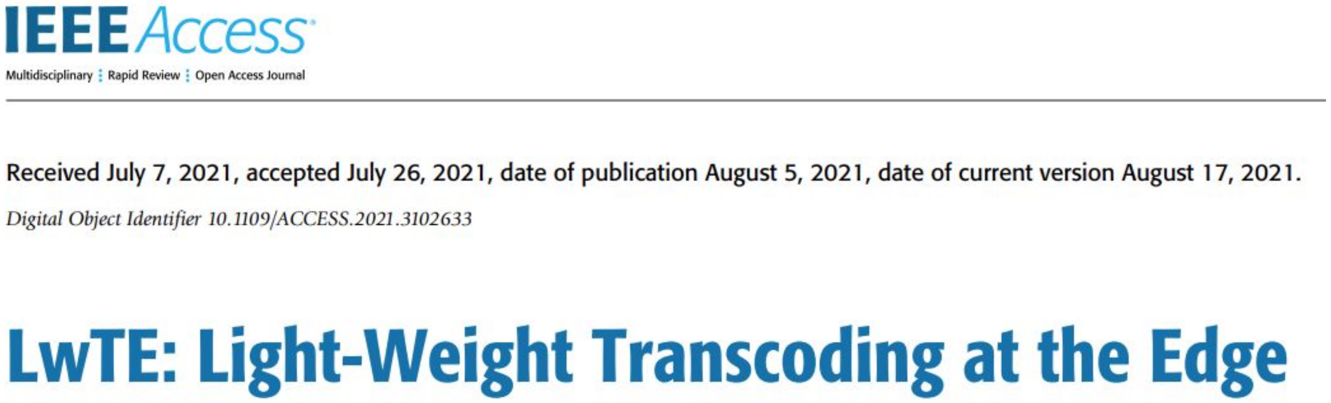
IMDEA Networks Institute

December 9, 2021, Madrid, Spain

LwTE: Light-Weight Transcoding at the Edge

Farzad Tashtarian,
farzad.tashtarian@aau.at | https://www.tashtarian.net/

Christian Doppler laboratory ATHENA | Klagenfurt University | Austria
https://athena.itec.aau.at/



IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 7, 2021, accepted July 26, 2021, date of publication August 5, 2021, date of current version August 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3102633

LwTE: Light-Weight Transcoding at the Edge

Alireza Erfanian Hadi Amirpour Farzad Tashtarian Christian Timmerer Hermann Hellwagner

Christian Doppler Laboratory ATHENA, Institute of Information Technology (ITEC),
Alpen-Adria-Universität Klagenfurt, 9020 Klagenfurt, Austria

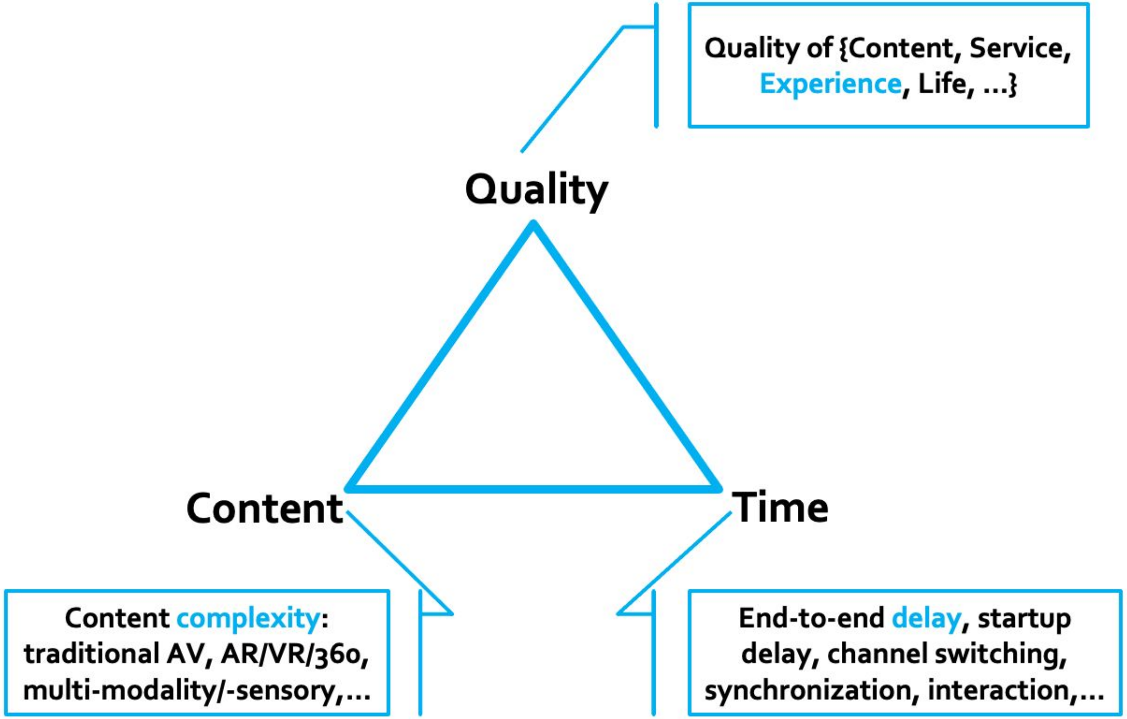
“By 2022, Internet video will represent 82% of all Internet traffic.”

Cisco Visual Networking Index: Forecast and Trends, 2017–2022 (White Paper), Cisco, February 2019.

Agenda

- Introduction
- Transcoding at the edge
  - Challenges and opportunities
  - Transcoding for VOD and Live streaming
- LwTE - for VOD applications
- Results, Conclusion and Future work
- Q&A

Multimedia Systems Challenges and Tradeoffs



Quality of {Content, Service, Experience, Life, ...}

Quality

Content

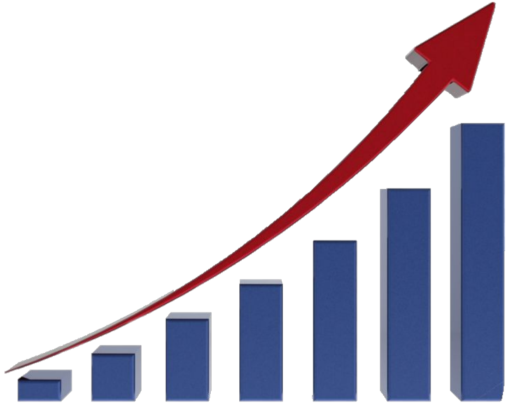
Time

Content complexity: traditional AV, AR/VR/360, multi-modality/-sensory,...

End-to-end delay, startup delay, channel switching, synchronization, interaction,...

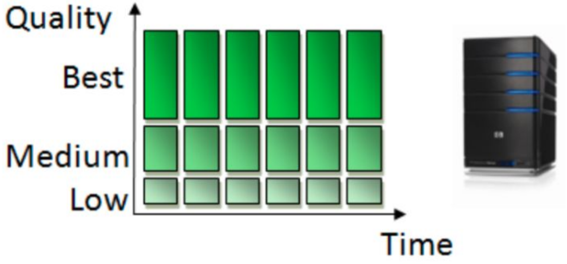
Basic figure by Klara Nahrstedt, University of Illinois at Urbana–Champaign, IEEE MIPR 2018

Motivation ...



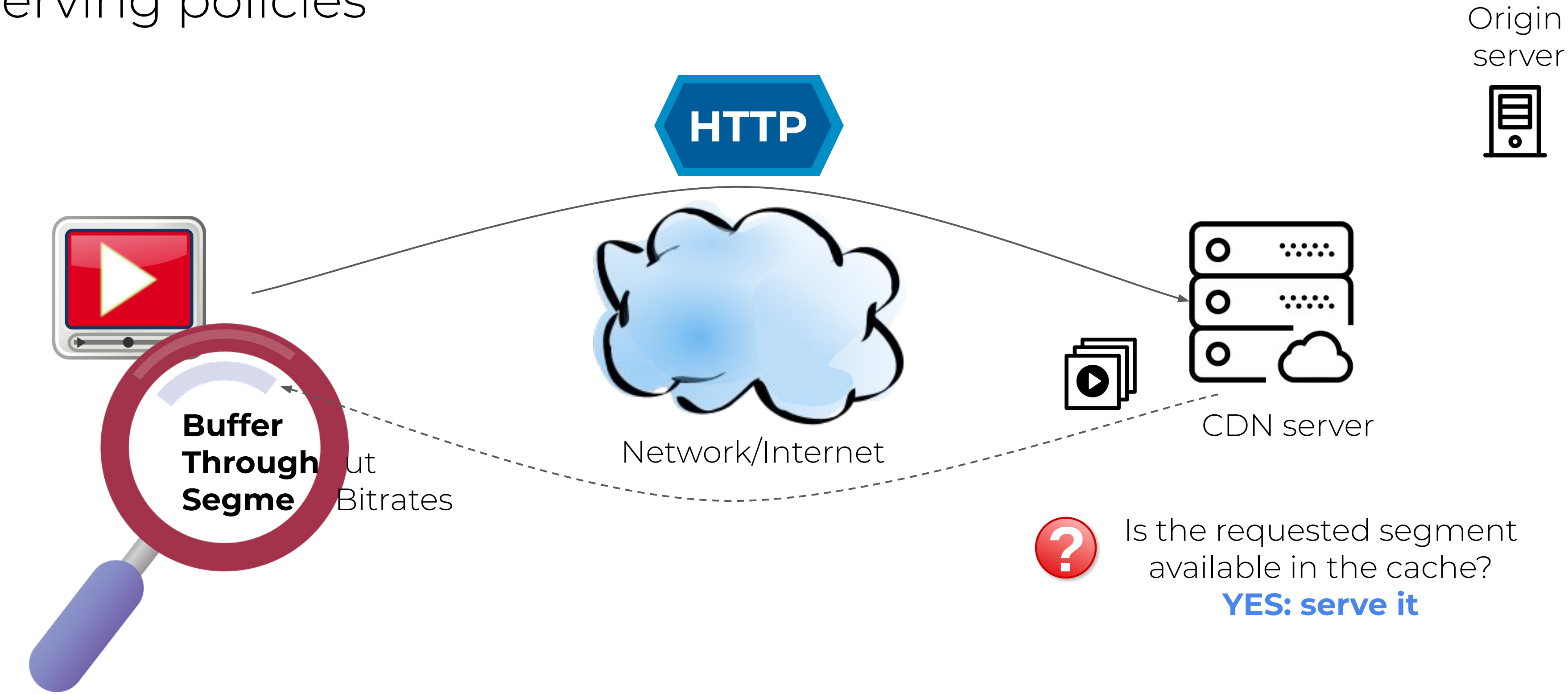
Increasing demands
on video streaming

Heterogeneous
environment



Provide streaming services
with different quality levels

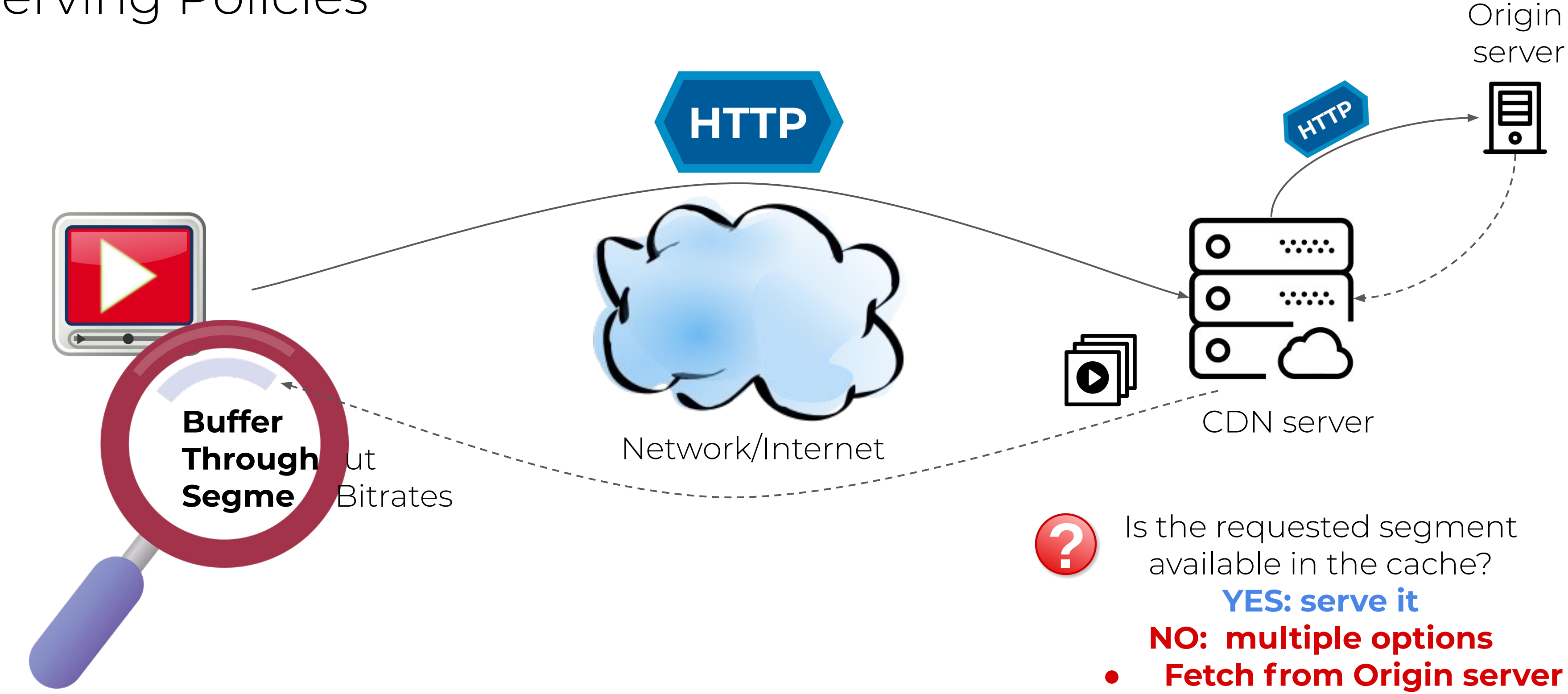
Serving policies



Is the requested segment available in the cache?

YES: serve it

Serving Policies



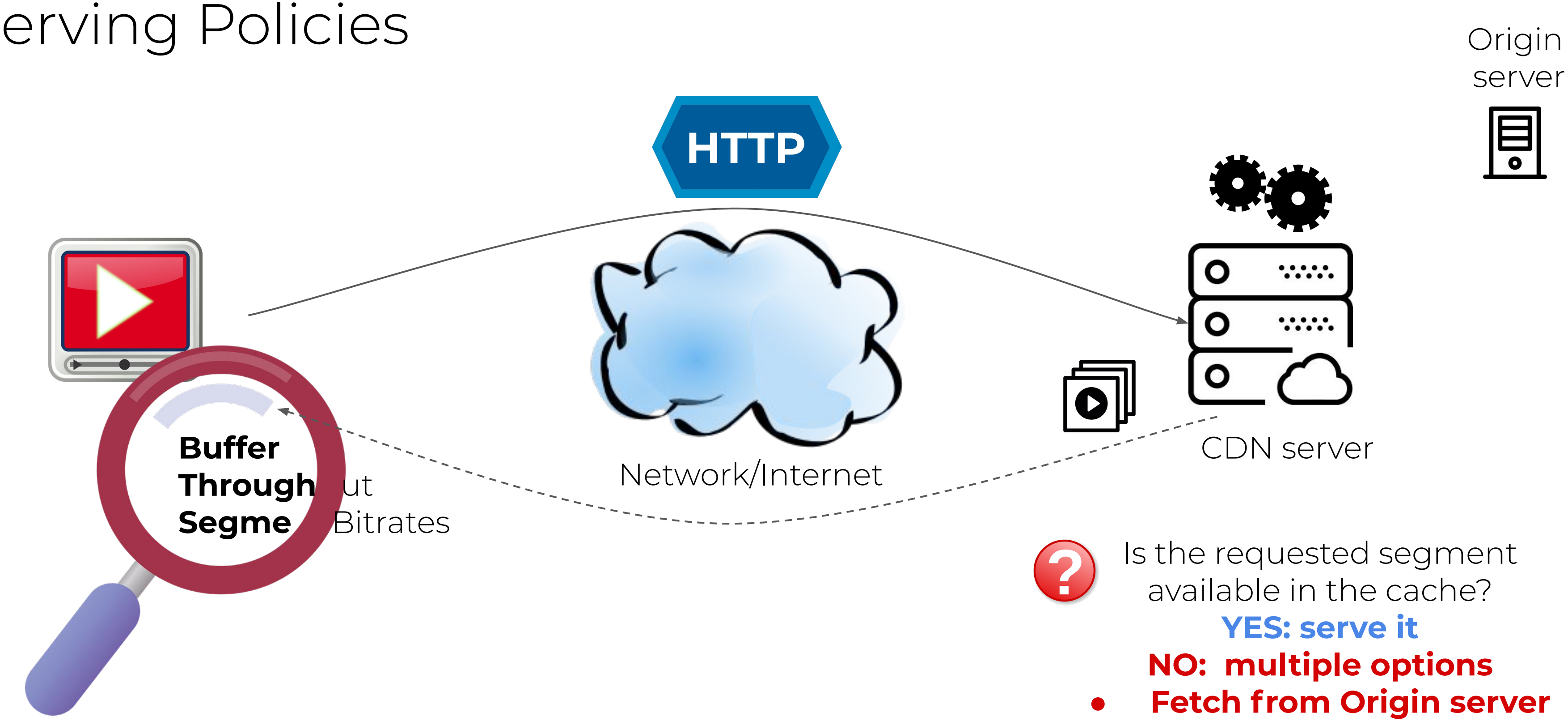
Is the requested segment available in the cache?

YES: serve it

NO: multiple options

- **Fetch from Origin server**

Serving Policies



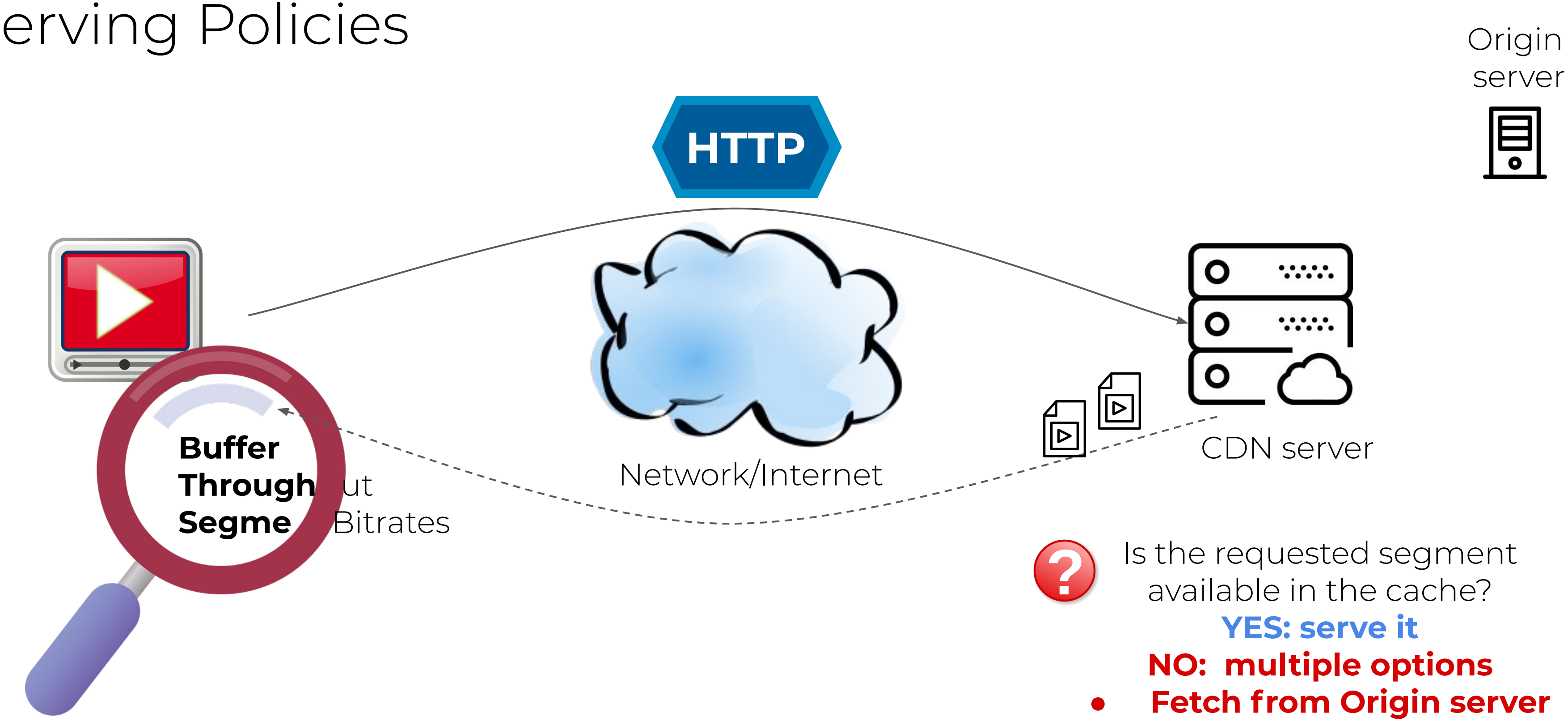
Is the requested segment available in the cache?

YES: serve it

NO: multiple options

- **Fetch from Origin server or another CDN server**
- **Transcode at the edge**

Serving Policies



Is the requested segment available in the cache?

YES: serve it

NO: multiple options

- **Fetch from Origin server or another CDN server**
- **Transcode at the edge**
- **Serve with lower quality**

Strategies of Employing Transcoding

The **highest bitrate** is stored, and the remaining bitrates are transcoded online

The **cloud paradigm** with virtually unlimited resources enables many cloud service providers like **Amazon Web Services** or **Google Cloud Platform** to provide cost-effective transcoding services.

Transcoding tasks are computationally intensive and **time-consuming**, which impose significant operational costs on service providers.

On-the-fly transcoding

Strategies of Employing Transcoding

The pre-transcoding method commonly **used in the industry** store all bitrates to meet all users' requests.

It incurs high **overhead in storage** to store all bitrates, especially for video segments/bitrates that are rarely requested.

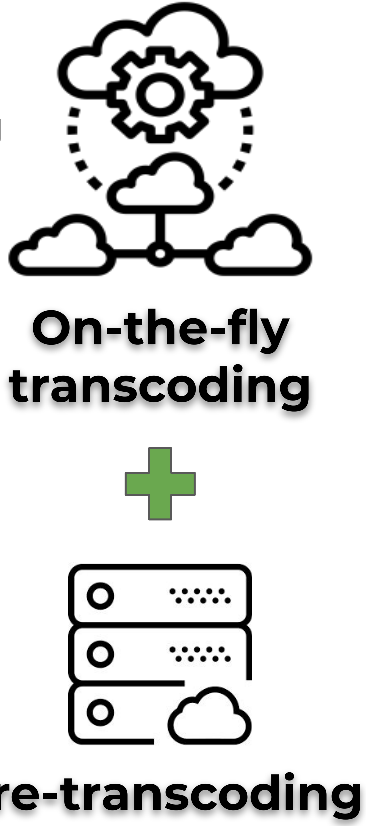
Although storage is becoming cheaper, this approach **is not cost-efficient.**

pre-transcoding

Hybrid Approach

Strategies of Employing Transcoding

Some studies try to **minimize video streaming** costs by combining **on-the-fly transcoding** and **pre-transcoding** approaches [7], [10], [11] by trading off storage costs and computation costs, considering various constraints.



On-the-fly transcoding

+

pre-transcoding

The **main issues** of partial transcoding are:

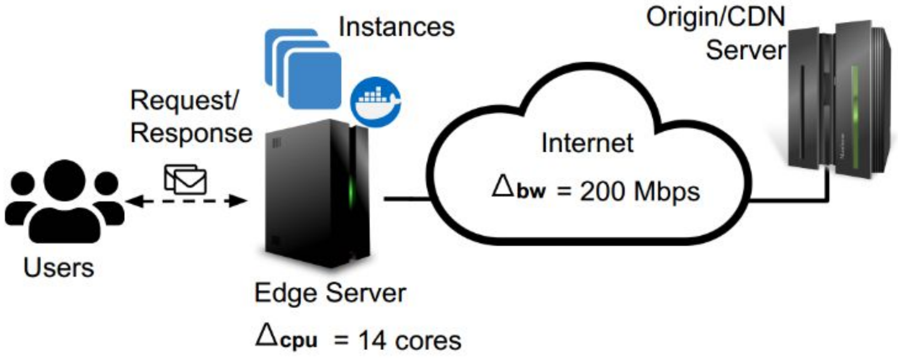
- determining the optimal set of video segments/bitrates which should be stored and
- the high computation time of the transcoding process that imposes noticeable computation cost and delay

Challenges of Transcoding at the Edge

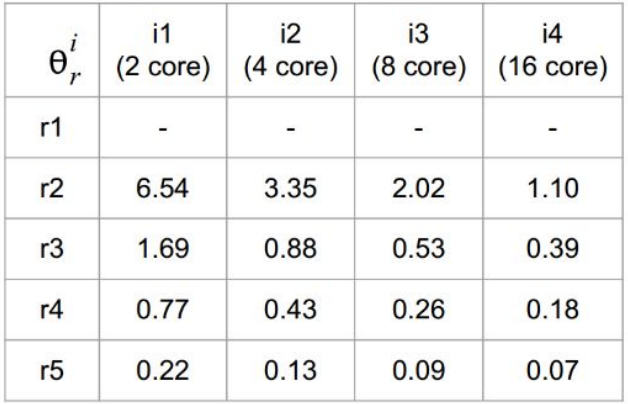
- For **which requests** do we run transcoding function? **Which properties** of requests, videos, segments, and bitrates should be considered.?
- **Where** do we **run** transcoding function?
- What is the **configuration** of **machine/VNF** running transcoding function?
- **Which bitrate should be** used to be transcoded to the requested one?

Challenges of Transcoding at the Edge

A simple example



(a) LwTE-Live in a scenario with five requested representations (r1-r5) and four instances (i1-i4)



θ_r^i	i1 (2 core)	i2 (4 core)	i3 (8 core)	i4 (16 core)
r1	-	-	-	-
r2	6.54	3.35	2.02	1.10
r3	1.69	0.88	0.53	0.39
r4	0.77	0.43	0.26	0.18
r5	0.22	0.13	0.09	0.07

(b) transcoding time in sec.

Opportunities of Transcoding at the Edge

- Decreasing response **delay**
- Optimizing **backhaul** traffic
- Minimizing the **costs** of storage and bandwidth
- Increasing **cache hit** ratio

Some related work

Hybrid Solution

[21]

Minimize storage and computation costs

Considering segment popularity and a weighted transcoding graph

- For VoD applications
- they did not consider the video quality drop caused by using lower bitrate segments for transcoding

Hybrid Solution

[11], ([26] in 5G)

Minimize storage, transcoding, and bandwidth costs

Transcoding from the highest bitrate
Consider popularity of video/segment

- store the full representation set for a few popular videos
- only keep the highest bitrate for the rest

Hybrid Solution

[19]

Minimize the backhaul network cost

collaborative joint caching and transcoding

- Problem as an Integer Linear Program (ILP)

Federated-Fog Delivery Network

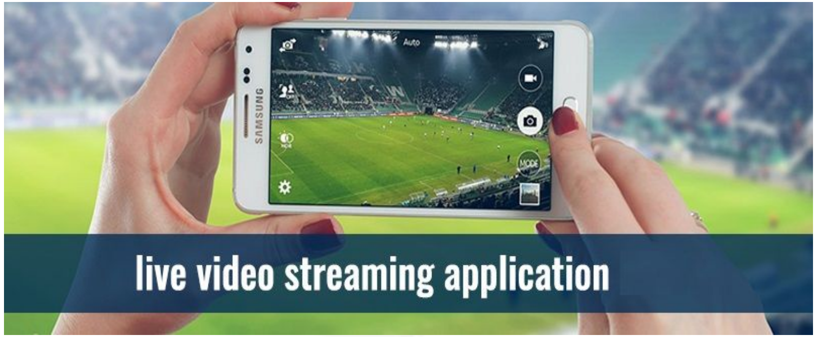
[24]

Maximize QoE

determine edge servers for transcoding operations

- A distributed platform at the edge named Federated-Fog Delivery Network (F-FDN)

How to utilize transcoding for VoD and Live Streaming Applications?



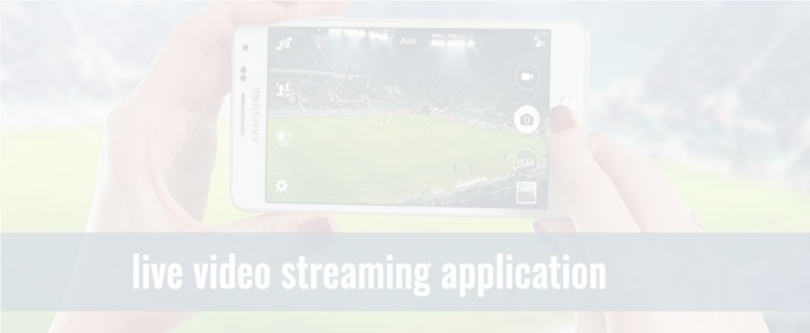
Live Streaming Applications

- Delay sensitive
- Serving policy
  - Transcoding
  - Fetch from CDN/Origin
- Cost functions
  - Computation
  - Bandwidth
- Resource limitations

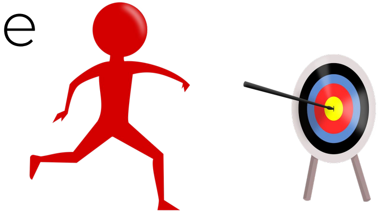
VOD Applications

- Serving policy:
  - **Cache**
  - Transcoding
  - Fetch from CDN/Origin
- Cost functions:
  - Storage
  - Computation
  - Bandwidth
- Resource limitations

Transcoding at the Edge for VoD Applications



Light-Weight Transcoding (LwTE) at the Edge for VoD Applications



01 **Extracts metadata**

- During the encoding process
- Employs it during the transcoding process at the edge

02 **Minimizing the total cost**

- **Storage and computation (transcoding) costs,**
- Prove its NP-completeness.
- Serving policy: **from cache and transcoding**

03 **Heuristic algorithm**

- To mitigate the time complexity of the proposed MILP model
- determine a near-optimal solution

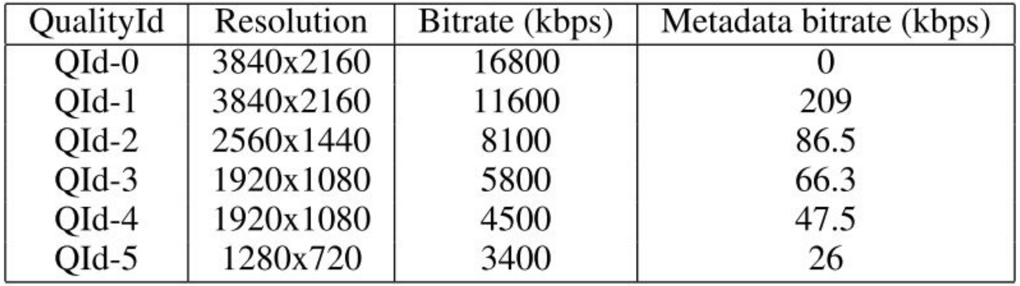
04 **Performance evaluation**

- LwTE achieves at least 80% reduction in transcoding time
- decrease the total cost by up to 70%

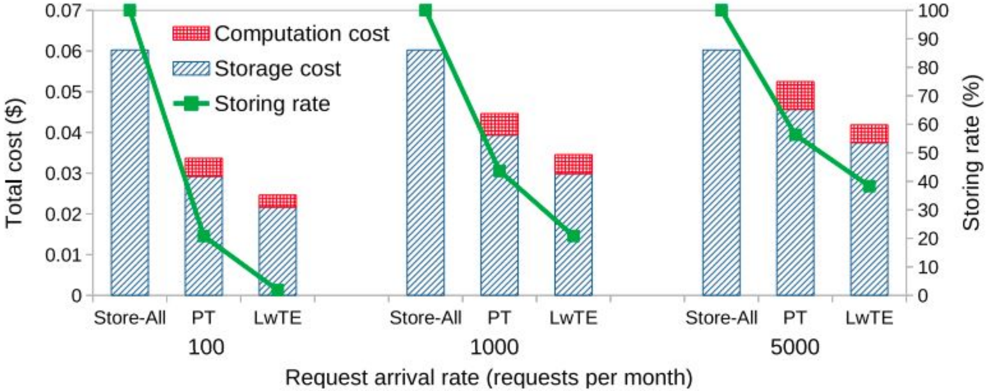
Motivating Example

Compare LwTE with two basic strategies:

- store all bitrates (**Store-All**)
- conventional partial-transcoding (**PT**) [21].

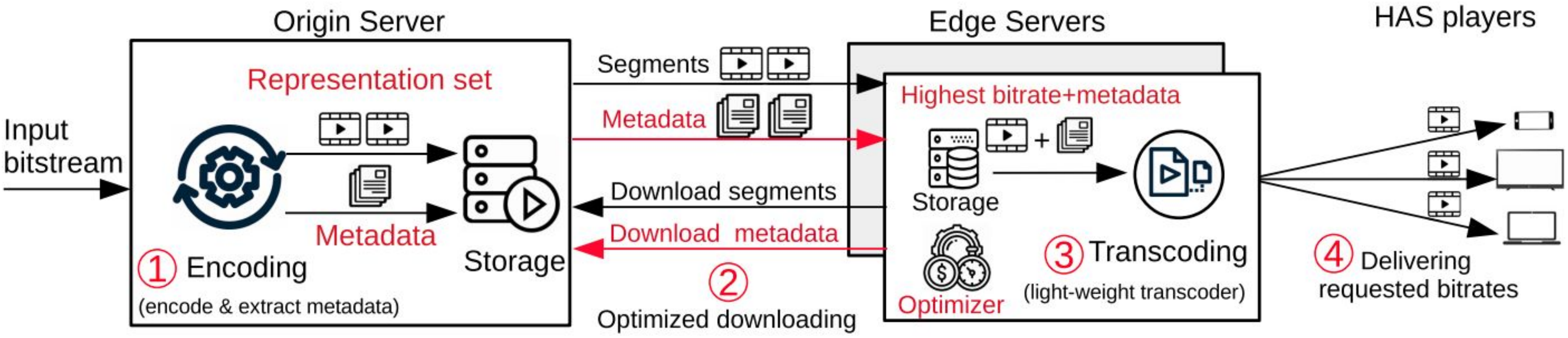


QualityId	Resolution	Bitrate (kbps)	Metadata bitrate (kbps)
QId-0	3840x2160	16800	0
QId-1	3840x2160	11600	209
QId-2	2560x1440	8100	86.5
QId-3	1920x1080	5800	66.3
QId-4	1920x1080	4500	47.5
QId-5	1280x720	3400	26



Proposed LwTE Architecture

Finally, requested bitrates are delivered from the edge server to the clients.



During the encoding process, selected features are extracted and stored as **metadata** for all bitrates (except for the highest bitrate) at **no additional costs**.

For **popular sets**, all video segments/bitrates are downloaded. In **unpopular sets**, only the highest bitrate plus corresponding metadata generated during the encoding process are made available at the edge

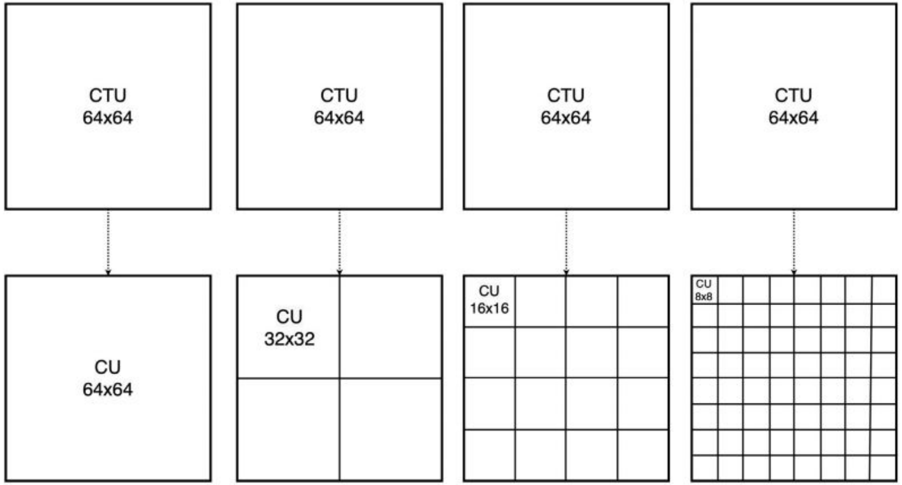
This stage will be applied to those segments/bitrates available in the unpopular set.

Extracting Metadata

In HEVC, frames are divided **into 64 × 64 pixel** blocks, called Coding Tree Units (**CTUs**)

To encode CTUs, each of them is partitioned into **equally sized square blocks** known as Coding Units (CUs)

The **rate distortion cost** is calculated for all of these CUs to find the optimal CTU partitioning structure with the minimum cost.

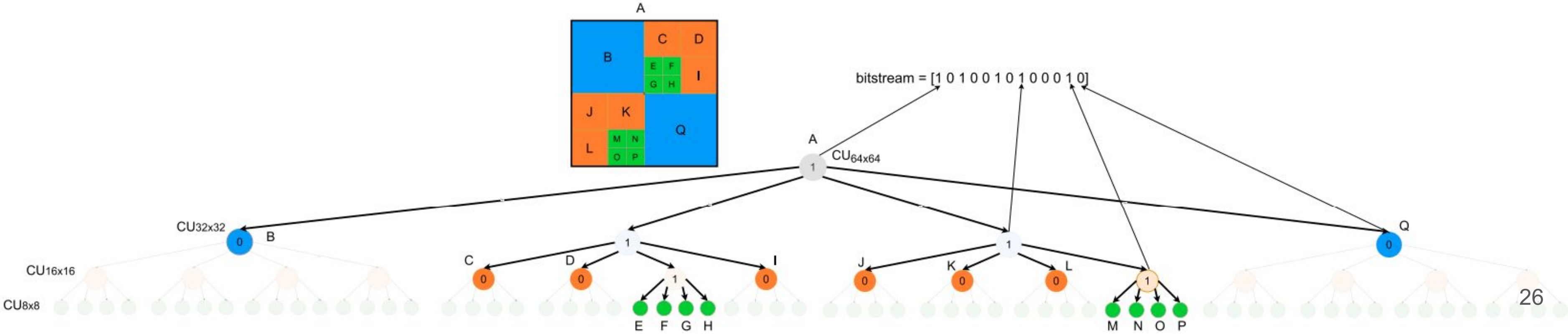


The search to find the optimal CTU partitioning into CUs using a brute-force approach takes the largest amount of time in the encoding process

Extracting Metadata

The search to **find the optimal CTU partitioning** into CUs using a **brute-force** approach takes the largest amount of time in the encoding process.

To avoid a brute-force search process at the edge, we extract the optimal partitioning structure for CTUs during encoding in the origin server and store this as metadata for each segment bitrate except the highest bitrate.



Adding PUs into the Metadata

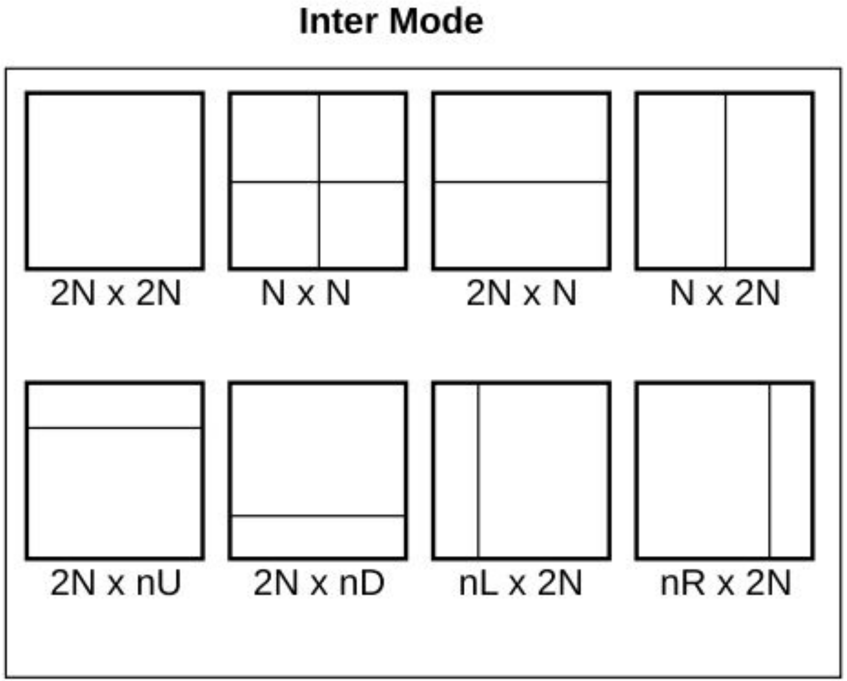
In HEVC, each CU is further split into **Prediction Units (PUs)** [29].

In addition to the CTU partitioning structure, the optimal PU partitioning mode for each CU is extracted and added to the metadata.

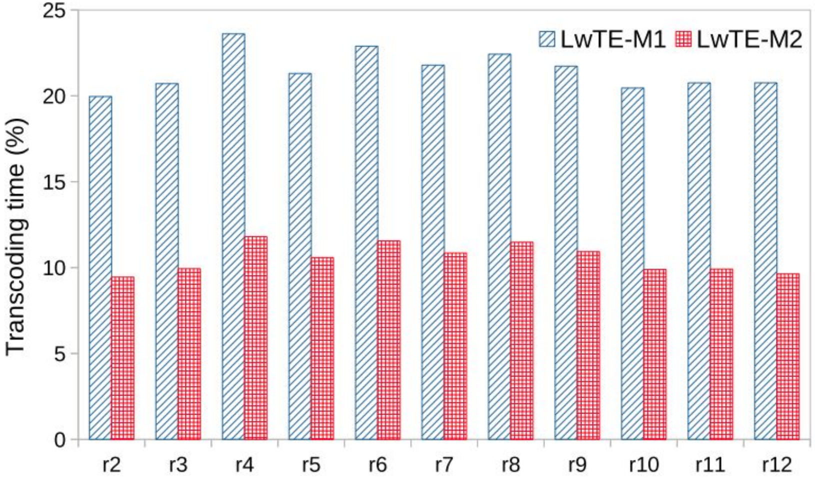
To further reduce the size of the metadata (or bitstream), we use the **Huffman algorithm** to encode the metadata losslessly.

LwTE-M1 (mode1) is used if the metadata contains only the optimal CU partitioning structure,

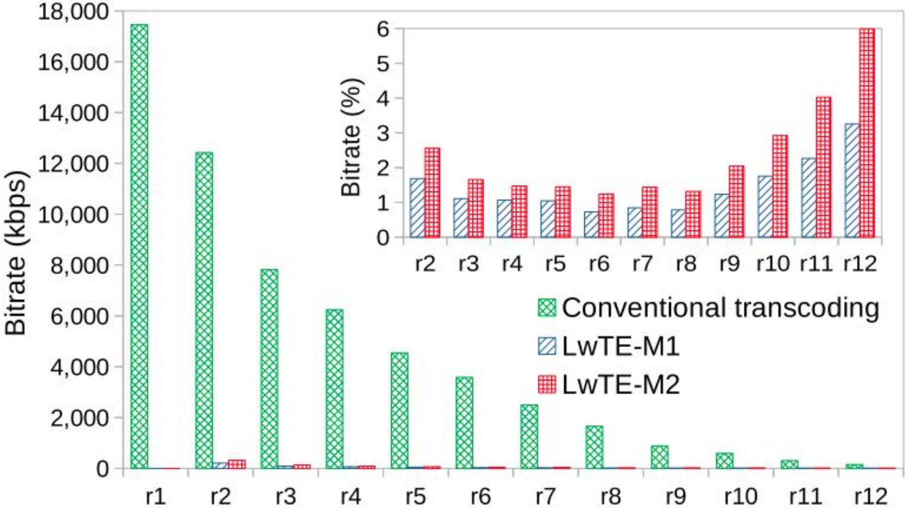
LwTE-M2 (mode2) is used if the metadata contains both the optimal CU partitioning structure and the optimal PU partitioning mode



LwTE transcoding times and bitrates in two different modes



Transcoding times of LwTE-M1 and LwTE-M2 relative to conventional mode's transcoding times for each bitrate in the representation set



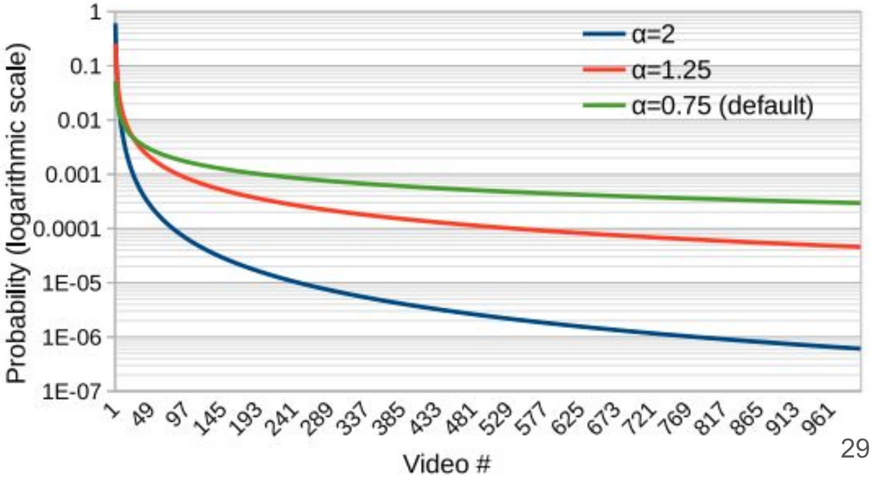
Bitrates of all representations and their corresponding metadata. The embedded plot shows the bitrates of the metadata relative to the corresponding representations.

Access Pattern

A long-tail distribution of access pattern ⇨ a small percentage of videos are requested frequently

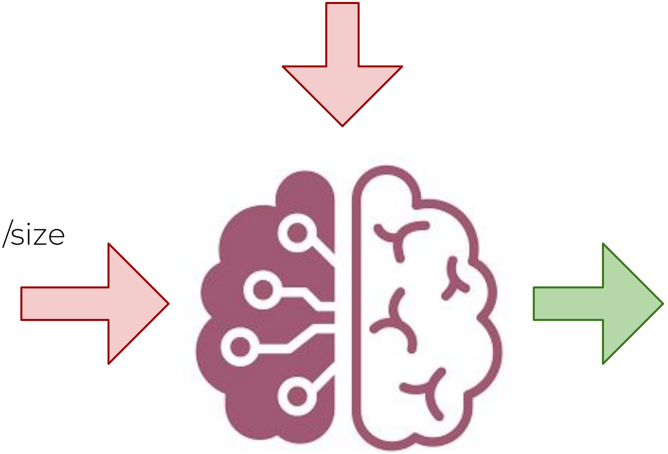
For instance, in the case of **YouTube**, it has been shown that only **5% of the videos are popular** [32].

For example, the **beginning portion of a video** or a popular highlight part in a video is typically streamed more often than the rest of the video [31].



LwTE Problem Formulation

{**Constraints** & **Objective function** }



Inputs:

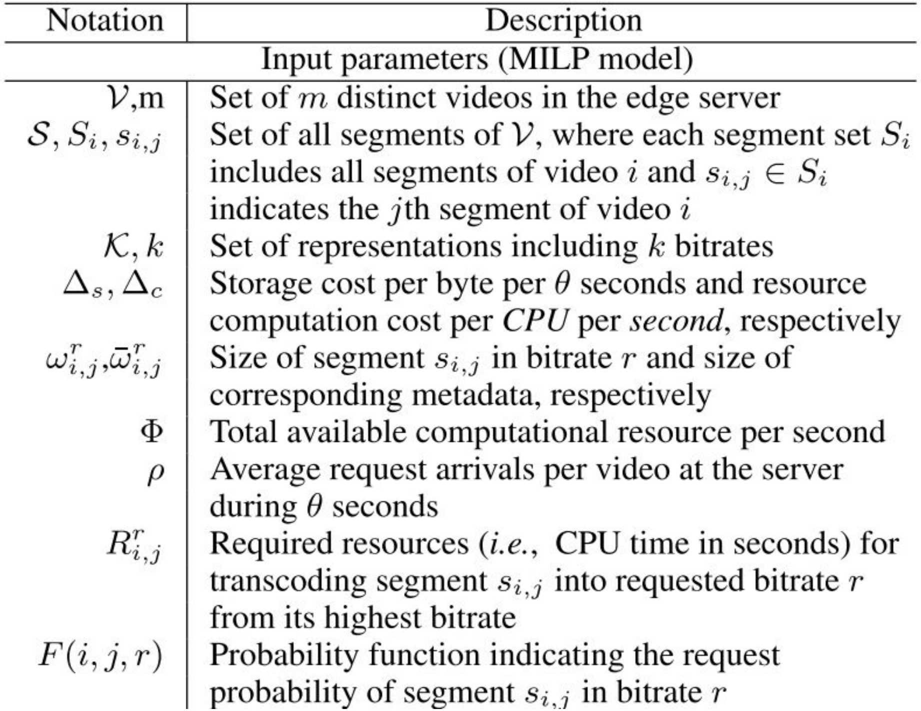
- Videos/Segments/metadata/size
- Representations
- Storage/computation cost
- Available resources
- Probability function

Outputs:

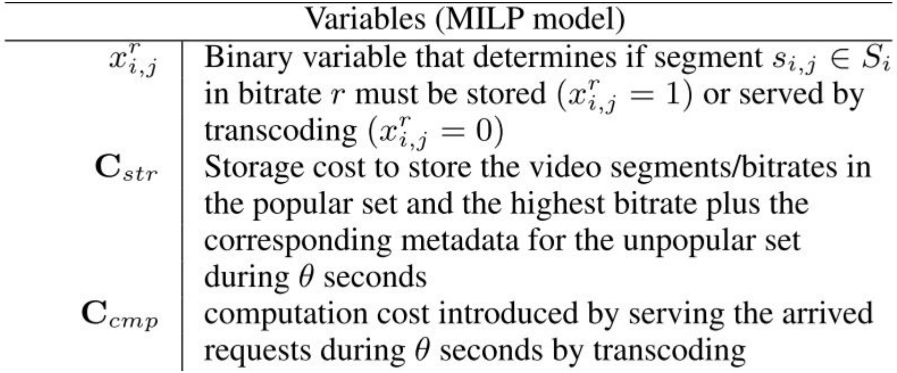
- Which segments should be stored/transcoded
- storage and computation costs

MILP Optimization model

Inputs and Outputs of LwTE



Notation	Description
	Input parameters (MILP model)
$\mathcal{V}$,m	Set of m distinct videos in the edge server
$\mathcal{S}, S_i, s_{i,j}$	Set of all segments of $\mathcal{V}$, where each segment set S_i includes all segments of video i and $s_{i,j} \in S_i$ indicates the jth segment of video i
$\mathcal{K}, k$	Set of representations including k bitrates
Δ_s, Δ_c	Storage cost per byte per θ seconds and resource computation cost per *CPU* per *second*, respectively
$\omega^r_{i,j}$,$\bar{\omega}^r_{i,j}$	Size of segment $s_{i,j}$ in bitrate r and size of corresponding metadata, respectively
Φ	Total available computational resource per second
ρ	Average request arrivals per video at the server during θ seconds
$R^r_{i,j}$	Required resources (*i.e.*, CPU time in seconds) for transcoding segment $s_{i,j}$ into requested bitrate r from its highest bitrate
$F(i,j,r)$	Probability function indicating the request probability of segment $s_{i,j}$ in bitrate r



	Variables (MILP model)
$x^r_{i,j}$	Binary variable that determines if segment $s_{i,j} \in S_i$ in bitrate r must be stored ($x^r_{i,j} = 1$) or served by transcoding ($x^r_{i,j} = 0$)
$\mathbf{C}_{str}$	Storage cost to store the video segments/bitrates in the popular set and the highest bitrate plus the corresponding metadata for the unpopular set during θ seconds
$\mathbf{C}_{cmp}$	computation cost introduced by serving the arrived requests during θ seconds by transcoding

Selecting Policy Constraint

For each segment/bitrate, we should decide the policy of servering. In other words, we should determine whether segment j of video i in bitrate r must be stored(=1) or served by transcoding (=0):

$$x_{i,j}^{r} = 1, \quad \forall i = \{1, \ldots, m\},\ j = \{1, \ldots, |S_i|\},\ r = 1$$

Note: we should fetch the highest bitrate to serve requests for popular and unpopular (for transcoding) segments;

The Storage Cost Constraint

The storage cost is a function of the **storage duration** and **volume**. The storage capacity is consumed by storing the video segments/bitrates for the popular set and the **highest bitrate plus the corresponding metadata** for the unpopular set

$$\underbrace{\Delta_s \times}_{\substack{\text{The storage cost per}\\ \text{byte per } \theta \text{ seconds}}} \sum_{i=1}^{m}\sum_{j=1}^{|S_i|}\sum_{r=1}^{|\mathcal{K}|} \underbrace{(x_{i,j}^{r} \times \omega_{i,j}^{r})}_{\substack{\text{The size of the bitrate}\\ \text{that needs to be stored}}} + \underbrace{(1 - x_{i,j}^{r}) \times \bar{\omega}_{i,j}^{r}}_{\text{the required storage for metadata}} \leq \mathbf{C}_{str},$$

The Computation Cost Constraint

Resource computation cost per CPU per second,

The request probability of segment si,j in bitrate r based on the given popularity pattern

$$\rho \times m \times \Delta_c \times \sum_{i=1}^{m} \sum_{j=1}^{|S_i|} \sum_{r=1}^{|\mathcal{K}|} (1 - x_{i,j}^{r}) \times F(i,j,r) \times R_{i,j}^{r} \leq \mathbf{C}_{cmp}$$

Average request arrivals per video in θ seconds

The required resources (i.e., CPU time in seconds) for transcoding the segment si,j into requested bitrate r from its highest bitrate

Available Computation Resource

The simulation duration in seconds

$$\rho \times m \times \sum_{i=1}^{m} \sum_{j=1}^{|S_i|} \sum_{r=1}^{|\mathcal{K}|} (1 - x_{i,j}^{r}) \times F(i, j, r) \times R_{i,j}^{r} \leq \Phi \times \theta,$$

the total available computation resource per second

The MILP Model

Objective Function

$$\left\{ Minimize\ \mathbf{C}_{str} + \mathbf{C}_{cmp} \right\}$$

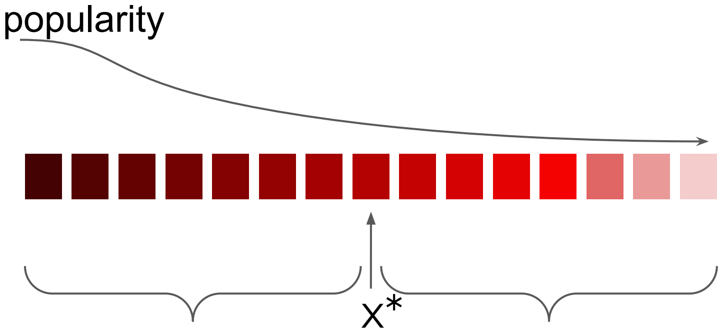
Constraints

$$\begin{cases} x_{i,j}^{r} = 1, \quad \forall i = \{1, \ldots, m\},\ j = \{1, \ldots, |S_i|\},\ r = 1 \\ \Delta_s \times \sum_{i=1}^{m} \sum_{j=1}^{|S_i|} \sum_{r=1}^{|\mathcal{K}|} (x_{i,j}^{r} \times \omega_{i,j}^{r}) + (1 - x_{i,j}^{r}) \times \bar{\omega}_{i,j}^{r} \leq \mathbf{C}_{str}, \\ \rho \times m \times \Delta_c \times \sum_{i=1}^{m} \sum_{j=1}^{|S_i|} \sum_{r=1}^{|\mathcal{K}|} (1 - x_{i,j}^{r}) \times F(i,j,r) \times R_{i,j}^{r} \leq \mathbf{C}_{cmp} \\ \rho \times m \times \sum_{i=1}^{m} \sum_{j=1}^{|S_i|} \sum_{r=1}^{|\mathcal{K}|} (1 - x_{i,j}^{r}) \times F(i,j,r) \times R_{i,j}^{r} \leq \Phi \times \theta, \end{cases}$$

NP-complete problem

Heuristic Algorithm

high time complexity of the proposed MILP



should be stored

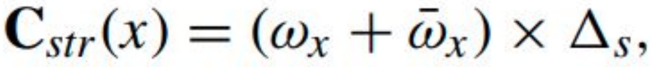
Should be transcoded

stores x* video segments/bitrates

stored in the highest bitrate plus corresponding metadata

Cost Function for the Heuristic Algorithm

The storage cost can be formulated as follows:



$$\mathbf{C}_{str}(x) = (\omega_x + \bar{\omega}_x) \times \Delta_s,$$

Where ω_x and $\bar{\omega}_x$ are the cumulative size of the video segments/bitrates that are stored up to point x and |X | − x segments/bitrates that are stored at the highest bitrate plus corresponding metadata, respectively

Cost Function for the Heuristic Algorithm …

Moreover, we can formulate the computation cost in a similar way

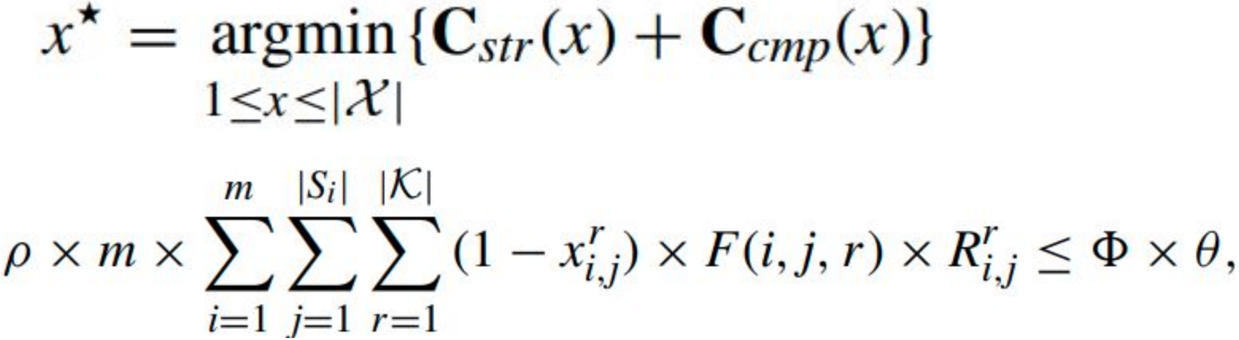
$$\mathbf{C}_{cmp}(x) = (\bar{P}_{|\mathcal{X}|} - \bar{P}_x) \times \rho \times m \times \Delta_c,$$

where $\bar{P}x$ specifies the cumulative required resources (i.e., CPU time in seconds) for transcoding up to point x; thus, by setting $\bar{P}0 = 0$, we have

$$\bar{P}_x = \bar{P}_{x-1} + (P(x) \times R_x),$$

where P(x) and Rx are **the request probability function** x and the required resources (i.e., CPU time in seconds) for transcoding segment/bitrate x from its highest bitrate, respectively

To consider the computational resource limitation at the edge, we should limit the x in the following equation. Thus, the boundary point x* for the given ρ can be obtained as follows

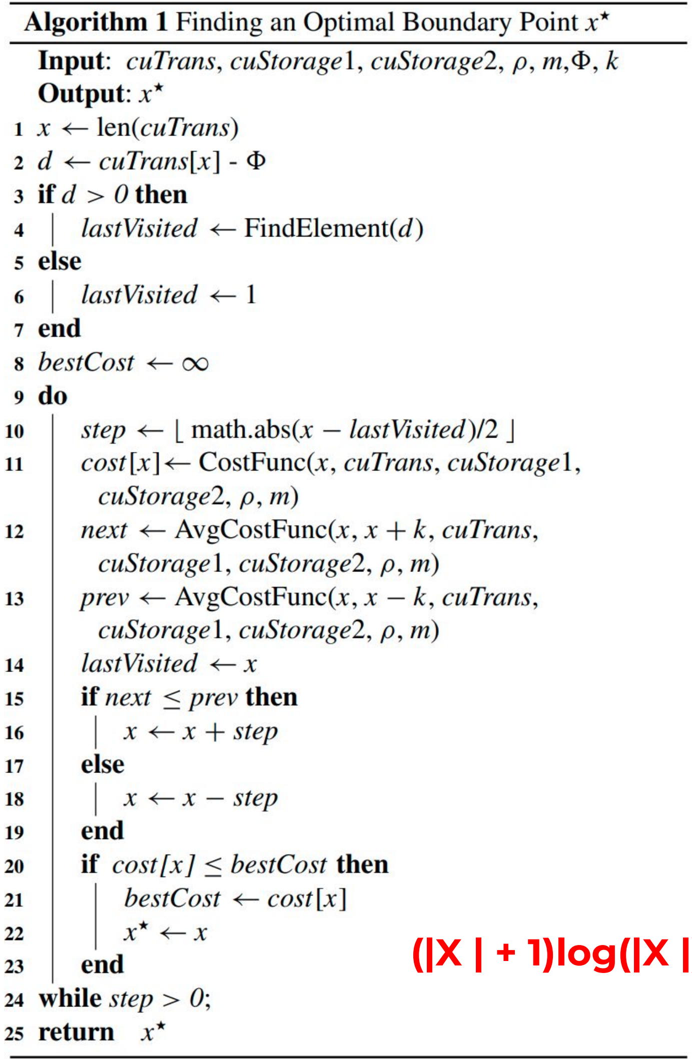


$$x^{\star} = \operatorname*{argmin}_{1 \le x \le |\mathcal{X}|} \{\mathbf{C}_{str}(x) + \mathbf{C}_{cmp}(x)\}$$

$$\rho \times m \times \sum_{i=1}^{m} \sum_{j=1}^{|S_i|} \sum_{r=1}^{|\mathcal{K}|} (1 - x_{i,j}^{r}) \times F(i,j,r) \times R_{i,j}^{r} \le \Phi \times \theta,$$

The value of x*can be achieved **by differentiating** the total cost function (Cstr(x) + Ccmp(x)) with respect to x; however, we first need to estimate the probability function P.

We can estimate the probability function P. However, **to avoid the time complexity of the non-linear function** estimation process and its potential error, here we propose a simple heuristic approach based on the binary search algorithm to find x* in a limited number of iterations.



Algorithm 1 Finding an Optimal Boundary Point $x^\star$

```
Input: $cuTrans$, $cuStorage1$, $cuStorage2$, $\rho$, $m$, $\Phi$, $k$
Output: $x^\star$
$x \leftarrow$ len($cuTrans$)
$d \leftarrow cuTrans[x]$ - $\Phi$
if $d > 0$ then
    $lastVisited \leftarrow$ FindElement($d$)
else
    $lastVisited \leftarrow 1$
end
$bestCost \leftarrow \infty$
do
    $step \leftarrow \lfloor$ math.abs($x - lastVisited$)/2 $\rfloor$
    $cost[x] \leftarrow$ CostFunc($x$, $cuTrans$, $cuStorage1$,
      $cuStorage2$, $\rho$, $m$)
    $next \leftarrow$ AvgCostFunc($x$, $x + k$, $cuTrans$,
      $cuStorage1$, $cuStorage2$, $\rho$, $m$)
    $prev \leftarrow$ AvgCostFunc($x$, $x - k$, $cuTrans$,
      $cuStorage1$, $cuStorage2$, $\rho$, $m$)
    $lastVisited \leftarrow x$
    if $next \leq prev$ then
        $x \leftarrow x + step$
    else
        $x \leftarrow x - step$
    end
    if $cost[x] \leq bestCost$ then
        $bestCost \leftarrow cost[x]$
        $x^\star \leftarrow x$
    end
while $step > 0$;
return $x^\star$
```

(|X | + 1)log(|X |)

Performance Evaluation- Setup & Assumptions

- The performance is evaluated for a fixed time interval,
- Resource costs (i.e., storage and computation) remain fixed during the considered time interval, and
- Arriving requests are distributed uniformly in the time interval
- We transcode the full set of representations (i.e., we adopt the bitrate configuration of HEVC/H.265 30 fps from [35]) of the BasketballDrive [13] sequence using HEVC HM-16.20 [13] with four-second segment length

Performance Evaluation- Setup & Assumptions ...

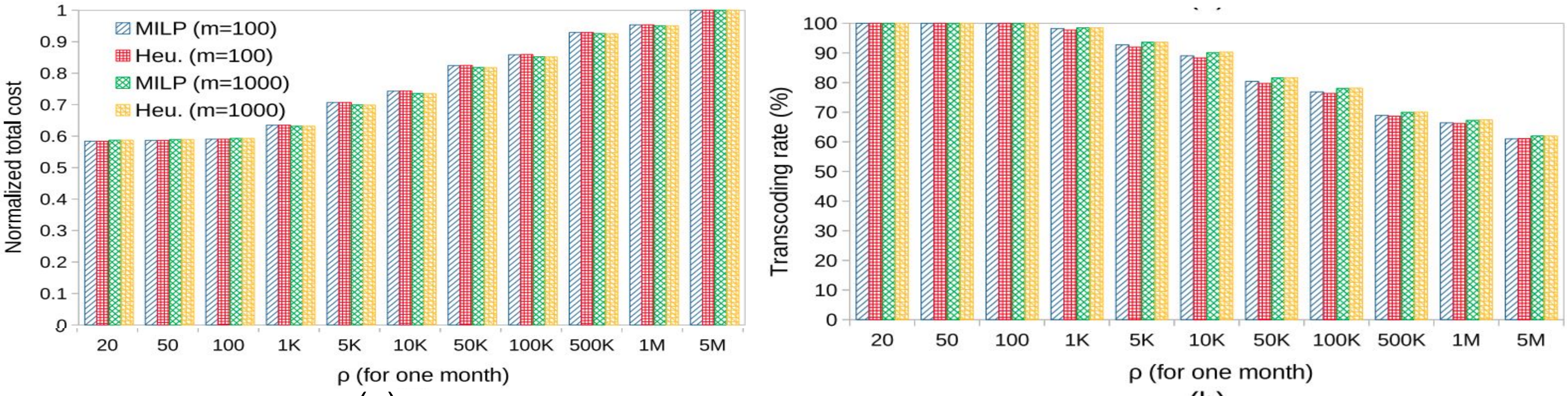
- Requests are generated independently and follow a Poisson process. For the access probability, we use a Zipf-like distribution
- The storage and computation costs are set to 0.024$ per GB per month and 0.029$ per CPU per hour, respectively [1]
- Transcoding is performed on Docker containers with one 3.4 GHz CPU and 2 GB memory

[1] https://calculator.aws/, last access: April 25, 2021.

Scenario I

execution times → MILP model form 1.2 seconds and 19.2 seconds, proposed heuristic algorithm → is less than one millisecond.

we investigate the performance of the proposed MILP model and the heuristic algorithm using LwTEM1 in terms of the **transcoding rate and total cost** (i.e., storage and computation costs)



(a)

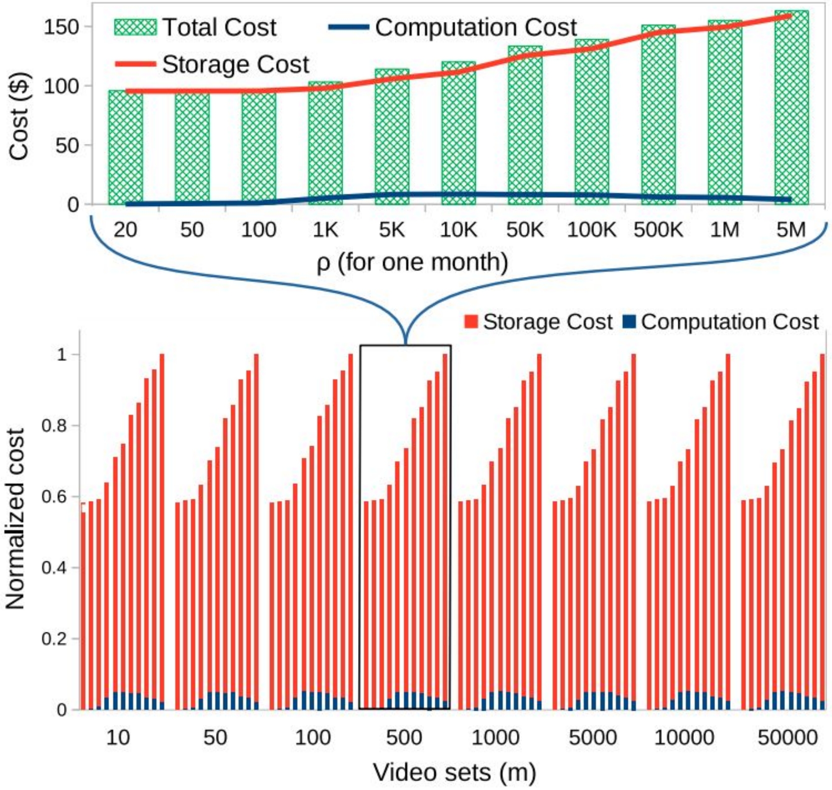
(b)

Comparison of the proposed MILP model and the heuristic algorithm in terms of (a) normalized total cost and (b) transcoding rate for two video sets, including 100 and 1000 videos, and various average request arrivals per video (ρ) for one month.

Scenario II

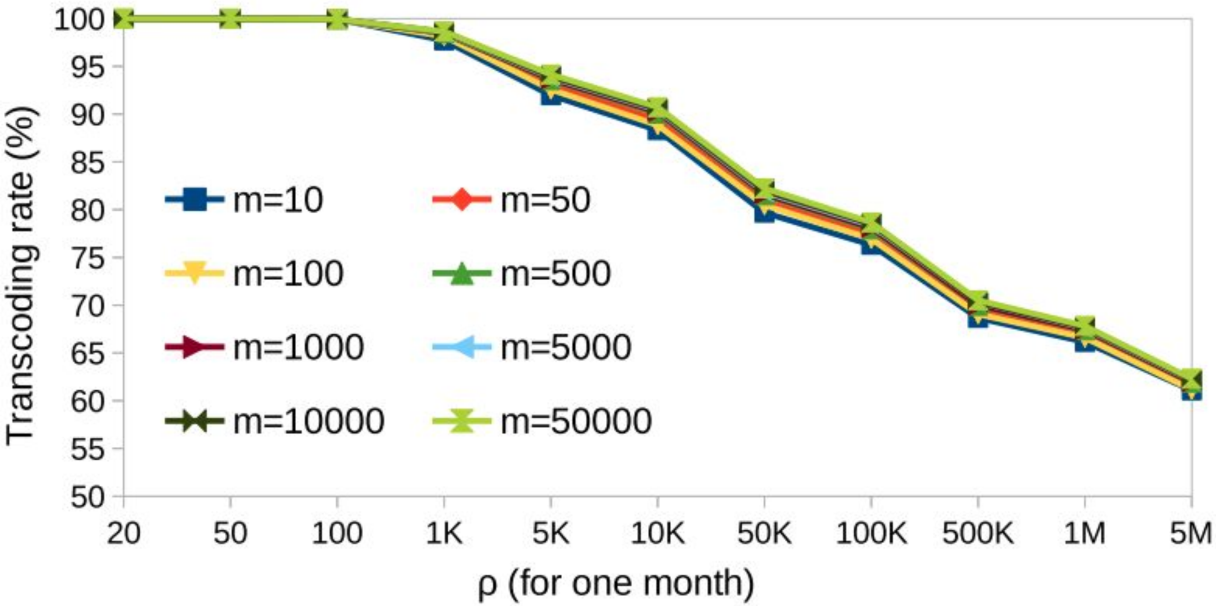
Investigates the proposed heuristic algorithm's performance employing **LwTE-M1 for eight video sets** and **various average request arrivals per video (ρ)** for one month in various aspects.

Since the ρ values are fixed for all video sets, **we can conclude** that determining the transcoding rate mainly **depends on ρ,** and the number of video segments/ bitrates has a negligible impact on it.



Normalized values for storage and computation costs of the proposed heuristic algorithm employing LwTE-M1 for eight video sets and various average request arrivals per video (ρ) for one month.

Scenario II ...

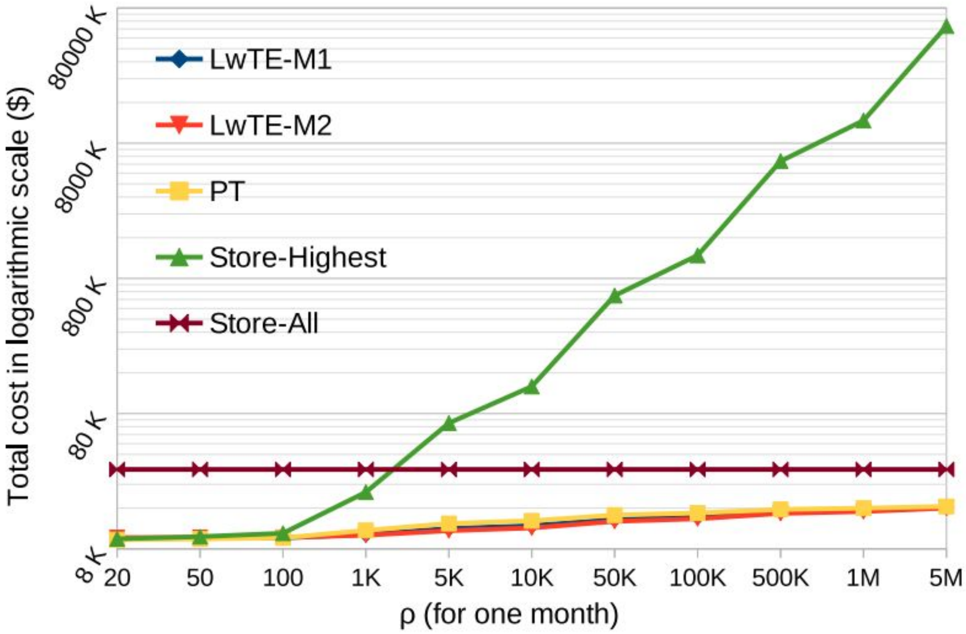


Transcoding rate of the proposed heuristic algorithm using LwTE-M1 for eight video sets and various average request arrivals per video (ρ) for one month.

Scenario III

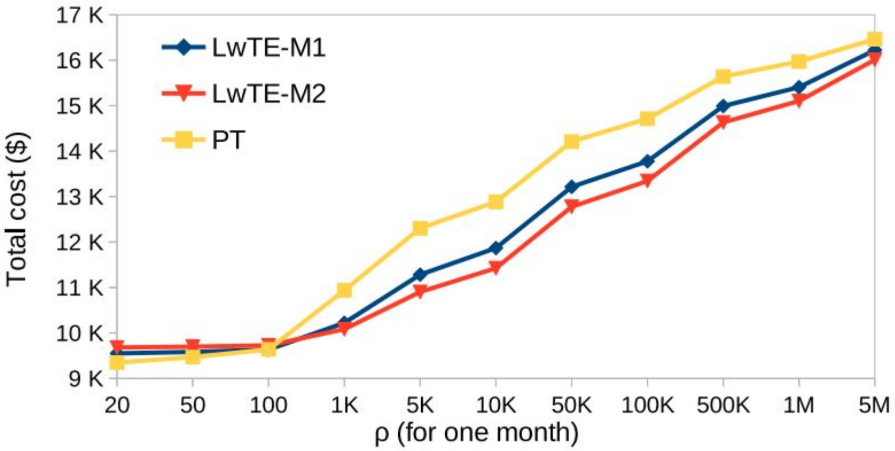
Compares the LwTE approach in different modes with some state-of-the-art and industrial approaches.

(i) Store-All
(ii) Store-Highest
(iii) Conventional Partial-transcoding (PT)
(iv) LwTE-M1 and LwTE-M2

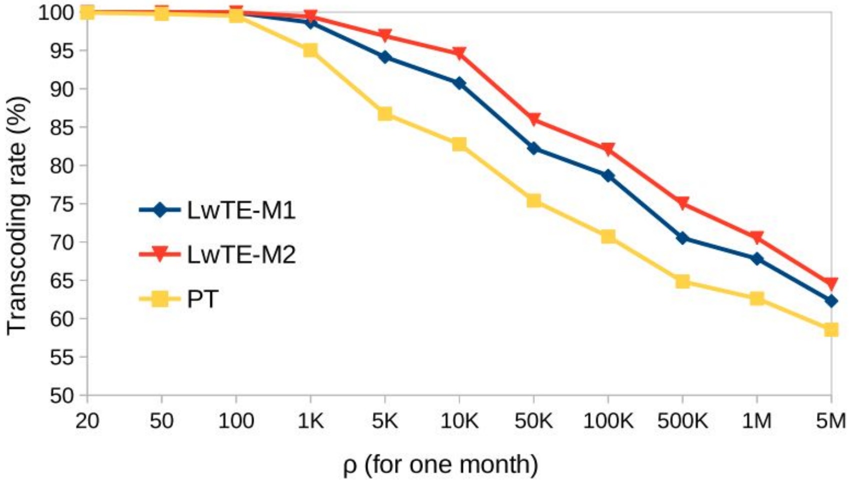


Comparison of the proposed LwTE approach's performance in different modes with some state-of-the-art methods in terms of the total cost.

Scenario III …



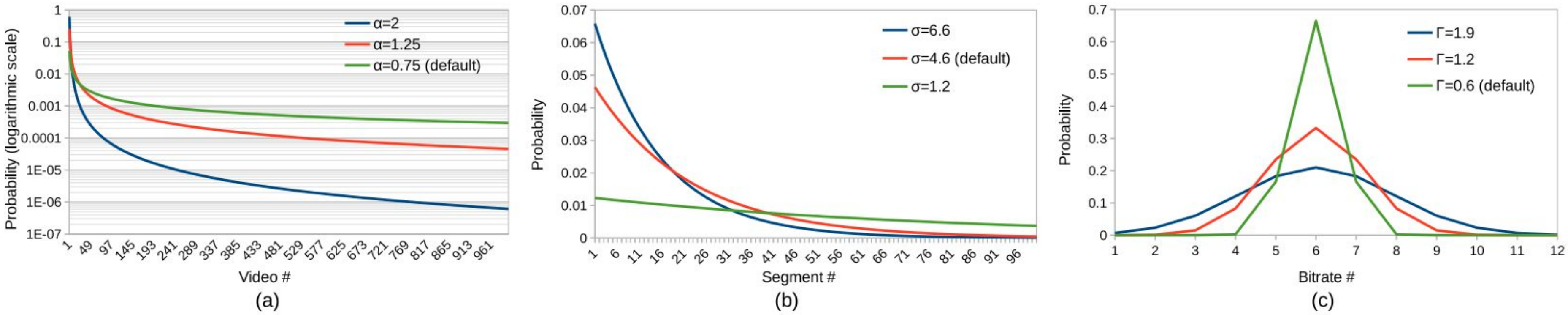
Comparison of the proposed LwTE approach in different modes with PT in terms of the total cost.



Comparison of the proposed LwTE approach in different modes with PT in terms of the transcoding rate.

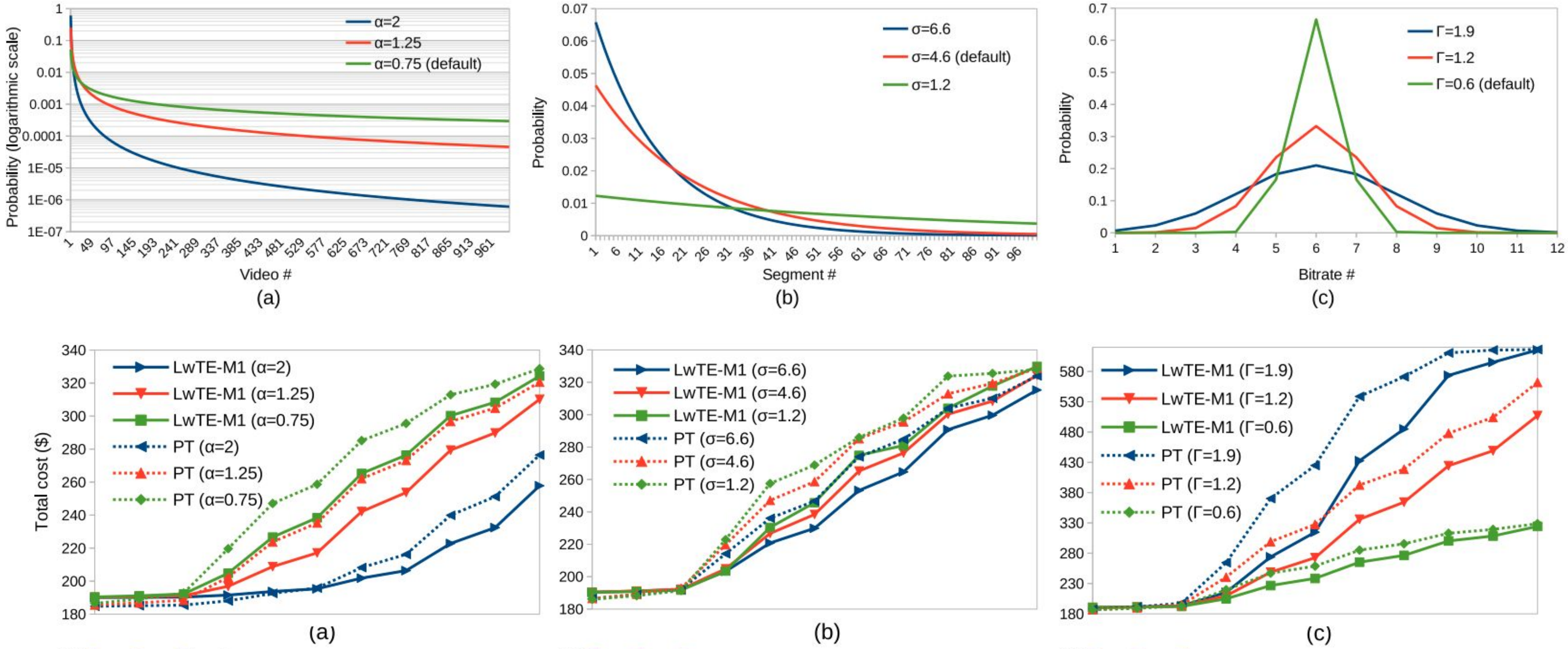
Scenario IV

Various probability distributions

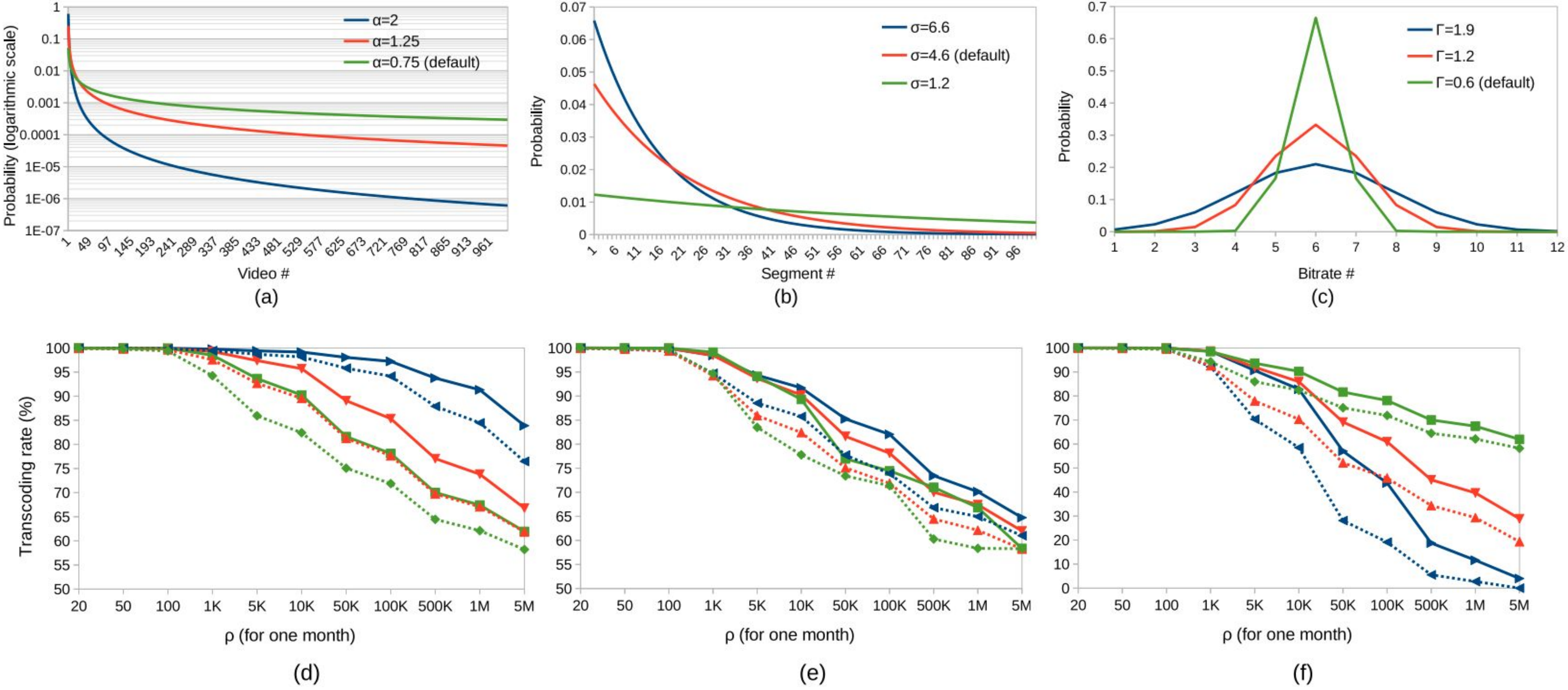


(a) (b) (c)

Probability distributions for (a) 1000 videos, (b) 100 segments within a video, and (c) bitrates in a representation set.



Impact of Zipf distribution parameters on total cost with (a) various **α** values, (b) various **σ** values, and (c) various 0 values, and on transcoding rate with (d) various **α** values, (e) various **σ** values, and (f) various 0 values.



Impact of Zipf distribution parameters on total cost with (a) various **α** values, (b) various **σ** values, and (c) various 0 values, and on transcoding rate with (d) various **α** values, (e) various **σ** values, and (f) various 0 values.

Conclusion and Future Work

- A novel cost-effective video transcoding approach called LwTE
- Store the optimal search results/decisions in the encoding process as metadata
- For unpopular segments/bitrates, LwTE stores only the highest bitrate plus corresponding metadata
- Transcoding processes at least 80% faster than the conventional transcoding method.
- Experimental results indicate up to 70% and 12% cost saving compared to the conventional Store-All and PT approaches, respectively.

Conclusion and Future Work …

- More realistic assumptions and constraints,
    - e.g., by considering resource limitations at the edge (i.e., storage, bandwidth, and computation)
- Multiple time-slots with variable duration into both optimization models and heuristic algorithms
- Implement LwTE in a large scale scenario

Thank you for your attention