中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

Lumos: towards Better Video Streaming QoE through Accurate Throughput Prediction

Gerui Lv, Qinghua Wu, Weiran Wang, Zhenyu Li, Gaogang Xie

Institute of Computing Technology, Chinese Academy of Sciences
University of Chinese Academy of Sciences
Purple Mountain Laboratories
Computer Network Information Center, Chinese Academy of Sciences

Adaptive Bitrate (ABR) Streaming

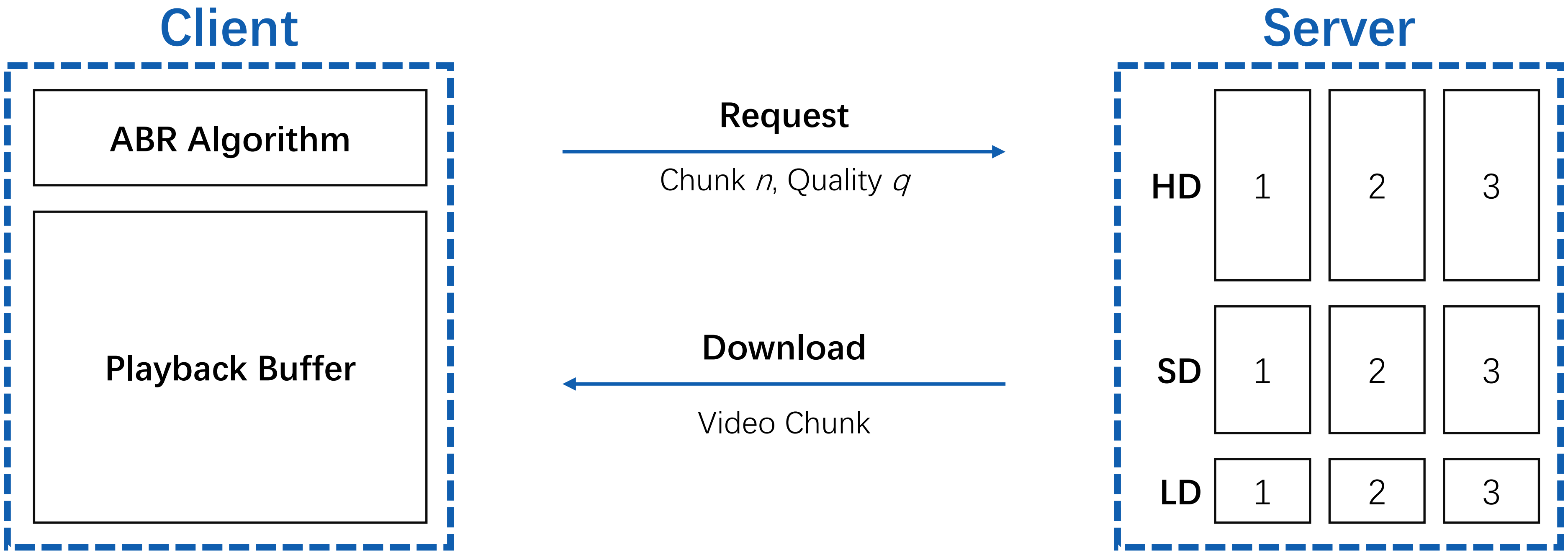
- HTTP-based video streaming dominates the Internet traffic nowadays, standardized as **DASH** (Dynamic Adaptive Streaming over HTTP)
- In DASH, the player runs **ABR** (Adaptive Bitrate) algorithm to select bitrate for each chunk, in order to optimize **QoE** (quality of experience)

$$QoE = \sum_{k=1}^{N} \boldsymbol{q(R_k)} - \mu \sum_{k=1}^{N} \mathbf{max}\left(\left(\frac{\boldsymbol{d_k(R_k)}}{\boldsymbol{T_k}} - \boldsymbol{B_k}\right), \mathbf{0}\right) - \lambda \sum_{k=1}^{N-1} \left|\boldsymbol{q(R_{k+1})} - \boldsymbol{q(R_k)}\right|$$

Quality	Rebuffering Time	Quality Switch

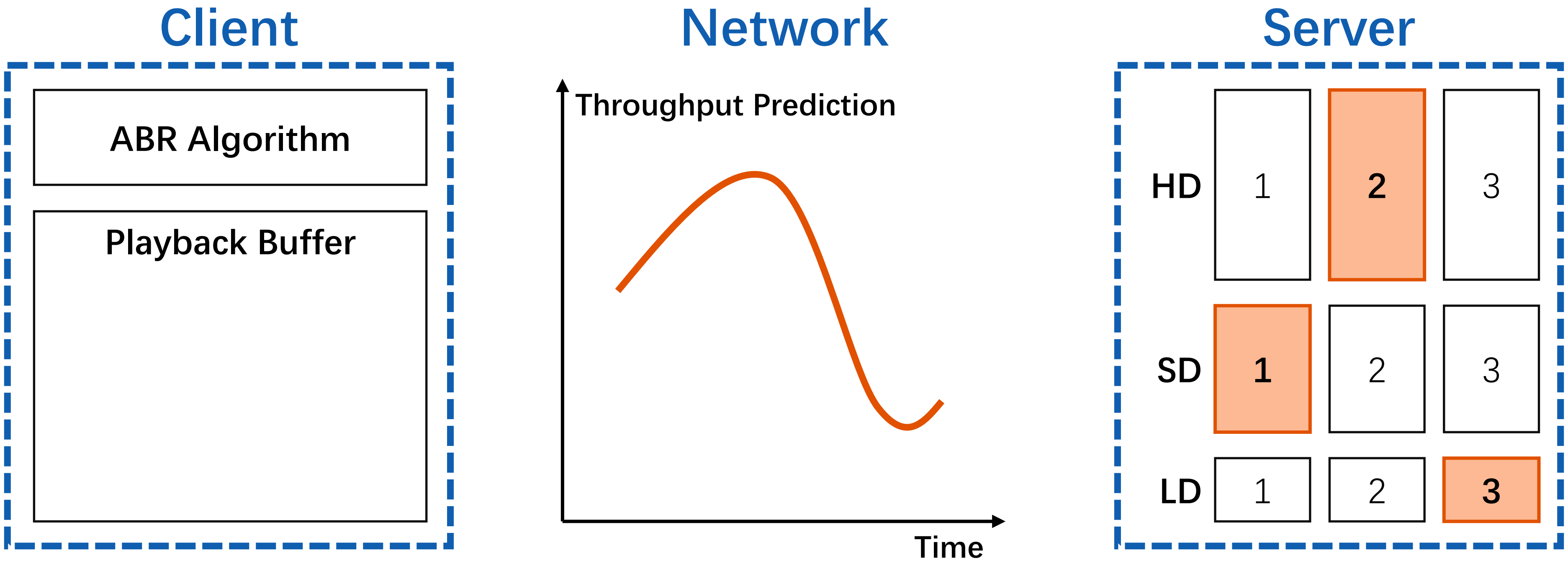
Adaptive Bitrate (ABR) Streaming

- DASH player runs **ABR** algorithm to optimize **QoE**
  - Goal: Higher quality; Lower rebuffering time; Fewer quality switches
  - Input: Application throughput, buffer occupancy, etc.
  - Output: Quality *q* (usually represented as bitrate level *r*) of chunk n



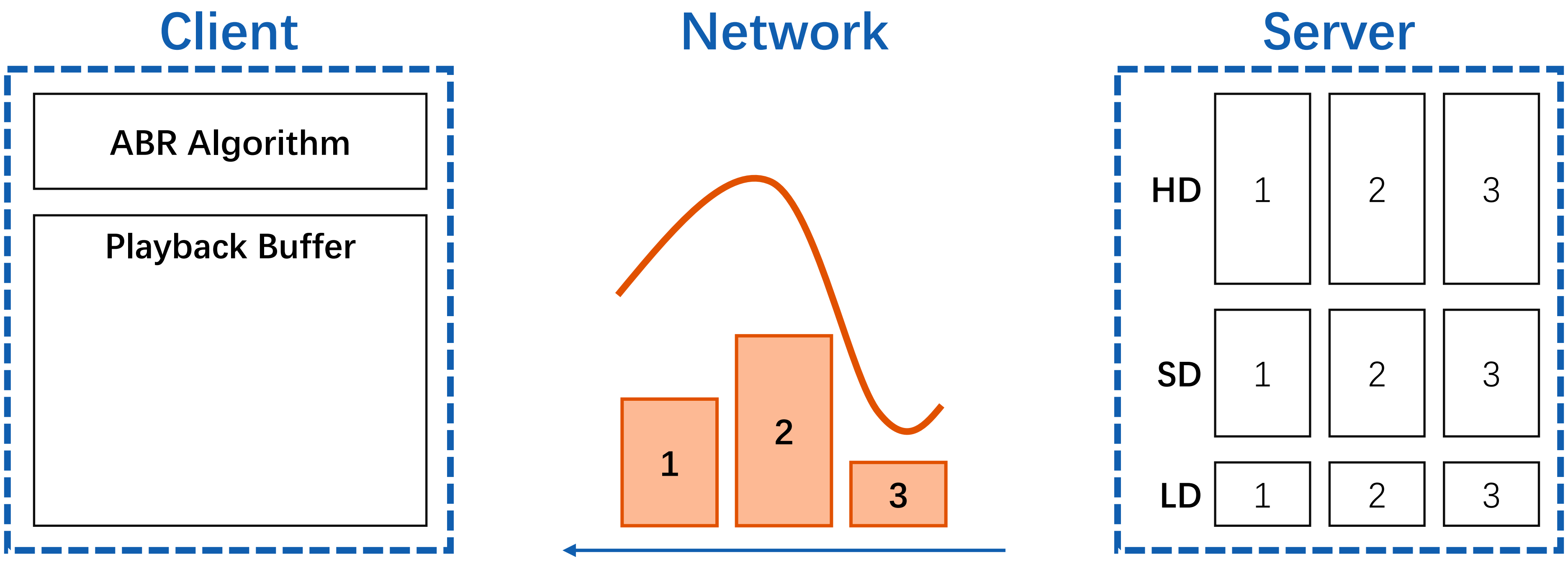
Adaptive Bitrate (ABR) Streaming

- DASH player runs **ABR** algorithm to optimize **QoE**
  - Goal: Higher quality; Lower rebuffering time; Fewer quality switches
  - Input: Application throughput, buffer occupancy, etc.
  - Output: Quality *q* (usually represented as bitrate level *r*) of chunk n



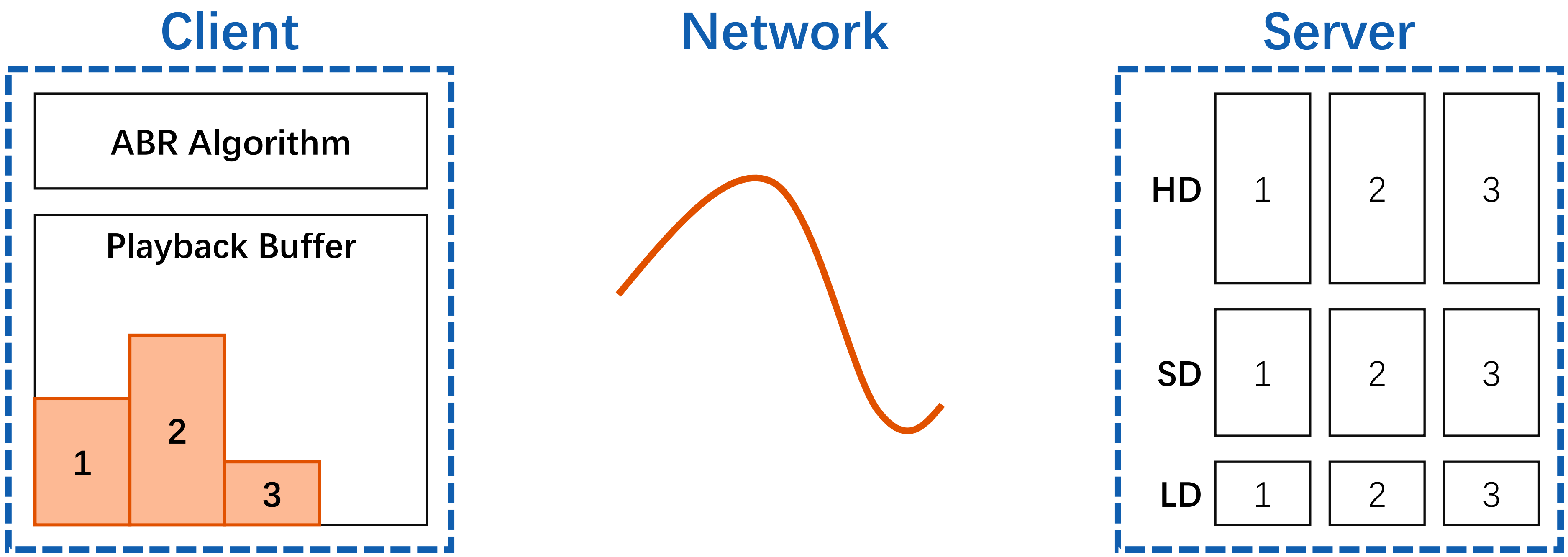
Adaptive Bitrate (ABR) Streaming

- DASH player runs **ABR** algorithm to optimize **QoE**
  - Goal: Higher quality; Lower rebuffering time; Fewer quality switches
  - Input: Application throughput, buffer occupancy, etc.
  - Output: Quality *q* (usually represented as bitrate level *r*) of chunk n



Adaptive Bitrate (ABR) Streaming

- DASH player runs **ABR** algorithm to optimize **QoE**
  - Goal: Higher quality; Lower rebuffering time; Fewer quality switches
  - Input: Application throughput, buffer occupancy, etc.
  - Output: Quality *q* (usually represented as bitrate level *r*) of chunk n



Throughput Prediction in ABR

❖ ABR algorithms can be classified into four categories

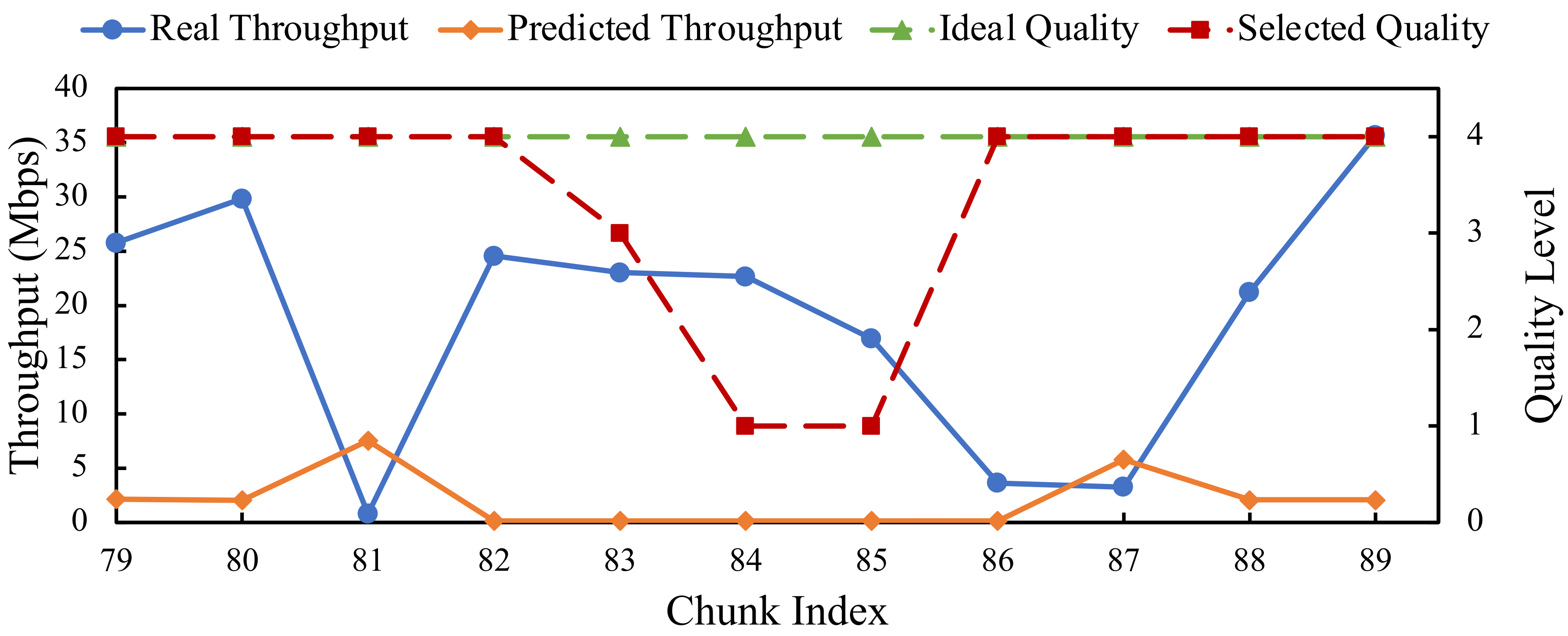
1 Rate-based	2 Buffer-based	3 Mixed	4 Learning-based
FESTIVE (CoNEXT'12) CS2P (SIGCOMM'16)	BBA (SIGCOMM'14) BOLA (INFOCOM'16)	MPC (SIGCOMM'15) PIA (INFOCOM'17) Fugu (NSDI'20)	Pensieve (SIGCOMM'17) Comyco (MM'19) Stick (INFOCOM'20)

- ①③: originally **require throughput prediction**
- ②④: **tend to rely on throughput prediction** when deployed in real-world environments
  - BOLA (INFOCOM'16) → DYNAMIC (MMSys'18)
  - Pensieve (SIGCOMM'17) → ABRL (ICML RL4RealLife'19)

Accurate throughput prediction is vital to improve QoE of ABR algorithms

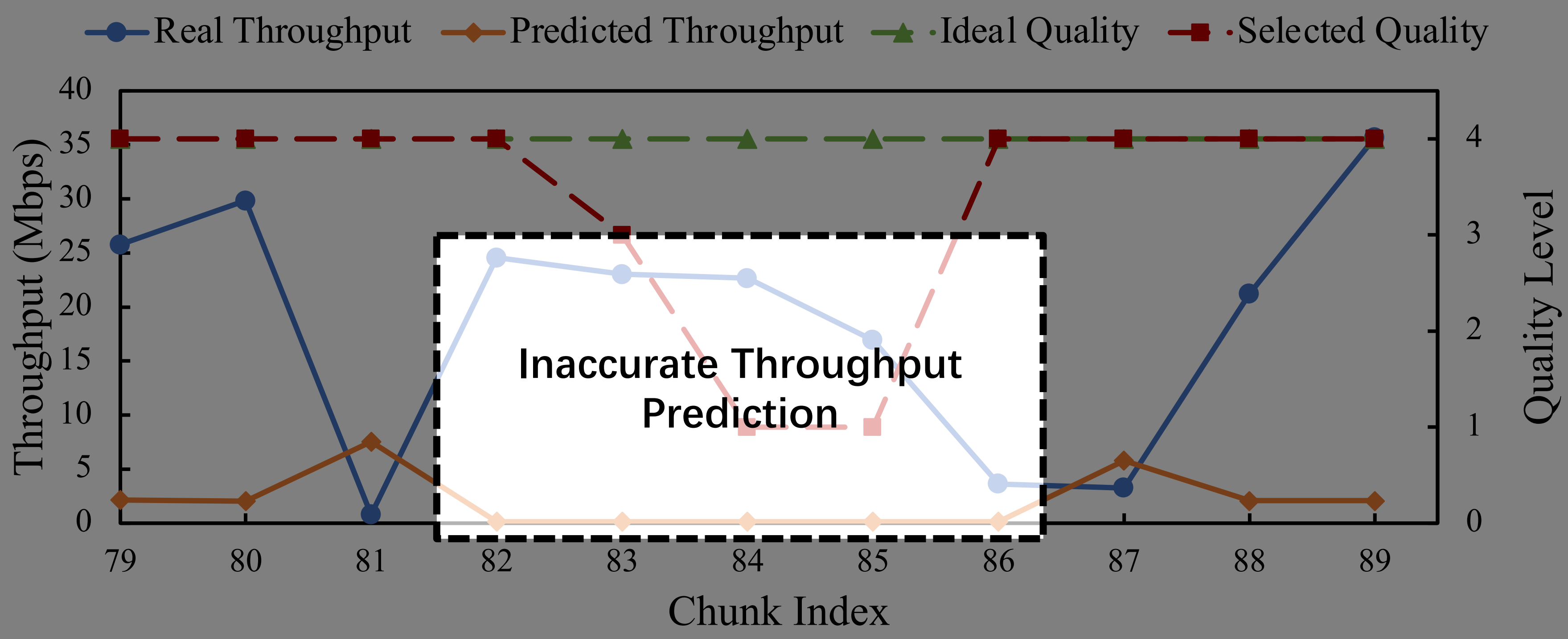
Harm of Inaccurate Prediction

- Inaccurate throughput prediction decreases QoE
  - A real case of **RobustMPC** (SIGCOMM'15)
  - Predicted Throughput = Harmonic Mean of past samples / (1 + max error rate)



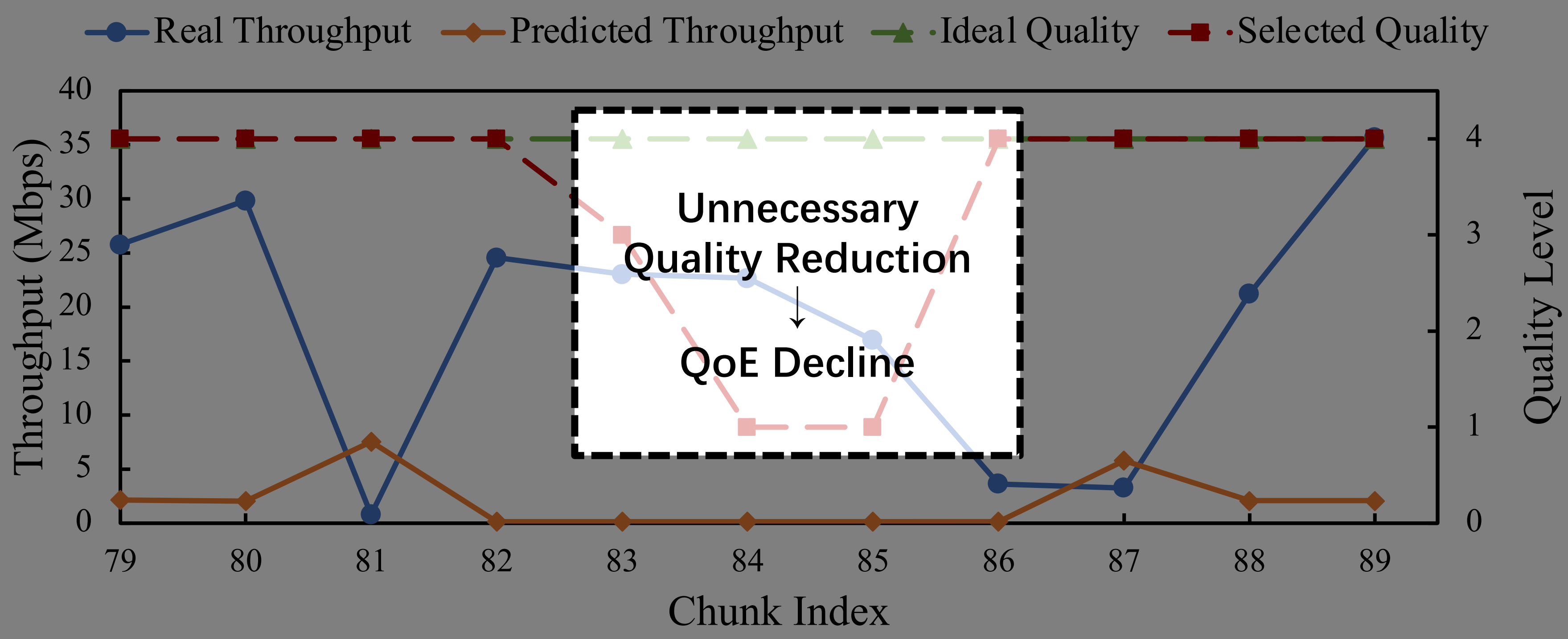
Harm of Inaccurate Prediction

- Inaccurate throughput prediction decreases QoE
  - A real case of **RobustMPC** (SIGCOMM'15)
  - Predicted Throughput = Harmonic Mean of past samples / (1 + max error rate)



Harm of Inaccurate Prediction

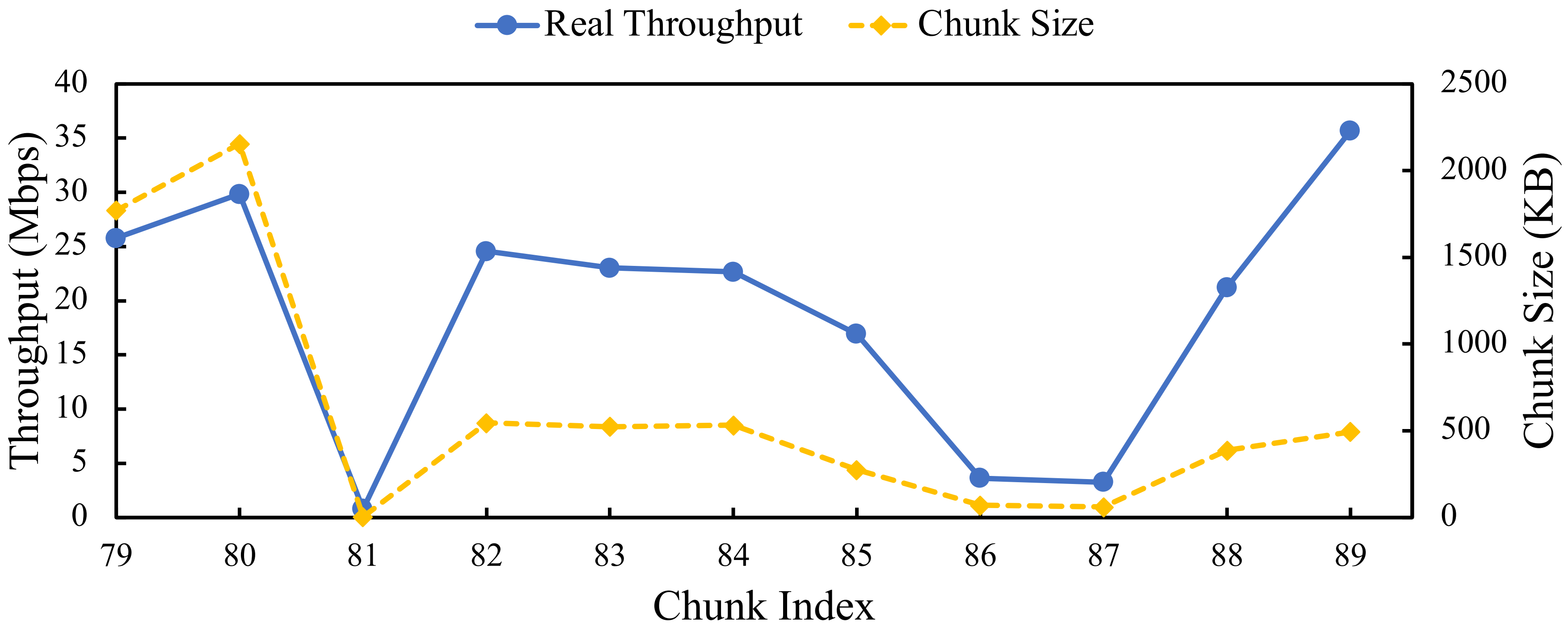
- Inaccurate throughput prediction decreases QoE
  - A real case of **RobustMPC** (SIGCOMM'15)
  - Predicted Throughput = Harmonic Mean of past samples / (1 + max error rate)



Dose Throughput Mean Network Conditions?

❖ Throughput Fluctuation

- Previous works attribute throughput fluctuation only to the change of **network conditions**
- However, it seems that **throughput changes in the same trend as chunk size does**



Chunk throughput may be not only determined by network conditions

Our Goal

Fundamental Problem in ABR streaming:

How to achieve accurate throughput prediction to assist ABR algorithms to optimize QoE?

Key Problems

- A predictor contains 3 components

$$Prediction = f(Features)$$

1

Input Features

What **factors** assist to achieve better prediction?

2

Output Target

Which one of **throughput** and **delivery time** is a better target to predict?

3

Mapping Function

Which **model** is more suitable for prediction?

Solution

- Correspondingly, our solution includes 3 steps

1	2	3
Identify Features	**Select Target & Model**	**Implement Predictor**
Qualitative Analysis Quantitative Verification	Controlled Experiment	Design Principle Real-world Deployment

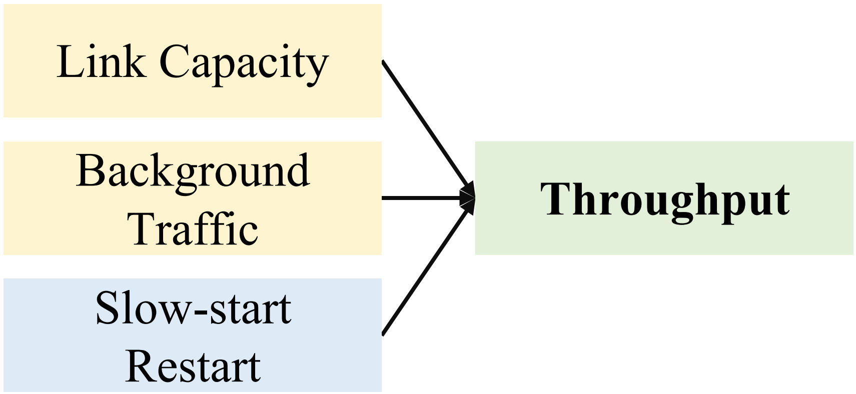
Solution

- Correspondingly, our solution includes 3 steps

1	2	3
Identify Features	**Select Target & Model**	**Implement Predictor**
Qualitative Analysis Quantitative Verification	Controlled Experiment	Design Principle Real-world Deployment

Identify Features

❖ Academic research evolves in this area

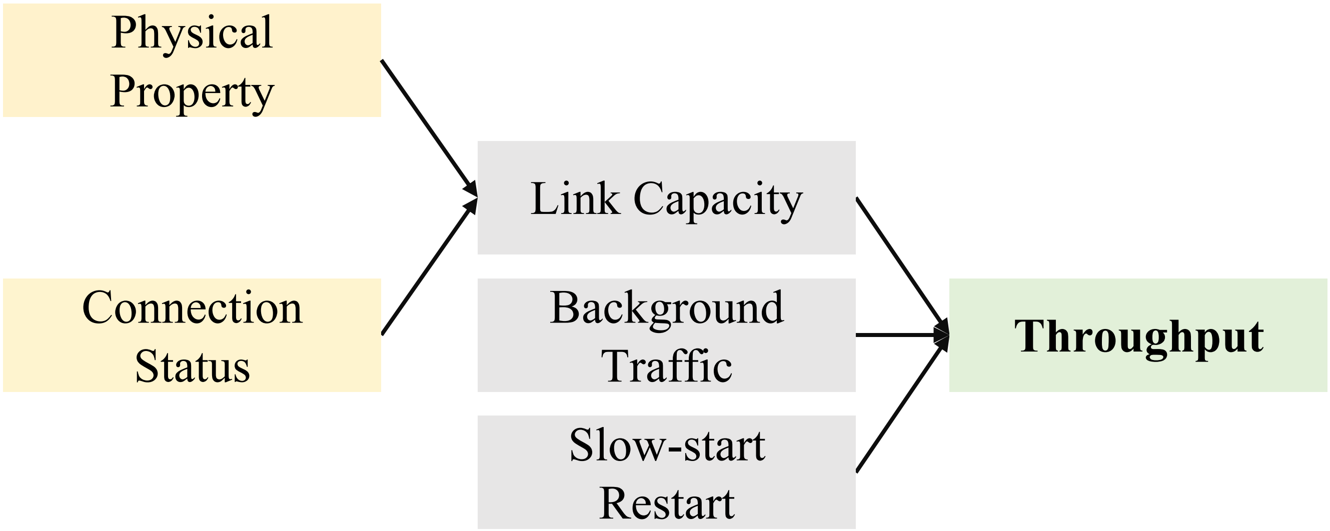


2012

Background traffic makes it hard to predict chunk throughput

Identify Features

❖ Academic research evolves in this area

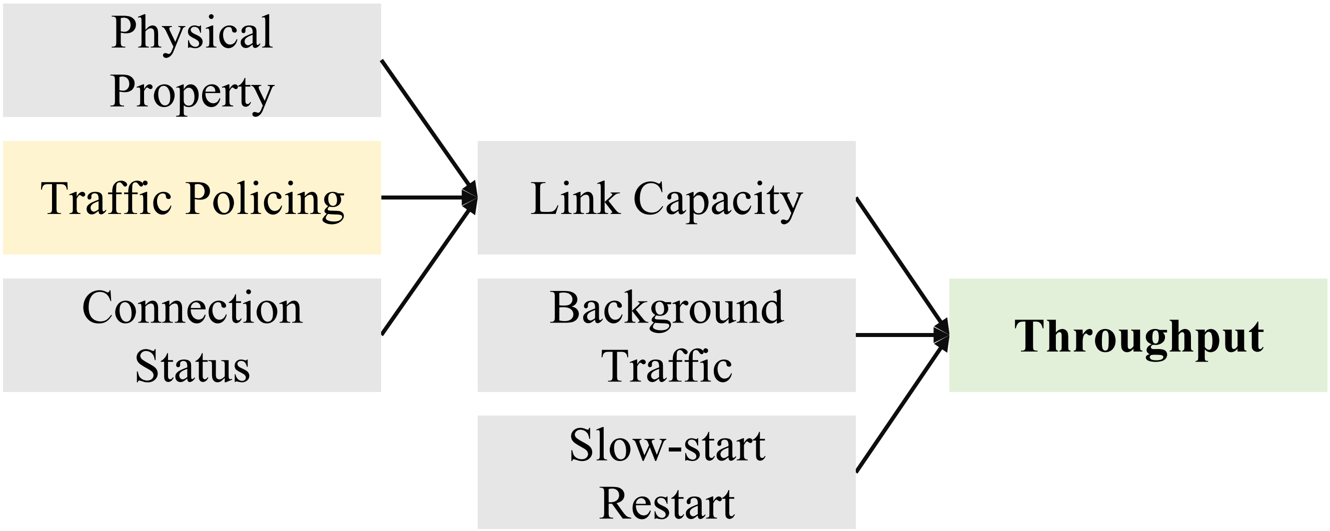


2016

Connections with similar status
exhibit similar throughput patterns

Identify Features

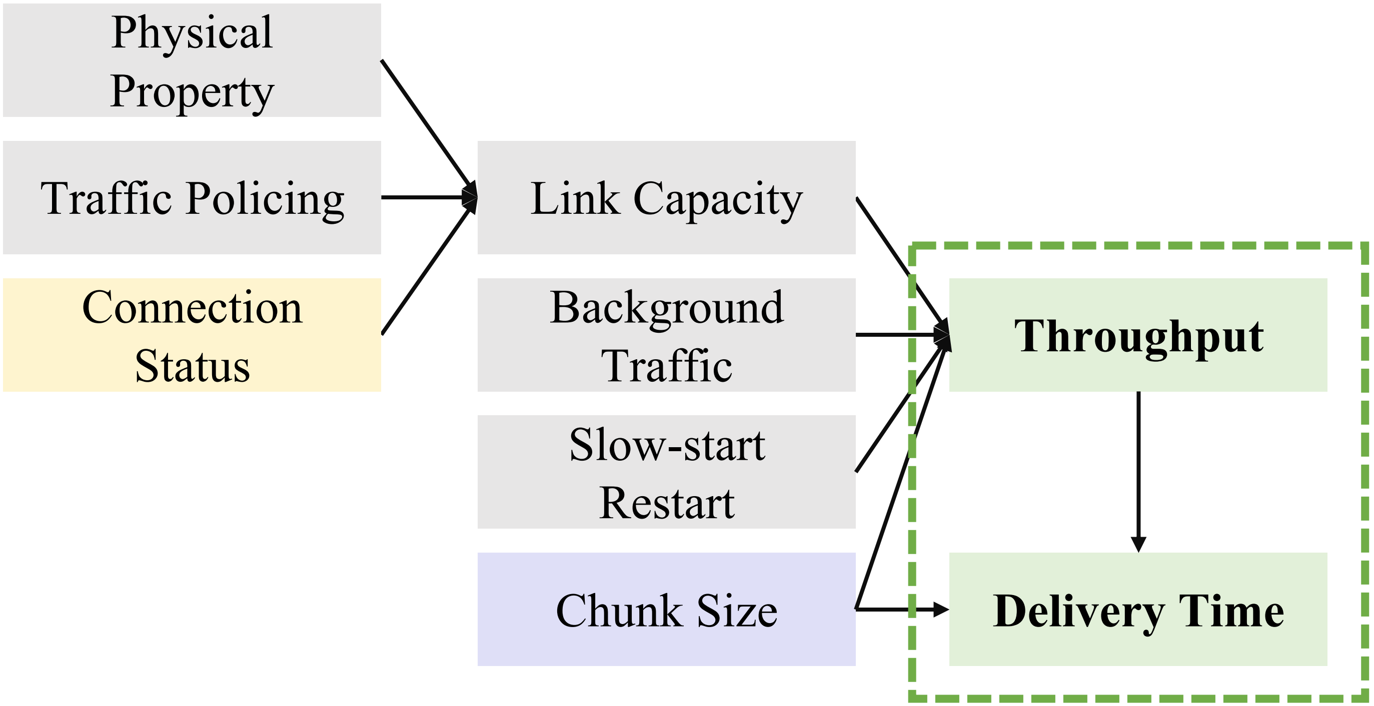
❖ Academic research evolves in this area



2016 | **Traffic Policing** impacts video throughput

Identify Features

❖ Academic research evolves in this area

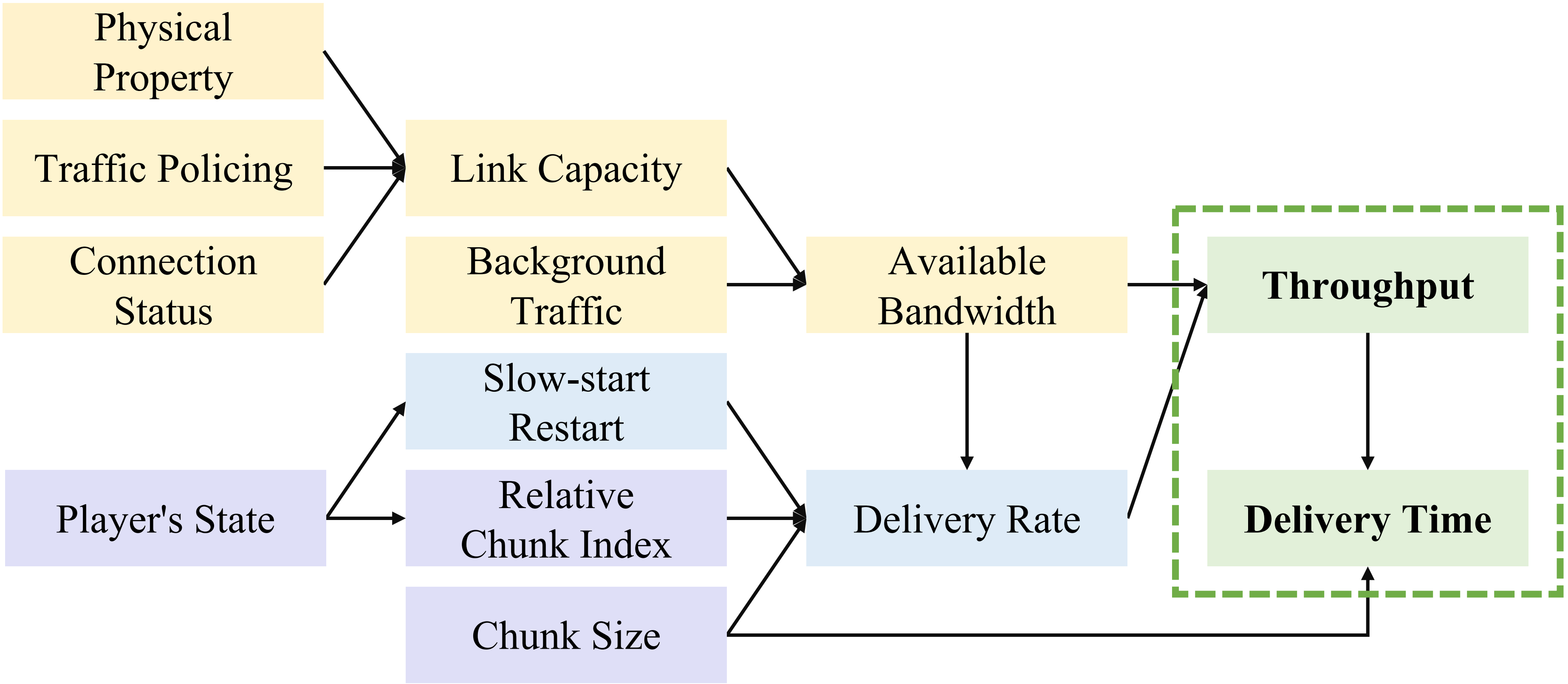


2020

Chunk size is useful to predict **delivery time**

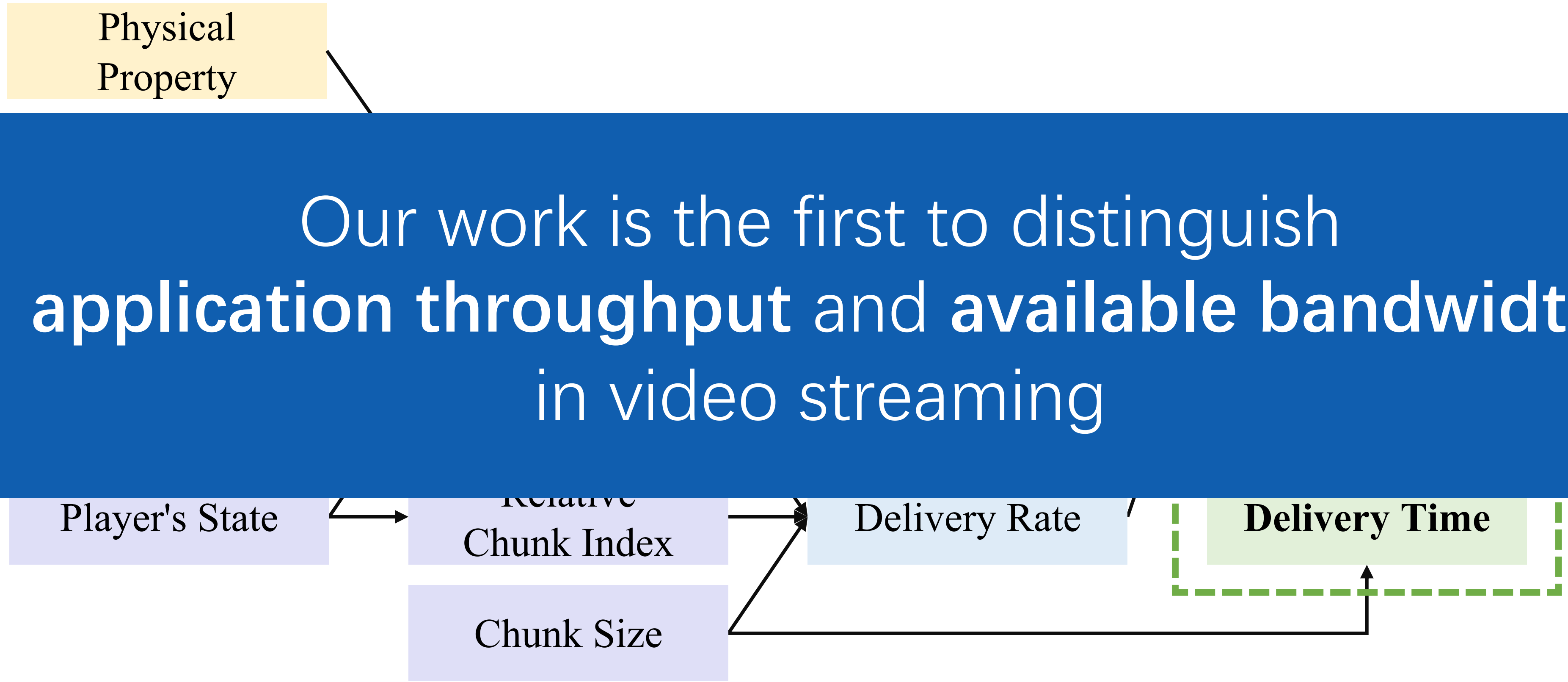
Identify Features

❖ **Theoretical Framework**: all factors impacting throughput and delivery time



Identify Features

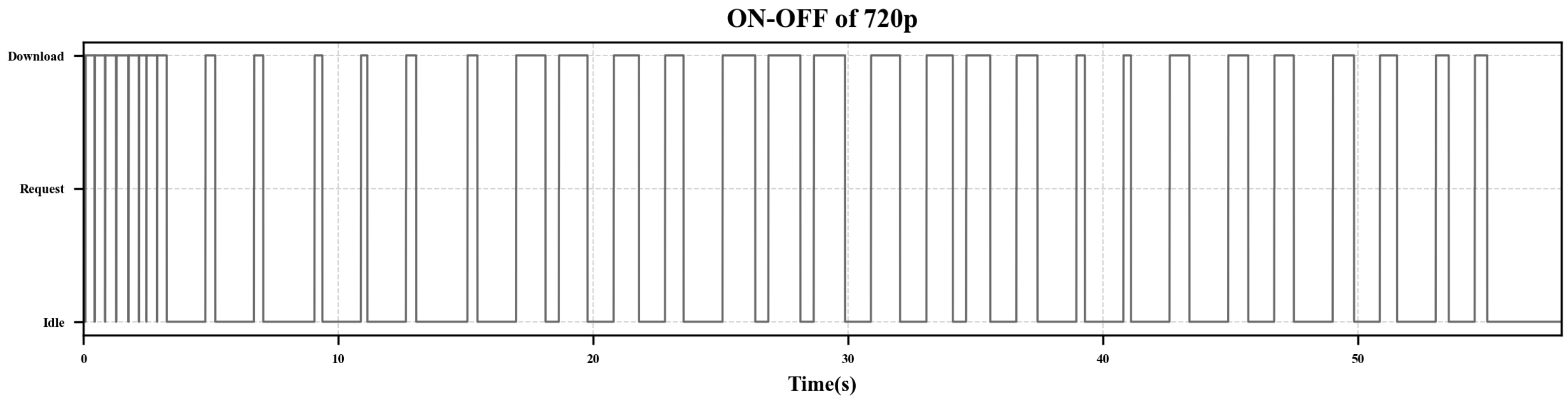
❖ **Theoretical Framework**: all factors impacting throughput and delivery time



Our work is the first to distinguish **application throughput** and **available bandwidth** in video streaming

ON-OFF Period in Video Streaming

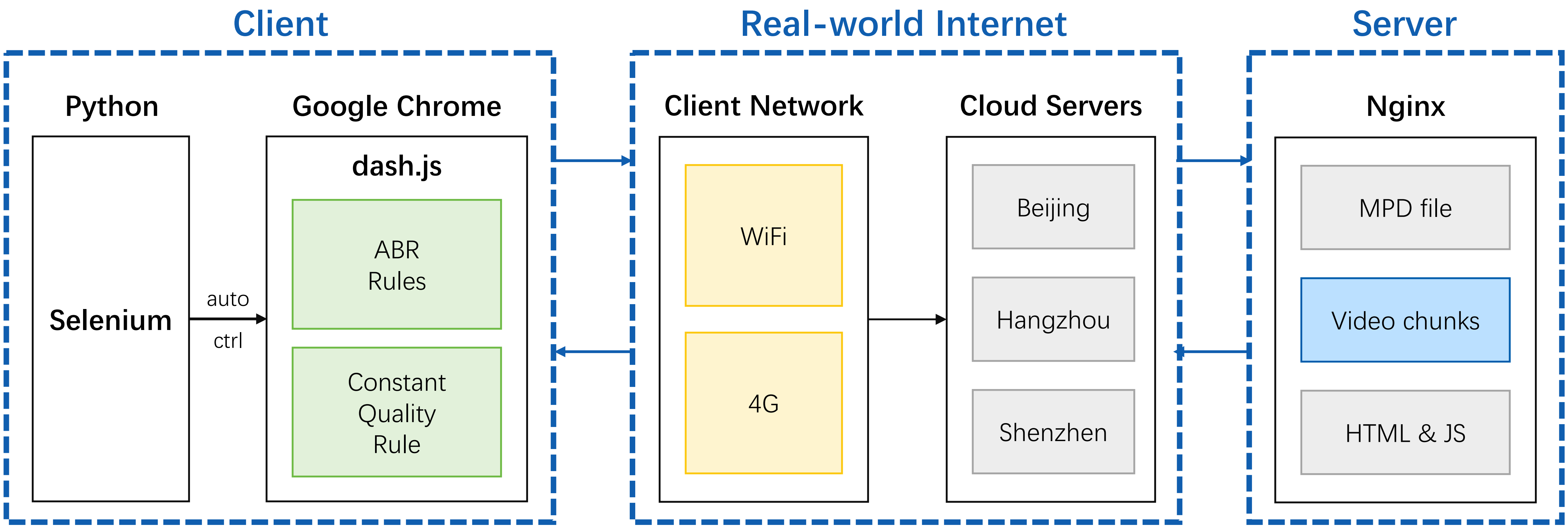
- ON-OFF: A unique behavior of video streaming
  - Player requests for chunks **periodically** rather than **continuously** after the start-up phase
- Slow-start Restart
  - If OFF period exceeds a timeout (200ms in Linux), the server will reset *cwnd* to its initial size, and return to slow-start phase for the next transmission
- Player's State
  - **Buffering State**: *cwnd* is adjusted continuously in subsequent chunks
  - **Steady State**: slow-start restart occurs when delivering each chunk



Collect Dataset

❖ Automated video streaming measurement platform

- **Video**: Elephant Dream & Big Buck Bunny at [300, 750, 1200, 1850, 2850, 4300] Kbps
- **Client**: based on Selenium to automictically control Chrome browser to run dash.js player
- **Server**: deployed on cloud servers in 3 cities, hosting video contents and HTML & JS codes



Collect Dataset

❖ Automated video streaming measurement platform

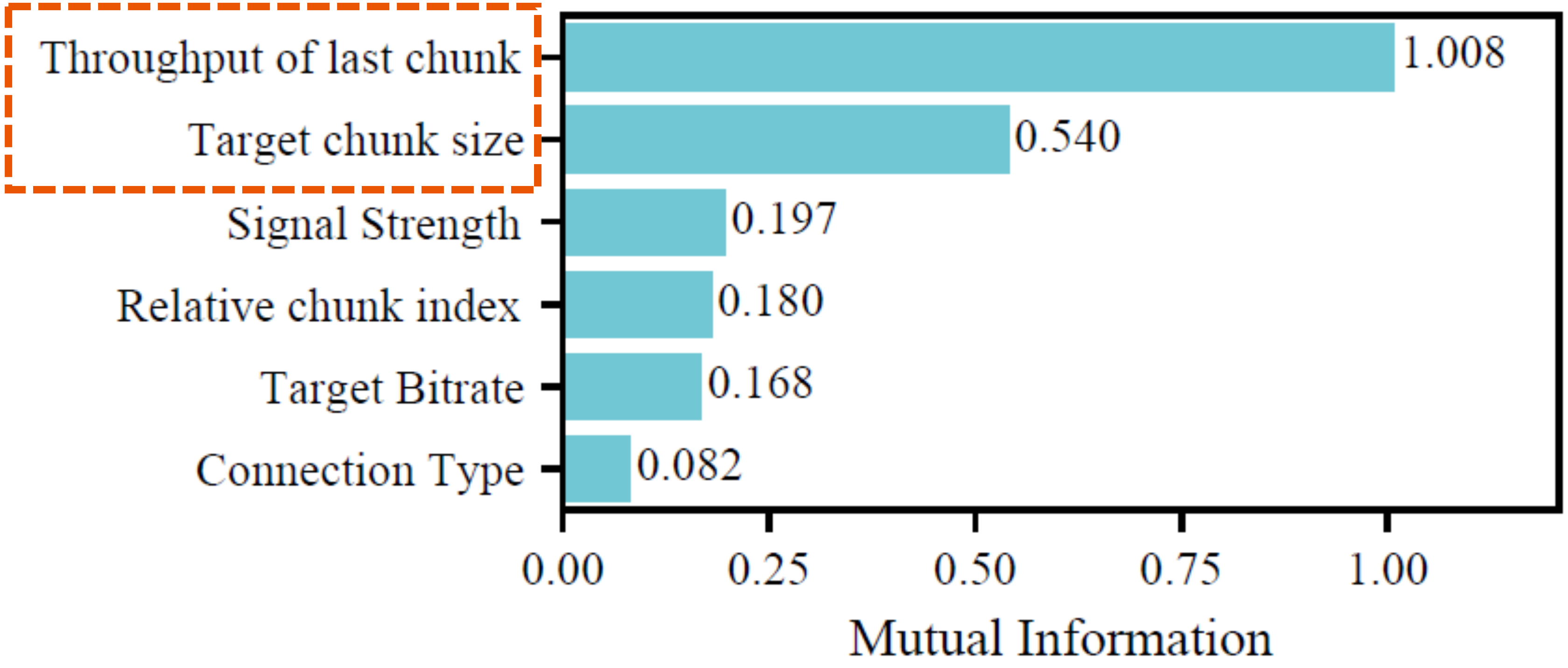
- Collected information

Client Player	Server Transport Layer
Constant bitrate rule & ABR rules	tcp_probe
Playback Information chunk size, delivery time, application throughput, buffer level, inactivity time, rebuffering time, etc. **Signal Strength** WiFi RSSI, 4G RSRP & SINR	**TCP Information** *cwnd* size, *ssthresh*, smoothed RTT, inflight size, etc.

We collected data of **2500+ video sessions**, containing **300,000+ video chunks** from 2019.12.30 to 2021.05.18

Identify Features

❖ **Observation 1**: Throughput of last chunk and target chunk size are the two most important factors in throughput prediction

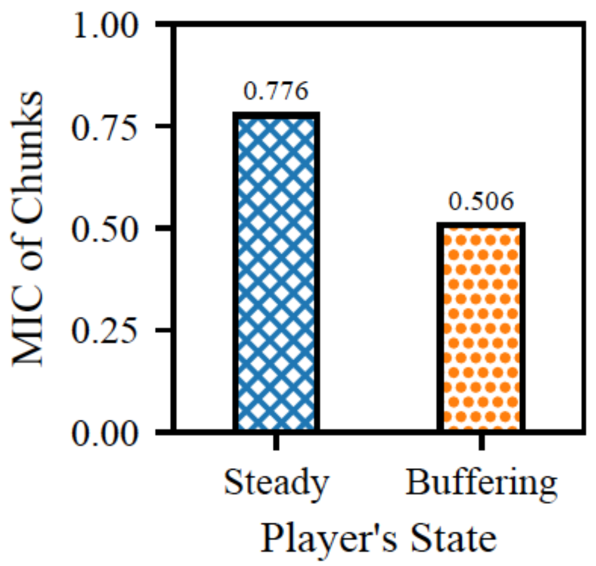


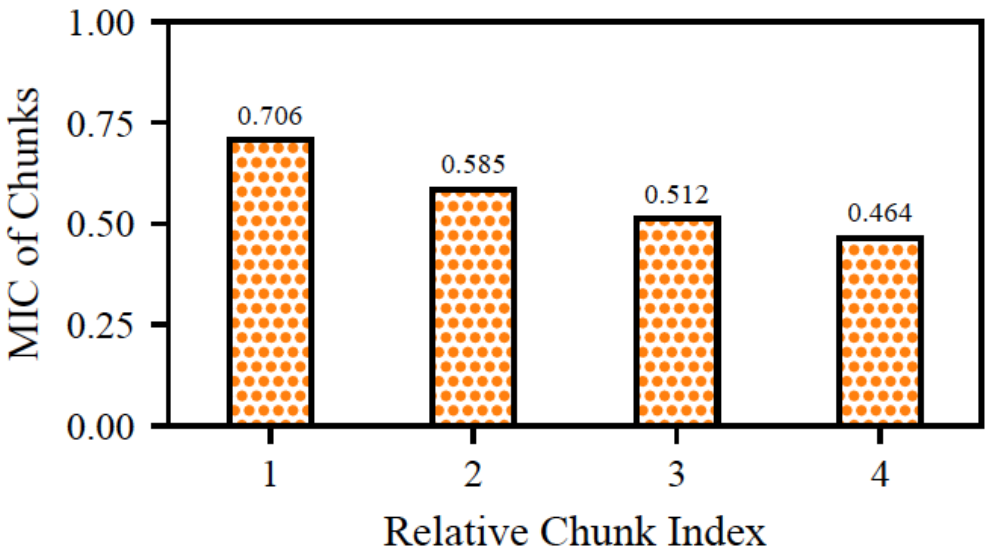
How does video chunk size affect application throughput?

Identify Features

❖ **Observation 2:** Correlation between throughput and chunk size is deeply affected by player's state, relative chunk index, and signal strength of the client

- 1. Player's state
  - The correlation in the Steady-State is higher than that in the Buffering-State
  - The correlation decreases as relative chunk index increases

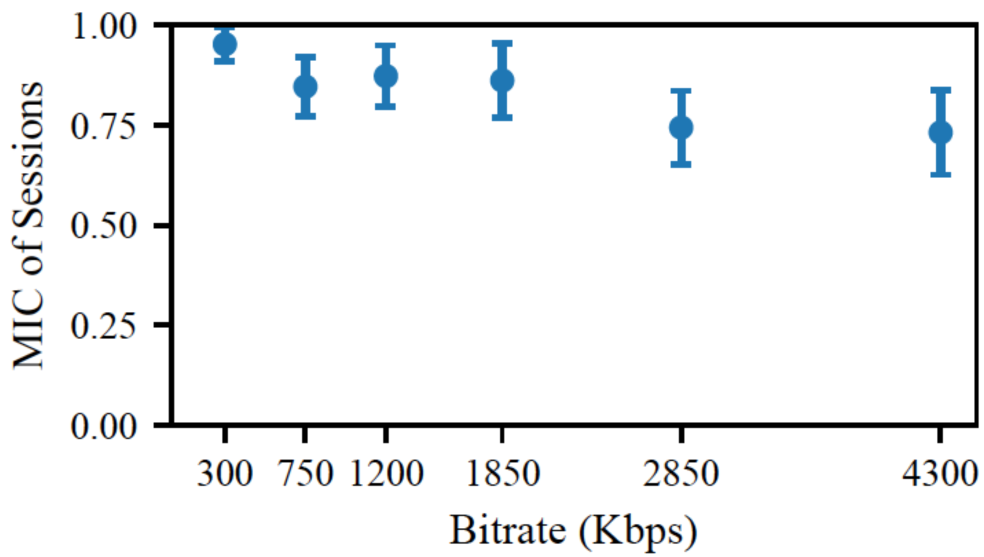




Identify Features

❖ **Observation 2:** Correlation between throughput and chunk size is deeply affected by <u>player's state</u>, <u>relative chunk index</u>, and <u>signal strength of the client</u>

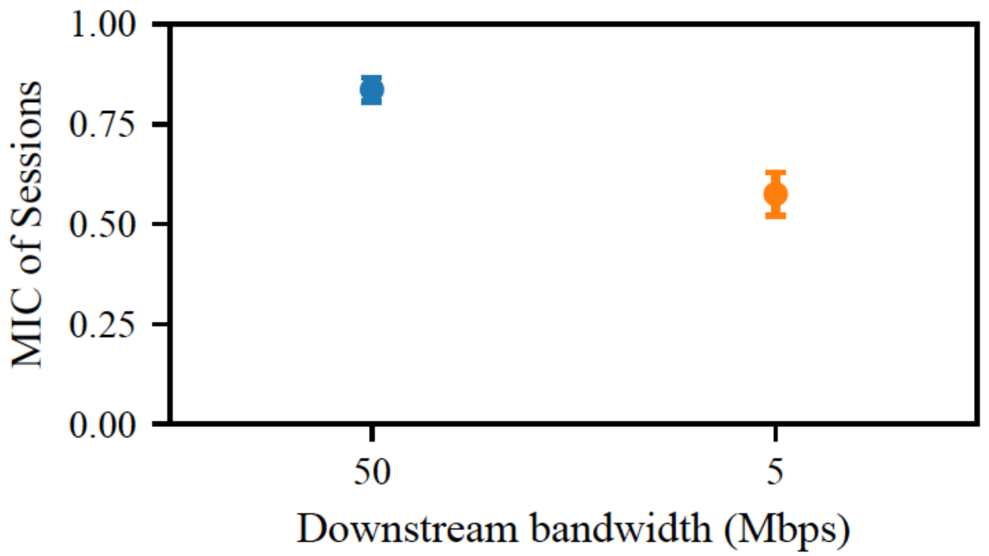
- 2. Bitrate level
  - The correlation becomes lower as the bitrate increases

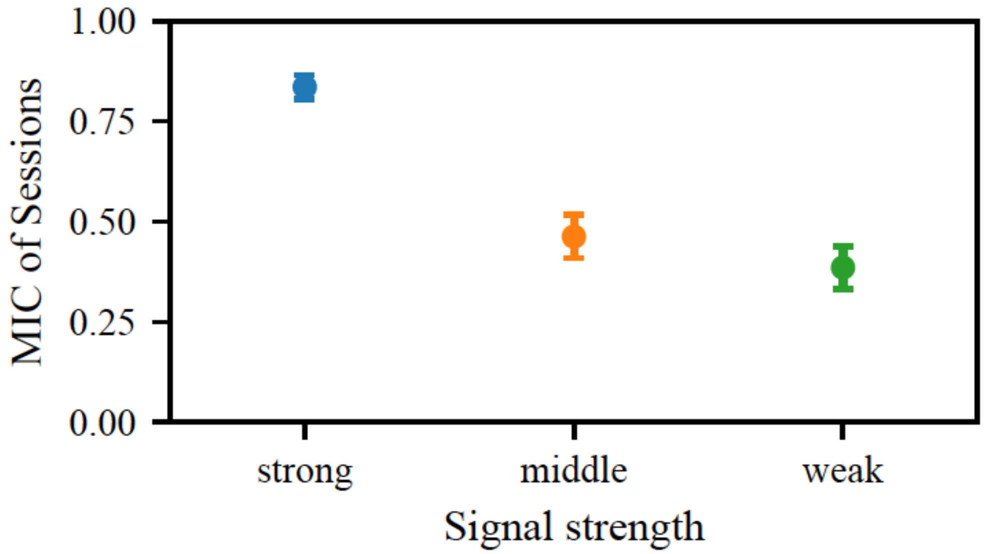


Identify Features

❖ **Observation 2:** Correlation between throughput and chunk size is deeply affected by player's state, relative chunk index, and signal strength of the client

- 3. Network condition
  - Downstream bandwidth of the server
  - Wireless signal strength of the client





Application throughput is determined by both **available bandwidth** and **delivery rate**

Identify Features

❖ Features of Predictor

1

Player's State

relative index of the last chunk

2

Chunk Information

bitrate & size of the last chunk
bitrate & size of the target chunk

3

Network conditions

max throughput of past chunks
max delivery time of past chunks
connection type
throughput of the last chunk

Solution

❖ Correspondingly, our solution includes 3 steps

1	2	3
Identify Features	**Select Target & Model**	Implement Predictor
Qualitative Analysis Quantitative Verification	Controlled Experiment	Design Principle Real-world Deployment

Select Target & Model

- Output Target: Throughput and Delivery Time can be converted to each other with chunk size given
  - **Throughput**: corresponding to bitrate (output of ABR algorithms)
  - **Delivery Time**: used to calculate QoE (target of ABR algorithms)

- Mapping Function: Data-driven methods
  - Multiple Linear Regression (MLR)
  - Decision Trees
  - Deep Neural Networks (DNN)
    - Effective, however heavyweight and short of interpretability, and thus hard to deploy[1][2]

[1] Z. Meng, J. Chen, Y. Guo, C. Sun, H. Hu, and M. Xu, "Pitree: Practical implementation of abr algorithms using decision trees", MM, 2019
[2] Z. Meng, M. Wang, J. Bai, M. Xu, H. Mao, and H. Hu, "Interpreting deep learning-based networking systems", SIGCOMM, 2020

Select Target & Model

- Controlled Experiment
  - **Target:** Throughput Predictor vs. Delivery Time Predictor
  - **Model:** Decision Tree Predictor vs. MLR Predictor

	Throughput	Delivery Time
Decision Tree	Tree-Thput	Tree-DTime
MLR	MLR-Thput	MLR-DTime

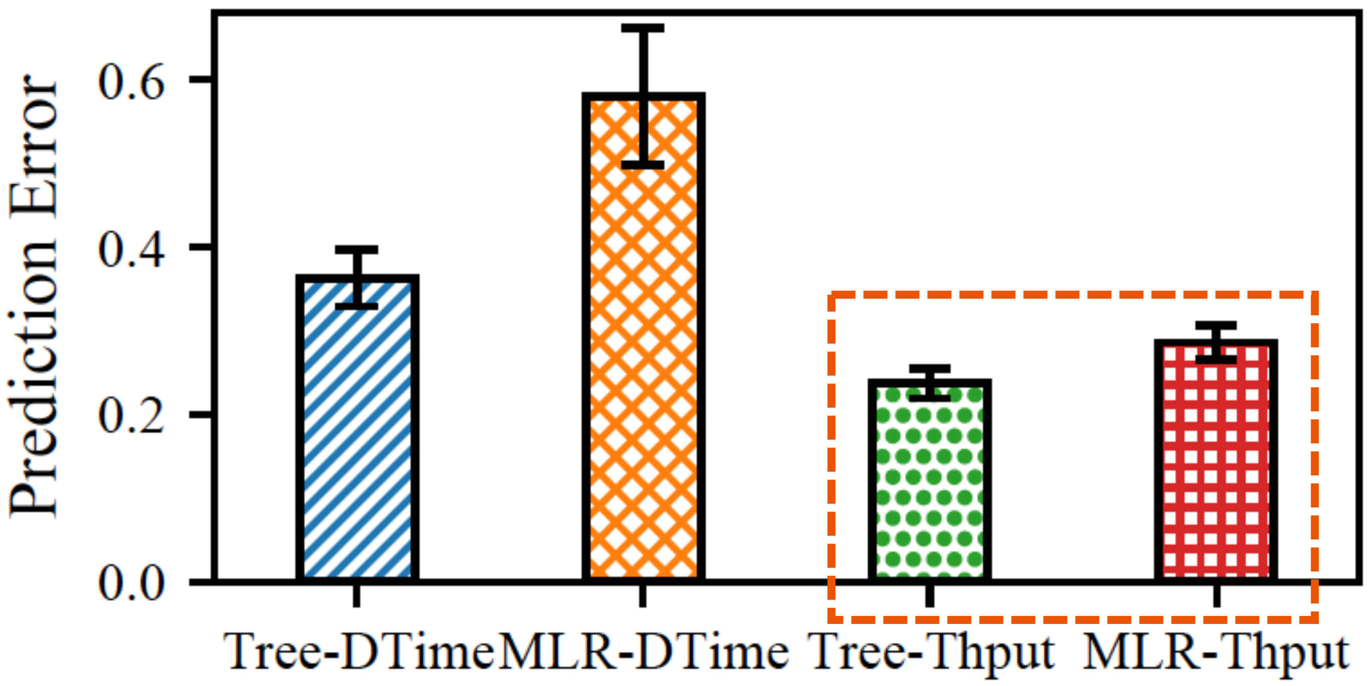
- Metric: Absolute Normalized Prediction Error

$$Err(DTime) = \frac{1}{N}\sum_{k=1}^{N}\frac{\widehat{DT_k} - DT_k}{DT_k}$$

* The predicted throughput is converted to delivery time for comparison with the directly predicted delivery time

Select Target & Model

- **Observation 3**: <u>Throughput prediction achieves better accuracy</u> than delivery time prediction does

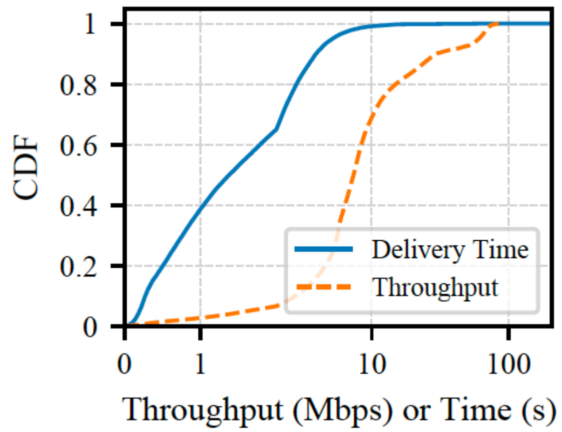


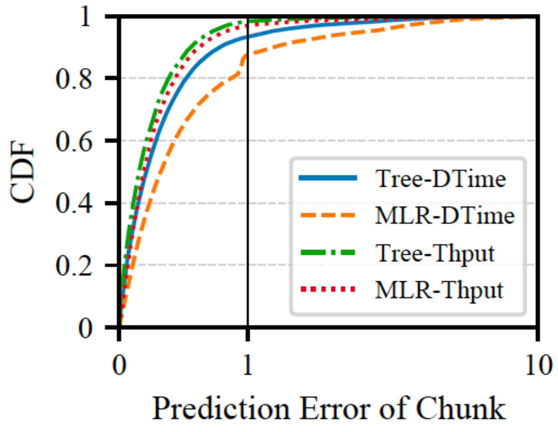
* The predicted throughput is converted to delivery time for comparison with the directly predicted delivery time

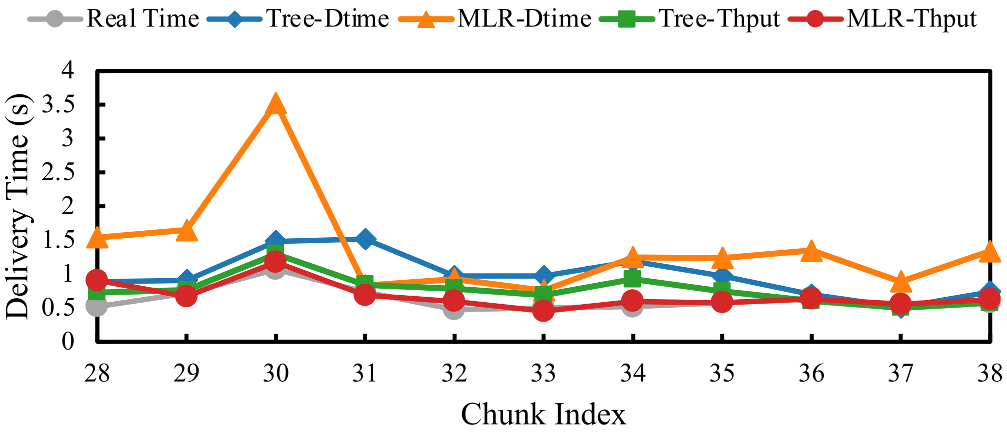
Select Target & Model

❖ **Observation 3**: Throughput prediction achieves better accuracy than delivery time prediction does

- The **long-tailed distribution** of delivery time causes data-driven predictors tend to overestimate the real values
  - 0.8% of chunks lie in the 95% tail of time interval (from 10s to 208s)
- Delivery time predictors perceive prediction error of over 100% for more chunks (6.6%~12.4%) than throughput predictors (1.7%~3.0%)







Solution

❖ Correspondingly, our solution includes 3 steps

1	2	3
Identify Features	**Select Target & Model**	**Implement Predictor**
Qualitative Analysis Quantitative Verification	Controlled Experiment	Design Principle Real-world Deployment

Key Problems

- A predictor builds a map between input and output, containing 3 components

$$Prediction = f(Features)$$

1 **Input Features**

What **factors** assist to achieve better prediction?

2 **Output Target**

Which one of **throughput** and **delivery time** is a better target to predict?

3 **Mapping Function**

Which **model** is more suitable for prediction?

Implement Predictor

❖ Design Principle

1	2	3
Input Features	**Output Target**	**Mapping Function**
What **factors** assist to achieve better prediction?	Which one of **throughput** and **delivery time** is a better target to predict?	Which **model** is more suitable for prediction?
Player's State **Chunk Information** **Network Conditions**	**Throughput**	**Decision Trees**

Lumos: decision-tree-based throughput predictor for ABR streaming

Lumos Mechanism

❖ Train

- Method: Classification and Regression Tree (CART)
  - Regression tree
  - Loss function: mean squared error (MSE)
- Dataset: 700+ ABR video sessions in the real world, containing 69,000+ chunks
  - Training set accounts for 70%
  - Data of various combinations of network conditions is balanced
- Optimization:
  - Prepruning and cost complexity pruning
  - Exhaustive grid search
  - K-fold cross validation

❖ Deploy

- Parameters of Lumos models are dumped in JavaScript, loaded in dash.js player

Lumos with ABR algorithms

❖ Lumos as a plug-in of ABR algorithms

RB (CoNEXT'12)	**MPC** (SIGCOMM'15)	**BBA** (SIGCOMM'14)
replace HM predictor	**replace HM predictor**	**combine with prediction**
$R_k = \max_{1 \le j \le m} \{r_j, \frac{d_k(r_j)}{\hat{T}_{k\|r_j}} \le L\}$	$R_k = \arg \max_{r_j, 1 \le j \le m} \sum_{l=k}^{k+n-1} \widehat{QoE}\ (R_{l\|r_j})$	$R_k = \max_{1 \le j \le m} \{r_j, B_{lower} + \frac{d_k(r_j)}{\hat{T}_{k\|r_j}} - \frac{d_k(r_1)}{\hat{T}_{k\|r_1}} \le B_k\}$

Evaluation Setup

❖ Baselines

- Predictors
  - Lumos / MLR / HM / Robust-HM
- ABR algorithms
  - RB (CoNEXT'12) / MPC & RobustMPC (SIGCOMM'15) / BBA (SIGCOMM'14) / Pensieve (SIGCOMM'17)

❖ Environment: Real-world Internet

- On our video streaming measurement platform
- ~300 sessions under various network environments
  - Downstream bandwidth of the server: 50Mbps / 5Mbps
  - Connection types: WiFi / 4G
  - Signal strength: Strong / Middle / Weak

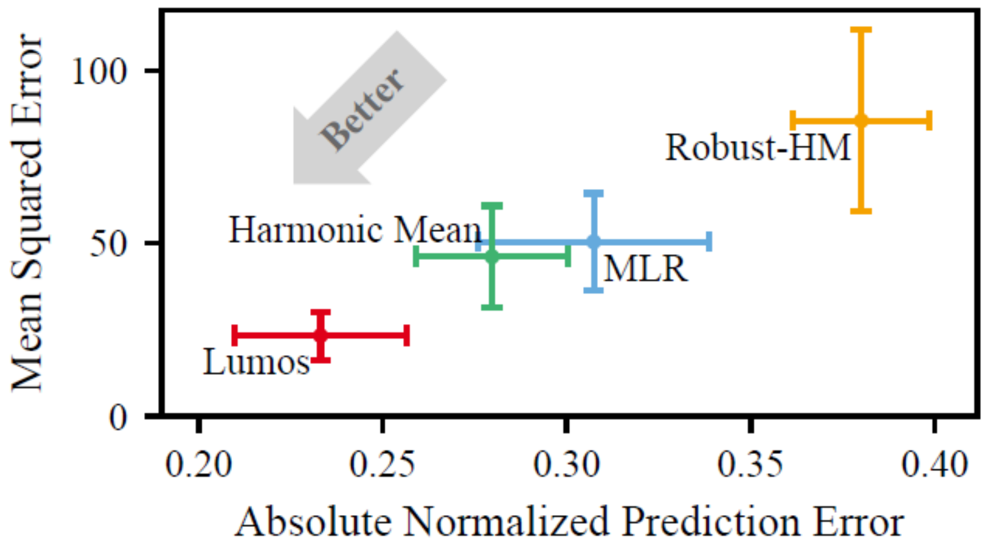
❖ Metrics

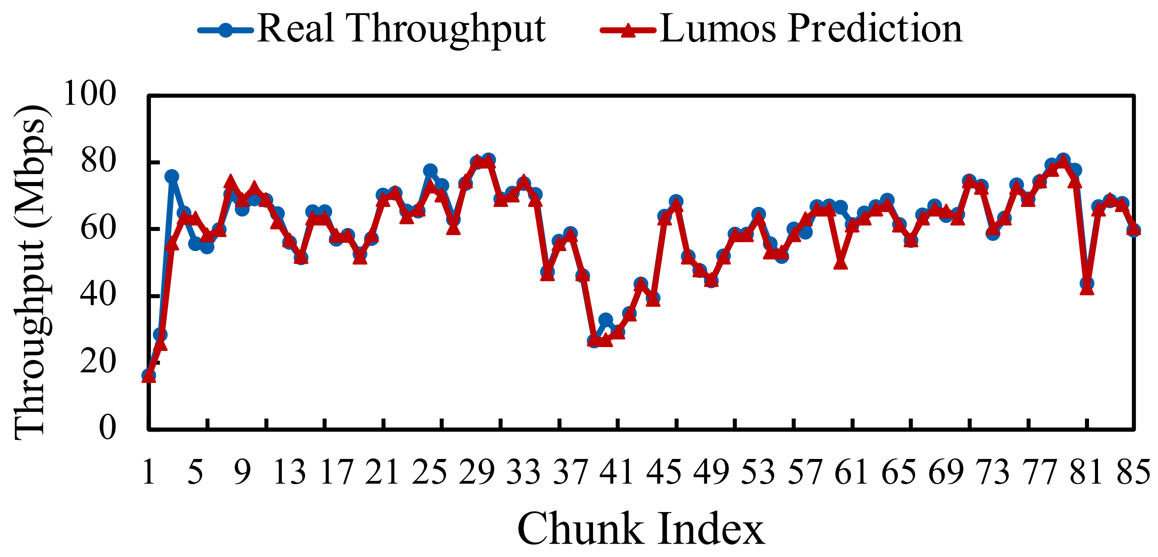
- Prediction accuracy of predictors
  - Prediction Error / Mean Squared Error (MSE)
- QoE performance of ABR algorithms
  - Higher quality / Lower rebuffering time / Fewer quality switches

Evaluation in real-world Internet

❖ Prediction Accuracy

- Lumos reduces prediction error by **16.8%~38.7%**, and MSE by **49.6%~72.8%**
- **Strong signal strength** of WiFi: prediction error is only **7.4%** in average
  - 57.7%~74% and 78.5%~90.5% improvement of the two metrics than others
- **Weak signal strength** where network is hard to predict: improve prediction accuracy over the two metrics by **9.1%~28.9%** and **17.8%~61.7%** respectively

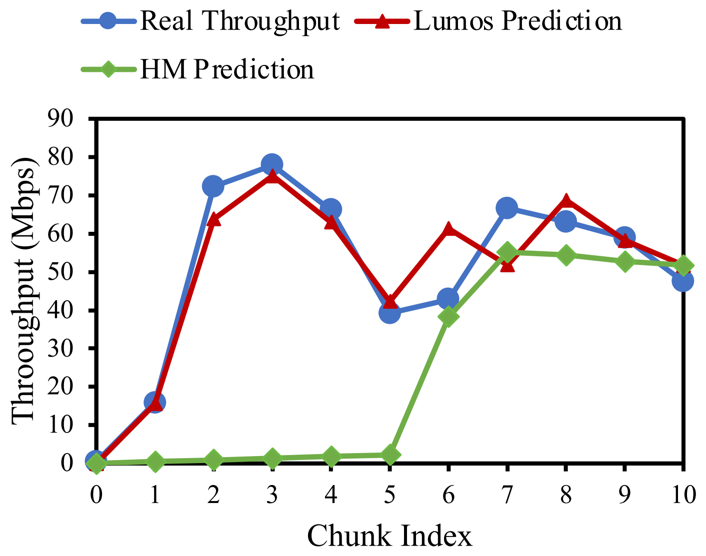


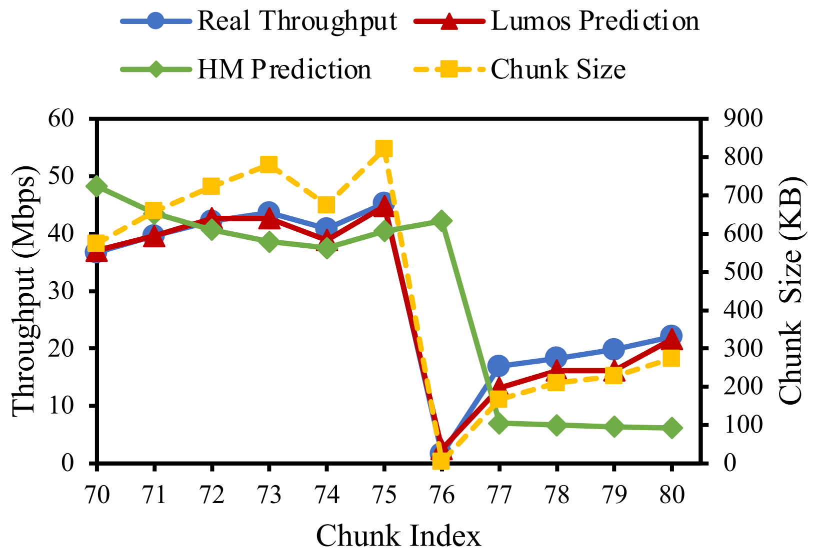


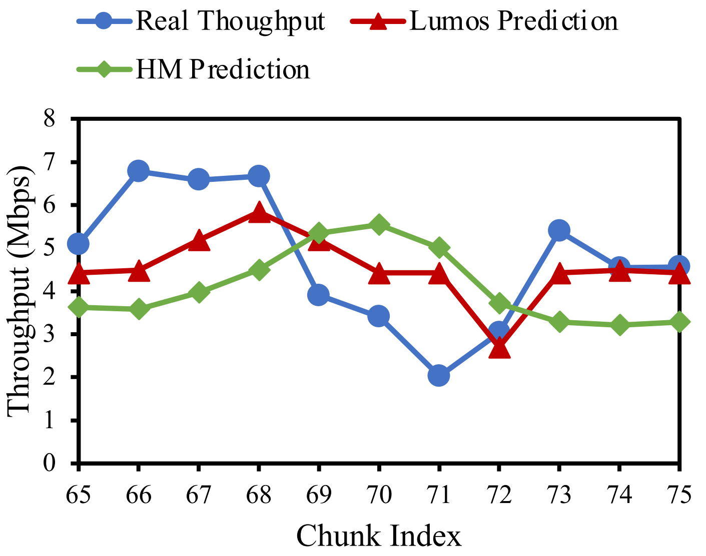
Evaluation in real-world Internet

❖ The Advantage of Lumos over HM (Harmonic Mean)

- Aware of network conditions and player's state: **start-up** phase
- Consider the fluctuation of chunk sizes: **strong** signal strength
- React quickly to bandwidth fluctuation: **weak** signal strength





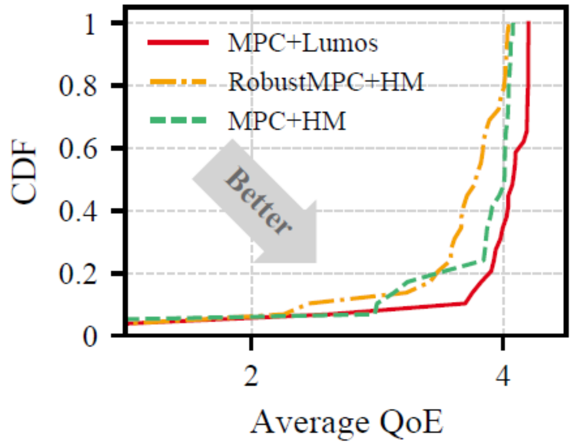


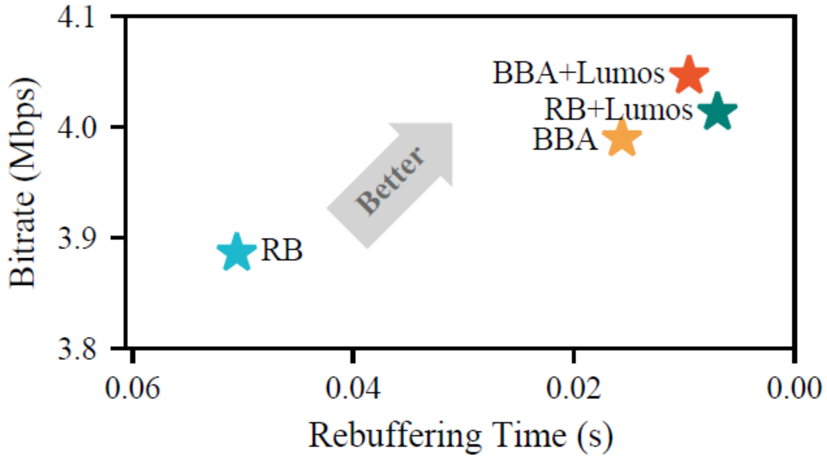
Lumos can distinguish the change of network conditions and application behavior

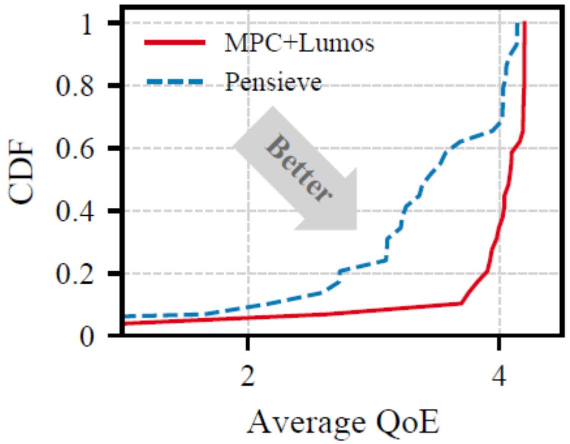
Evaluation in real-world Internet

❖ QoE of Lumos-assisted ABR Algorithms

- with **MPC**: improve QoE by **6.3%** over MPC and **8.7%** over RobustMPC
- with **RB** / **BBA**: reduce rebuffering time by **86.3%** and **37.5%**, and improve bitrate by **3.3%** and **1.3%**, compared with RB and BBA respectively
- vs. **Pensieve**: MPC+Lumos improves QoE by **19.2%** in average
  - Difference between Pensieve's simulator and the real world







Summary

❖ Contributions

- We construct a theoretical framework containing all the impacting factors in predicting throughput and delivery time for video streaming, and distinguish **application throughput** from **available bandwidth** for the first time.
- We build a **real-world** video streaming measurement platform, and collect dataset containing **2500+ sessions**. By data analysis and controlled experiments, we find that:
  - **Strong correlation exists between chunk size and throughput**. This correlation is deeply affected by player's state, relative chunk index, and signal strength of the client.
  - **Throughput is a better prediction target than delivery time** in terms of prediction error for data-driven predictors.
- We propose **Lumos**, a decision-tree-based accurate throughput predictor for ABR streaming. As a plug-in, Lumos assists ABR algorithms to achieve better QoE.

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

Thanks!

Q&A

Presented by **Gerui Lv** from ICT, CAS