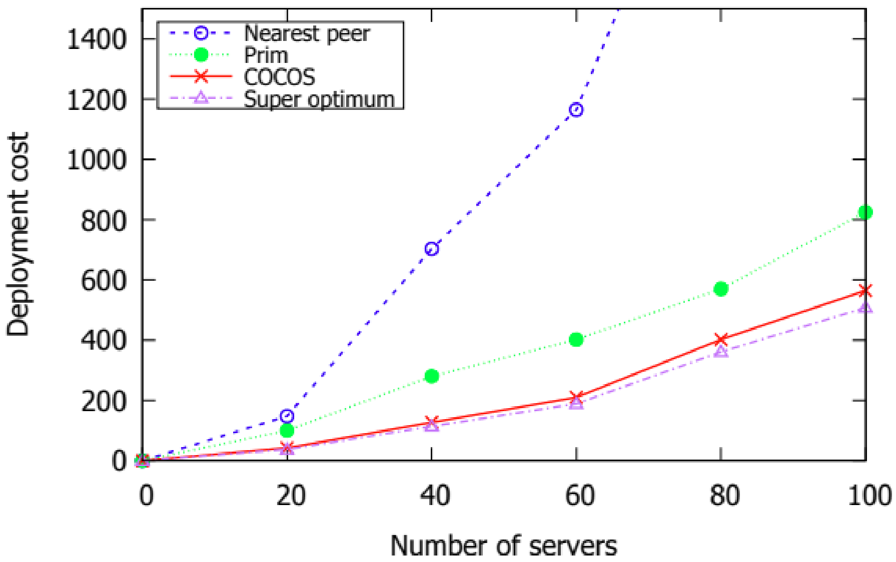
Synthetic data:

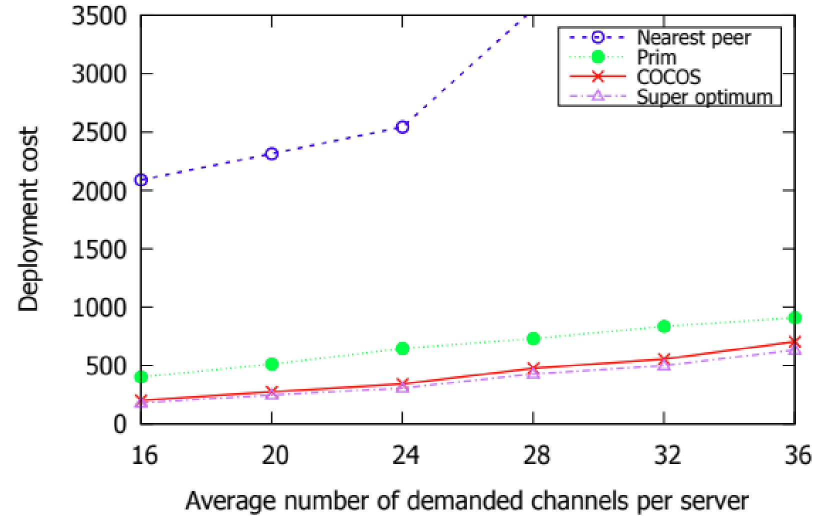
- Verify the performance given difference video popularity

Stable performance:

- Given different video popularity

Scalable Scheme





Better performance for a larger system

Contents

Conclusion

Bi-criteria problem formulation and complexity analysis	• Comprehensive model: 1. Various cost components 2. Streaming constraints for QoE • NP-hardness proof
COCOS: A novel bi-criteria approximation algorithm	• Meet the QoE constraints tightly • LP-based solution • Polynomial time algorithmic complexity • Proven approximation ratio
Extensive trace-driven experimental results	• Outperform the state-of-the-art schemes • Reduce the optimality gap by multiple times

Thank You!

Any Questions?