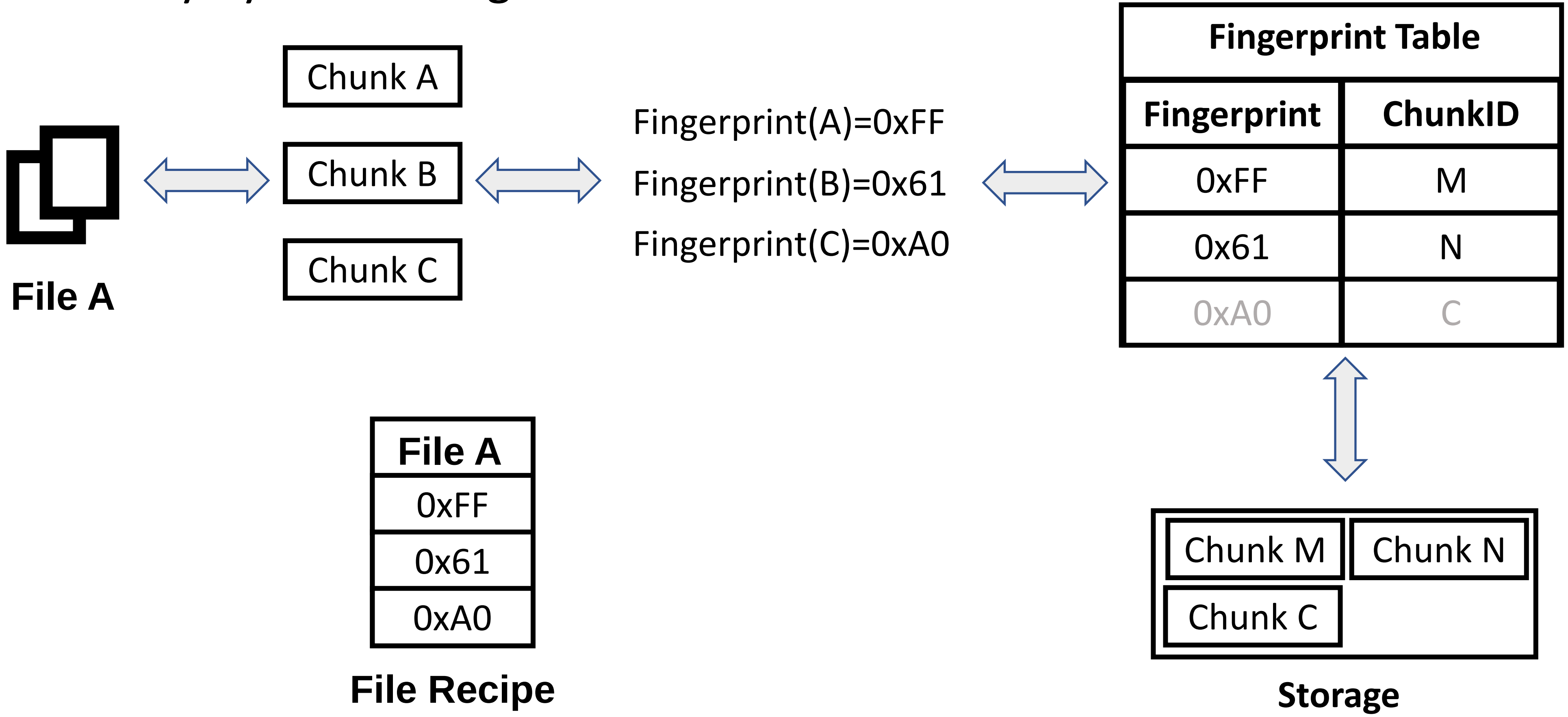
HotDedup: Managing Hot Data Storage at Network Edge through Optimal Distributed Deduplication

INFOCOM '20

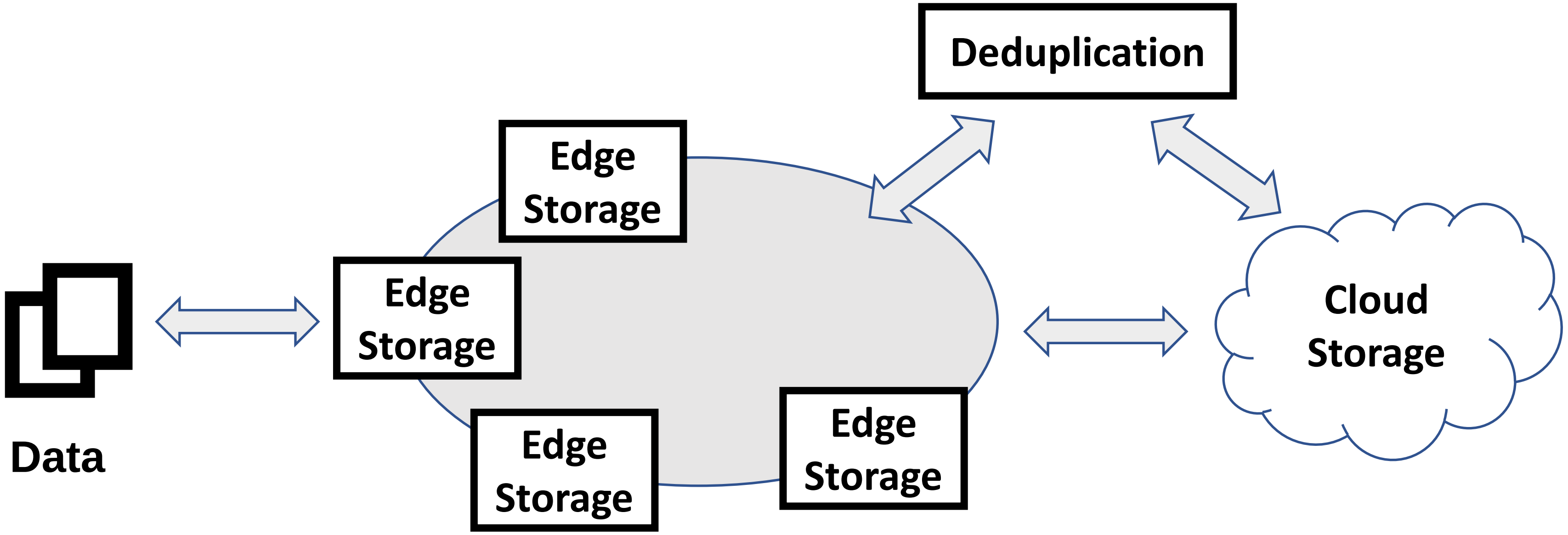
Background

- The goal of **deduplication** is to reduce storage requirements and improve efficiency by eliminating **redundant** information.



Background

- Placing **hot data** at **network edge** helps increasing **performance of edge applications** by making hot data locally accessible
- Partitioning **similar files** with high redundancy into the **same storage tier** helps improving **space efficiency**



Edge and Cloud Storage

Motivation

The capacity of the edge storage is 4 chunks

F_1	A	B	C		$\lambda_1 = 2$
F_2	A	B	C	F	$\lambda_2 = 1$
F_3	A	F			$\lambda_3 = 1$
F_4	B	C	G		$\lambda_4 = 4$
F_5	H	G	J	K	$\lambda_5 = 5$

	Edge Files	Edge Service Rate
• **most-popular-first:**	F_5	λ_5=5
• **space-only:**	F_1 ,F_2,F_3	λ_1+ λ_2+ λ_3=4
• **joint optimal:**	F_1,F_4	λ_1+ λ_4=6

Motivation

- Can we consider both **data popularity** (for optimal data access performance) and **data similarity** (for optimal storage space efficiency) to solve the problem?

Partitioning Problem:

$$\begin{aligned} \max \quad & \sum_{i \in \mathcal{P}} \lambda_i \\ \text{s.t.} \quad & D(\mathcal{P}) \leq B = \sum_k C_k, \\ \text{var.} \quad & \mathcal{P} \subseteq \mathcal{F}. \end{aligned}$$

λ : Edge Service Rate

$\mathcal{P}$: for edge storage

B : total edge storage capacity

$\mathcal{F}$: a set of m files

C_k : storage capacity of edge node k

- In short, under the condition of meeting storage capacity, find the partition result that maximizes edge service rate

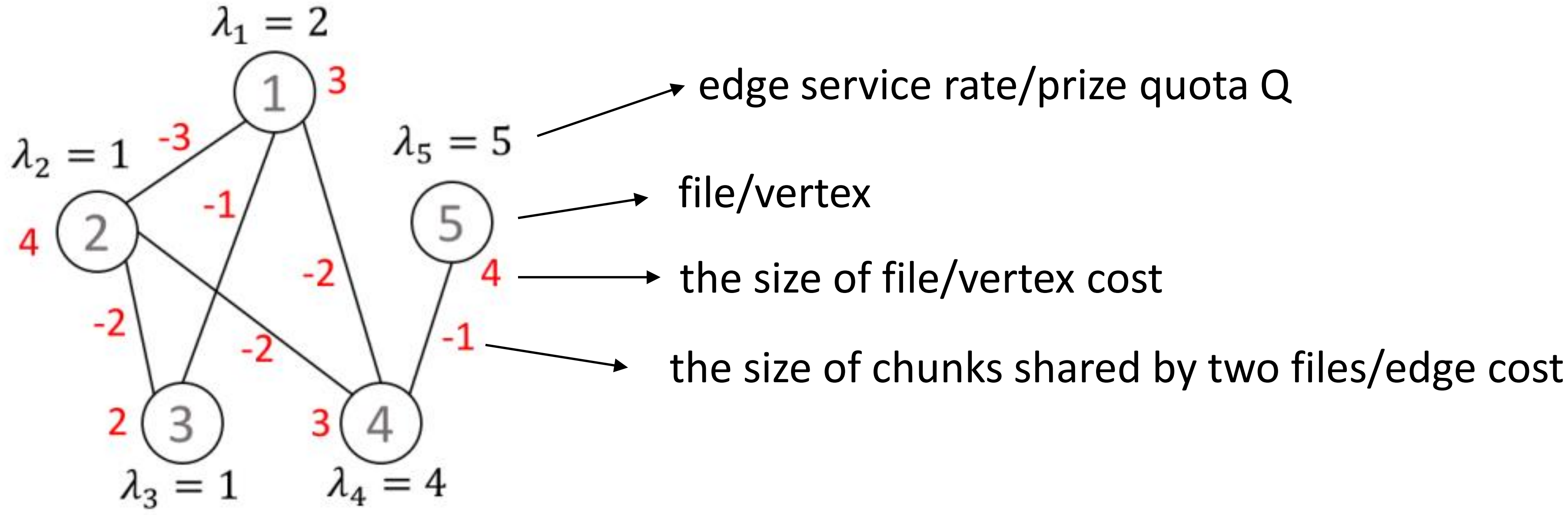
Main Idea

- **Delta Steiner Tree Partitioning (DSTP)**
  1. δ-similarity graph G = {V, E}
  2. k-MST algorithm

- **δ-similarity graph G** is used to express partitioning problems
- The **k-MST algorithm** is used to calculate and obtain the results

δ-similarity graph - express partitioning problems

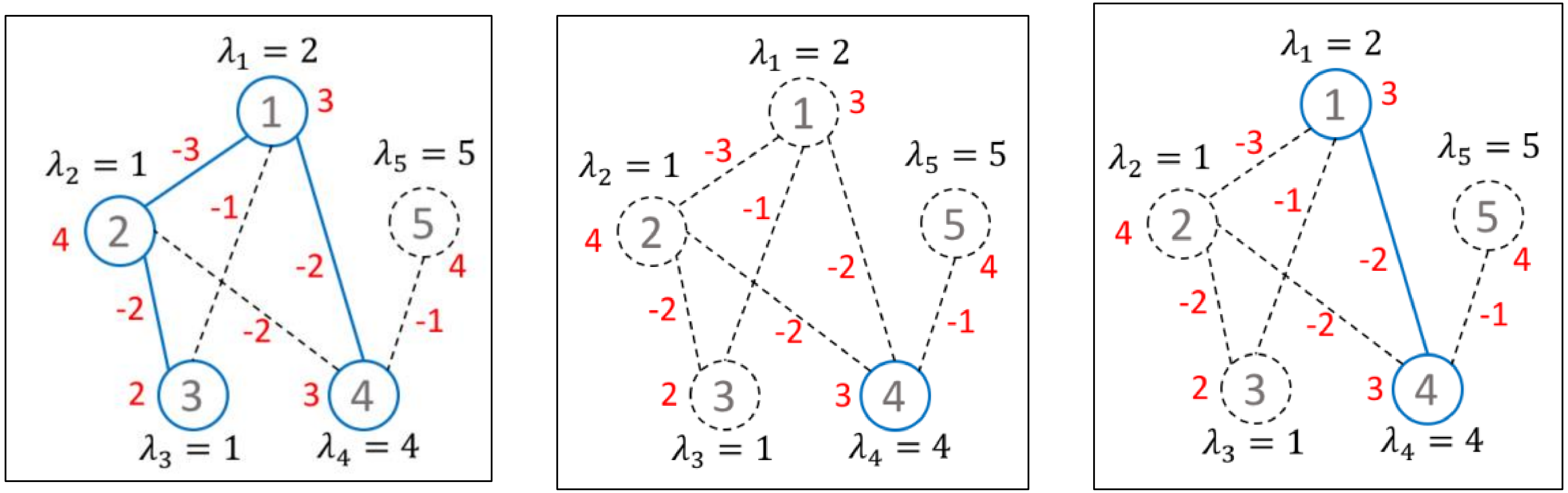
- How to express the partitioning problem?



- Find subgraphs in G that meet storage space constraint *B* - **quota problem**

δ-similarity graph - express partitioning problems

An example : edge storage capacity B = 4.



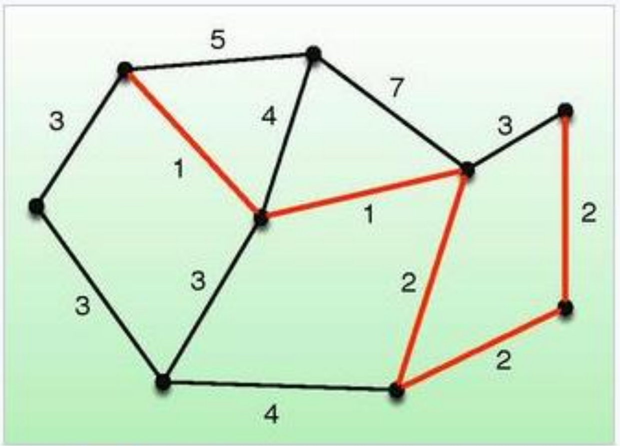
initial prize quota Q = 8

Q = (8 + 0)/2 = 4

Q = (4 + 8)/2 = 6

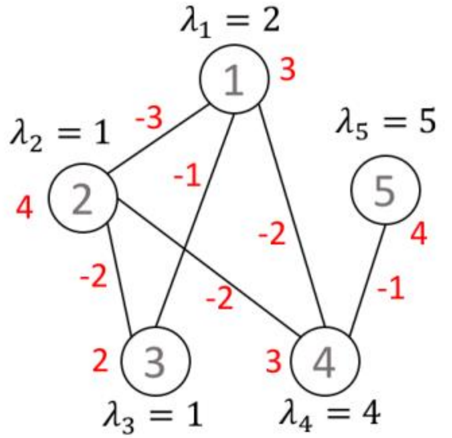
k-MST algorithm - calculate the δ-similarity graph G

- The k-minimum spanning tree problem asks for a tree of **minimum cost** that has exactly **k vertices** and forms a **subgraph** of a larger graph
- How to **calculate** the δ-similarity graph G?



The 6-MST of G

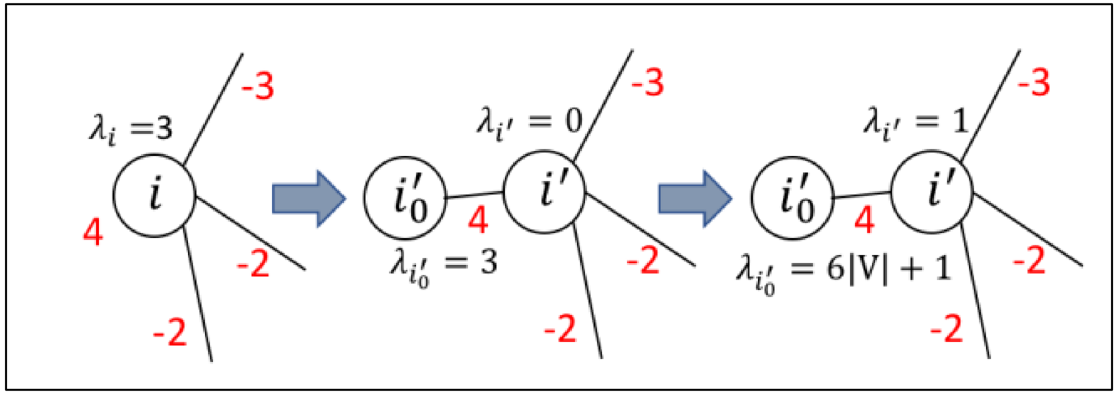
non-negative edge costs



vertex costs
negative edge costs
prize quota Q

k-MST algorithm - calculate the δ-similarity graph G

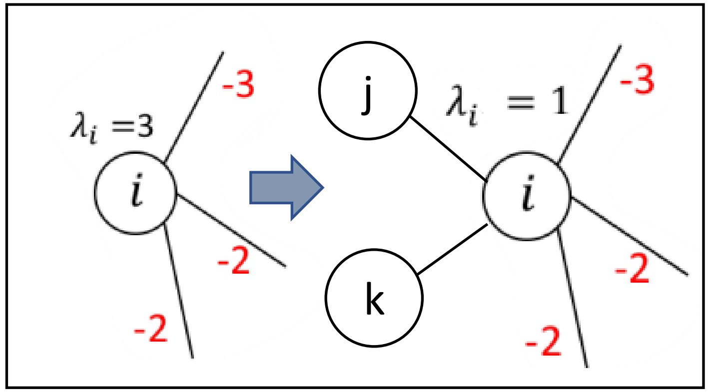
- **Key idea** is to transform graph G into G`



remove vertex costs

deal with zero-prize vertices

$$\hat{\lambda}_s = 2|V'|\lambda_s + 1 \qquad \hat{Q} = 2Q|V'|$$



convert Q into vertices

Delta Steiner Tree Partitioning

- **Correctness**

1. Any tree that is optimal for the G` is optimal for the G before vertex prize scaling.

2. Algorithm DSTP finds a feasible solution for the partitioning problem.

$$
\begin{aligned}
D(\mathcal{T}) &= |\cup_{i\in T} F_i| \\
&\leq \sum_{i\in T} |F_i| - \sum_{(i,j)\in T} |F_i \cap F_j| \\
&= \sum_{e\in T} v_e(G) + \sum_{i\in T} \mu_i(G) \leq B.
\end{aligned}
$$

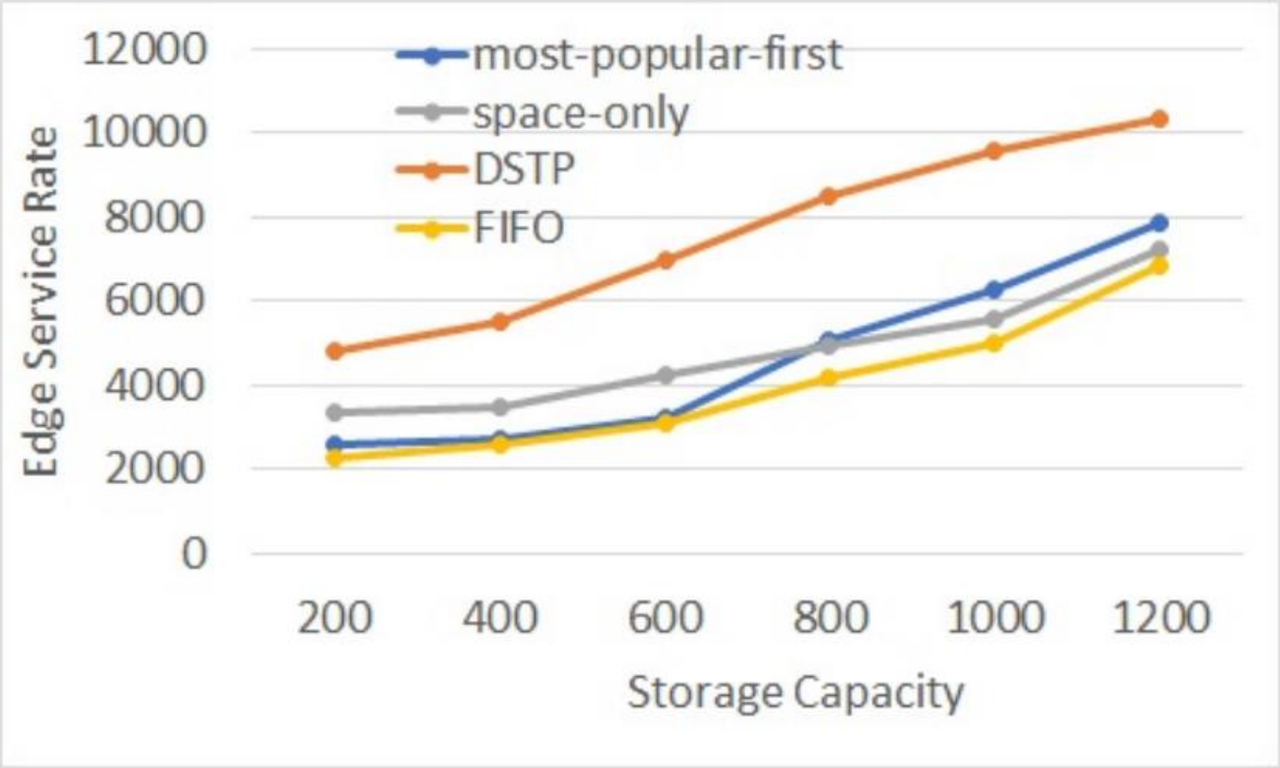
Brief Summary

- Transformed the **partitioning problem** into a **quota problem** and designed **DSTP**

- **δ-similarity graph G** may not be precise, but still provide a simple yet effective model for optimizing storage systems with deduplication

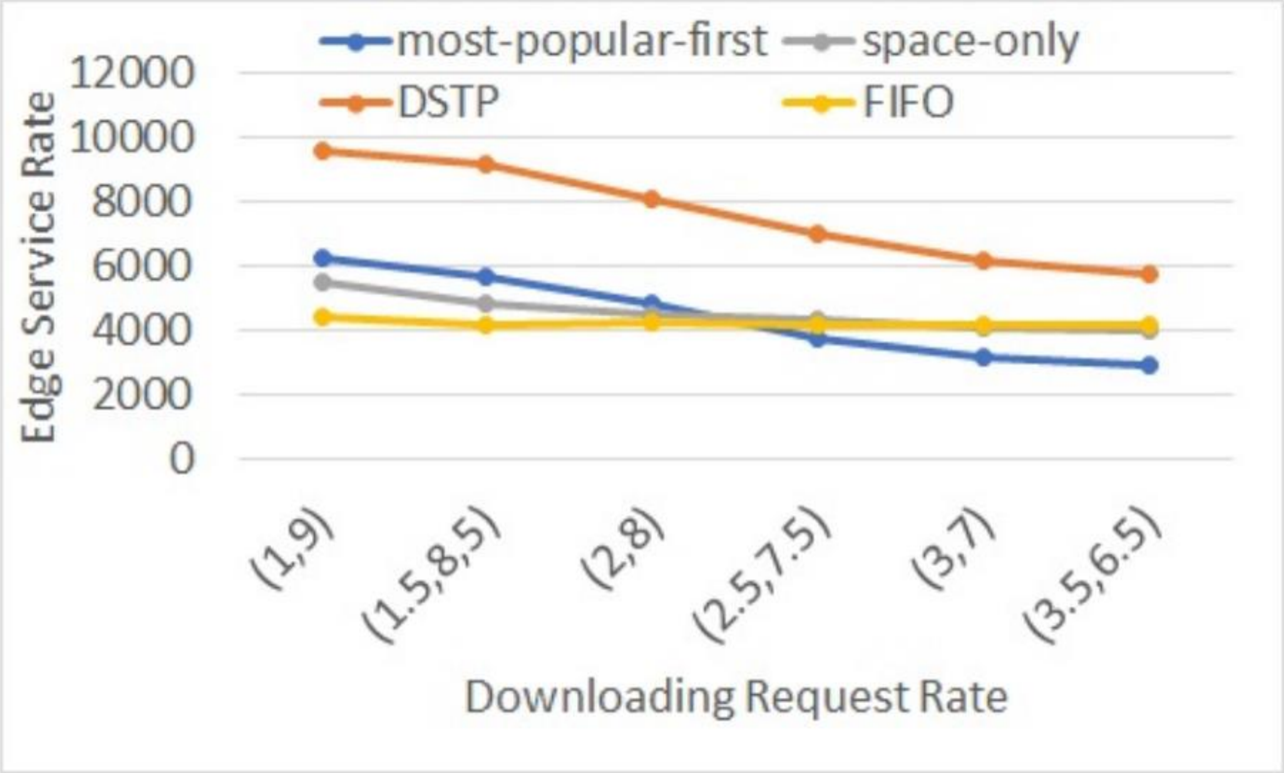
- To use the **k-MST algorithm**, convert G equivalently into G`

Evaluation



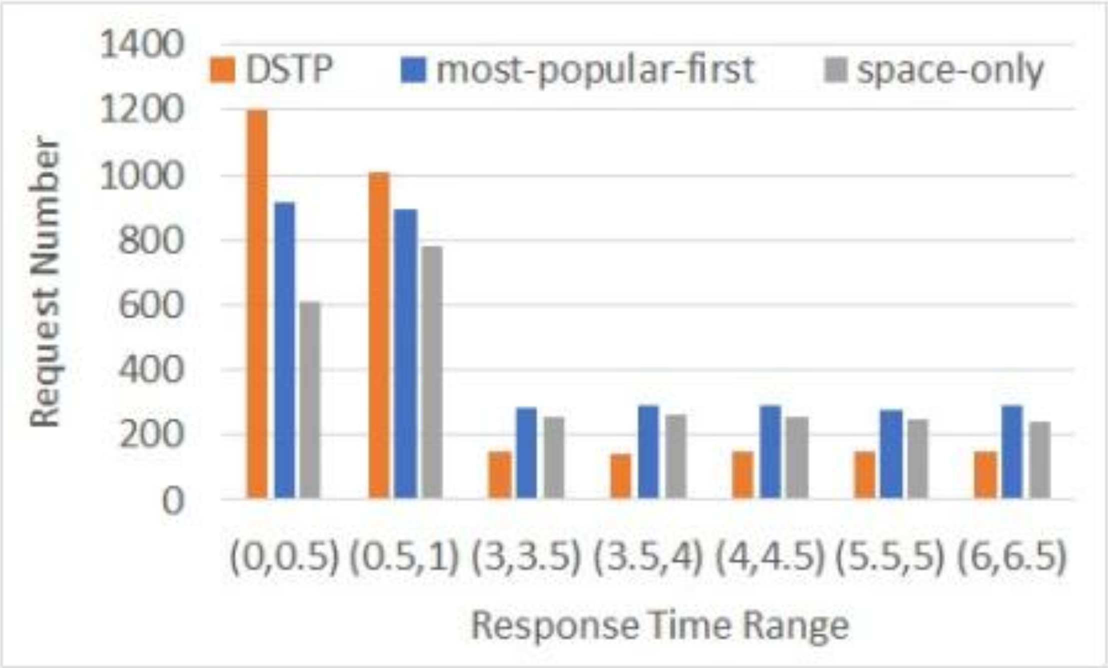
- DSTP is able to outperform the baselines by 43.4-118.5% in terms of edge service rate

Evaluation



- It almost doubles the edge service rate under uniform request distribution in (1, 9), which corresponds to large variation in file popularity and allows DSTP to better optimize over select hot files with high access rate

Evaluation



- achieve the minimum response time, as it is able to maximize the number of requests served by network edge and also achieve high space efficiency