Lecture 14: Learning for Networking

CS 234 / NetSys 210: Advanced Computer Networks

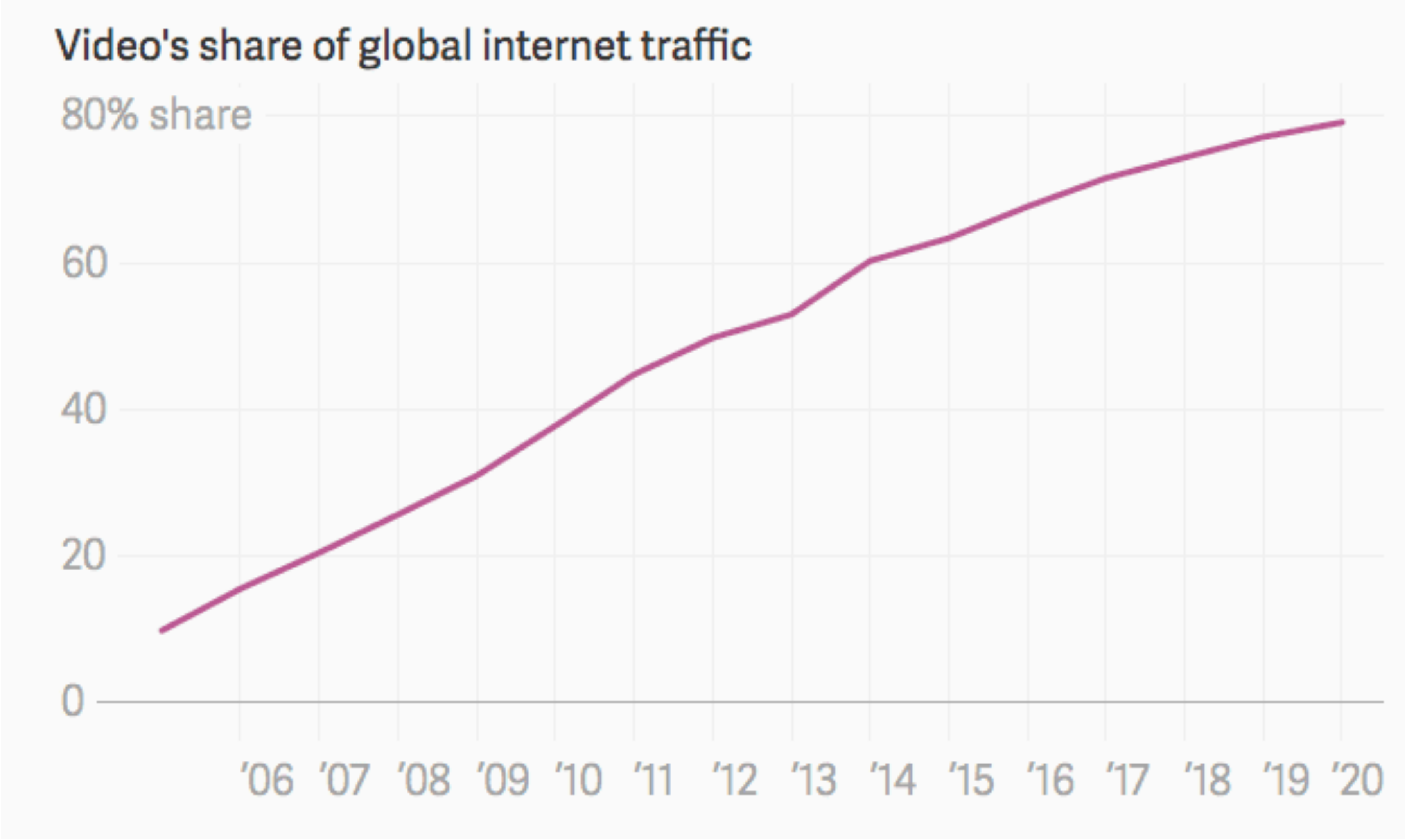
Sangeetha Abdu Jyothi

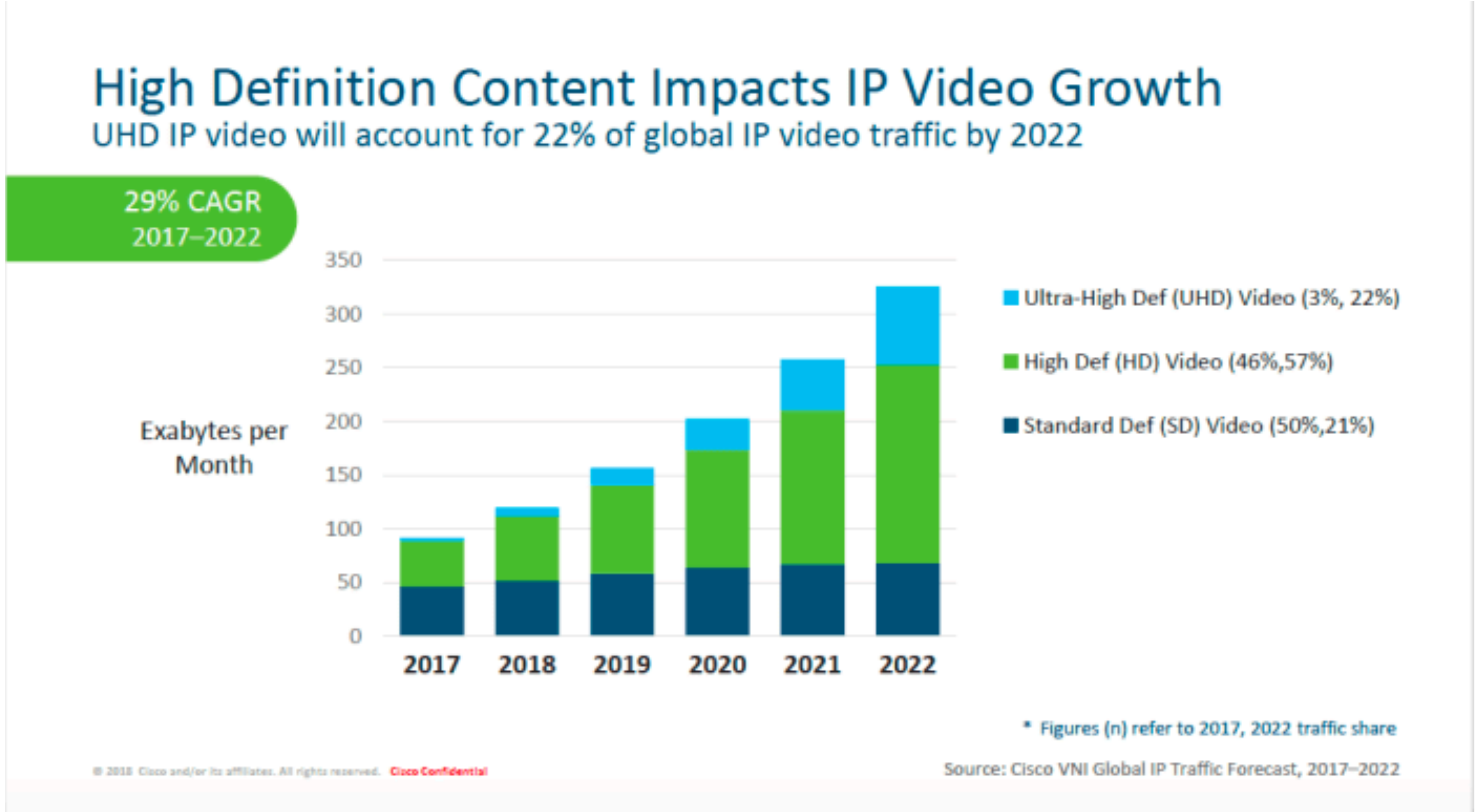
UCIrvine

Parts of this lecture were adapted from talks on Pensieve, CFA

Adaptive Bit Rate Selection

Video Traffic in the Internet

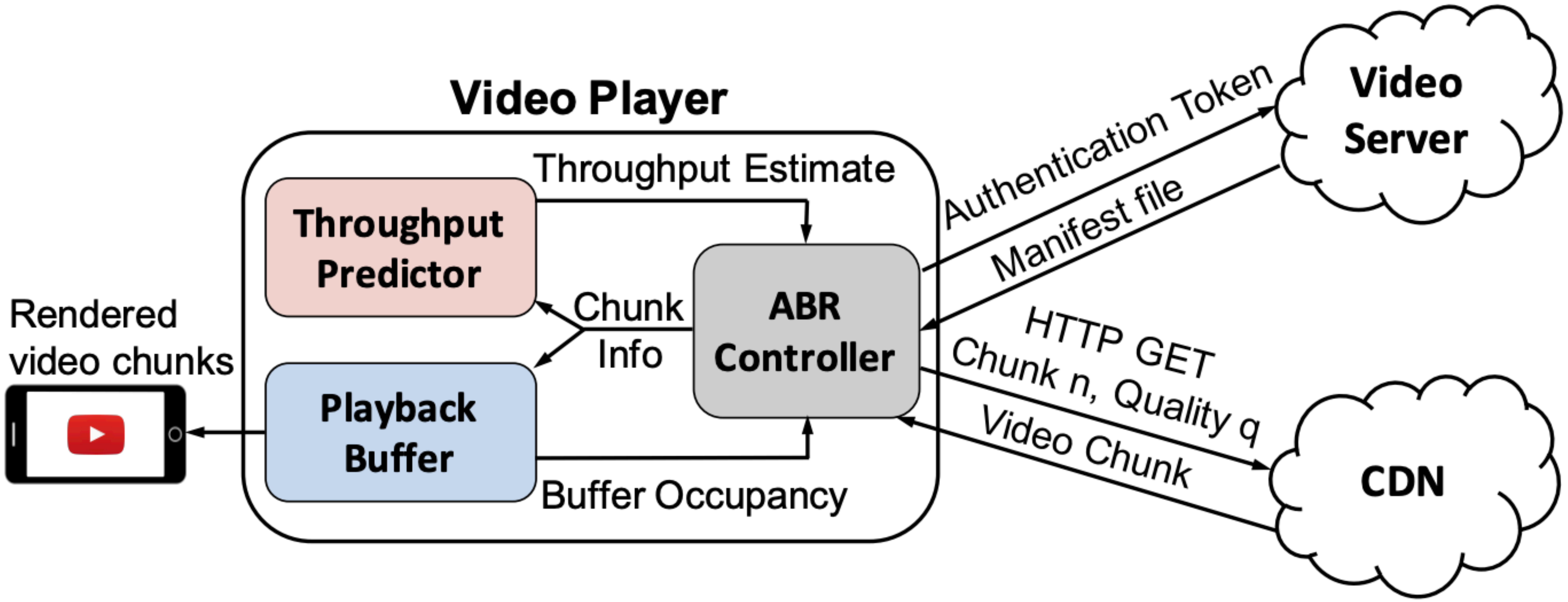




Adaptive Video Streaming

- Video streaming over the network
- Requirements
    - High resolution (high bitrate)
    - Smooth playback (no rebuffering)
    - Start playing immediately

Adaptive Video Streaming



Challenges with ABR

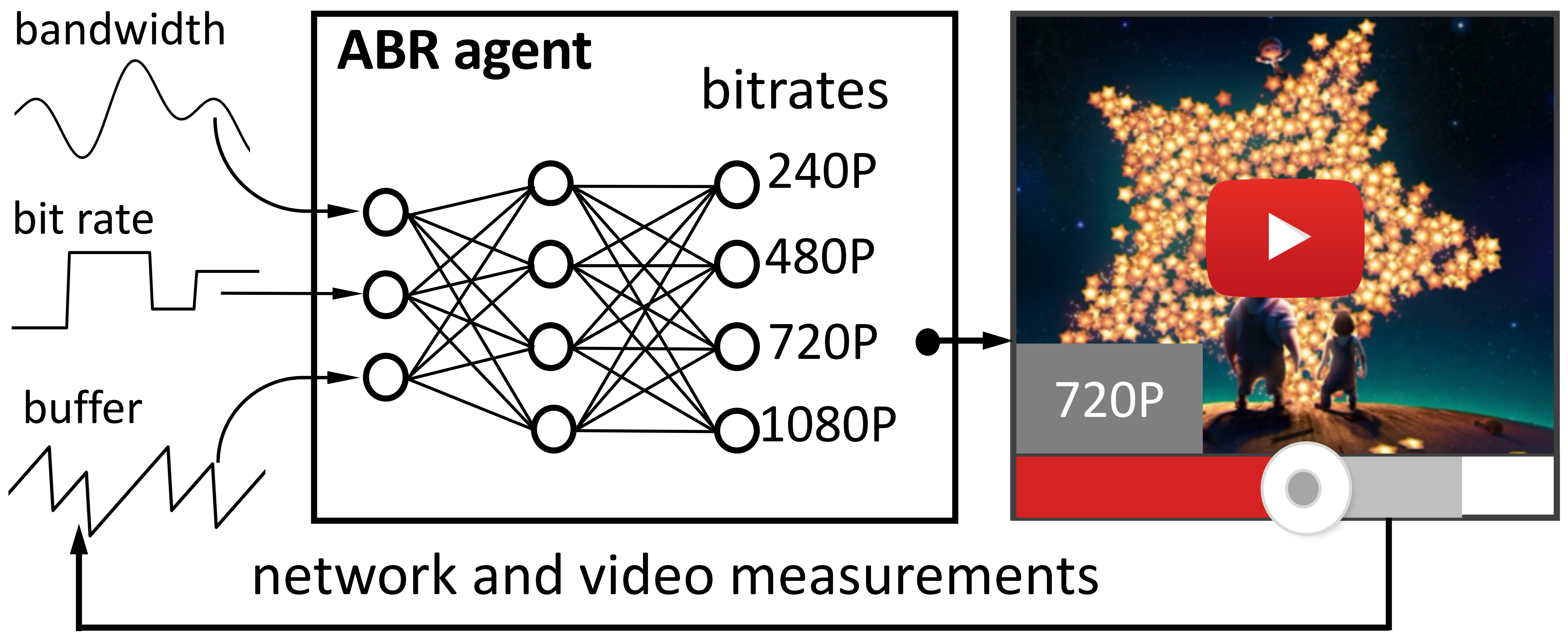
- Fluctuating network conditions
- A variety of QoE goals
- Cascading effects
- Coarse-grained decisions

Previous ABR algorithms

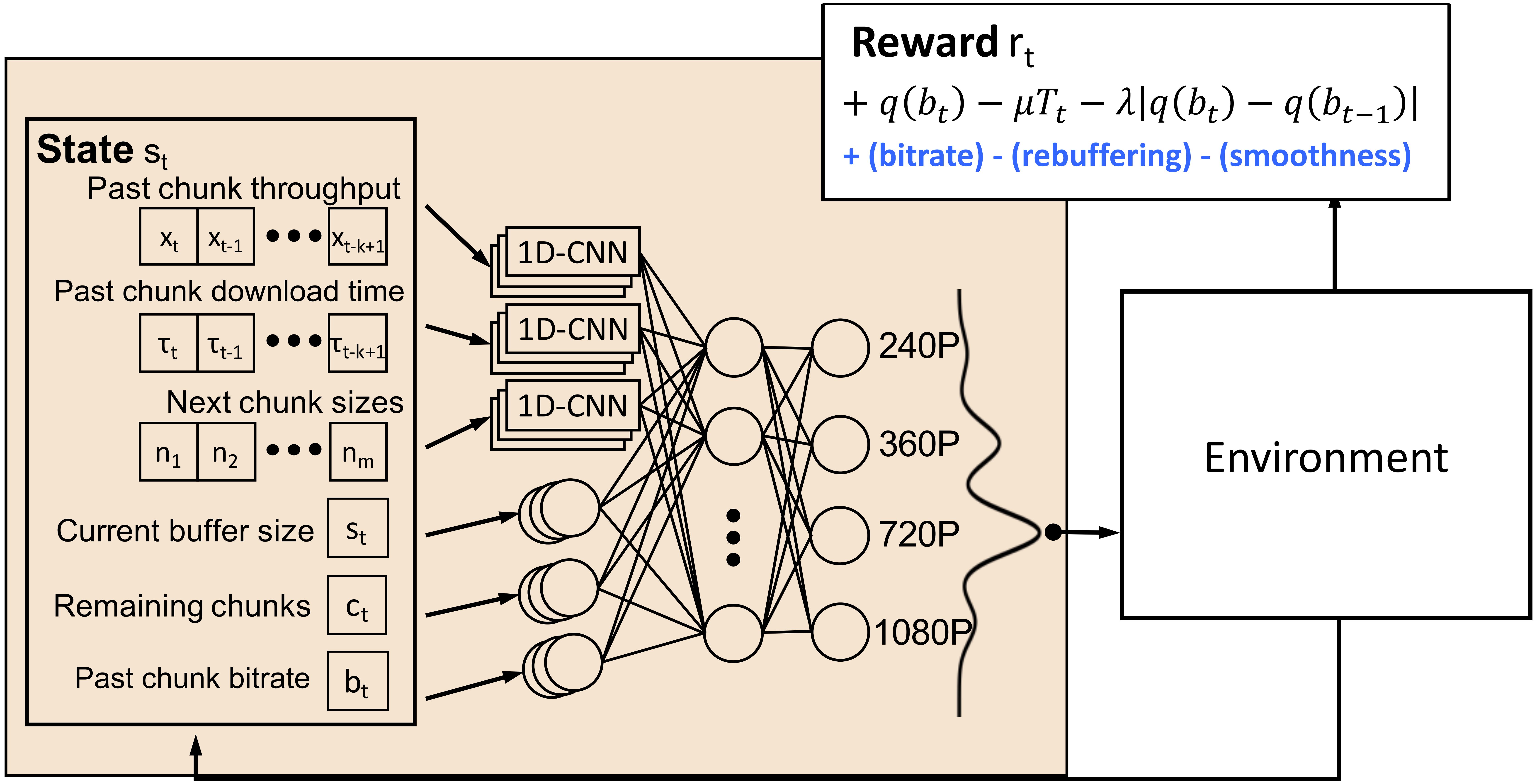
- Rate-based: pick bitrate based on predicted throughput
    - FESTIVE [CoNEXT'12], PANDA [JSAC'14], CS2P [SIGCOMM'16]
- Buffer-based: pick bitrate based on buffer occupancy
    - BBA [SIGCOMM'14], BOLA [INFOCOM'16]
- Hybrid: use both throughput prediction & buffer occupancy
    - PBA [HotMobile'15], MPC [SIGCOMM'15]

Pensieve [SIGCOMM'17]

- Learn from video streaming sessions in actual network conditions
- Deep RL-based solution
- Tailors ABR decisions for different network conditions in a data-driven way
- Delivers 12-25% better QoE, with 10-30% less rebuffering than previous ABR algorithms



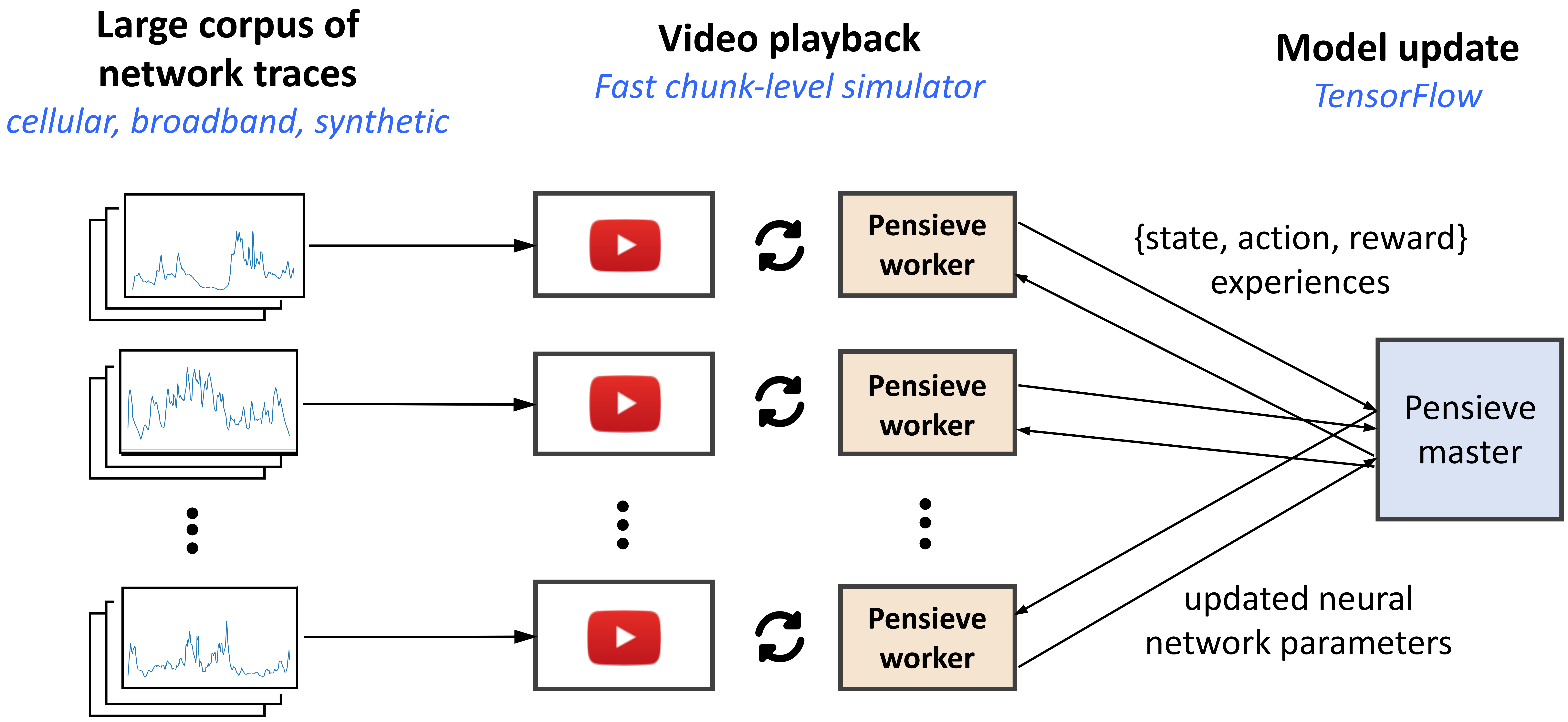
Pensieve Design



Pensieve Advantages

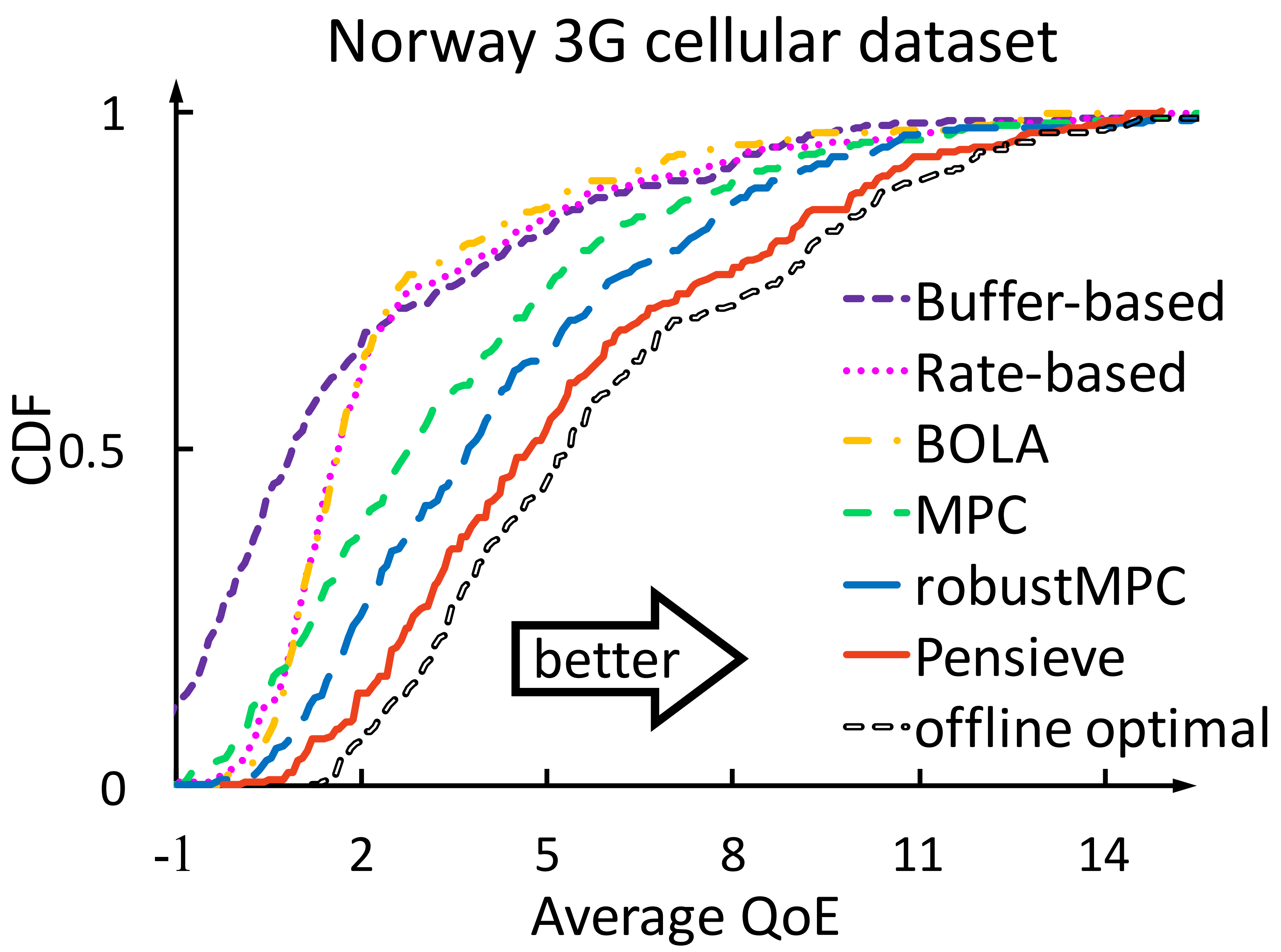
- Learn the dynamics directly from experience
- Optimize the high level QoE objective end-to-end
- Extract control rules from raw high-dimensional signals

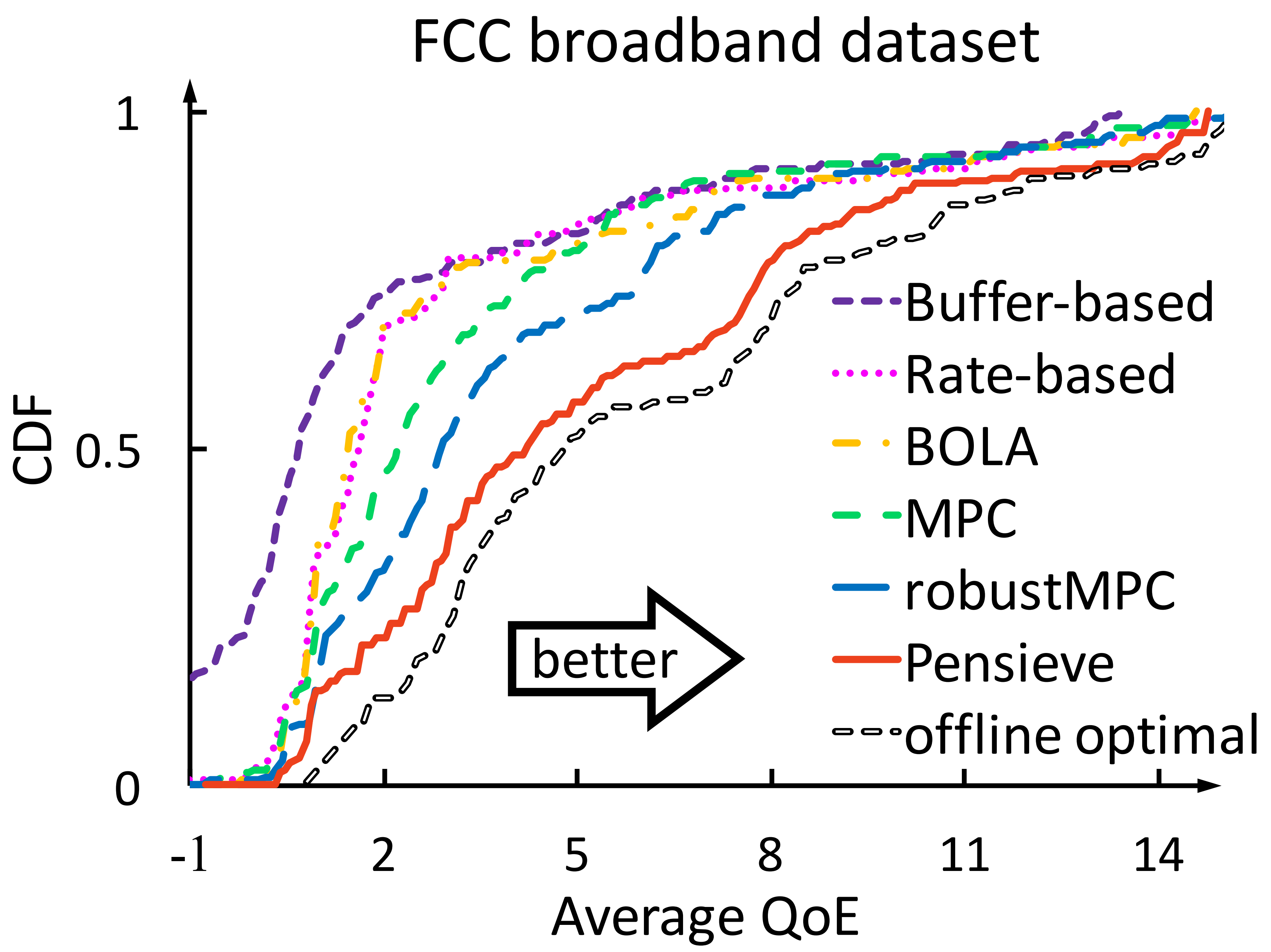
Pensieve Training System



Trace Driven Evaluation

- Dataset: Two datasets, each dataset consists of 1000 traces, each trace 320 seconds.
- Video: 193 seconds. encoded at bitrates: {300, 750, 1200, 1850, 2850, 4300} kbps.
- Video player: Google Chrome browser Video server: Apache server



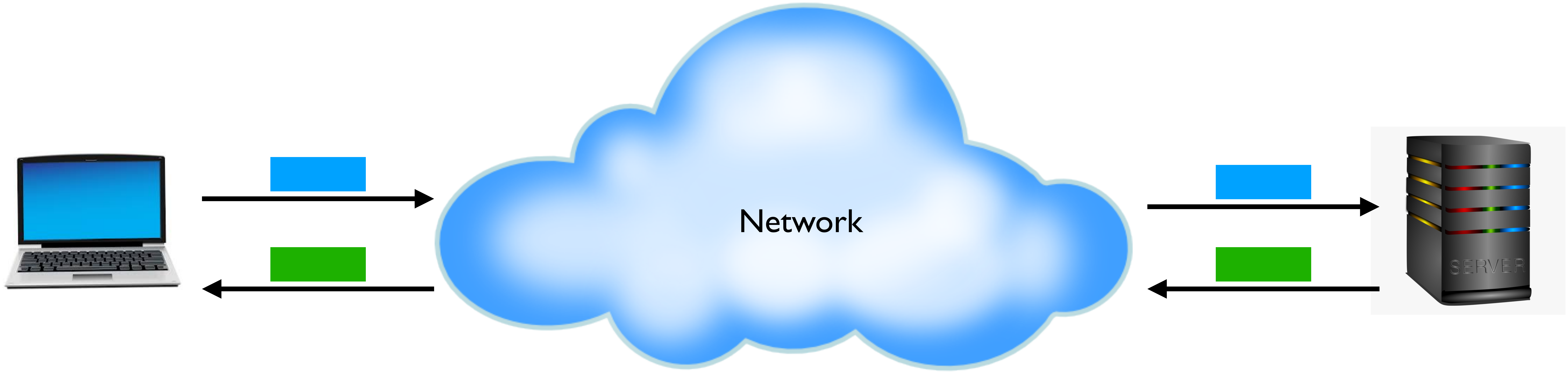


Congestion Control

Congestion Control

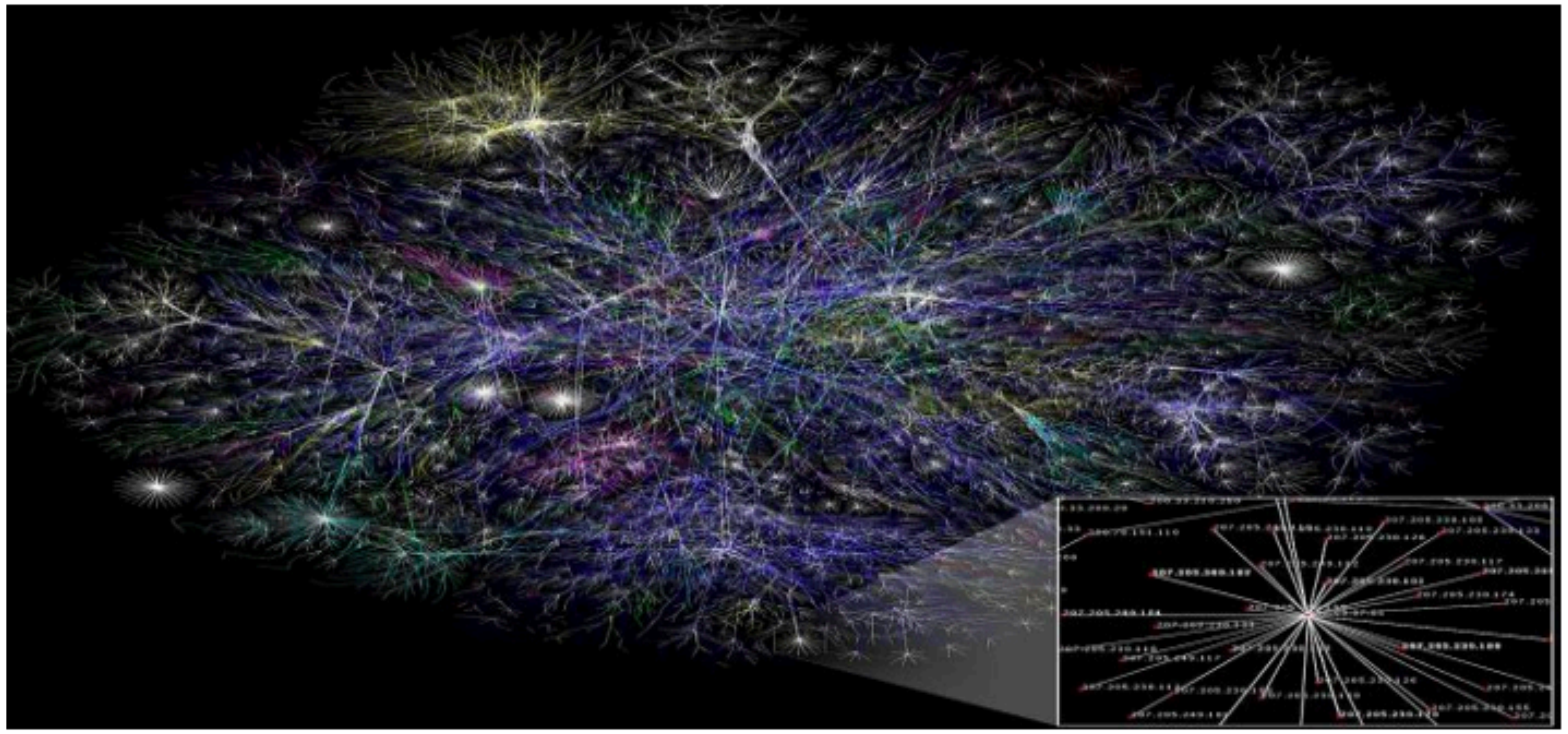
- A longstanding problem in communication networks
- Determines the bandwidth you get out of the network
- Essential for preventing collapse of the network

Congestion Control

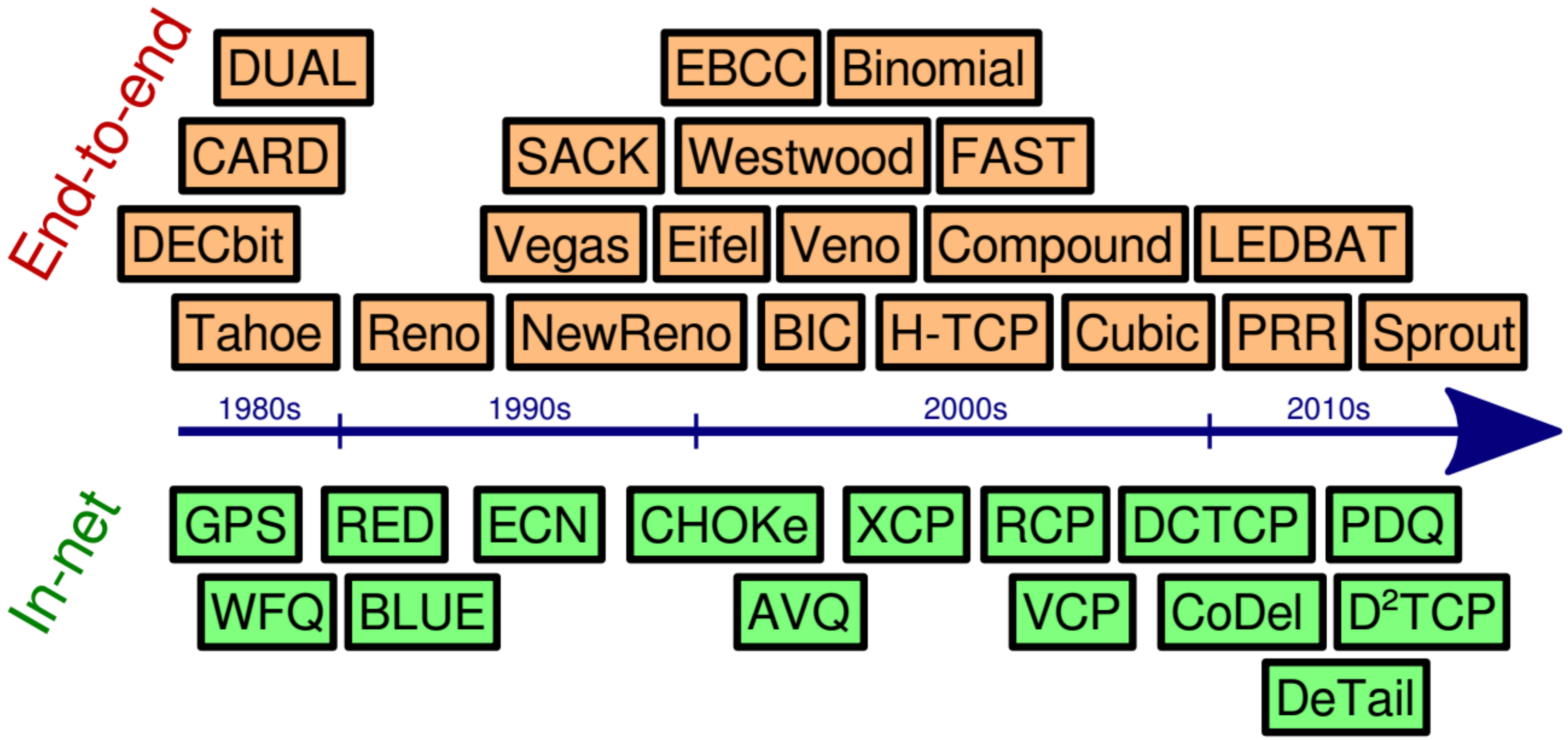


Underlying Complexities

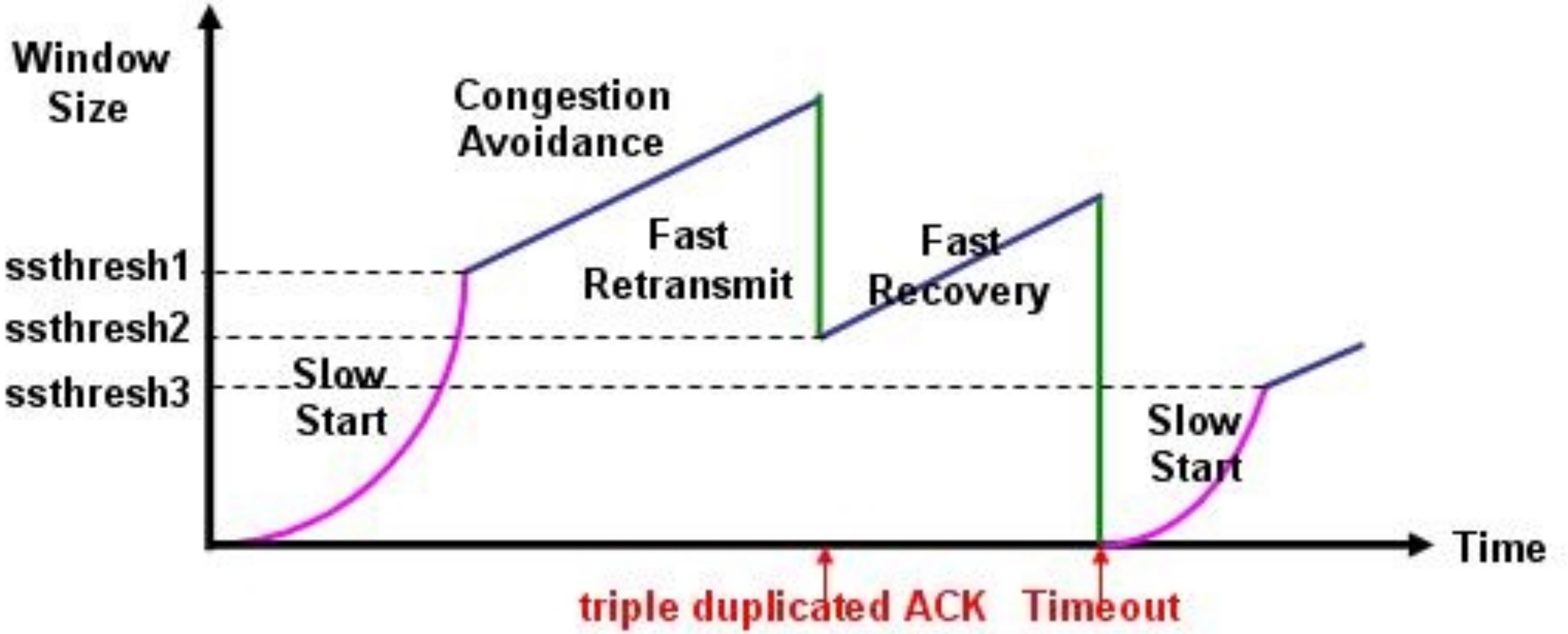
- Enormous, dynamic network
- Massive agent churn
  - (e.g., 80,000/sec for YouTube)
- Limited information at the endhost



History of Congestion Control Mechanisms

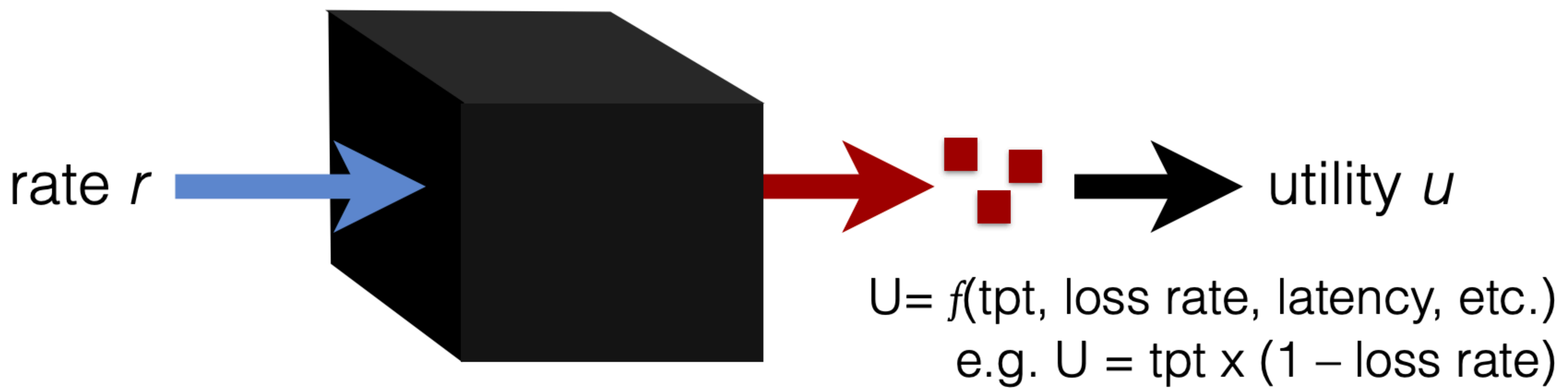


TCP

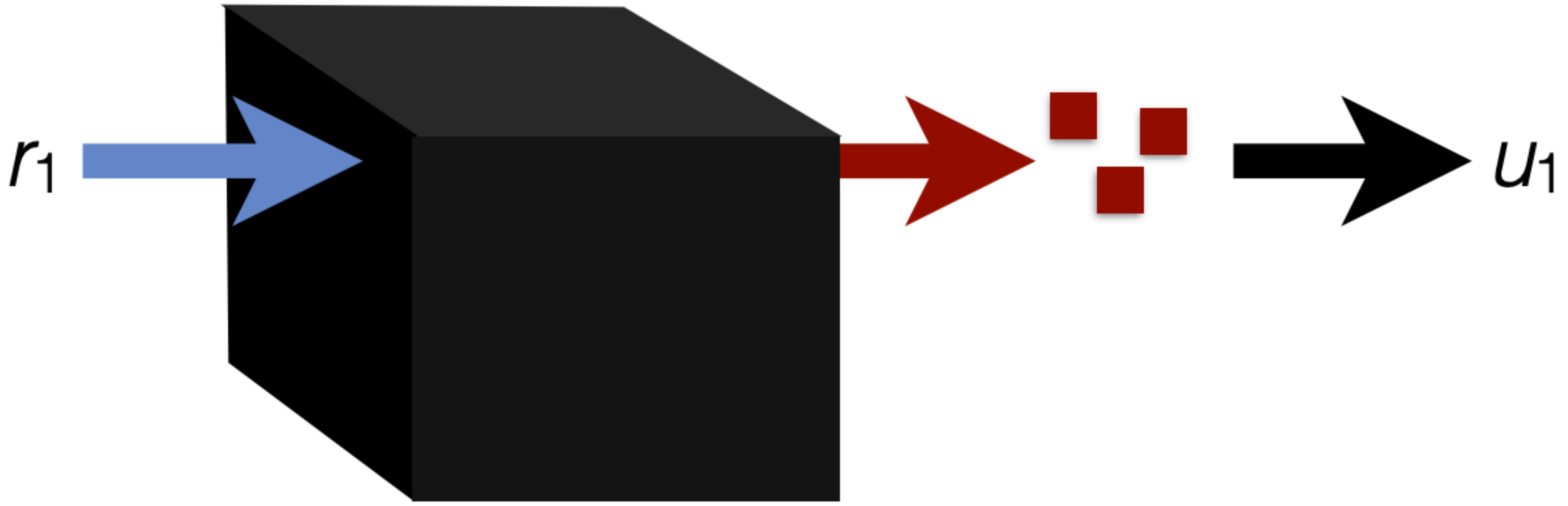

Window Size
Congestion Avoidance
Fast Retransmit
Fast Recovery
ssthresh1
ssthresh2
ssthresh3
Slow Start
Slow Start
Time
triple duplicated ACK
Timeout

PCC [NSDI'15]

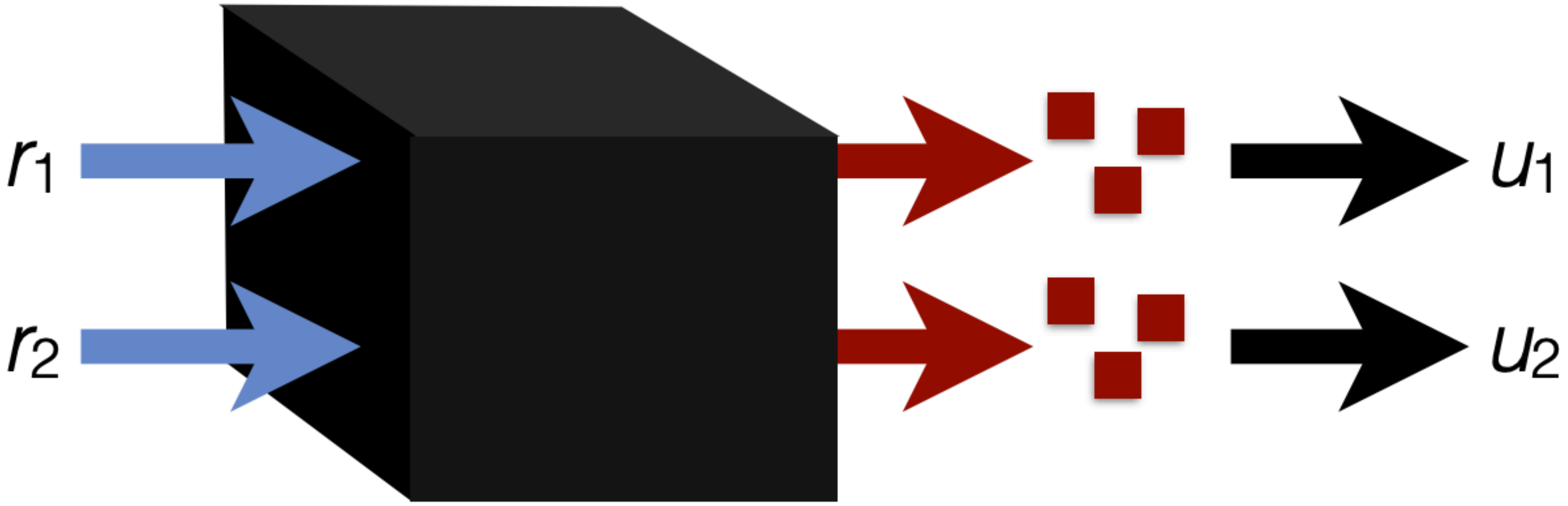
- Online solution
- Reward-based architecture
- Based on utility function



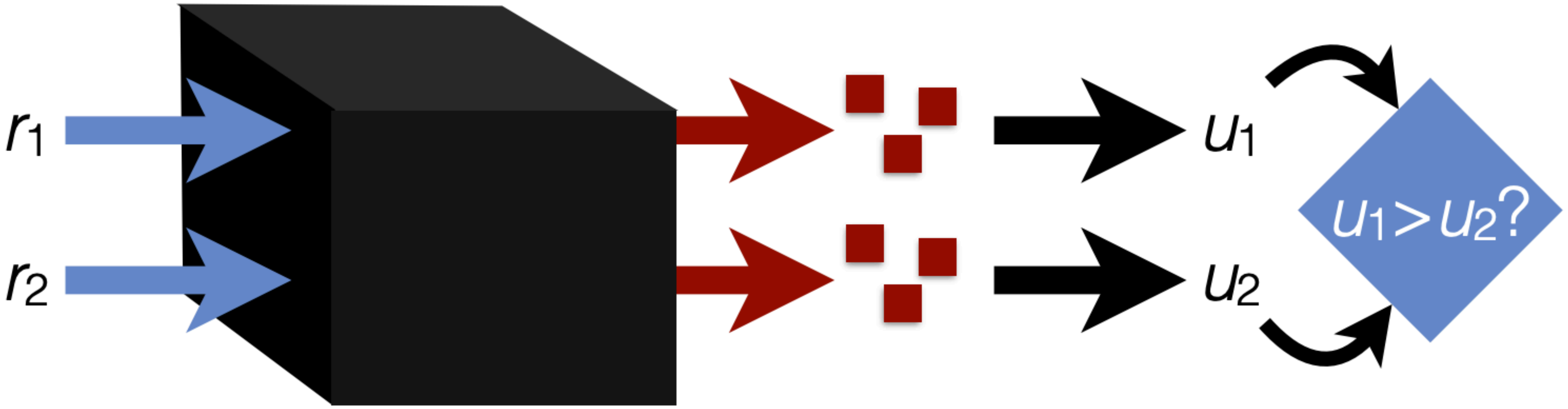
PCC (Performance-Oriented Congestion Control)



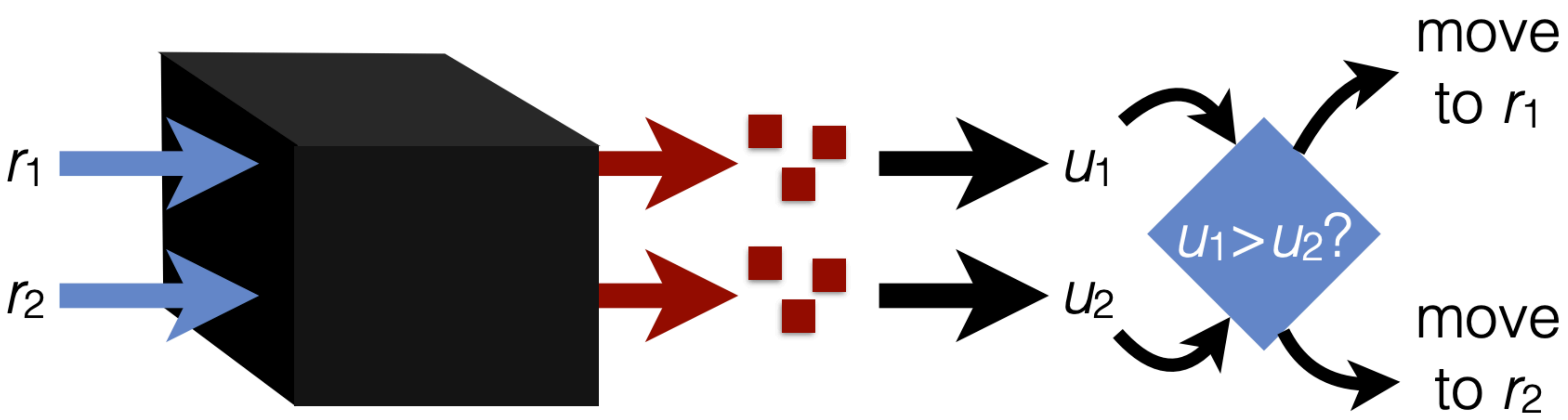
PCC (Performance-Oriented Congestion Control)



PCC (Performance-Oriented Congestion Control)

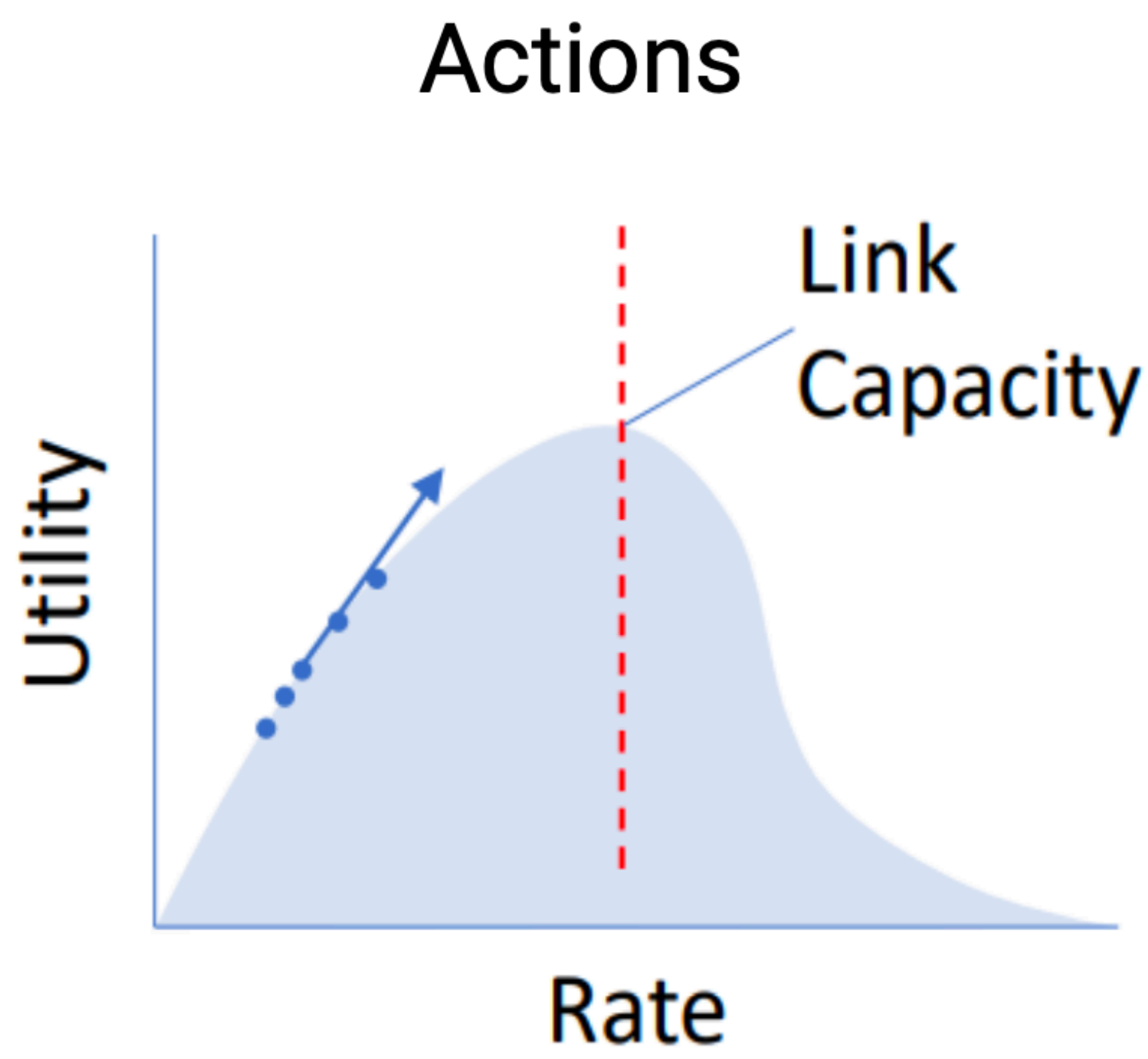


PCC (Performance-Oriented Congestion Control)



PCC Advantages

- Observes real performance
- Control based on empirical evidence
- Online learning algorithm that tracks the empirically-optimal sending rate similar to gradient ascent
- Yields consistent high performance



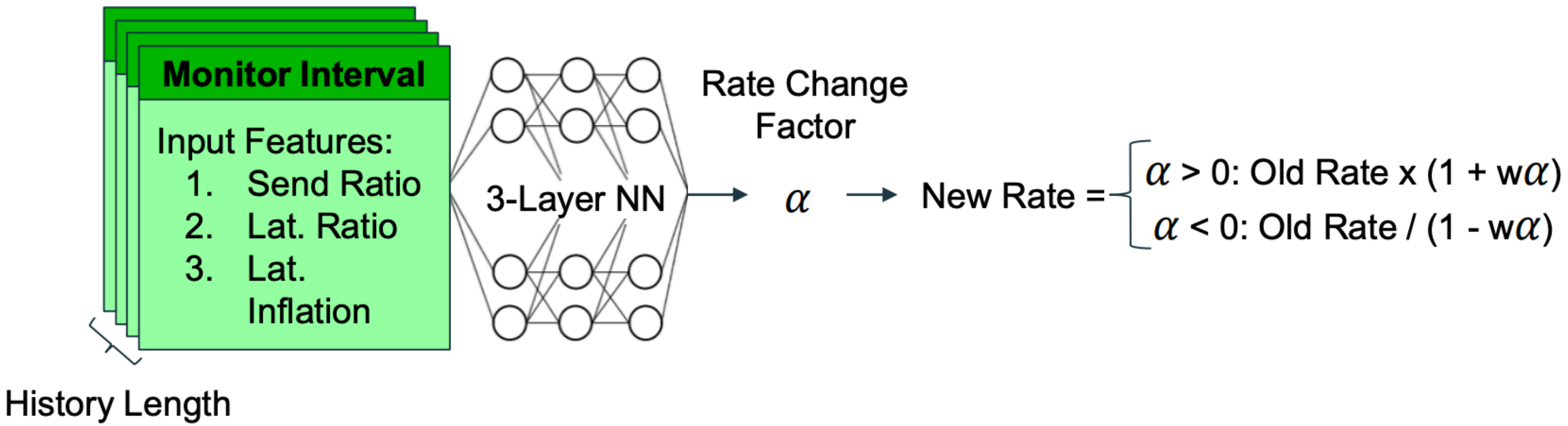
PCC Disadvantages

- Gradient descent does not work well in a highly dynamic network
- Does not adapt fast under churn

Aurora [ICML'19]

- Reinforcement Learning-based Congestion Control
- Faster adaptation than PCC

Aurora Agent Architecture



$$\text{Reward} = 10 * \text{throughput} - 1000 * \text{latency} - 2000 * \text{loss}$$

Training/Testing Environment

Training Environment:

- Simulated network
- Each episode chooses link parameters from a range:

Capacity	Latency	Loss	Queue
1 - 6mbps	50 - 500ms	0 - 5%	1 - ~3000pkt

Testing Environment:

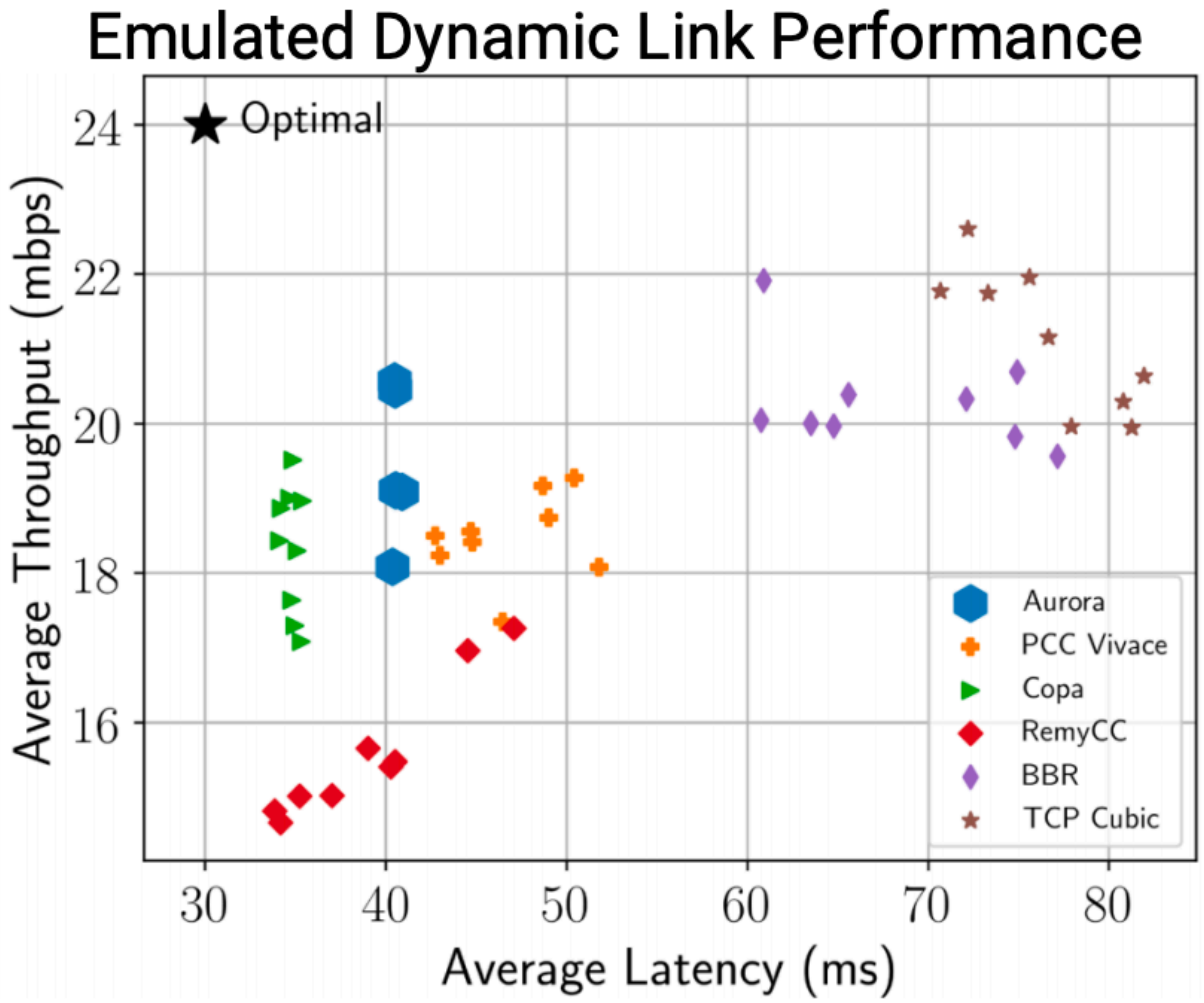
- Real packets in Linux kernel network emulation
- Much wider testing range:

Capacity	Latency	Loss	Queue
1 - 128mbps	1 - 512ms	0 - 20%	1 - 10000pkt

Experimental Results

Test Description:

- Emulated network, with real Linux kernel noise
- Time-varying link



Disadvantages of Aurora

- Arbitrary, fixed reward function. Different applications may have different performance goals

- Does not consider fairness

- Speed of adaptation in real-world networks ??

Other Directions in Learning

- Multi-agent scenarios
    - Cooperative
    - Selfish
- Online training
    - Few-shot training
    - Meta-learning
- Multi-objective Learning
    - File transfer
    - Live video

Congestion Control: Hybrid Approach

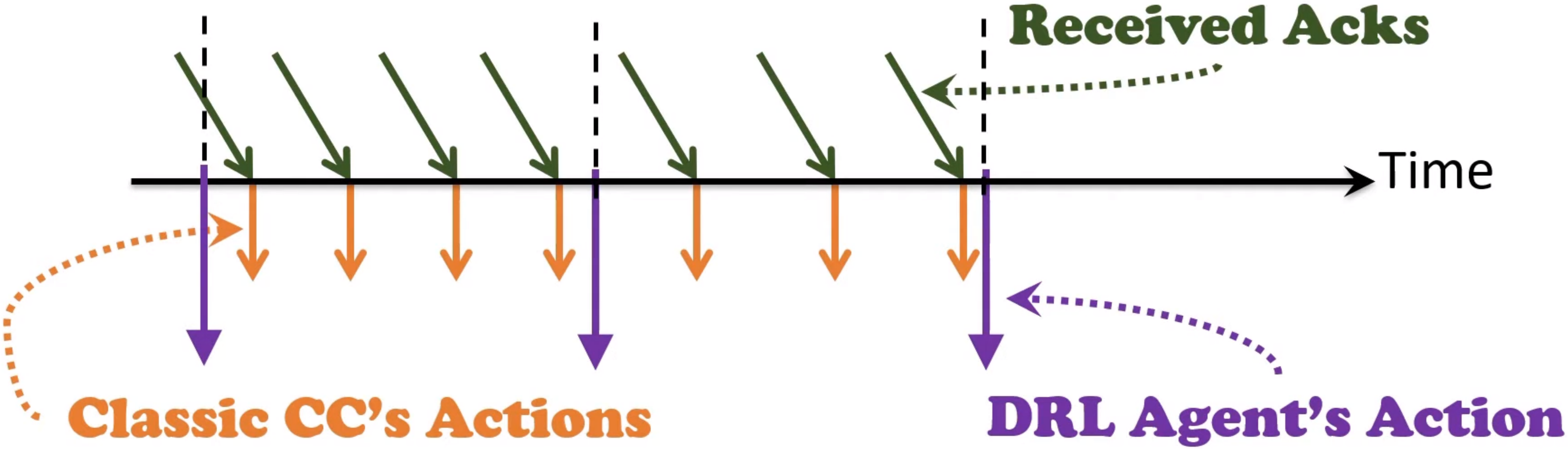
Towards Hybrid Approach

- Learning based Schemes
    - Needs time to adapt in unseen environments
    - May have high overhead in real-world settings
    - No safety guarantees
- Classic Heuristics
    - Typically lower performance
    - Difficult to design a one-size-fits-all solution

Orca: Hybrid Congestion Control [SIGCOMM'20]

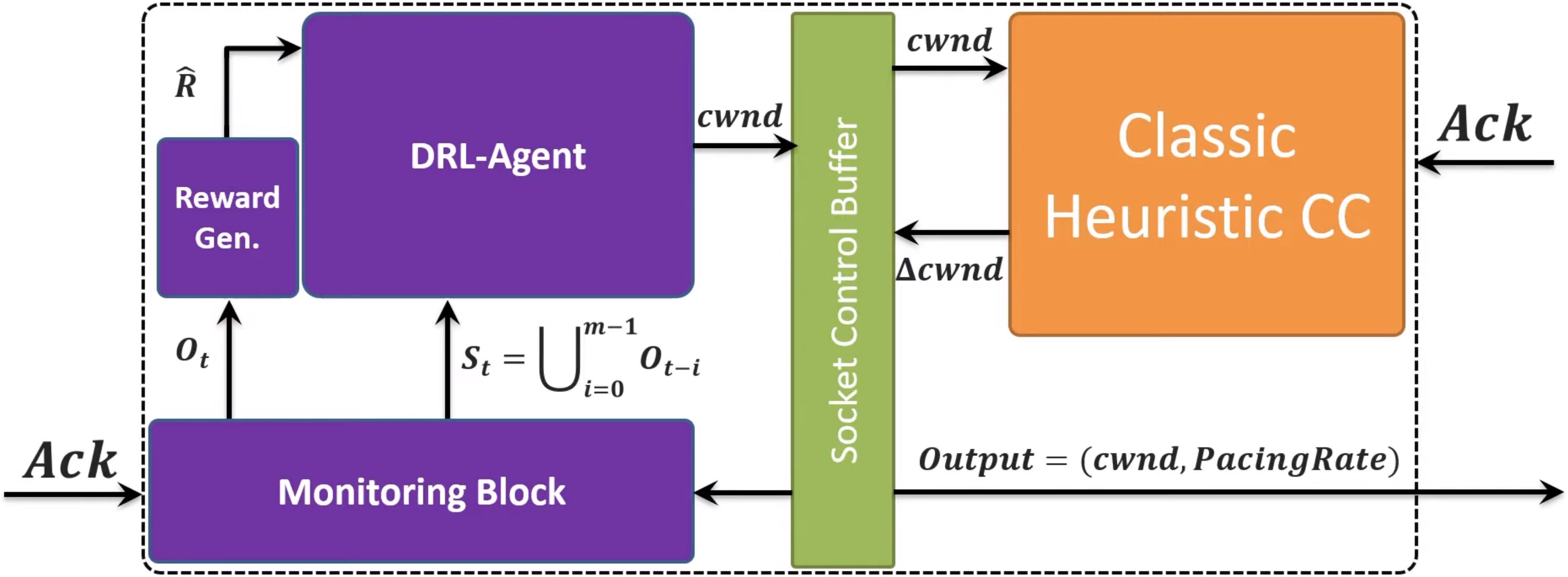
- Combining learning with classic heuristic
- Two level control hierarchy
    - Coarse-grained control using Deep RL
    - Fine-grained control using classic heuristics

Two-Level Hierarchy



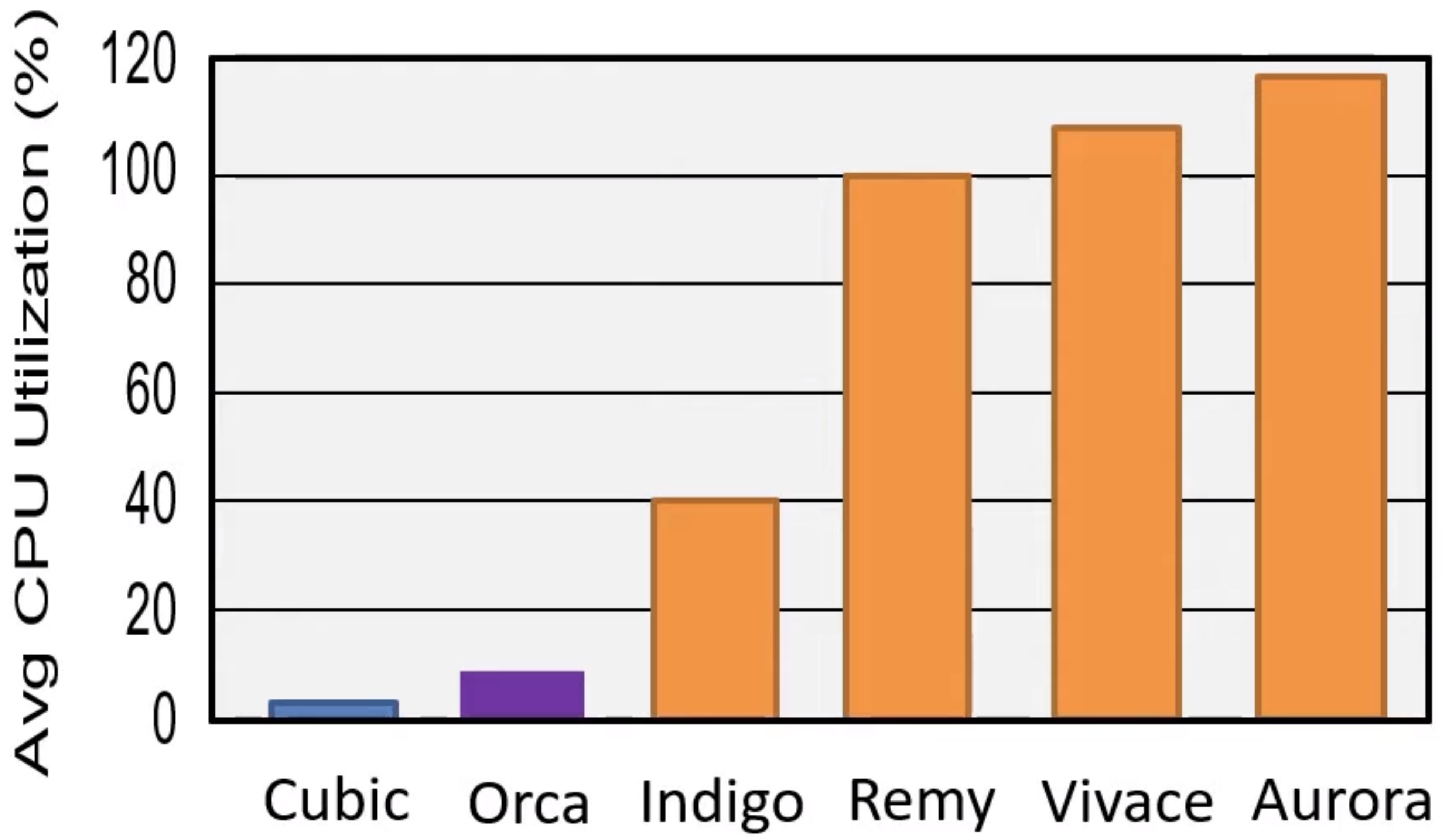
Orca System Design

$$\hat{R} = Norm.\left(\frac{Delivery\ Rate\ - \zeta \times loss}{Delay}\right)$$



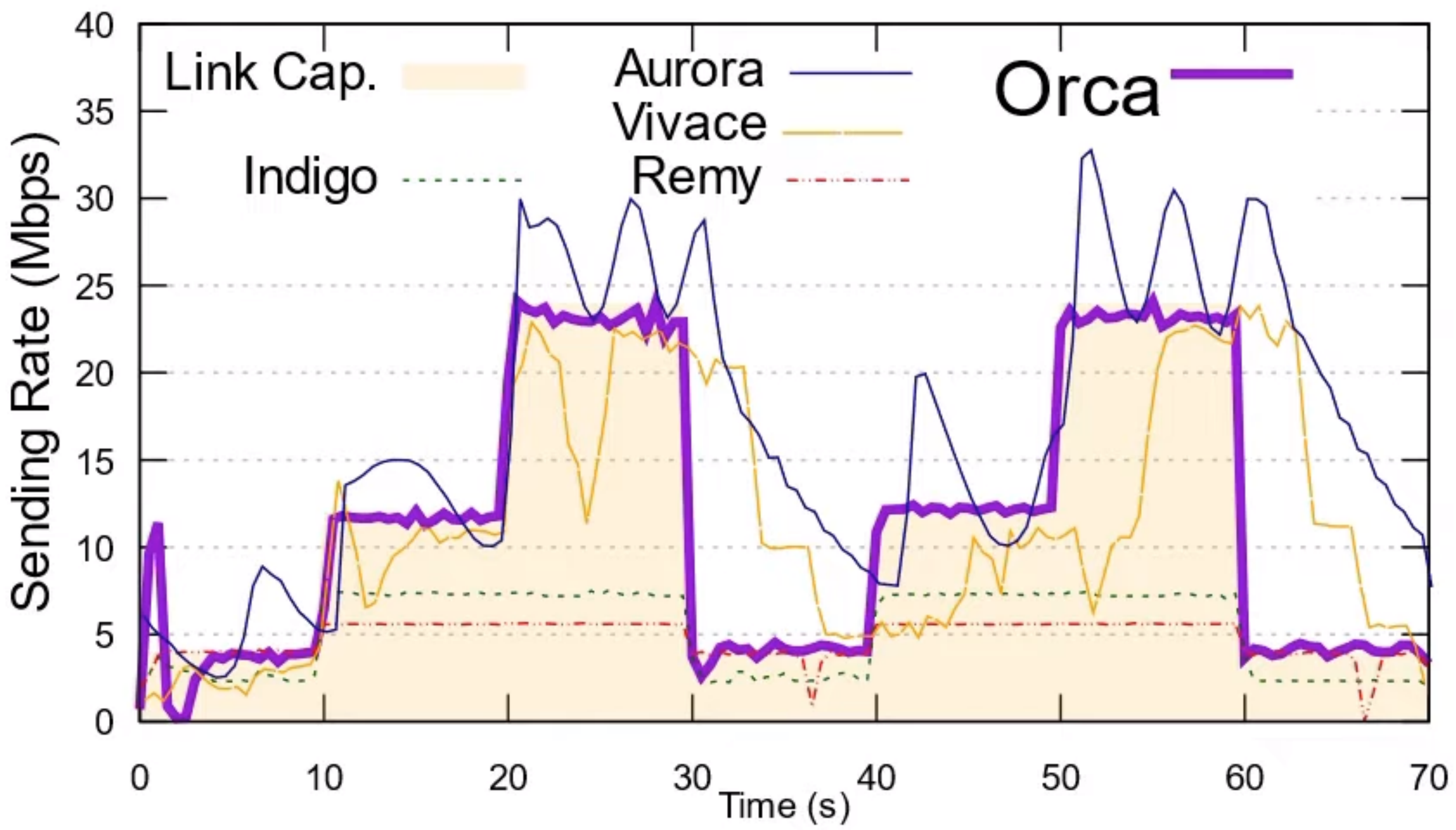
Orca Performance

Low Overhead



The Overhead

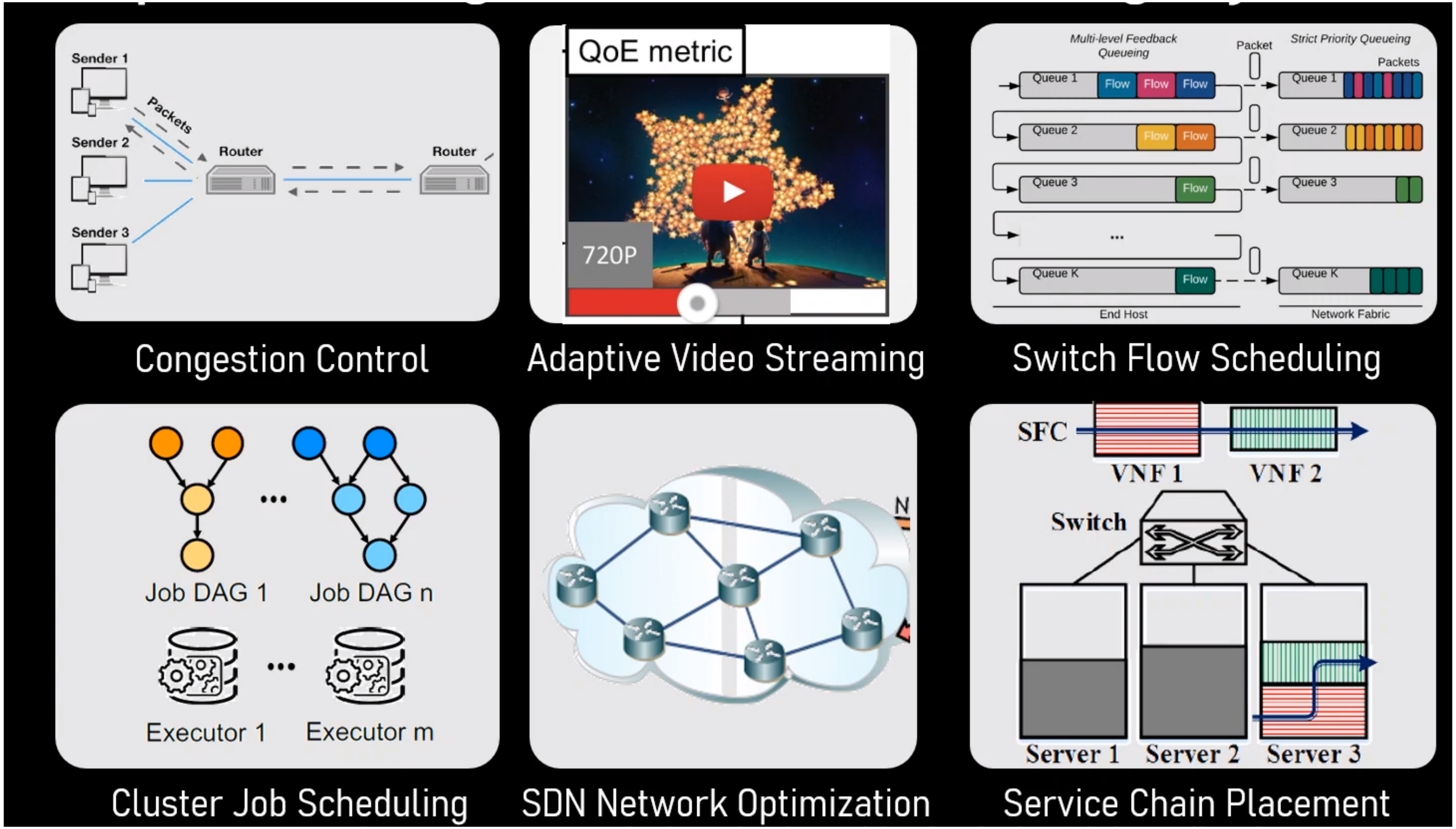
Better Dynamics



The Step Scenario

Other Applications

Deep Learning-Based Systems



Time-Series Based

- TSF Applications
    - Network flow prediction
    - Heavy-hitter detection
- Time Series Forecasting (TSF)
    - Traditional statistical analysis (e.g., ARIMA)
    - ML models (NN-based)
- Non-TSF Formulations
    - Flow size prediction (elephant/mice flows)
    - Flow count prediction

Anomaly Detection

- Applications
    - Intrusion Detection
    - Malware Detection
    - DDoS attacks
    - Phishing emails
- ML Techniques used
    - Supervised - with labeled datasets
    - Unsupervised - clustering-based techniques
    - Some RL based solutions also proposed

Thanks!