INTRODUCTION TO VIDEO STREAMING

Vaneet Aggarwal

Parts of work joint with:

A. Elgabli, A. Alabassi, A. Ghosh, M. Felemban (Purdue University)

S. Sen, S. Hao (AT&T Labs-Research)

F. Qian (IU)

PURDUE UNIVERSITY

MOBILE VIDEO STREAMING IS GETTING SO POPULAR

- **By 2021: 82% of Internet Traffic (3 ZB = $3*10^9$ TB) will be video** [CISCO]
- **Q4 2016: mobile video surpassed desktop videos in terms of online viewing time** [IAB]

PURDUE UNIVERSITY

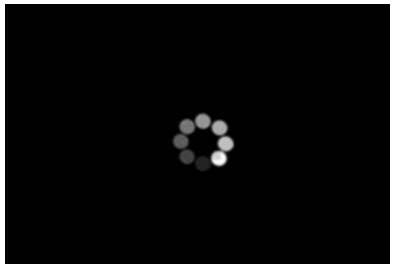
MOBILE VIDEO STREAMING IS GETTING SO POPULAR

- **By 2021: 82% of Internet Traffic (3 ZB = $3*10^9$ TB) will be video** [CISCO]
- **Q4 2016: mobile video surpassed desktop videos in terms of online viewing time** [IAB]

PURDUE UNIVERSITY

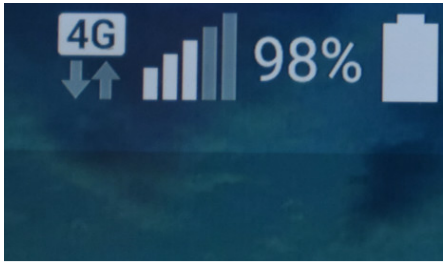
RESEARCH GOAL

Improve Quality of Experience (QoE) and Resource Efficiency for Video Streaming



Less Stall

Higher Quality



Less Energy/Data Usage

PURDUE UNIVERSITY

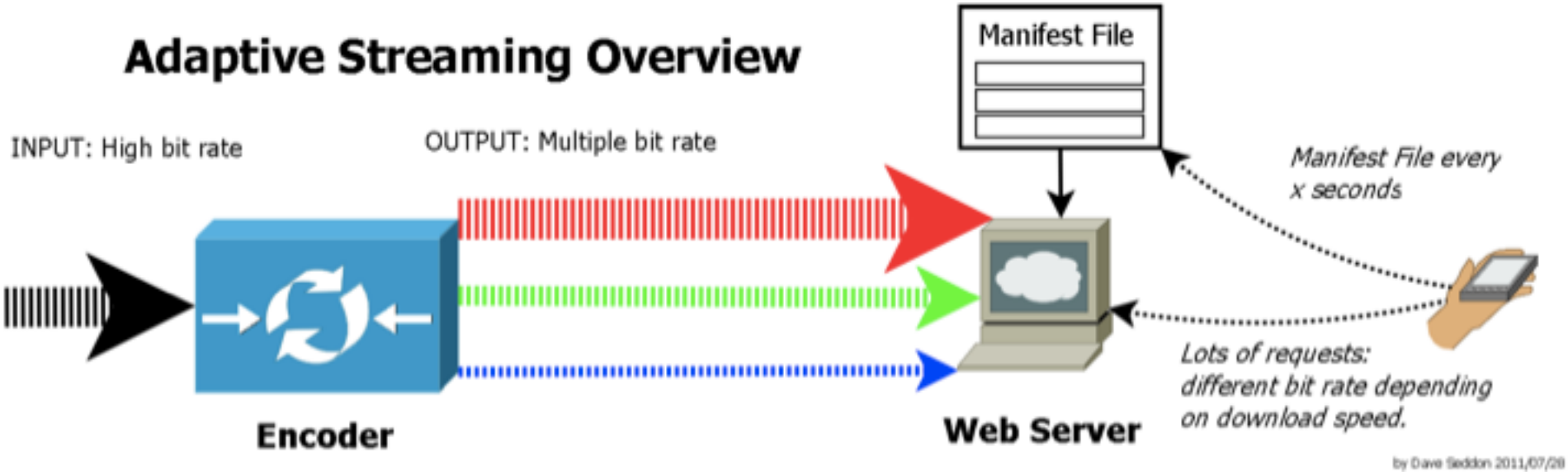
OUTLINE

- Video Streaming Algorithms
  - Single Path
  - Multiple Paths
  - Giant Client
- 360-degree Video Streaming
- Video Streaming over Cloud

OUTLINE

- Video Streaming Algorithms
  - Single Path
  - Multiple Paths
  - Giant Client
- 360-degree Video Streaming
- Video Streaming over Cloud

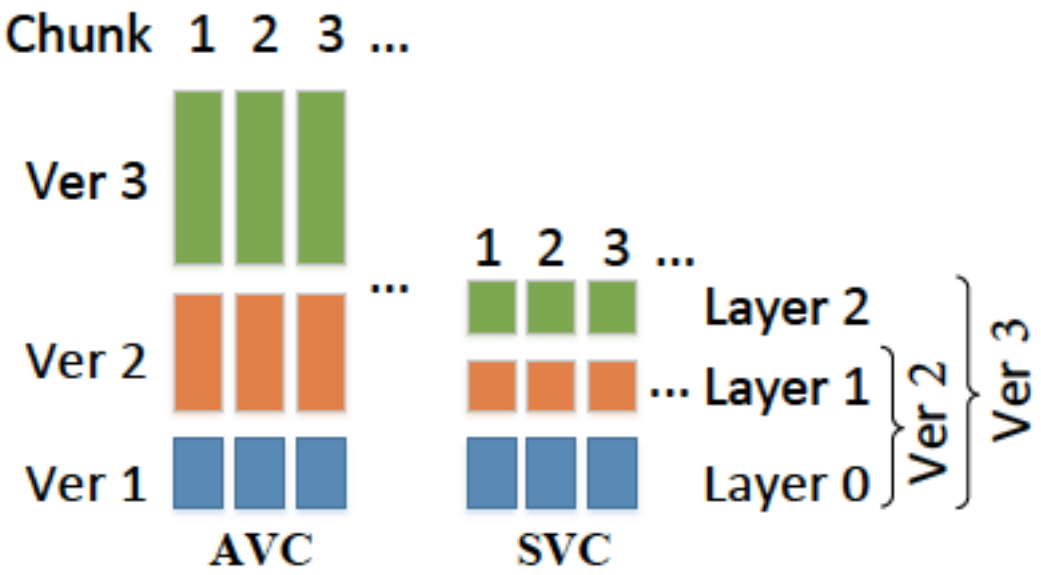
ADAPTIVE BIT-RATE VIDEO



- Each chunk is encoded into multiple bit-rates
- The user can fetch each chunk at one of these rates.

PURDUE UNIVERSITY

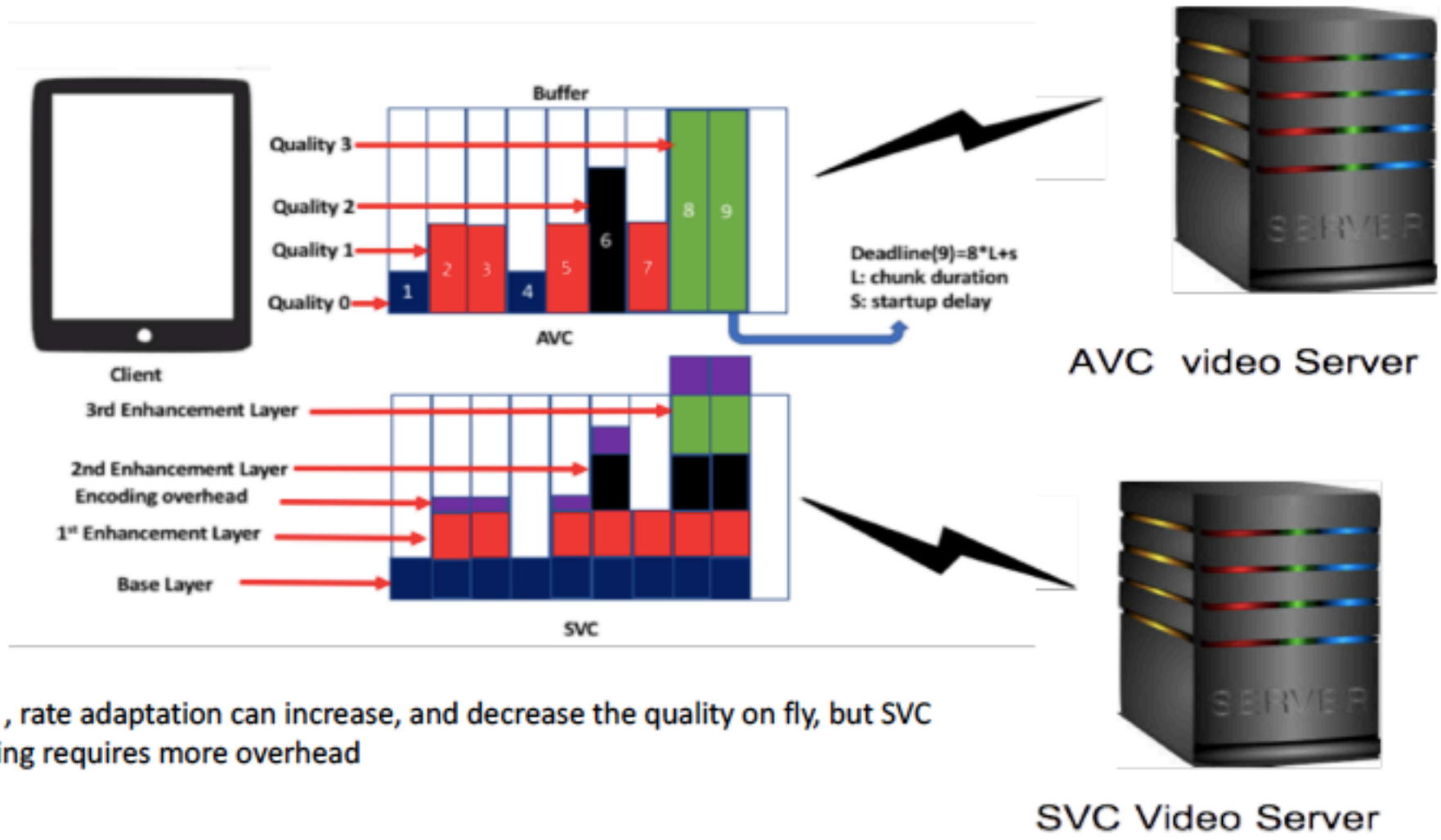
TWO TYPICAL FORMS OF ENCODING



- H.264/MPEG-4 AVC (Advanced Video Coding), Standardized 2003. One out of L chunks must be selected.

- SVC (Scalable Video Coding), Standardized in 2007 as a extension to H.264. Has one base layer and multiple enhancement layers.

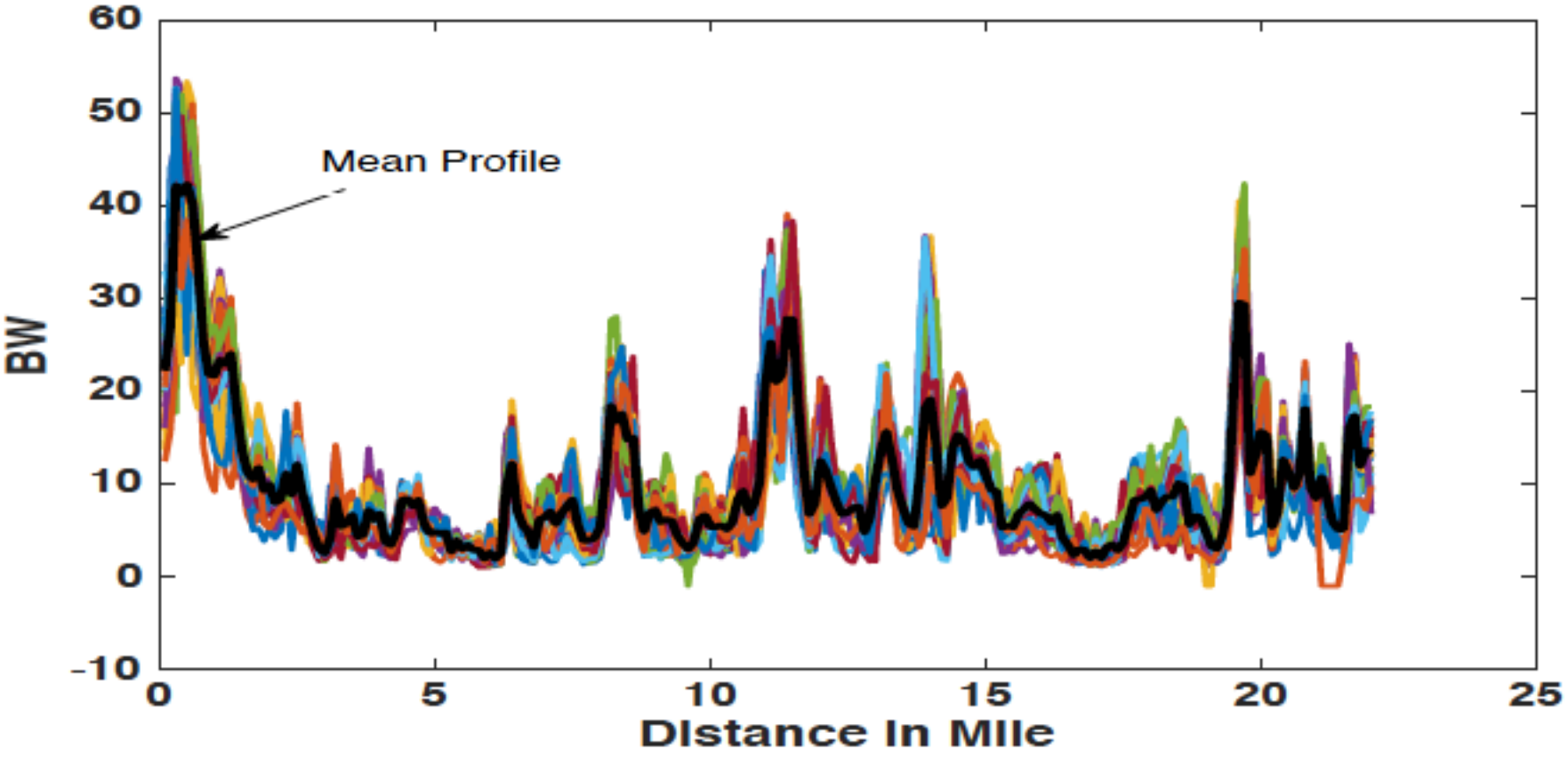
TWO TYPICAL FORMS OF ENCODING



In SVC , rate adaptation can increase, and decrease the quality on fly, but SVC encoding requires more overhead

BENEFIT OF PREDICTION

- LTE data collection over 30 different weekdays

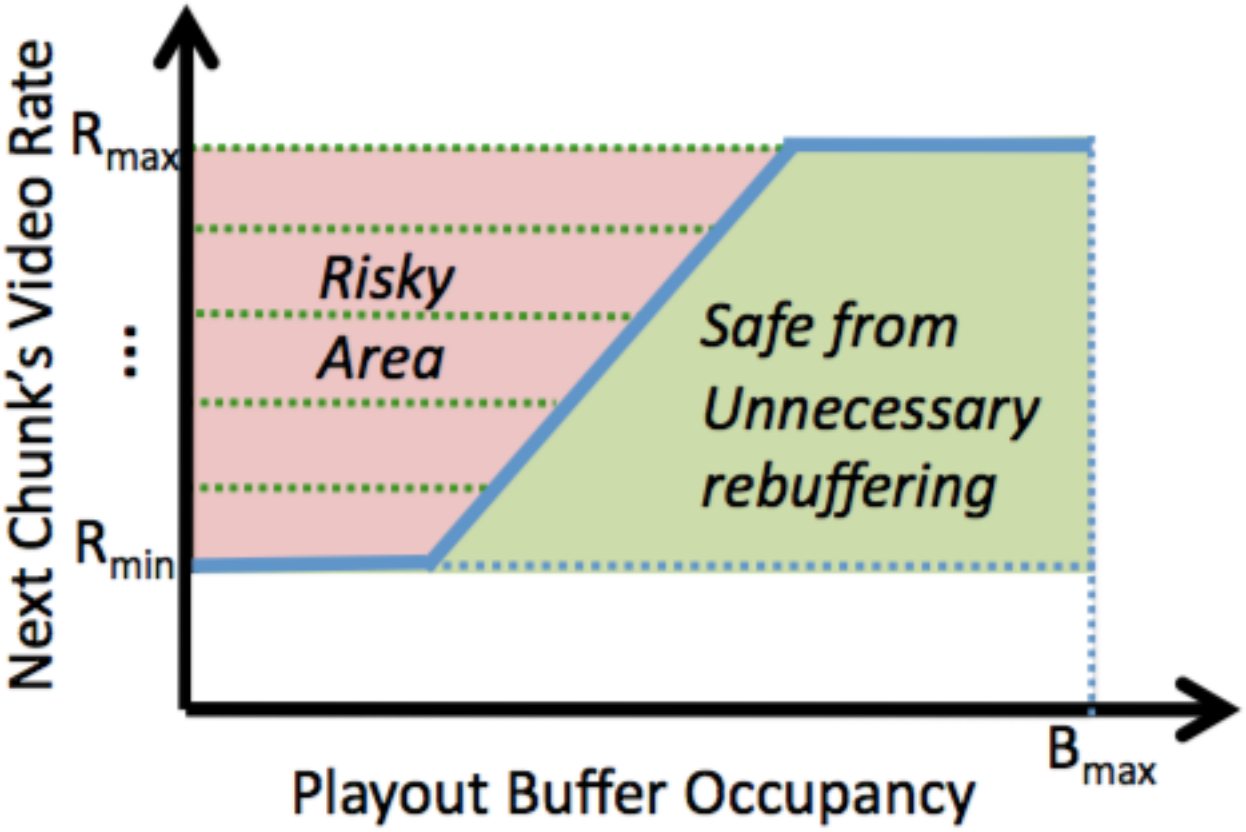


VIDEO STREAMING PROBLEM FORMULATION

- What quality should each of the video chunk be fetched at?
- Objective: Maximize the Quality of Experience of the end users.
- Some key metrics:
  - Stall Duration (Rebuffering)
  - Average quality
  - Quality Variations

BUFFER BASED ALGORITHM (BBA)

- Decide the quality of next chunk based on the buffer occupancy
- Higher video rate with higher buffer occupancy
- Gap: Buffer occupancy accounts for buffer and thus implicitly bandwidth, but not explicitly. Slow in learning bandwidth change.



Huang et al., Sigcomm 2014

PURDUE UNIVERSITY

MODEL PREDICTIVE CONTROL (MPC)

- An optimization approach for deciding the quality levels
- Weighted combination of different QoE metrics is chosen
  - Stall Duration
  - Average Quality
  - Quality variation between chunks
- The optimization problem has discrete constraints (quality levels are discrete)

$$\max_{R_1,\cdots,R_K,T_s} \quad QoE_1^K$$

$$s.t. \qquad t_{k+1} = t_k + \frac{d_k(R_k)}{C_k} + \Delta t_k,$$

$$C_k = \frac{1}{t_{k+1} - t_k - \Delta t_k} \int_{t_k}^{t_{k+1}-\Delta t_k} C_t \, dt,$$

$$B_{k+1} = \left(\left(B_k - \frac{d_k(R_k)}{C_k} \right)_+ + L - \Delta t_k \right)_+$$

$$B_1 = T_s, \quad B_k \in [0, B_{max}]$$

$$R_k \in \mathcal{R}, \quad \forall k = 1, \cdots, K.$$

Yin et al., Sigcomm 2015

MODEL PREDICTIVE CONTROL (MPC)

- Weighted combination of different QoE metrics is chosen
  - Stall Duration
  - Average Quality
  - Quality variation between chunks
- The optimization problem has discrete constraints (quality levels are discrete)
- A lookup table for nearby points is constructed, and used in real-time.

$$\max_{R_1,\cdots,R_K,T_s} \quad QoE_1^K$$

$$s.t. \qquad t_{k+1} = t_k + \frac{d_k(R_k)}{C_k} + \Delta t_k,$$

$$C_k = \frac{1}{t_{k+1} - t_k - \Delta t_k} \int_{t_k}^{t_{k+1}-\Delta t_k} C_t \, dt,$$

$$B_{k+1} = \left(\left(B_k - \frac{d_k(R_k)}{C_k} \right)_+ + L - \Delta t_k \right)_+$$

$$B_1 = T_s, \quad B_k \in [0, B_{max}]$$

$$R_k \in \mathcal{R}, \quad \forall k = 1, \cdots, K.$$

Yin et al., Sigcomm 2015

OUR PROPOSED FORMULATION

- As a first step, we assume SVC encoding, and will generalize it later.
- We also assume a skip-based formulation, where chunks if not received by deadline are skipped. The results will be extended to the stall-based version too.
- Notations
  - C chunks
  - N+1 layers – 0, 1, …, N
  - Length of chunk, L
  - Startup delay, s
  - Size of Layer n, Y_n
  - $z_n(i,j)$ – Amount of layer n of chunk i fetched in time j
  - Quality level at which chunk at layer n can be fetched $Z_n \epsilon \{0, Y_n\}$
  - Bandwidth limitation at every time j, B(j)
  - Buffer capacity in time, B_m

PURDUE UNIVERSITY

OUR PROPOSED FORMULATION

- Constraints:

Layer Size Constraints	$\sum_{j=1}^{(i-1)L+s} z_n(i,j) = Z_{n,i}, \quad \forall i, n$
Decoder Constraints	$Z_{n,i} \leq \frac{Y_n}{Y_{n-1}} Z_{n-1,i}, \quad \forall i, n > 0$
Bandwidth Constraints	$\sum_{n=0}^{N} \sum_{i=1}^{C} z_n(i,j) \leq B(j) \quad \forall j = 1, \cdots, (C-1)L + s,$
Buffer Constraints	$\sum_{i,(i-1)L+s>t} \mathbf{I}\left(\sum_{j=1}^{t} \left(\sum_{n=0}^{N} z_n(i,j) \right) > 0 \right) L \leq B_m \ \forall t$
Positivity, deadline, and feasibility constraints	$z_n(i,j) \geq 0 \ \forall i = 1, \cdots, C$ $z_n(i,j) = 0 \ \forall i, (i-1)L + s > j$ $Z_{n,i} \in \mathcal{Z}_n \triangleq \{0, Y_n\} \quad \forall i, n$

OUR PROPOSED FORMULATION

- What could be a good metric (objective)?
- The increase in QoE with chunk quality has diminishing returns
- Minimizing Skips = Maximizing number of chunks for which base layer is fetched.

OUR PROPOSED FORMULATION

- What could be a good metric (objective)?
- The increase in QoE with chunk quality has diminishing returns
- Minimizing Skips = Maximizing number of chunks for which base layer is fetched.
- So, one approach seems like (γ<<1)

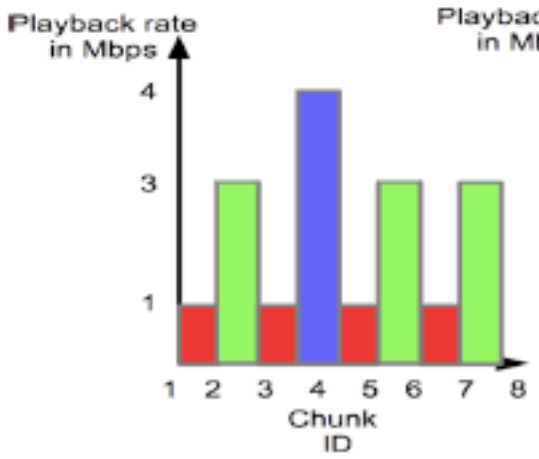
  (Number of chunks in BL) + γ (Number of chunks in EL1) + γ^2 (Number of chunks in EL2)+…

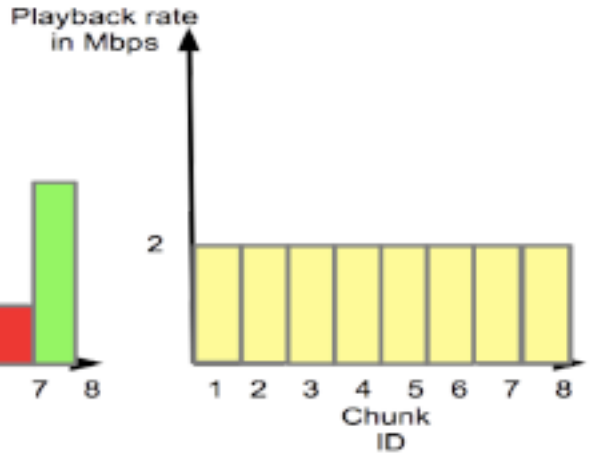
- This prefers going higher layers when there is no possibility where more lower layers can be obtained, thus matching the diminishing returns objective.

WHICH STRATEGY IS BETTER?

(Number of chunks in BL) + γ (Number of chunks in EL1) + γ^2 (Number of chunks in EL2)+…

Fetching Policy 1





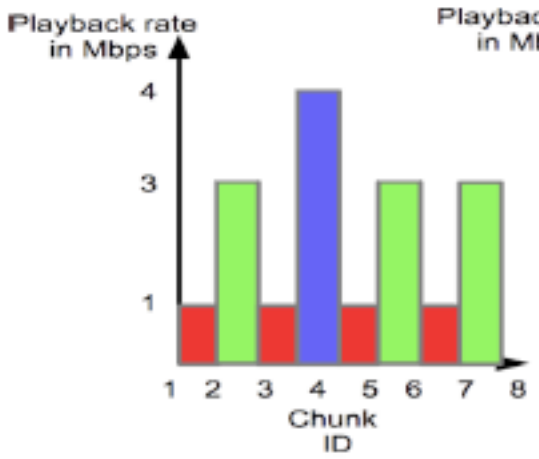
Fetching Policy 2

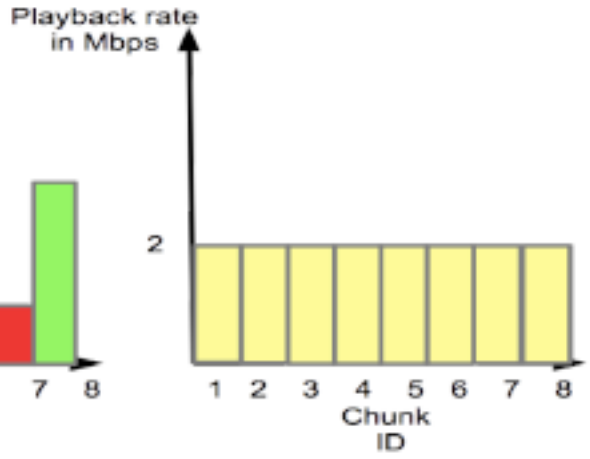
PURDUE UNIVERSITY

WHICH STRATEGY IS BETTER?

(Number of chunks in BL) + γ (Number of chunks in EL1) + γ^2 (Number of chunks in EL2)+…

Fetching Policy 1





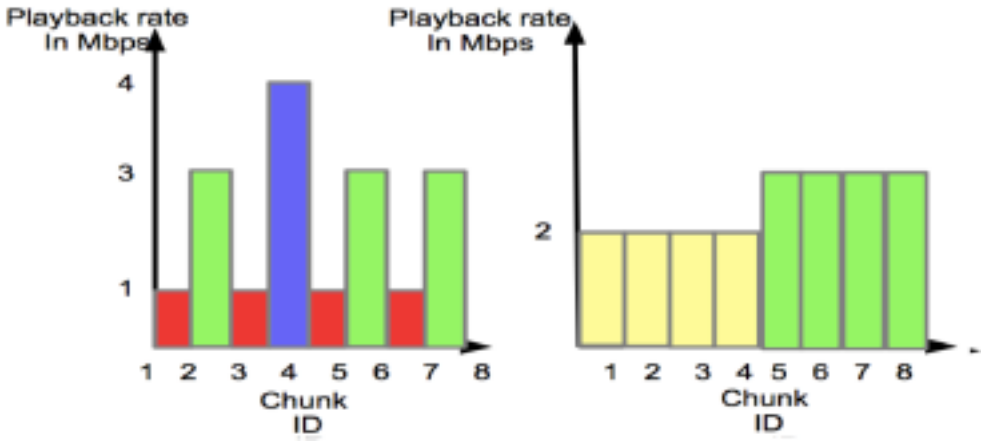
Fetching Policy 2

- Policy 2

OTHER CONSIDERATION?

(Number of chunks in BL) + γ (Number of chunks in EL1) + γ^2 (Number of chunks in EL2)+…

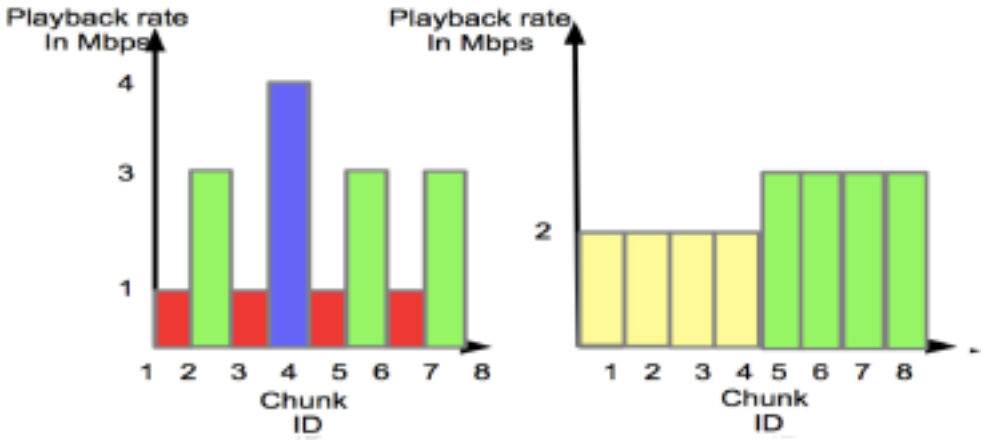
- The objective does not account for quality variations.



OTHER CONSIDERATION?

(Number of chunks in BL) + γ (Number of chunks in EL1) + γ^2 (Number of chunks in EL2)+…

- The objective does not account for quality variations.



- Approach: Favor later chunks.

$$\textbf{Maximize:} \left(\gamma^0 \sum_{i=1}^{C} \beta^i Z_{0,i} + \gamma^1 \sum_{i=1}^{C} \beta^i Z_{1,i} + \ldots\ldots + \gamma^n \sum_{i=1}^{C} \beta^i Z_{n,i} \right)$$

CONDITION ON γ

$$\gamma^a r_a > \sum_{k=a+1}^{N} \gamma^k r_k \sum_{i=1}^{C} \beta^i \quad \text{for } a = 0, \cdots, N-1.$$

- For every layer a, all higher layers achieve less objective than fetching a chunk at layer a
- This allows for layer prioritization.
- Can independent decision at layers be chosen with such constraints?

CONDITION ON γ

$$\gamma^a r_a > \sum_{k=a+1}^{N} \gamma^k r_k \sum_{i=1}^{C} \beta^i \quad \text{for } a = 0, \cdots, N-1.$$

- For every layer a, all higher layers achieve less objective than fetching a chunk at layer a
- This allows for layer prioritization.
- Can independent decision at layers be chosen with such constraints? No
  - This is because when the chunk is fetched impacts the available bandwidth for future
  - The chunks at all times have to satisfy buffer capacity constraints

IS THE PROBLEM NP-HARD?

- The problem has discrete constraints, since each layer of each chunk is wither fetched fully or not at all
- Also, there is a non-convex buffer constraint

Layer Size Constraints	$\sum_{j=1}^{(i-1)L+s} z_n(i,j) = Z_{n,i}, \quad \forall i,n$
Decoder Constraints	$Z_{n,i} \leq \frac{Y_n}{Y_{n-1}} Z_{n-1,i}, \quad \forall i, n > 0$
Bandwidth Constraints	$\sum_{n=0}^{N}\sum_{i=1}^{C} z_n(i,j) \leq B(j) \quad \forall j = 1, \cdots, (C-1)L+s,$
Buffer Constraints	$\sum_{i,(i-1)L+s>t} \mathbf{I}\left(\sum_{j=1}^{t}\left(\sum_{n=0}^{N} z_n(i,j)\right) > 0\right) L \leq B_m \; \forall t$
Positivity, deadline, and feasibility constraints	$z_n(i,j) \geq 0 \; \forall i = 1, \cdots, C$ $z_n(i,j) = 0 \; \forall i, (i-1)L+s > j$ $Z_{n,i} \in \mathcal{Z}_n \triangleq \{0, Y_n\} \quad \forall i, n$

ARE DISCRETE OPTIMIZATION PROBLEMS HARD?

- Many problems are.
  - Knapsack Problem
  - Cutting Stock Problem
  - Bin Packing Problem
  - Travelling Salesman Problem
  - -- and many more.

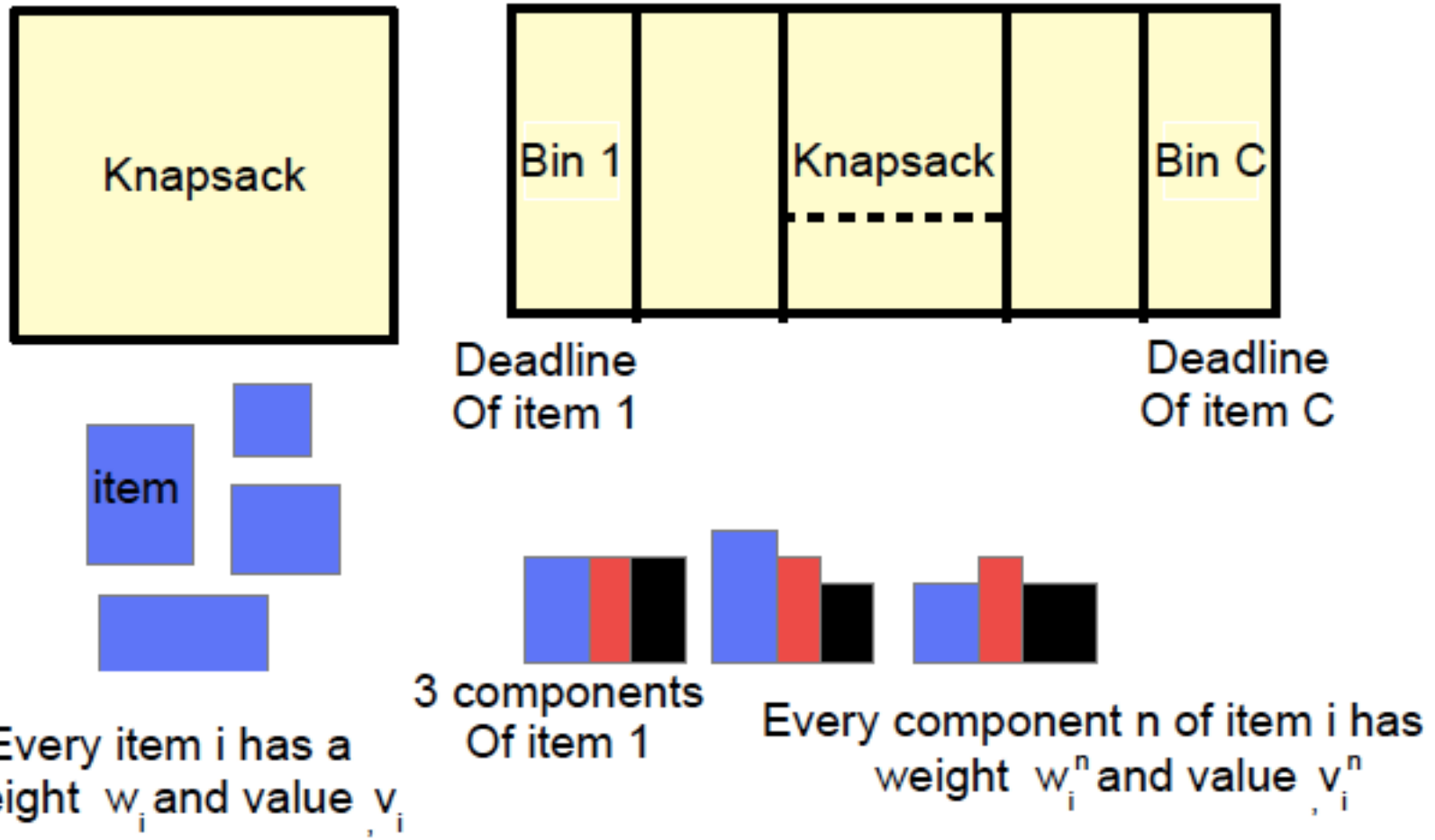
  - All these are discrete optimizations, and NP hard.

ARE DISCRETE OPTIMIZATION PROBLEMS HARD?

- In contrast, some problems are not
  - Shortest Path Trees
  - Flows and Circulations
  - Spanning Trees
  - Matching Problem
  - Matroid Problem

RELATION TO KNAPSACK PROBLEM

- The problem can be seen as bin-extension to the Knapsack problem



BIN-EXTENDED KNAPSACK PROBLEM

$$\textbf{maximize: } \sum_{n=0}^{N-1} \sum_{i=1}^{C} \sum_{t=1}^{C} v_i^n x_{i,n,t}$$

s.t

$$\sum_{n=0}^{N-1} \sum_{i=1}^{C} \sum_{j=1}^{t} x_{i,n,j} w_i^n \leq \sum_{j=1}^{t} B_j, \forall t$$

$$\sum_{t=1}^{C} x_{i,n,t} \leq \sum_{t=1}^{C} x_{i,n-1,t}, \forall i, n$$

$$x_{i,n,t} > 0 \forall i, n, t$$

$$\sum_{j=i+1}^{C} \mathbf{I}(\sum_{t=1}^{i} \sum_{n=0}^{N-1} x_{j,n,t} > 0) \leq B_{max}, \forall i$$

$$x_{i,n,t} = 0, \forall i, n, t > i$$

$$\sum_{t=1}^{C} x_{i,n,t} \in \{0, 1\}, \forall i, n$$

PURDUE UNIVERSITY

RELATION TO KNAPSACK PROBLEM

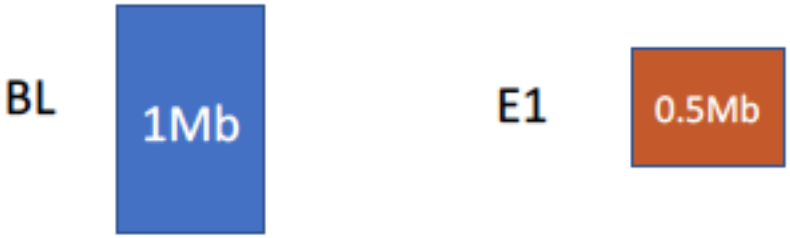
- Knapsack problem is NP hard, so is this extension.
- However, we consider a special case

1. $w_i^n = w^n \forall i$
2. $v_i^n = v^n \forall i$
3. $C \sum_{k=a+1}^{N-1} v^k < v^a, \quad \text{for all } a = 0, \cdots, N-2$

- We will see that the problem is polynomial-time solvable in this special case (indeed, linear time solvable in N and C).

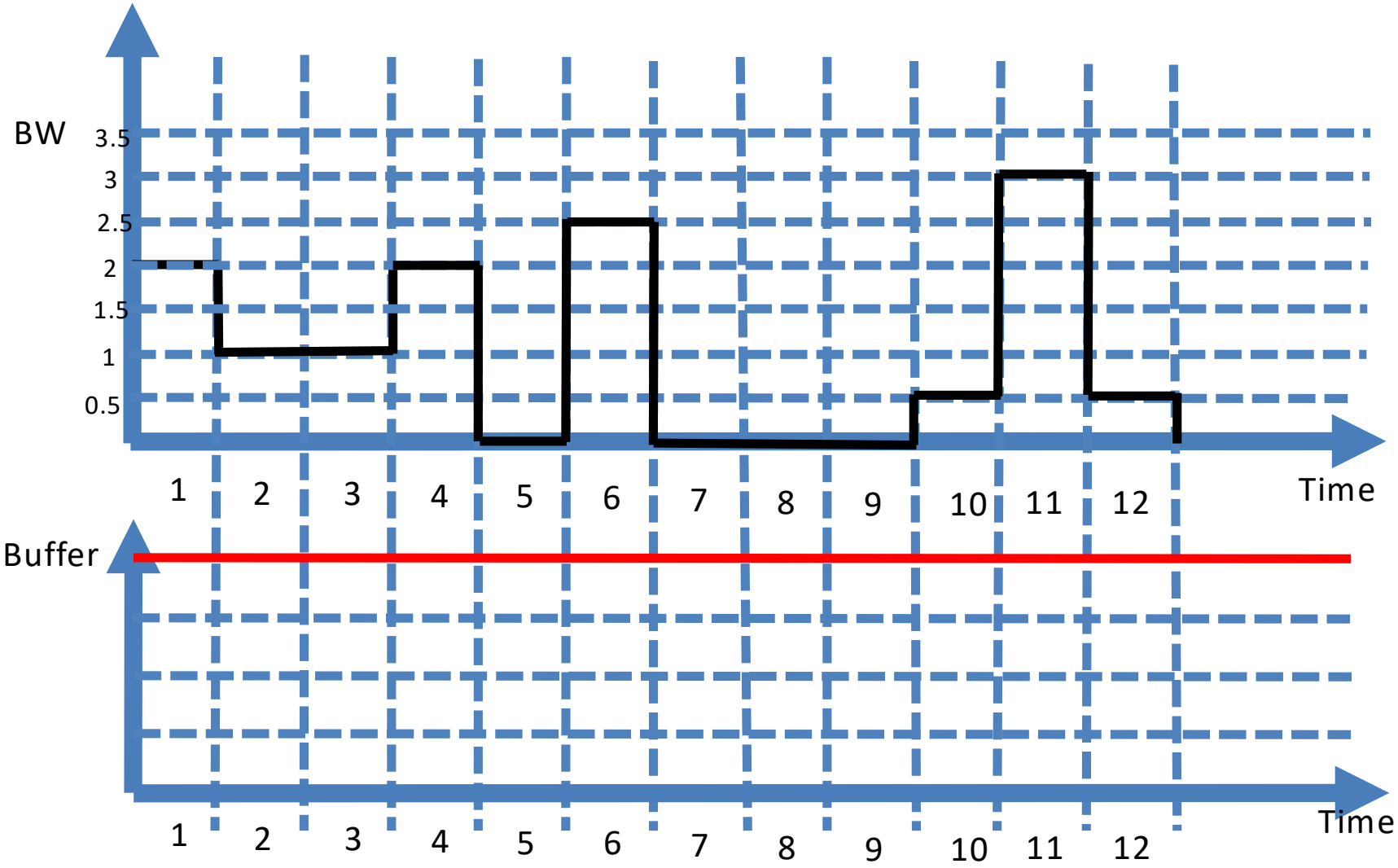
PROPOSED ALGORITHM

- Example Setup:
  - 10 chunks, 1 s each
  - Startup delay = 3s
  - Maximum Buffer size = 4s
  - 1 BL, 1 EL

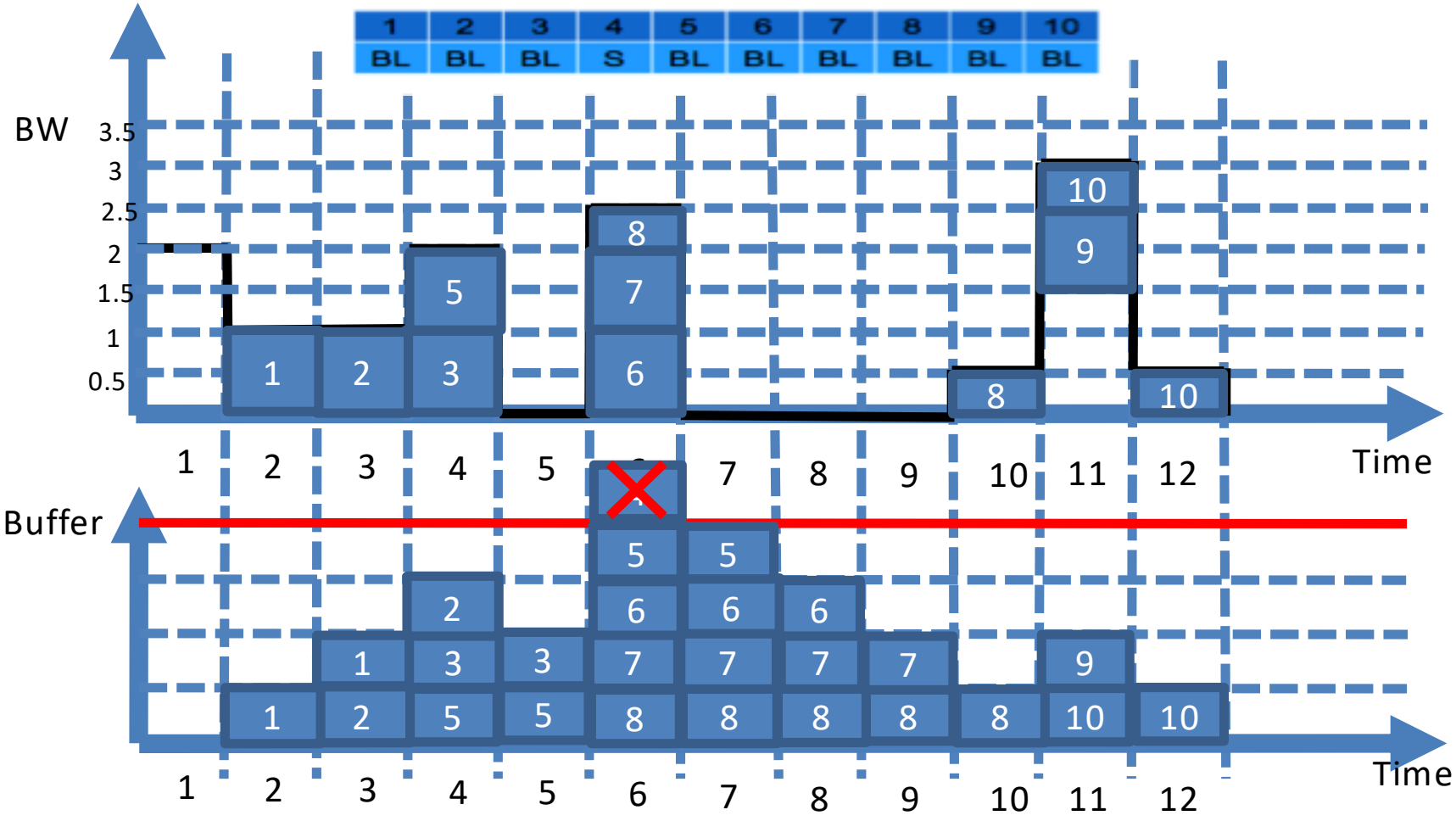


PURDUE UNIVERSITY

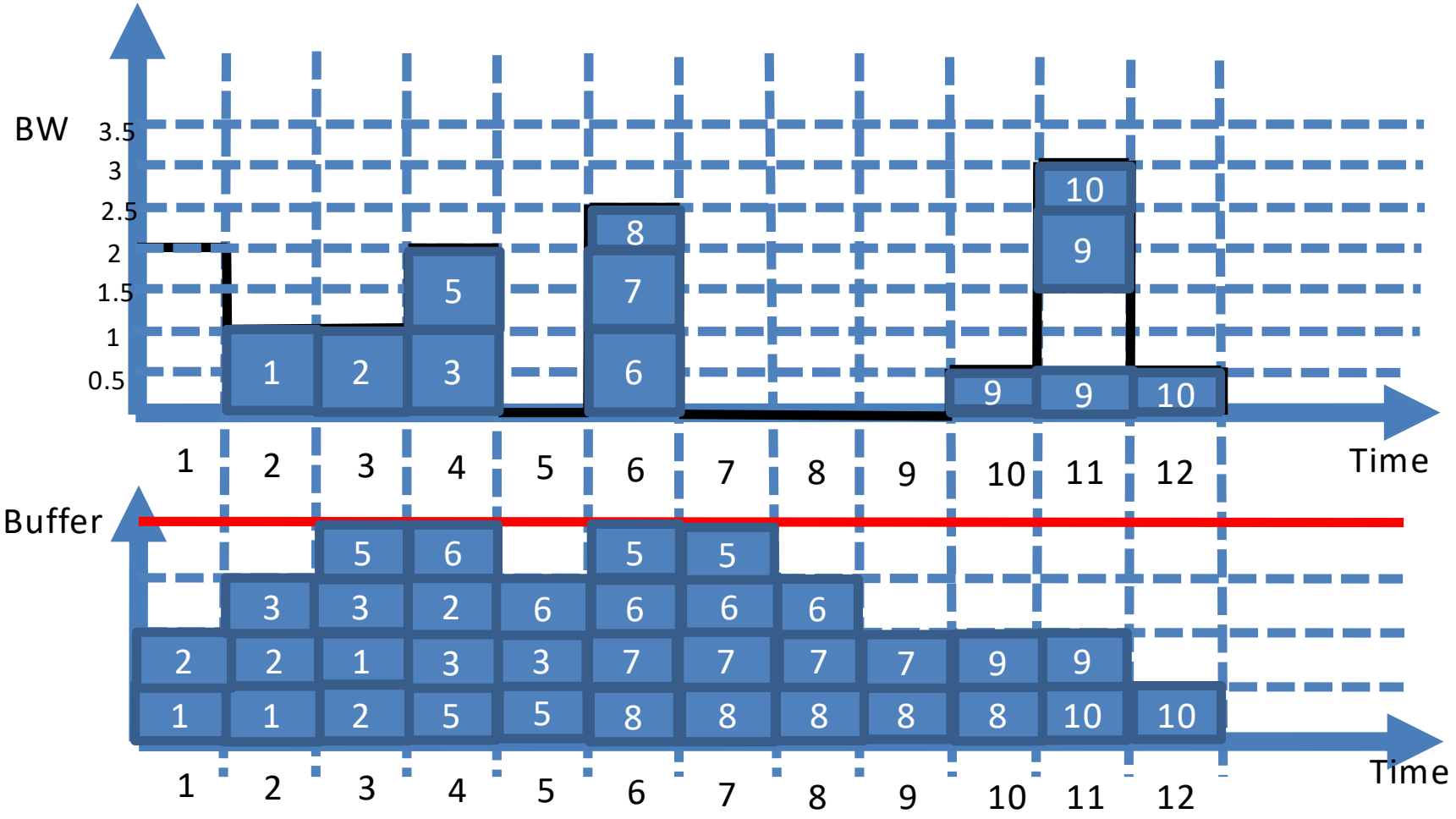
BANDWIDTH PROFILE AND BUFFER CAPACITY



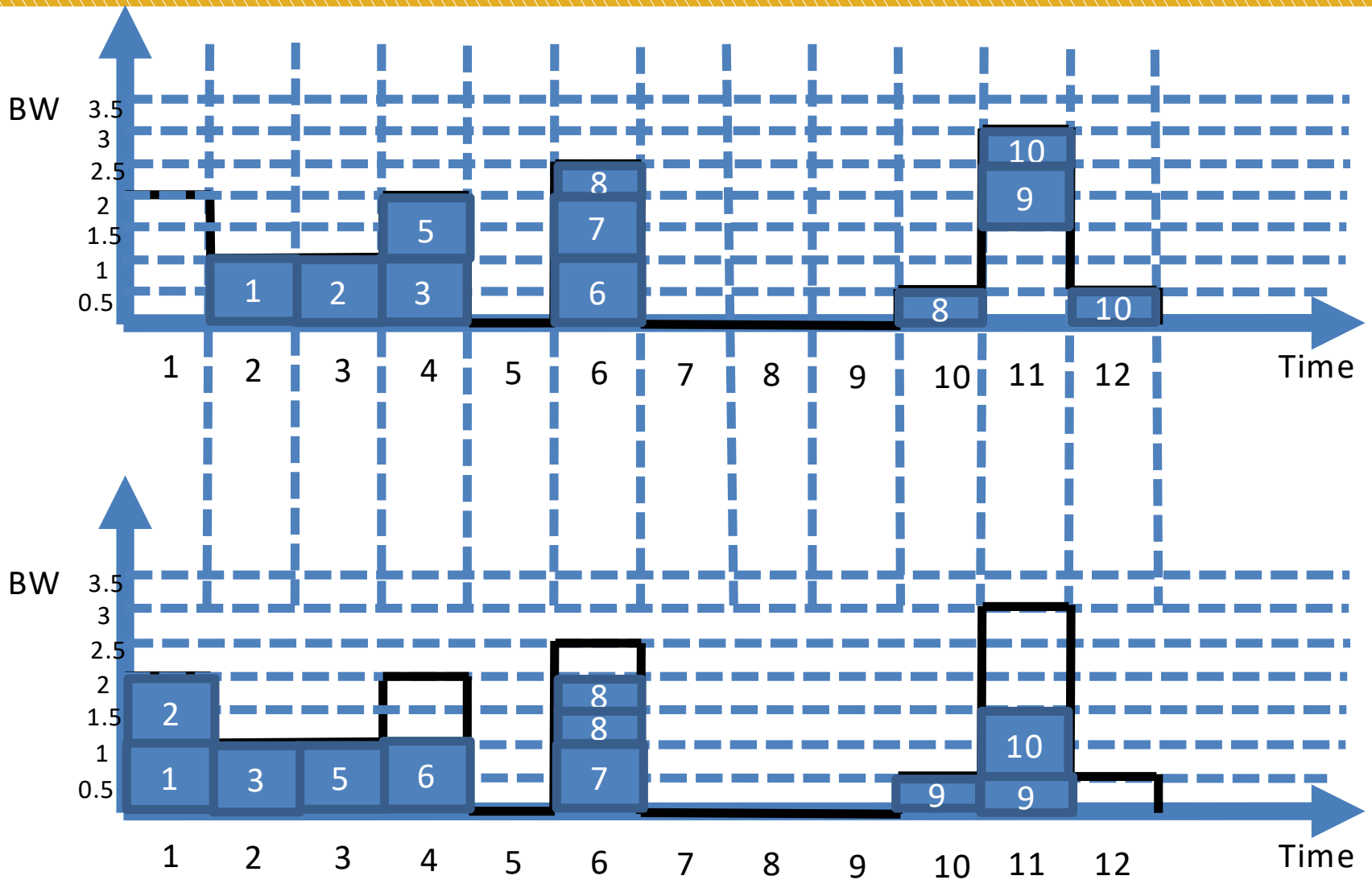
STEP 1: BACKWARD ALGORITHM FOR BL DECISIONS



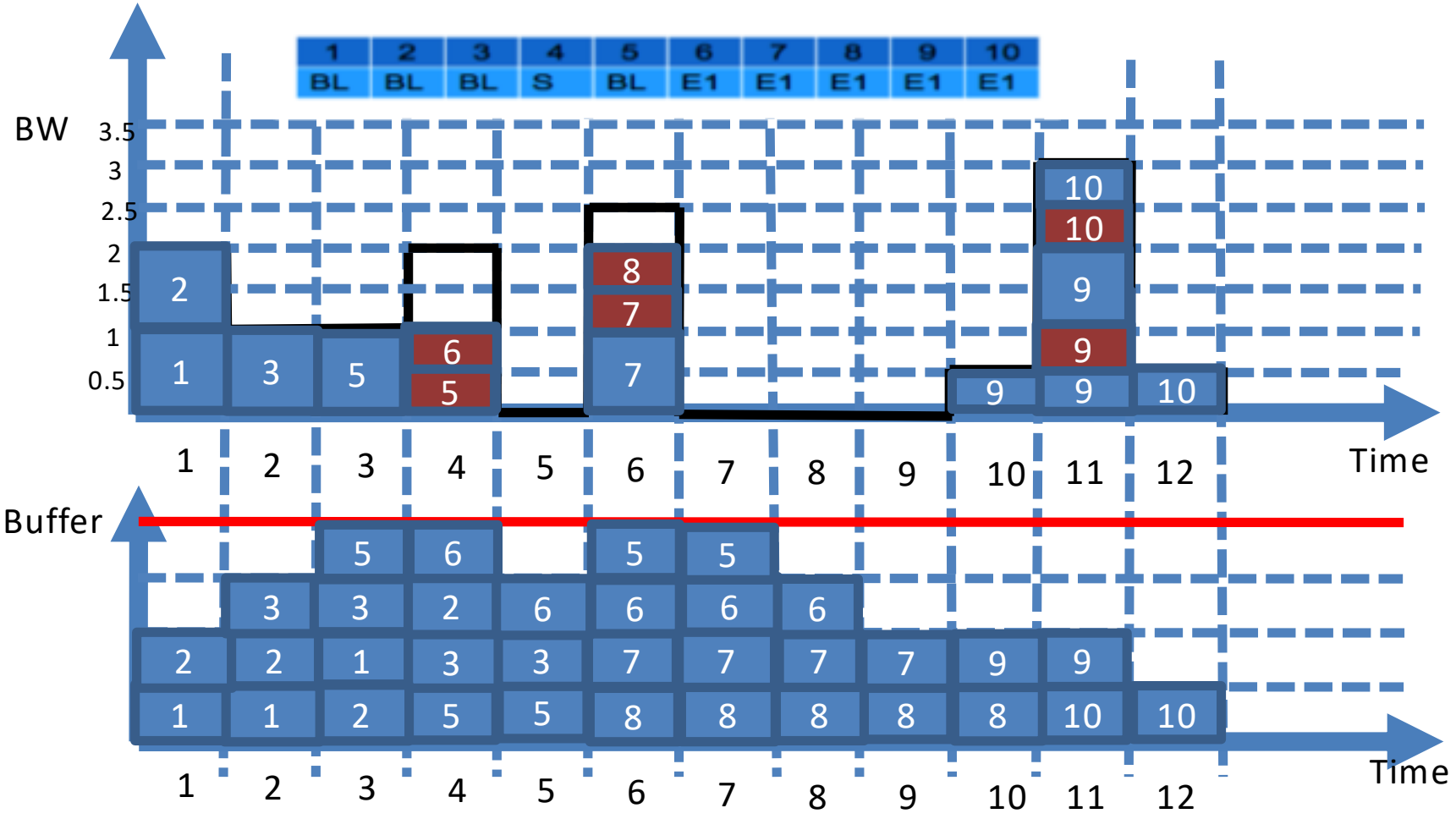
STEP 2: FORWARD ALGORITHM FOR BL DECISIONS



BEFORE AND AFTER STEP 2



STEP 3: BACKWARD ALGORITHM FOR EL1 DECISIONS



OPTIMALITY

- Step 1: Without loss of generality, can consider in-order fetching.
- Step 2: Fetching algorithm maximizes number of fetched chunks at BL.
  - There are skips only due to two reasons – bandwidth insufficient or buffer violations. In both cases, it can be shown that the skips cannot be decreased as compared to the proposed algorithm
- Step 3: Forward algorithm maximizes the available bandwidth for higher layers among all decisions with same decisions at lower layers
- Recursively using the results give the optimality of the proposed algorithm
- Complexity: O(NC) – linear in number of layers and number of chunks.

ONLINE ALGORITHM

- The predicted bandwidth may only be available for short time ahead
- Bandwidth prediction is inaccurate
- We Compute the scheduling decision only for the next W chunks ahead.
- Repeat after downloading every chunk to adjust to prediction errors and compute quality decisions for more chunks ahead.
- Our formulation works with any prediction technique. For evaluation, we consider harmonic mean and crowd-sourcing based predictions

NO-SKIP BASED STREAMING FORMULATION

The new objective function is:

$$\textbf{Maximize: } \sum_{n=1}^{N} \gamma^n \sum_{i=1}^{C} \beta^i Z_{n,i} - \lambda d(C)$$

Subject to all skip version constraints plus the following constraint

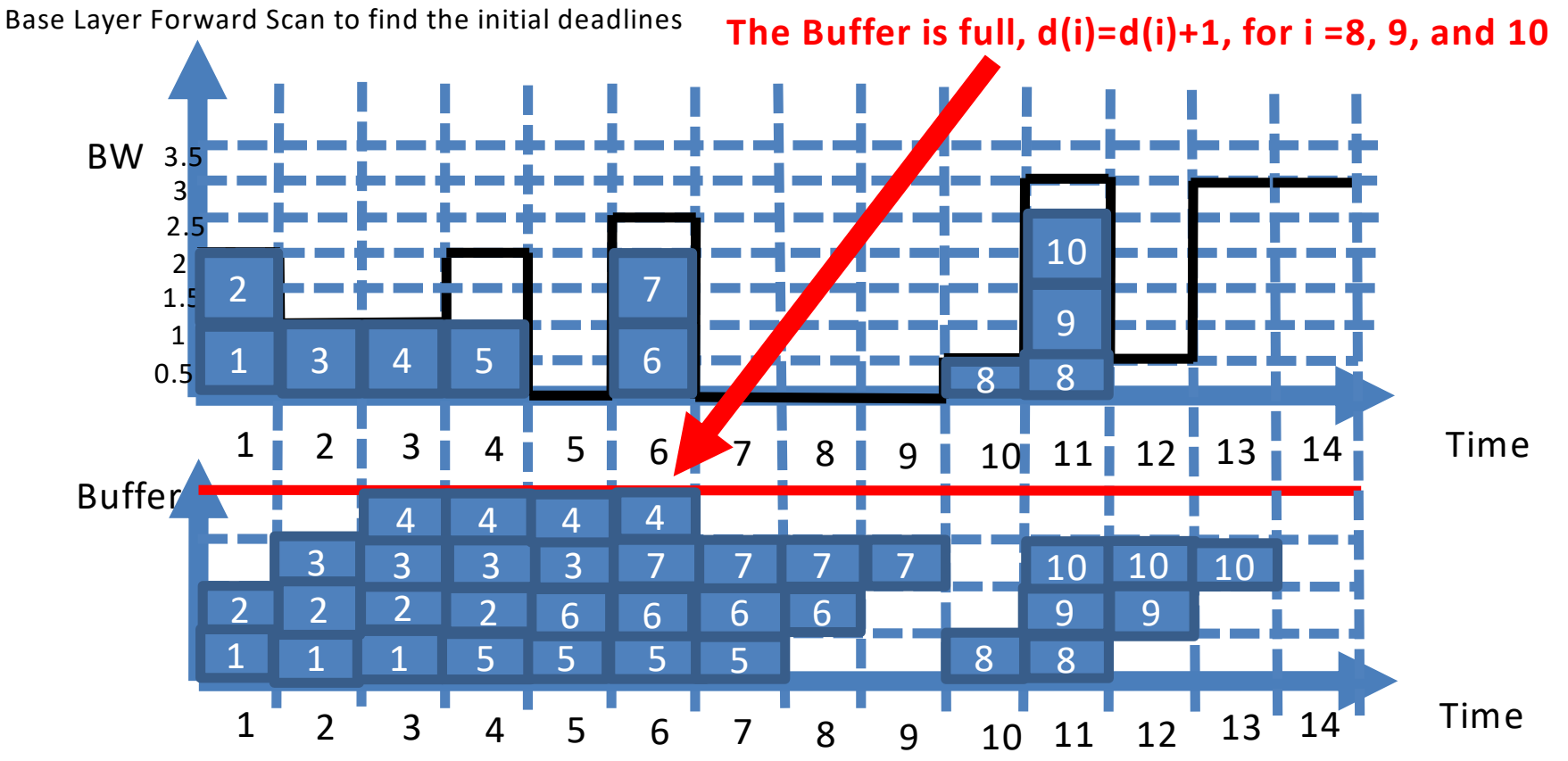
No Base Layer Skips

$$\sum_{j=1}^{(i-1)L+s+d(i)} z_0(i,j) = Y_0 \forall i$$

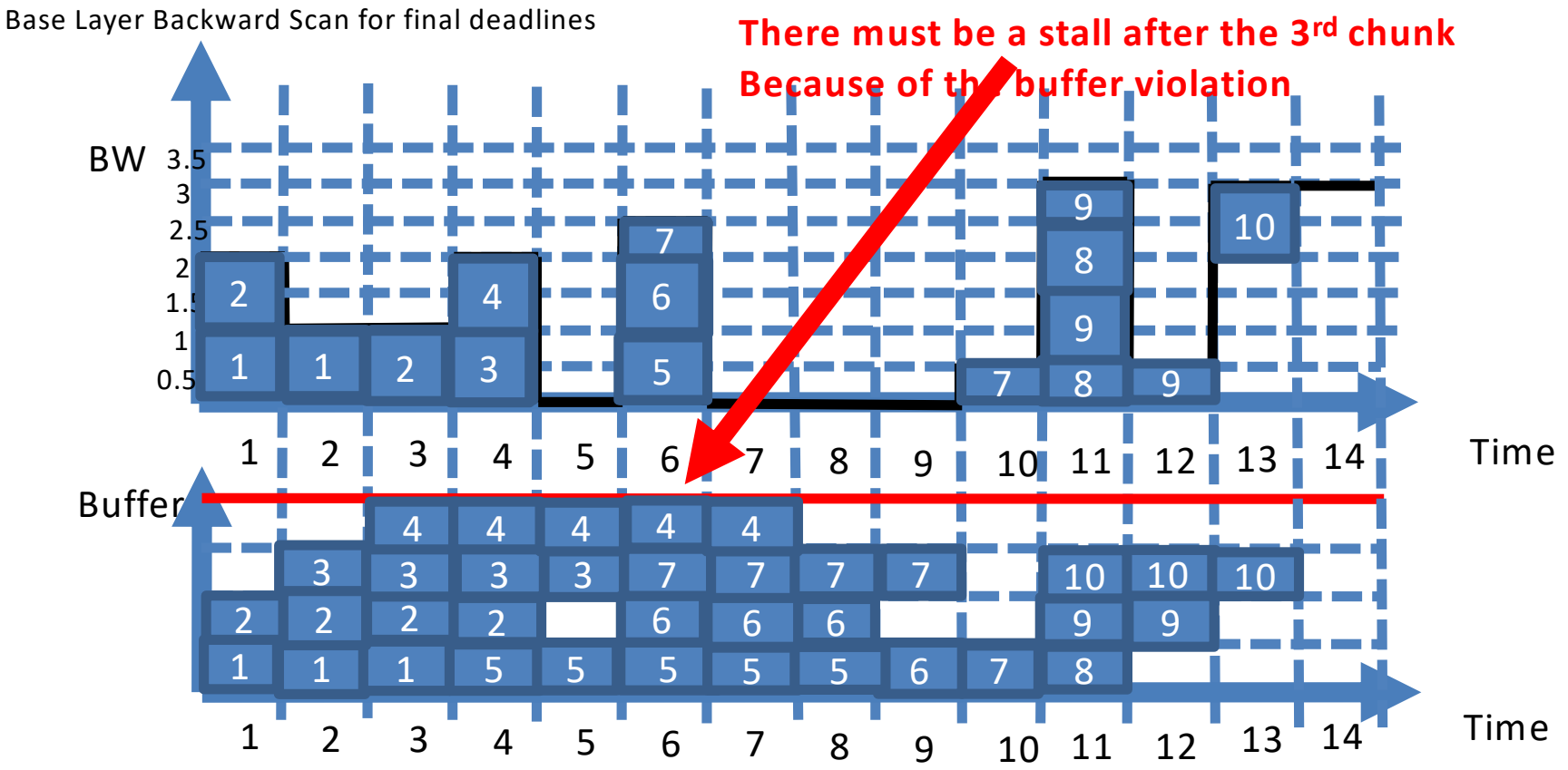
Stall duration up to the ith chunk

PURDUE UNIVERSITY

NO-SKIP BASED STREAMING FORMULATION



NO-SKIP BASED STREAMING FORMULATION



The rest is equivalent to skip version, backwar-forward scan starting from the BL

No-skip is a special case of skip in which chunk deadlines are chosen such that there are no skipsno skips

EVALUATIONS FOR SVC

- Simulation and TCP/IP-based emulation testbed in C++.
- For Performance measure, we compare our skip version with 3 Baselines
- Baseline 1: Conservative Algo, Fetch BL of all chunks before moving to E1 and so on. (Horizontal)
- Baseline 2: Optimistic Algo, Fetch all layers of current chunk and if there is still BW, fetch all possible layers of next chunk. (Vertical)
- Baseline 3: In between, Fetch all layers of current chunk and if there is still BW, fetch BL of later chunks
- We compare No-Skip version with
- Netflix BBA-0,
- Naive port of Microsoft Streamer to SVC (NMS).
- Slope based algorithm

TABLE I: SVC encoding bitrates used in our evaluation

playback layer	BL	EL1	EL2	EL3
nominal Cumulative rate (Mbps)	0.6	0.99	1.5	2.075

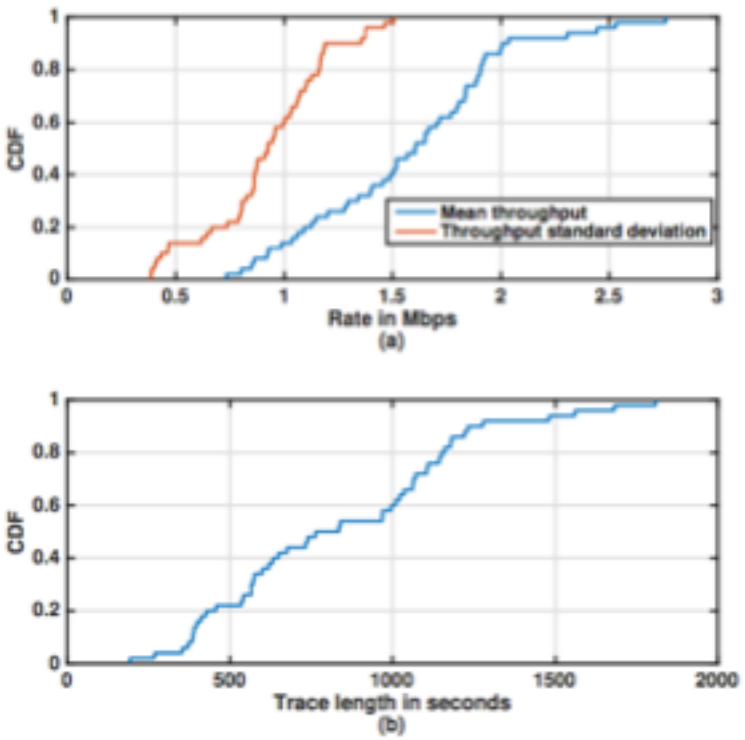
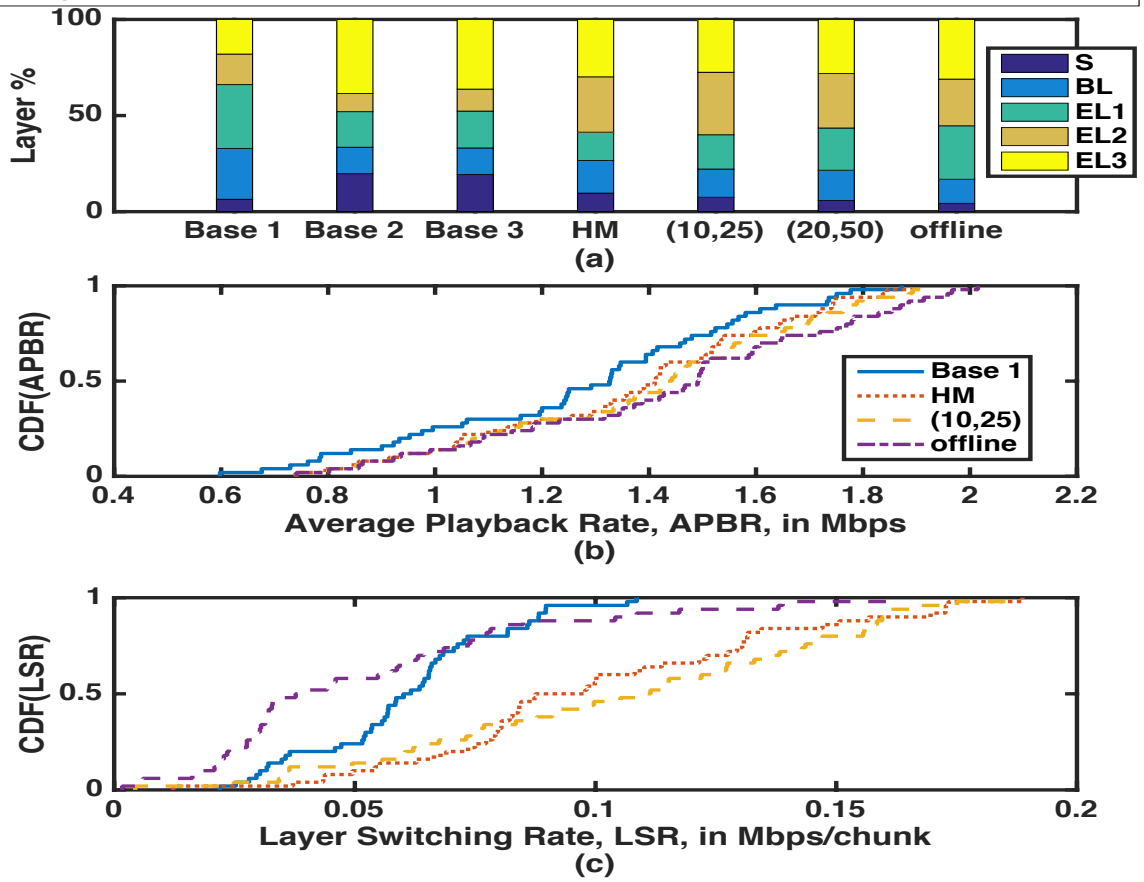


Fig. 4: Statistics of the bandwidth traces: (a) mean and standard deviation of each trace's throughput, and (b) trace length, across the 50 traces.

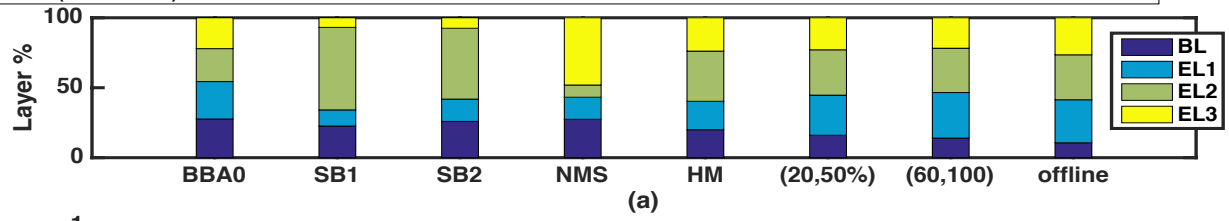
EVALUATION RESULTS FOR SVC - SKIP

%BL	26.5	14	14	16.9	14.5	15.7	12.5
APBR(Mbps)	1.05	1.19	1.18	1.3	1.31	1.31	1.334
%Skips	6.5	19.7	19.2	9	7	5.7	4.3

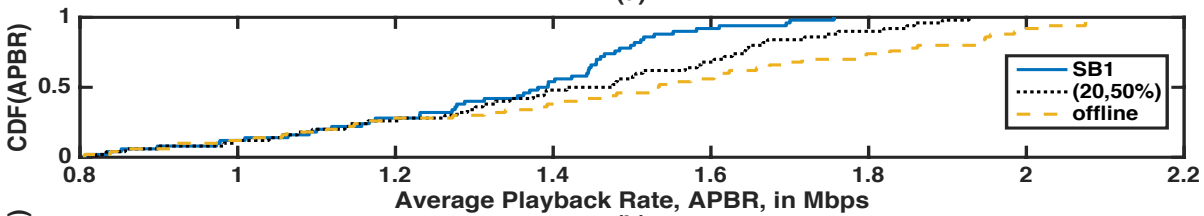


EVALUATION RESULTS FOR SVC - STALL

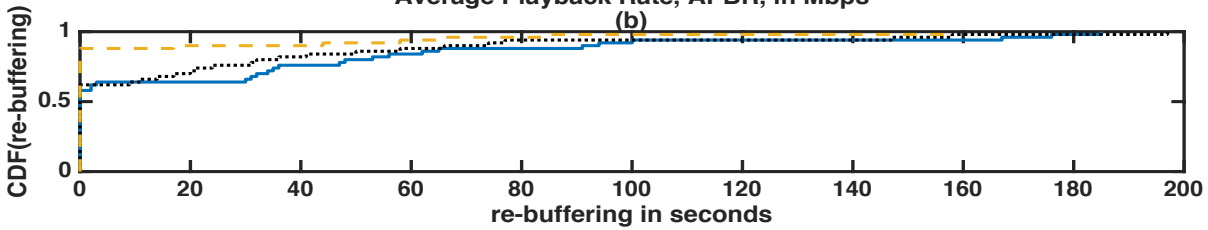
%BL	28	23	26	27.5	19	16	14	10.5
APBR(Mbps)	1.2	1.25	1.2	1.43	1.35	1.33	1.32	1.37
Stall time (minutes)	15	23	12	53	19	15	10	7



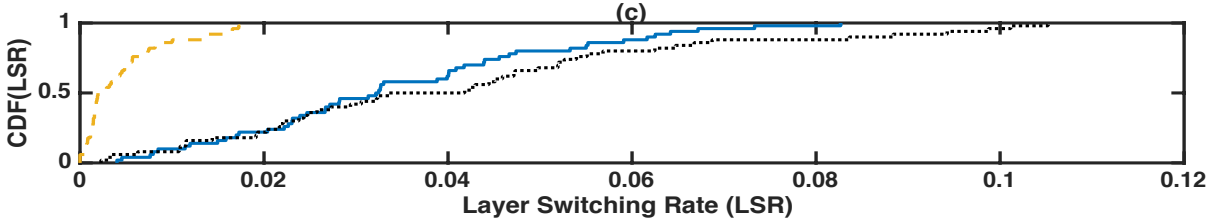
(a)



(b)



(c)



(d)

EVALUATION FOR AVC

- Real implementation in dash.js open source video platform.
- Comparison with FastMPC, Festive, BBA, RB, and dash.js reference
- We use real video ”Envivo”, chunk duration is 4s.

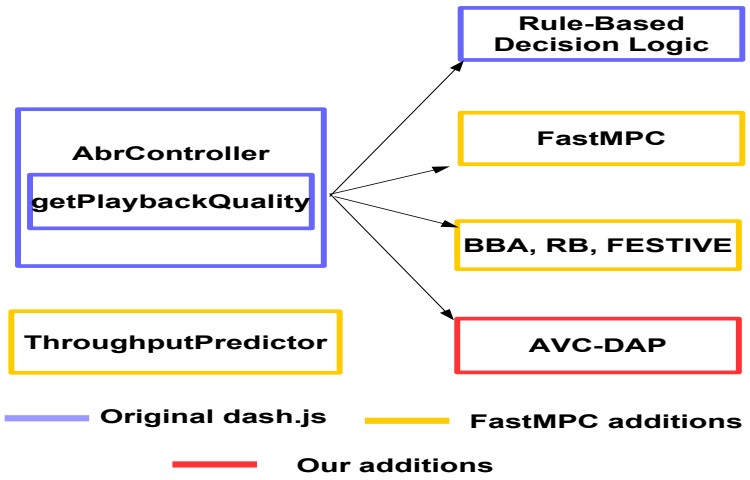
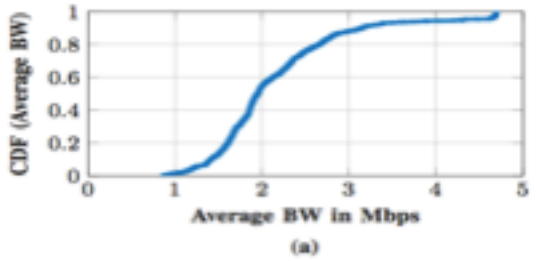


Table 1: AVC encoding chunk sizes in MB of the Envivio video used in our evaluation

Quality level	level-0	level-1	level-2	level-3	level-4
Min	0.0433	0.0786	0.1265	0.2213	0.3205
Mean	0.1693	0.2916	0.4795	0.949	1.403
Max	0.2342	0.3855	0.6217	1.286	1.918



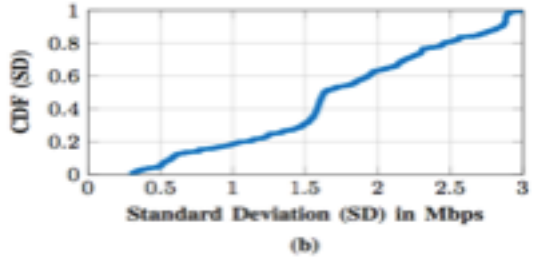


Fig. 3. Statistics of the two bandwidth traces: (a) mean, and (b) standard deviation of each trace's available bandwidth.

EVALUATION RESULTS FOR AVC

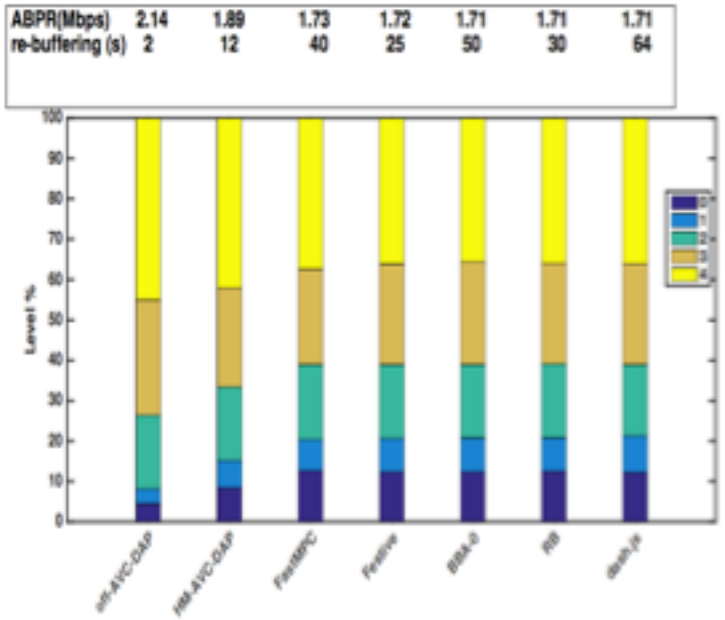
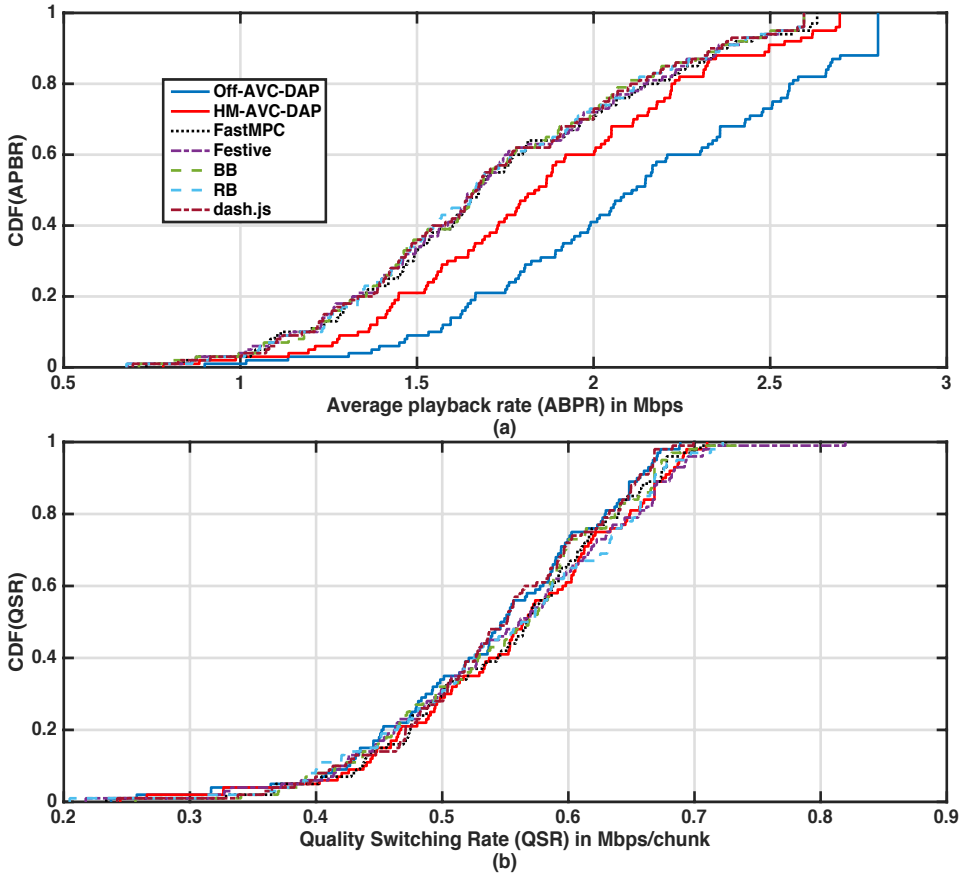


Figure 3: Comparison with other approaches: Video Quality distribution



SUMMARY

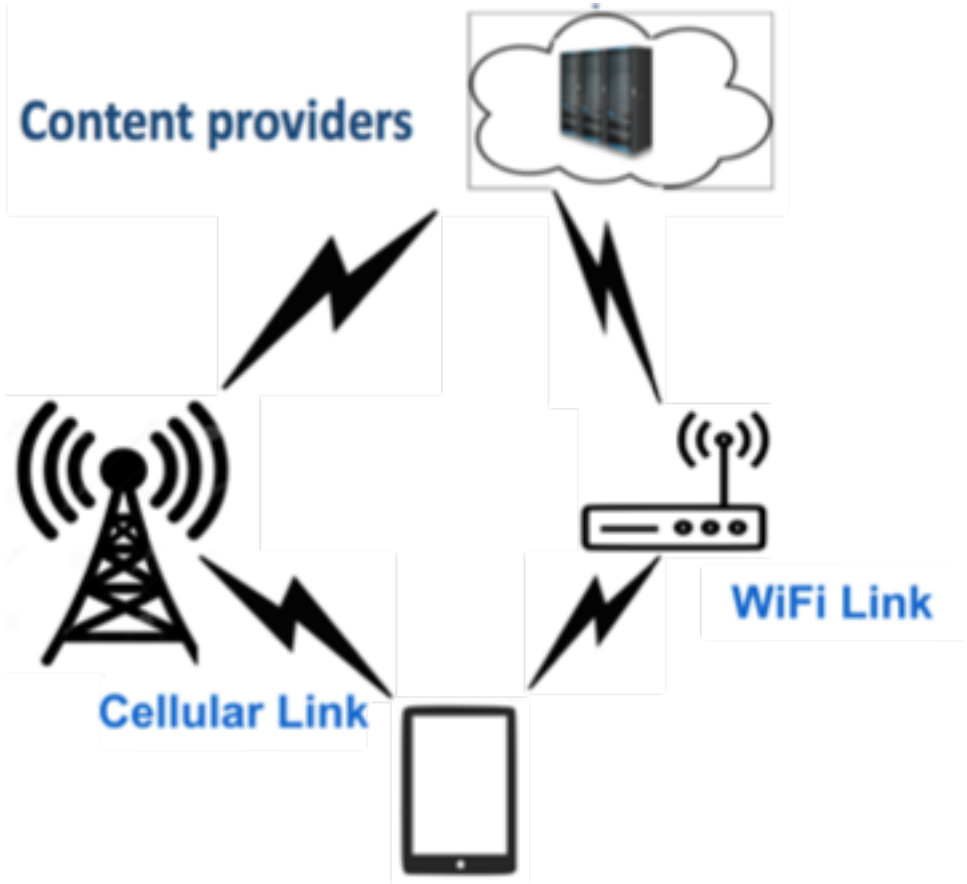
- Novel formulation for video streaming
- Novel algorithms for video streaming
- Low complexity, and improved performance as compared to the baselines
- Some discrete optimization problems can be solved with low complexity.

OUTLINE

- Video Streaming Algorithms
  - Single Path
  - Multiple Paths
  - Giant Client
- 360-degree Video Streaming
- Video Streaming over Cloud

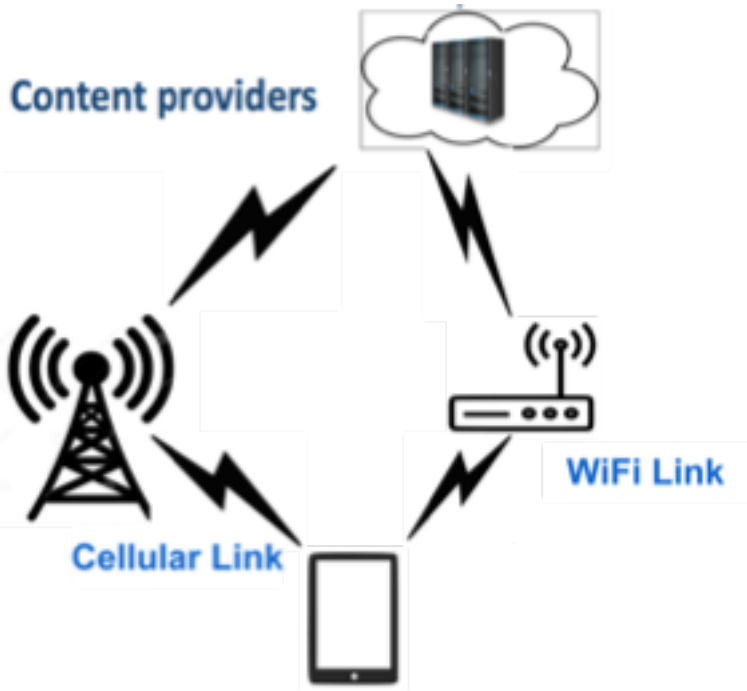
MULTIPATH STREAMING

- Use multiple interfaces (e.g WiFi and LTE) to download video chunks



MULTIPATH STREAMING

- MPTCP may or may not be allowed for using two links
- A vast amount of commercial content providers do not support MPTCP because it requires OS kernel update at both the client and server side.
- MPTCP uses special TCP extensions that are often blocked by middle-boxes of the commercial content providers (e.g MPTCP over Port 80/443 is blocked by most U.S. cellular carriers).
- Implementing MPTCP with link preference further requires message exchange between the rate adaptation logic at the application layer and MPTCP in order to disable/enable parallel TCP connections.



PROBLEM FORMULATION WITH AVC USING MPTCP

$$\lambda_a^{(1)} > C \cdot \Big(\sum_{n=a}^{N} \lambda_n^{(2)} + \sum_{n=a+1}^{N} \lambda_n^{(1)} \Big).$$

$$\lambda_a^{(2)} > C \cdot \Big(\sum_{n=a+1}^{N} \lambda_n^{(2)} Y_n \Big).$$

$$\mu \gg \max(\lambda_n^k)$$

Note that a chunk can be Split into the two links since splitting a chunk among links is Possible with MPTCP

Maximize: $\sum_{i=1}^{C} \sum_{j=1}^{(i-1)L+s} \sum_{n=0}^{N} \left(\lambda_n^1 z_n^{(1)}(i,j) + \lambda_n^2 z_n^{(2)}(i,j) \right) - \mu d(C)$

Stall duration

subject to

$$\sum_{j=1}^{deadline(i)} \left(z_0^1(i,j) + z_0^2(i,j) \right) = Y_0 \ \forall i$$

$$\sum_{j=1}^{deadline(i)} \left(z_n^1(i,j) + z_n^2(i,j) \right) = Z_{n,i} \ \forall i = i, n = 1, \cdots, N$$

$$Z_{n,i} \le \frac{Y_n}{Y_{n-1}} Z_{n-1,i} \ \forall i, n$$

$$z_n^{(2)}(i,j) = 0 \ \forall i, j, n > n_2$$

$$\sum_{n=0}^{N} \sum_{i=i'}^{C} z_n^k(i,j) \le B^k(j) \ \forall j, k \in \{1,2\}$$

$$z_n^k(i,j) \ge 0 \ \forall i, j, k \in \{1,2\}$$

$$z_n^k(i,j) = 0 \ \forall i, j > deadline(i), k \in \{1,2\}$$

$$Z_{n,i} \in \{0, Y_{n,i}\} \ \forall i, n$$

$$d(i) \ge 0, d(i+1) \ge d(i), deadline(i) = S + (i-1)L + d(i) \ \forall i$$

PURDUE UNIVERSITY

EXAMPLE

Example 1: MPTCP-AVC

In this example we consider a video consists of 10 chunks, 1s each

Startup delay =1 s

The video is encoded into two quality levels

- 0-th Quality level, Bit rate is 1Mb/s, so chunk size is 2Mb
- 1-st Quality level, Bitrate 3Mb/s, so chunk size is 3Mb,

Therefore, the size difference between the two quality levels is 1Mb

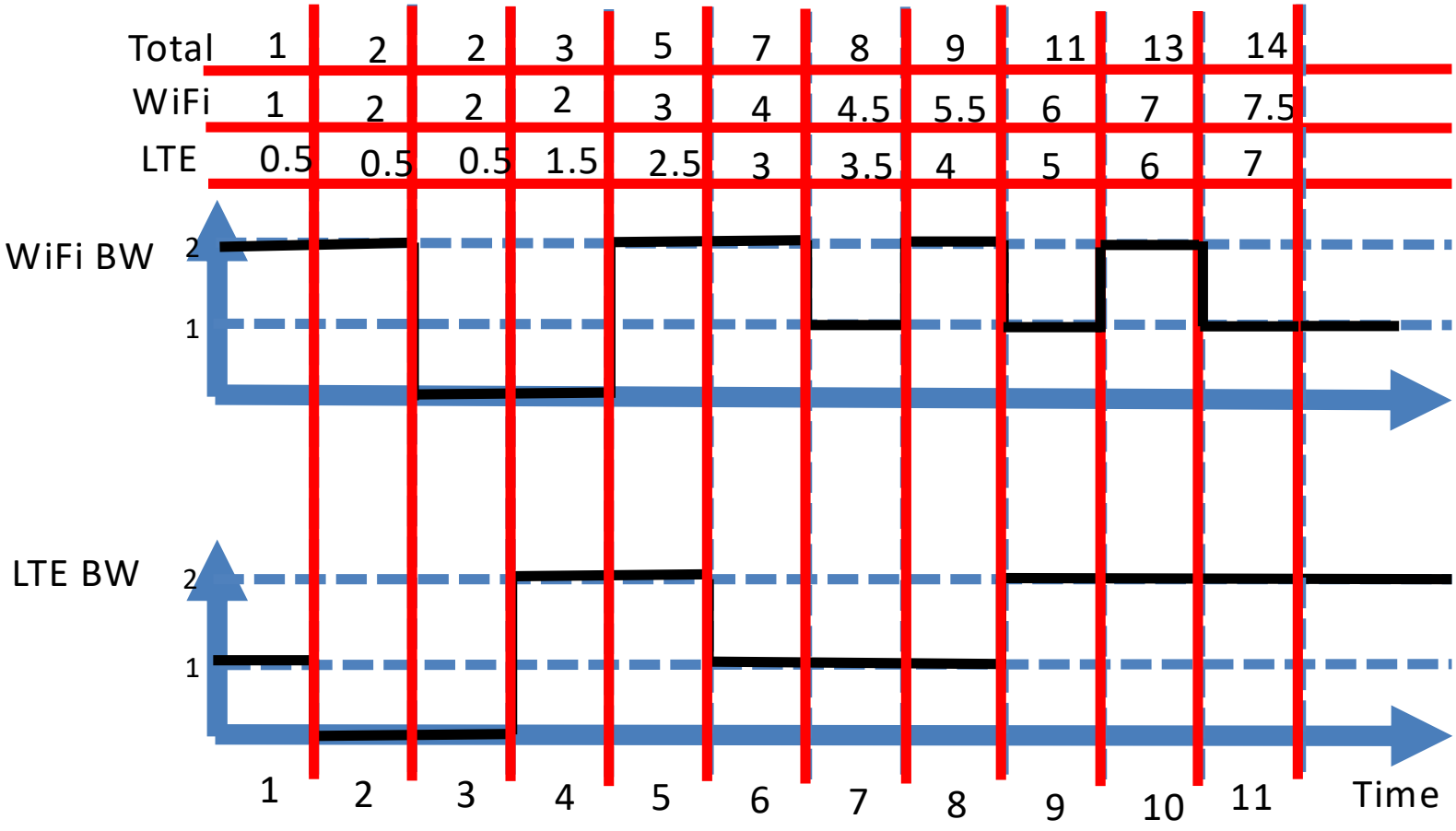
0-th Quality level

1-st Quality level

Size difference between
The two quality levels

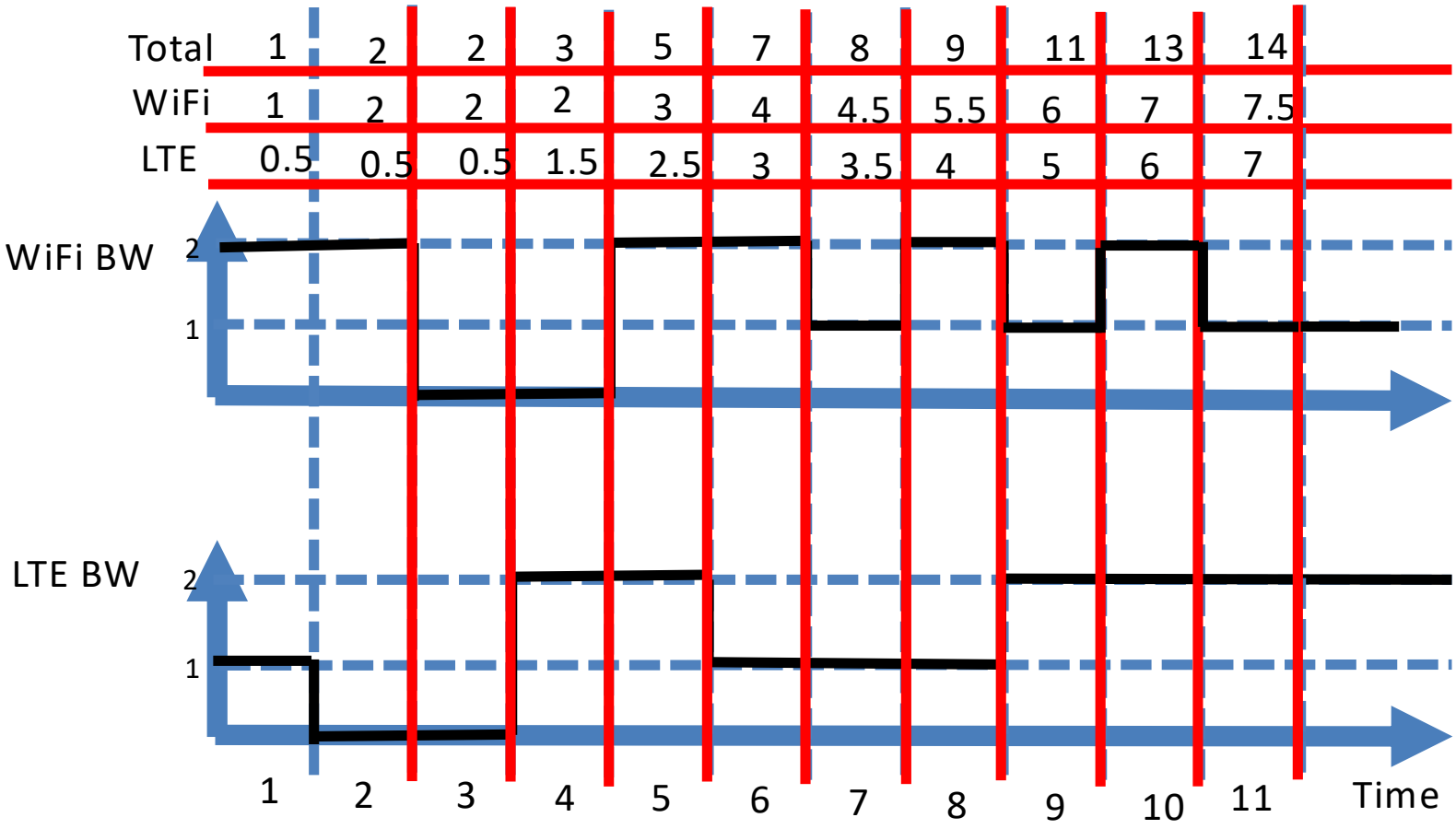
PURDUE
UNIVERSITY

FORWARD SCAN FOR 0-QUALITY LEVEL

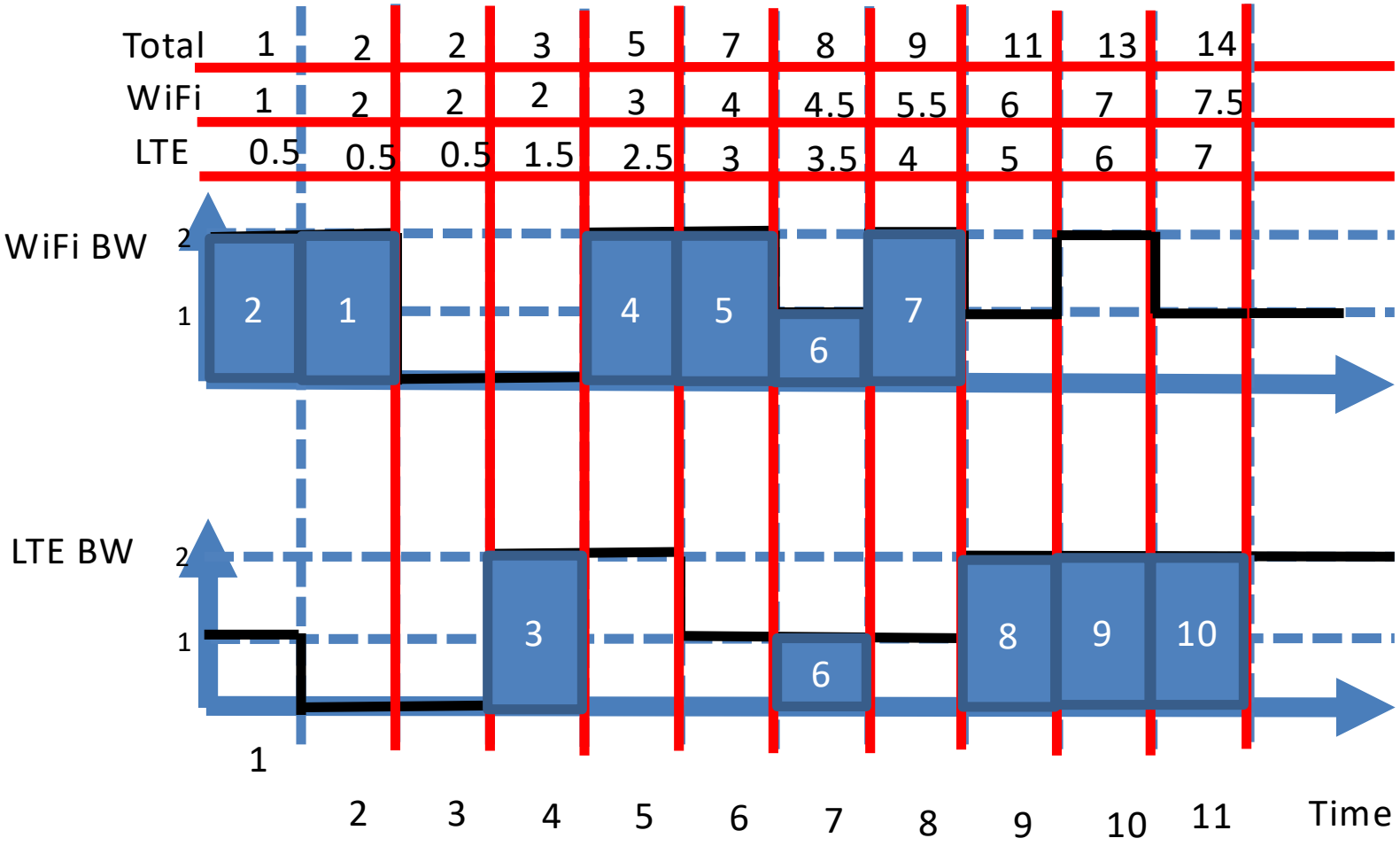


Total stall duration is 1 second

Deadline(i)=(i-1)L+s+d(i), s=1, and d(i)=1 for all I
i.e deadline(1)=2

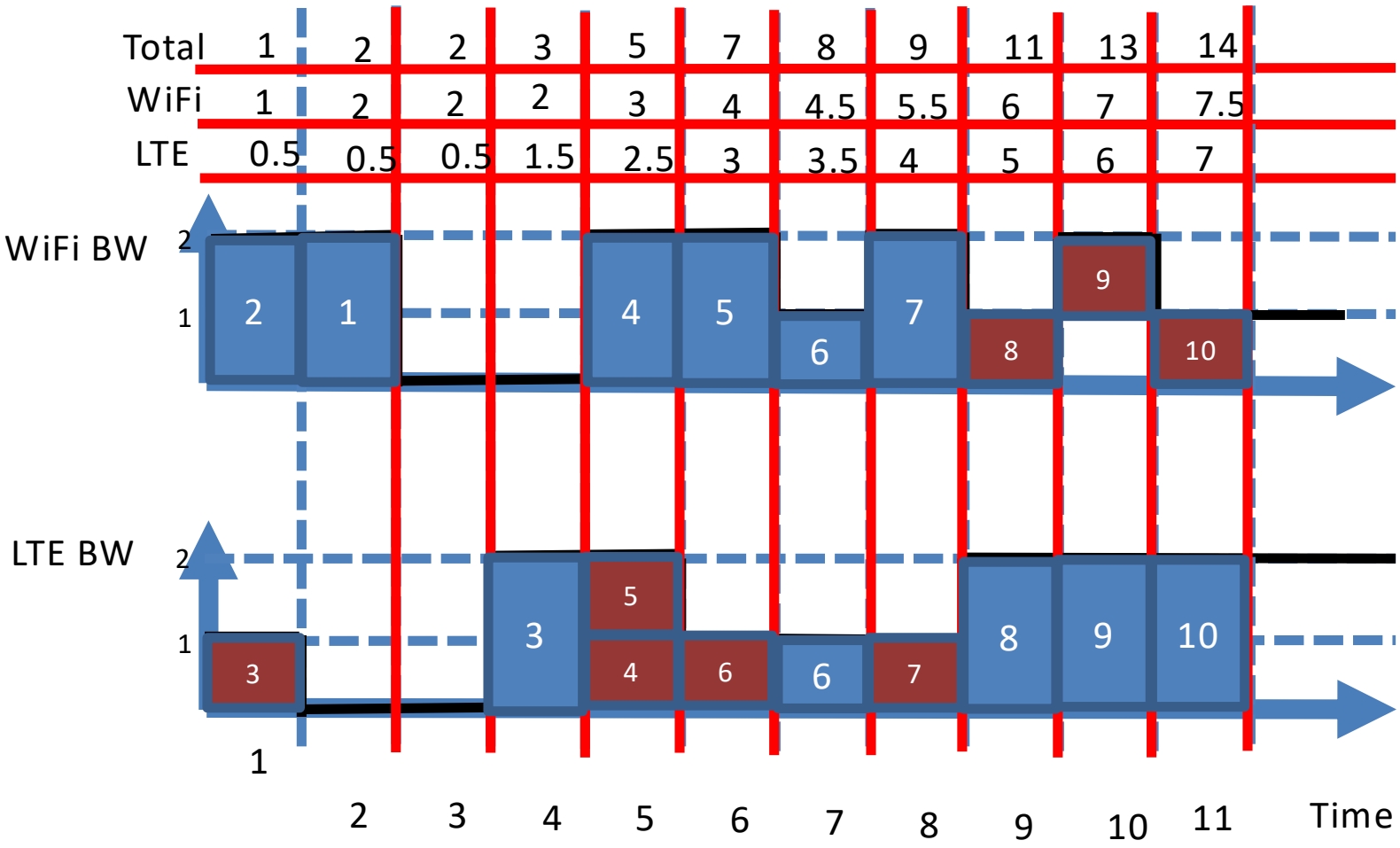


BACKWARD SCAN FOR 0-QUALITY LEVEL

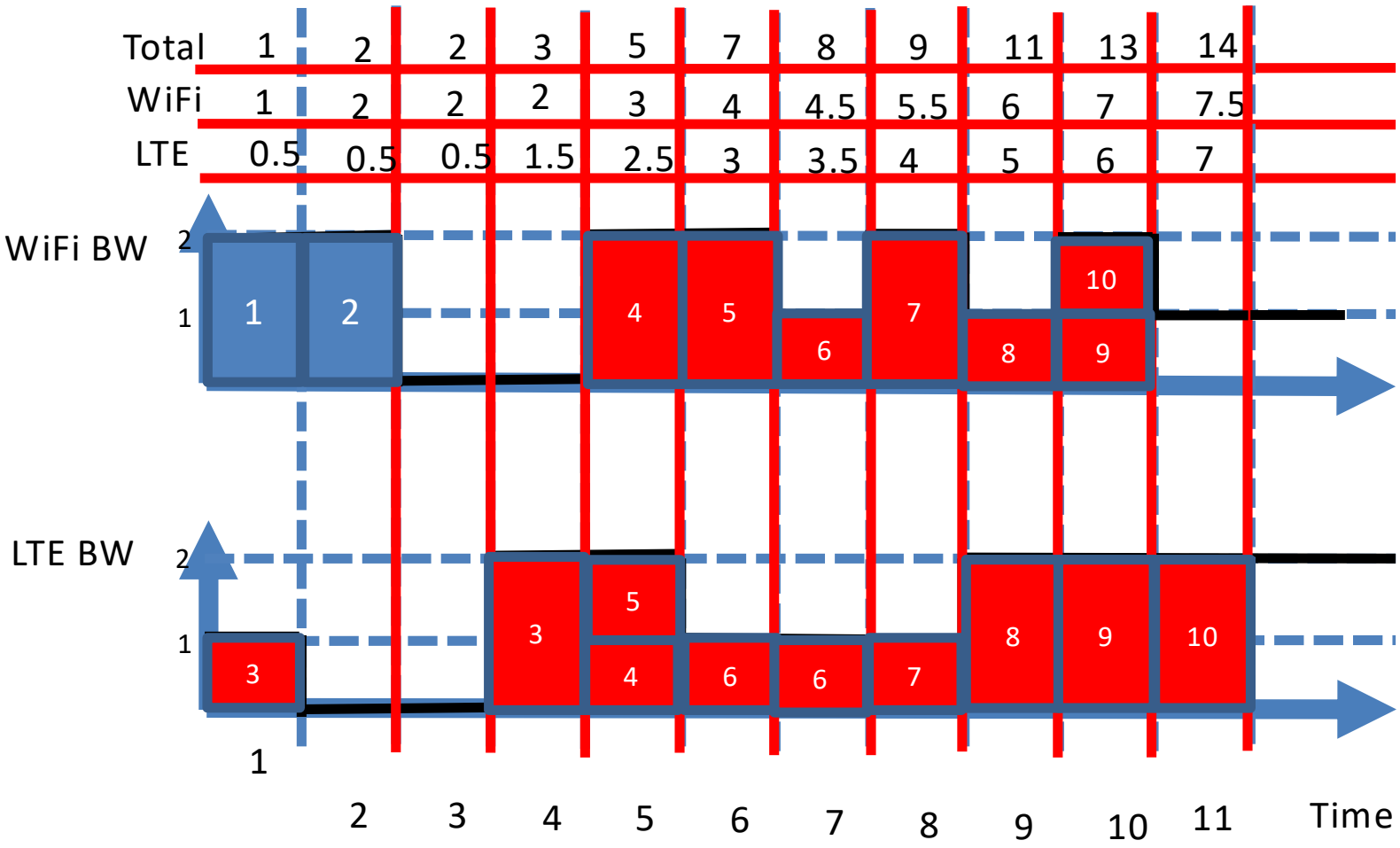


PURDUE UNIVERSITY

DECISION FOR 1-QUALITY LEVEL

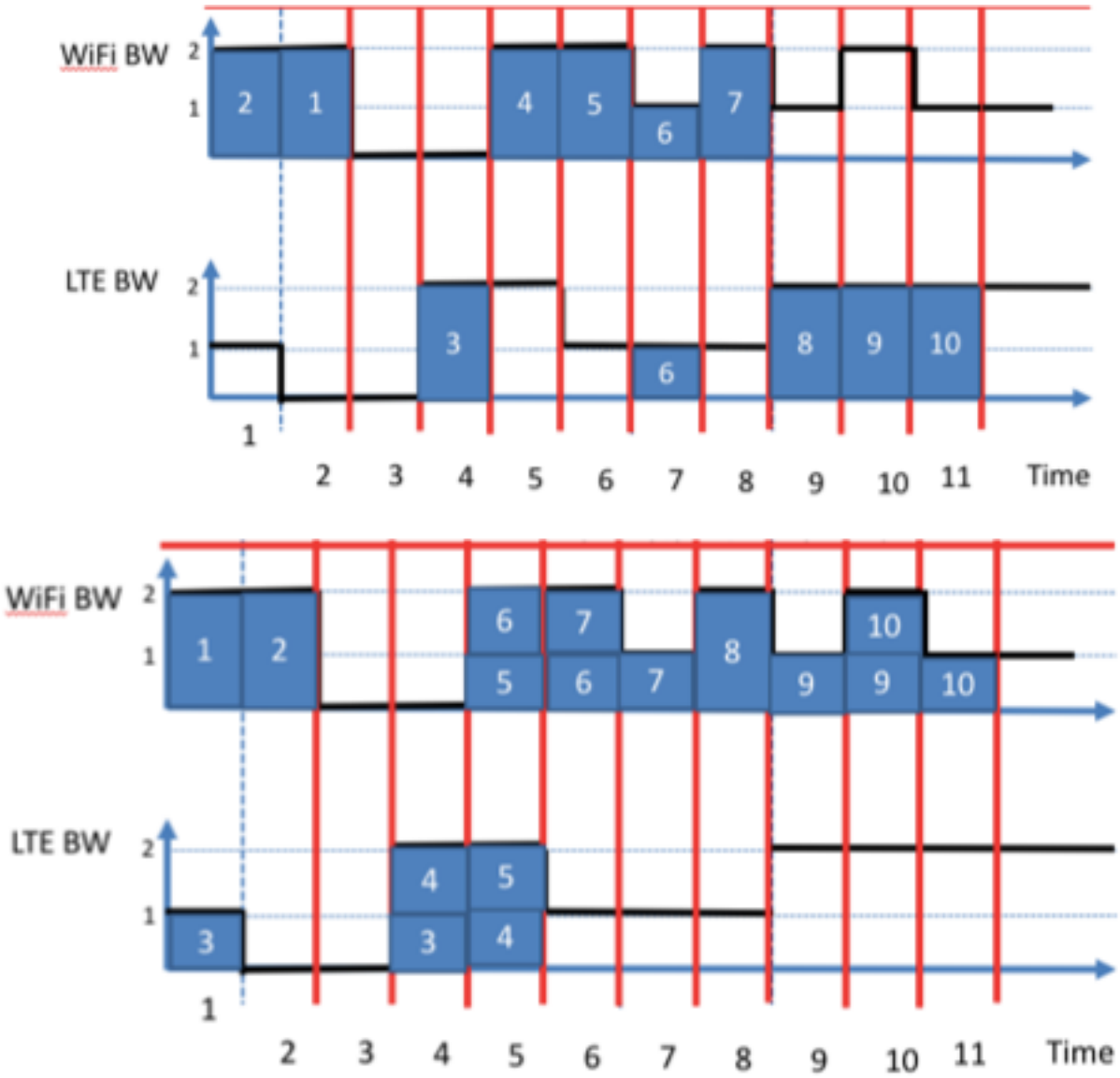


DECISION FOR 1-QUALITY LEVEL



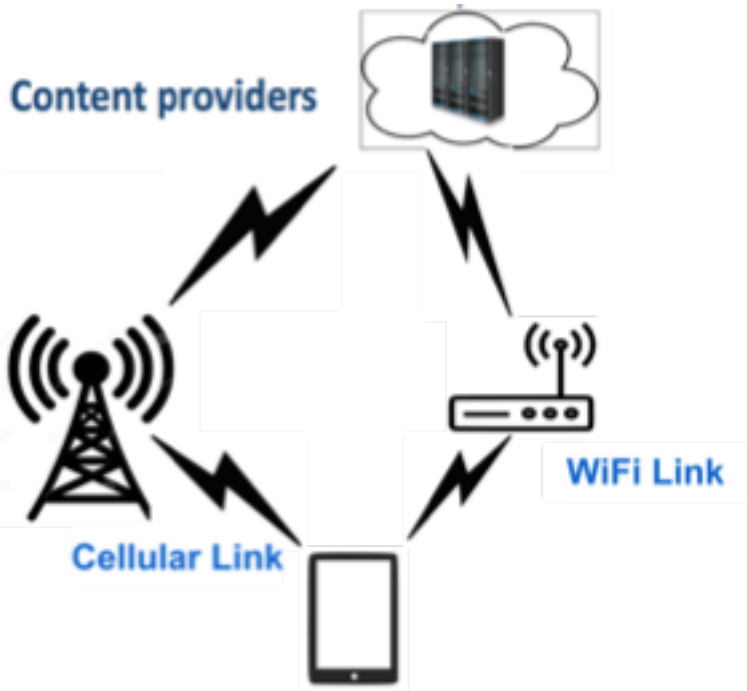
PURDUE UNIVERSITY

CONSIDERING PREFERENCES: EXCHANGE ALGORITHM



MULTIPATH STREAMING

- MPTCP may or may not be allowed for using two links
- A vast amount of commercial content providers do not support MPTCP because it requires OS kernel update at both the client and server side.
- MPTCP uses special TCP extensions that are often blocked by middle-boxes of the commercial content providers (e.g MPTCP over Port 80/443 is blocked by most U.S. cellular carriers).
- Implementing MPTCP with link preference further requires message exchange between the rate adaptation logic at the application layer and MPTCP in order to disable/enable parallel TCP connections.



PURDUE UNIVERSITY

WITHOUT MPTCP

- Without MPTCP, we use SVC since each layer can be obtained from different link.
- Forward scan: Finds stalls using greedy approach
- Other steps, similar to before, where we assume that LTE can only be used upto a layer and prefer WiFi to LTE.

EVALUATIONS

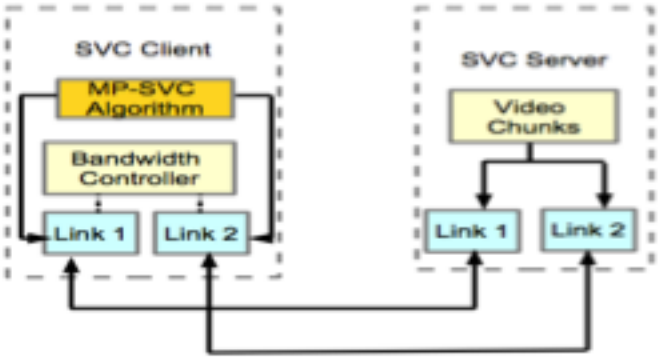


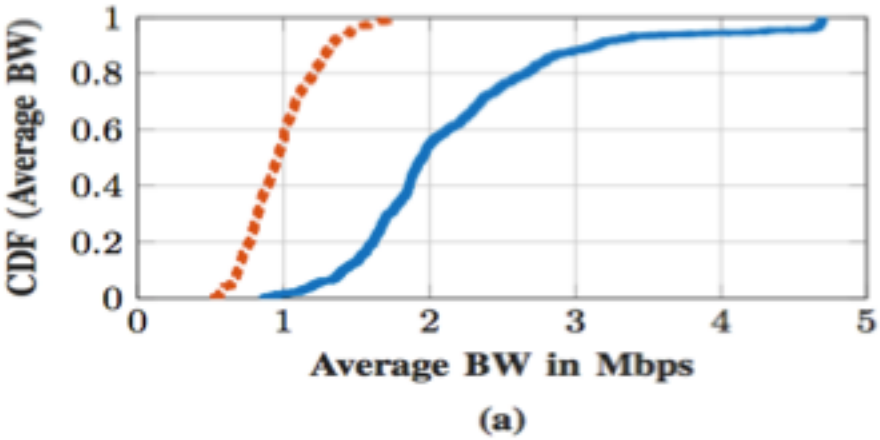
Fig. 1. System Architecture for Emulation of MP-SVC.

TABLE I
SVC ENCODING BITRATES USED IN OUR EVALUATION

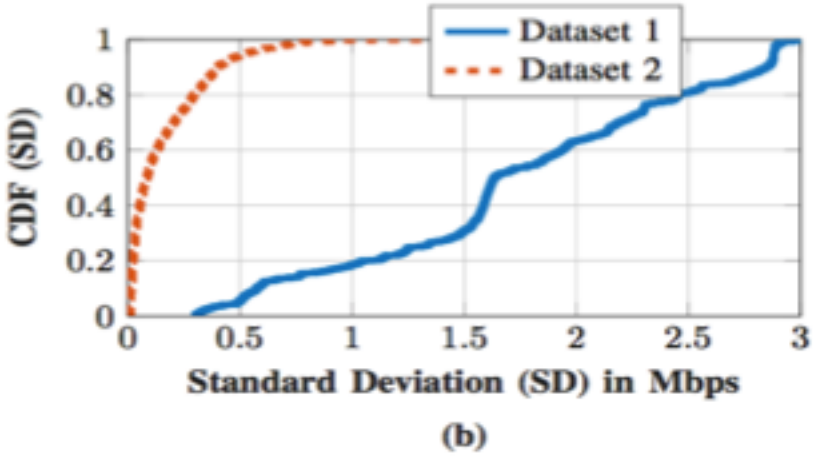
playback layer	BL	EL1	EL2	EL3
nominal Cumulative rate (Mbps)	0.6	0.99	1.5	2.075

Table 1: AVC encoding chunk sizes in MB of the Envivio video used in our evaluation

Quality level	level-0	level-1	level-2	level-3	level-4
Min	0.0433	0.0786	0.1265	0.2213	0.3205
Mean	0.1693	0.2916	0.4795	0.949	1.403
Max	0.2342	0.3855	0.6217	1.286	1.918



(a)



(b)

EVALUATION RESULTS FOR SVC

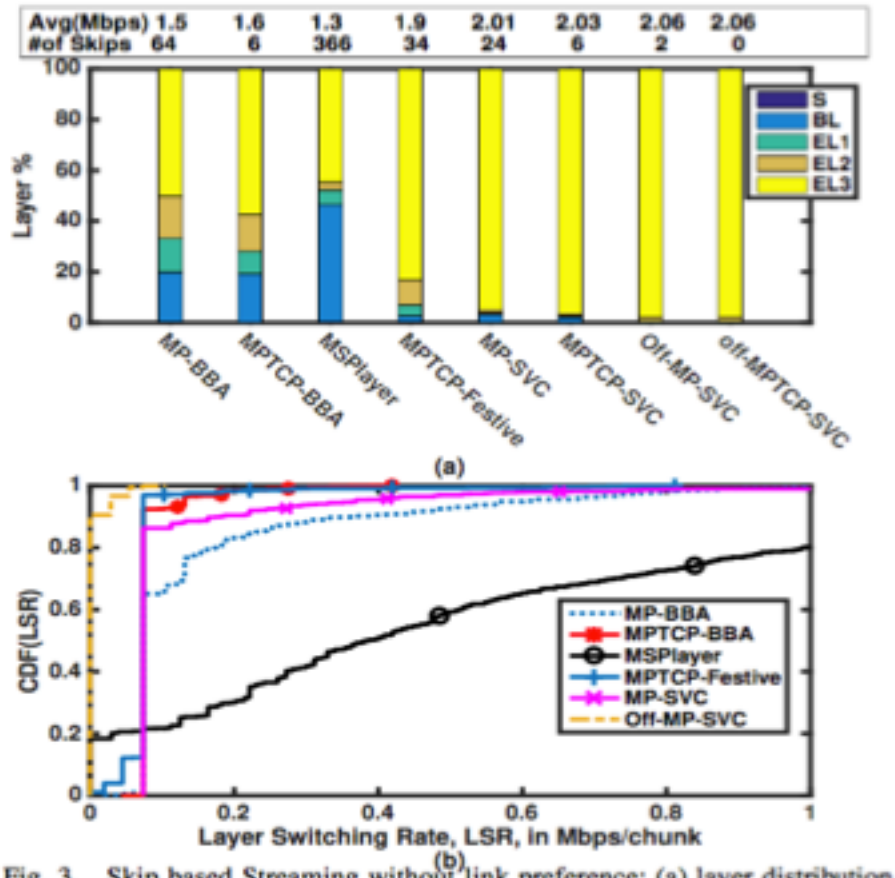


Fig. 3. Skip based Streaming without link preference: (a) layer distribution, and (b) CDF of layer switching rate.

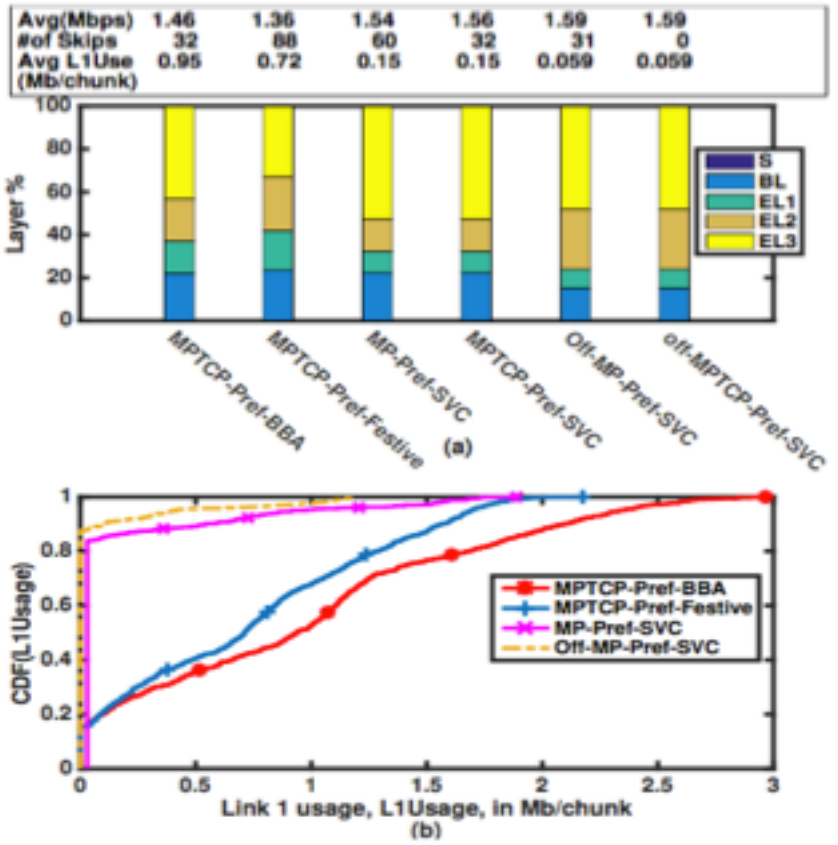
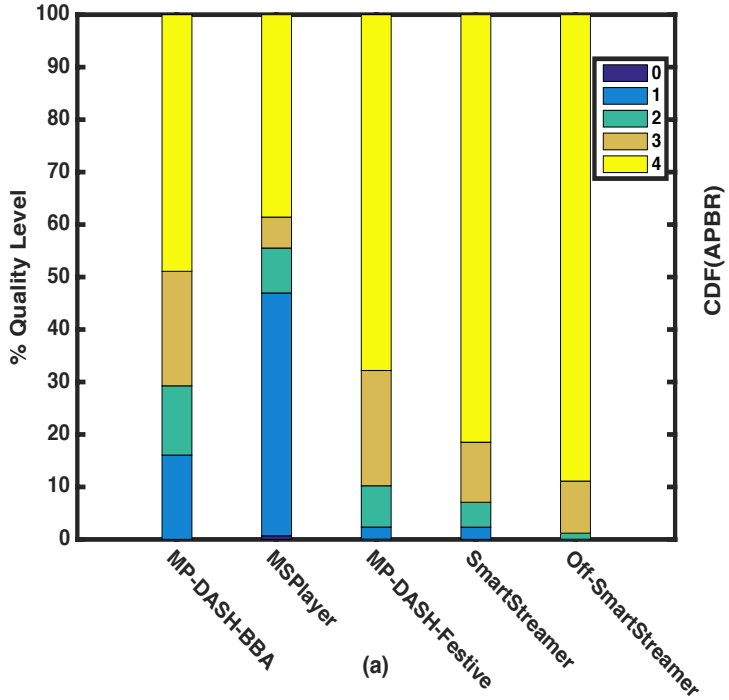


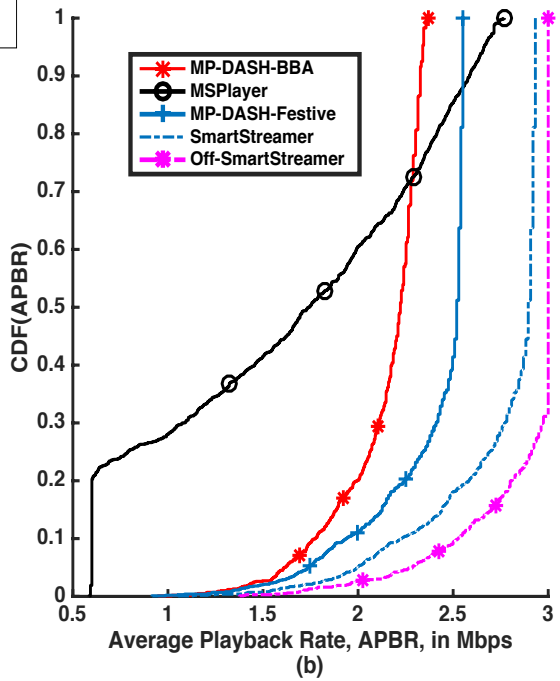
Fig. 4. Skip based streaming with Link Preference: (a) layer distribution, and (b) CDF of link 1 usage.

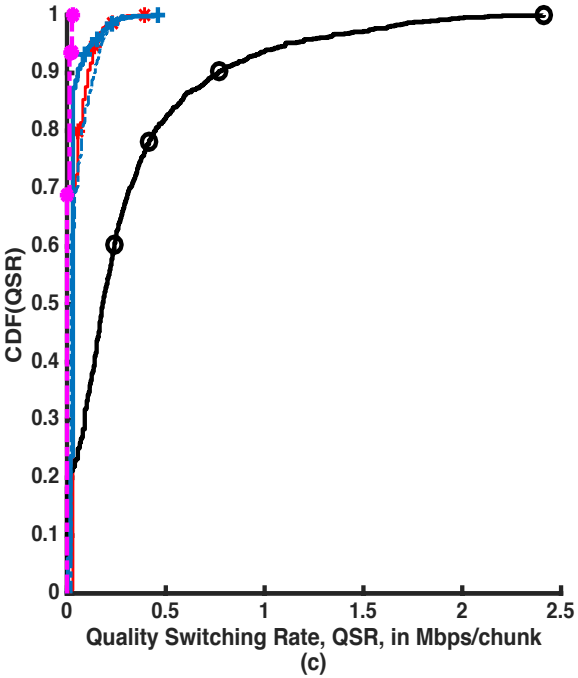
EVALUATION RESULTS FOR AVC

No Preference Results

Avg(Mbps)	2.2	1.65	2.48	2.75	2.87
Stall(seconds)	8	213	110	8	0



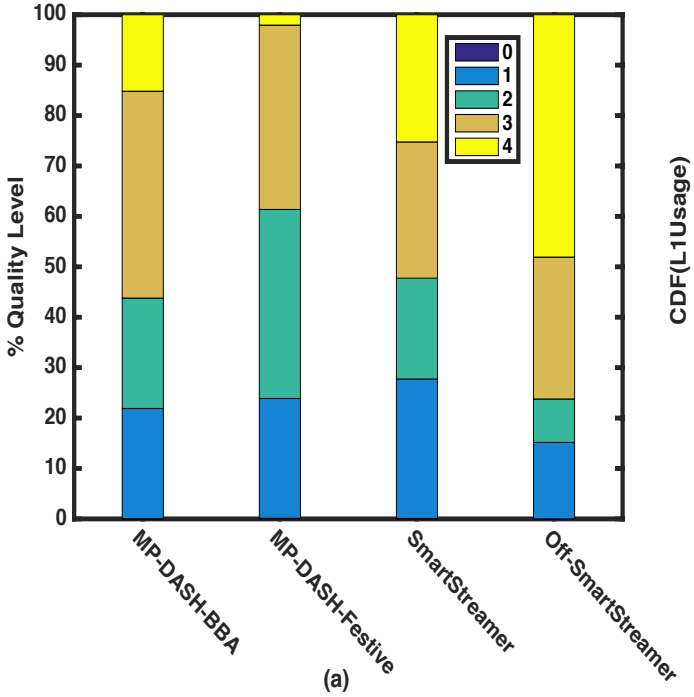


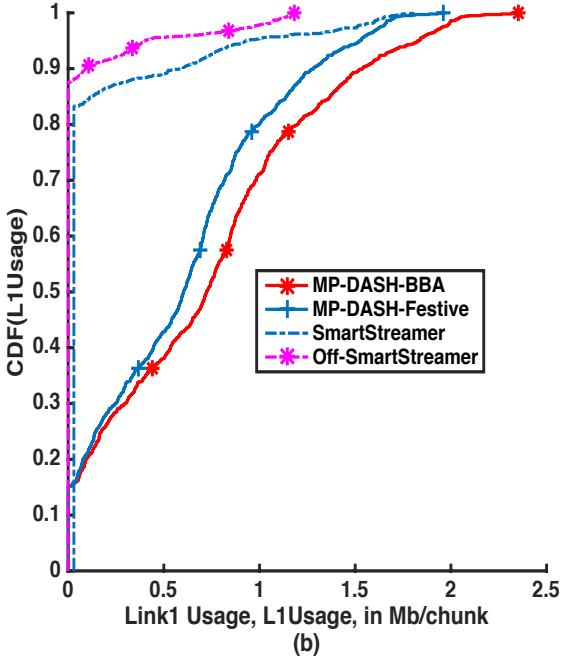


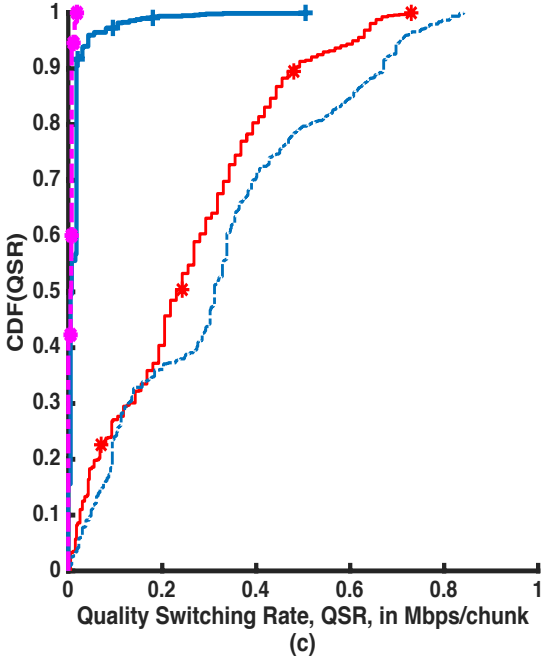
EVALUATION RESULTS FOR AVC

Preference Results

Avg(Mbps)	1.6	1.4	1.67	2.18
Stall(seconds)	8	110	8	0







SUMMARY

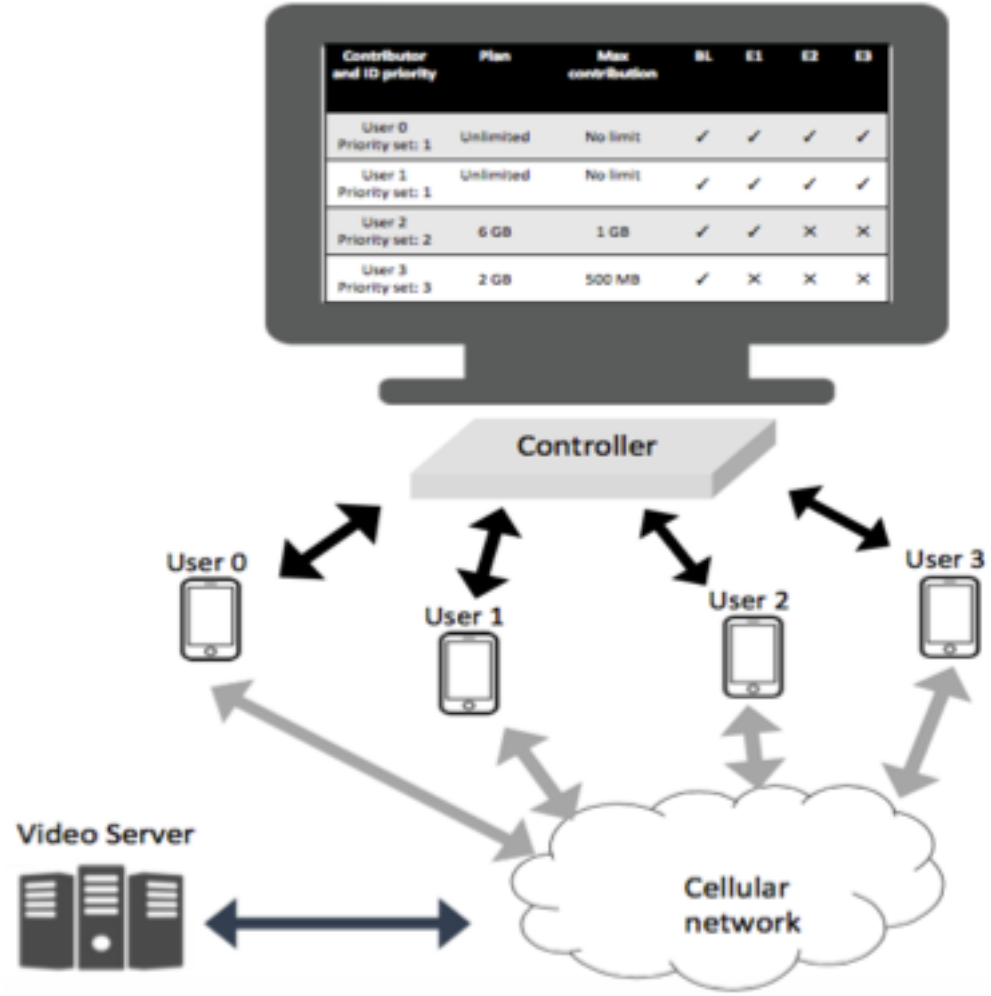
- Novel formulation for multi-path video streaming
- Considers link preference in the design of multi-path streaming algorithms
- Novel algorithms with low complexity and improved performance as compared to the baselines

OUTLINE

- Video Streaming Algorithms
  - Single Path
  - Multiple Paths
  - Giant Client
- 360-degree Video Streaming
- Video Streaming over Cloud

COOPERATIVE VIDEO STREAMING

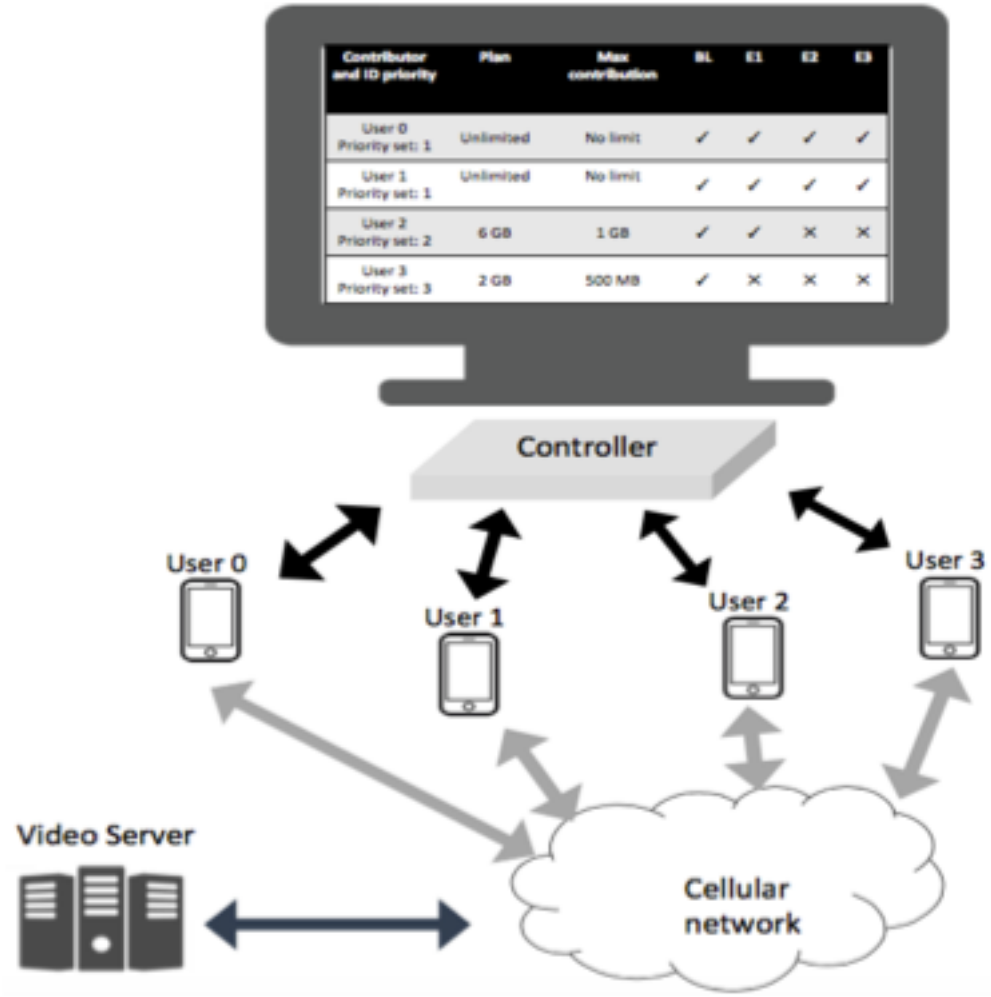
- Group of users are gathered in a low speed internet place.
- They are interested in watching a TV show on a common screen
- Willing to cooperate, but:
  - Users may have different data plan limits
  - Devices are running into different energy levels
- Maximum contribution of every user should not be violated



COOPERATIVE VIDEO STREAMING

- Multipath TCP cannot be used due to different users from potentially different carriers involved

- The approach for multipath can be used here for fetching chunks from servers.



EVALUATION

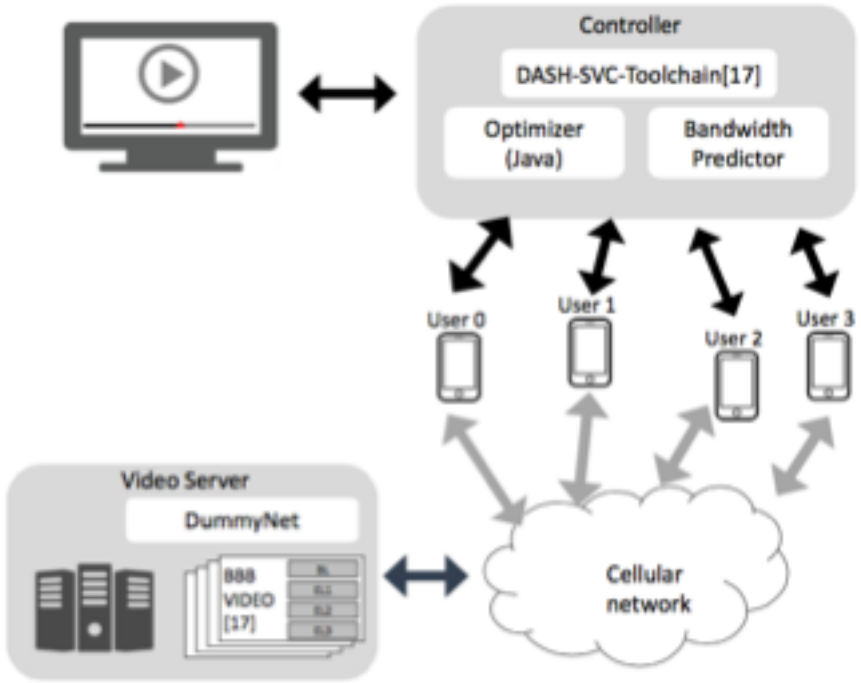


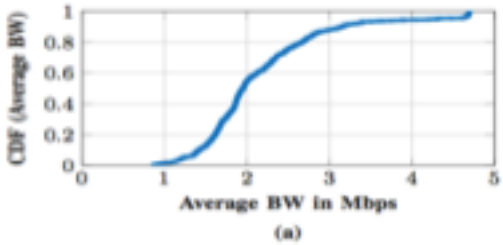
Fig. 2. System Setup.

TABLE I
SVC ENCODING BITRATES USED IN OUR EVALUATION

playback layer	BL	EL1	EL2	EL3
nominal Cumulative rate (Mbps)	1.5	2.75	4.8	7.8

TABLE II
MAX CONTRIBUTIONS USED IN OUR EVALUATION

User No.	1	2	3	4
Max Contribution (Mb)	700	500	100	100



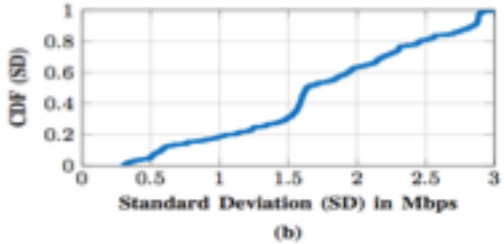
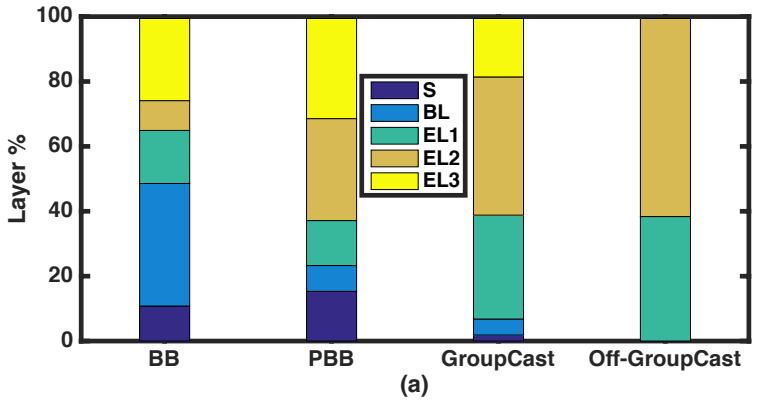
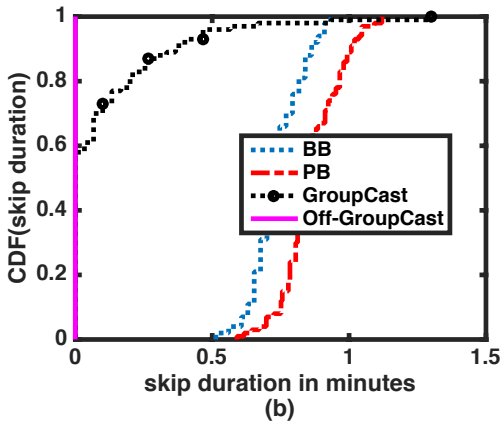
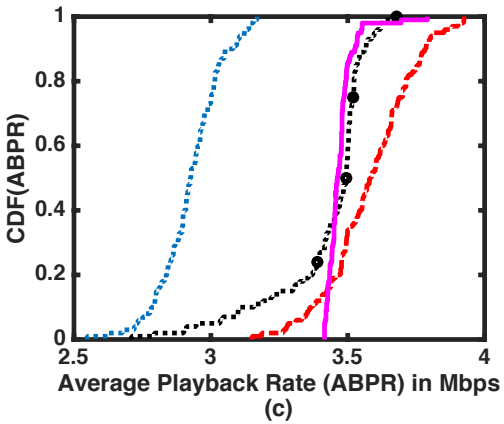


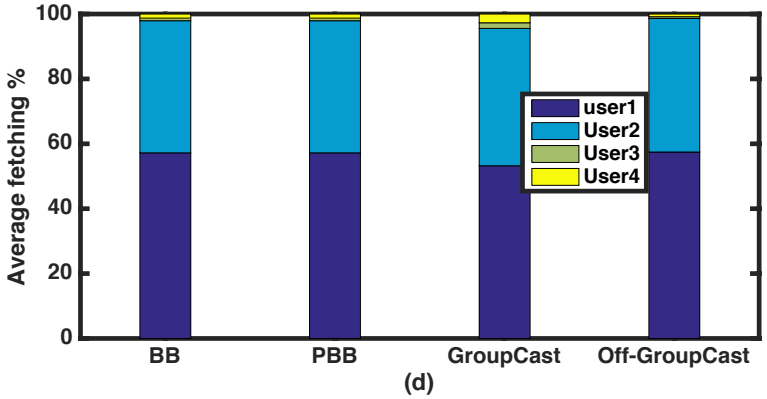
Fig. 3. Statistics of the two bandwidth traces: (a) mean, and (b) standard deviation of each trace's available bandwidth.

RESULTS









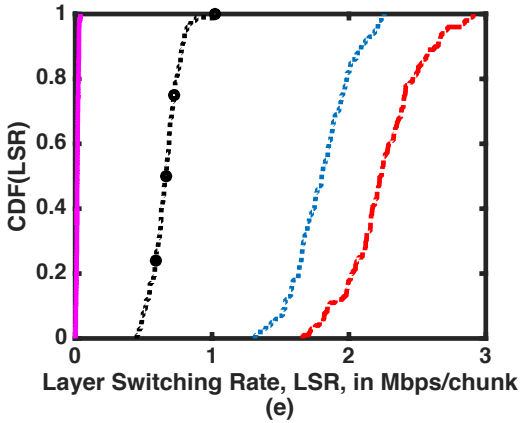


TABLE I
SVC ENCODING BITRATES USED IN OUR EVALUATION

playback layer	BL	EL1	EL2	EL3
nominal Cumulative rate (Mbps)	1.5	2.75	4.8	7.8

TABLE II
MAX CONTRIBUTIONS USED IN OUR EVALUATION

User No.	1	2	3	4
Max Contribution (Mb)	700	500	100	100

PURDUE UNIVERSITY

SUMMARY

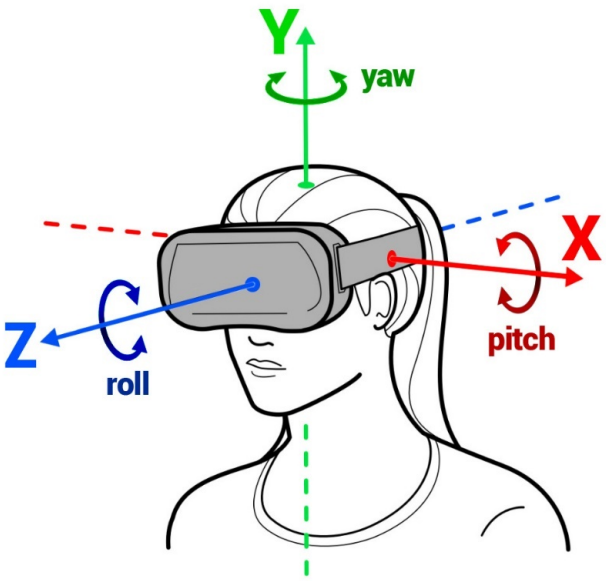
- Natural extension of the work to cooperative streaming
- Cooperation helps fetching content from multiple users

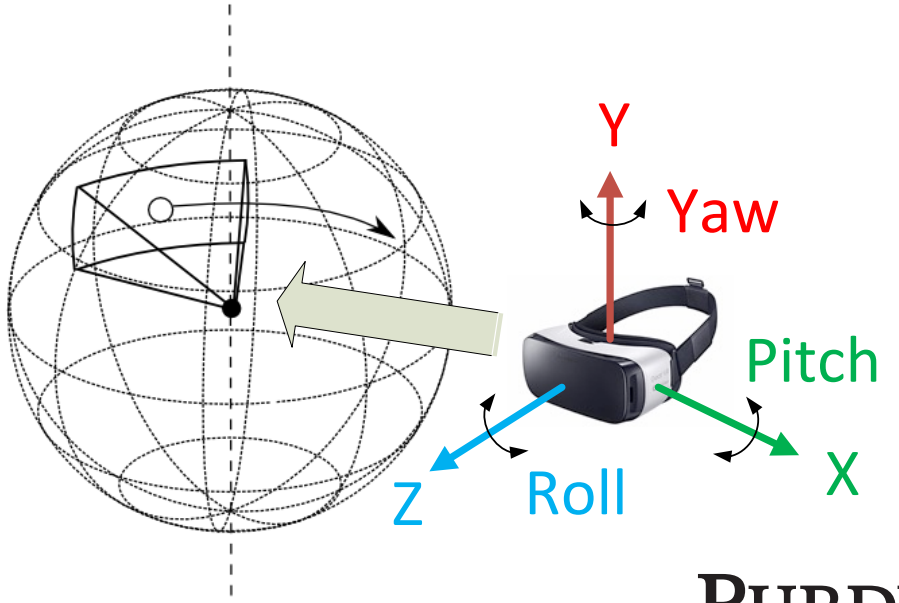
OUTLINE

- Video Streaming Algorithms
  - Single Path
  - Multiple Paths
  - Giant Client
- 360-degree Video Streaming
- Video Streaming over Cloud

360-DEGREE VIDEO STREAMING

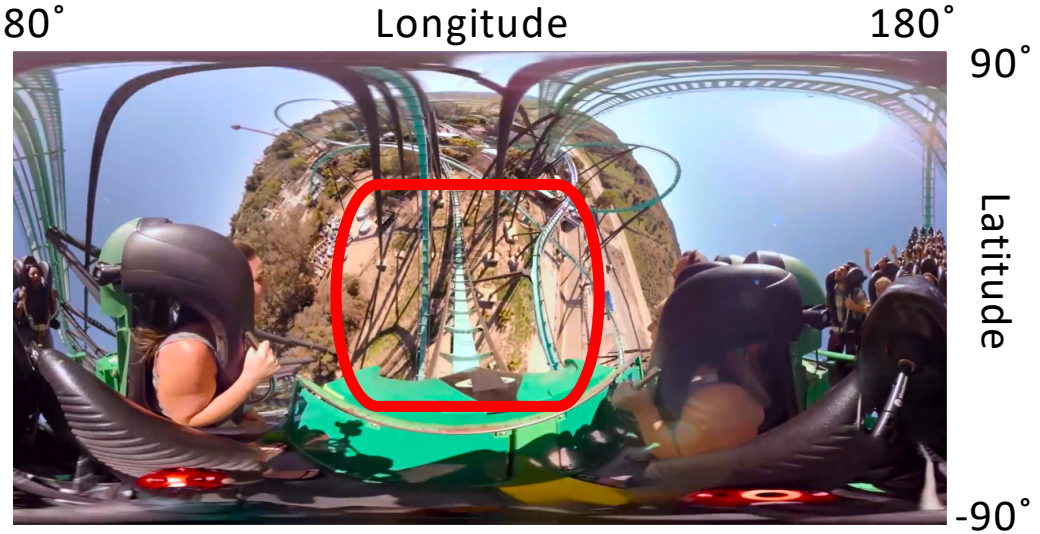
- A key application of Virtual Reality (VR)
- Provide users with panoramic views



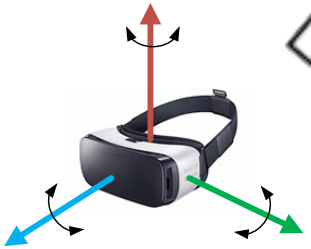


HOW ARE 360 VIDEOS CAPTURED & PLAYED?

Omni-directional Camera



Raw Panoramic Frame



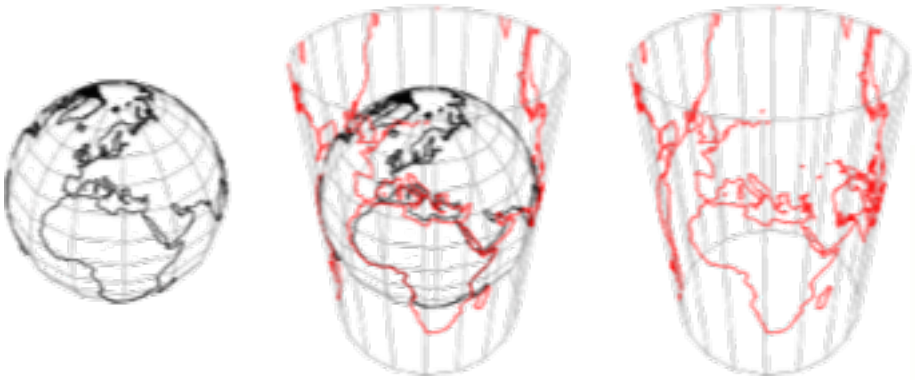
Viewing direction (lat/lon) &
Field-of-view (FoV) size

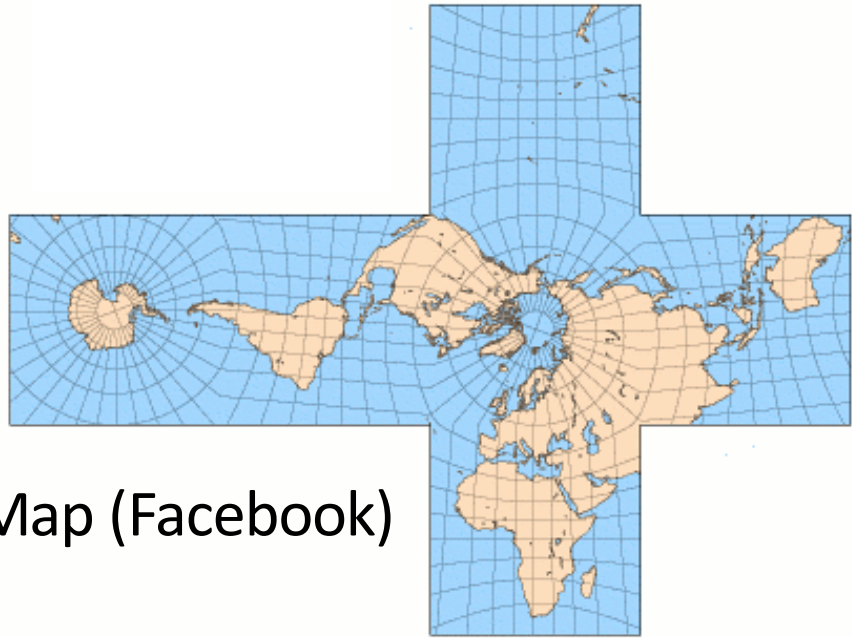
Project

Projected view in the viewport

PROJECTION ALGORITHM

Equirectangular (YouTube)





CubeMap (Facebook)

PURDUE UNIVERSITY

360-DEGREE VS REGULAR VIDEO STREAMING

360 videos inherit almost everything from regular videos

- Same encoding scheme
- Same rate adaptation algorithm
- Same client/server infrastructure
- Viewport agnostic from networking perspective: client downloads the entire raw panoramic frame regardless of user's viewport

360/VR videos

Regular video streaming techniques

Projection

PURDUE UNIVERSITY

360-DEGREE VS REGULAR VIDEO STREAMING

Tremendous bandwidth waste

- User only consumes ~15% of the downloaded contents

Under the same perceived quality, 360 videos are 4~6 times larger than regular videos.

- Roller coaster, 1'57", 4K, 250MB+

360/VR videos

Regular video streaming techniques

Projection

360-DEGREE VIDEO STREAMING

High-level idea: FoV-aware streaming!

- Predict user's head movement and only fetch portions that the user is about to see

Target commodity mobile devices & low-end VR headsets

Bare Smartphone

Google Cardboard, $10

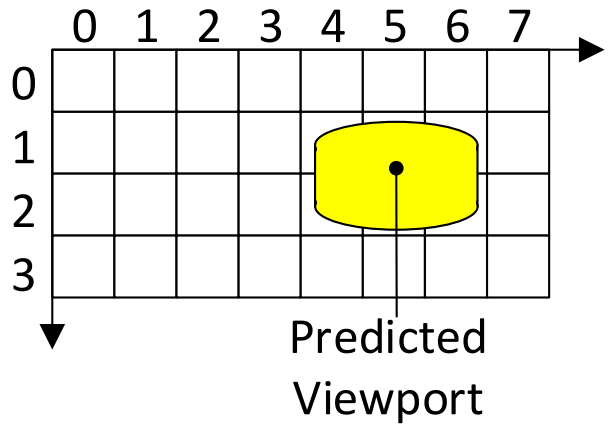
Samsung Gear VR, $80

Significantly reduce bandwidth utilization, or improve perceived quality

PURDUE UNIVERSITY

360-DEGREE VIDEO STREAMING

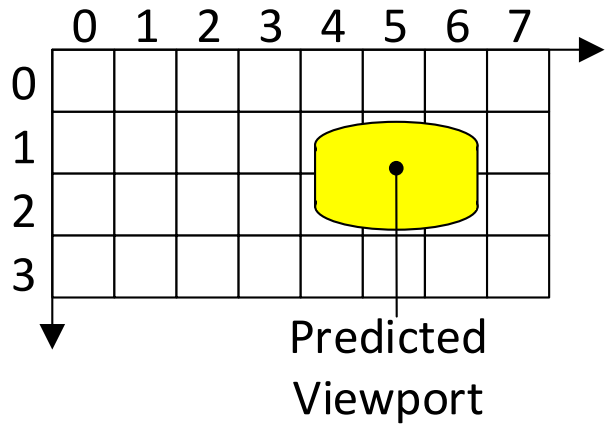
- A chunk is divided into tiles
- Efficient prediction of FoV can be obtained for some time-ahead
- Statistics of viewing each tile can be known
- How to account for FoV to obtain new streaming algorithm?



PURDUE UNIVERSITY

360-DEGREE VIDEO STREAMING

- What are the metrics?
  - Minimum rate in FoV
  - Average Rate in FoV
  - Stalls
  - Quality variations in FoV

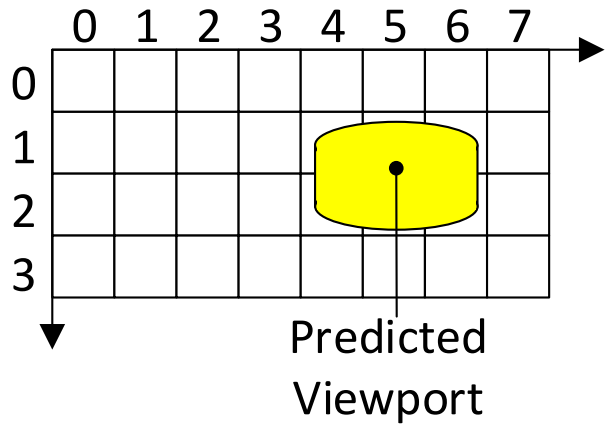


- What if prediction is wrong – how about stalls?

- We propose minimum quality for all tiles, so that we do not have to stall due to incorrect prediction

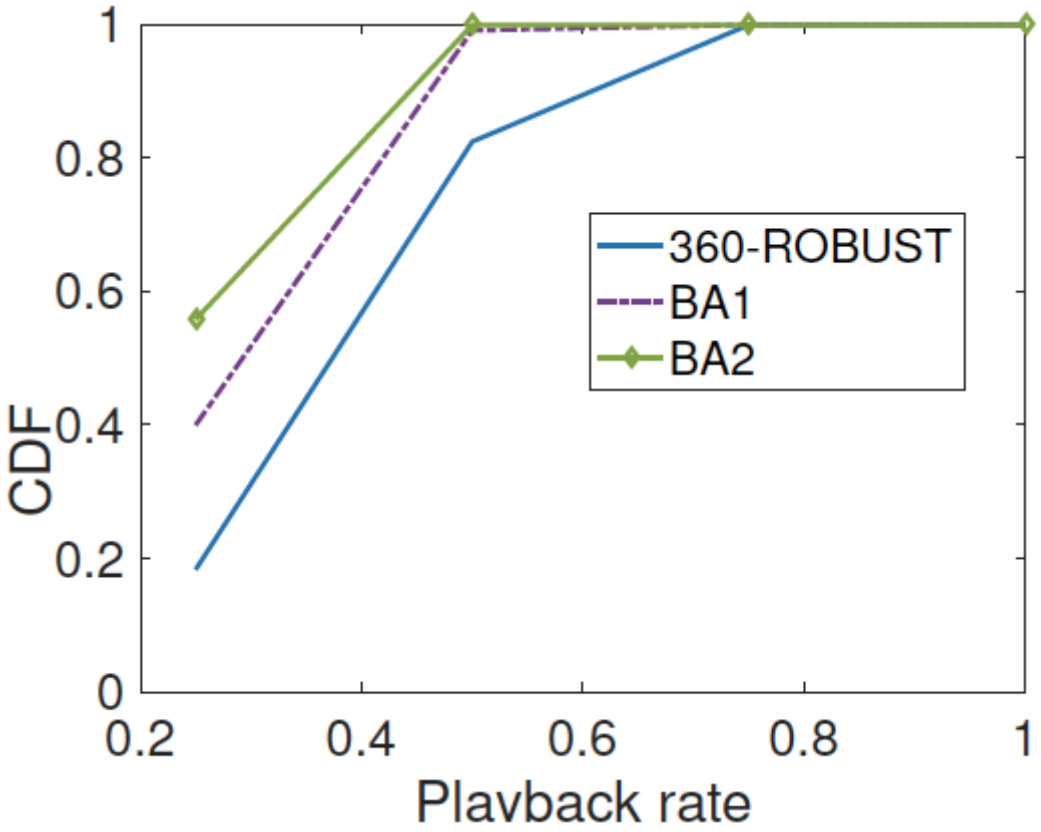
360-DEGREE VIDEO STREAMING

- What are the metrics?
  - Minimum rate in FoV
  - Average Rate in FoV
  - Stalls
  - Quality variations in FoV



- One metric: Choose 99th percentile area, and fetch higher quality in that, BL in rest
- Flexible: What quality for this bigger area?
- Akin to VBR – and efficient streaming strategies have been proposed.

RESULTS



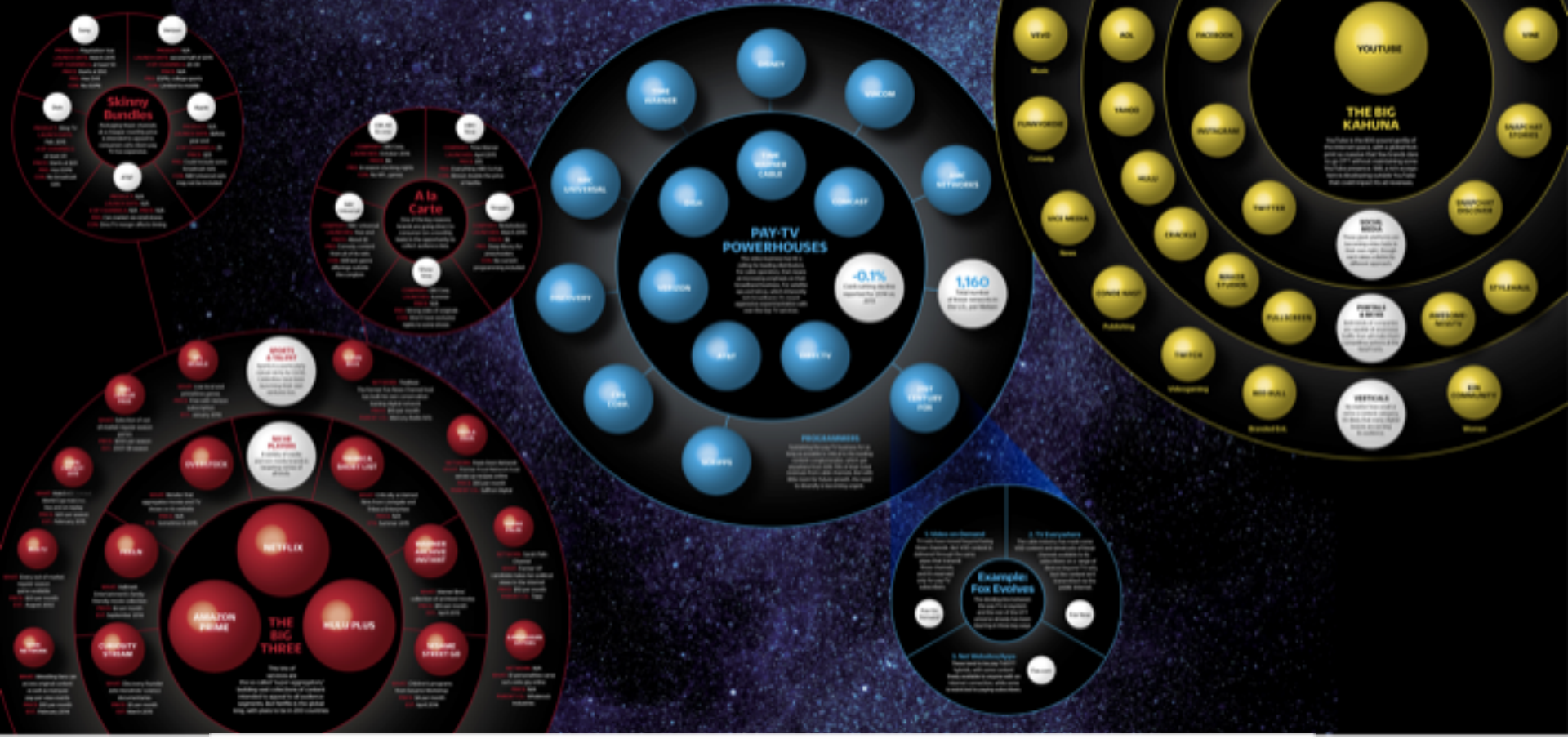
- Improves Playback rate and reduce stalls and quality variations as compared to approaches getting all tiles at same quality.

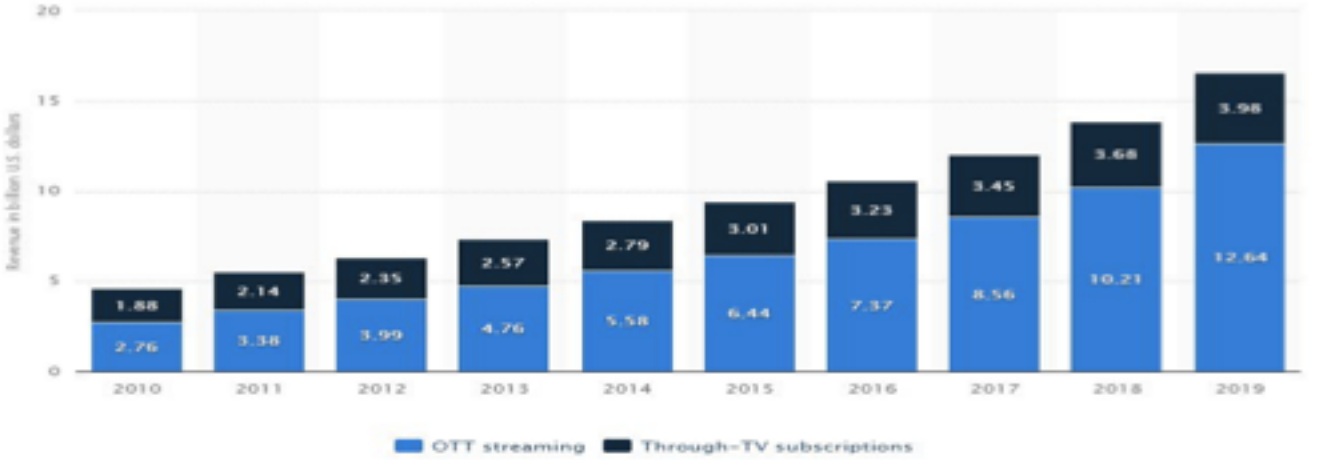
SUMMARY

- 360-degree adds a challenge of FoV prediction
- Exploiting the FoV prediction with tile-based streaming
- A new formulation is provided, and algorithms are proposed

OUTLINE

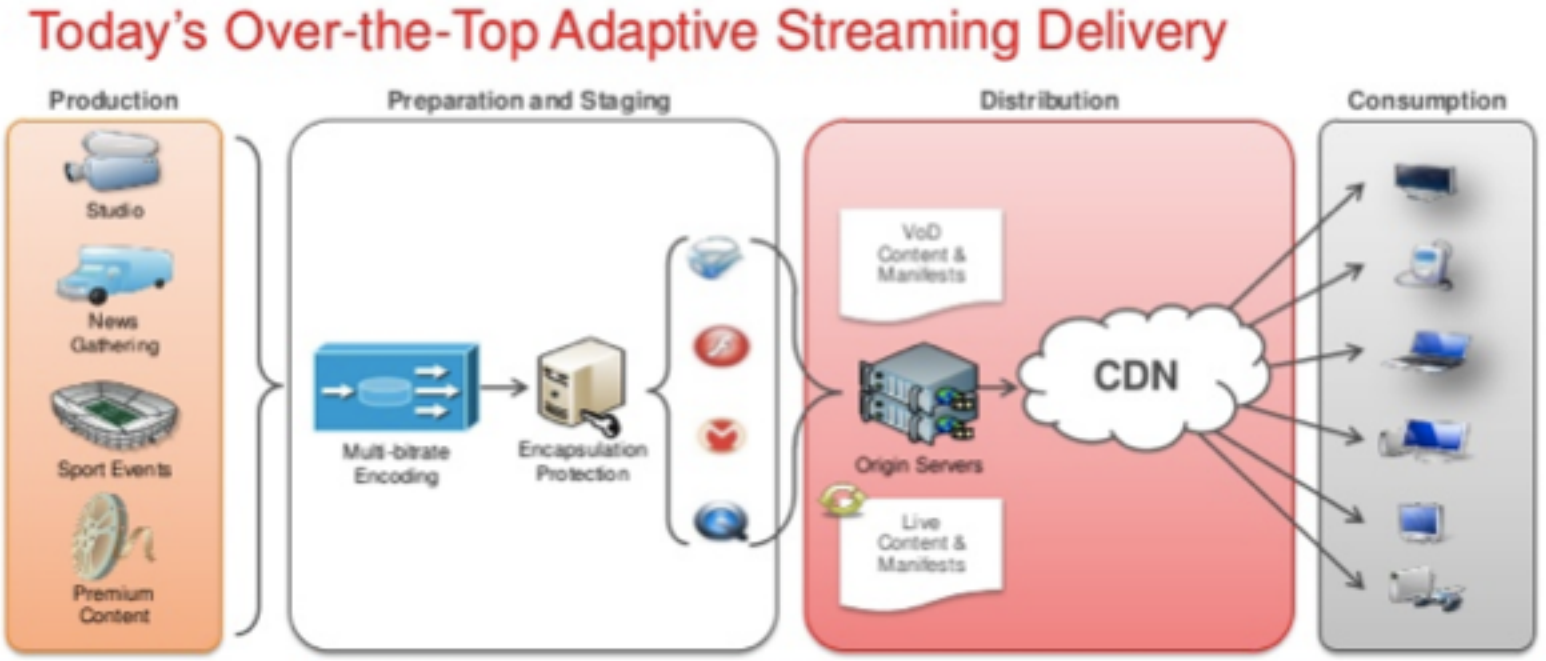
- Video Streaming Algorithms
  - Single Path
  - Multiple Paths
  - Giant Client
- 360-degree Video Streaming
- Video Streaming over Cloud


Skinny Bundles
A la Carte
PAY-TV POWERHOUSES
-0.1%
1,160
YOUTUBE
THE BIG KAHUNA
NETFLIX
AMAZON PRIME
HULU PLUS
THE BIG THREE
Example: Fox Evolves


20
15
10
5
0
revenue in billion U.S. dollars
1.88
2.76
2.14
3.38
2.35
3.99
2.57
4.76
2.79
5.58
3.01
6.44
3.23
7.37
3.45
8.56
3.68
10.21
3.98
12.64
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
OTT streaming
Through-TV subscriptions
© Statista 2016

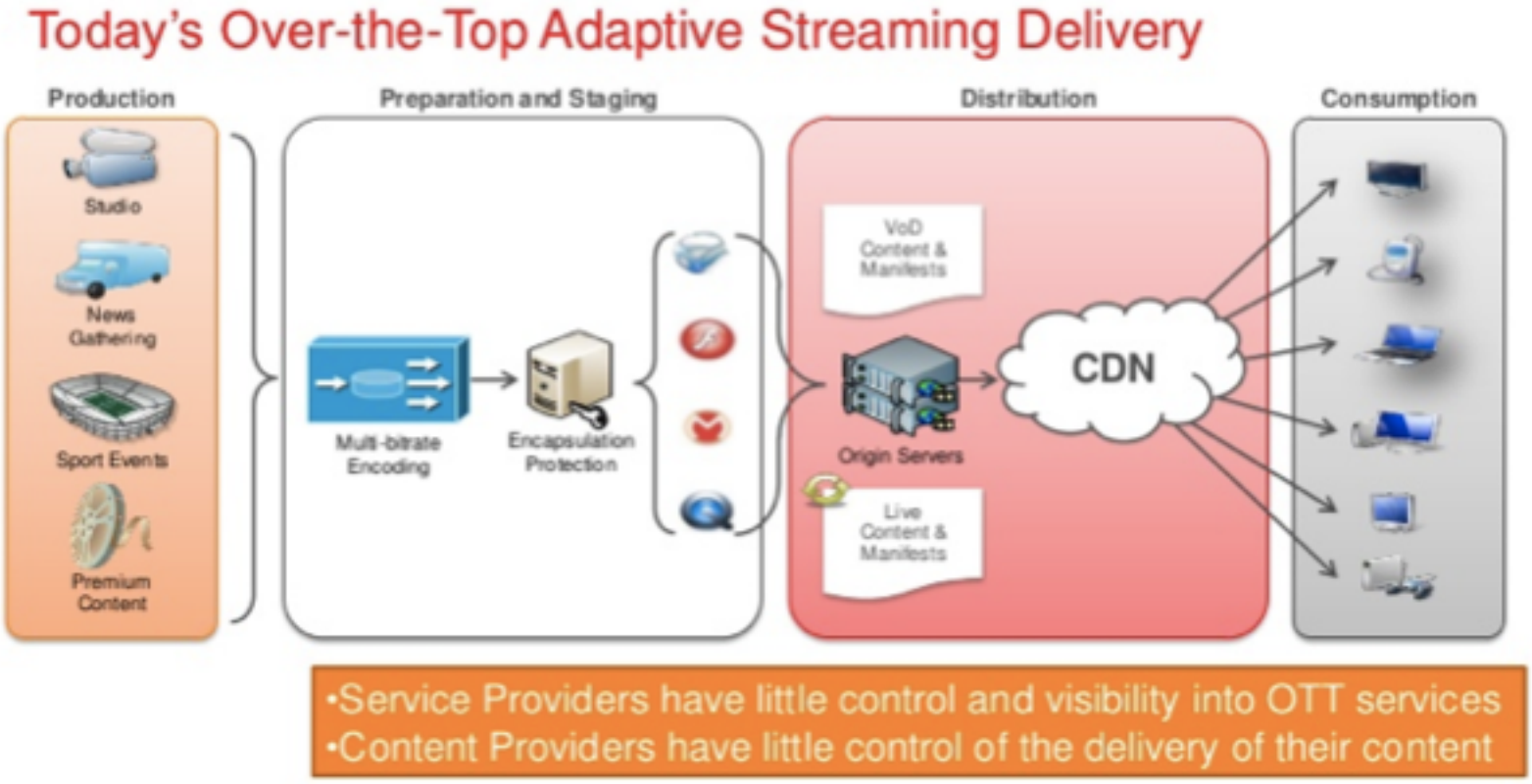
MOTIVATION

- Video streaming applications represents 62% of the Internet traffic in US
- More than 50% of over-the-top video traffic is now delivered through CDNs
- What are the key challenges?



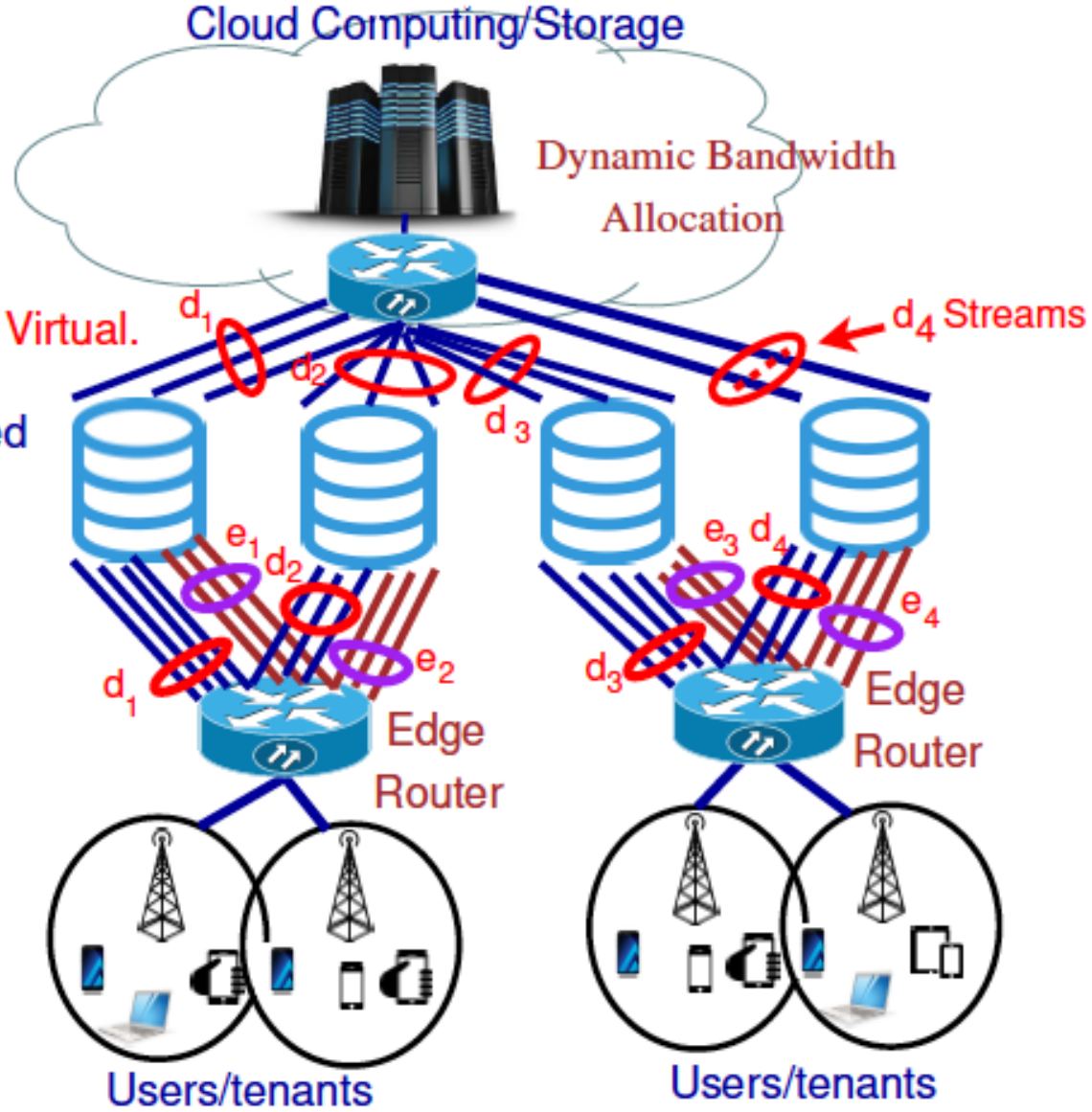
RESEARCH PROBLEMS

- Network Optimization: Determination of Caching, Bandwidths, and Server access for Optimized User QoE
- Computation Optimization: Multiple MapReduce Jobs for Video Processing and Data Analytics, optimized completions.
- Video Streaming Algorithms: Adaptive bit-rate algorithms for optimized user QoE

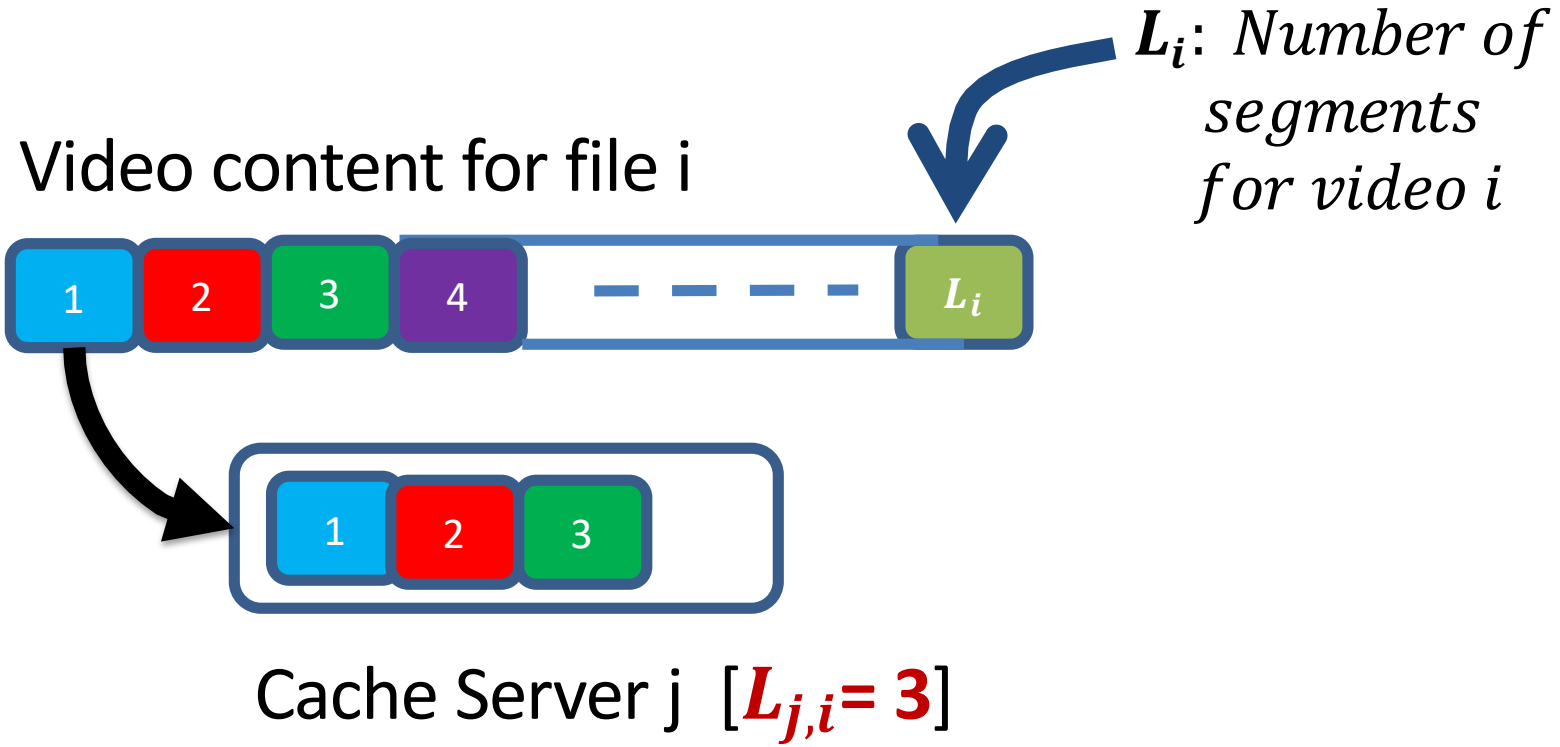


SETUP

- Part of each video is cached at cache servers.
- The content from cache is downloaded from one of e_j streams and that from the central cloud though d_j
- Each video request chooses one of the distributed server, and correspondingly one of d_j and one of e_j streams.



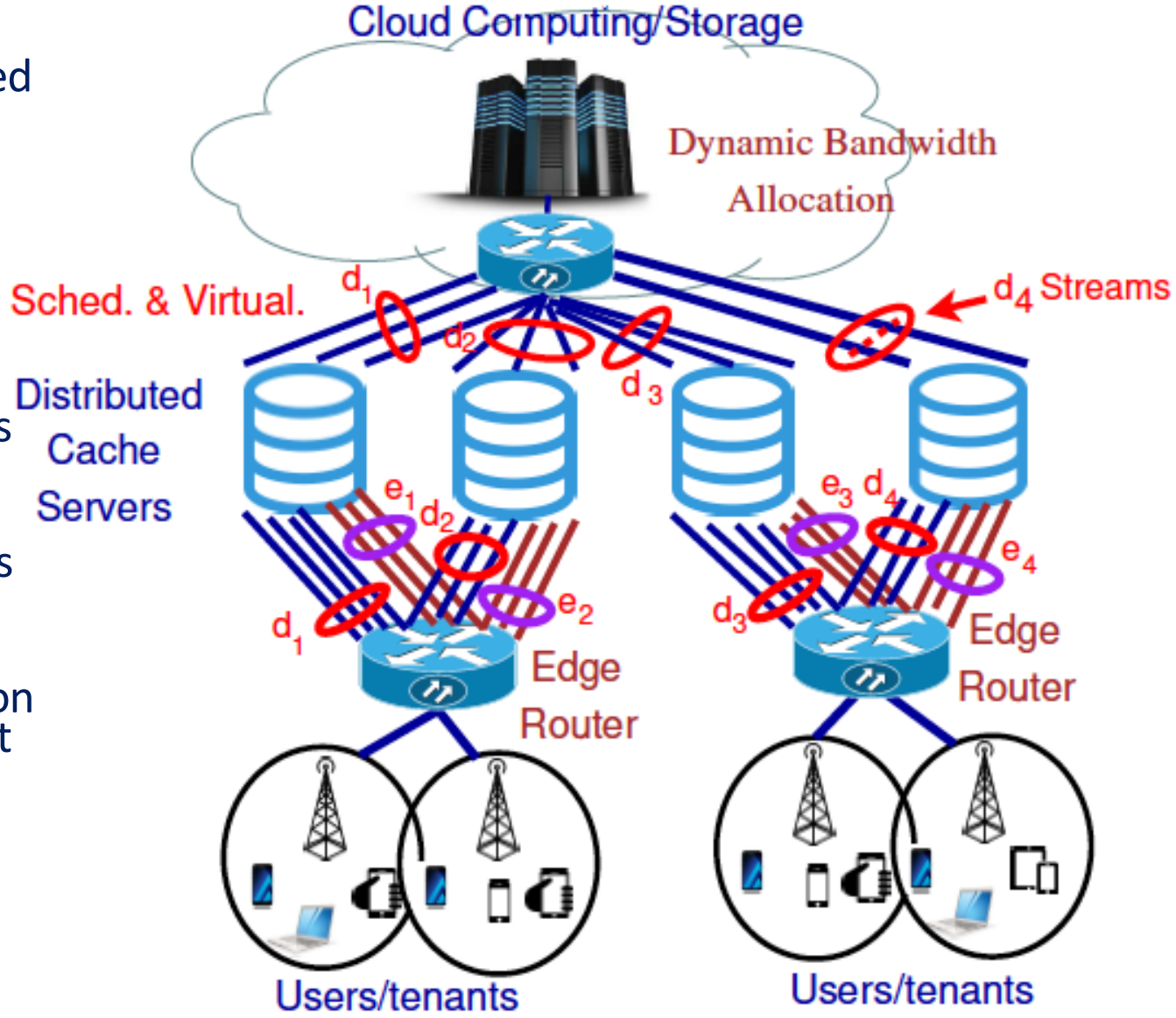
SETUP



- Part of each video is cached at cache servers.

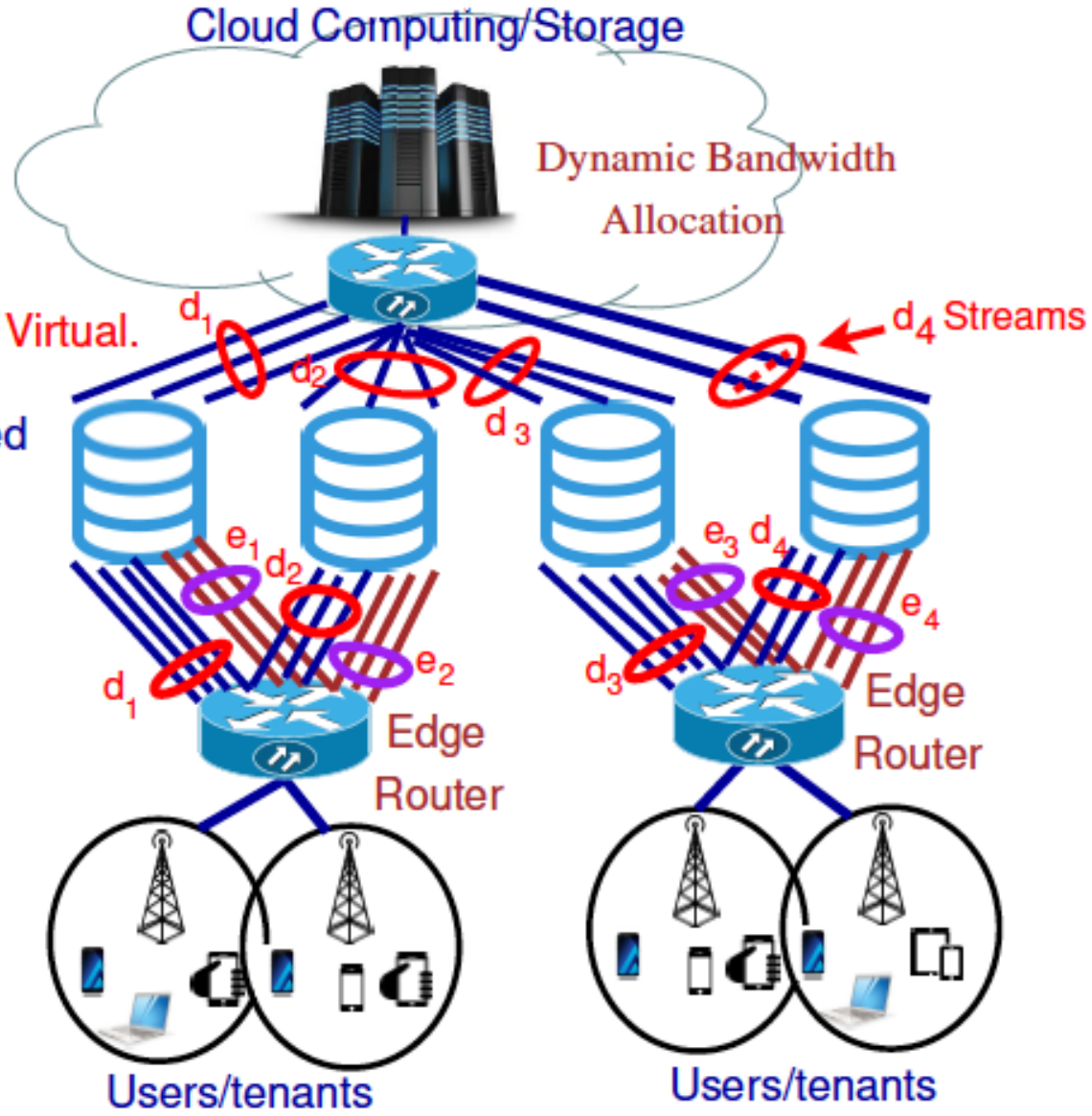
DECISION VARIABLES

- Choice of Distributed Server
- Amount of each video in cache
- Choice of e_j streams
- Choice of d_j streams
- Bandwidth allocation among the different streams



OBJECTIVE

- Wish to Minimize Stalls
- One of the key metric is to minimize the probability that the stalls are above a threshold.
- Metric is called Stall Duration Tail Probability



CHALLENGES

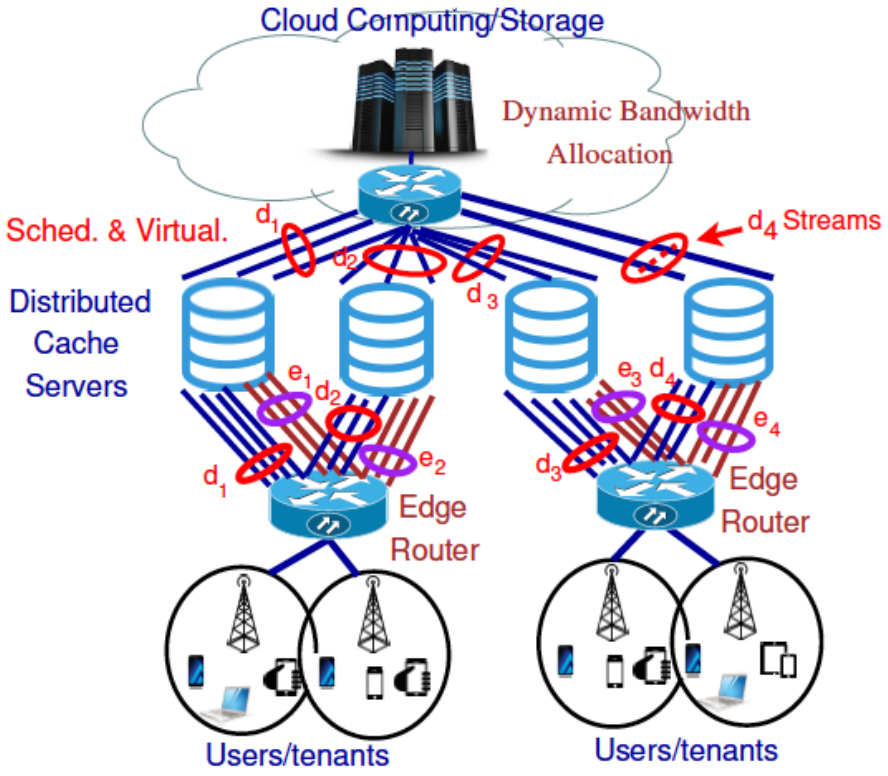
- Computation of objective is hard
- The key reason is that the choice of server, and the streams is a decision process, and the embedded computation of Markov Chain is challenging.
- Approach: Have the choice be independent of the queue state. [Proposed earlier for erasure-coded cloud storage by Aggarwal et al., Sigmetrics 2014, TON 2016.]
- Probabilistic Scheduling: Make the choice randomly with certain probabilities. However, the probabilities become design variables.
- Approach can give an upper bound for the objective

CHALLENGES

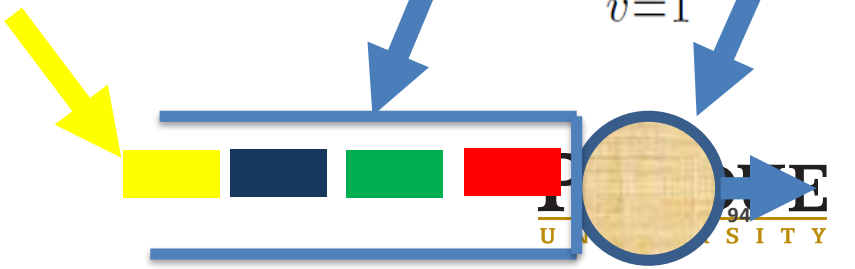
- Stall duration does not only depend on download of a chunk
- We need to compute download time of all chunks and stalls between chunks.
- Makes the metric harder to compute since download times are stochastic and the stall times are non-convex function of download times.
- Approach: Use efficient bounds.

STEPS FOR COMPUTATION

- For a file request for video i, assume that the servers, streams, and the cache contents at the server are known.
- **Step 1:** Download Time for Chunks from the Cache Servers:
- Download time of first chunk only depends on the waiting time for previous videos, and the service time of 1st chunk.
- Download time of later chunks can be found by adding service times.

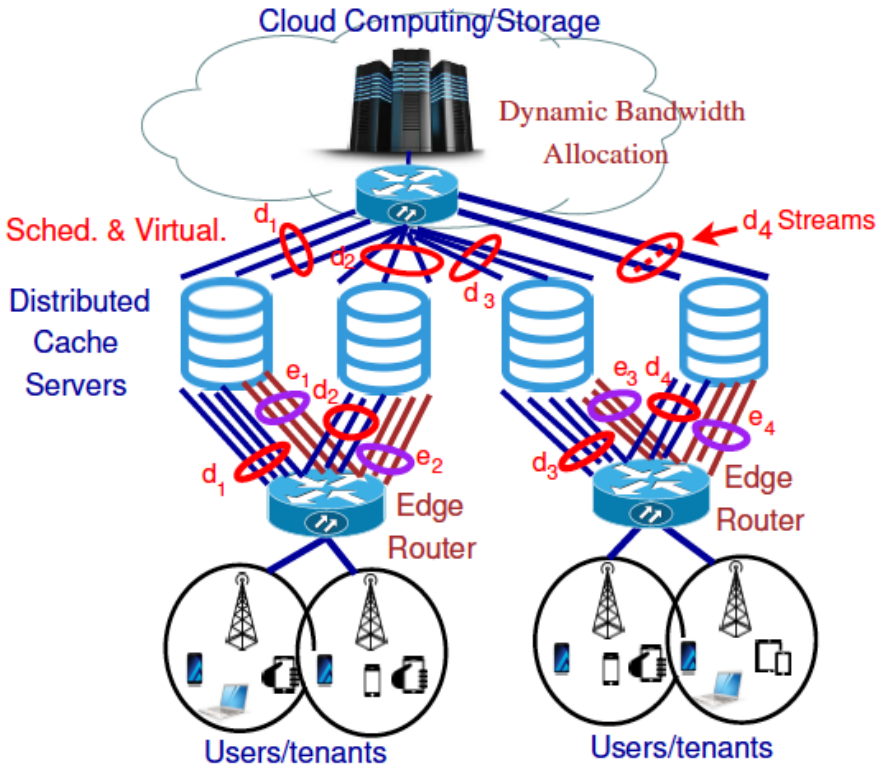


$$D^{(g)}_{i,j,\beta_j,\nu_j} = W^{(e)}_{j,\nu_j} + \sum_{v=1} Y^{(e,v)}_{i,j,\nu_j}$$



STEPS FOR COMPUTATION

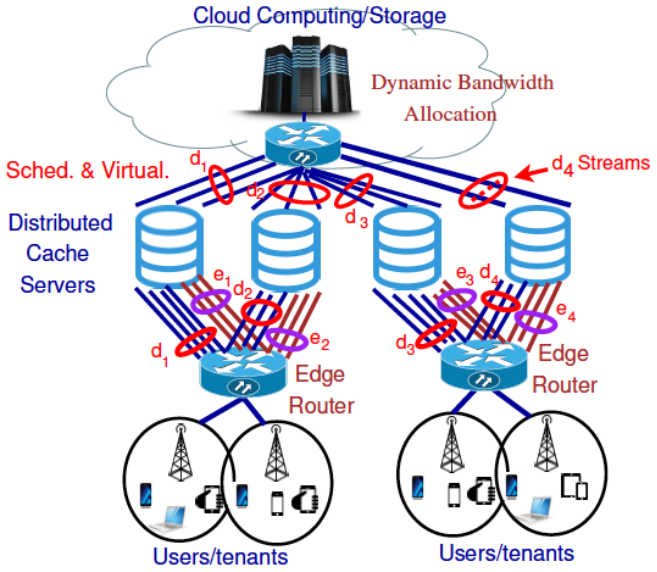
- For a file request for video i, assume that the servers, streams, and the cache contents at the server are known.
- **Step 2:** Download Time for Chunks from the Cloud Storage:
- This is challenging due to two queues. The chunk first has to come on top pipe, be downloaded, and then be enqueued in the bottom pipe.

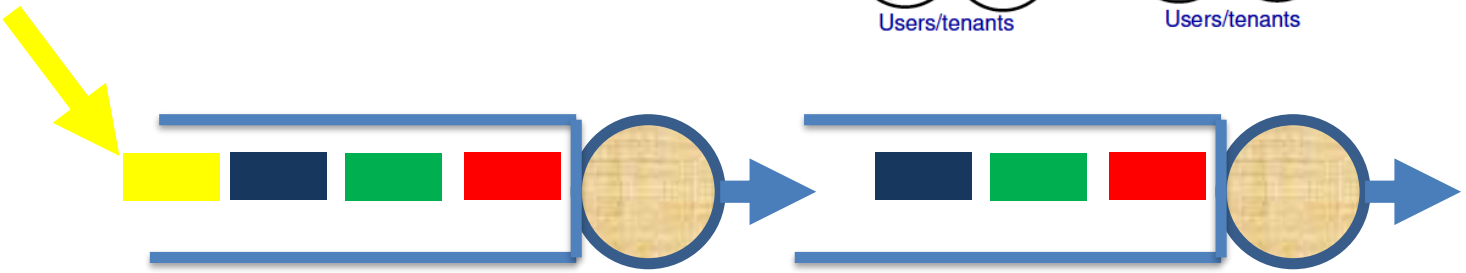


STEPS FOR COMPUTATION

- **Step 2:** Download Time for Chunks from the Cloud Storage:

- This is challenging due to two queues. The chunk first has to come on top pipe, be downloaded, and then be enqueued in the bottom pipe.





Datacenter

Cache Server

$$D_{i,j,\beta_j,\nu_j}^{(L_{j,i}+1)} = \max(W_{j,\beta_j}^{(c)}, E_{j,\beta_j}^{(L_{j,i}+1)}) + Y_{j,\beta_j}^{(c,L_{j,i}+1)}$$

$$D_{i,j,\beta_j,\nu_j}^{(g)} = \max(D_{i,j,\beta_j,\nu_j}^{(g-1)}, E_{i,j,\beta_j}^{(g)}) + Y_{j,\beta_j}^{(c,g)}$$

PURDUE UNIVERSITY

STEPS FOR COMPUTATION

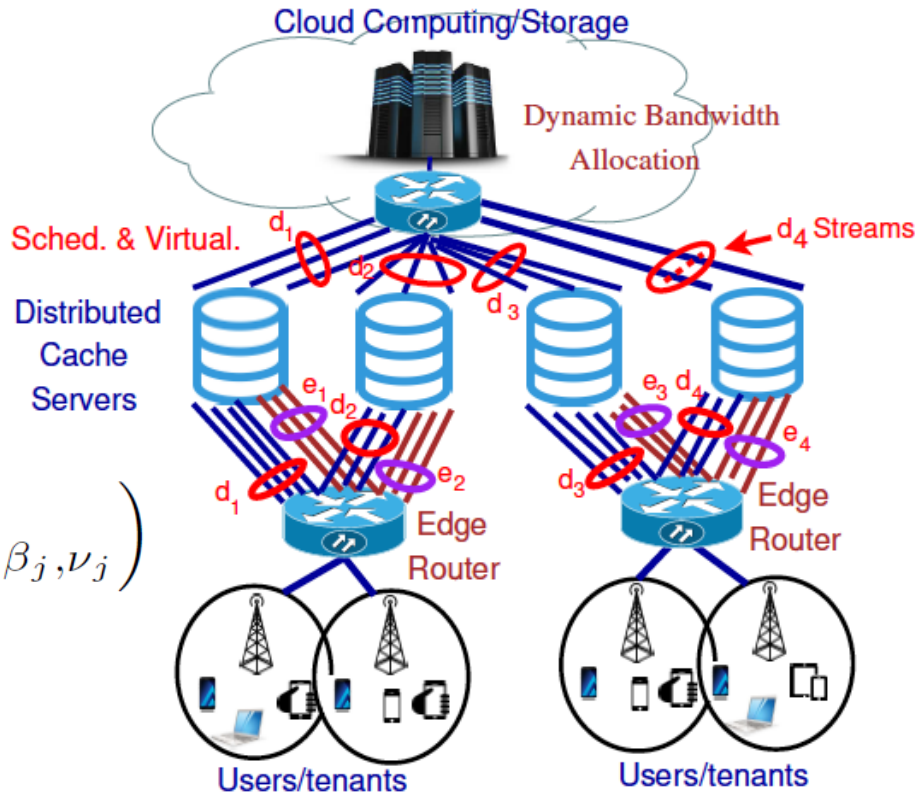
- **Step 3:** Play Time for Chunks from the Cloud Storage:

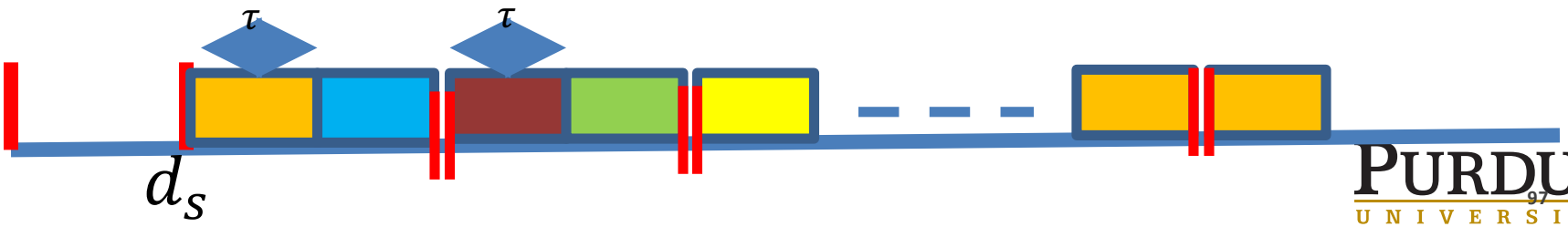
- Knowing download time of all chunks, the play times can be calculated.

$$T^{(1)}_{i,j,\beta_j,\nu_j} = \max\left(d_s,\ D^{(1)}_{i,j,\beta_j,\nu_j}\right)$$

$$T^{(q)}_{i,j,\beta_j,\nu_j} = \max\left(T^{(q-1)}_{i,j,\beta_j,\nu_j} + \tau,\ D^{(q)}_{i,j,\beta_j,\nu_j}\right)$$

- Thus, the statistics of Play times can be calculated.

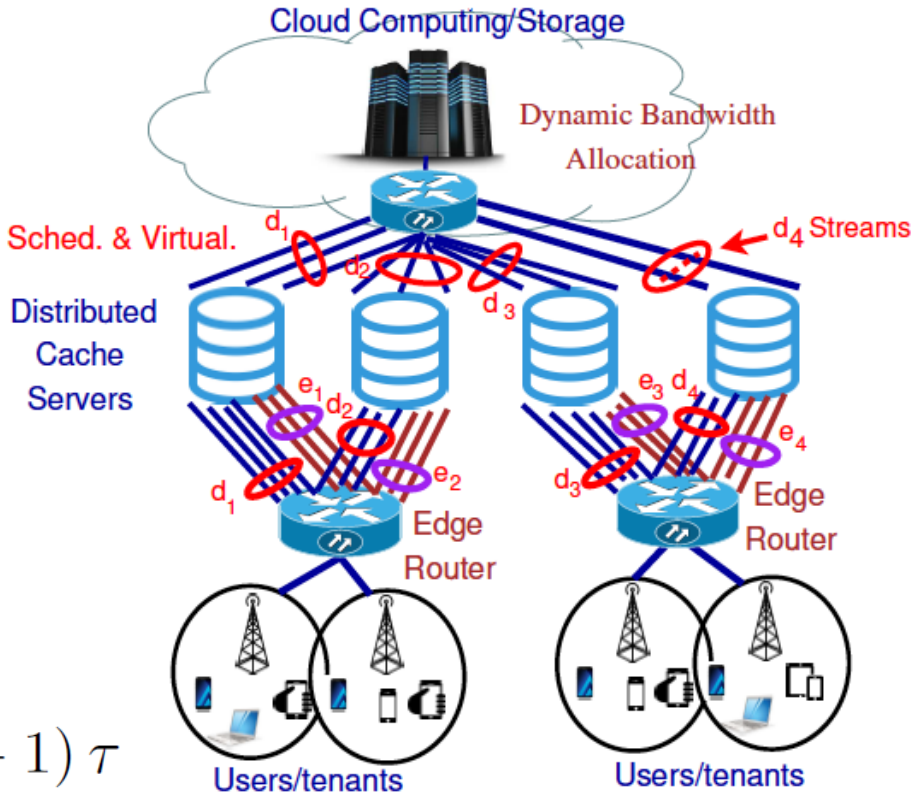




STEPS FOR COMPUTATION

- **Step 4:** Stall Duration:
- Stall duration can be calculated from the play time of the last chunk.
- Ideally, last chunk must be played at Startup Delay + (C-1) Chunk Duration
- The delay between the actual play time and the ideal time is the stall duration.

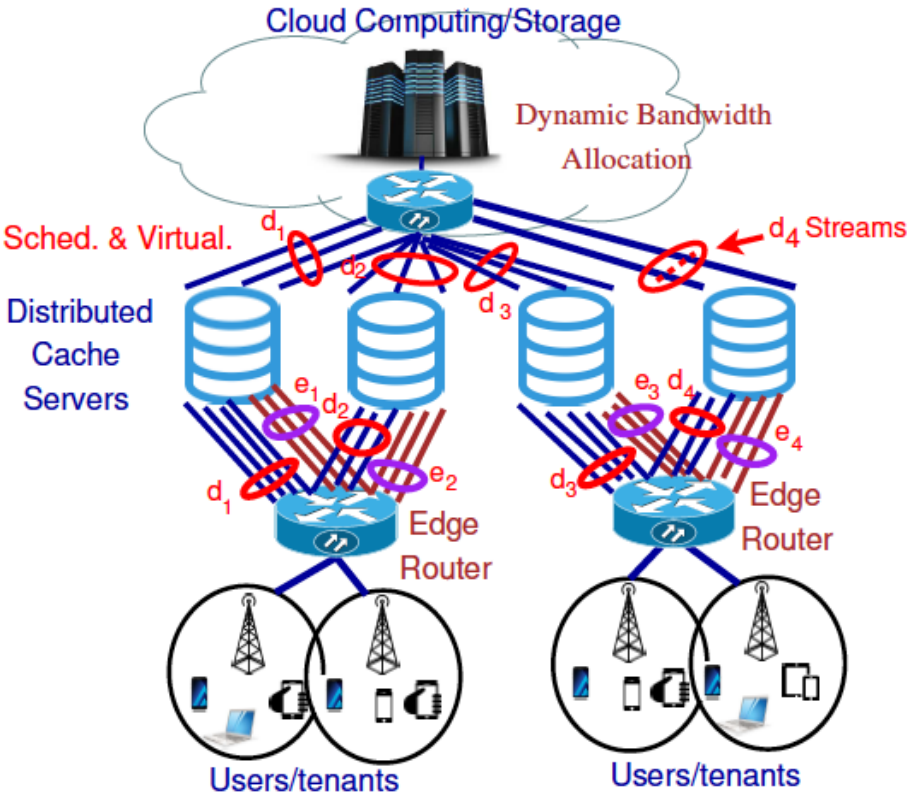
$$\Gamma^{(i,j,\beta_j,\nu_j)} = T^{(L_i)}_{i,j,\beta_j,\nu_j} - d_s - (L_i - 1)\,\tau$$



STEPS FOR COMPUTATION

- **Step 5:** Stall Duration Tail Probability:

- Knowing the moment generating function of the stall duration, bounds on tail probability can be calculated, over the statistics of the probabilistic scheduling.



$$\Pr\left(\Gamma^{(i,j,\beta_j,\nu_j)} \geq \sigma\right) \overset{(a)}{=} \Pr\left(T^{(L_i)}_{i,j,\beta_j,\nu_j} \geq \sigma + d_s + (L_i - 1)\,\tau\right)$$

OPTIMIZATION

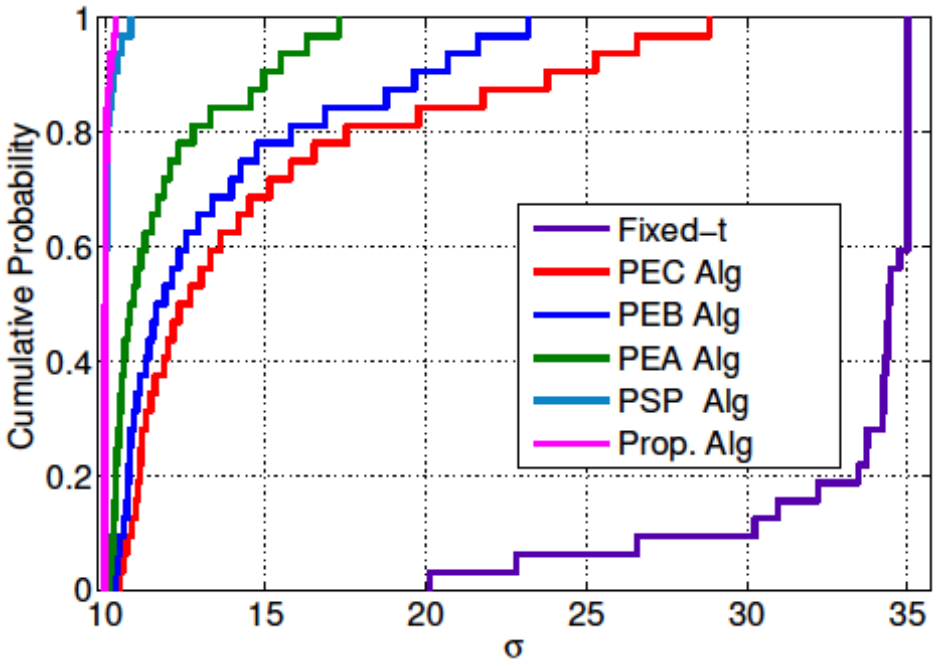
- We wish to optimize the stall duration tail probability, weighted for different video requests, to determine the choice of bandwidth allocation, caching, and the probabilistic scheduling.
- The problem is non-convex, has integer constraints.
- We use an alternating minimization based algorithm over different variable groups, and solve each variable group using iNner cOnVex Approximation (NOVA) algorithm.
- This guarantees convergence to a local optima.

EVALUATIONS

- Comparisons:
  - Projected Equal Server-PSs Scheduling, Optimized Auxiliary variables, Caching, and Bandwidth Weights (PEA)
  - Projected Equal Bandwidth, Optimized Access Servers and PS scheduling Probabilities, Auxiliary variables and cache placement (PEB)
  - Projected Equal Caching, Optimized Scheduling Probabilities, Auxiliary variables and Bandwidth Allocation Weights (PEC)
  - Fixed-t: Fixed Auxiliary Variables.
  - Projected Proportional Service-Rate, Optimized Auxiliary variables, Bandwidth Weights, and Cache Placement (PSP)

EVALUATIONS

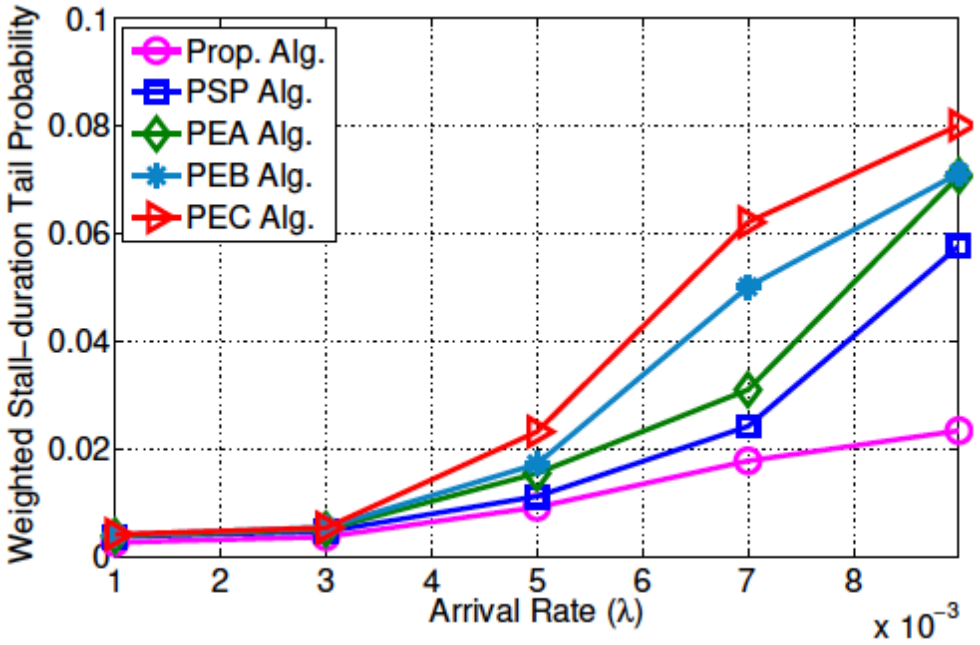
- Comparisons:
  - Projected Equal Server-PSs Scheduling, Optimized Auxiliary variables, Caching, and Bandwidth Weights (**PEA**)
  - Projected Equal Bandwidth, Optimized Access Servers and PS scheduling Probabilities, Auxiliary variables and cache placement (**PEB**)
  - Projected Equal Caching, Optimized Scheduling Probabilities, Auxiliary variables and Bandwidth Allocation Weights (**PEC**)
  - Fixed-t: Fixed Auxiliary Variables.
  - Projected Proportional Service-Rate, Optimized Auxiliary variables, Bandwidth Wights, and Cache Placement (**PSP**)



Better than Baselines

EVALUATIONS

- Comparisons:
  - Projected Equal Server-PSs Scheduling, Optimized Auxiliary variables, Caching, and Bandwidth Weights (**PEA**)
  - Projected Equal Bandwidth, Optimized Access Servers and PS scheduling Probabilities, Auxiliary variables and cache placement (**PEB**)
  - Projected Equal Caching, Optimized Scheduling Probabilities, Auxiliary variables and Bandwidth Allocation Weights (**PEC**)
  - Fixed-t: Fixed Auxiliary Variables.
  - Projected Proportional Service-Rate, Optimized Auxiliary variables, Bandwidth Wights, and Cache Placement (**PSP**)

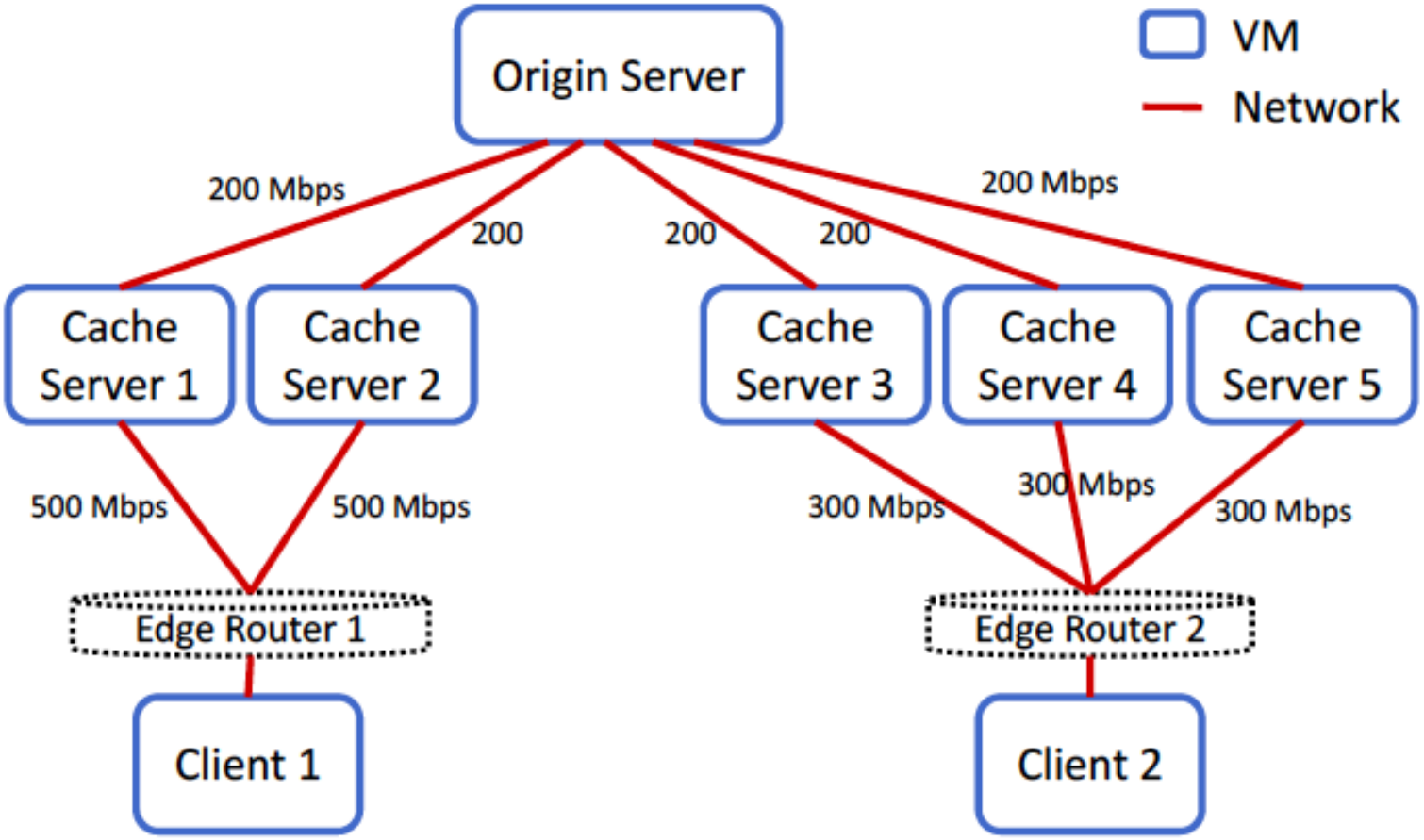


Benefit increases with load

PURDUE UNIVERSITY

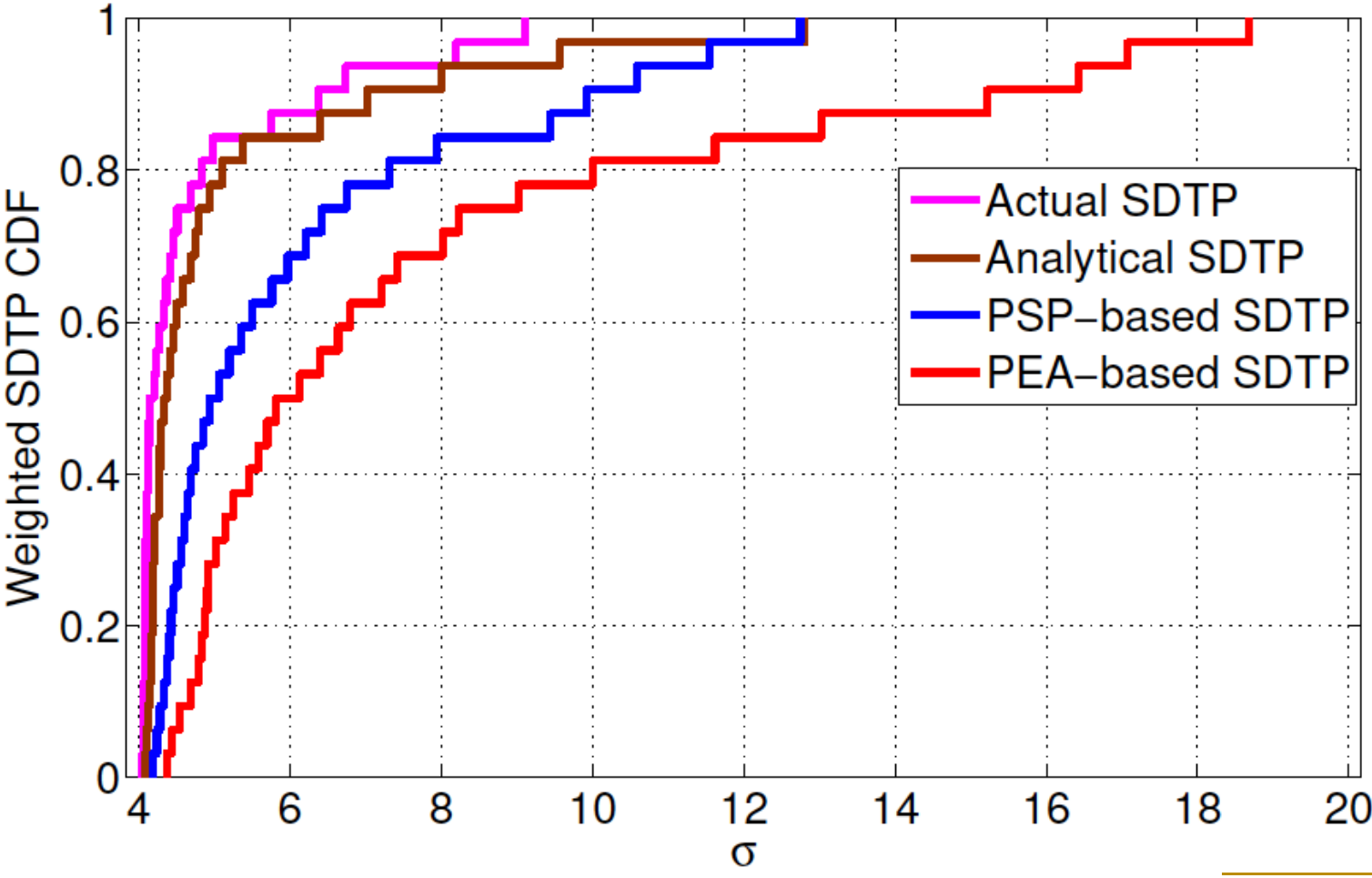
REAL IMPLEMENTATION

- The proposed algorithm was implemented on real servers.



REAL IMPLEMENTATION

- The proposed algorithm was implemented on real servers.



HOW ABOUT EDGE CACHE?

- Edge Router also has limited cache
- What is good caching policy at the edge router?

HOW ABOUT EDGE CACHE?

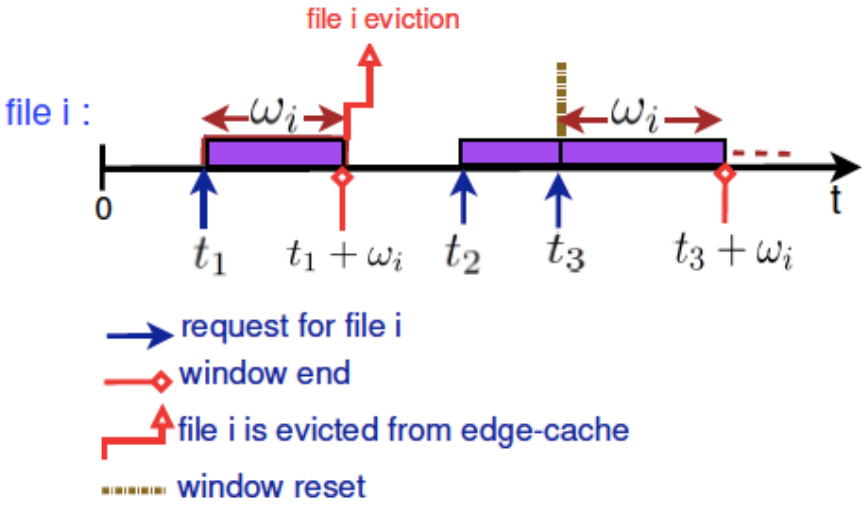
- Edge Router also has limited cache
- What is good caching policy at the edge router?
- LRU (Least Recently Used): Place last requested file in cache, remove the least recently used file.
- Edge cache can allow for multicast since the same content can be sen to another user even while being downloaded.

HOW ABOUT EDGE CACHE?

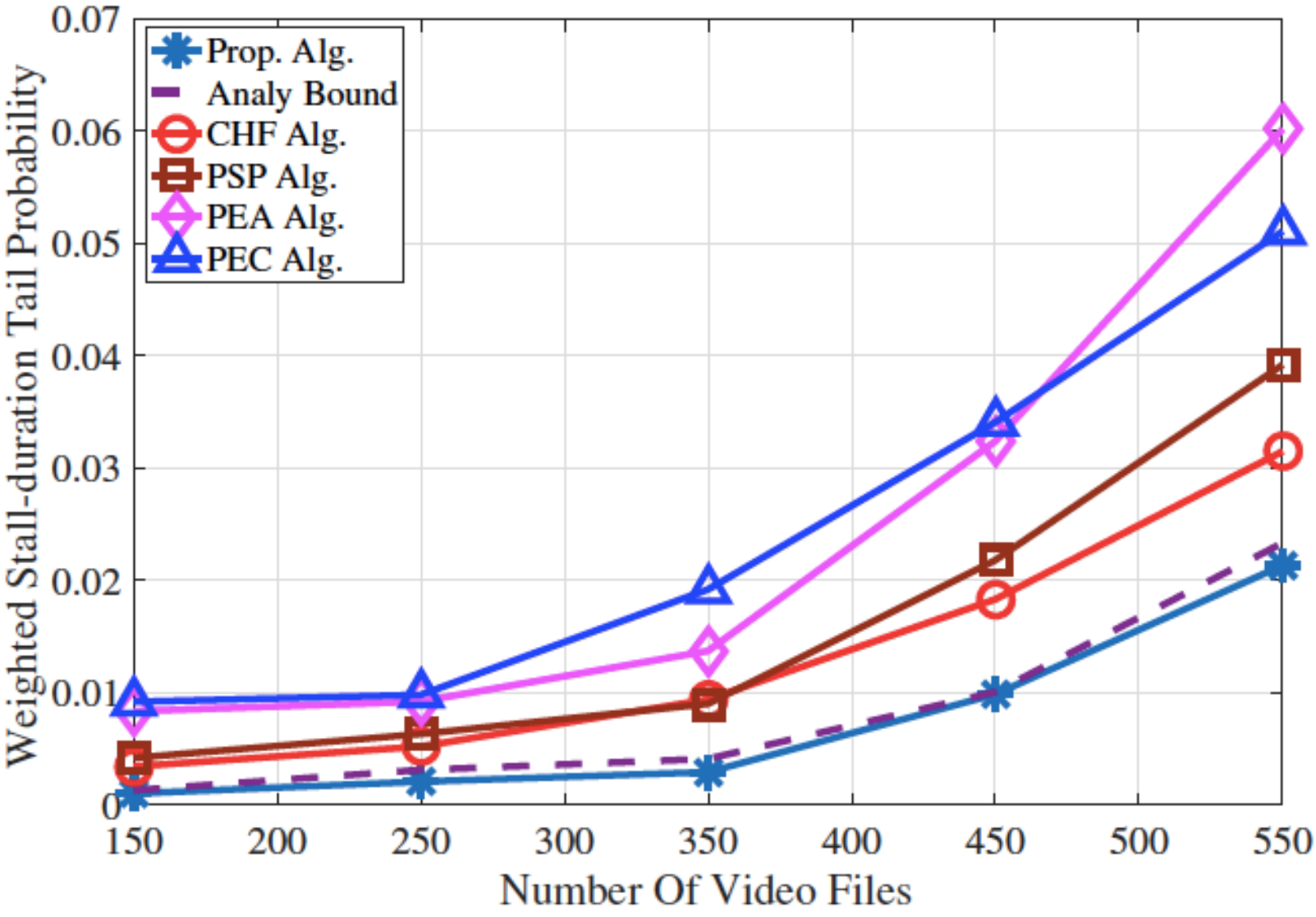
- Edge Router also has limited cache
- What is good caching policy at the edge router?
- LRU (Least Recently Used): Place last requested file in cache, remove the least recently used file.
- Edge cache can allow for multicast since the same content can be sen to another user even while being downloaded.
- LRU does not account for file priority, and is known to be bad for asymmetric file sizes. Thus, we consider TTL (time-to-live) where each file is removed after certain time (parameter) if not accessed.

CHANGES WITH TTL

- We consider TTL (time-to-live) where each file is removed after certain time (parameter) if not accessed.
- The cache has limited size, so parameters can be found to keep probability of cache over-size very small. Online adaptation can be done to manage such low probability violation.
- With this edge cache strategy and multicast, stall duration statistics can be calculated. The rest of the procedure is similar as described earlier.



IMPLEMENTATION RESULTS



DUE UNIVERSITY

SUMMARY

- CDN based video delivery system is considered
- Stall duration statistics are analyzed
- An optimized system is developed with knobs of scheduling, resource allocation, and caching

PURDUE UNIVERSITY

Thank You!

PURDUE UNIVERSITY