A Light-weight Content Distribution Scheme for Cooperative Caching in Telco-CDNs

Takuma Nakajima, Masato Yoshimi,
Celimuge Wu, Tsutomu Yoshinaga

The University of Electro-Communications

Summary

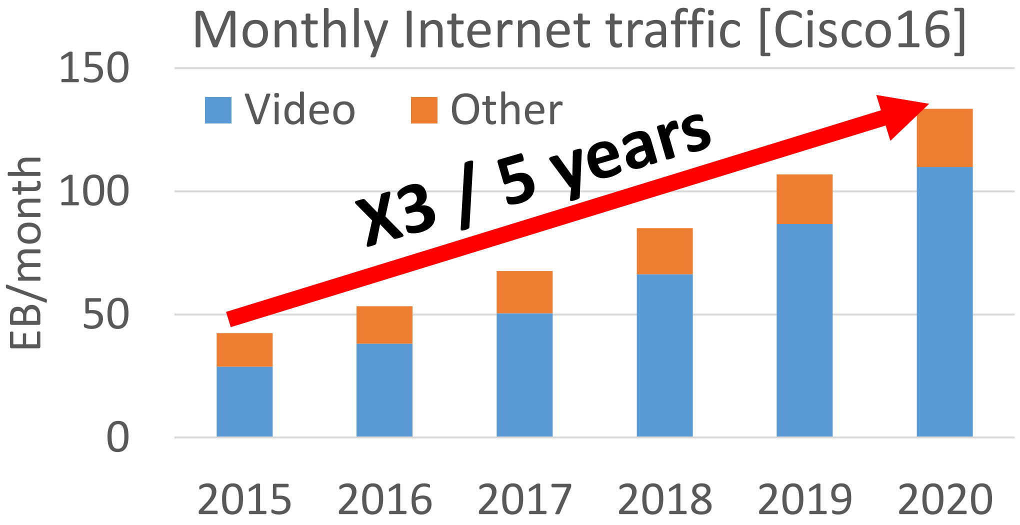
- **Proposal**: A light-weight scheme to utilize cache servers to reduce growing internet traffic
- **Major Contributions**:
  - **Enhanced traffic reduction by distributing contents** with a simple grouping scheme of cache servers and contents
  - **Following a rapid change in access patterns** by utilizing a hybrid caching scheme of LFU and LRU algorithms
- **Evaluation**:
  - Case study using a backbone network in Japan and YouTube access patterns
  - Comparison of traffic reduction and computational overhead with a sub-optimal result calculated by Genetic Algorithm

Outline

- **Introduction**
  - Rapid growth of video traffic
  - Efficient utilization of cache servers
- **Proposal**
  - Adjusting cache distribution by a simple scheme
  - Following rapid change in access pattern by LFU/LRU hybrid caching
- **Evaluation**
  - Traffic reduction and computational overhead compared with a sub-optimal result
- **Conclusion and Future work**

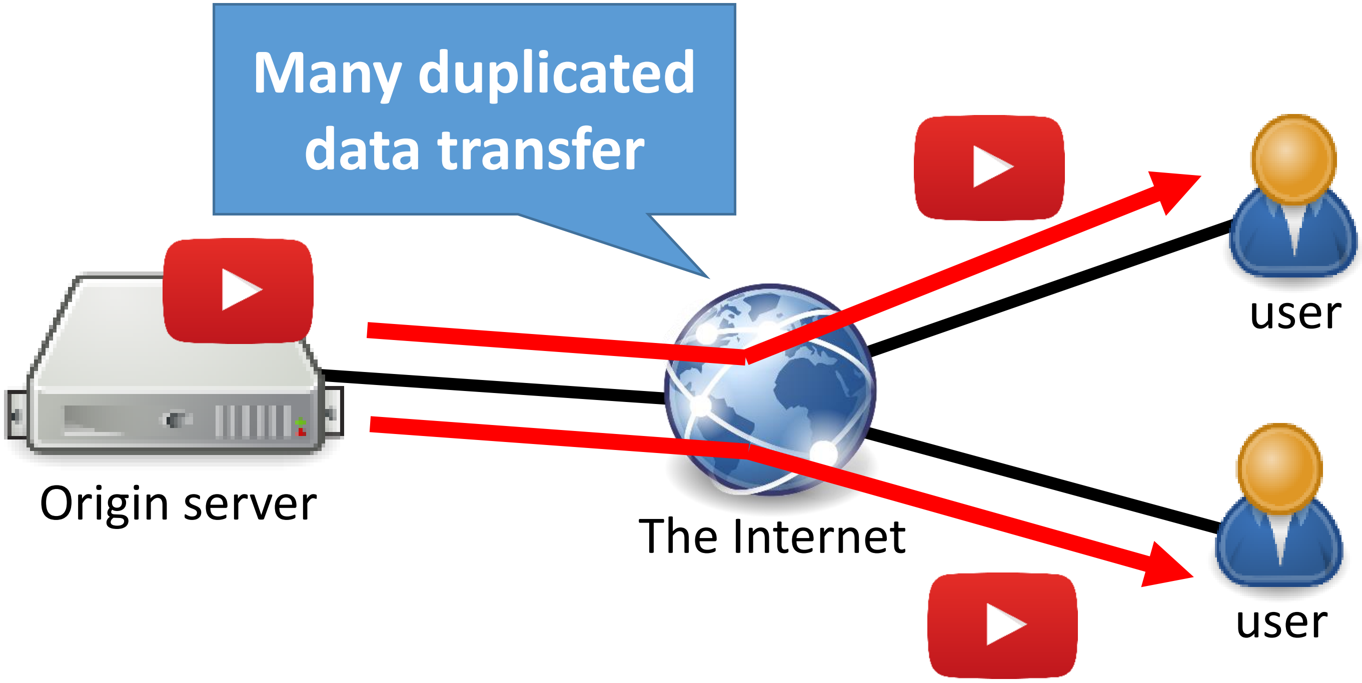
Rapid growth of video traffic

- **Video-on-Demand** (VoD) services will contribute more than **80% of internet traffic** in 2020 [Cisco16]
- Such enormous traffic will cause many congested links and **degrade network performances**
- **Efficient utilization of cache servers** is a key to reduce the internet traffic


Monthly Internet traffic [Cisco16]

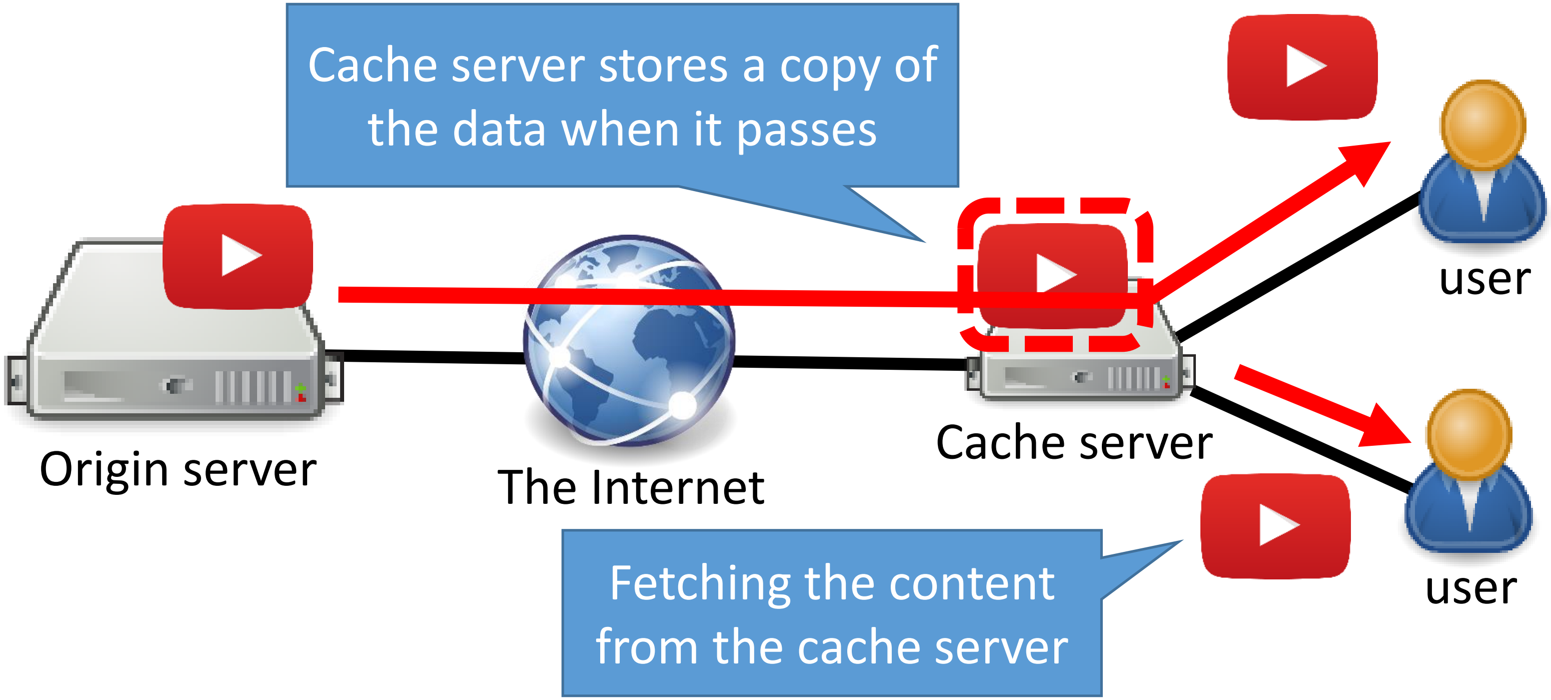
How internet traffic increases?

- The network **transfers the same data** to different users many times increasing the internet traffic



Traffic reduction by cache servers

- The cache server **stores a copy of contents** when they pass the server
- The cache server **responds the copy to users** to reduce the traffic from the cache server to the origin

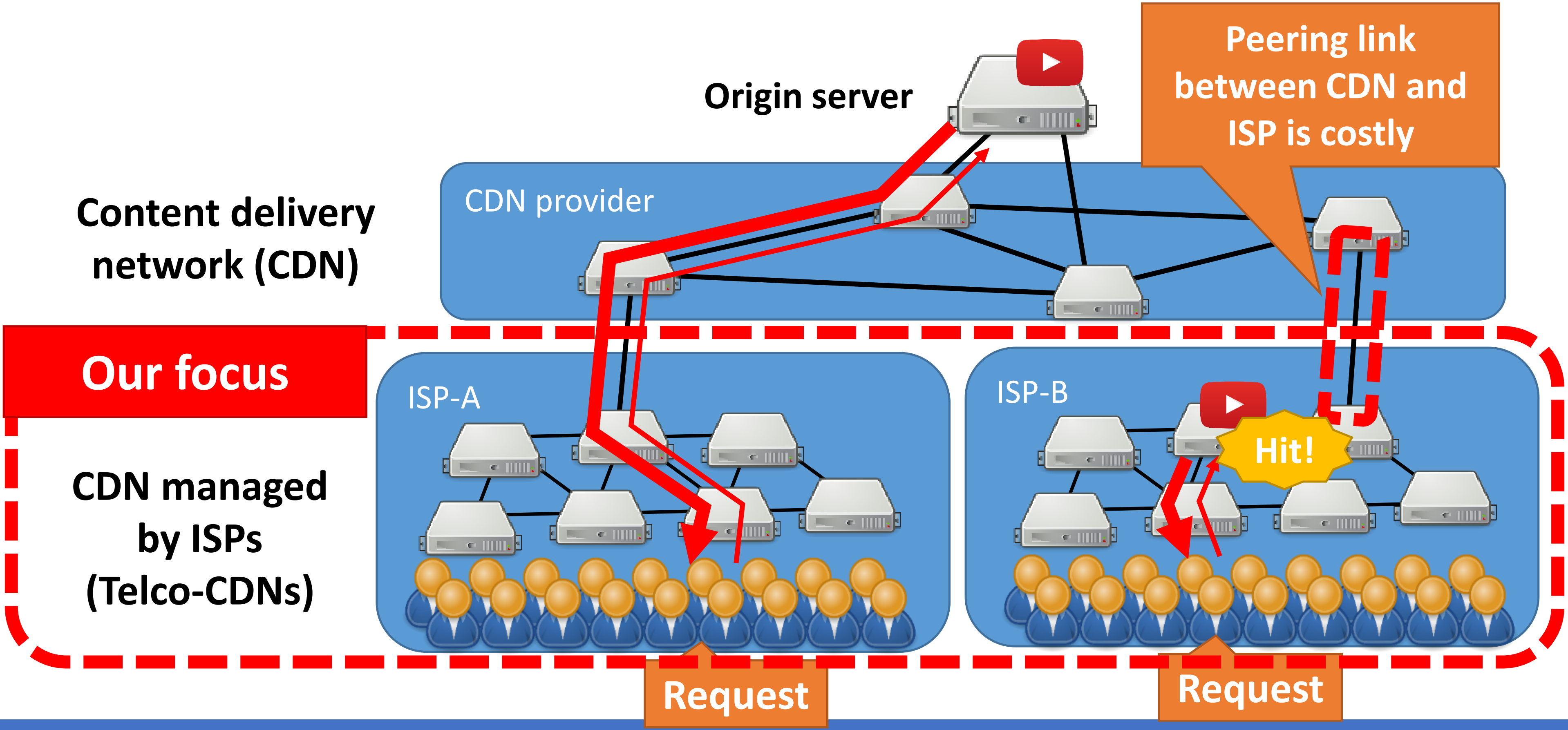


Cache server stores a copy of the data when it passes

Fetching the content from the cache server

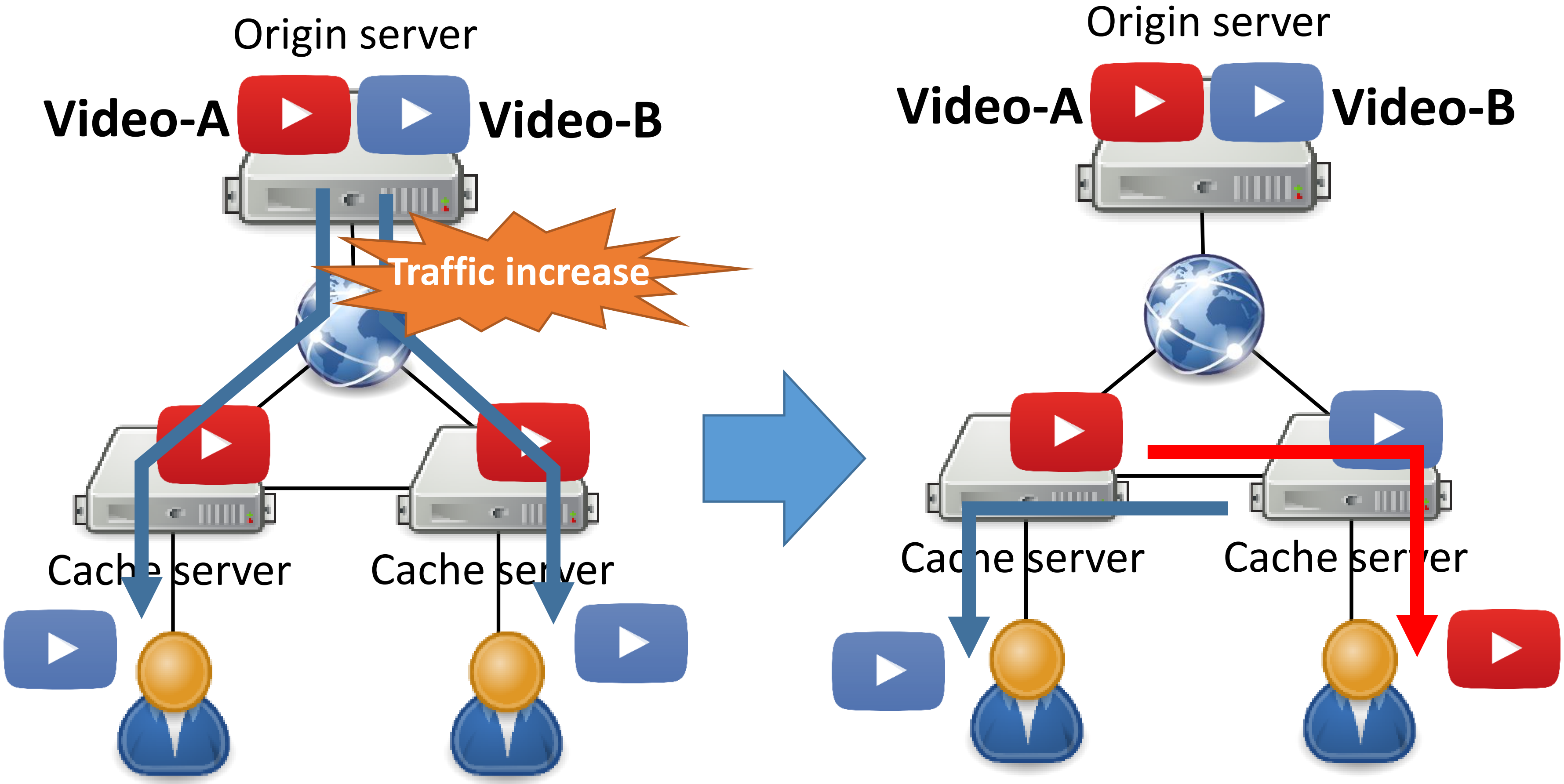
Tiered cache networks

- It is better to **complete requests in an ISP network** to reduce traffic and communication costs


Origin server
Peering link between CDN and ISP is costly
Content delivery network (CDN)
CDN provider
Our focus
ISP-A
ISP-B
Hit!
CDN managed by ISPs (Telco-CDNs)
Request
Request

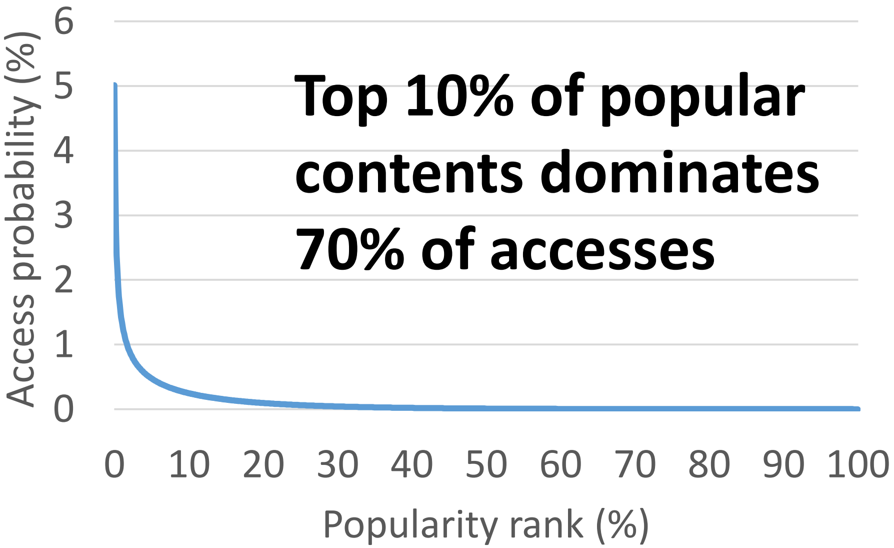
Cooperative caching

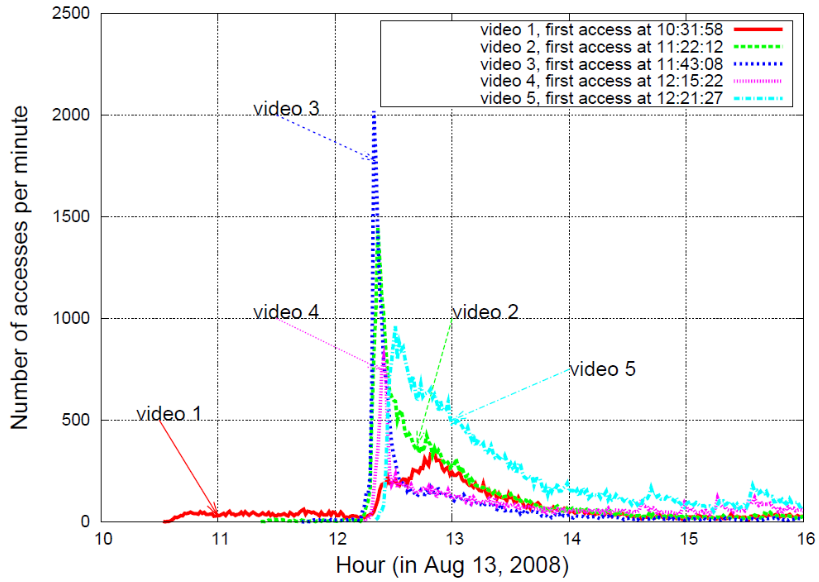
- **Increasing effective storage size** by grouping several cache servers



Characteristics of video accesses

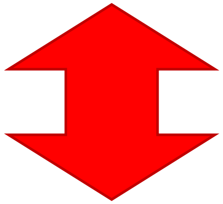
- **Skewed accesses** [Cheng13]:
  Most accesses request limited popular contents
- **Rapid change in contents’ popularities** [Yin09]:
  Access patterns often change widely due to news and viral communications in SNS


Top 10% of popular contents dominates 70% of accesses

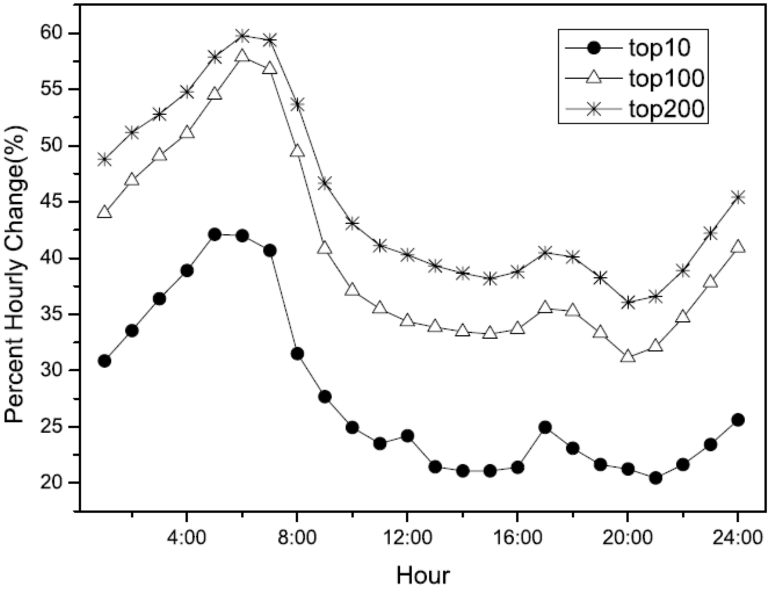


Finding optimal cache allocation

- Calculating **sub-optimal allocations of contents** to minimize the traffic with Genetic Algorithm (GA) [Li13]
  - It takes around **10 hours' calculation**



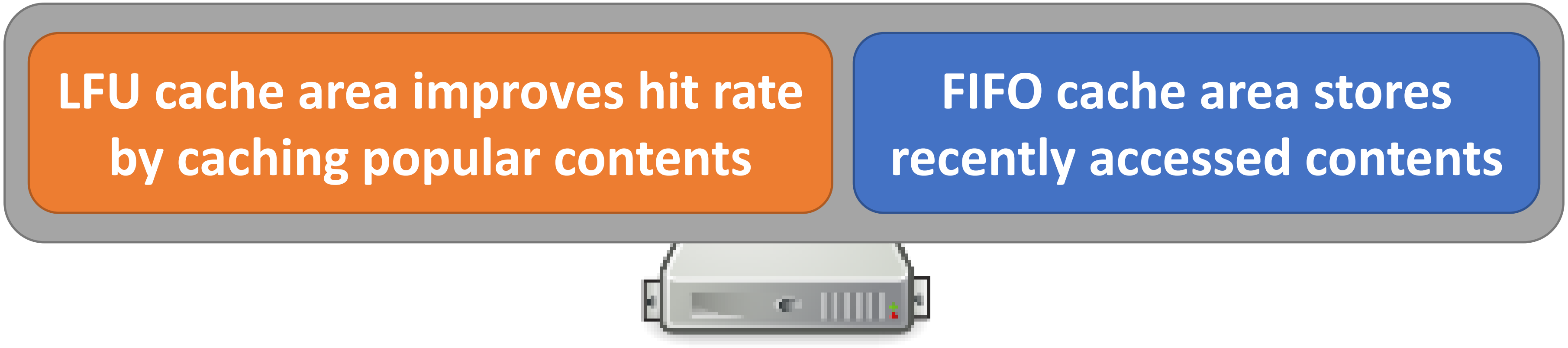
- **Access patterns change 20-60% every hour** [Yu06]
  - Long calculation time causes mismatches in the allocation



Hourly popularity change

LFU/FIFO hybrid caching [Zhou15]

- LFU/FIFO hybrid caching **improves cache hit rate** and **follow changes in access patterns**
  - LFU: Improving hit rate of each cache server
  - FIFO: Following change in access patterns
- It **does not support cooperative caching**



LFU cache area improves hit rate by caching popular contents

FIFO cache area stores recently accessed contents

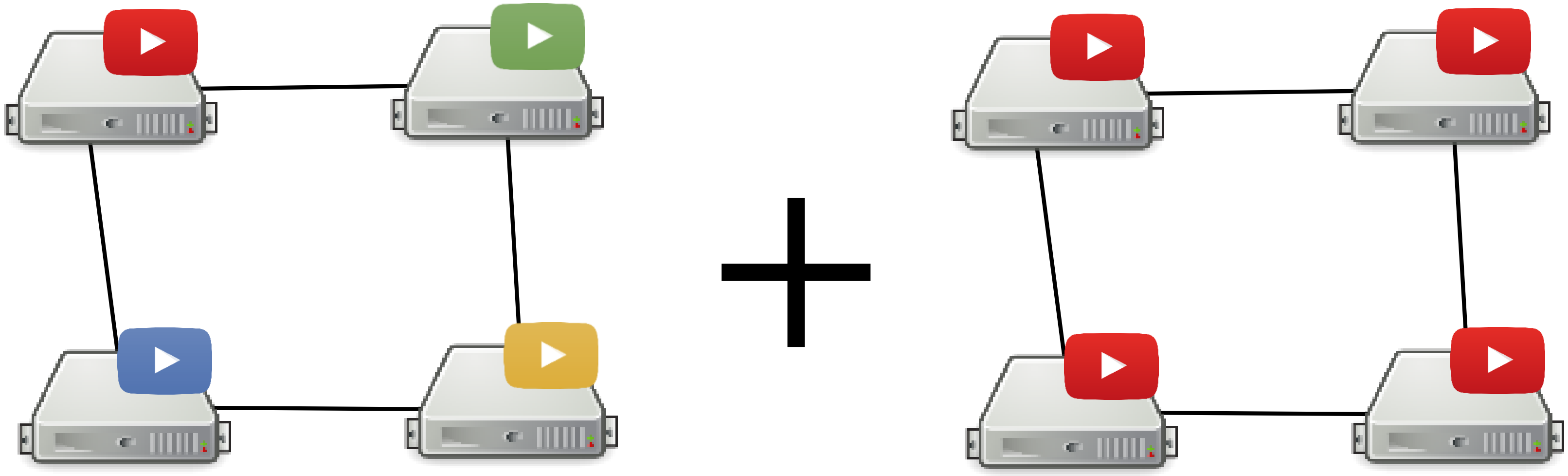
LFU/FIFO Hybrid Cache

Outline

- **Introduction**
  - Rapid growth of video traffic
  - Efficient utilization of cache servers
- **Proposal**
  - Adjusting cache distribution by a simple scheme
  - Following rapid change in access pattern by LFU/LRU hybrid caching
- **Evaluation**
  - Traffic reduction and computational overhead compared with a sub-optimal result
- **Conclusion and Future work**

Efficient use of cache servers

- A key factor of an efficient cache management is a combination use of **content distribution** and **duplication of popular contents**



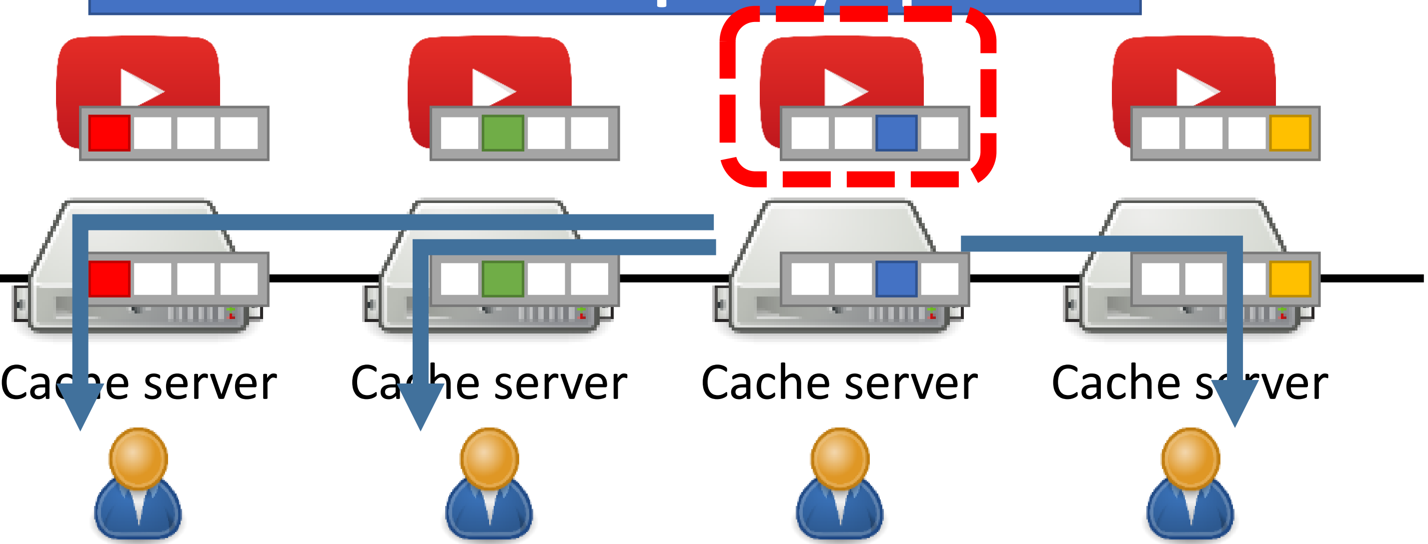
Cache distribution:
increase effective storage size

Duplication of popular contents:
increase hit rates of servers

Content distribution by color tags

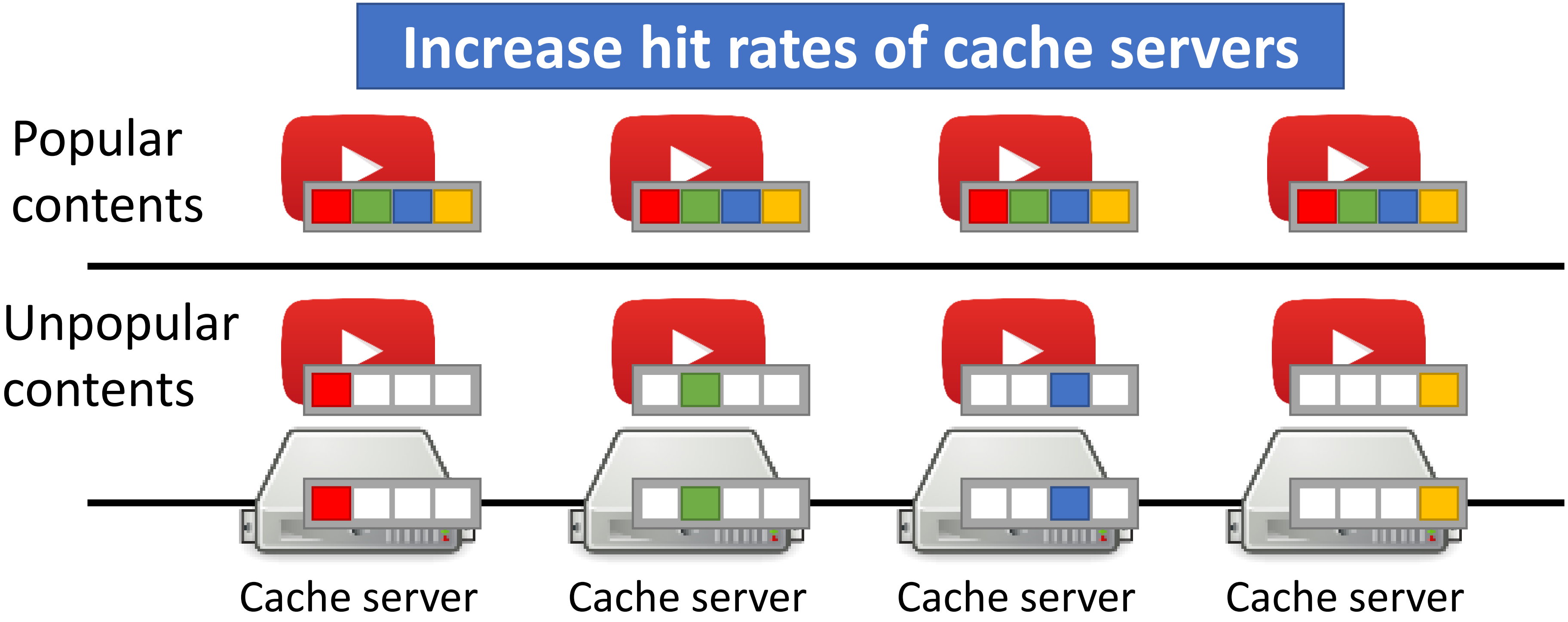
- **Increasing cache capacity** by explicitly storing contents among cache servers
- Grouping and associating cache servers and contents **with color tags with a specific color**
  - Each cache server stores contents if the color matches

Increase cache capacity up to X4



Content duplication by color tags

- **Eliminating traffic** among cache servers
- **Duplicate popular contents** by **applying multiple colors** to them to increase hit rates

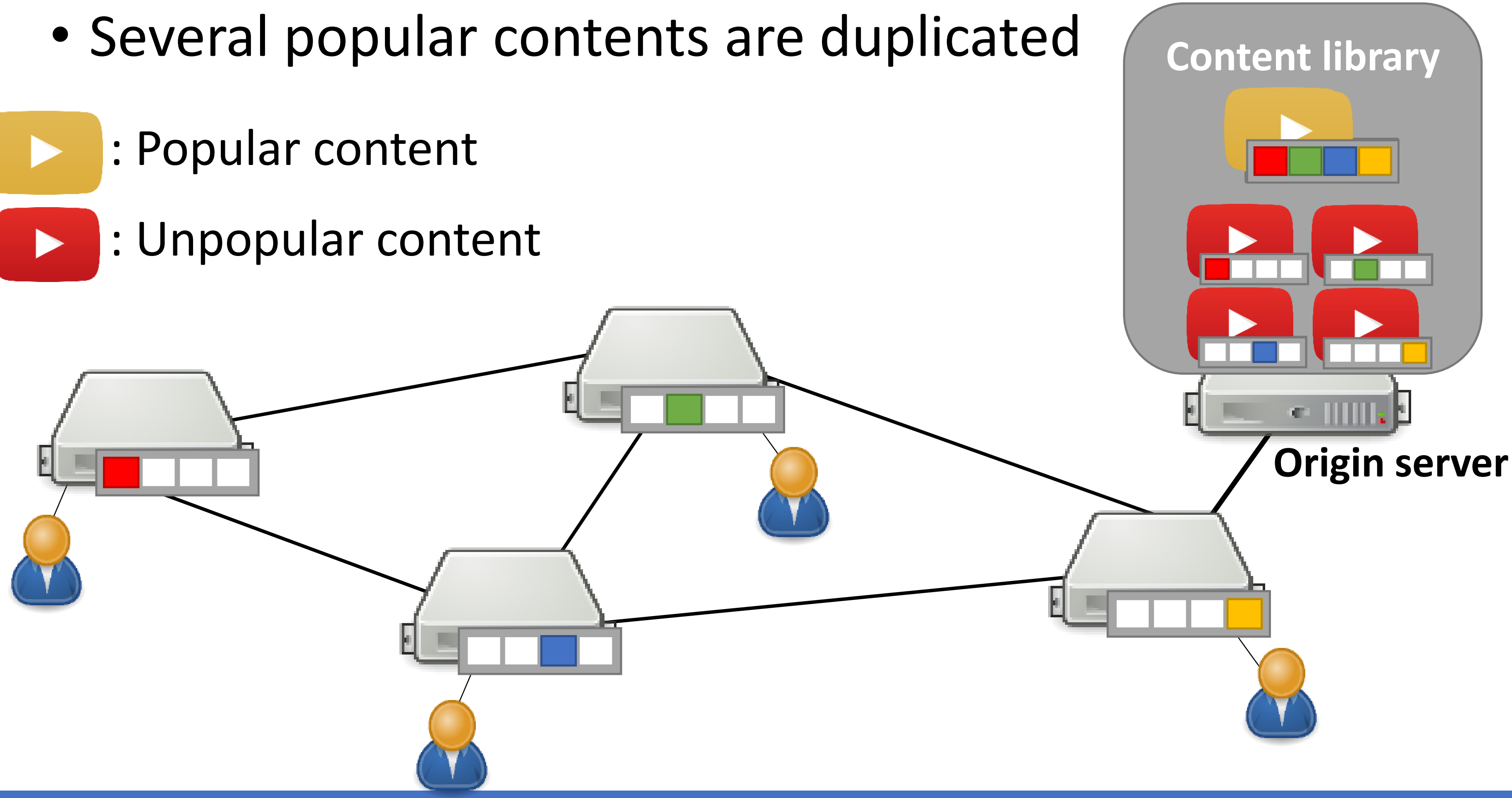


Example cache distribution

- Contents are basically distributed
- Several popular contents are duplicated

▶ : Popular content

▶ : Unpopular content



Preparing color tags

- A color tag is a set of bits, and each bit stands for a specific color

🟥🟩🟦🟨 1111

🟥🟩🟦⬜ 1110

🟥🟩⬜⬜ 1100

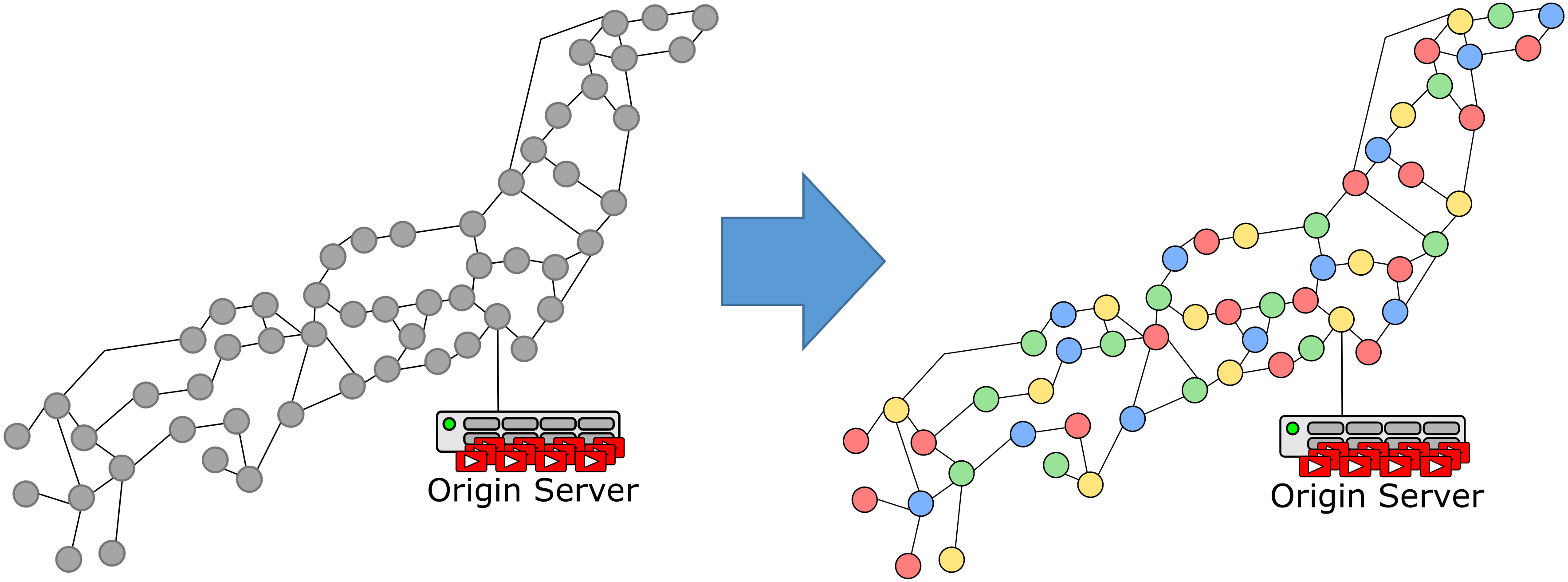
🟥⬜⬜⬜ 1000

- Popular contents have tags with many 1-bit to increase hit rates

# of colors	R	G	B	Y
4	1	1	1	1
3	1	1	1	0
	1	1	0	1
	1	0	1	1
	0	1	1	1
2	1	1	0	0
	1	0	1	0
	1	0	0	1
	0	1	1	0
	0	1	0	1
	0	0	1	1
1	1	0	0	0
	0	1	0	0
	0	0	1	0
	0	0	0	1
0	0	0	0	0

Coloration of cache networks

- Each cache server is preliminarily colorized with a specific color like the **four-color theorem**
  - For a case study, we colorized the network by preferring longer distances between the same colors



Coloration of contents

- **Sorting contents** by their popularities and **set color tags in a cyclic fashion**

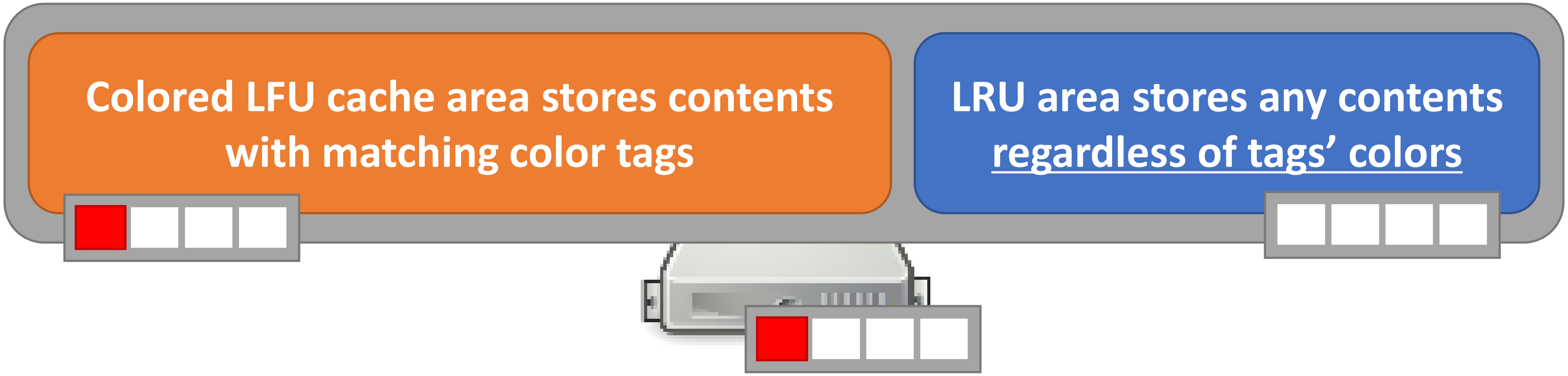
# of colors	R	G	B	Y
4	1	1	1	1
3	1	1	1	0
	1	1	0	1
	1	0	1	1
	0	1	1	1
2	1	1	0	0
	1	0	1	0
	1	0	0	1
	0	1	1	0
	0	1	0	1
	0	0	1	1
1	1	0	0	0
	0	1	0	0
	0	0	1	0
	0	0	0	1
0	0	0	0	0

Rank	Popularity class	Content name	Tag
1	High	Video01.mp4	1111
2	High	Video02.mp4	1111
		⋮	
11	Mid-High	Video11.mp4	1110
12	Mid-High	Video12.mp4	1101
13	Mid-High	Video13.mp4	1011
14	Mid-High	Video14.mp4	0111
15	Mid-High	Video15.mp4	1110
		⋮	
130	Middle	Video130.mp4	1100
131	Middle	Video131.mp4	1010
132	Middle	Video132.mp4	1001
133	Middle	Video133.mp4	0110

Set color tags from popular contents

Following rapid access changes

- We adopt a **hybrid caching scheme** with colored LFU and no-color Modified LRU [Vleeschauwer11] areas
  - Modified LRU achieves better hit rate than LRU
- **Colored LFU area** stores contents with matching tags, while the **Modified LRU** area stores contents without matching tags



Colored LFU cache area stores contents with matching color tags

LRU area stores any contents regardless of tags' colors

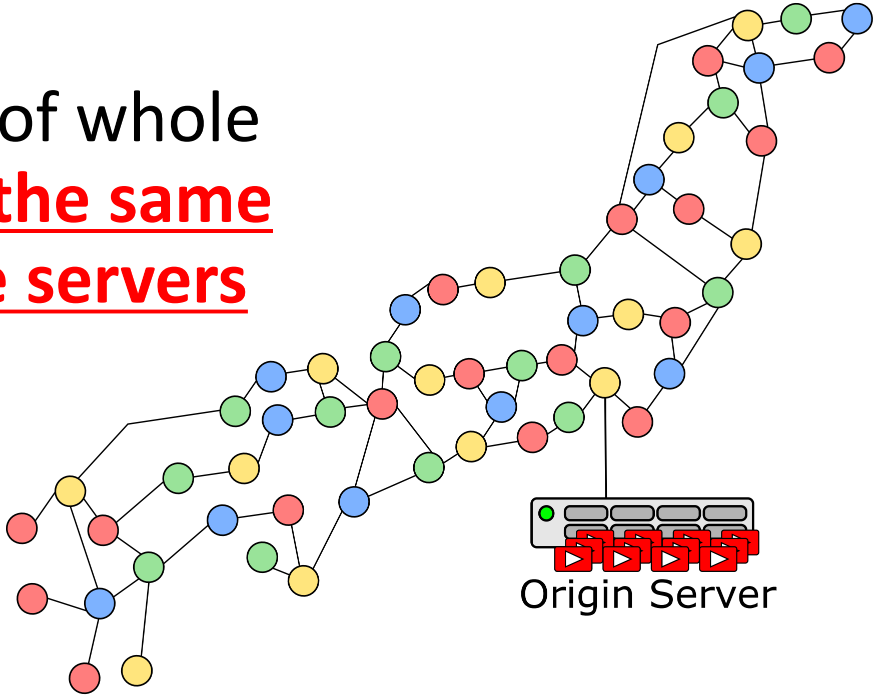
Hybrid Cache with Red color tag

Outline

- **Introduction**
  - Rapid growth of video traffic
  - Efficient utilization of cache servers
- **Proposal**
  - Adjusting cache distribution by a simple scheme
  - Following rapid change in access pattern by LFU/LRU hybrid caching
- **Evaluation**
  - Traffic reduction and computational overhead compared with a sub-optimal result
- **Conclusion and Future work**

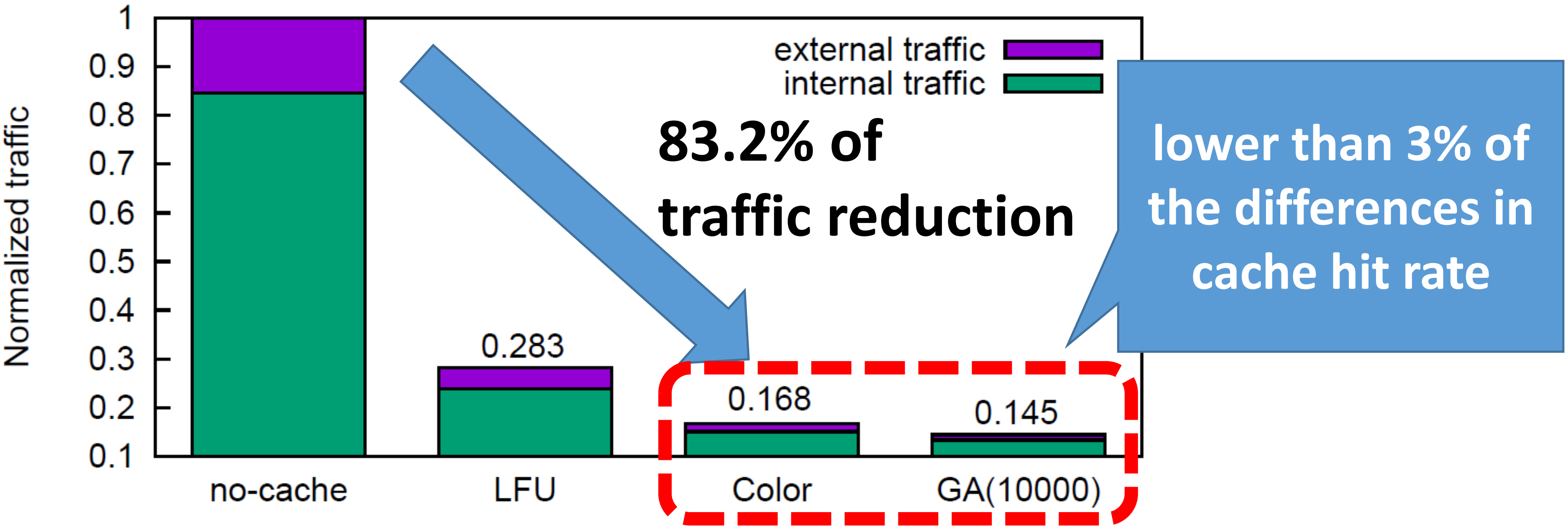
Evaluation environment

- **NTT-like topology** in Japan [Arteta07]
- Origin server is connected to a cache server in Tokyo
- Skewed access pattern with **YouTube video traffic** [Cheng13]
- Each cache can store 10% of whole contents, which is almost **the same capacity as Netflix's cache servers**



Traffic reduction

- Proposed color-cache scheme could **achieve close to the sub-optimal result** calculated by GA



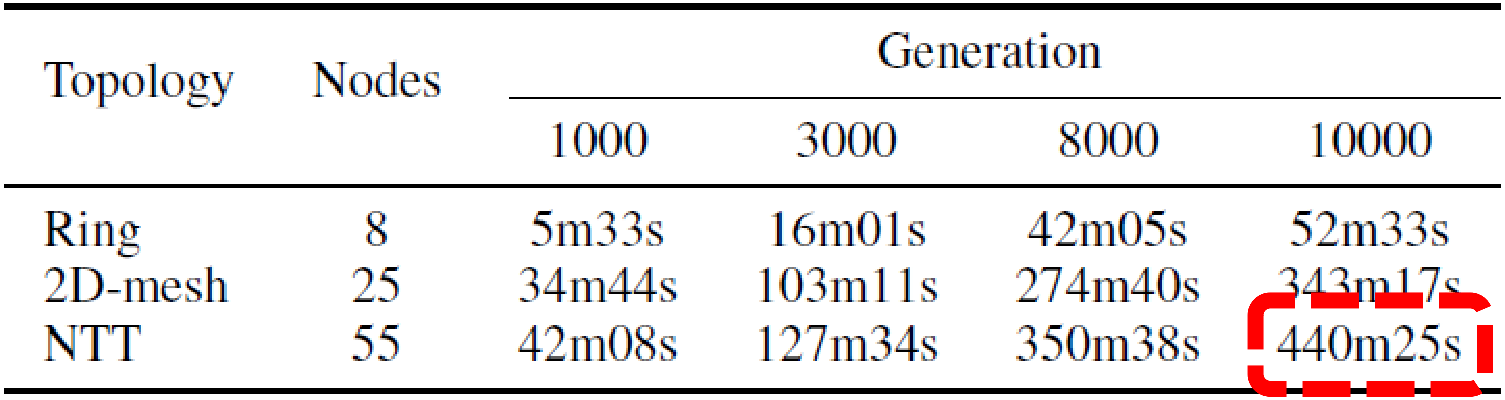
83.2% of traffic reduction

lower than 3% of the differences in cache hit rate

Computational overhead

- **Colorization overhead is limited to a few seconds** since it only have to sort and update tags in a cyclic fashion

Table: Computational time of GA



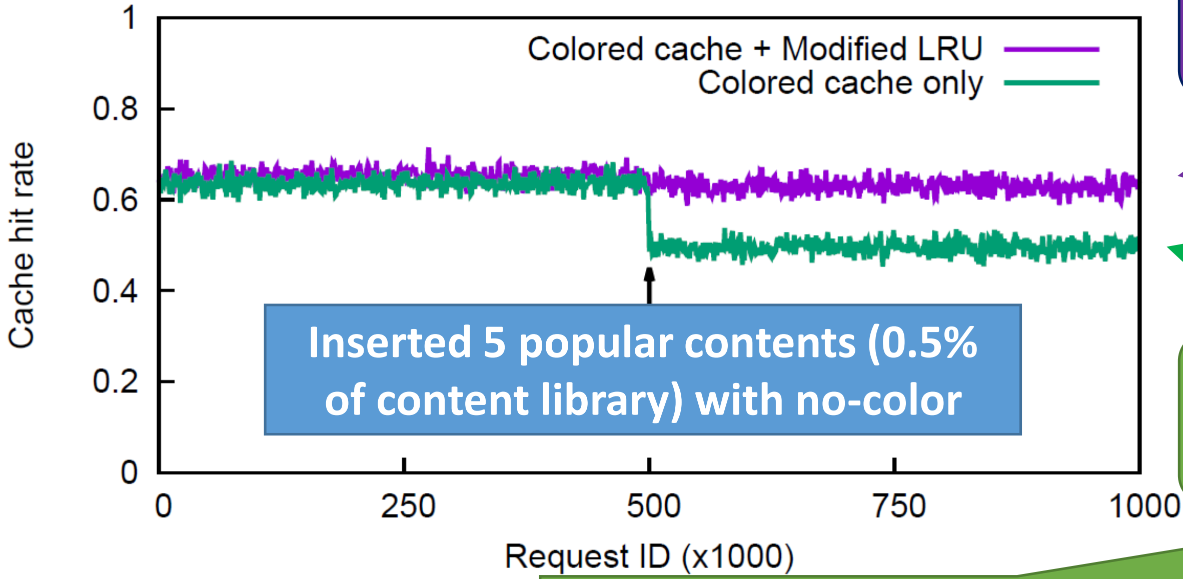
Topology	Nodes	Generation			
		1000	3000	8000	10000
Ring	8	5m33s	16m01s	42m05s	52m33s
2D-mesh	25	34m44s	103m11s	274m40s	343m17s
NTT	55	42m08s	127m34s	350m38s	440m25s

GA takes more than 7 hours until the conversion when using recent Core i7 CPU

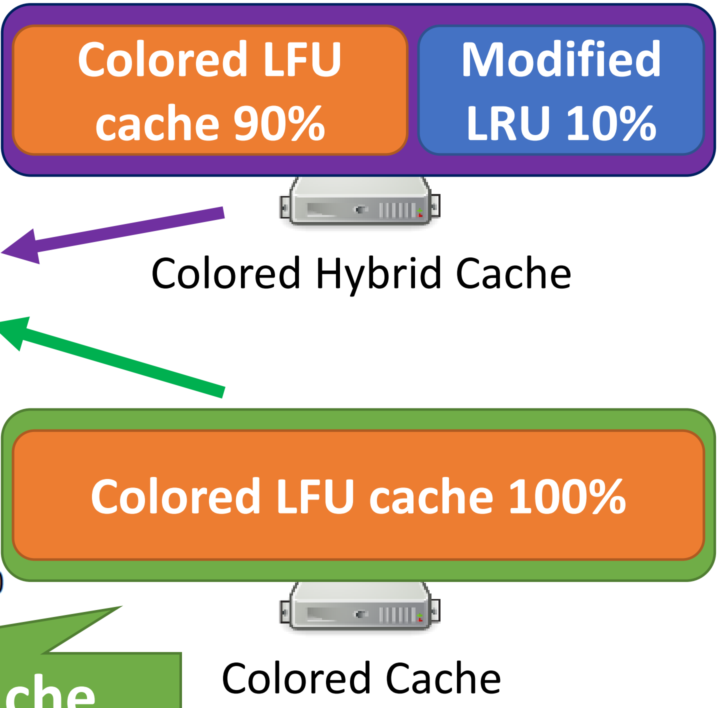
Following dynamic accesses

- The **Colored hybrid caching** scheme could **maintain its hit rate** even when new contents are inserted

Colored hybrid caching could limit the degradation to 2.3%



Inserted 5 popular contents (0.5% of content library) with no-color



Single colored LFU cache drops the hit rate by 13.9%

Conclusion and Future work

- **Proposal**: A light-weight scheme to manage cache servers by focusing on **content distribution** and **duplication** with a simple colorization scheme
- **Evaluation**:
  - Colored caching scheme could **achieve close to the sub-optimal result** with less than 3% of difference in hit rates
  - Computational **overhead is limited to a few seconds**
  - Colored hybrid caching scheme could also **follow the rapid change in access patterns** limiting the degradation to 2.3%
- **Future work**
  - More efficient ways to colorize cache servers and routing algorithms for further enhancing the traffic reduction