Bandwidth-aware Prefetching for Proactive Multi-video Preloading and Improved HAS Performance

Vengatanathan Krishnamoorthi[1], Niklas Carlsson[1],
Derek Eager[2], Anirban Mahanti[3], Nahid Shahmehri[1]

[1] Linköping university, Sweden
[2] University of Saskatchewan, Canada
[3] NICTA, Australia

Proc. ACM Multimedia, Brisbane, Australia, Oct. 2015

Users of the Web are very impatient and want instantaneous response for every action ...

Users of the Web are very impatient and want instantaneous response for every action …

- Loading a web page

Users of the Web are very impatient and want instantaneous response for every action ...

- Loading a web page
- Response to search query

Users of the Web are very impatient and want instantaneous response for every action ...

- Loading a web page
- Response to search query
- Start playing a video

Users of the Web are very impatient and want instantaneous response for every action ...

- Loading a web page
- Response to search query
- Start playing a video

Delays in executing these actions leads to ...

- Annoyed users

Users of the Web are very impatient and want instantaneous response for every action ...

- Loading a web page
- Response to search query
- Start playing a video

Delays in executing these actions leads to ...

- Annoyed users
- Dissatisfaction with the service and service providers

Users of the Web are very impatient and want instantaneous response for every action ...

- Loading a web page
- Response to search query
- Start playing a video

Delays in executing these actions leads to ...

- Annoyed users
- Dissatisfaction with the service and service providers
- Terminated sessions

→ Lost revenue!!

Users of the on-demand video streaming services ...

watch the beginning of several videos (~5 seconds)

before actually watching a video until the end[1].

[1]- L. Chen, Y. Zhou and D. Chiu. **A study of user behavior in online vod services.** *Computer Communications,* 2014.

Users of the on-demand video streaming services ...

watch the beginning of several videos (~5 seconds)

before actually watching a video until the end[1].

- Knowing these patterns, popular streaming services offer several related videos to chose from, based on
  - current video choice
  - user viewing history
  - popular videos in the geographical area
  - many other information sources ...

[1] - L. Chen, Y. Zhou and D. Chiu. **A study of user behavior in online vod services.** *Computer Communications*, 2014.

However, there is a startup time associated with every new video ...

However, there is a startup time associated with every new video ...

and we all know that it is annoying to wait

However, there is a startup time associated with every new video ...

and we all know that it is annoying to wait

In order to reduce startup times and improve user retention

In order to reduce startup times and improve user retention

- Effective prefetching strategies are required

In order to reduce startup times and improve user retention

- Effective prefetching strategies are required
- Alternate videos must be readily available for playback and played instantaneously

In order to reduce startup times and improve user retention

- Effective prefetching strategies are required
- Alternate videos must be readily available for playback and played instantaneously
- Prefetching must be quality-adaptive and have no negative effects on the current video's playback

In order to reduce startup times and improve user retention

- Effective prefetching strategies are required
- Alternate videos must be readily available for playback and played instantaneously
- Prefetching must be quality-adaptive and have no negative effects on the current video's playback
- These goals need to be achieved with the current state-of-the-art

Contributions

- We present a HAS-based solution that:
  - enables quality adaptive prefetching and instantaneous playback of alternative videos

Contributions

- We present a HAS-based solution that:
    - enables quality adaptive prefetching and instantaneous playback of alternative videos
    - improves the playback quality of the current video, by addressing the well known on-off problem in HAS

Contributions

- We present a HAS-based solution that:
    - enables quality adaptive prefetching and instantaneous playback of alternative videos
    - improves the playback quality of the current video, by addressing the well known on-off problem in HAS
    - ensures stall free playback of the current video with improved playback experience

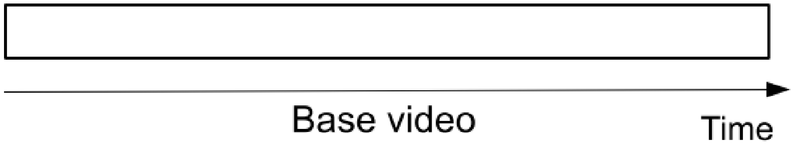
Contributions

- We present a HAS-based solution that:
    - enables quality adaptive prefetching and instantaneous playback of alternative videos
    - improves the playback quality of the current video, by addressing the well known on-off problem in HAS
    - ensures stall free playback of the current video with improved playback experience
- Our policy classes captures a diverse set of use cases

Contributions

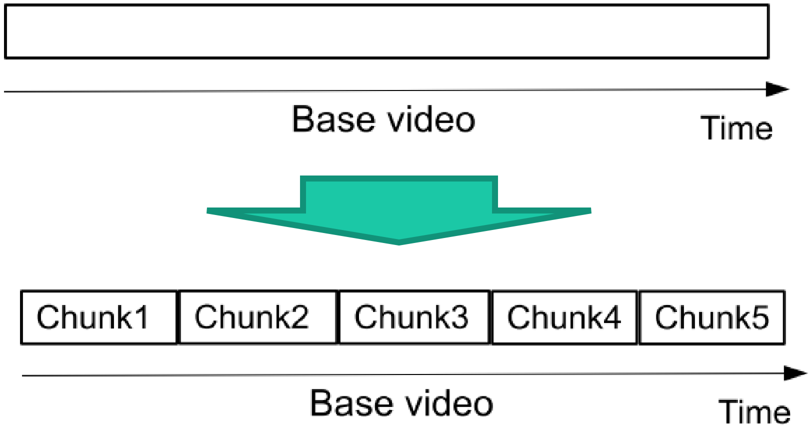
- We present a HAS-based solution that:
    - enables quality adaptive prefetching and instantaneous playback of alternative videos
    - improves the playback quality of the current video, by addressing the well known on-off problem in HAS
    - ensures stall free playback of the current video with improved playback experience
- Our policy classes captures a diverse set of use cases
- We characterize and show the benefits of our prefetching policies through our proof-of-concept implementation

HTTP-based Adaptive Streaming (HAS)



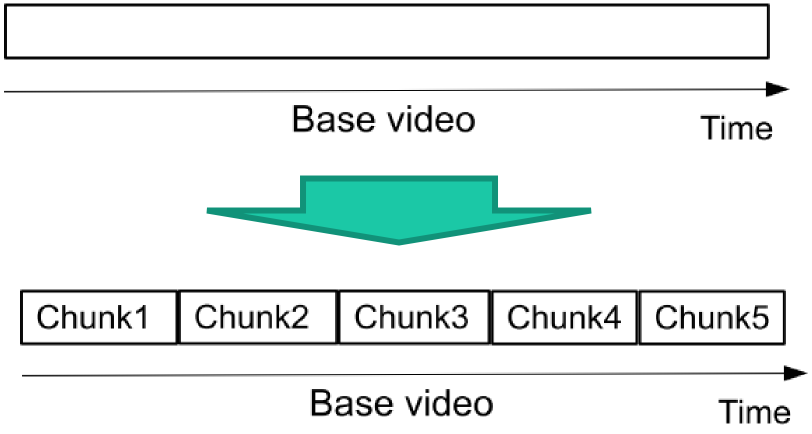
- HTTP-based streaming

HTTP-based Adaptive Streaming (HAS)



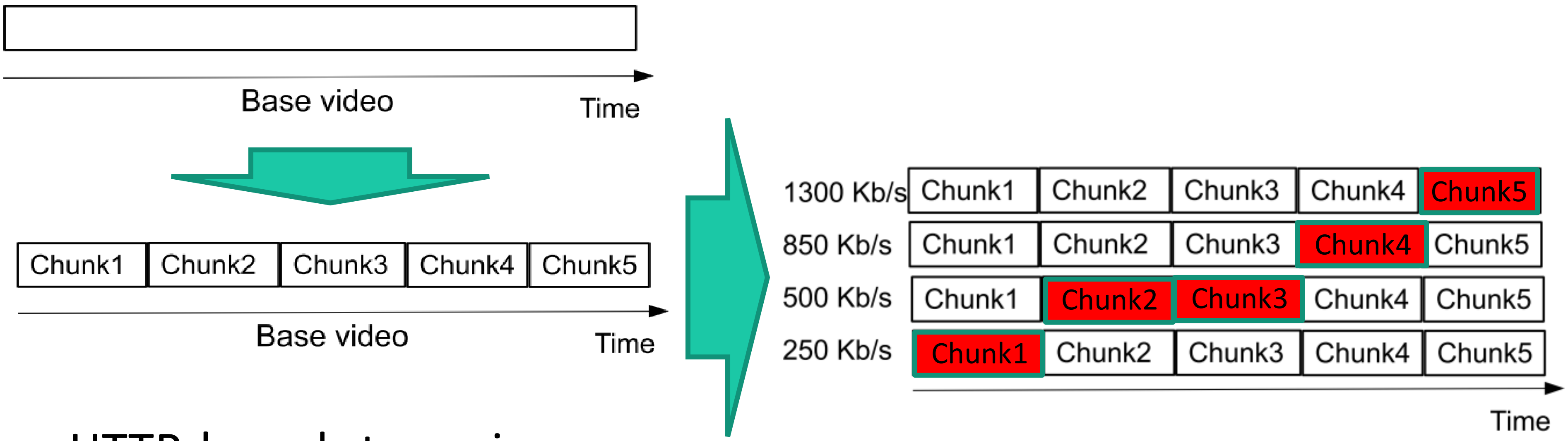
- HTTP-based streaming
  - Video is split into chunks

HTTP-based Adaptive Streaming (HAS)



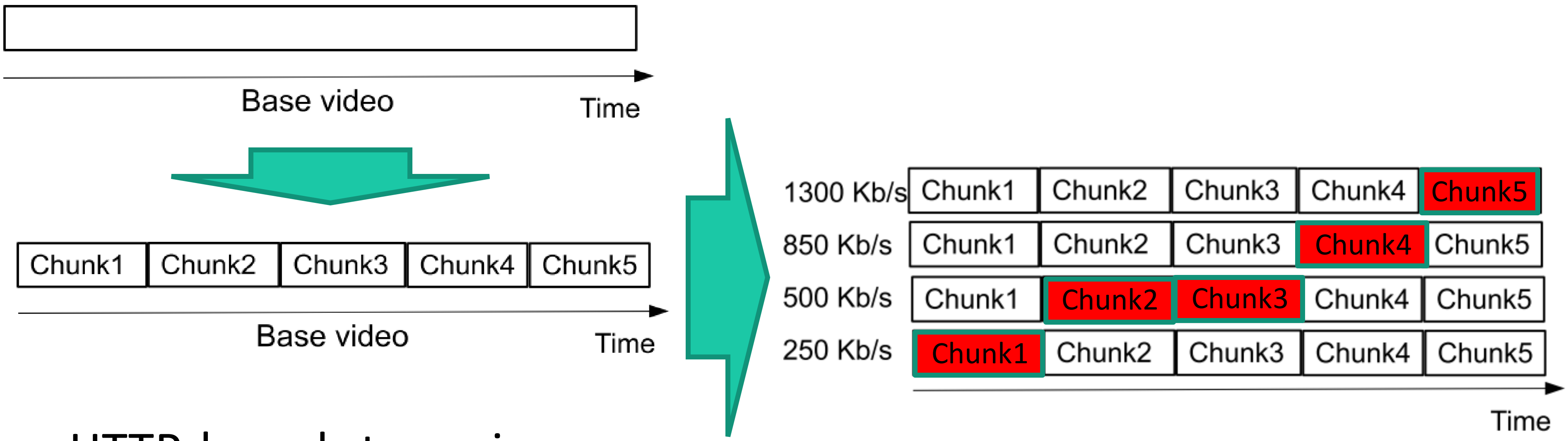
- HTTP-based streaming
  - Video is split into chunks
  - Easy firewall traversal and caching

HTTP-based Adaptive Streaming (HAS)



- HTTP-based streaming
  - Video is split into chunks
  - Easy firewall traversal and caching
- HTTP-based adaptive streaming
  - Clients adapt quality encoding based on buffer/network conditions

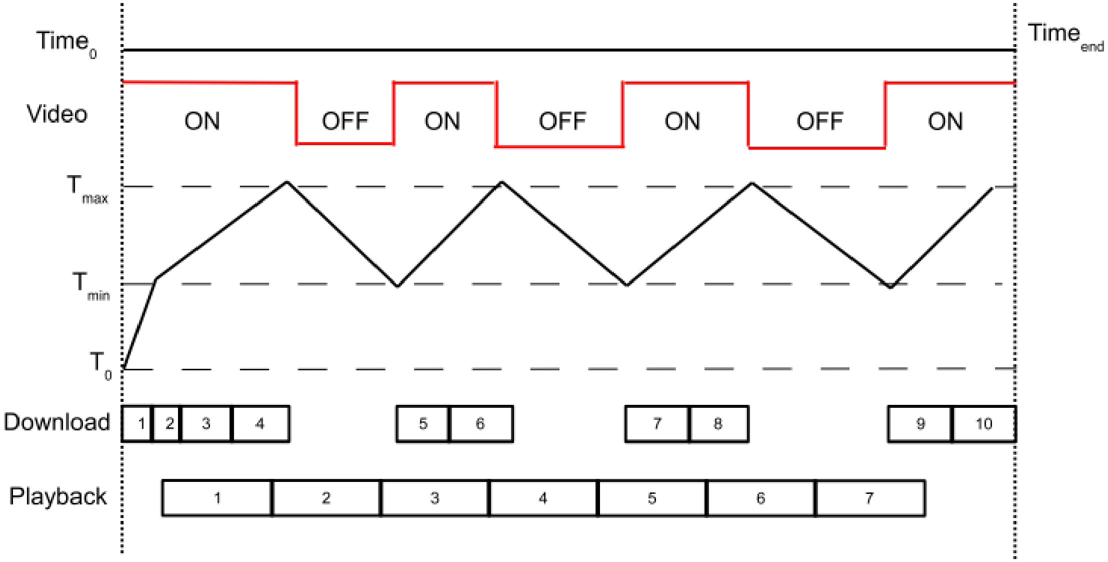
HTTP-based Adaptive Streaming (HAS)



- HTTP-based streaming
  - Video is split into chunks
  - Easy firewall traversal and caching
- HTTP-based adaptive streaming
  - Clients adapt quality encoding based on buffer/network conditions
  - Support for interactive VoD

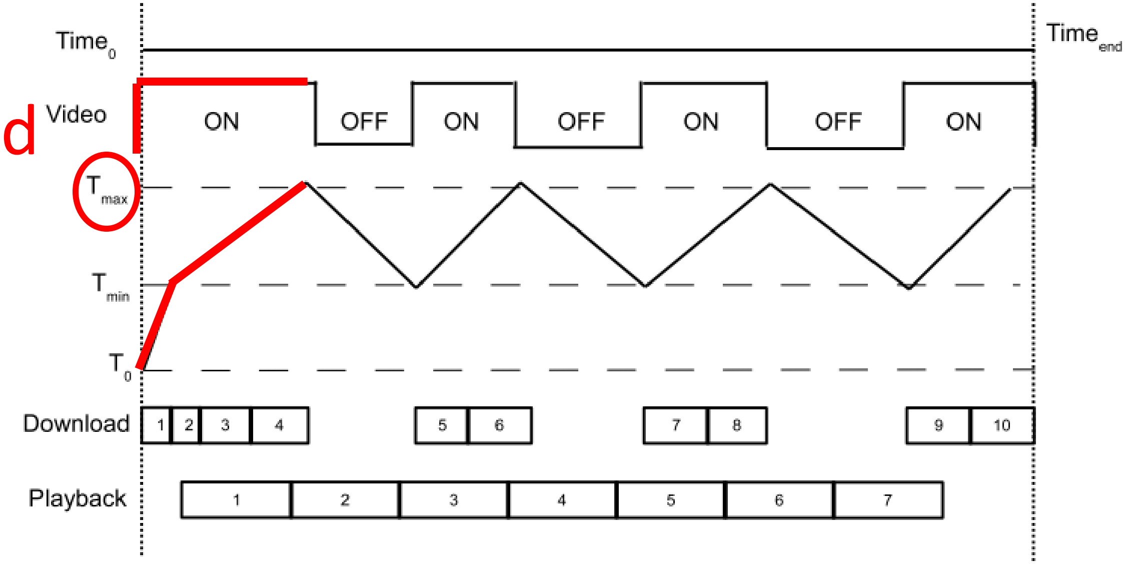
On-off switching in HAS

- Most HAS players perform ON-OFF switching based on two buffer thresholds: T_{min} and T_{max}



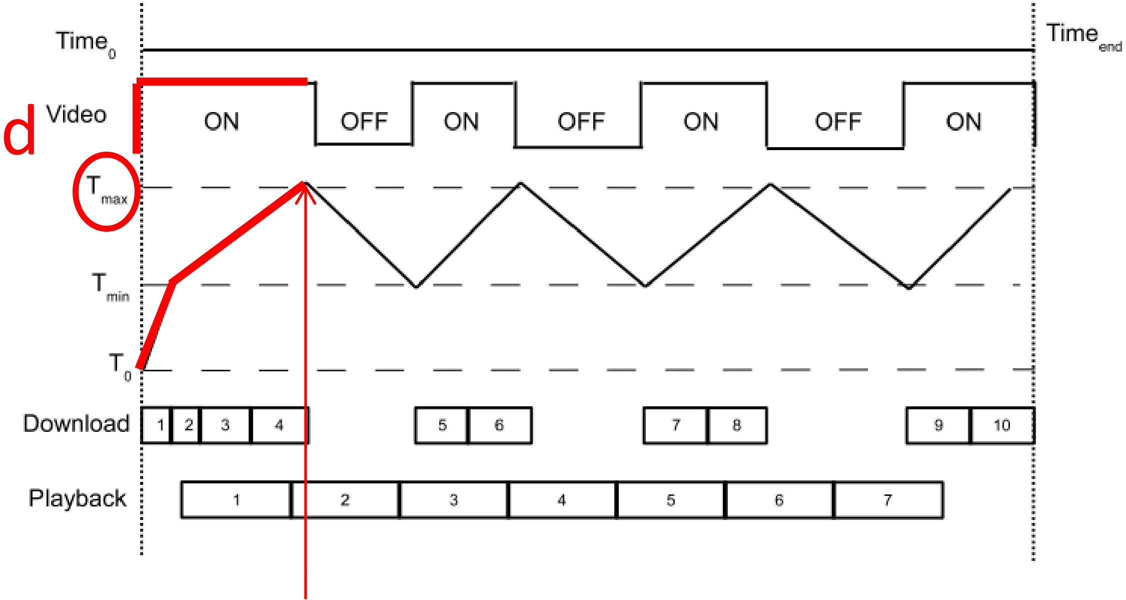
On-off switching in HAS

- Most HAS players perform ON-OFF switching based on two buffer thresholds: T_{min} and T_{max}
- If buffer > T_{max}
  - Suspend download



On-off switching in HAS

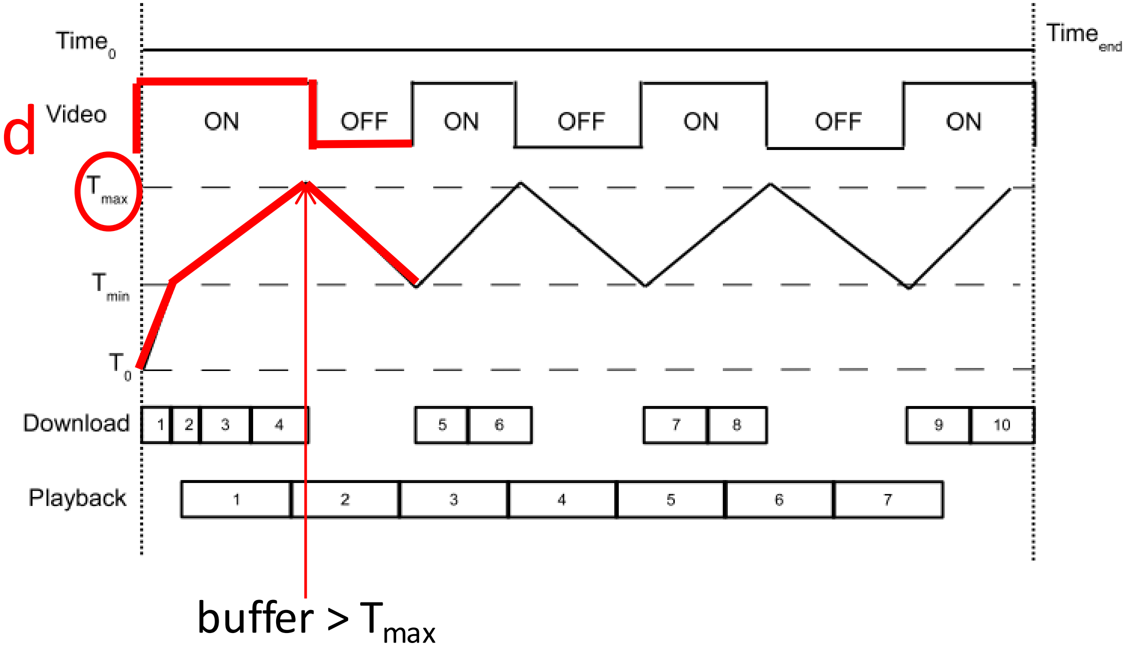
- Most HAS players perform ON-OFF switching based on two buffer thresholds: T_{min} and T_{max}
- If buffer > T_{max}
  - Suspend download



buffer > T_{max}

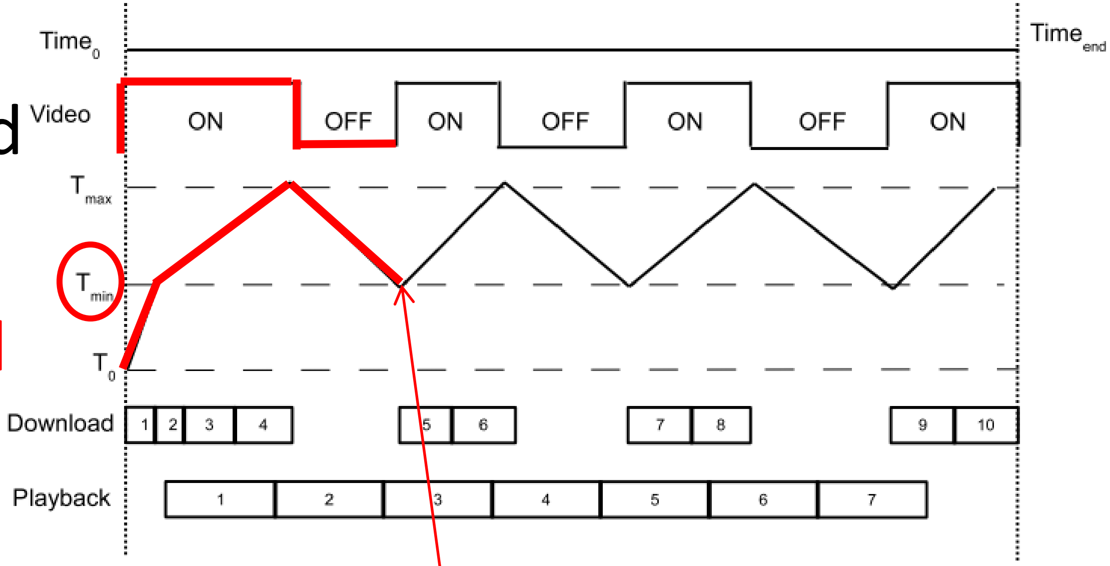
On-off switching in HAS

- Most HAS players perform ON-OFF switching based on two buffer thresholds: T_{min} and T_{max}
- If buffer > T_{max}
  - Suspend download



On-off switching in HAS

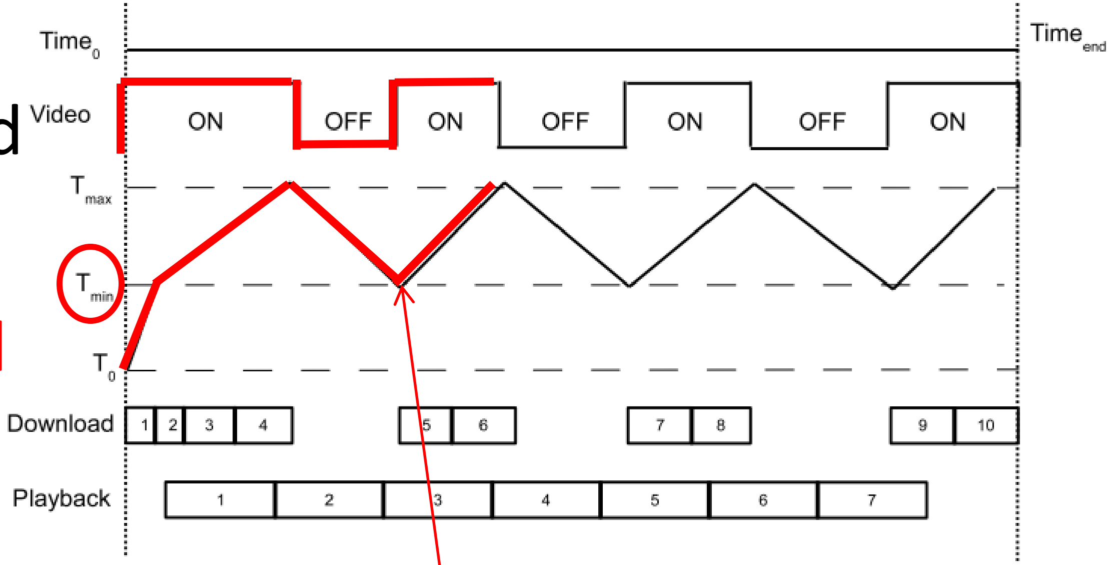
- Most HAS players perform ON-OFF switching based on two buffer thresholds: T_{min} and T_{max}
- If buffer > T_{max}
  - Suspend download
- If buffer < T_{min}
  - Resume download



buffer < T_{min}

On-off switching in HAS

- Most HAS players perform ON-OFF switching based on two buffer thresholds: T_{min} and T_{max}
- If buffer > T_{max}
  - – Suspend download
- If buffer < T_{min}
  - – Resume download



buffer < T_{min}

Issues with on-off switching in HAS

- Although thresholds on the buffer is beneficial, on-off switching has been shown to lead to:

Issues with on-off switching in HAS

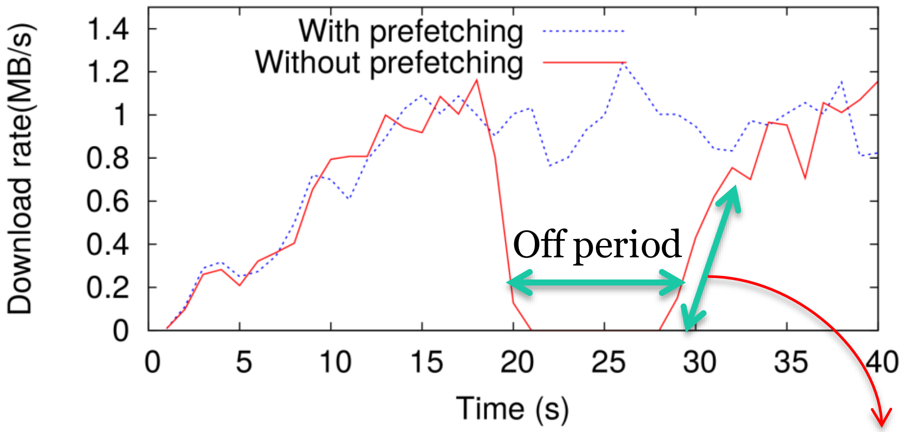
- Although thresholds on the buffer is beneficial, on-off switching has been shown to lead to:
  - Unfair bandwidth allocation
  - Under utilization of bandwidth
  - Unnecessary fluctuations in quality adaptation

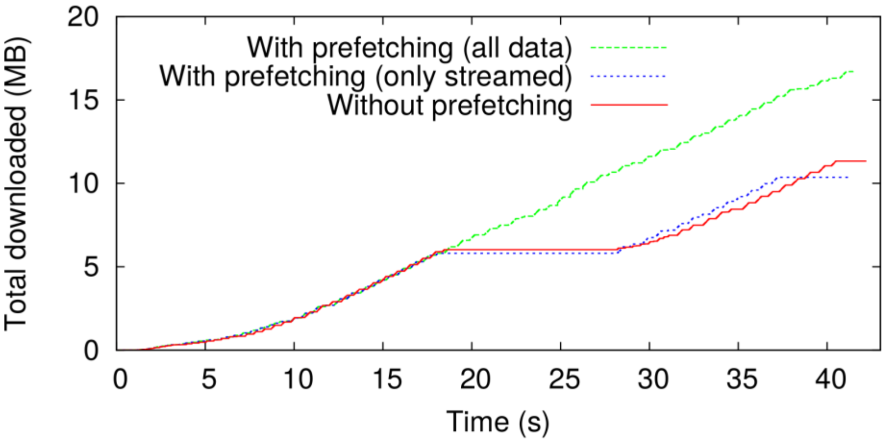
Prefetch alternative videos during off periods

- Allow instantaneous playback of alternative videos
- In addition, prefetching during off periods:

Prefetch alternative videos during off periods

- Allow instantaneous playback of alternative videos
- In addition, prefetching during off periods:
  - Avoids the need to ramp-up from slow-start

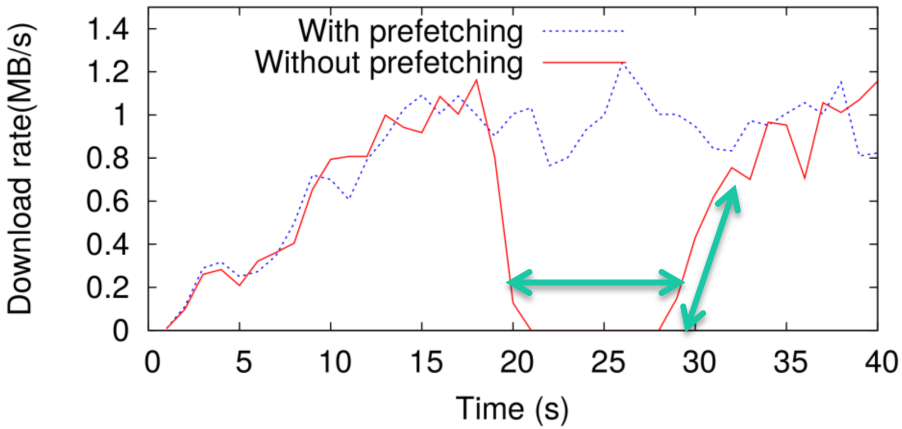


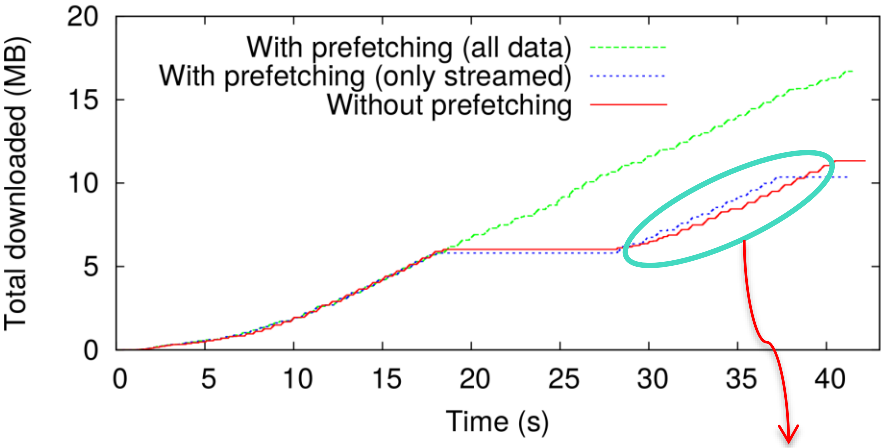


Slow-start and ramp up

Prefetch alternative videos during off periods

- Allow instantaneous playback of alternative videos
- In addition, prefetching during off periods:
  - Avoids the need to ramp-up from slow-start
  - Client remains active throughout the duration



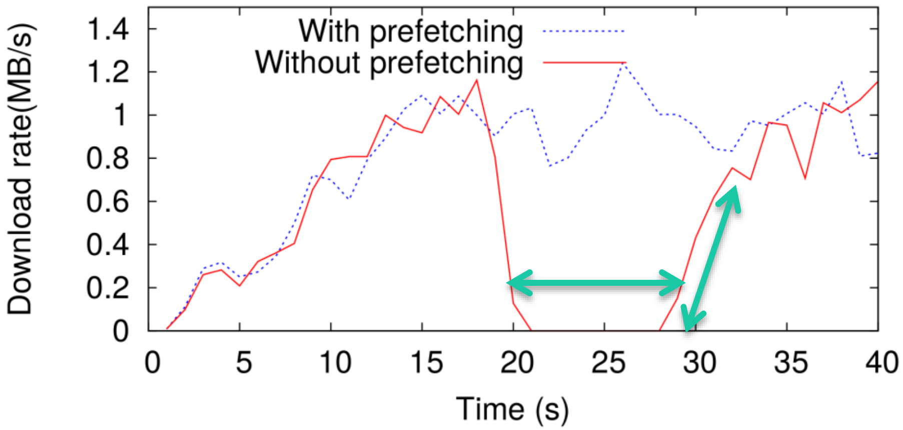


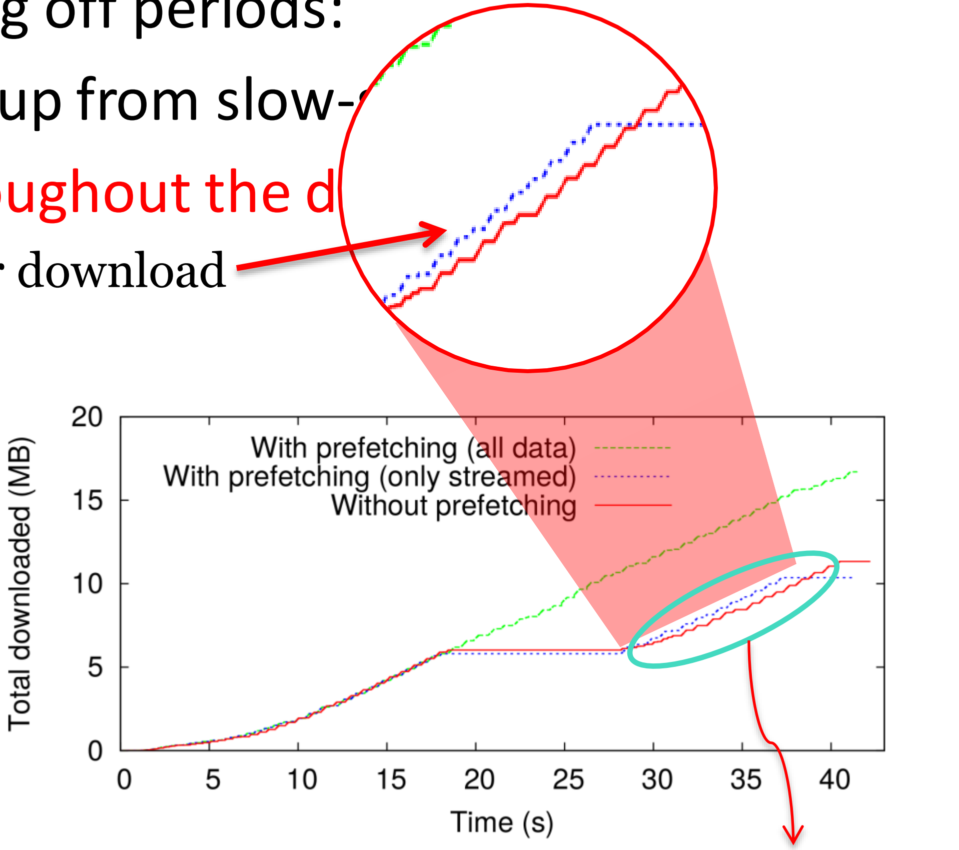
With prefetching, data is downloaded faster and the next off period is reached sooner

Prefetch alternative videos during off periods

- Allow instantaneous playback of alternative videos
- In addition, prefetching during off periods:
  - Avoids the need to ramp-up from slow-
  - Client remains active throughout the d

Greater slope → faster download





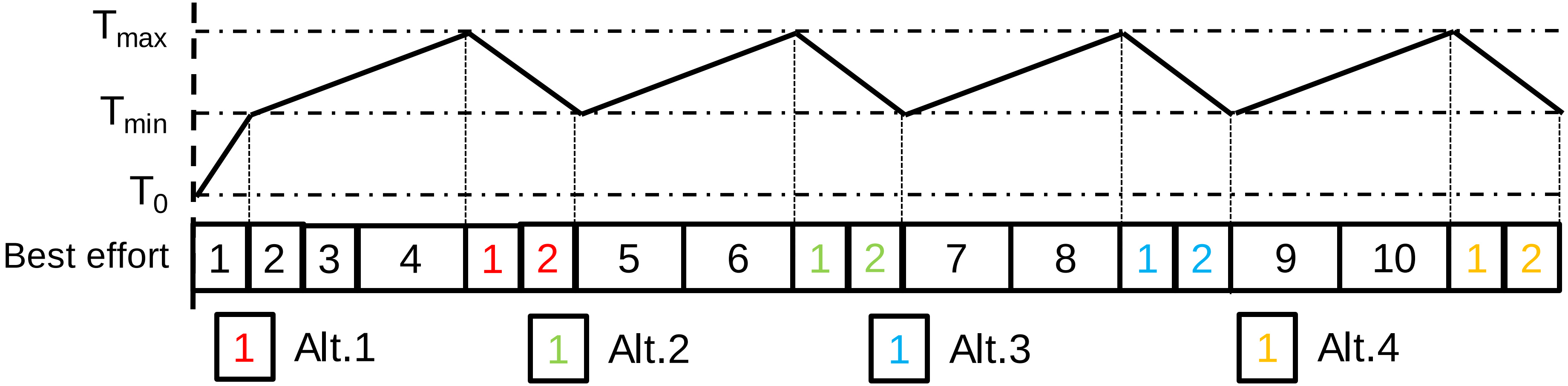
With prefetching, data is downloaded faster and the next off period is reached sooner

Prefetching policies

- In order to control the number of prefetched chunks and the time at which alternate videos will be available for playback, we consider three broad classes of prefetching policies:
  - Best-effort
  - Token-based
  - Deadline-based

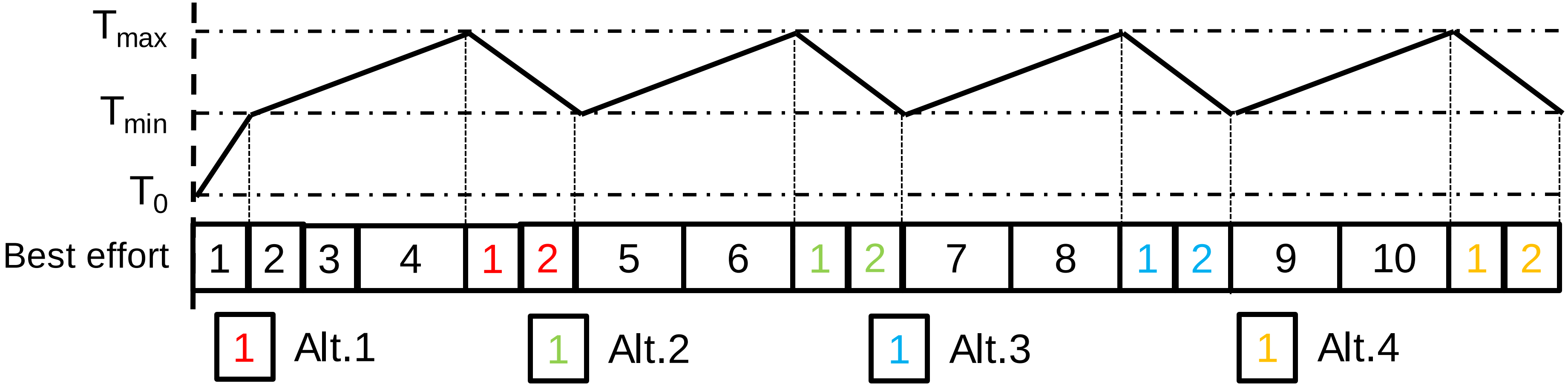
Prefetching policy: Best-effort

- Prefetching rules:
  - Prefetch alternative chunks when $T \geq T_{max}$ and $r(T - T_{min})$ > prefetched chunk size



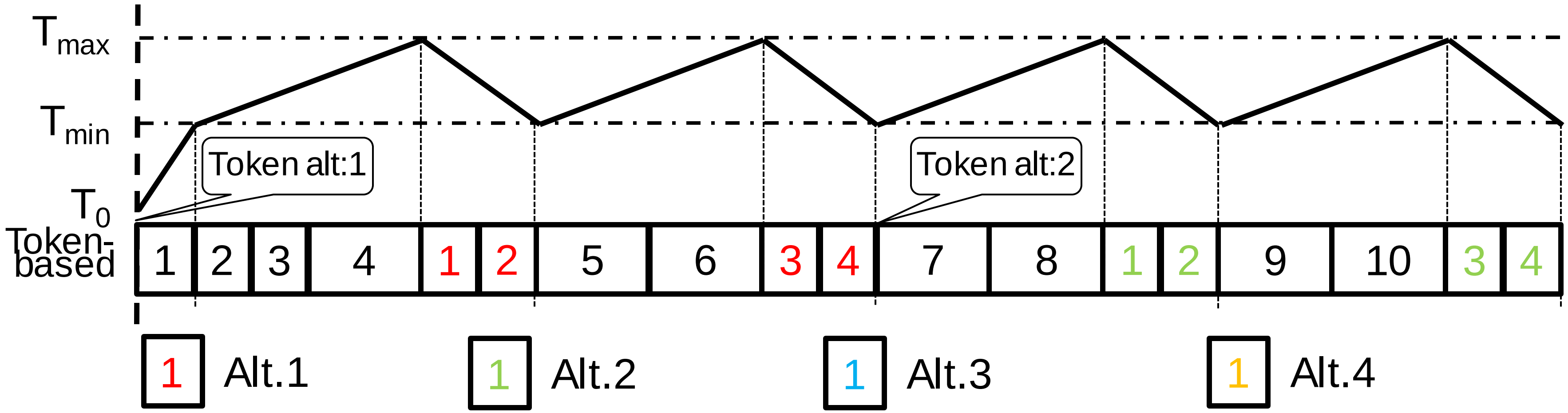
Prefetching policy: Best-effort

- Prefetching rules:
  - Prefetch alternative chunks when $T \geq T_{max}$ and $r(T - T_{min})$ > prefetched chunk size
  - Number of chunks per alternate video is controlled by parameter ‘*n*’



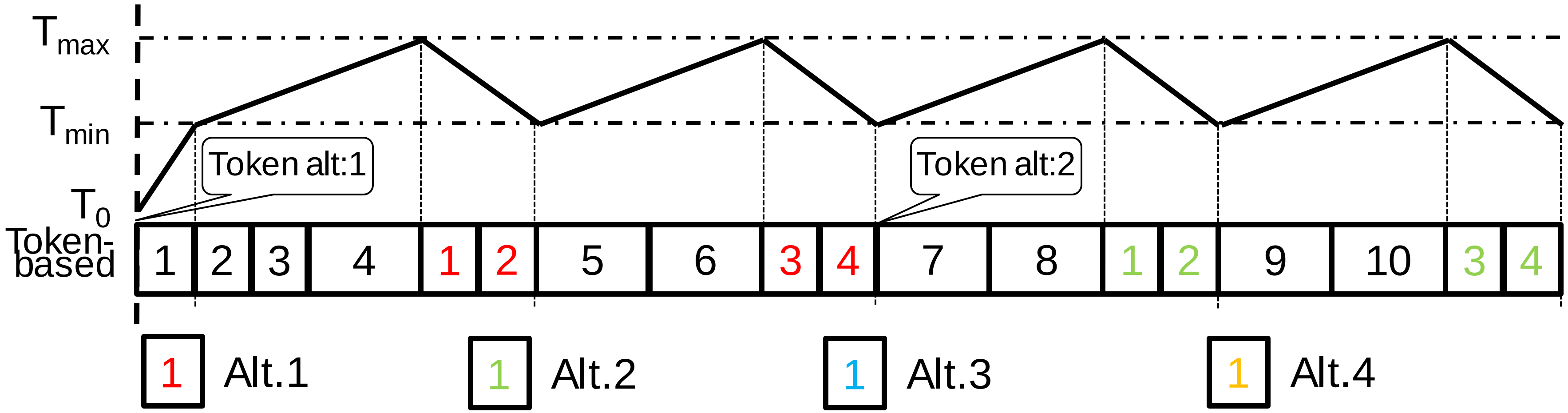
Prefetching policy: Token-based

- Prefetching rules:
  - Prefetch alternative chunks when $T \geq T_{max}$ and $r(T - T_{min})$ > prefetched chunk size



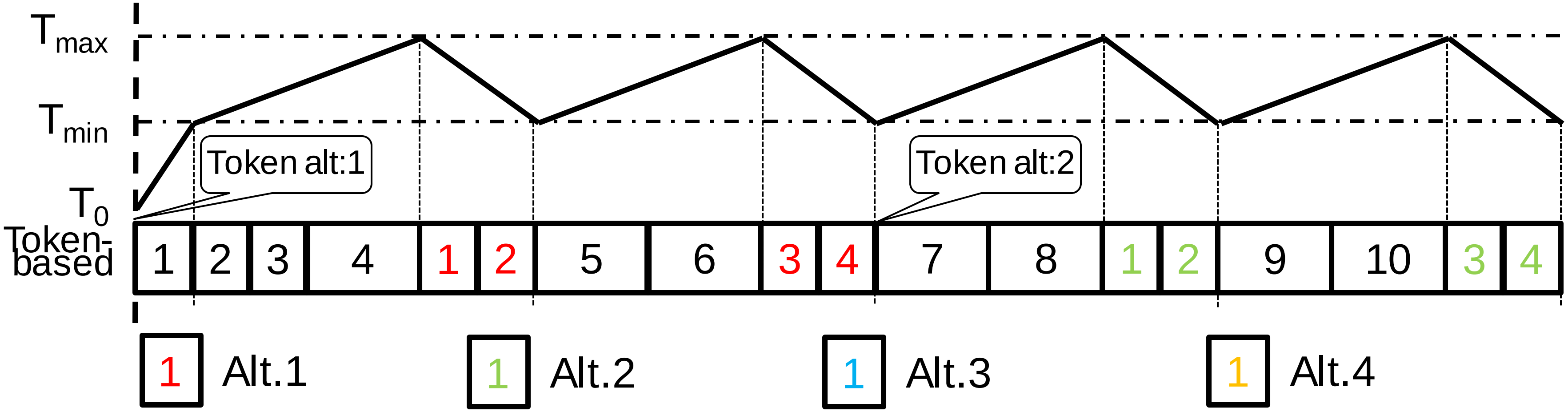
Prefetching policy: Token-based

- Prefetching rules:
  - Prefetch alternative chunks when $T \geq T_{max}$ and $r(T - T_{min})$ > prefetched chunk size
  - Token determines which alternative video to prefetch



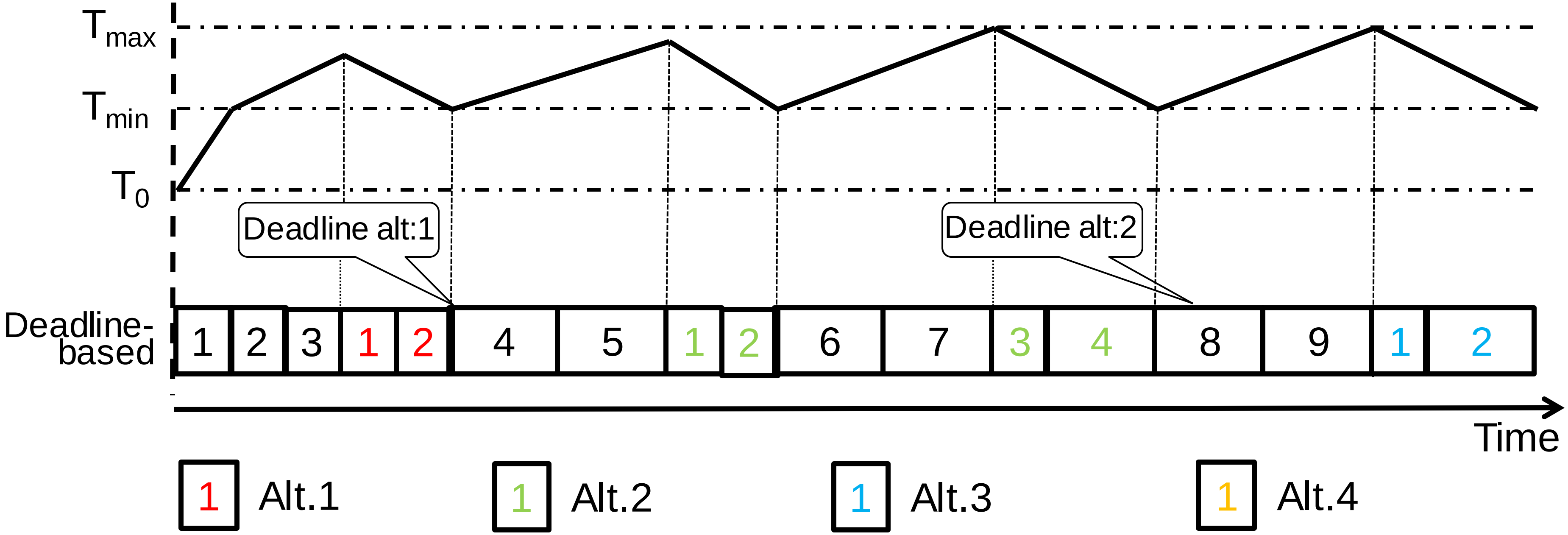
Prefetching policy: Token-based

- Prefetching rules:
  - Prefetch alternative chunks when $T \geq T_{max}$ and $r(T - T_{min})$ > prefetched chunk size
  - Token determines which alternative video to prefetch
  - Time Δ determines time between prefetching of alternative videos



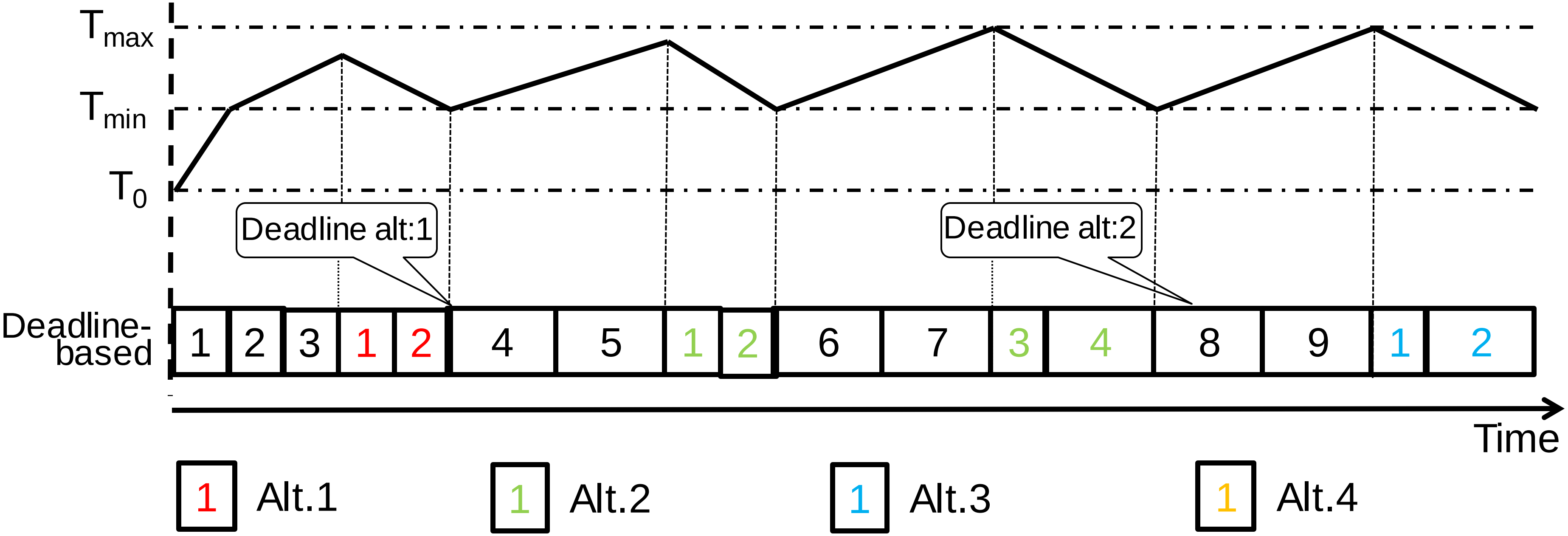
Prefetching policy: Deadline-based

- Prefetching rules:
  - Strict deadlines by which ‘*n*’ chunks of alternative videos (and ‘*m*’ chunks of current video) must be downloaded



Prefetching policy: Deadline-based

- Prefetching rules:
  - Strict deadlines by which '*n*' chunks of alternative videos (and '*m*' chunks of current video) must be downloaded
  - Quality of the streaming video is adapted to satisfy the deadlines based on an optimization framework



Prefetching policy: Deadline-based

- Prefetching rules:
  - Strict deadlines by which ‘*n*’ chunks of alternative videos (and ‘*m*’ chunks of current video) must be downloaded
  - Quality of the streaming video is adapted to satisfy the deadlines based on an optimization framework

$$\text{maximize} \sum_{i=1}^{m_a+1} q_i^s l_i^s + \sum_{j=1}^{n} q_j^a l_j^a$$

Playback quality of streamed video

Playback quality alternative video chunks

The optimization formulation above originally appeared in our ACM MM 2014 paper.

Quality-adaptive Prefetching for Interactive Branched Video using HTTP-based Adaptive Streaming

In Proceedings of the ACM International Conference on Multimedia (ACM Multimedia), Orlando, FL, Nov. 2014.

Policy characterization

- All experiments performed with at least one competing flow
  - Generated from a large file download from a second server
- Results are averages over 20 experiments

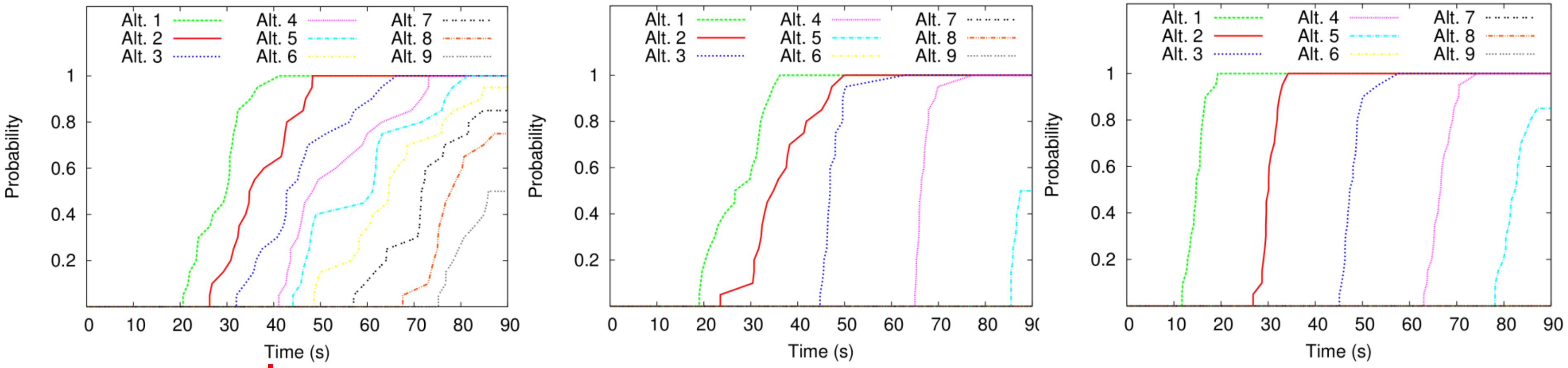
Policy characterization

- All experiments performed with at least one competing flow
  - Generated from a large file download from a second server
- Results are averages over 20 experiments

- Example results
  - 4 Mbps (shared) link
  - 1 competing flow
  - 150ms RTT
  - $T_{min}/T_{max} = 8/12$
  - *n* (chunks to prefetch per alternative video) = 2

LINKÖPING UNIVERSITY

Policy characterization: Number of alternative videos

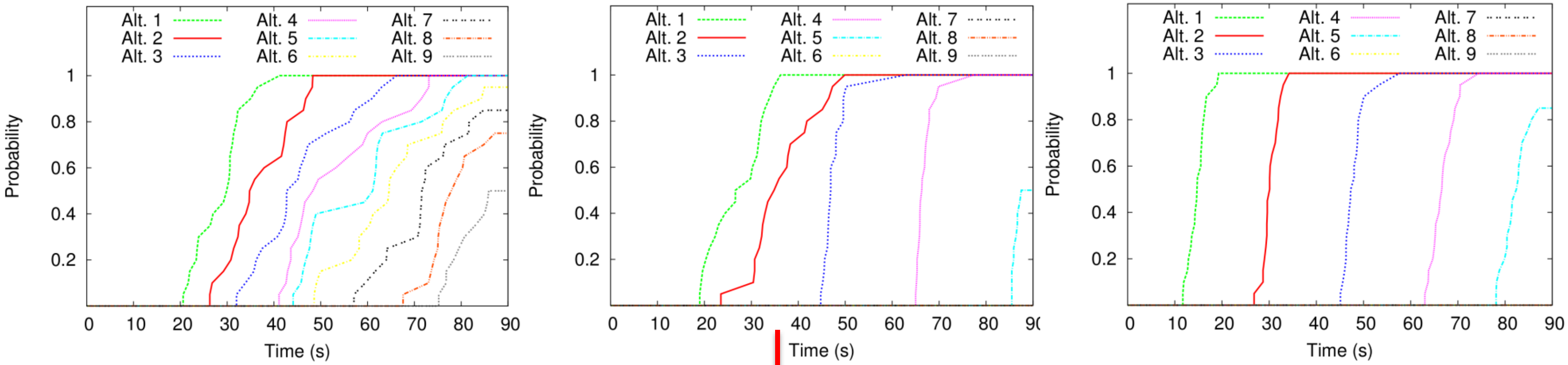


- Best-effort policy
  - Moves to the next alternative video after 2 chunks of the previous alternative video is completed

Figures: Download completion time of the *n*=2 chunks of each alternative video

LINKÖPING UNIVERSITY

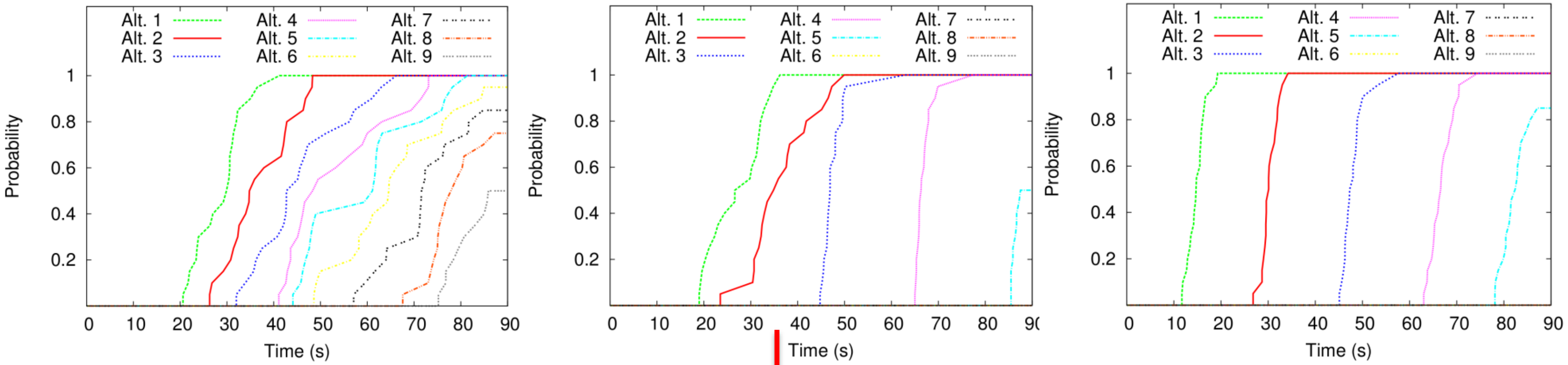
Policy characterization: Number of alternative videos



- Token-based policy
  - Moves to the next alternative video only when the next token is released

Figures: Download completion time of the *n*=2 chunks of each alternative video

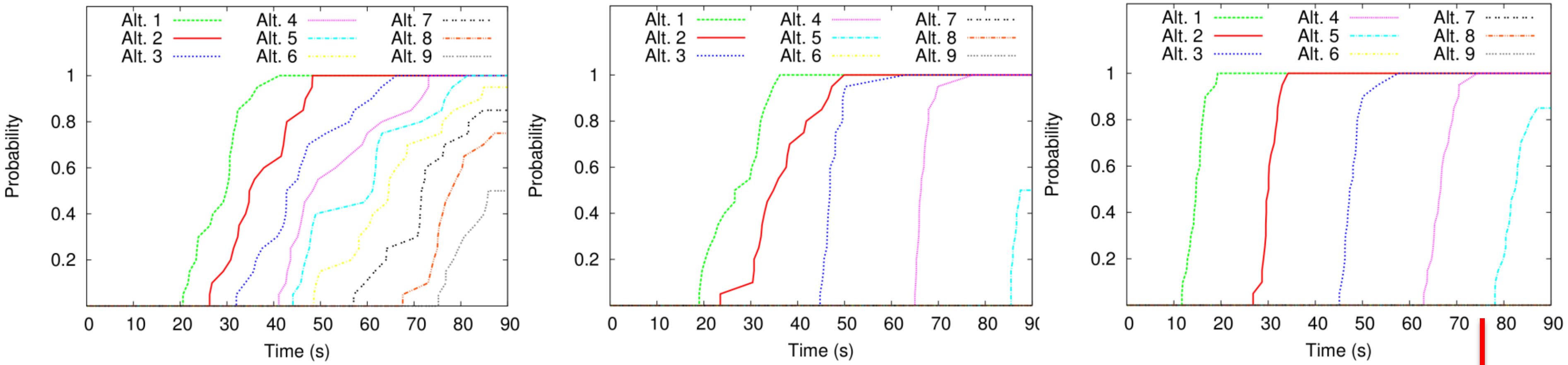
Policy characterization: Number of alternative videos



- Token-based policy
  - Moves to the next alternative video only when the next token is released
  - Prefetches more chunks of alternative videos in the absence of new tokens

Figures: Download completion time of the *n*=2 chunks of each alternative video

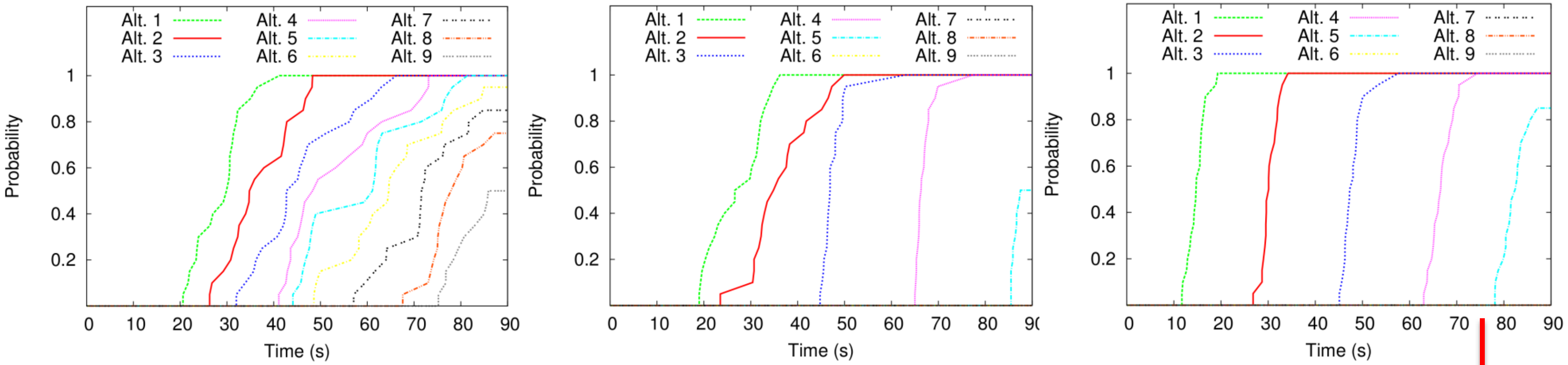
Policy characterization: Number of alternative videos



- Deadline-based policy
  - Deadlines every 20s

Figures: Download completion time of the *n*=2 chunks of each alternative video

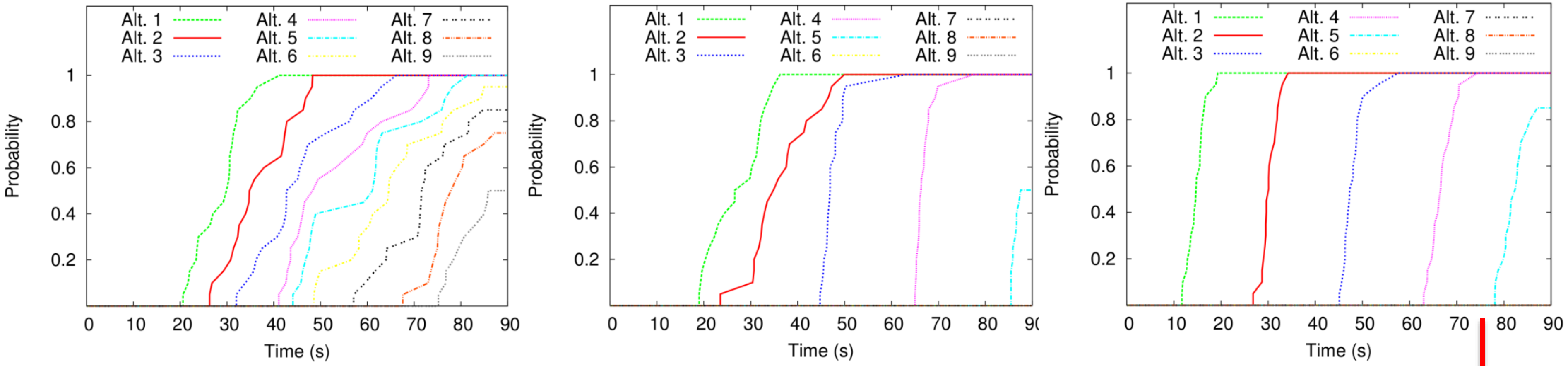
Policy characterization: Number of alternative videos



- Deadline-based policy
  - Deadlines every 20s
  - Evenly spaced download completions, respecting their download deadlines

Figures: Download completion time of the *n*=2 chunks of each alternative video

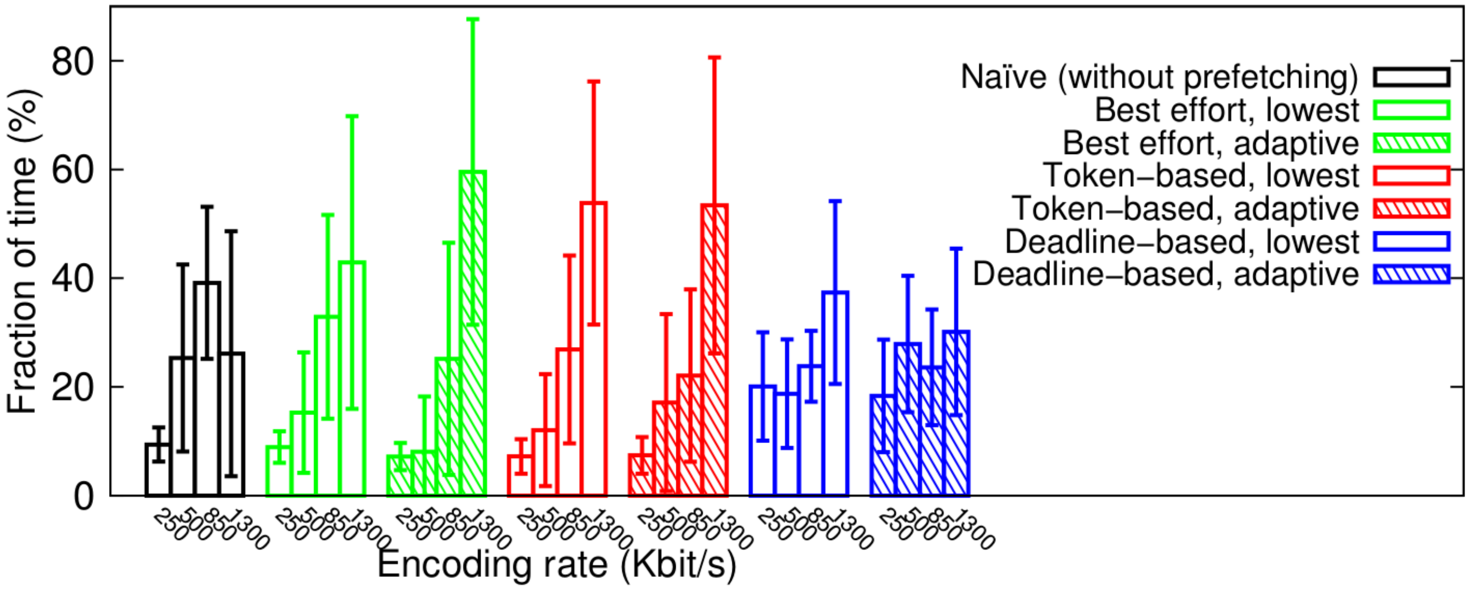
Policy characterization: Number of alternative videos



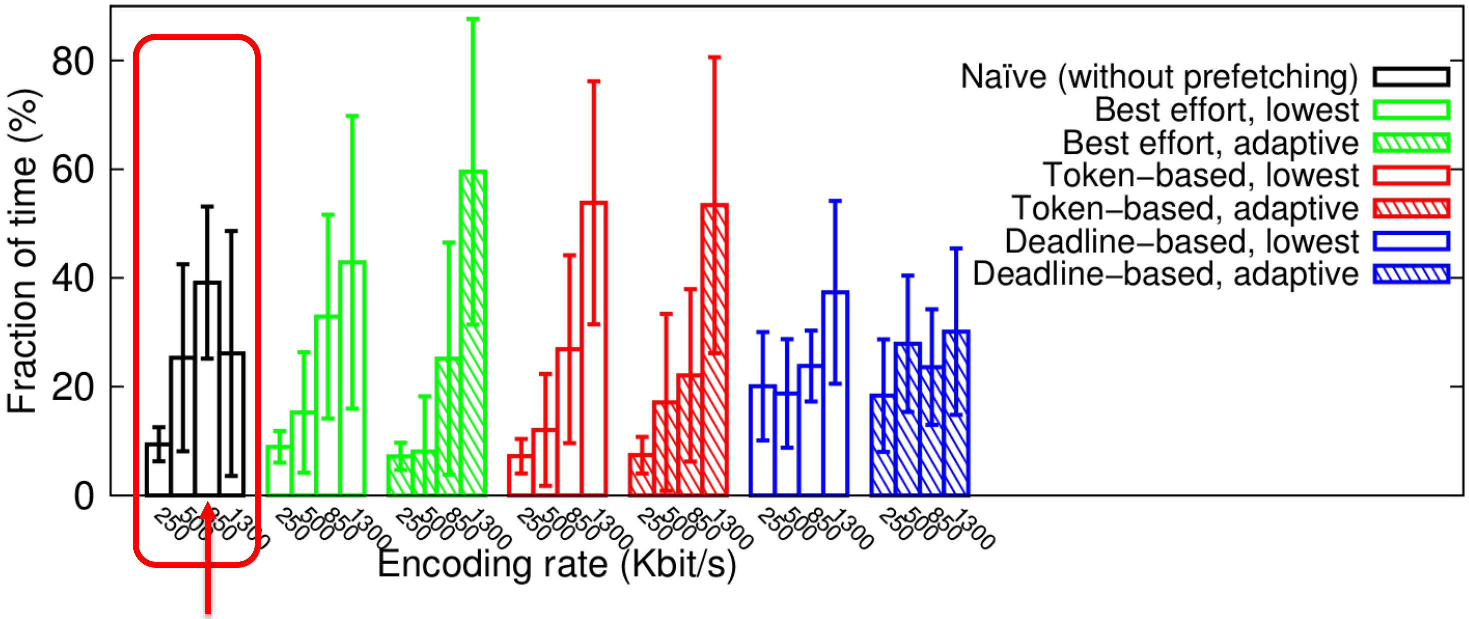
- Deadline-based policy
  - Deadlines every 20s
  - Evenly spaced download completions, respecting their download deadlines
  - When the deadline is satisfied, the player moves ahead with the next deadline

Figures: Download completion time of the *n*=2 chunks of each alternative video

Policy characterization: Playback quality of streamed video chunks

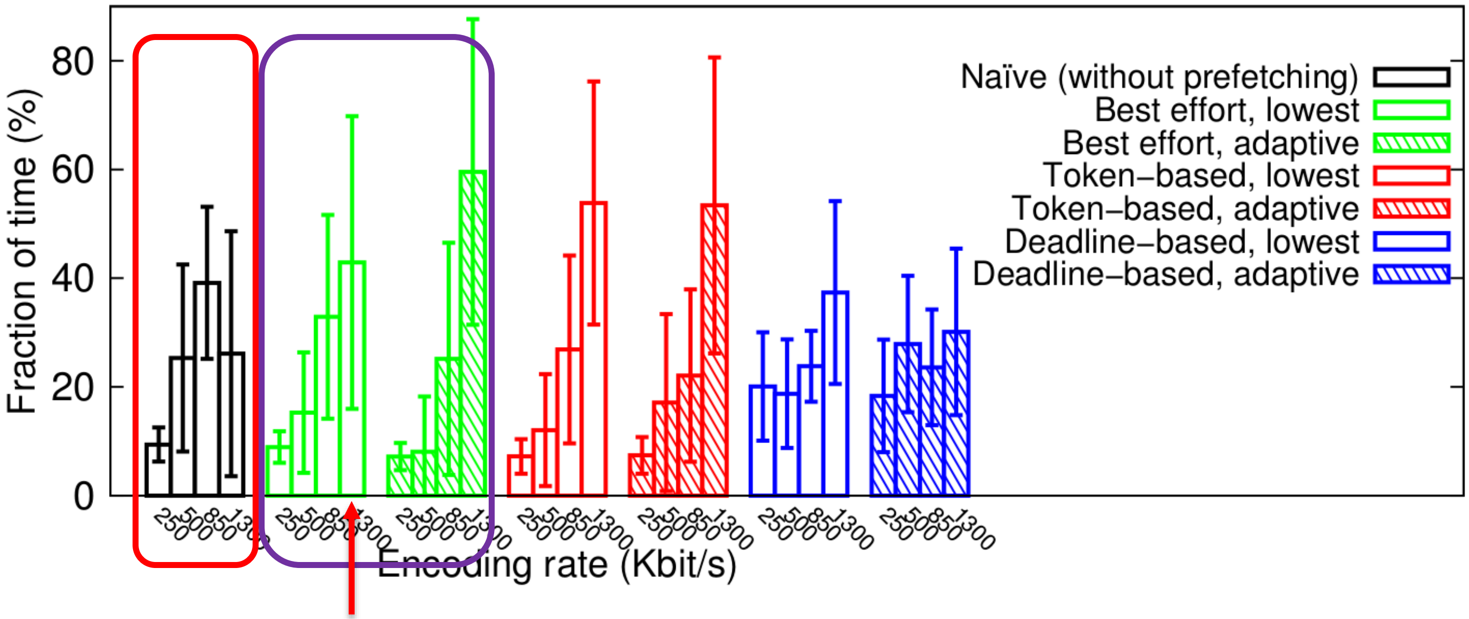


Policy characterization: Playback quality of streamed video chunks



Medium quality

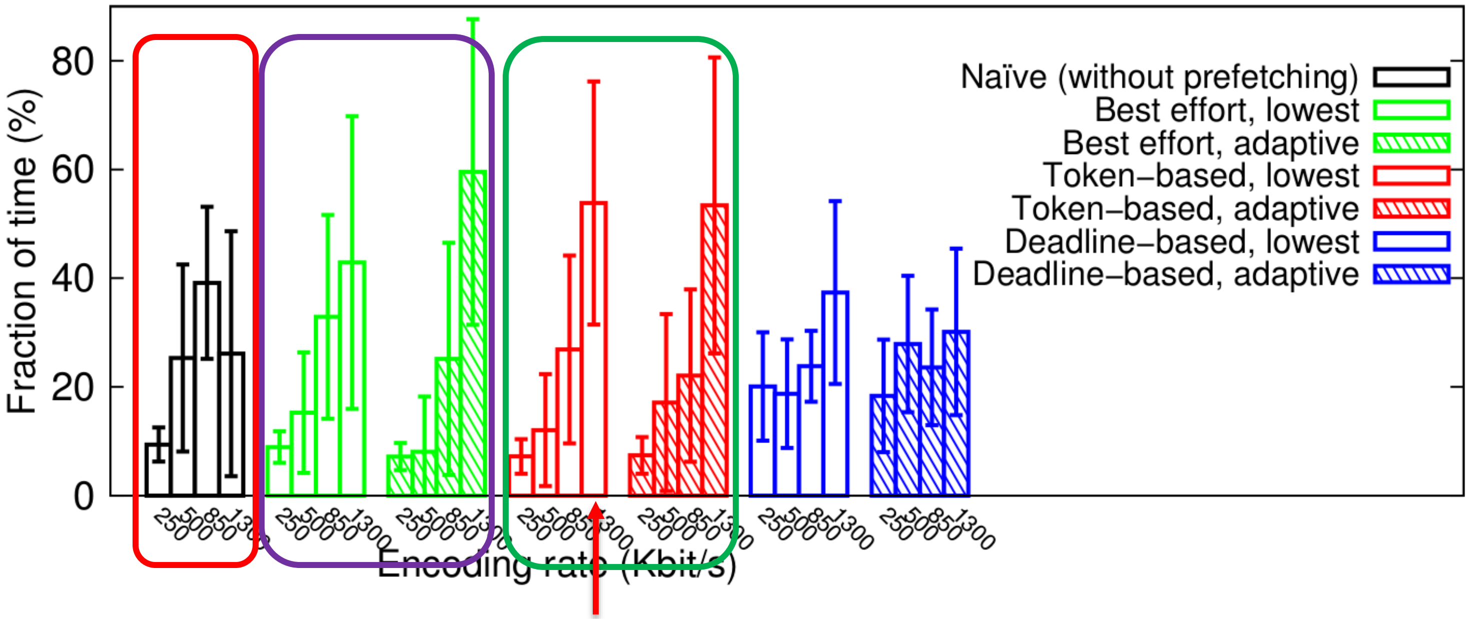
Policy characterization: Playback quality of streamed video chunks



High quality

- Best-effort policy achieves better playback rates than the naïve player

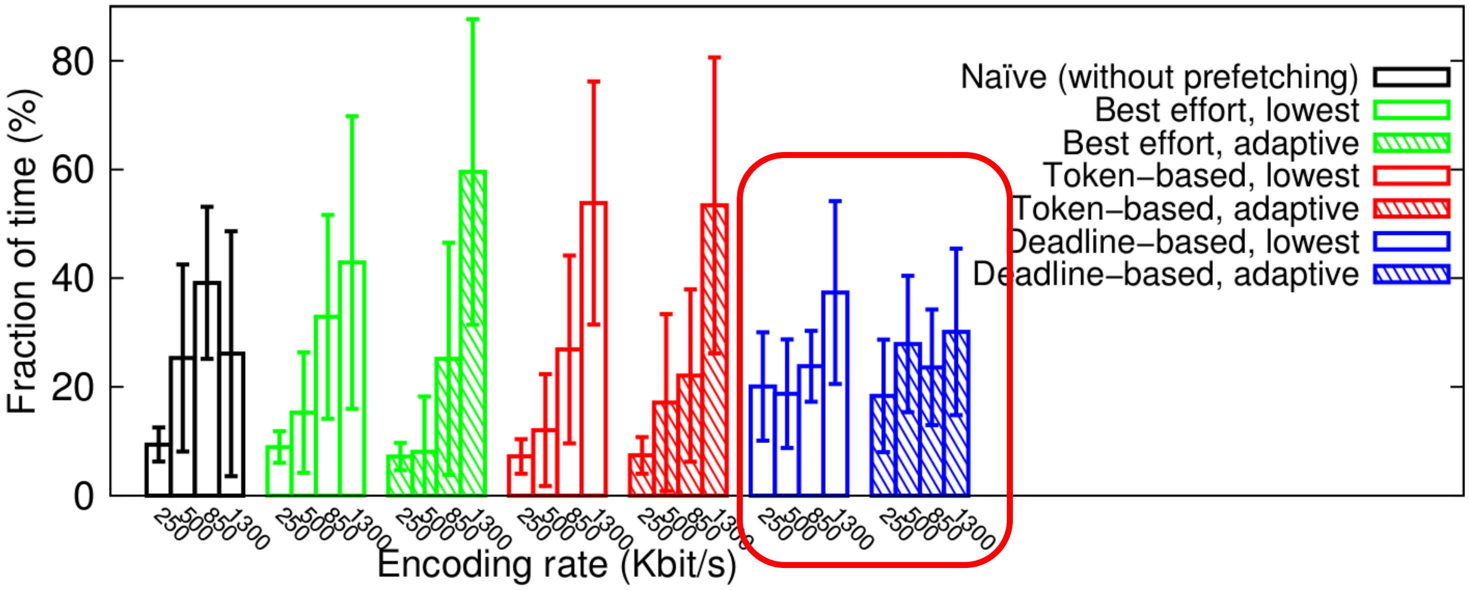
Policy characterization: Playback quality of streamed video chunks



High quality

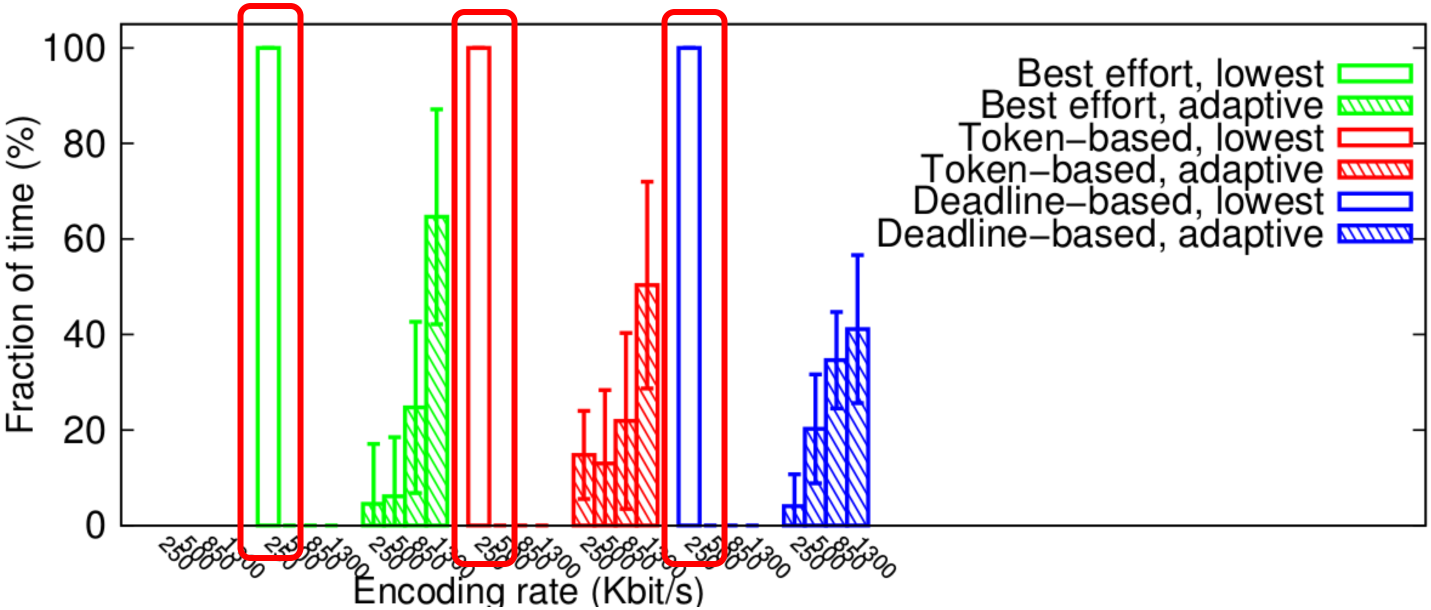
- Best-effort policy achieves better playback rates than the naïve player
- Token-based policy also achieves better playback rates than the naïve player

Policy characterization: Playback quality of streamed video chunks



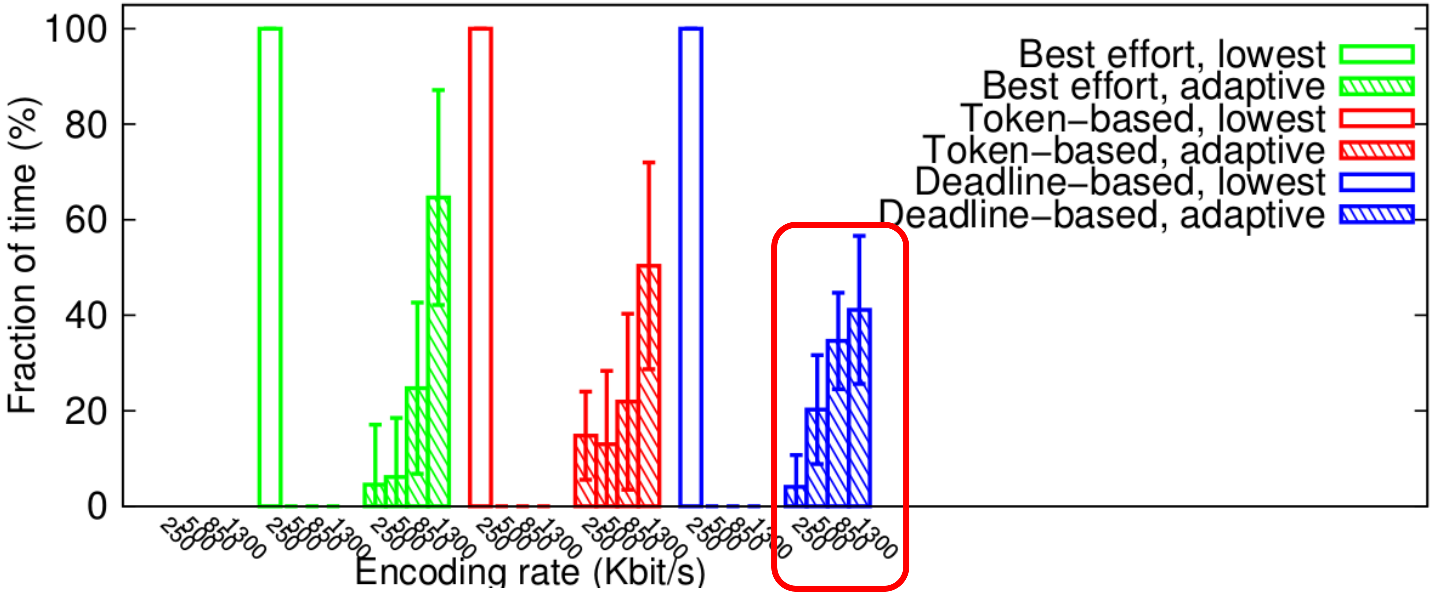
- Best-effort policy achieves better playback rates than the naïve player
- Token-based policy also achieves better playback rates than the naïve player
- Comparatively, deadline-based policies achieve lower playback qualities due to deadline constraints

Policy characterization: Playback quality of alternate video chunks



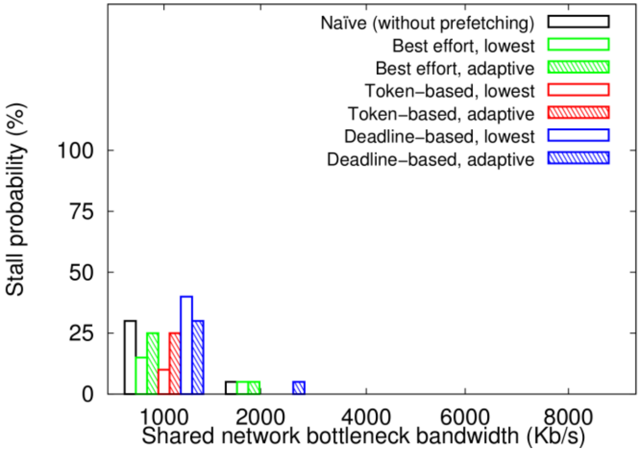
- Lowest-quality prefetching always chooses the lowest encoding available

Policy characterization: Playback quality of alternate video chunks

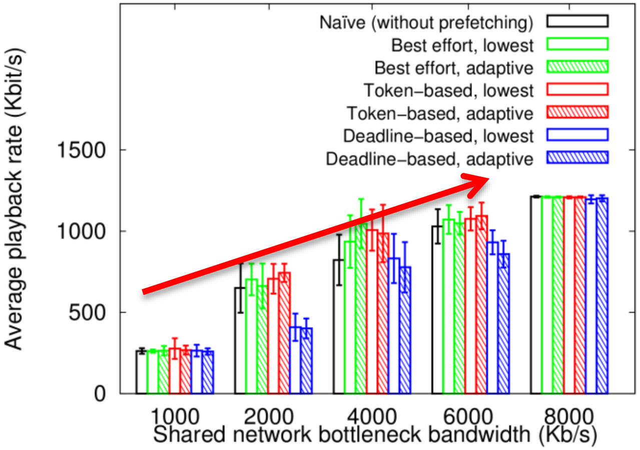


- Lowest-quality prefetching always chooses the lowest encoding available
- Deadline-based policy trades-off quality of both streamed and alternate videos to achieve the deadlines

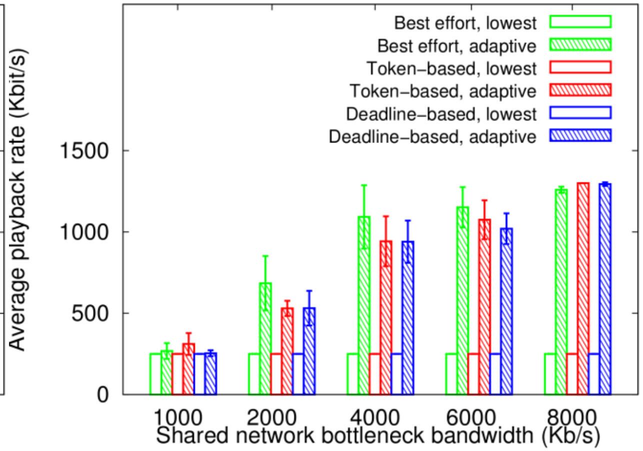
Impact of network conditions: Bandwidth



Stall probabilities



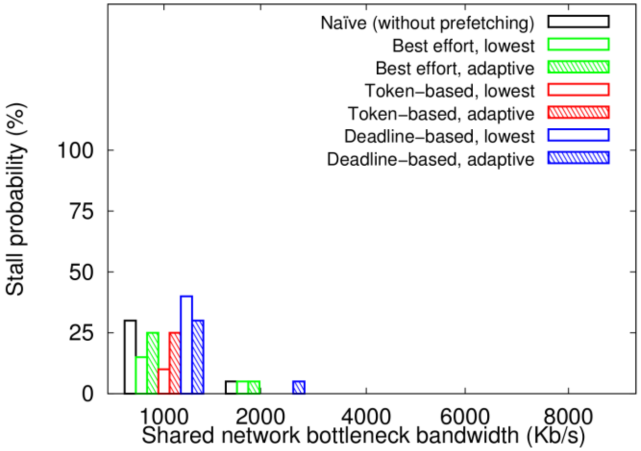
Quality of streamed video



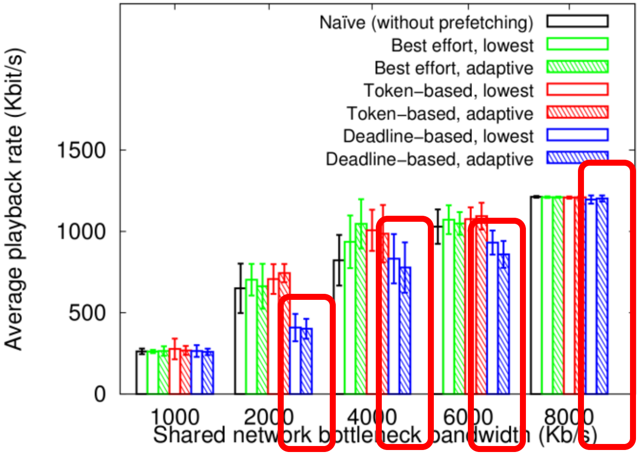
Quality of alternate video

- All policies adapt playback quality based on bandwidth

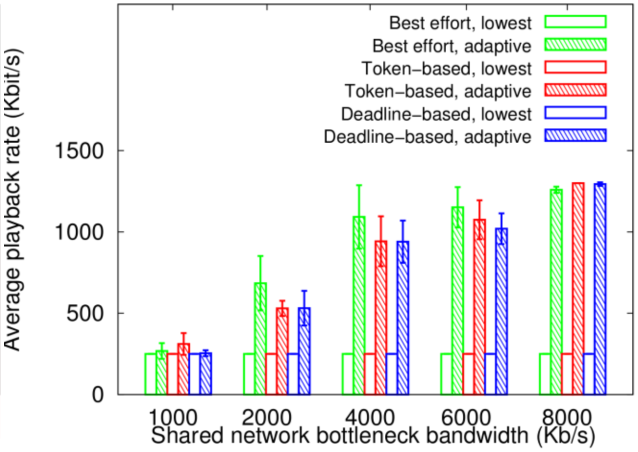
Impact of network conditions: Bandwidth



Stall probabilities



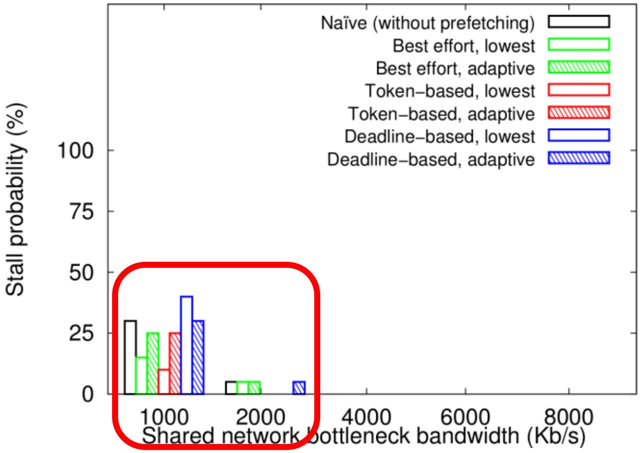
Quality of streamed video



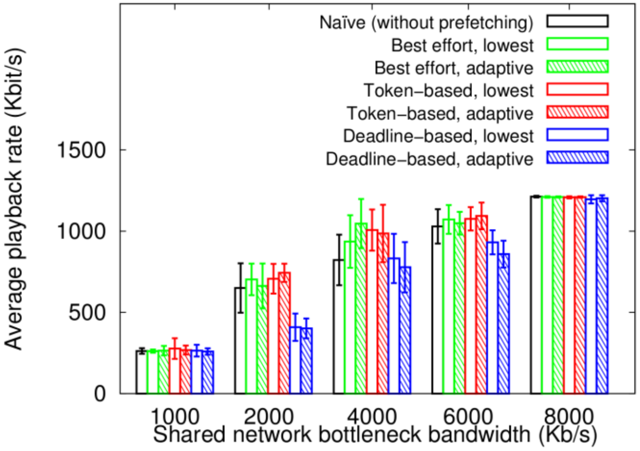
Quality of alternate video

- All policies adapt playback quality based on bandwidth
- Deadline-based policy consistently trades-off playback quality in order to meet deadlines

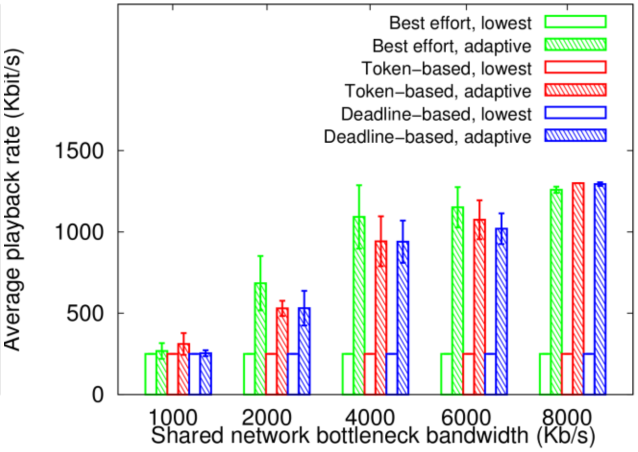
Impact of network conditions: Bandwidth



Stall probabilities



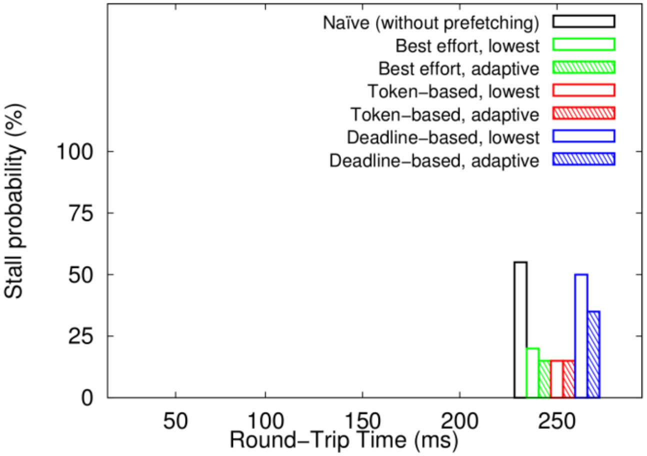
Quality of streamed video



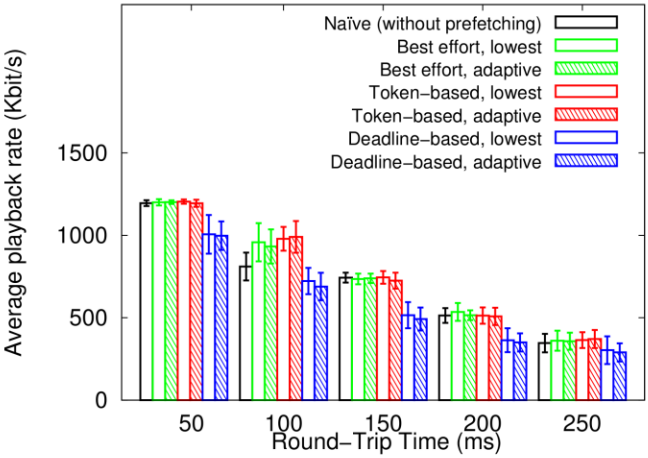
Quality of alternate video

- All policies adapt playback quality based on bandwidth
- Deadline-based policy consistently trades-off playback quality in order to meet deadlines
- Best-effort and token-based policies perform slightly better than the naïve player at low bandwidths

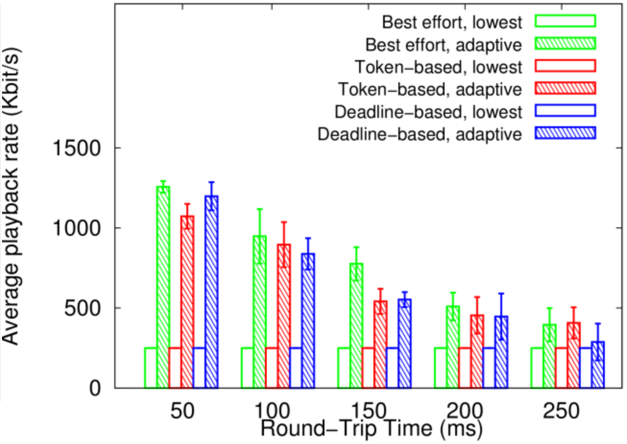
Impact of network conditions: RTT



Stall probabilities



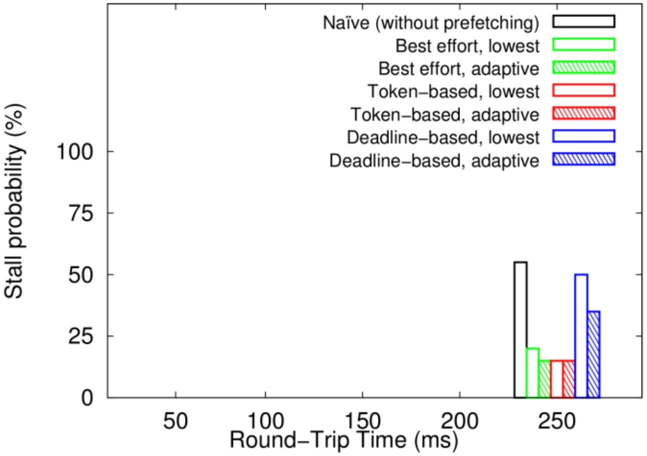
Quality of streamed video



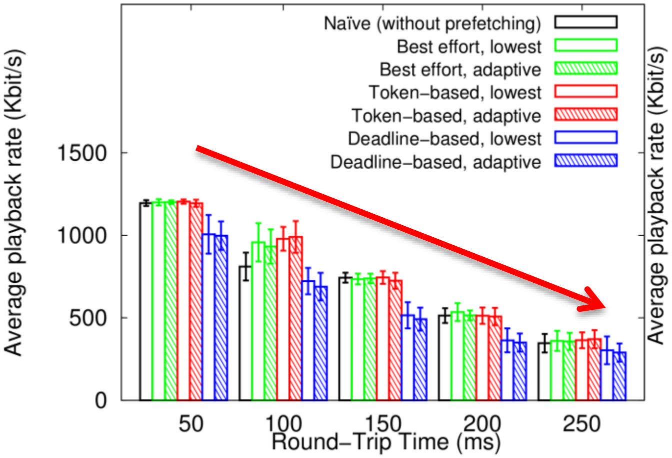
Quality of alternate video

- The video flow experiences increasing RTTs while the competing flows RTT remains constant at 50ms

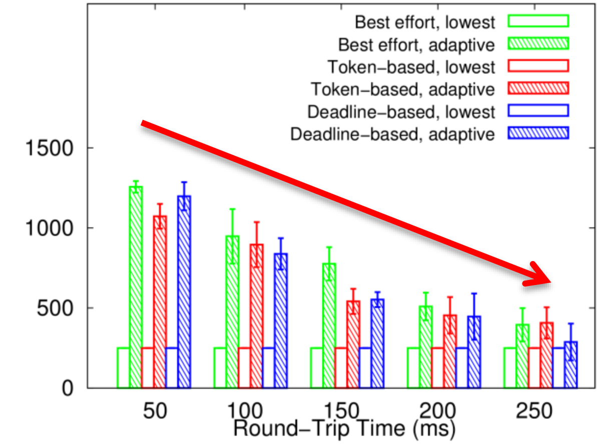
Impact of network conditions: RTT



Stall probabilities



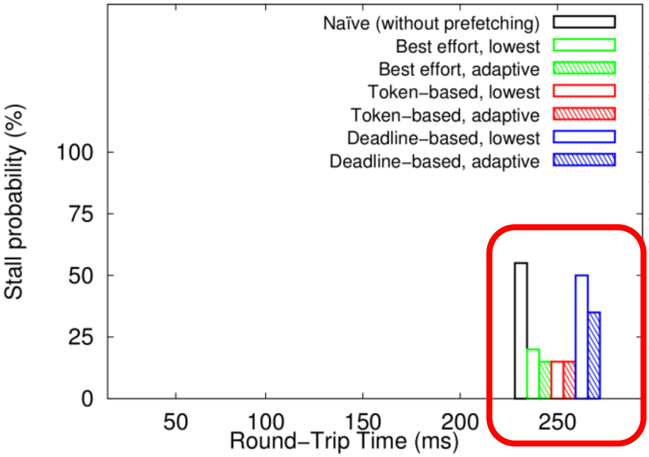
Quality of streamed video



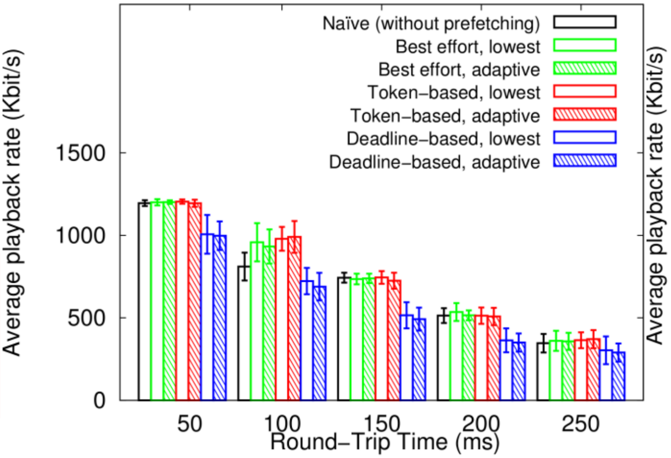
Quality of alternate video

- In general, TCP throughput decreases with increasing RTTs, as shown by playback qualities

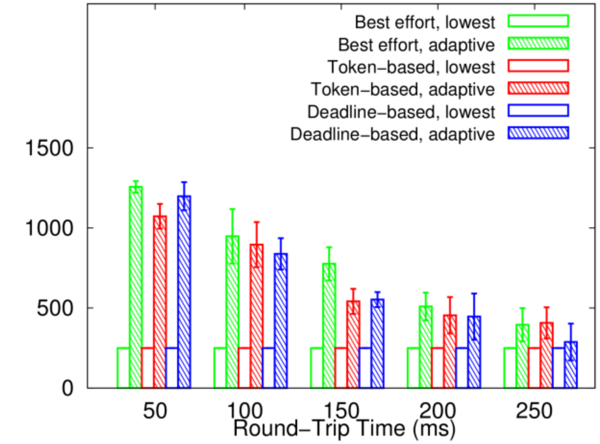
Impact of network conditions: RTT



Stall probabilities



Quality of streamed video



Quality of alternate video

- In general, TCP throughput decreases with increasing RTTs, as shown by playback qualities
- Playback stalls experienced at high RTTs, although all three policies out perform the naïve player

Startup times

	Startup time/SD (seconds)
Prefetched chunk	0.6/0.15
No prefetching, 2000 Kb/s, 150ms RTT	10/4.1
No prefetching, 4000 Kb/s, 150ms RTT	5.8/2.3
No prefetching, 6000 Kb/s, 150ms RTT	4.0/1.2
No prefetching, 8000 Kb/s, 150ms RTT	3.6/1.4

- With prefetching, startup times are low and independent of throughput or RTT
- Fetch time of the chunk from cache ~0.1 second, the additional time is required to change player states

Startup times

	Startup time/SD (seconds)
Prefetched chunk	0.6/0.15
No prefetching, 2000 Kb/s, 150ms RTT	10/4.1
No prefetching, 4000 Kb/s, 150ms RTT	5.8/2.3
No prefetching, 6000 Kb/s, 150ms RTT	4.0/1.2
No prefetching, 8000 Kb/s, 150ms RTT	3.6/1.4

- With prefetching, startup times are low and independent of throughput or RTT
- Fetch time of the chunk from cache ~0.1 second, the additional time is required to change player states
- Startup times decrease with increasing bandwidth, but are always constrained by the larger RTT and network conditions to reach the server

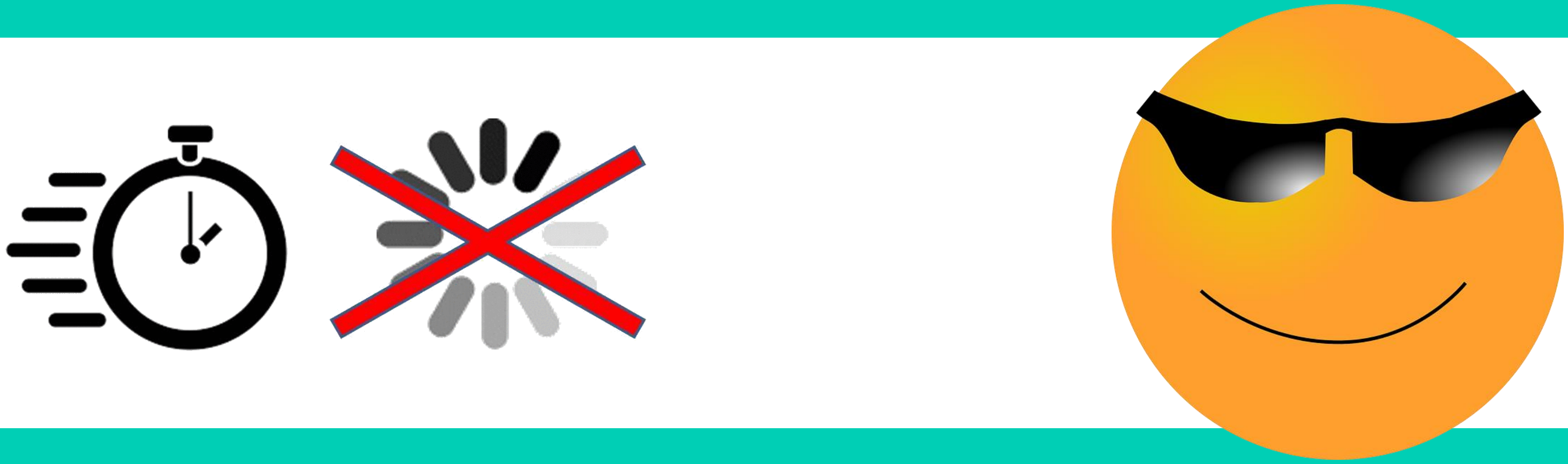
Other experimental results

- Also performed experiments under wide range of other scenarios, including
  - different buffer sizes (*4/6, 8/16, 12/24, 12/30* seconds)
  - different real-world bandwidth traces
  - different number of competing flows
  - different OSes running different TCP versions
- Our conclusions and relative performance across policies remain consistent in all scenarios

Conclusions

- We have designed, implemented and evaluated a HAS-based solution, which:
  - enables quality-adaptive prefetching and instantaneous playback of alternative videos
  - leads to perceptible gains for the streamed video in terms of stall-free playback and better playback quality
- Considered three different policy classes and two quality adaptation methods to cater for different real-world use cases
- Overall, our solutions improve the bandwidth utilization and playback experience by leveraging off periods to download alternative videos that are most likely to be watched

Bandwidth-aware Prefetching for Proactive Multi-video Preloading and Improved HAS Performance



Our source codes are available for download here:
http://www.ida.liu.se/~nikca/papers/mm15.html

University of Saskatchewan

Linköping University

NICTA