Interactive Branched Video Streaming and Cloud Assisted Content Delivery

Niklas Carlsson

Linköping University, Sweden

@ Umeå University, October 1, 2015

LINKÖPINGS UNIVERSITET

Much of the work in collaboration ...

- ***Primary research collaborators (alphabetic)****:*
  - *Martin Arlitt (HP Labs, USA)*
  - *György Dan (KTH, Sweden)*
  - *Derek Eager (University of Saskatchewan, Canada)*
  - *Phillipa Gill (Stony Brook University, USA)*
  - *Emir Halepovic (AT&T research, USA)*
  - *Anirban Mahanti (NICTA, Australia)*
  - *Aniket Mahanti (University of Auckland, New Zeeland)*
  - *Nahid Shahmehri (Linköping University, Sweden)*
  - *Carey Williamson (University of Calgary, Canada)*

… and with students

- ***PhD students at LiU***
  - *Vengatanathan Krishnamoorthi*
  - *Rahul Hiran*
  - *Anna Vapen*

- ***Researchers/postdocs at LiU***
  - *Cyriac James (now at University of Calgary)*
  - *Ajay Gopinathan (now at Google)*

- ***Also assisting PhD students elsewhere***
  - *Raoufehsadat Hashemian (U Calgary, Canada)*
  - *Benoy Varghese (NICTA, Australia)*
  - *Youmna Borghol (NICTA, Australia)* **Graduated!**
  - *Aniket Mahanti (U Calgary, Canada)* **Graduated!**

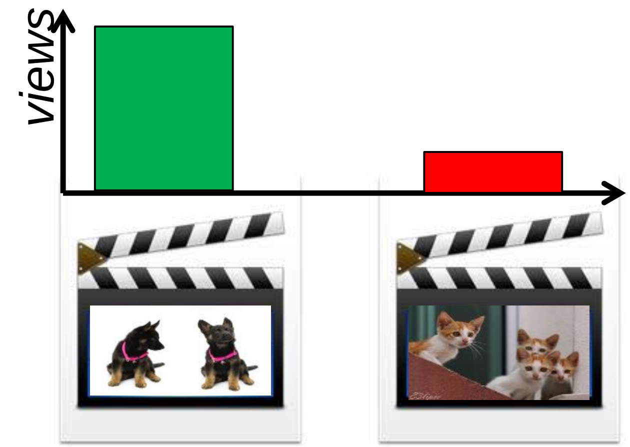
LINKÖPINGS UNIVERSITET

UNIVERSITY OF CALGARY

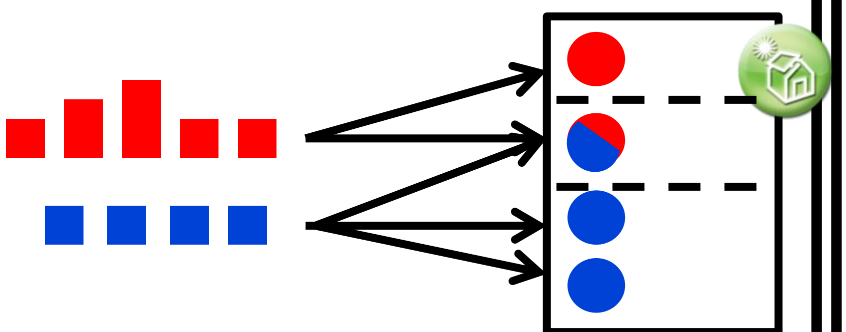
NICTA

Background: Research overview

Scalable content delivery



Characterization, analytics, modeling



Efficiency and sustainability

Network security

Design, modeling, and performance evaluation of distributed systems and networks

In this talk I will talk about …

… innovative new streaming media …

… cost-efficient delivery ...

… and determine who should serve who.

Quality-adaptive Prefetching for Interactive Branched Video using HTTP-based Adaptive Streaming
Proc. ACM Multimedia 2014.

Empowering the Creative User: Personalized HTTP-based Adaptive Streaming of Multi-path Nonlinear Video
Proc. ACM FhMN@SIGCOMM 2013. (Also in *ACM CCR*). **Best paper award**

Bandwidth-aware Prefetching for Proactive Multi-video Preloading and Improved HAS Performance
Proc. ACM Multimedia 2015.

We have all seen a movie that (in our taste) is...

We have all seen a movie that (in our taste) is...

too sad

We have all seen a movie that (in our taste) is...

too sad

too violent

We have all seen a movie that (in our taste) is...

too sad

too violent

too scary

...

We have all seen a movie that (in our taste) is...

too sad

too violent

too scary

...

… or where we may have wanted our favorite character to make a different choice...

We have all seen a movie that (in our taste) is...

too sad

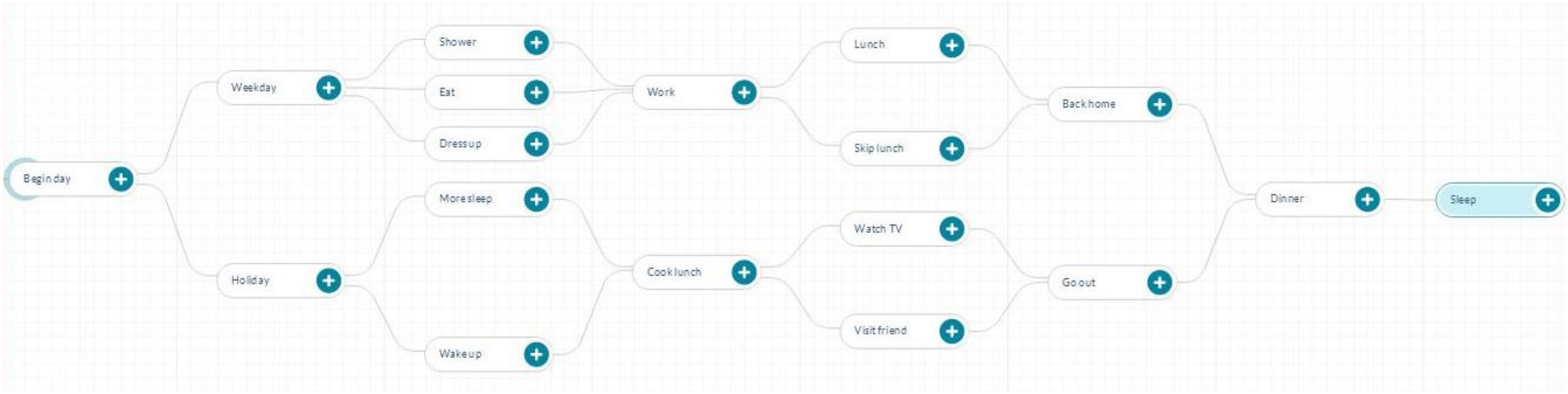
too violent

too scary

...

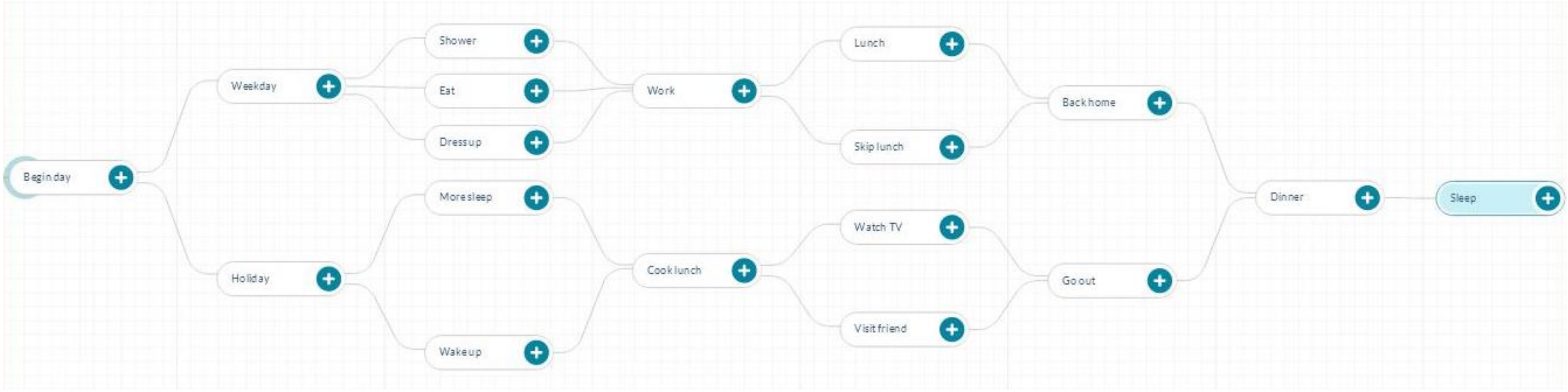
… or where we may have wanted our favorite character to make a different choice...

Interactive Branched Video



Allow user to selects between multiple storylines or alternative endings

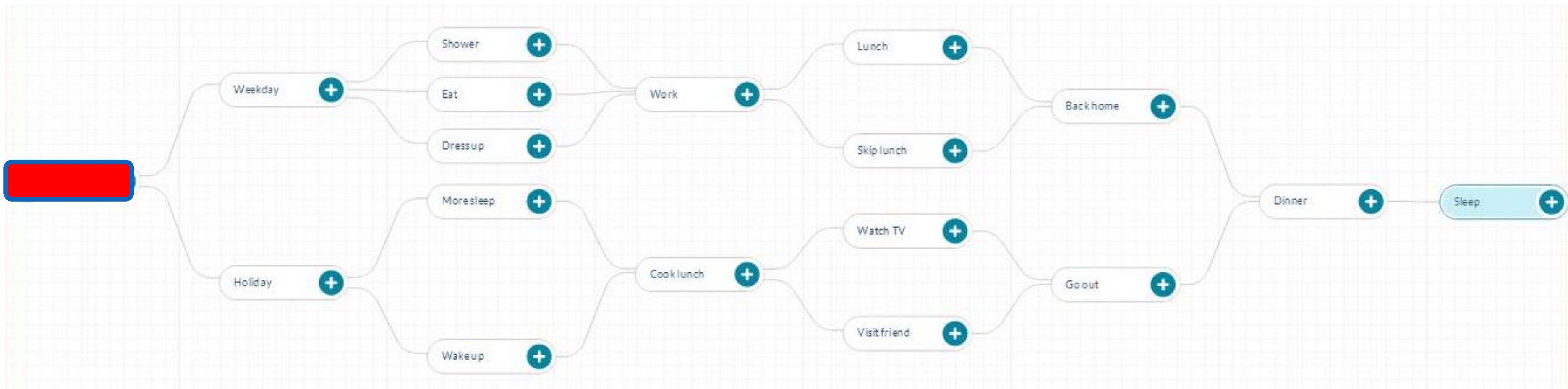
Interactive Branched Video



Allow user to selects between multiple storylines or alternative endings

Clickable objects allow the user to interact with the player and influence the storyline

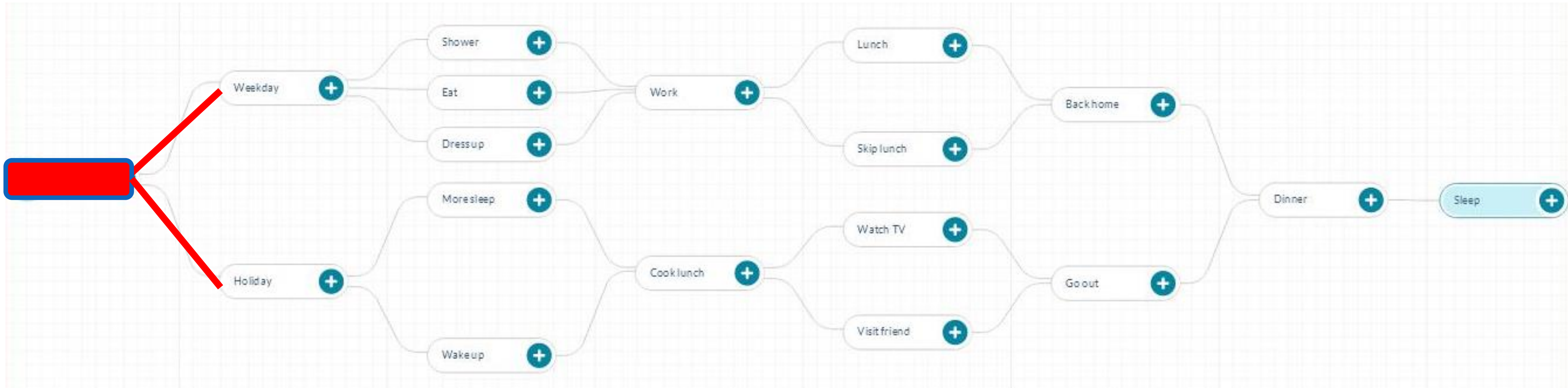
Interactive Branched Video



Allow user to selects between multiple storylines or alternative endings

Clickable objects allow the user to interact with the player and influence the storyline

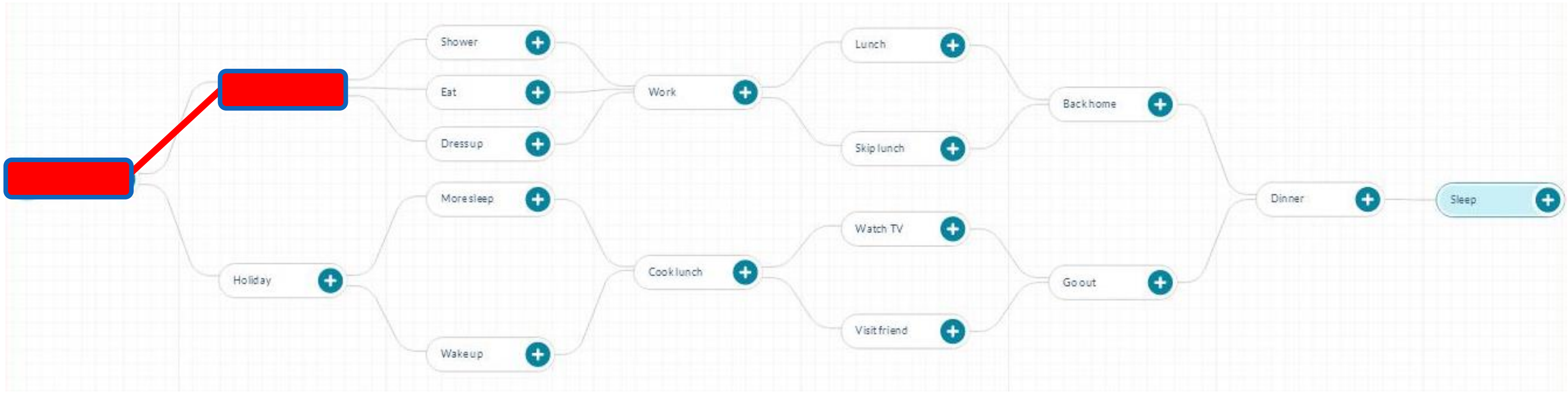
Interactive Branched Video



Allow user to selects between multiple storylines or alternative endings

Clickable objects allow the user to interact with the player and influence the storyline

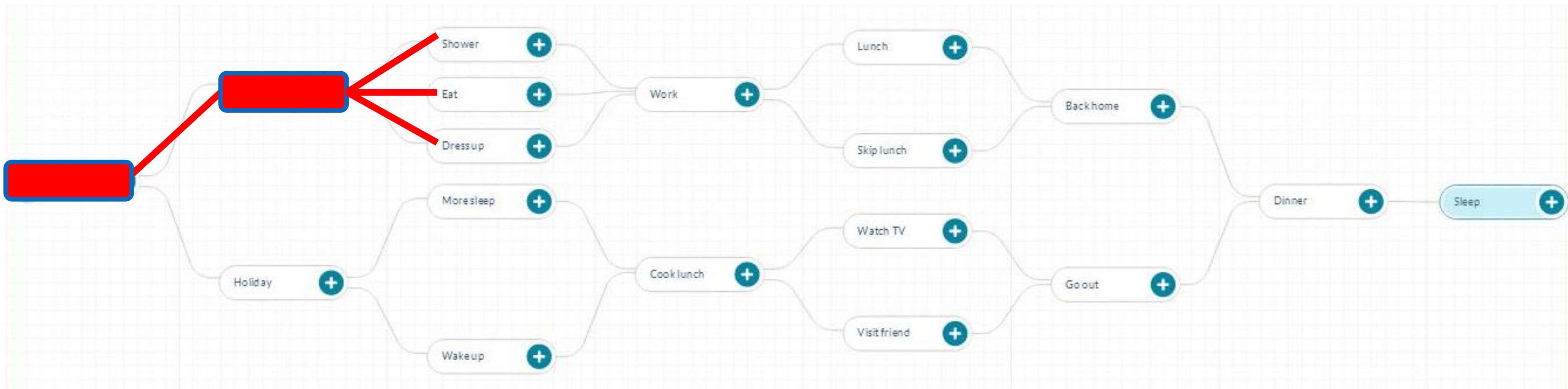
Interactive Branched Video



Allow user to selects between multiple storylines or alternative endings

Clickable objects allow the user to interact with the player and influence the storyline

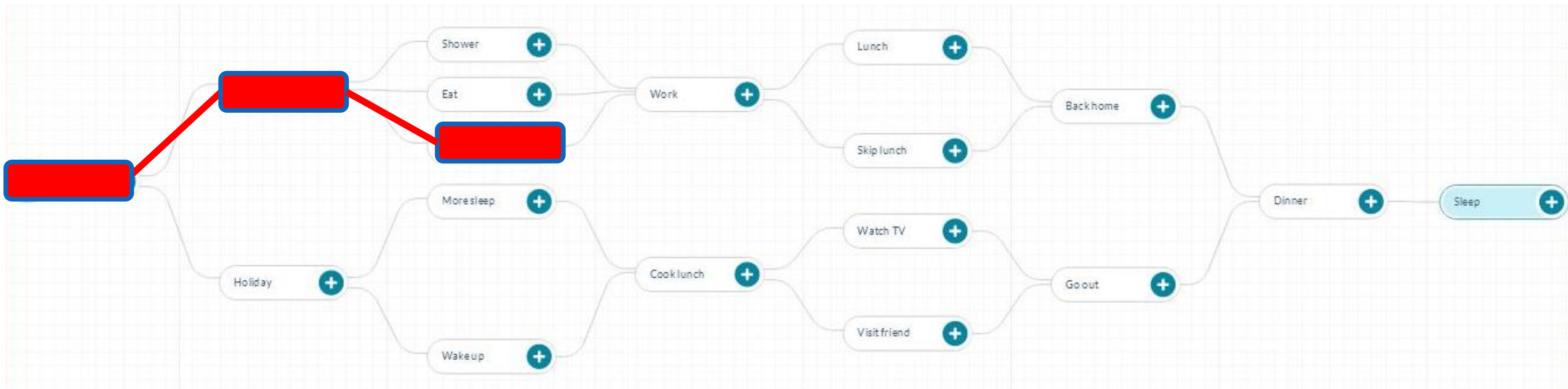
Interactive Branched Video



Allow user to selects between multiple storylines or alternative endings

Clickable objects allow the user to interact with the player and influence the storyline

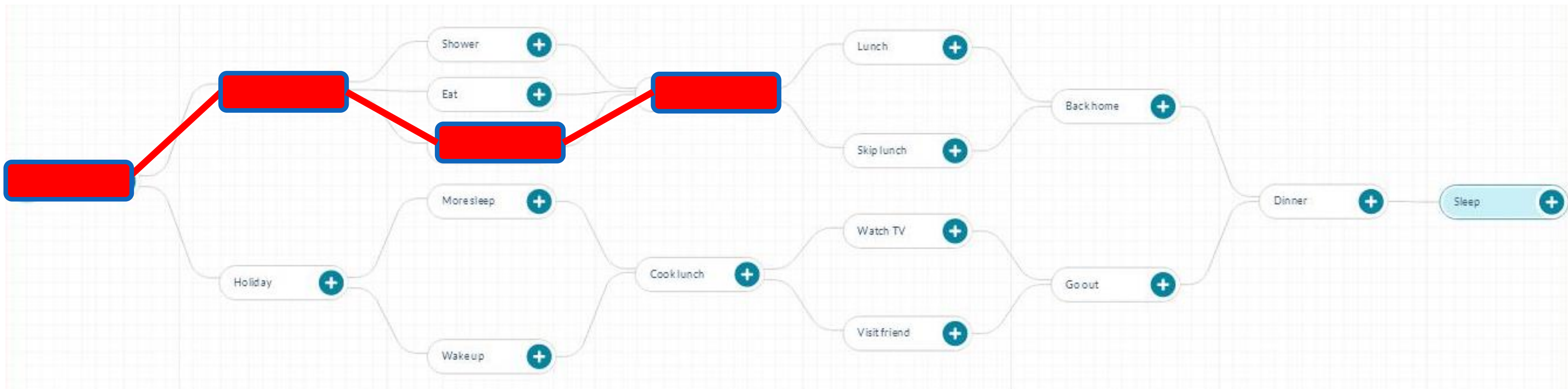
Interactive Branched Video



Allow user to selects between multiple storylines or alternative endings

Clickable objects allow the user to interact with the player and influence the storyline

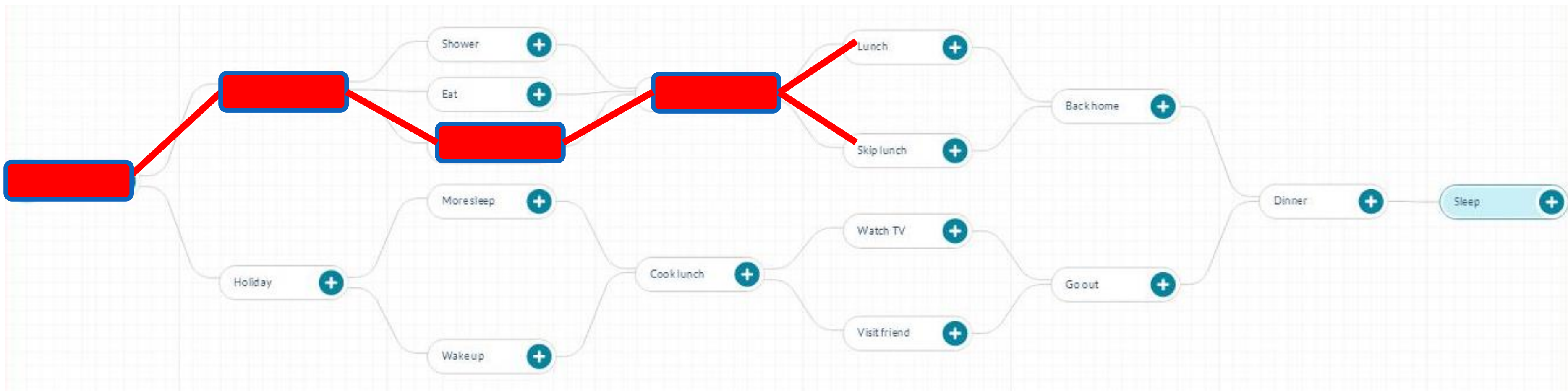
Interactive Branched Video



Allow user to selects between multiple storylines or alternative endings

Clickable objects allow the user to interact with the player and influence the storyline

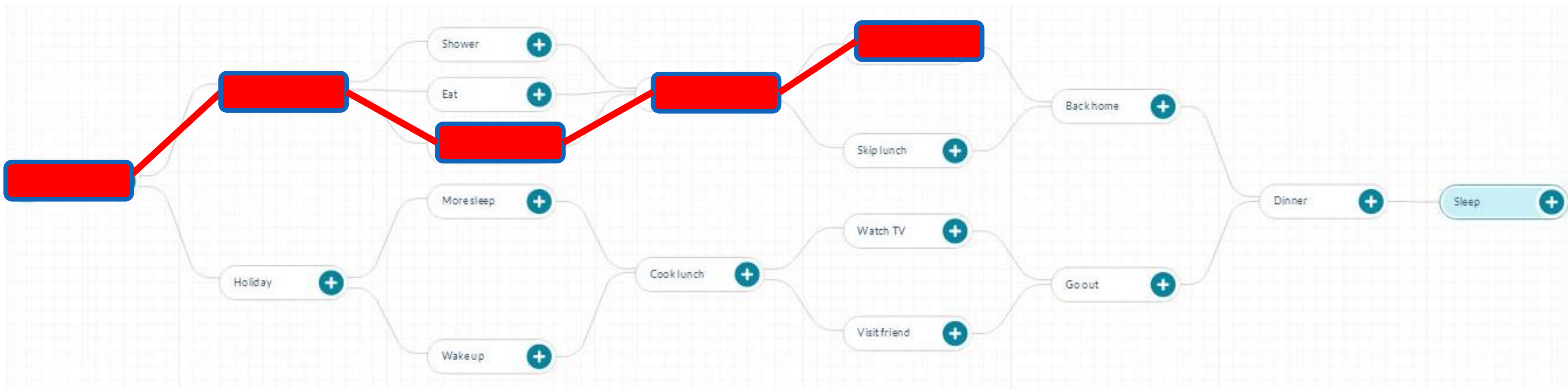
Interactive Branched Video



Allow user to selects between multiple storylines or alternative endings

Clickable objects allow the user to interact with the player and influence the storyline

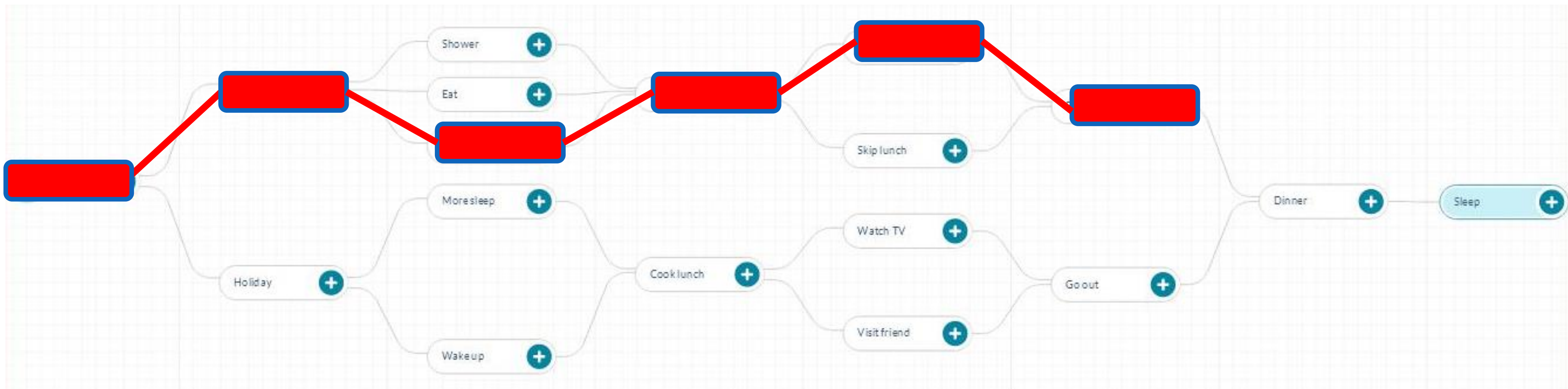
Interactive Branched Video



Allow user to selects between multiple storylines or alternative endings

Clickable objects allow the user to interact with the player and influence the storyline

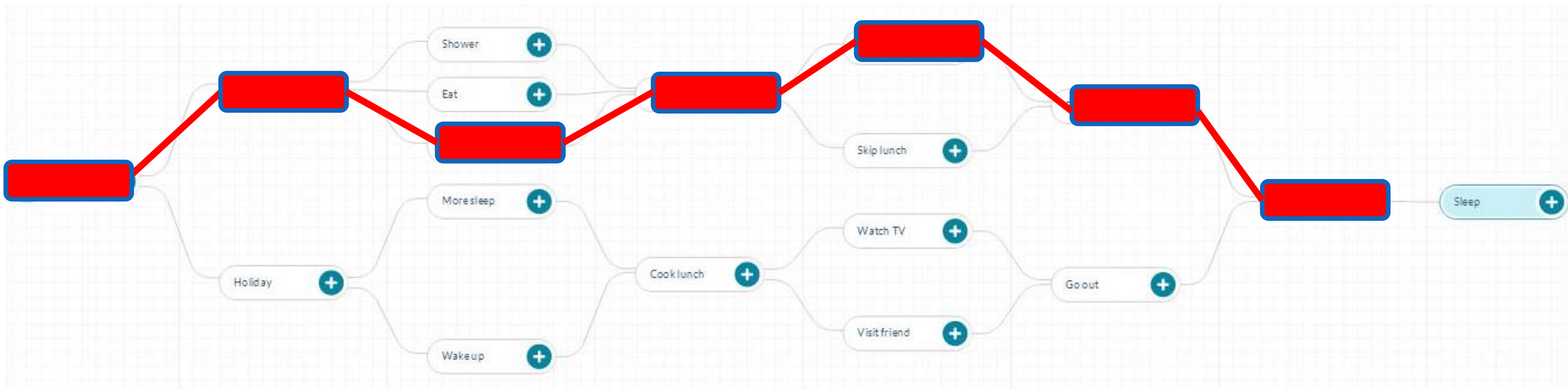
Interactive Branched Video



Allow user to selects between multiple storylines or alternative endings

Clickable objects allow the user to interact with the player and influence the storyline

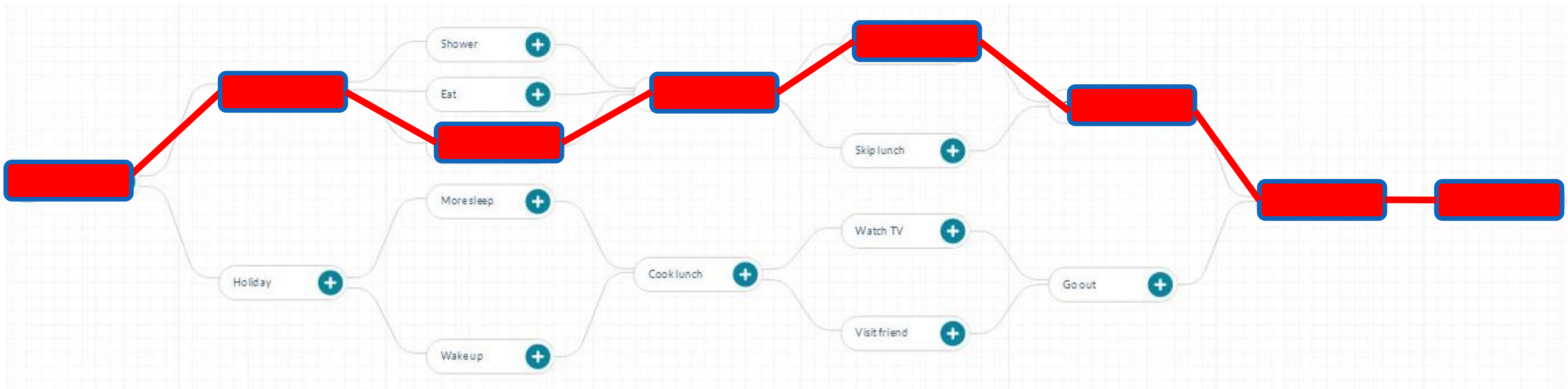
Interactive Branched Video



Allow user to selects between multiple storylines or alternative endings

Clickable objects allow the user to interact with the player and influence the storyline

Interactive Branched Video



Allow user to selects between multiple storylines or alternative endings

Clickable objects allow the user to interact with the player and influence the storyline

We have solved …

The problem of providing seamless playback in the presence of multiple branch options

We have solved …

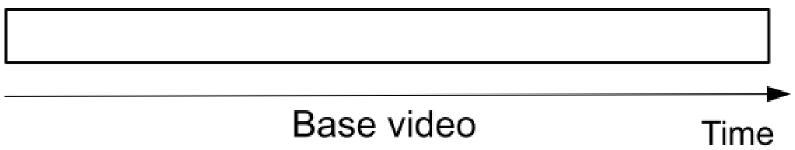
The problem of providing seamless playback in the presence of multiple branch options

We have solved …

The problem of providing seamless playback in the presence of multiple branch options

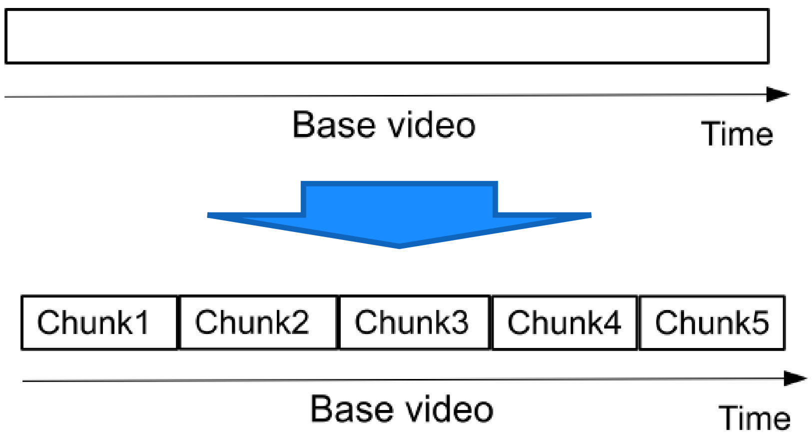
- *HTTP-based Adaptive Streaming*
- *Path and quality-aware prefetching*

HTTP-based Adaptive Streaming (HAS)



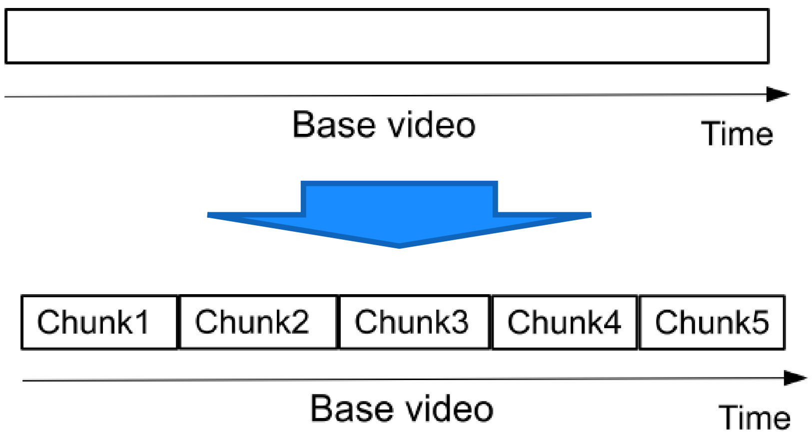
- HTTP-based streaming
  - Video is split into chunks

HTTP-based Adaptive Streaming (HAS)



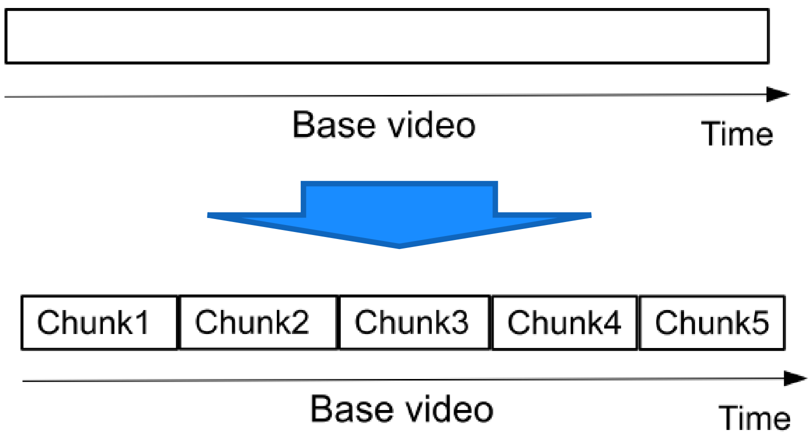
- HTTP-based streaming
  - Video is split into chunks
  -
  -
-
  -
  -

HTTP-based Adaptive Streaming (HAS)



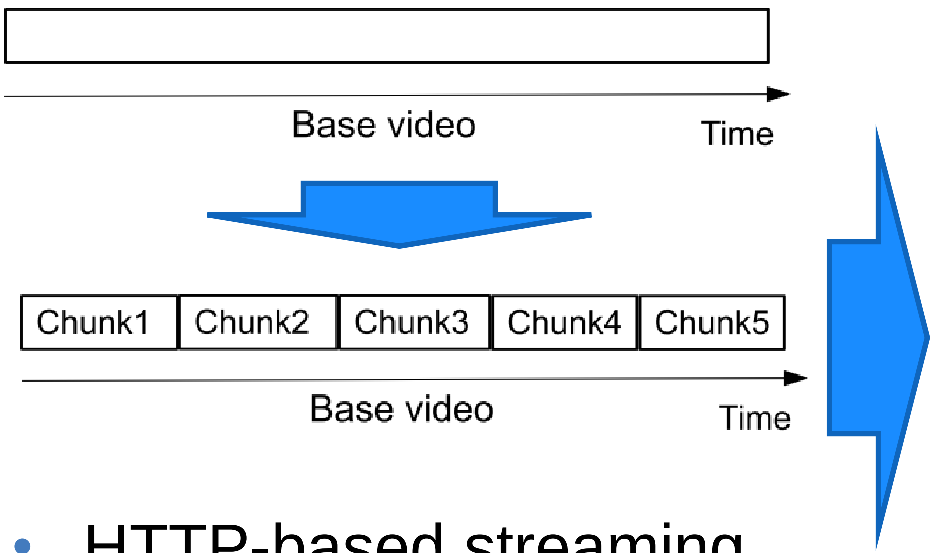
- HTTP-based streaming
  - Video is split into chunks
  - Easy firewall traversal and caching
  -
-
  -
  -

HTTP-based Adaptive Streaming (HAS)



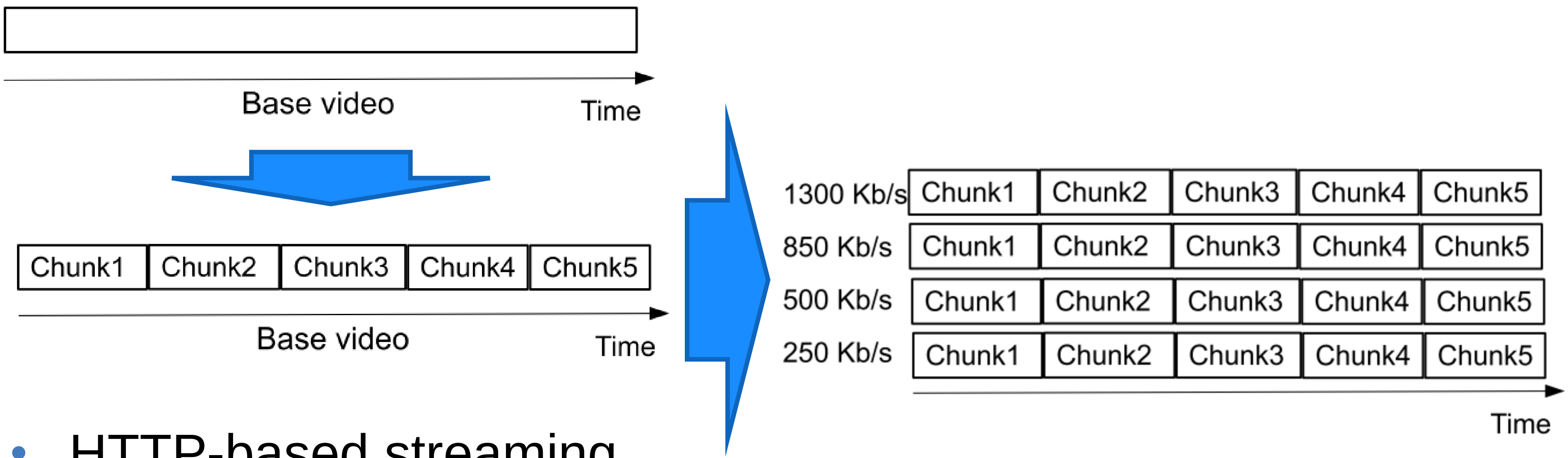
- HTTP-based streaming
  - Video is split into chunks
  - Easy firewall traversal and caching
  - Easy support for interactive VoD
-
  -
  -

HTTP-based Adaptive Streaming (HAS)



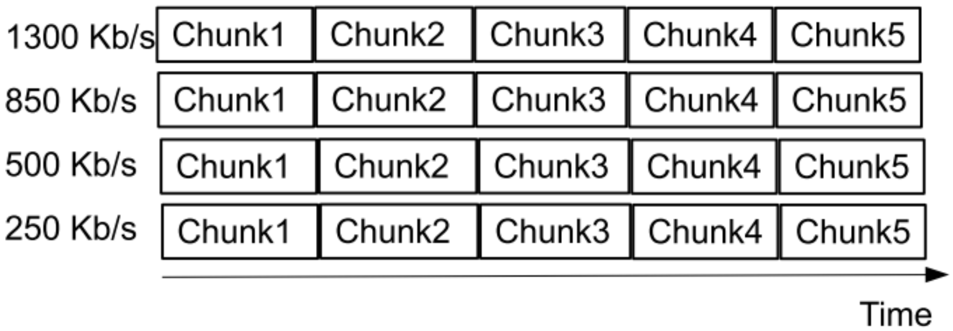
- HTTP-based streaming
  - Video is split into chunks
  - Easy firewall traversal and caching
  - Easy support for interactive VoD
- HTTP-based adaptive streaming
  -
  -

HTTP-based Adaptive Streaming (HAS)



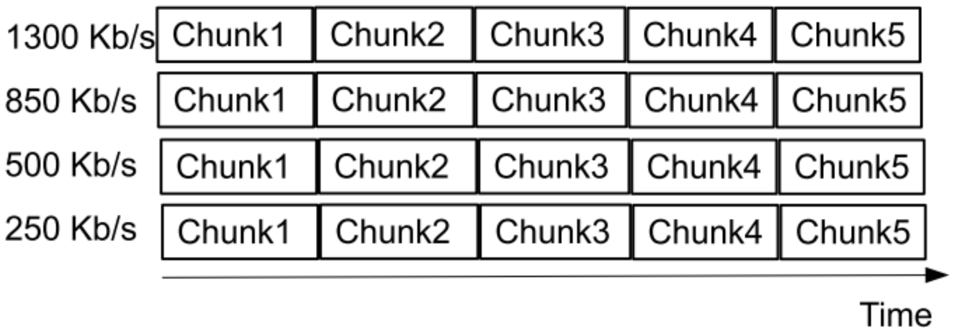
- HTTP-based streaming
  - Video is split into chunks
  - Easy firewall traversal and caching
  - Easy support for interactive VoD
- HTTP-based adaptive streaming
  - Multiple encodings of each chunk (defined in manifest file)
  -

HTTP-based Adaptive Streaming (HAS)



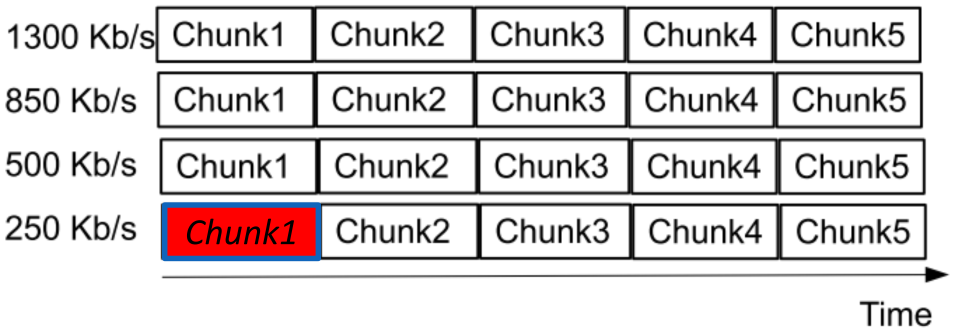
- HTTP-based streaming
  - Video is split into chunks
  - Easy firewall traversal and caching
  - Easy support for interactive VoD
- HTTP-based adaptive streaming
  - Multiple encodings of each chunk (defined in manifest file)
  -

HTTP-based Adaptive Streaming (HAS)



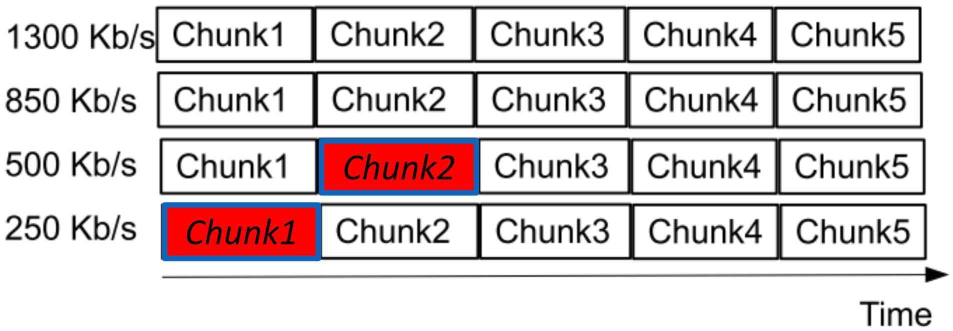
- HTTP-based streaming
  - Video is split into chunks
  - Easy firewall traversal and caching
  - Easy support for interactive VoD
- HTTP-based adaptive streaming
  - Multiple encodings of each chunk (defined in manifest file)
  - Clients adapt quality encoding based on buffer/network conditions

HTTP-based Adaptive Streaming (HAS)



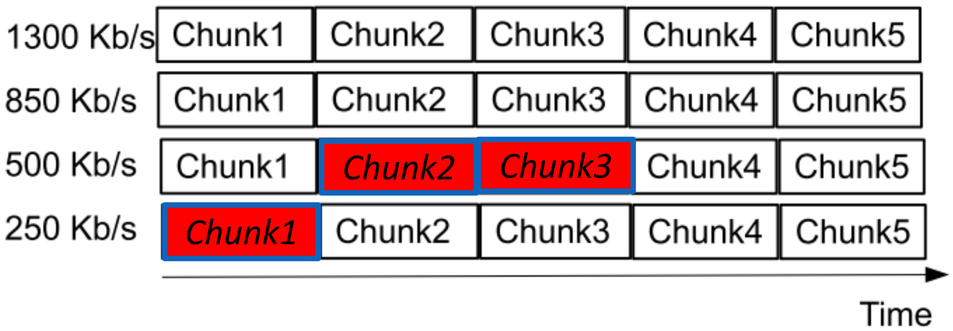
- HTTP-based streaming
  - Video is split into chunks
  - Easy firewall traversal and caching
  - Easy support for interactive VoD
- HTTP-based adaptive streaming
  - Multiple encodings of each chunk (defined in manifest file)
  - Clients adapt quality encoding based on buffer/network conditions

HTTP-based Adaptive Streaming (HAS)



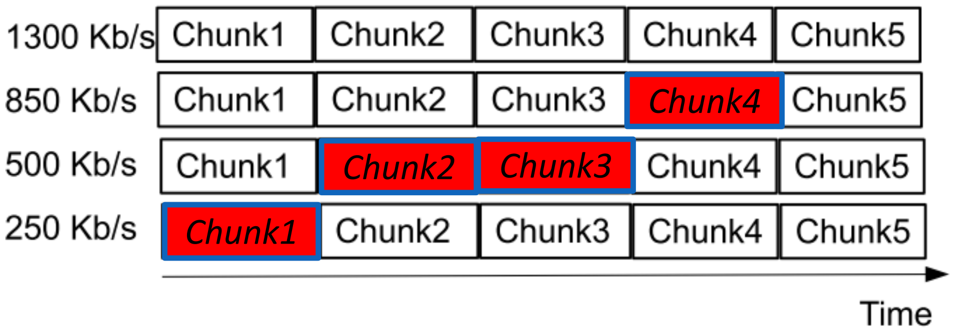
- HTTP-based streaming
  - Video is split into chunks
  - Easy firewall traversal and caching
  - Easy support for interactive VoD
- HTTP-based adaptive streaming
  - Multiple encodings of each chunk (defined in manifest file)
  - Clients adapt quality encoding based on buffer/network conditions

HTTP-based Adaptive Streaming (HAS)



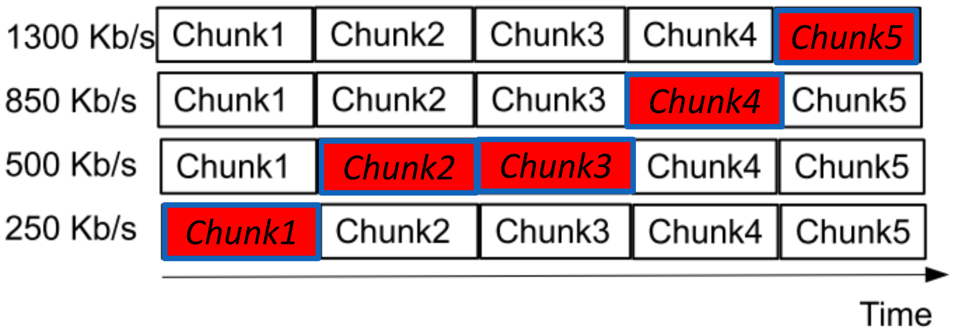
- HTTP-based streaming
  - Video is split into chunks
  - Easy firewall traversal and caching
  - Easy support for interactive VoD
- HTTP-based adaptive streaming
  - Multiple encodings of each chunk (defined in manifest file)
  - Clients adapt quality encoding based on buffer/network conditions

HTTP-based Adaptive Streaming (HAS)



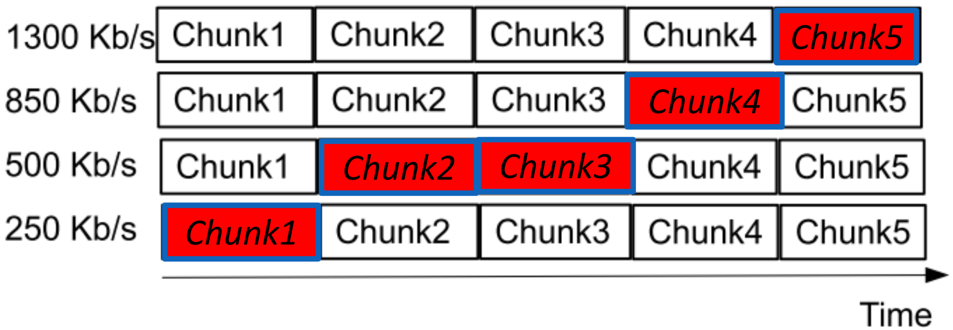
- HTTP-based streaming
  - Video is split into chunks
  - Easy firewall traversal and caching
  - Easy support for interactive VoD
- HTTP-based adaptive streaming
  - Multiple encodings of each chunk (defined in manifest file)
  - Clients adapt quality encoding based on buffer/network conditions

HTTP-based Adaptive Streaming (HAS)



- HTTP-based streaming
  - Video is split into chunks
  - Easy firewall traversal and caching
  - Easy support for interactive VoD
- HTTP-based adaptive streaming
  - Multiple encodings of each chunk (defined in manifest file)
  - 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
  - Easy support for interactive VoD
- HTTP-based adaptive streaming
  - Multiple encodings of each chunk (defined in manifest file)
  - Clients adapt quality encoding based on buffer/network conditions

Contributions

- We develop a simple analytic model which allows us to define the prefetching problem as an optimization problem
  - Maximizes expected playback quality while avoiding stalls
- Based on our findings, we design optimized policies that determine:
  1. When different chunks should be downloaded
  2. What quality level should be selected for each of these chunks
  3. How to manage playback buffers and (multiple) TCP connections

  such as to ensure smooth playback experience without excessive workahead (buffering)
- The design and implementation of the framework
- Experimental evaluation of our policies, which provide insights into the importance of careful adaptive policies

Contributions

- We develop a simple analytic model which allows us to define the prefetching problem as an optimization problem
  - Maximizes expected playback quality while avoiding stalls
- Based on our findings, we design optimized policies that determine:
  1. When different chunks should be downloaded
  2. What quality level should be selected for each of these chunks
  3. How to manage playback buffers and (multiple) TCP connections

  such as to ensure smooth playback experience without excessive workahead (buffering)
- The design and implementation of the framework
- Experimental evaluation of our policies, which provide insights into the importance of careful adaptive policies

Contributions

- We develop a simple analytic model which allows us to define the prefetching problem as an optimization problem
  - Maximizes expected playback quality while avoiding stalls
- Based on our findings, we design optimized policies that determine:
  1. When different chunks should be downloaded
  2. What quality level should be selected for each of these chunks
  3. How to manage playback buffers and (multiple) TCP connections

  such as to ensure smooth playback experience without excessive workahead (buffering)
- The design and implementation of the framework
- Experimental evaluation of our policies, which provide insights into the importance of careful adaptive policies

Contributions

- We develop a simple analytic model which allows us to define the prefetching problem as an optimization problem
  - Maximizes expected playback quality while avoiding stalls
- Based on our findings, we design optimized policies that determine:
  1. When different chunks should be downloaded
  2. What quality level should be selected for each of these chunks
  3. How to manage playback buffers and (multiple) TCP connections

  such as to ensure smooth playback experience without excessive workahead (buffering)
- The design and implementation* of the framework
- Experimental evaluation of our policies, which provide insights into the importance of careful adaptive policies

**Software: http://www.ida.liu.se/~nikca/mm14.html*

Contributions

- We develop a simple analytic model which allows us to define the prefetching problem as an optimization problem
  - Maximizes expected playback quality while avoiding stalls
- Based on our findings, we design optimized policies that determine:
  1. When different chunks should be downloaded
  2. What quality level should be selected for each of these chunks
  3. How to manage playback buffers and (multiple) TCP connections

  such as to ensure smooth playback experience without excessive workahead (buffering)
- The design and implementation* of the framework
- Experimental evaluation of our policies, which provide insights into the importance of careful adaptive policies

**Software: http://www.ida.liu.se/~nikca/mm14.html*

Contributions

- We develop a simple analytic model which allows us to define the prefetching problem as an optimization problem
  - Maximizes expected playback quality while avoiding stalls
- Based on our findings, we design optimized policies that determine:
  1. When different chunks should be downloaded
  2. What quality level should be selected for each of these chunks
  3. How to manage playback buffers and (multiple) TCP connections

  such as to ensure smooth playback experience without excessive workahead (buffering)
- The design and implementation* of the framework
- Experimental evaluation of our policies, which provide insights into the importance of careful adaptive policies

**Software: http://www.ida.liu.se/~nikca/mm14.html*

Contributions

- We develop a simple analytic model which allows us to define the prefetching problem as an optimization problem
  - Maximizes expected playback quality while avoiding stalls
- Based on our findings, we design optimized policies that determine:
  1. When different chunks should be downloaded
  2. What quality level should be selected for each of these chunks
  3. How to manage playback buffers and (multiple) TCP connections

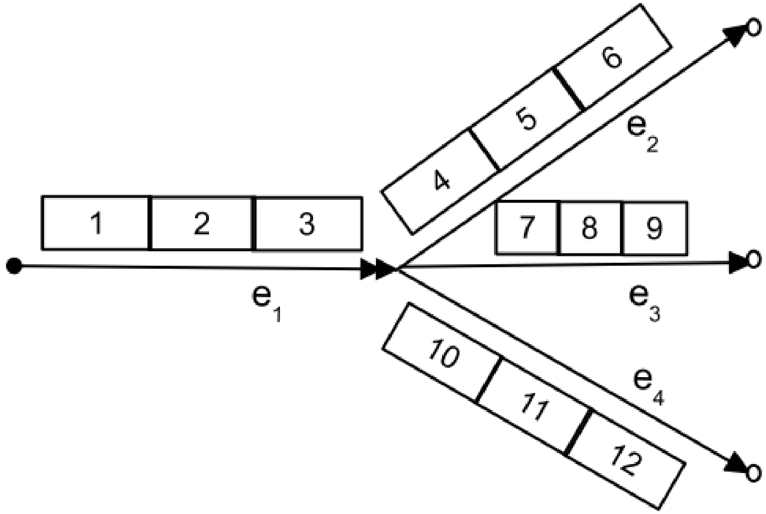
  such as to ensure smooth playback experience without excessive workahead (buffering)
- The design and implementation* of the framework
- Experimental evaluation of our policies, which provide insights into the importance of careful adaptive policies

**Software: http://www.ida.liu.se/~nikca/mm14.html*

Problem Description and Constraints

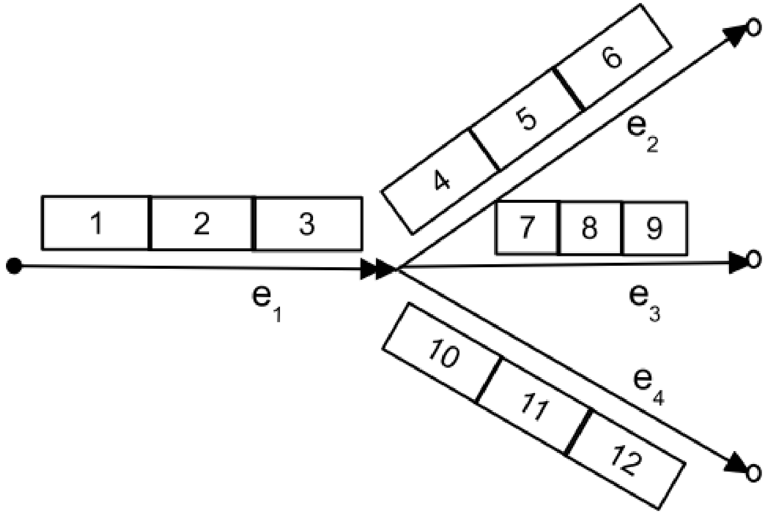
- Problem: Maximize quality, given playback deadlines and bandwidth conditions

Problem Description and Constraints



- Problem: Maximize quality, given playback deadlines and bandwidth conditions

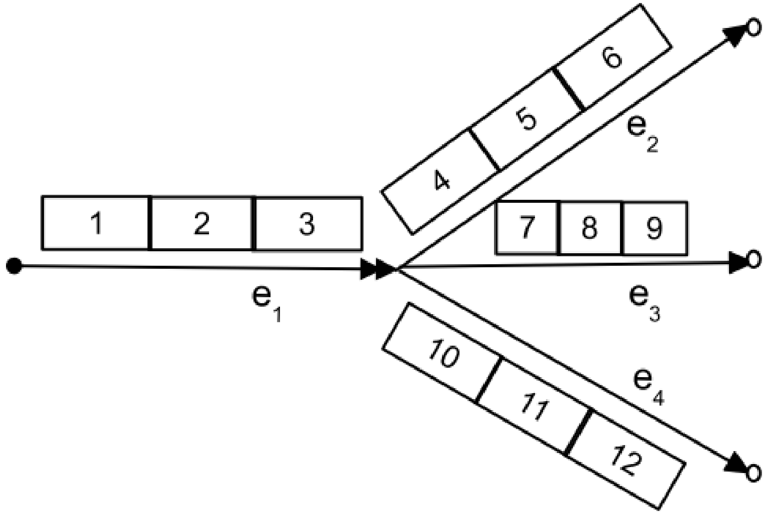
Problem Description and Constraints



- Objective function

maximize *playback quality*

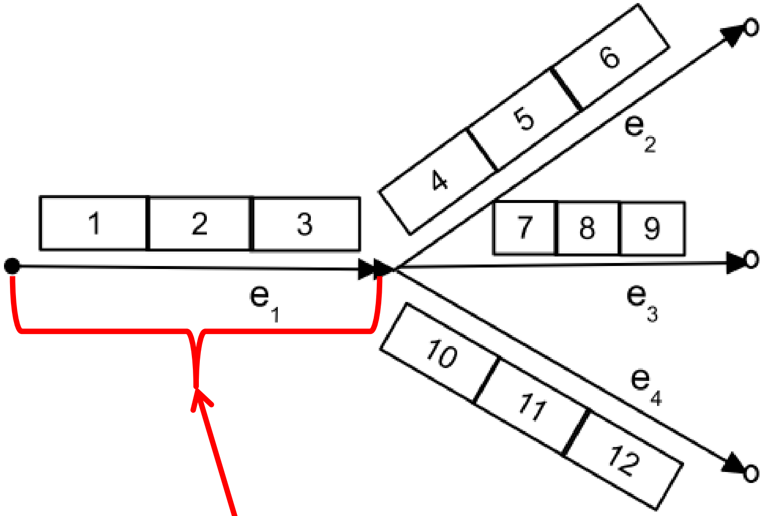
Problem Description and Constraints



- Objective function

$$\text{maximize} \sum_{i=1}^{n_e} q_i l_i + \sum_{i=n_e+1}^{n_e+|\mathcal{E}^b|} q_i l_i$$

Problem Description and Constraints

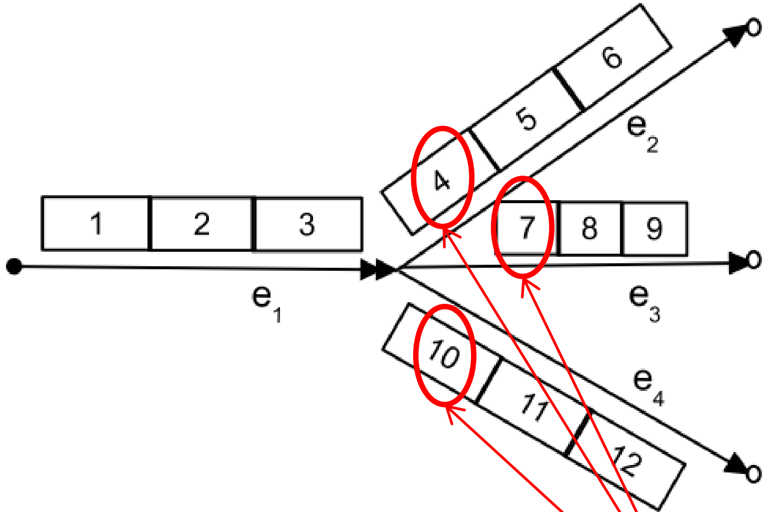


- Objective function

$$\text{maximize} \sum_{i=1}^{n_e} q_i l_i + \sum_{i=n_e+1}^{n_e+|\mathcal{E}^b|} q_i l_i$$

Current segment

Problem Description and Constraints

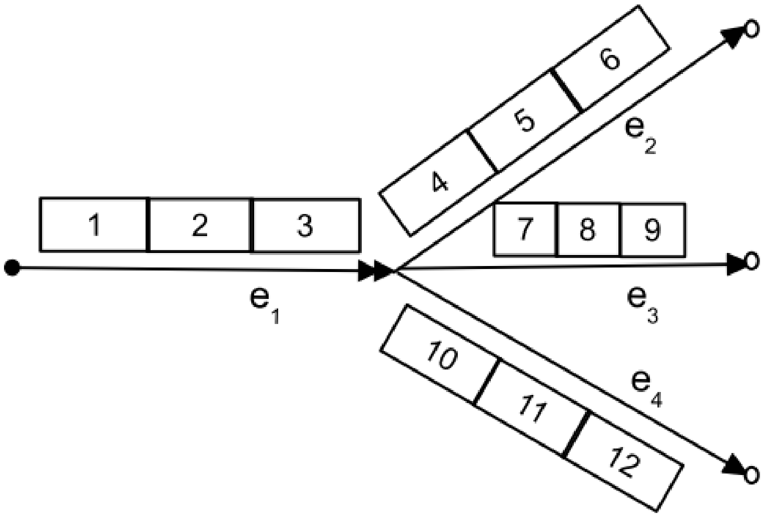


- Objective function

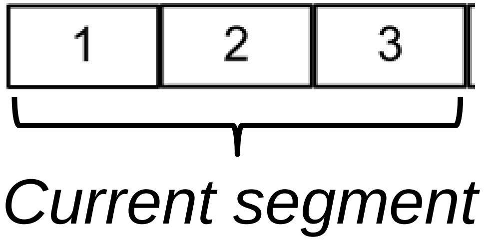
$$\text{maximize} \sum_{i=1}^{n_e} q_i l_i + \sum_{i=n_e+1}^{n_e+|\mathcal{E}^b|} q_i l_i$$

Beginning of next segment

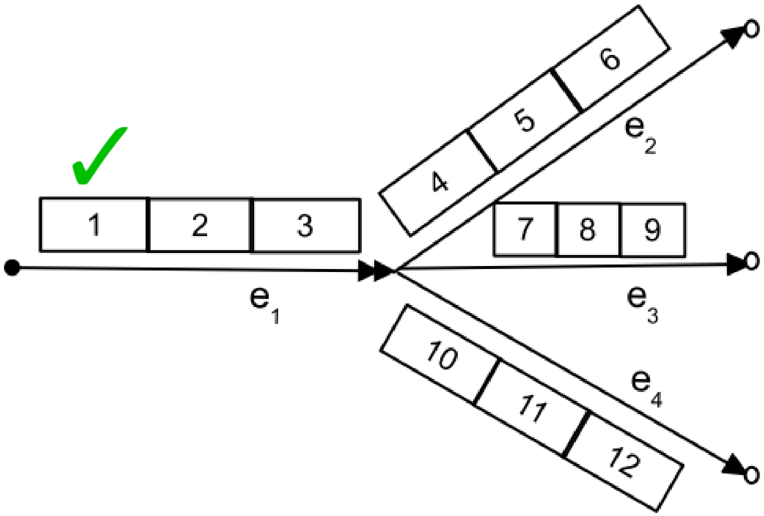
Problem Description and Constraints



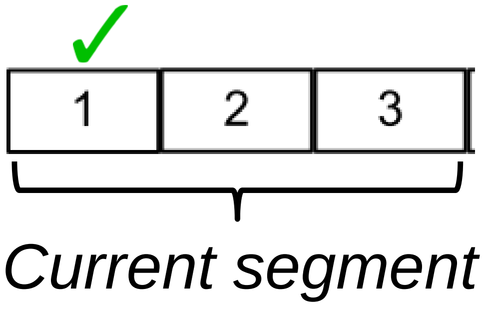
- Download order: round robin (optimal)



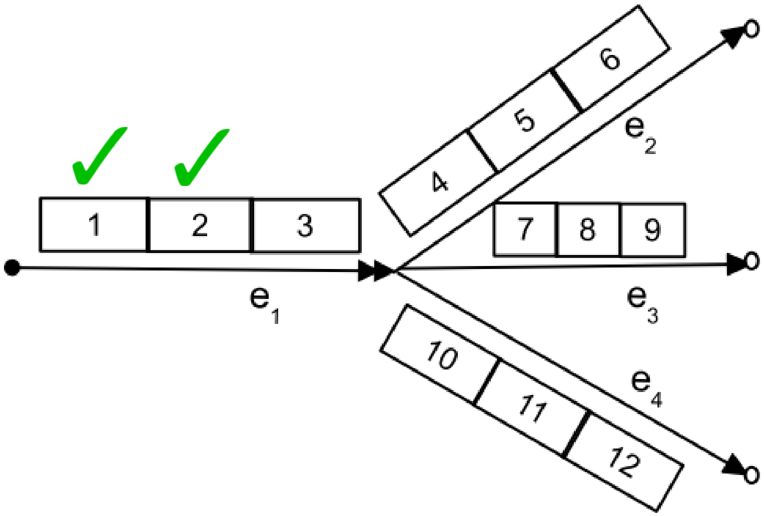
Problem Description and Constraints



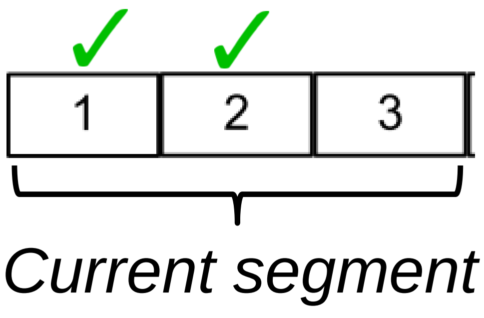
- Download order: round robin (optimal)



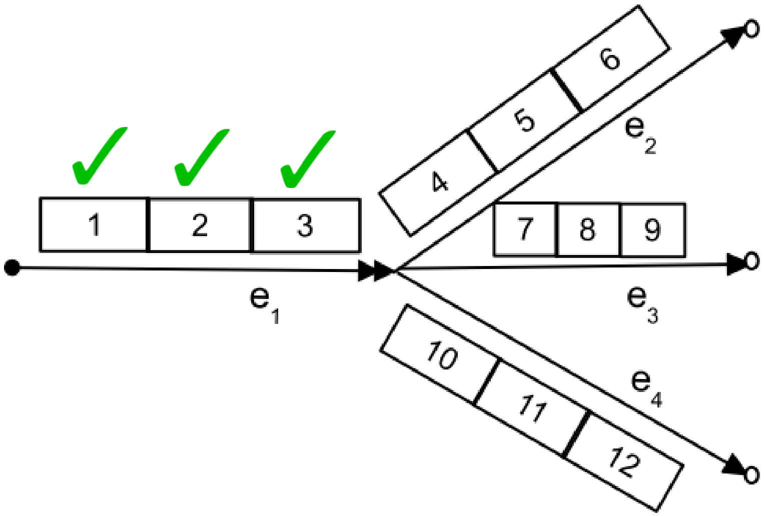
Problem Description and Constraints



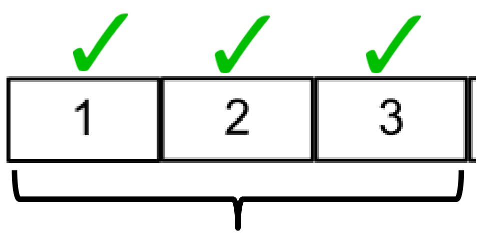
- Download order: round robin (optimal)



Problem Description and Constraints

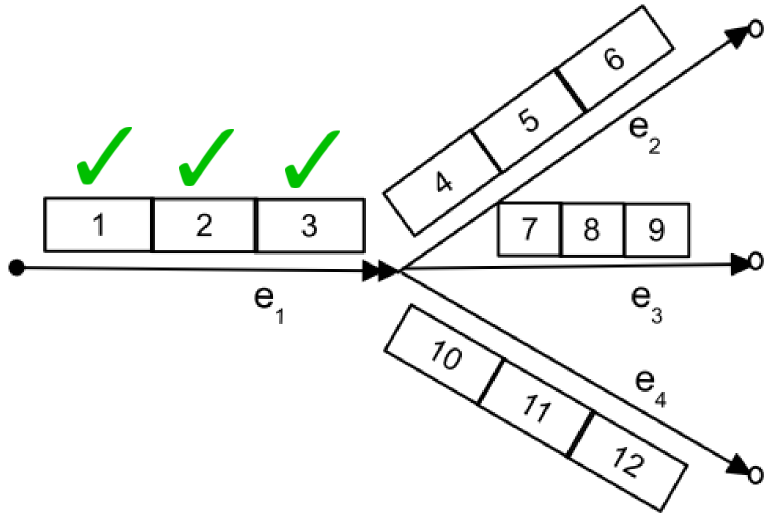


- Download order: round robin (optimal)

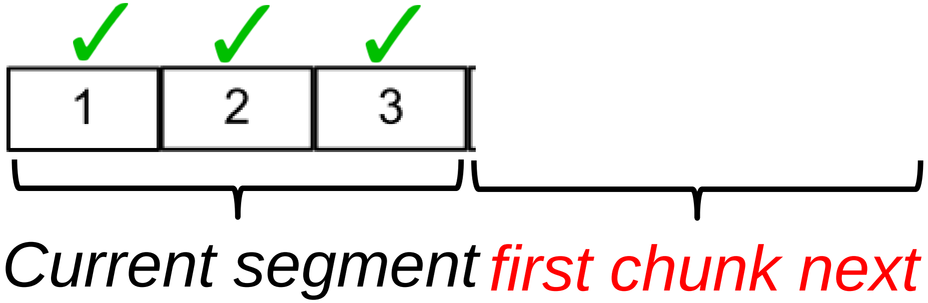


Current segment

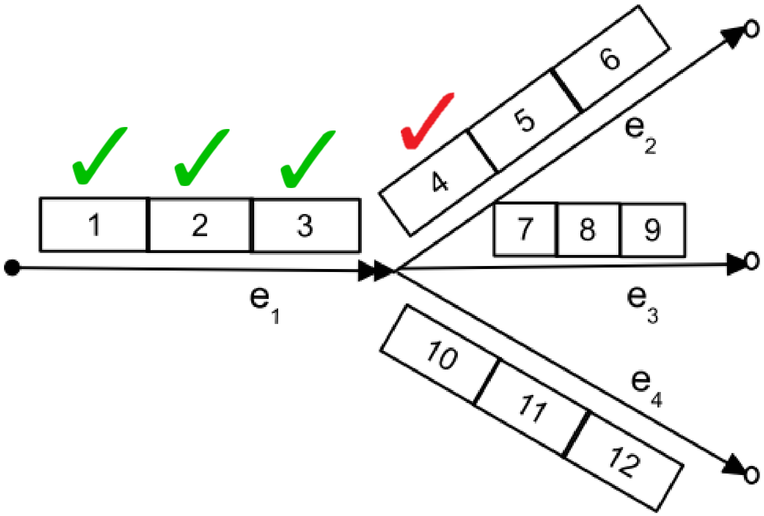
Problem Description and Constraints



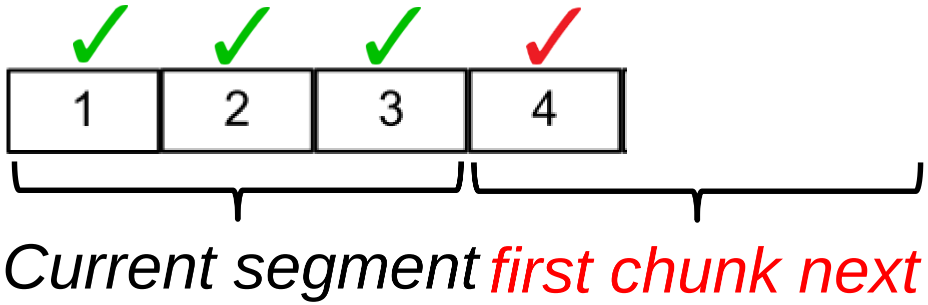
- Download order: round robin (optimal)



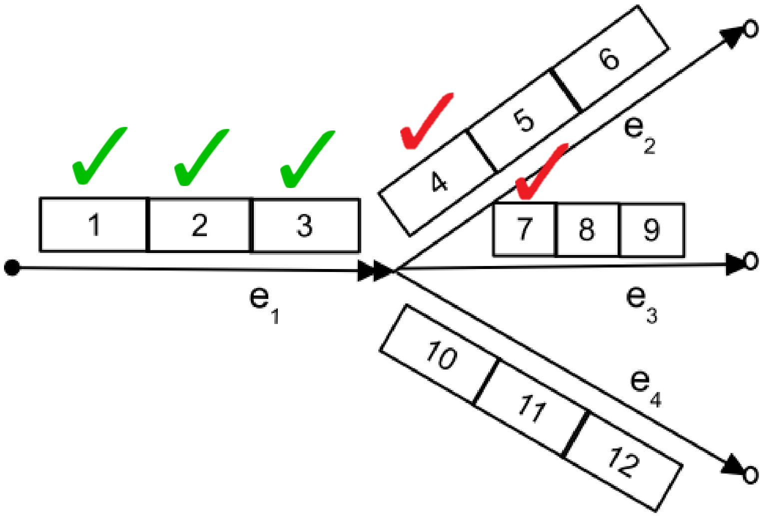
Problem Description and Constraints



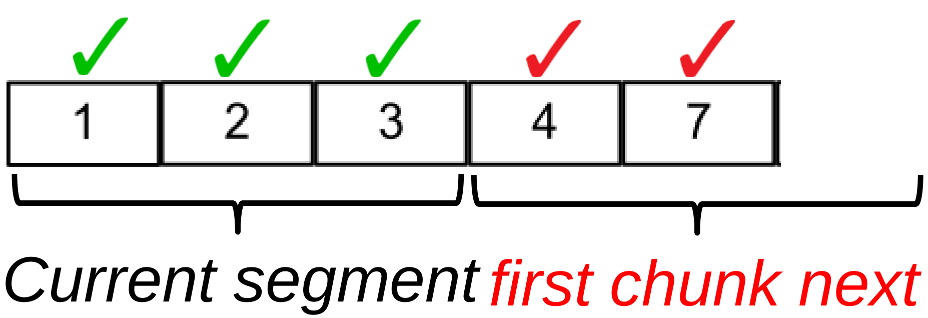
- Download order: round robin (optimal)



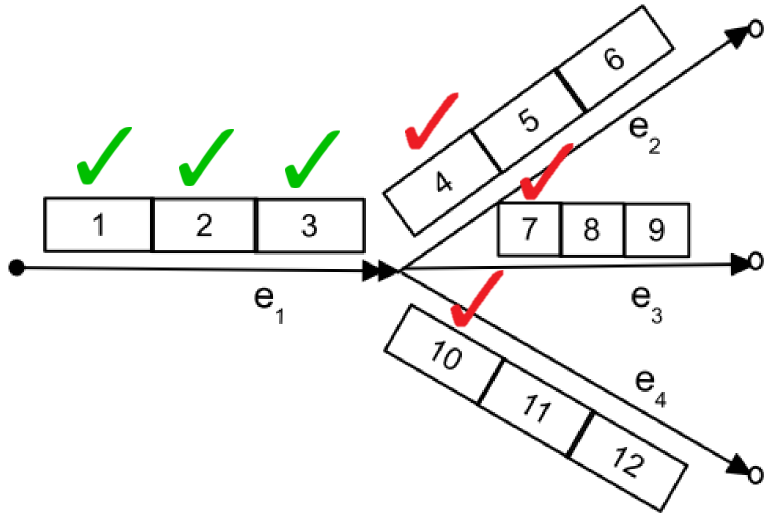
Problem Description and Constraints



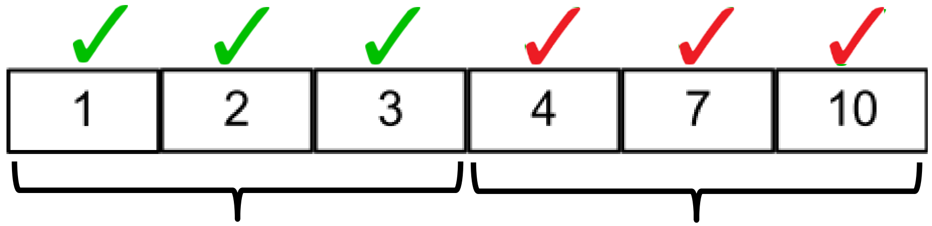
- Download order: round robin (optimal)



Problem Description and Constraints

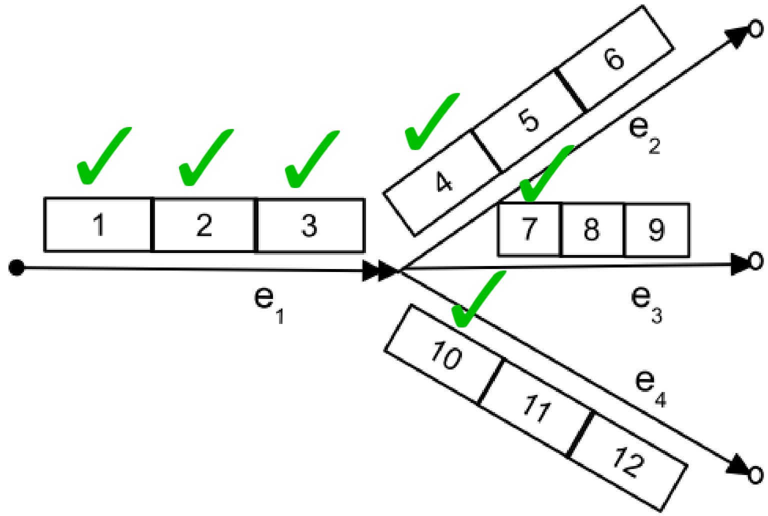


- Download order: round robin (optimal)

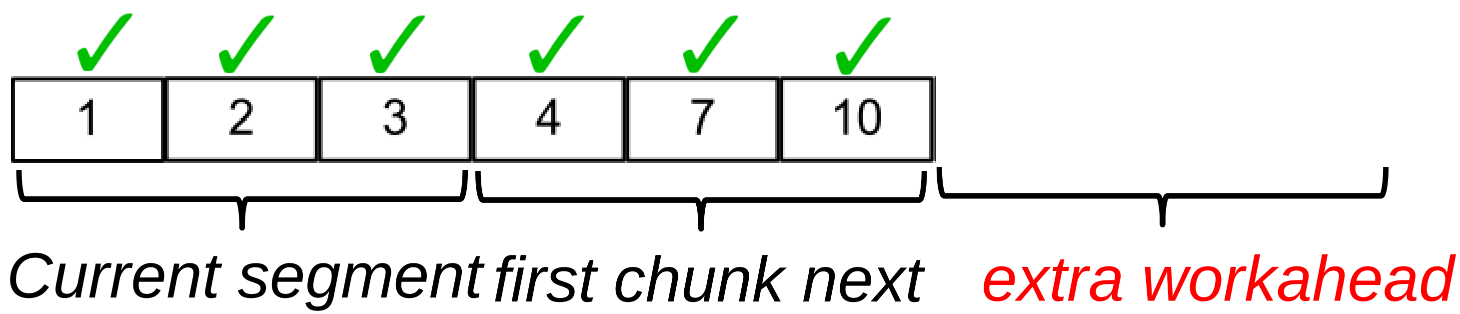


Current segment *first chunk next*

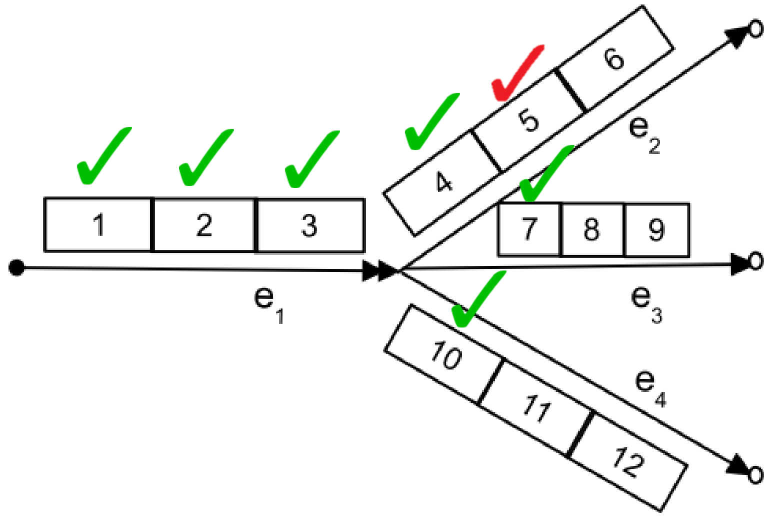
Problem Description and Constraints



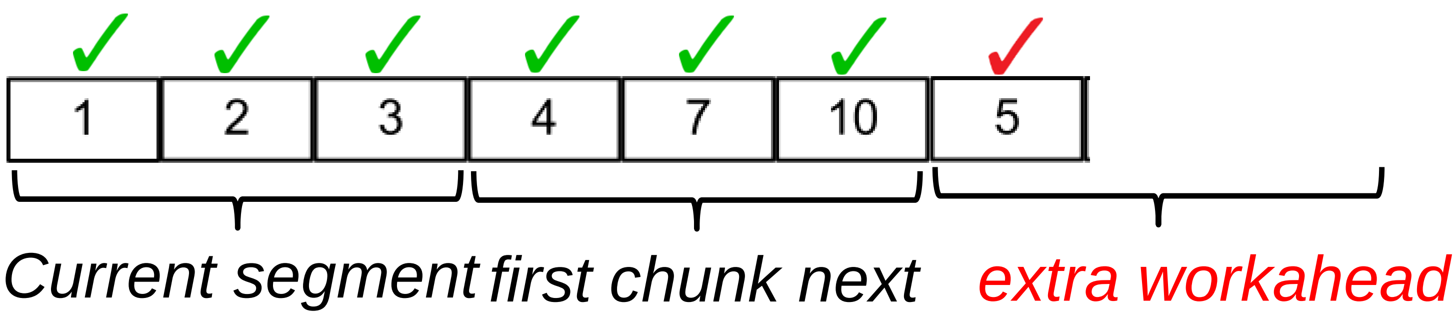
- Download order: round robin (extra workahead)



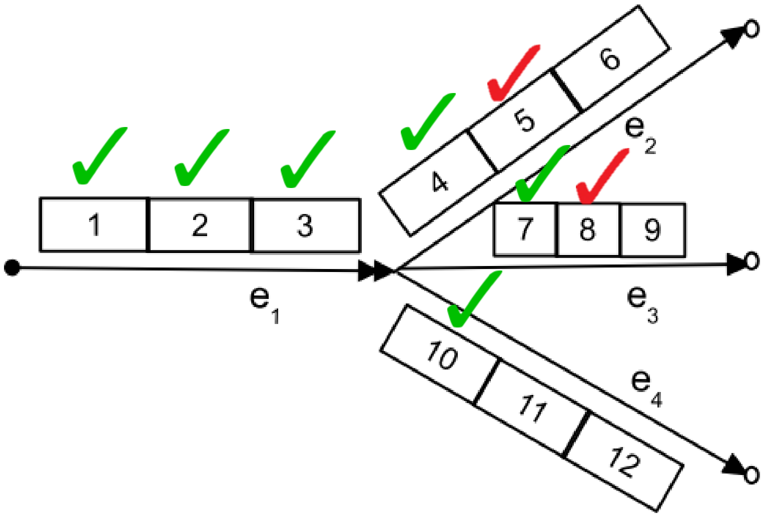
Problem Description and Constraints



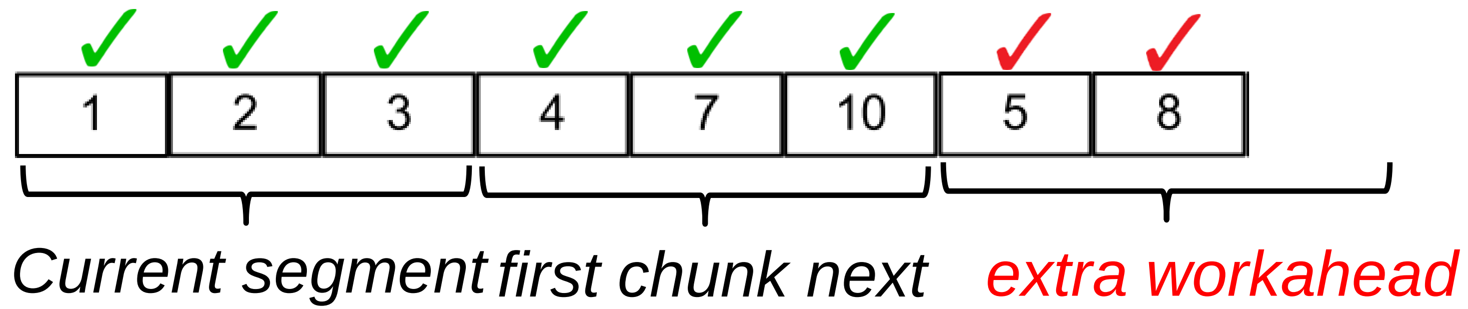
- Download order: round robin (extra workahead)



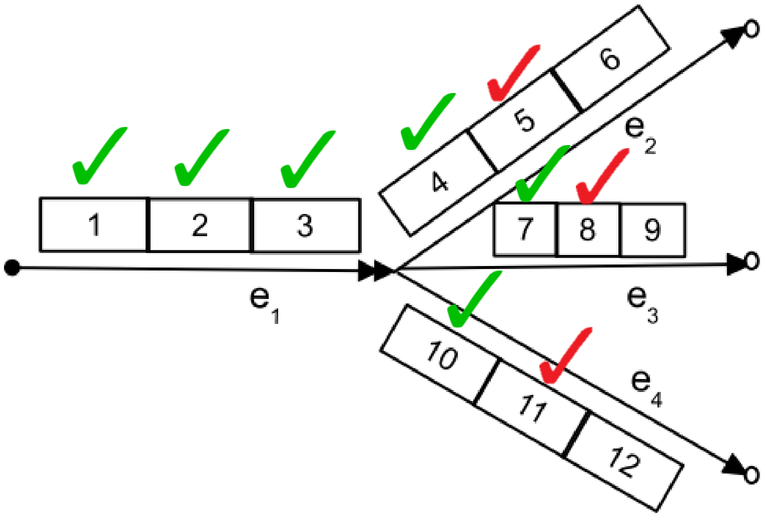
Problem Description and Constraints



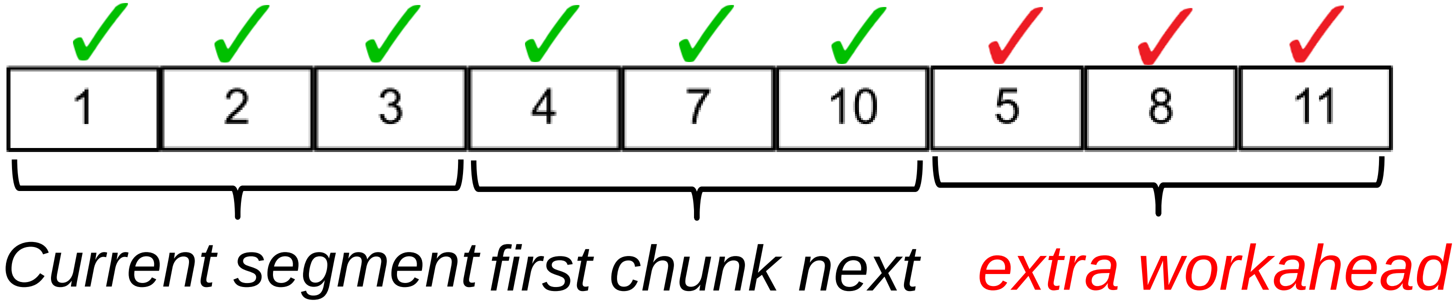
- Download order: round robin (extra workahead)



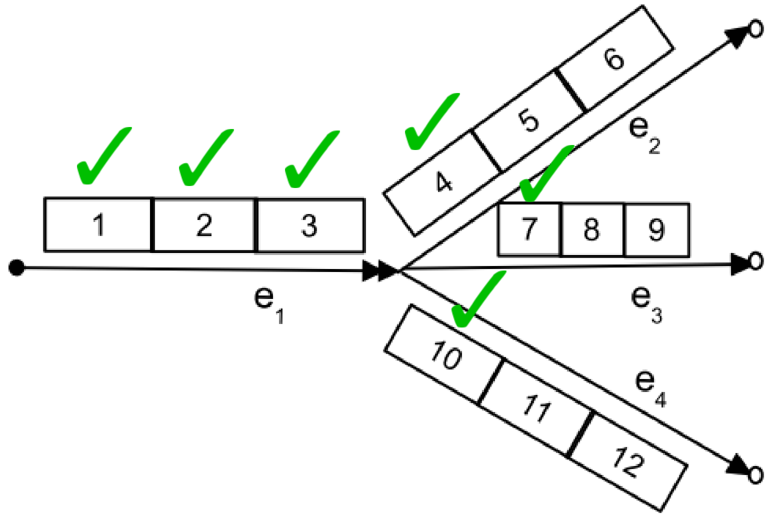
Problem Description and Constraints



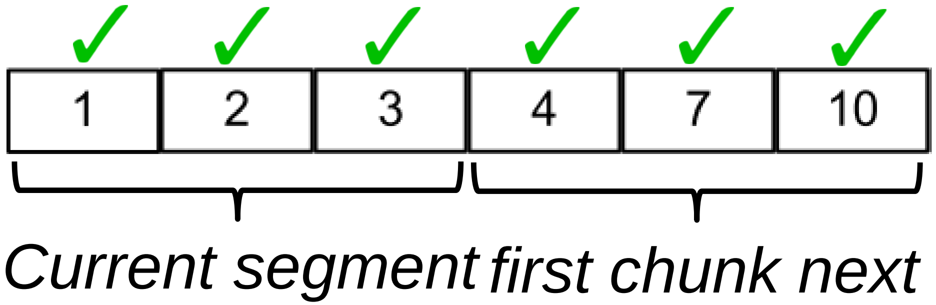
- Download order: round robin (extra workahead)



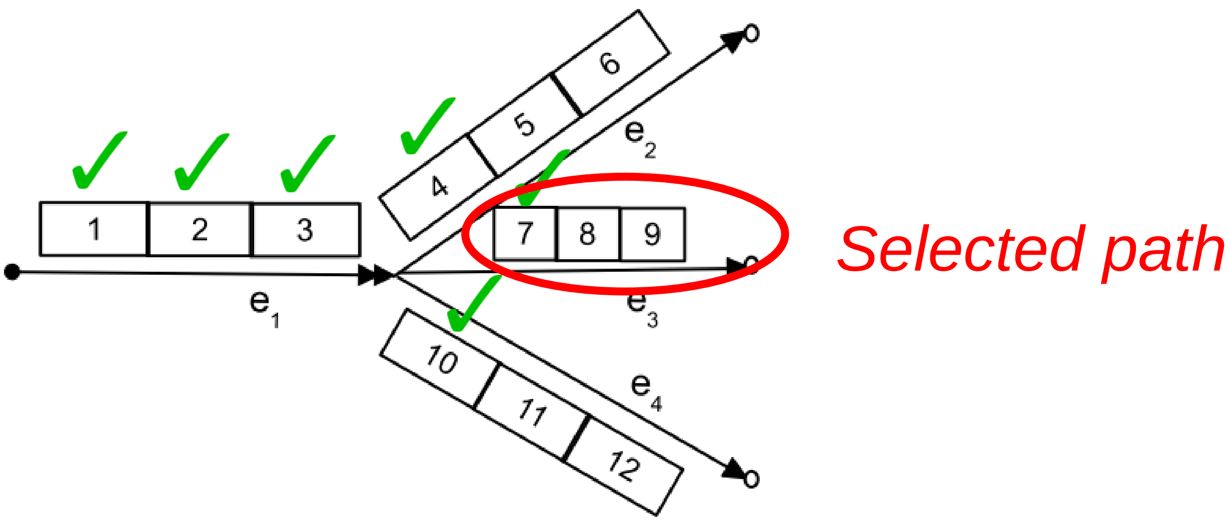
Problem Description and Constraints



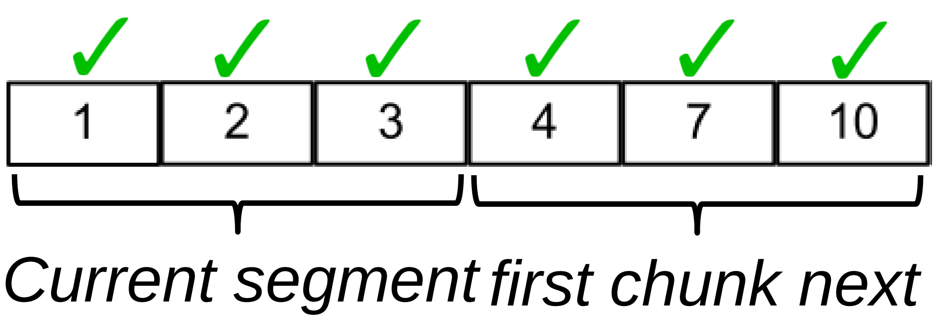
- Once branch point has been traversed, move on to next segment ...



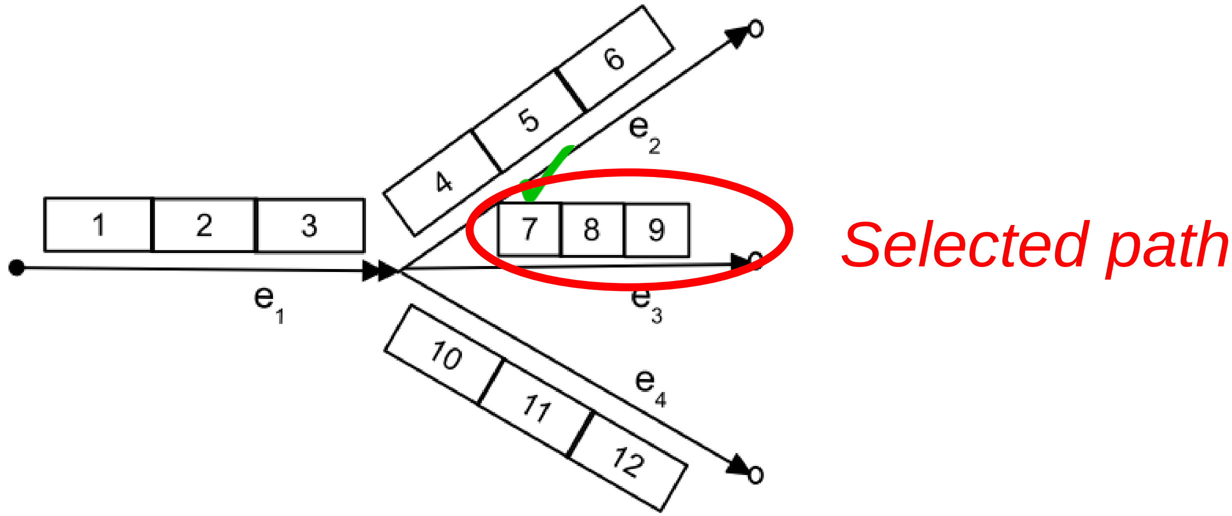
Problem Description and Constraints



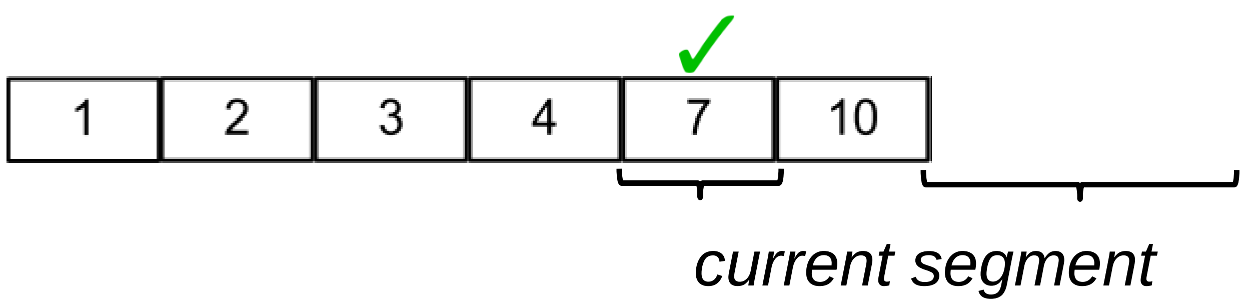
- Once branch point has been traversed, move on to next segment ...



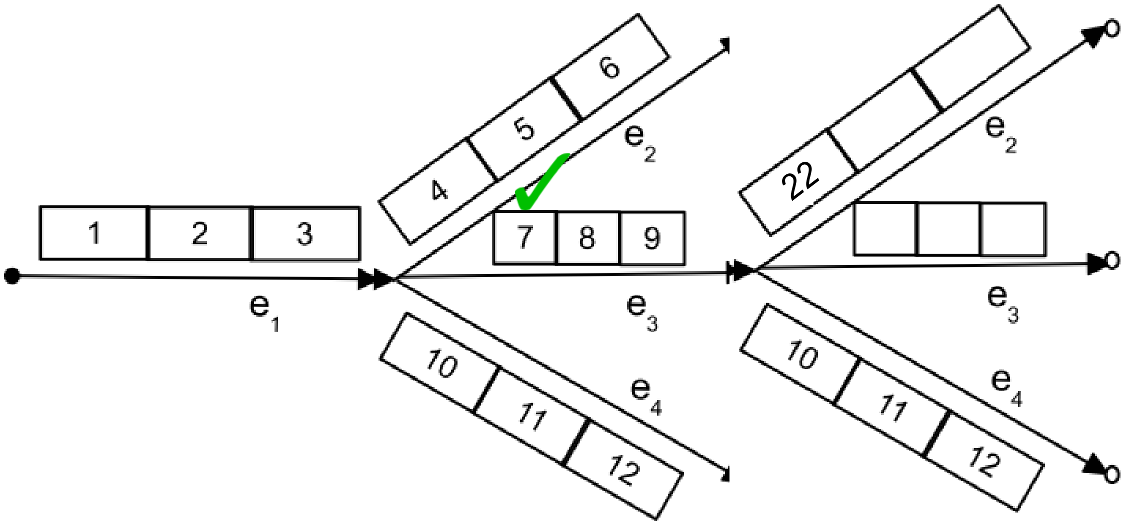
Problem Description and Constraints



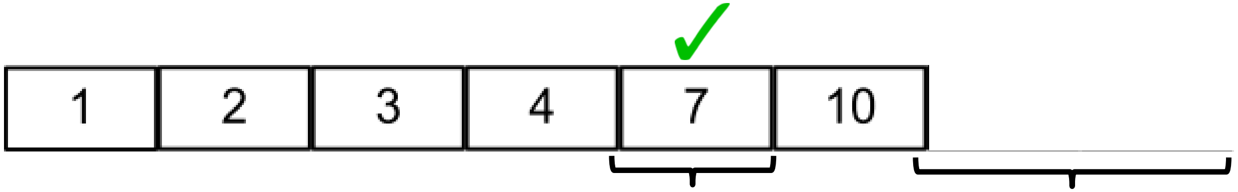
- Once branch point has been traversed, move on to next segment ...



Problem Description and Constraints

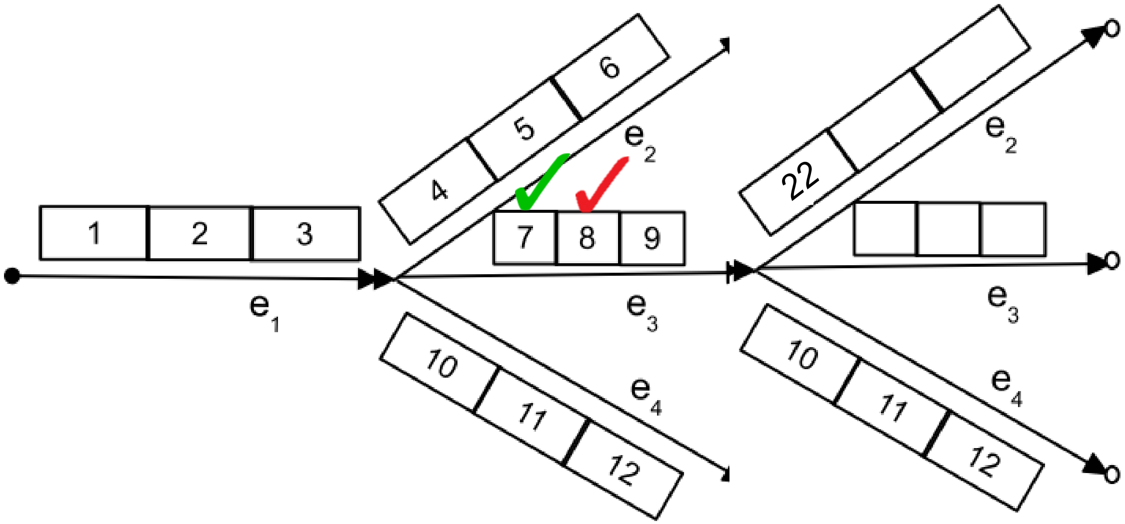


- Once branch point has been traversed, move on to next segment ...

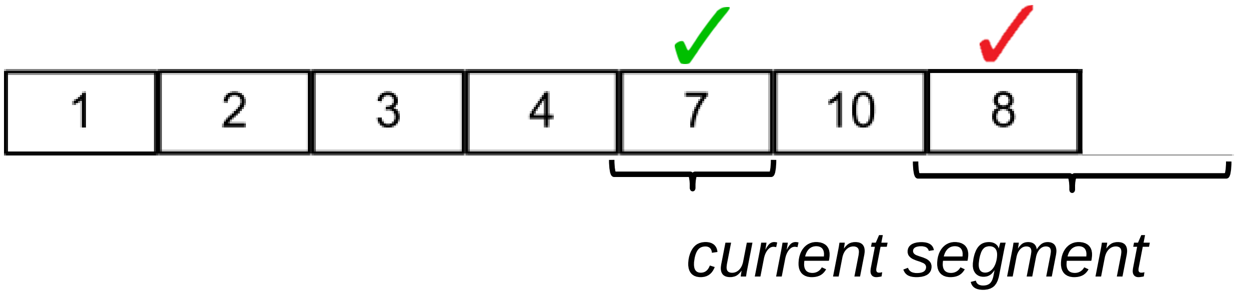


current segment

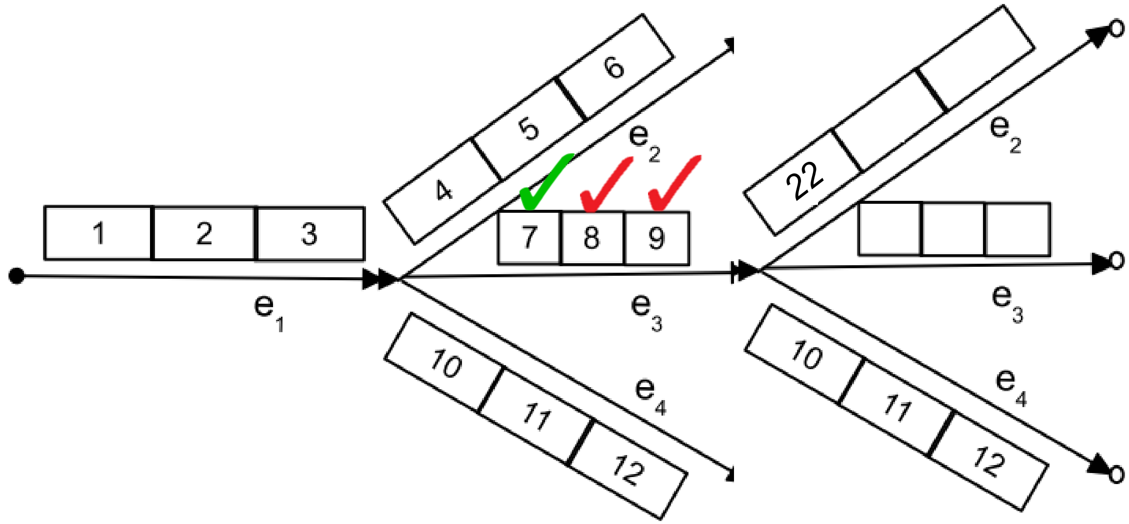
Problem Description and Constraints



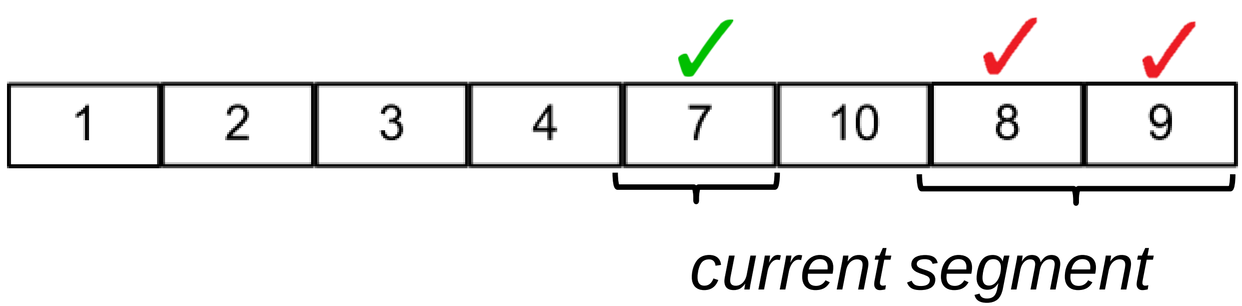
- Once branch point has been traversed, move on to next segment ...



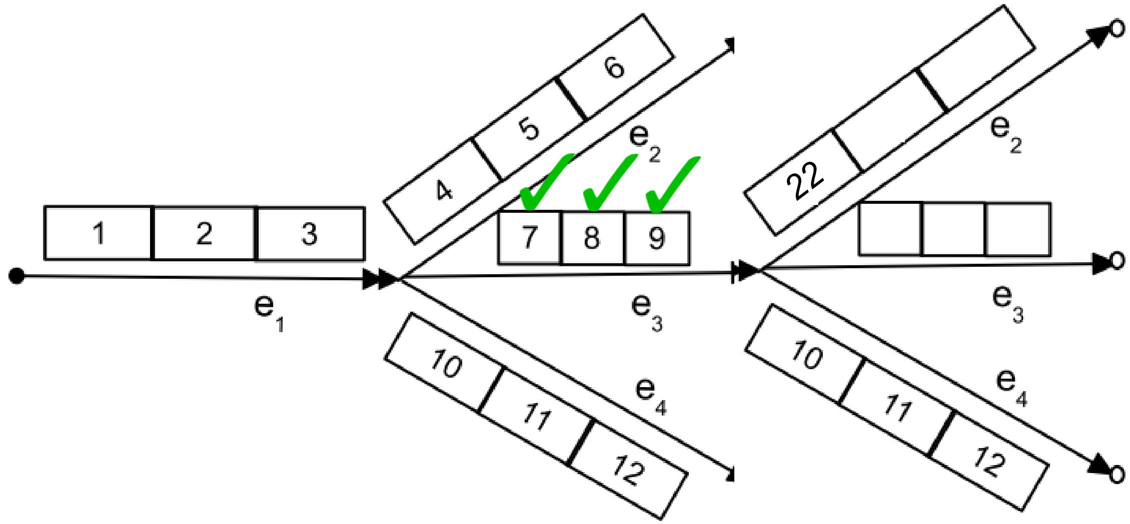
Problem Description and Constraints



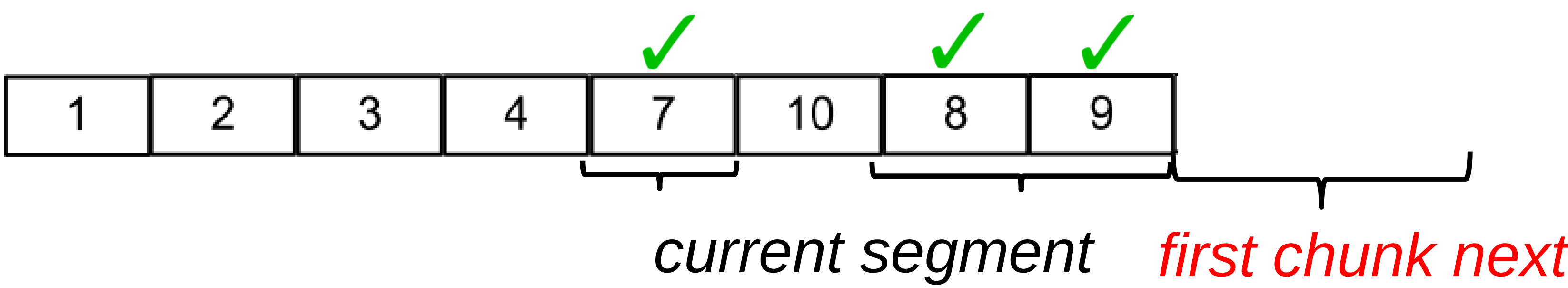
- Once branch point has been traversed, move on to next segment ...



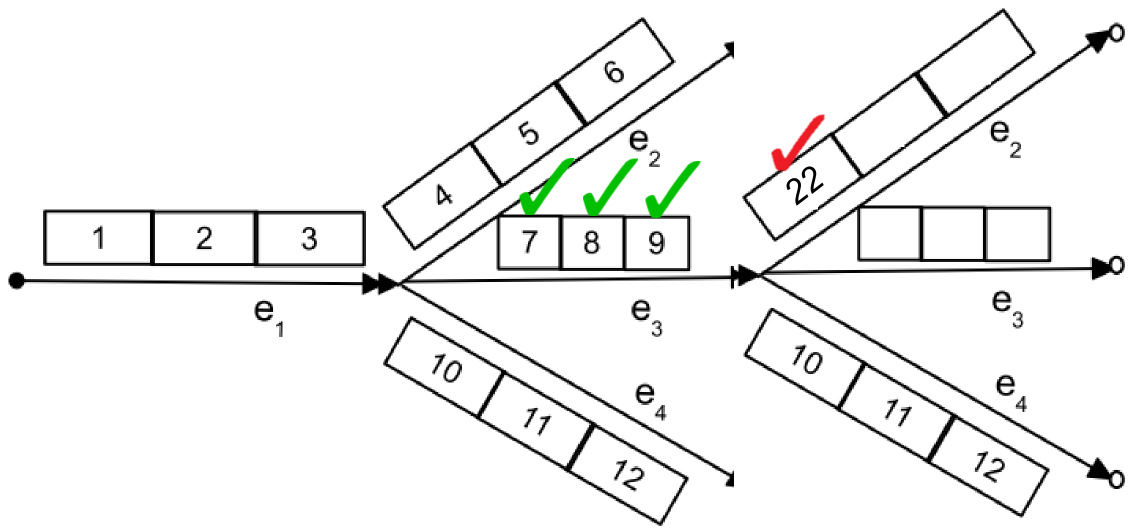
Problem Description and Constraints



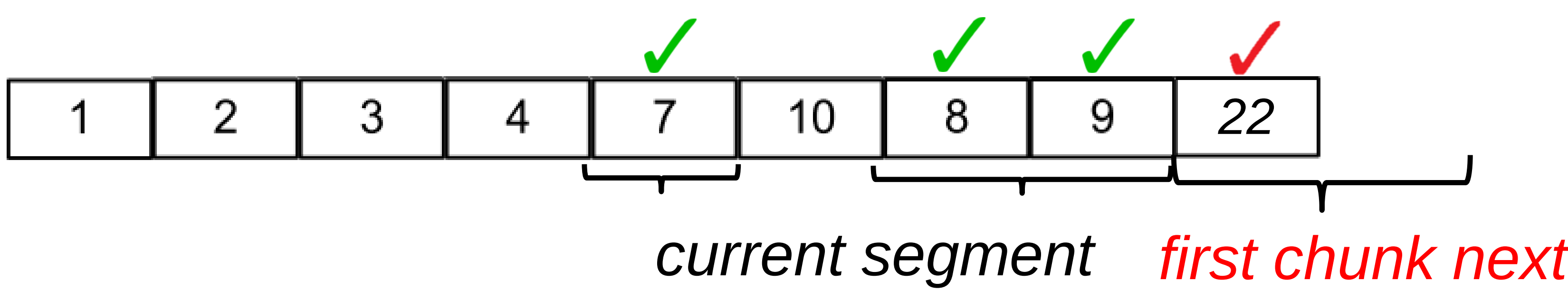
- Once branch point has been traversed, move on to next segment ...



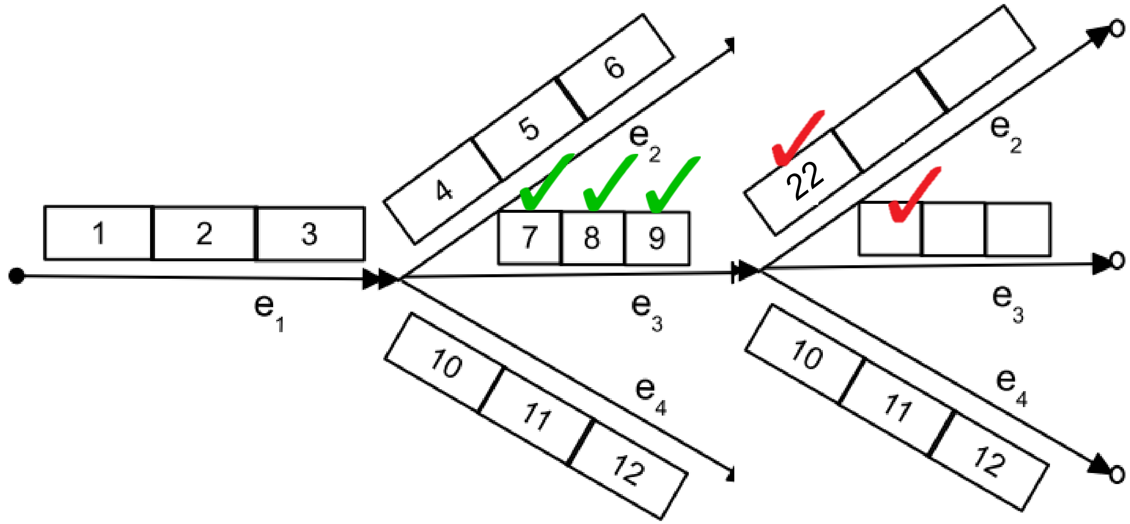
Problem Description and Constraints



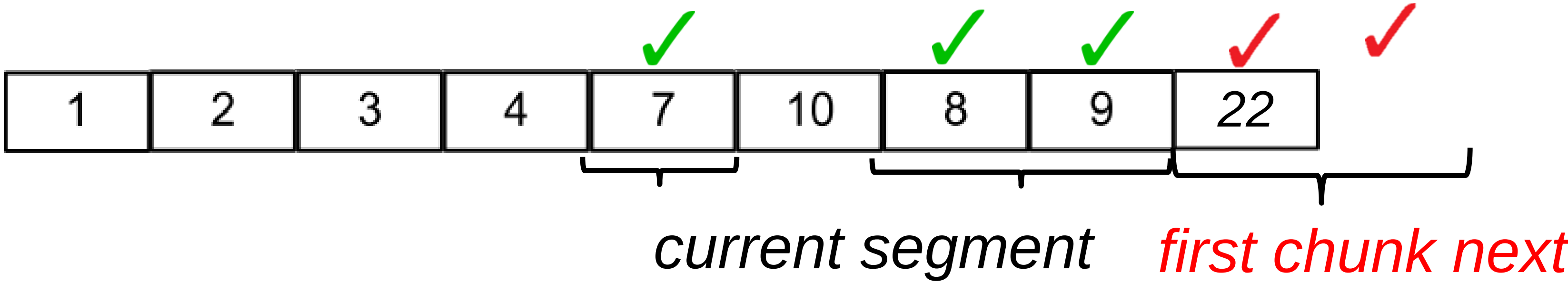
- Once branch point has been traversed, move on to next segment ...



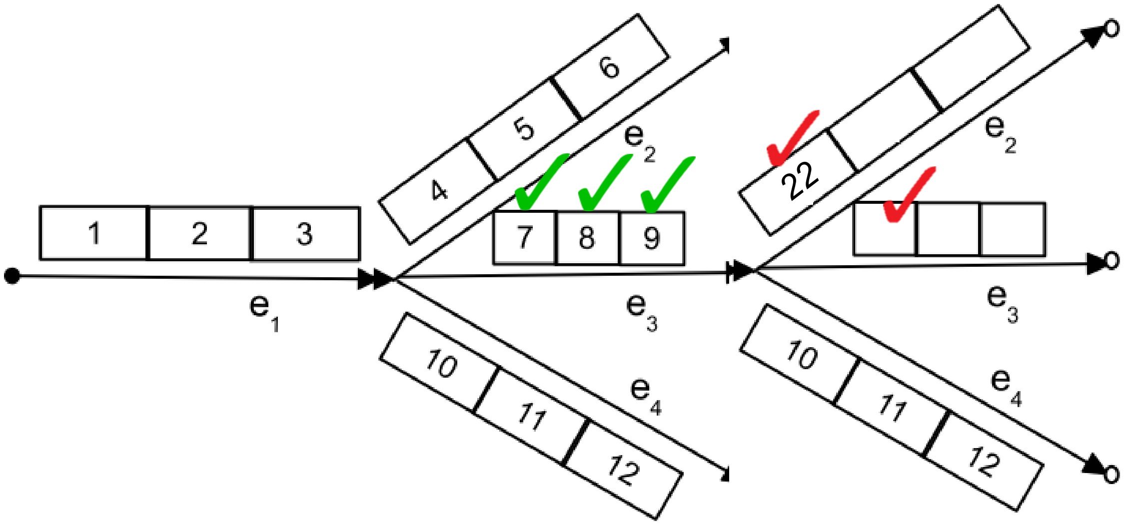
Problem Description and Constraints



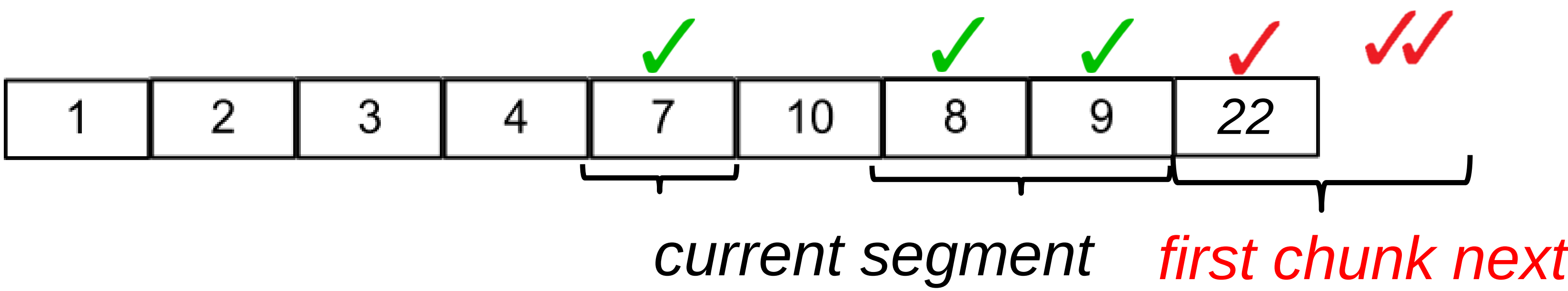
- Once branch point has been traversed, move on to next segment ...



Problem Description and Constraints

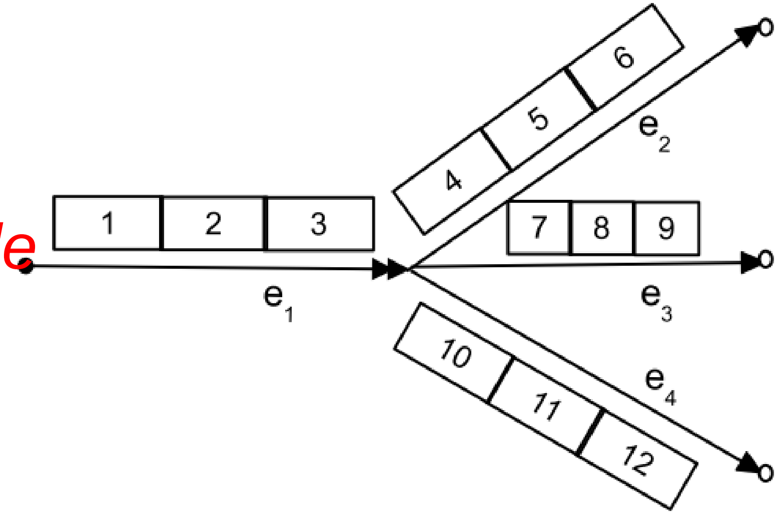


- Once branch point has been traversed, move on to next segment ...



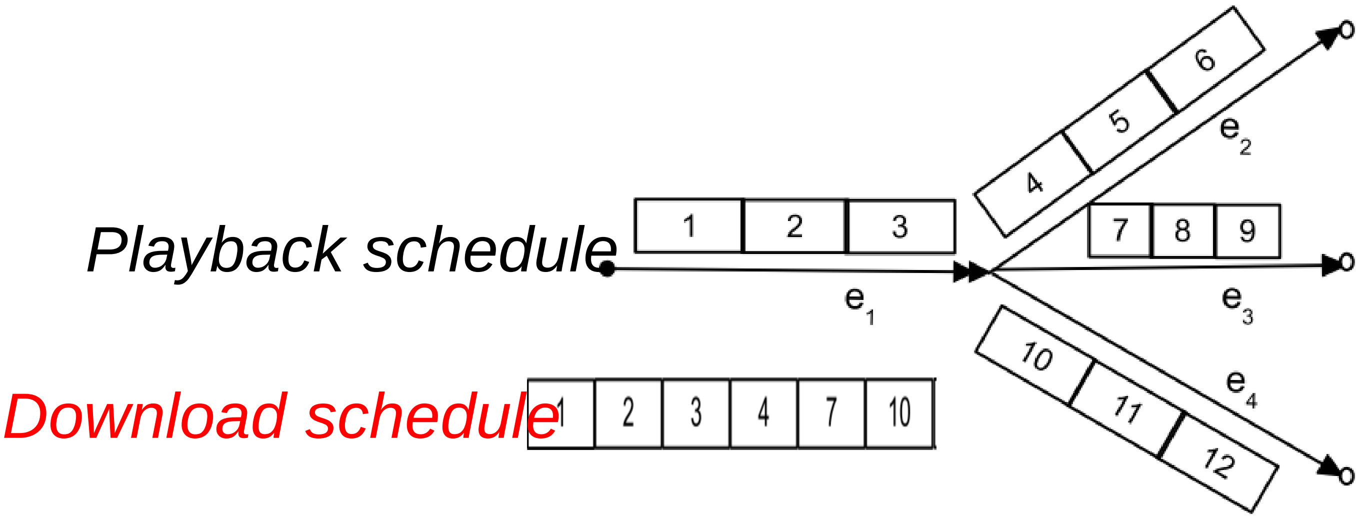
Problem Description and Constraints

Playback schedule



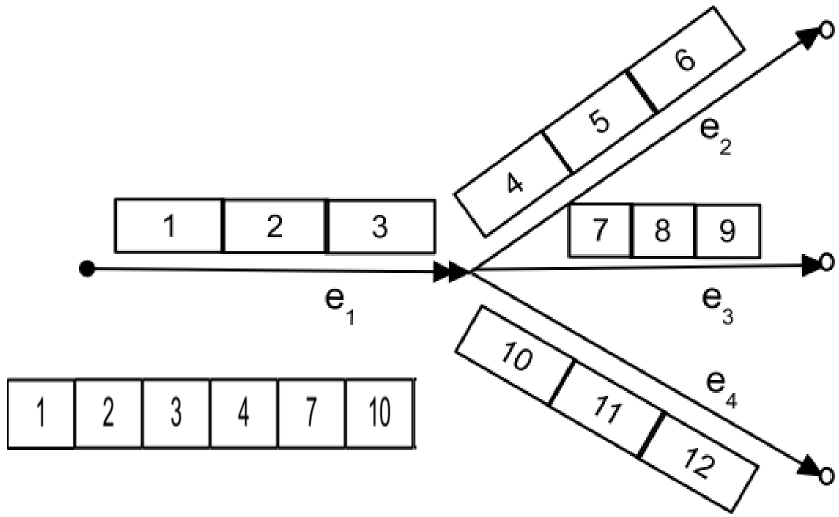
- Playback deadlines
  - for seamless playback without stalls

Problem Description and Constraints



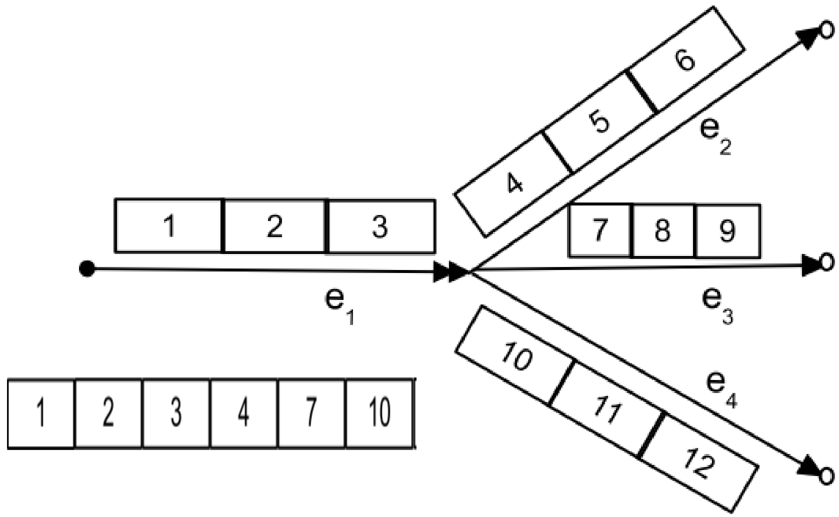
- Playback deadlines
  - for seamless playback without stalls

Problem Description and Constraints



- Playback deadlines
  - for seamless playback without stalls
  - Current segment: e.g., 2 and 3

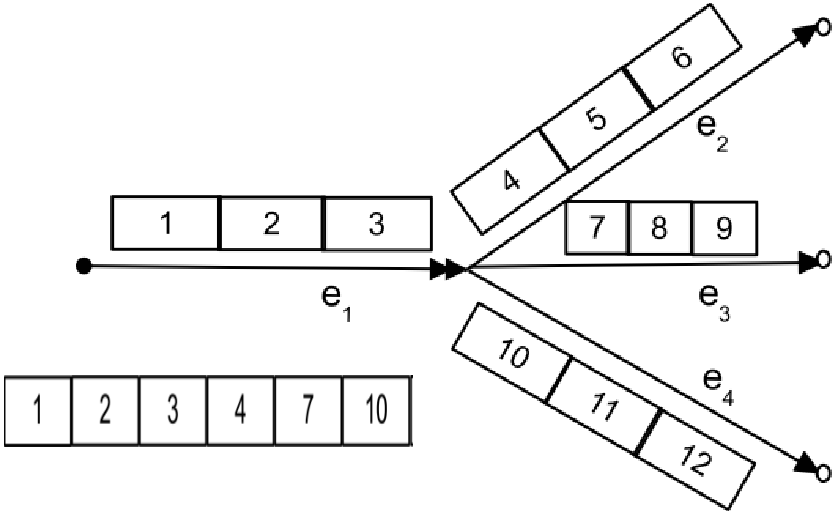
Problem Description and Constraints



- Playback deadlines
  - for seamless playback without stalls
  - Current segment: e.g., 2 and 3

$$t_i^c \leq t_i^d = \tau + \sum_{j=1}^{i-1} l_j, \quad \text{if } 1 \leq i \leq n_e$$

Problem Description and Constraints

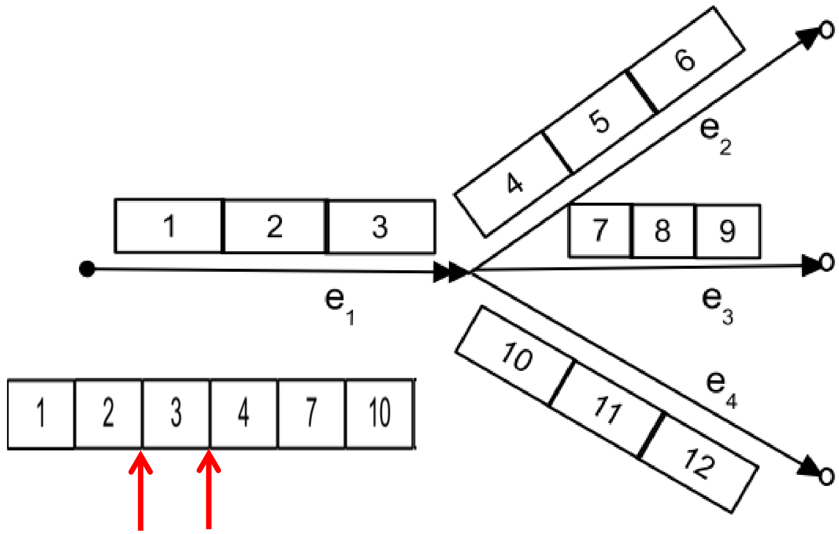


- Playback deadlines
  - for seamless playback without stalls
  - Current segment: e.g., 2 and 3

$$t_i^c \leq t_i^d = \tau + \sum_{j=1}^{i-1} l_j, \quad \text{if } 1 \leq i \leq n_e$$

Download completion time

Problem Description and Constraints



Download completion times

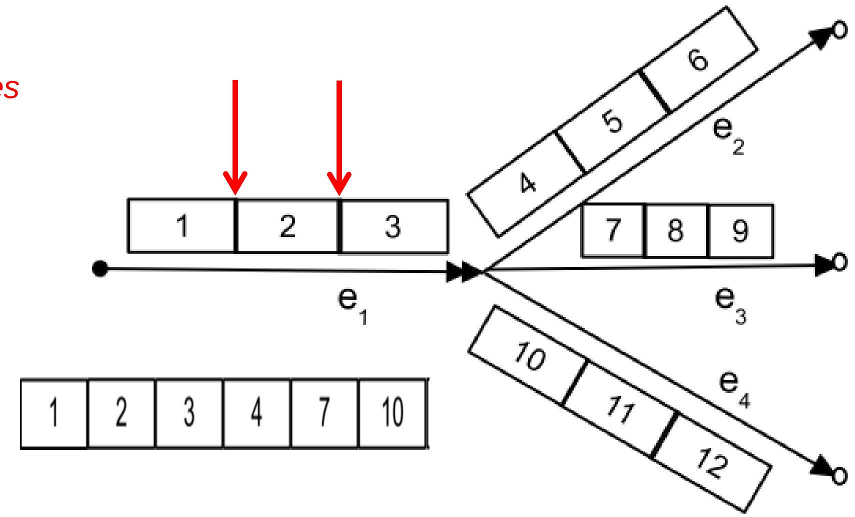
- Playback deadlines
  - for seamless playback without stalls
  - Current segment: e.g., 2 and 3

$$t_i^c \leq t_i^d = \tau + \sum_{j=1}^{i-1} l_j, \quad \text{if } 1 \leq i \leq n_e$$

Download completion time

Problem Description and Constraints

Playback deadlines



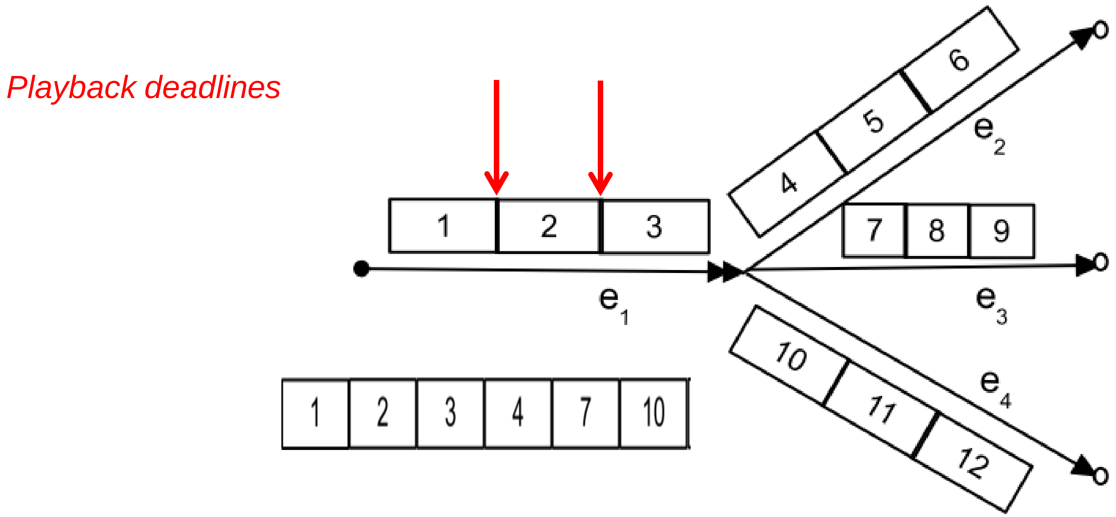
- Playback deadlines
  - for seamless playback without stalls
  - Current segment: e.g., 2 and 3

$$t_i^c \le t_i^d = \tau + \sum_{j=1}^{i-1} l_j, \quad \text{if } 1 \le i \le n_e$$

Time of playback deadline

Download completion time

Problem Description and Constraints



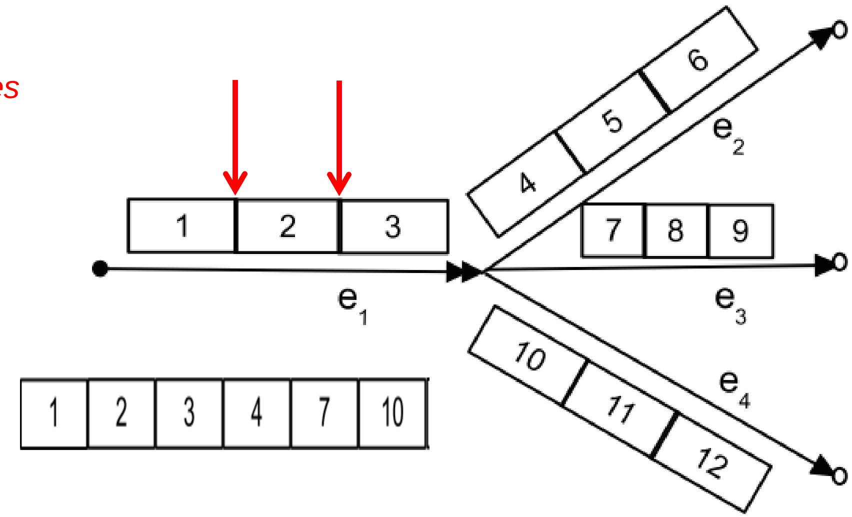
- Playback deadlines
  - for seamless playback without stalls
  - Current segment: e.g., 2 and 3

$$t_i^c \leq t_i^d = \tau + \sum_{j=1}^{i-1} l_j, \quad \text{if } 1 \leq i \leq n_e$$

Time of playback deadline

Problem Description and Constraints

Playback deadlines

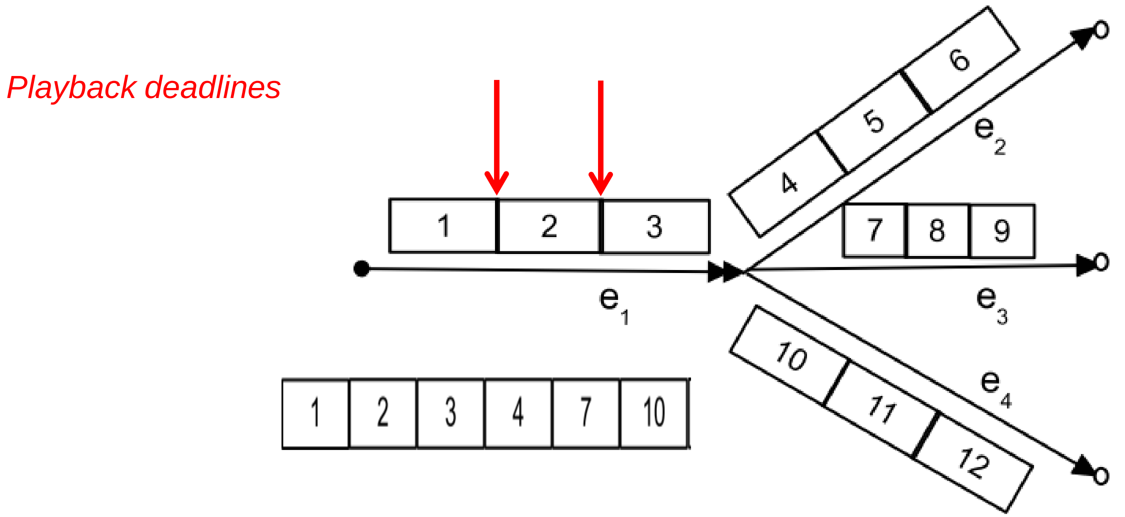


- Playback deadlines
  - for seamless playback without stalls
  - Current segment: e.g., 2 and 3

$$t_i^c \leq t_i^d = \tau + \sum_{j=1}^{i-1} l_j, \quad \text{if } 1 \leq i \leq n_e$$

Startup delay

Problem Description and Constraints



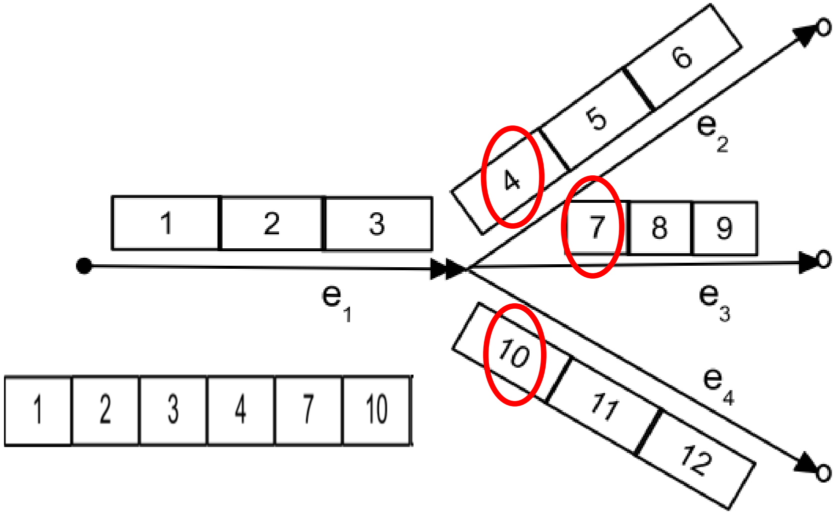
- Playback deadlines
  - for seamless playback without stalls
  - Current segment: e.g., 2 and 3

$$t_i^c \leq t_i^d = \tau + \sum_{j=1}^{i-1} l_j, \quad \text{if } 1 \leq i \leq n_e$$

Startup delay

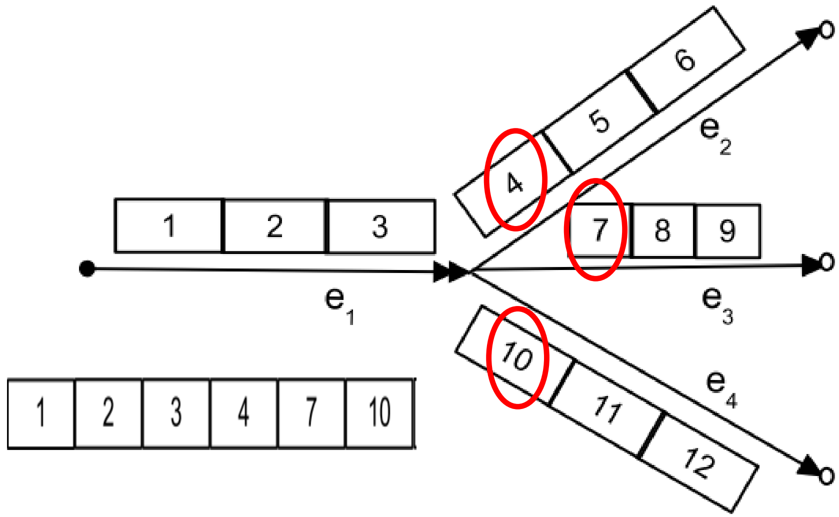
Playtime of earlier chunks

Problem Description and Constraints



- Playback deadlines
  - for seamless playback without stalls
  - First chunks next segment: e.g., 4, 7, and 10

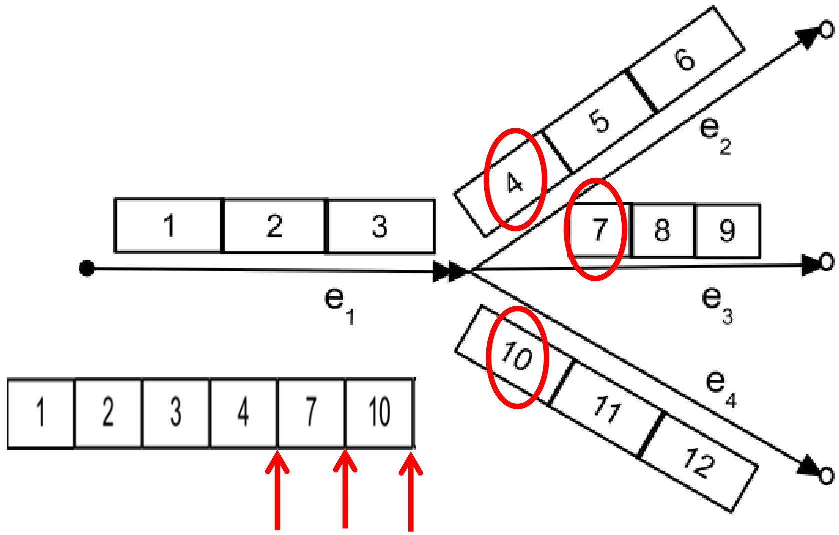
Problem Description and Constraints



- Playback deadlines
  - for seamless playback without stalls
  - First chunks next segment: e.g., 4, 7, and 10

$$t_i^c \le t_i^d = \tau + \sum_{j=1}^{n_e} l_j, \quad \text{if } n_e < i \le n_e + |\mathcal{E}^b|$$

Problem Description and Constraints



Download completion times

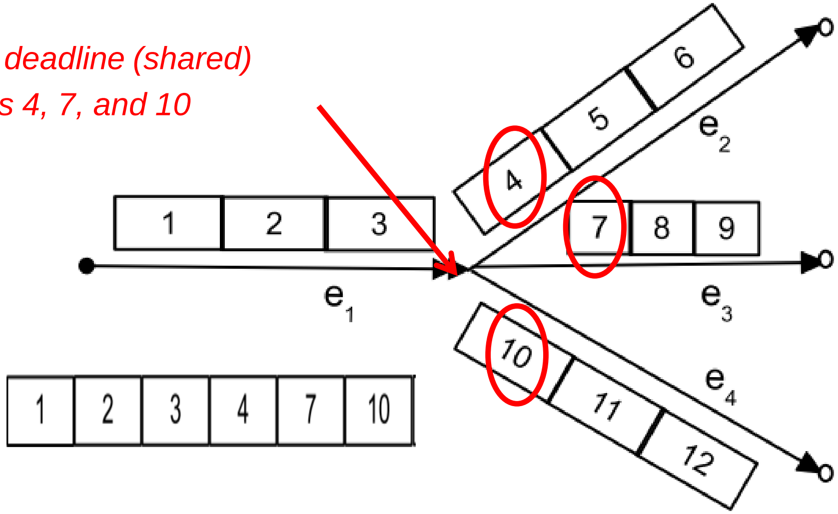
- Playback deadlines
  - for seamless playback without stalls
  - First chunks next segment: e.g., 4, 7, and 10

$$t_i^c \leq t_i^d = \tau + \sum_{j=1}^{n_e} l_j, \quad \text{if } n_e < i \leq n_e + |\mathcal{E}^b|$$

Download completion times

Problem Description and Constraints

Playback deadline (shared) for chunks 4, 7, and 10



- Playback deadlines
  - for seamless playback without stalls
  - First chunks next segment: e.g., 4, 7, and 10

$$t_i^c \leq t_i^d = \tau + \sum_{j=1}^{n_e} l_j, \quad \text{if } n_e < i \leq n_e + |\mathcal{E}^b|$$

Time at which branch point is reached

Download completion times

Problem Description and Constraints

1	2	3	4	7	10

Download completion times

$$t_i^c \leq t_i^d = \tau + \sum_{j=1}^{n_e} l_j, \quad \text{if } n_e < i \leq n_e + |\mathcal{E}^b|$$

Download completion times

Problem Description and Constraints

- Download times t_i^c , rate estimations, and parallel connections
  -
  -
  -

Problem Description and Constraints

- Download times t_i^c , rate estimations, and parallel connections
  - At the end of a chunk download, schedule new downloads and new TCP connections
  - Assume that an additional TCP connection will not increase the total download rate
  - New connections are initiated only if it is not expected to lead to playback deadline violations

Problem Description and Constraints

- Download times t_i^c , rate estimations, and parallel connections
  - At the end of a chunk download, schedule new downloads and new TCP connections
  - Assume that an additional TCP connection will not increase the total download rate
  - New connections are initiated only if it is not expected to lead to playback deadline violations

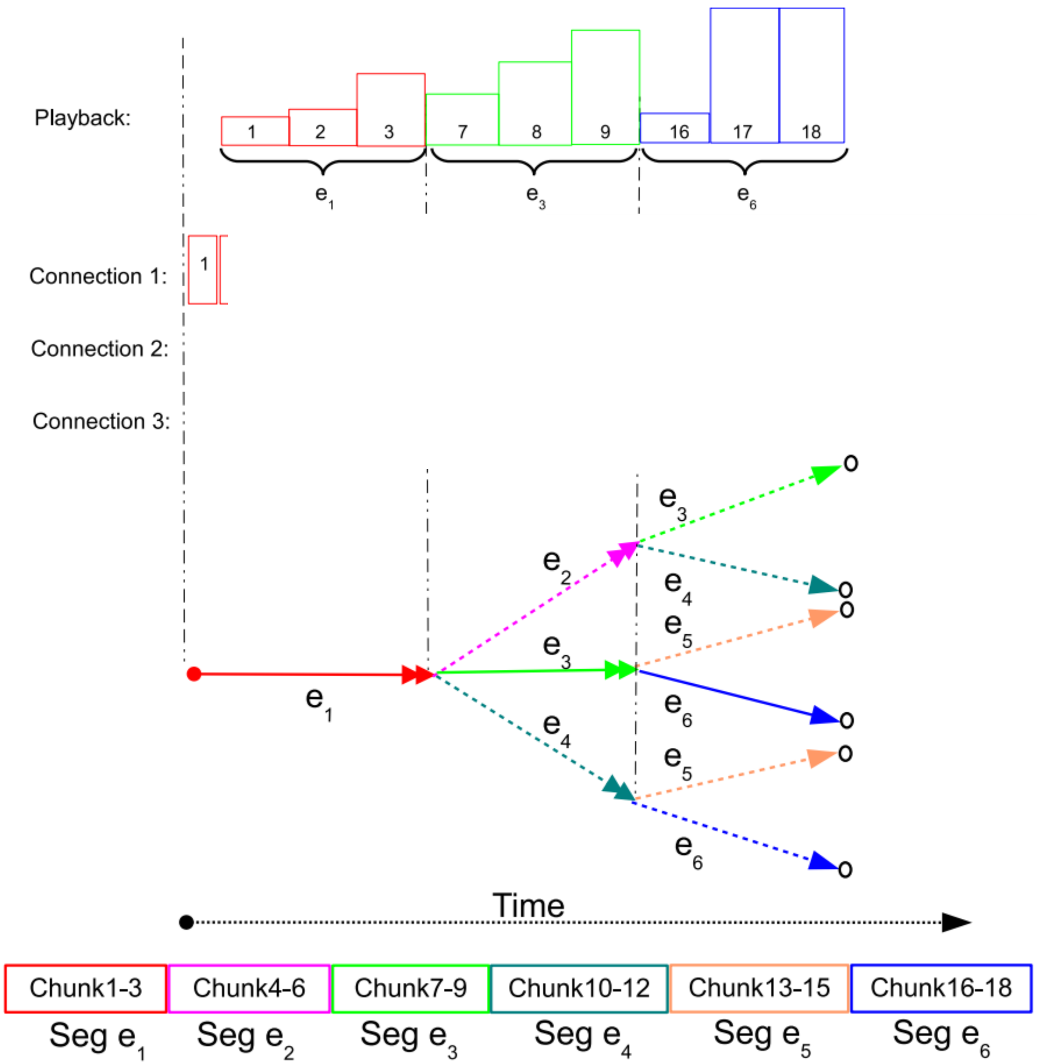
Problem Description and Constraints

- Download times t_i^c , rate estimations, and parallel connections
  - At the end of a chunk download, schedule new downloads and new TCP connections
  - Assume that an additional TCP connection will not increase the total download rate
  - New connections are initiated only if it is not expected to lead to playback deadline violations

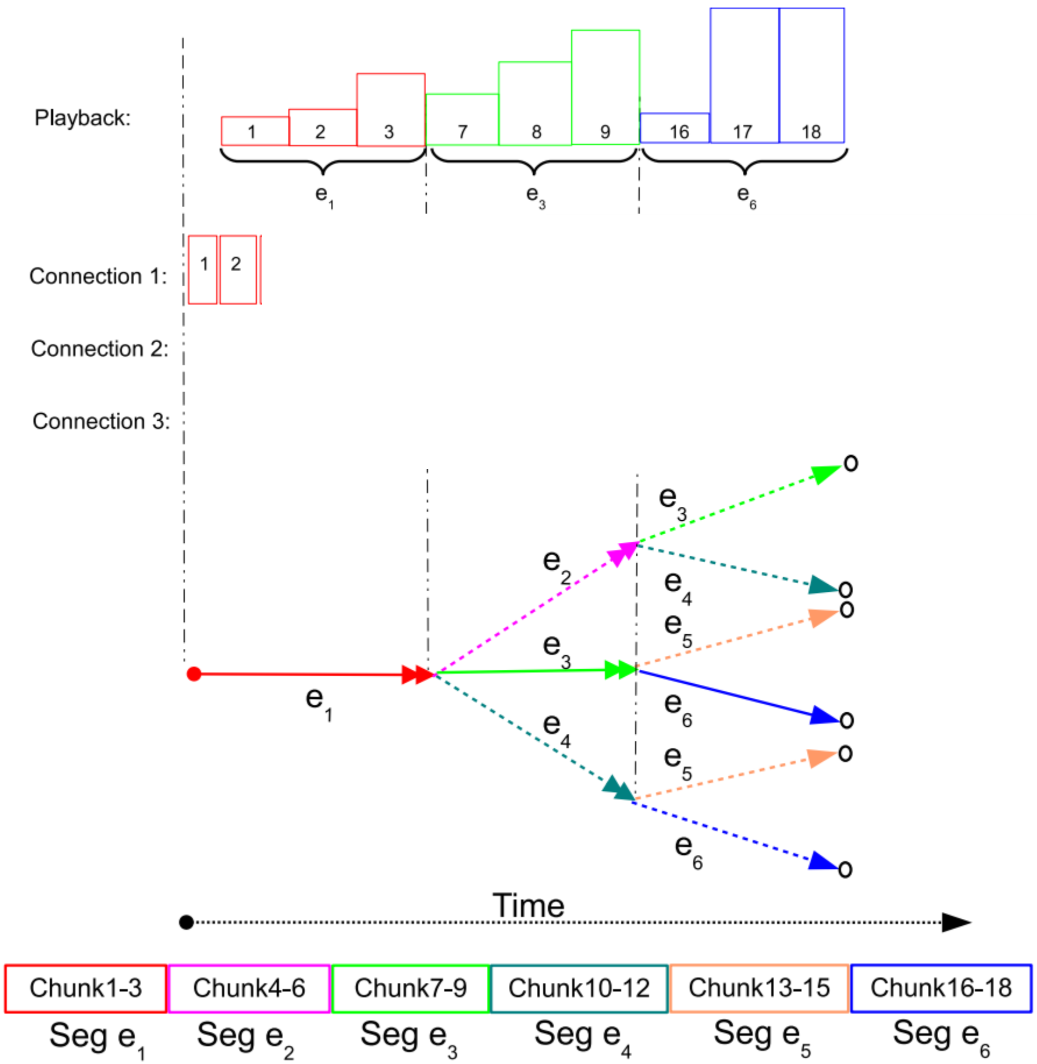
Problem Description and Constraints

- Download times t_i^c, rate estimations, and parallel connections
  - At the end of a chunk download, schedule new downloads and new TCP connections
  - Assume that an additional TCP connection will not increase the total download rate
  - New connections are initiated only if it is not expected to lead to playback deadline violations

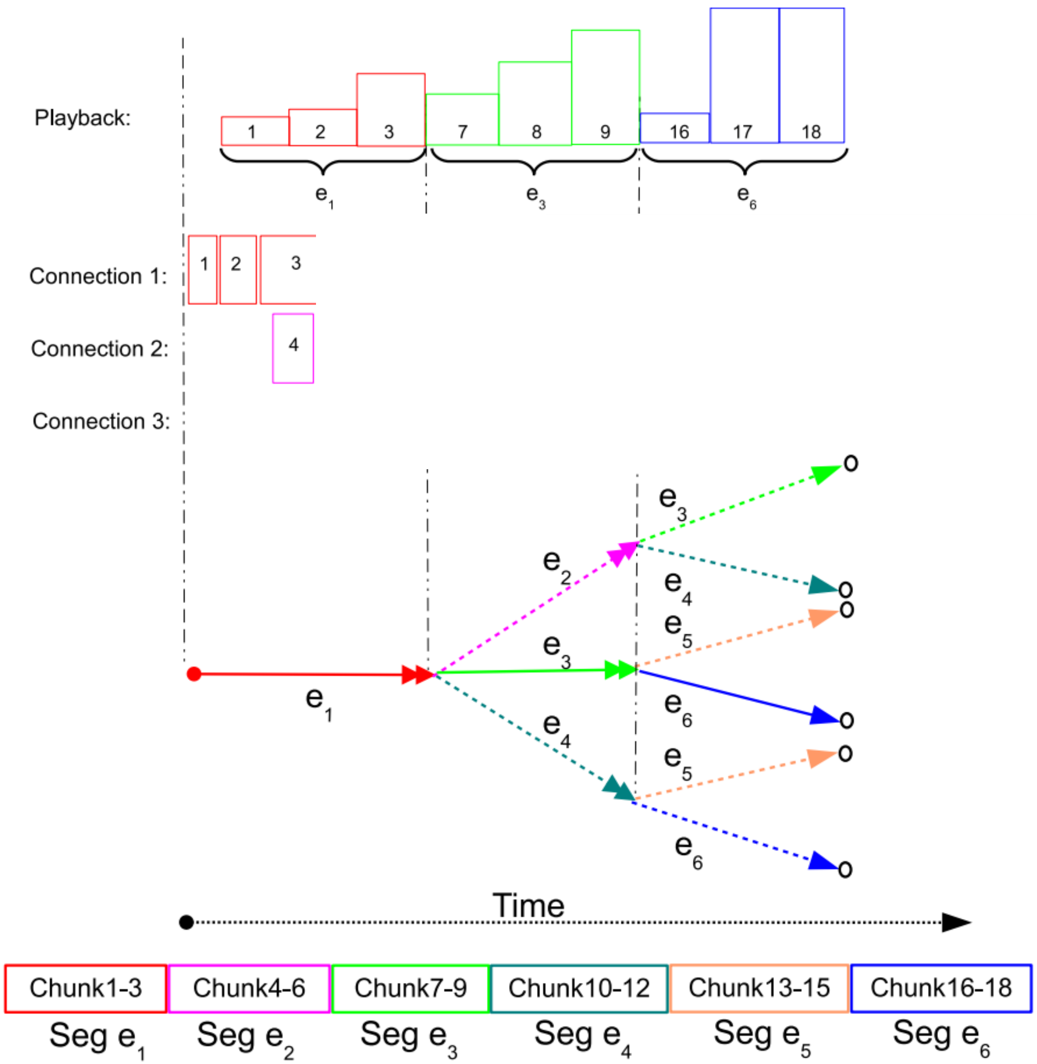
Concurrent Download Example



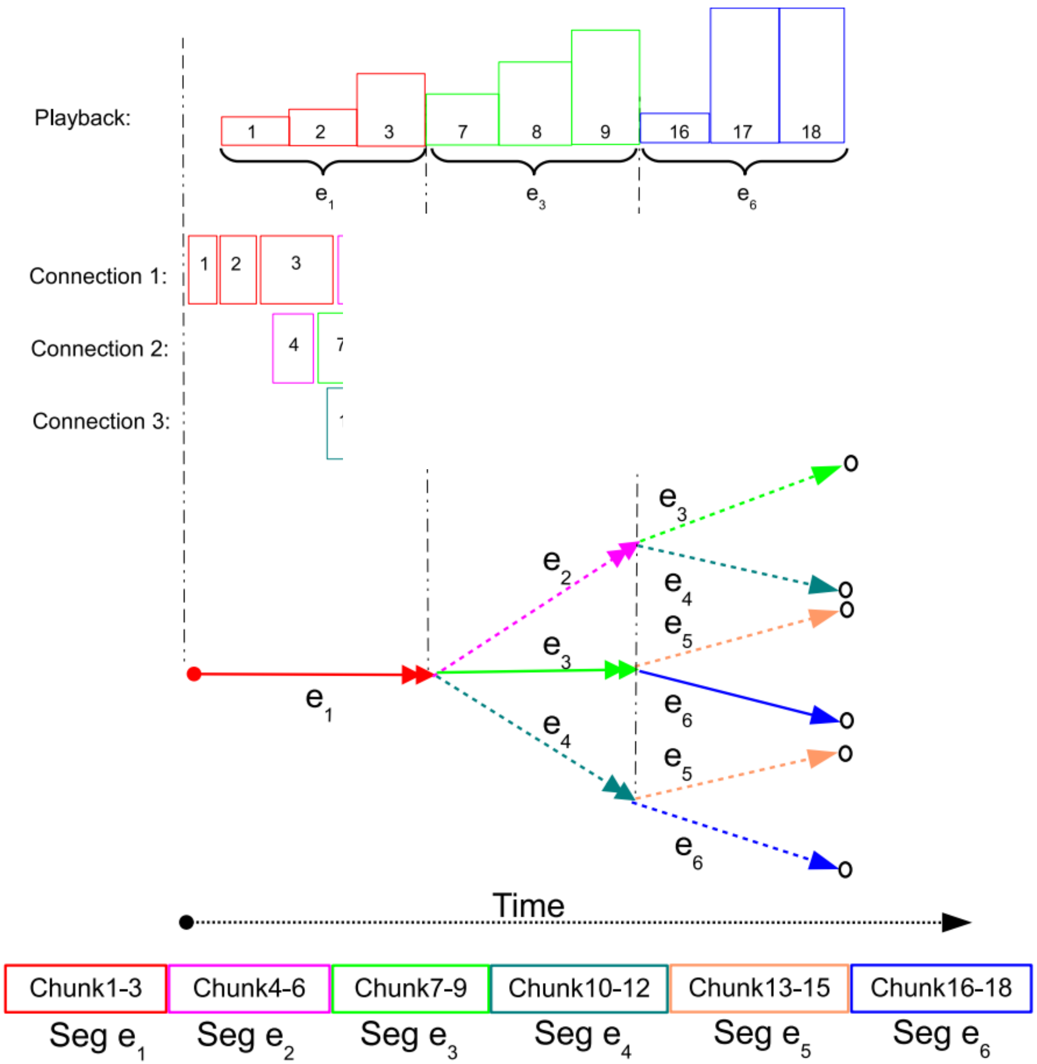
Concurrent Download Example



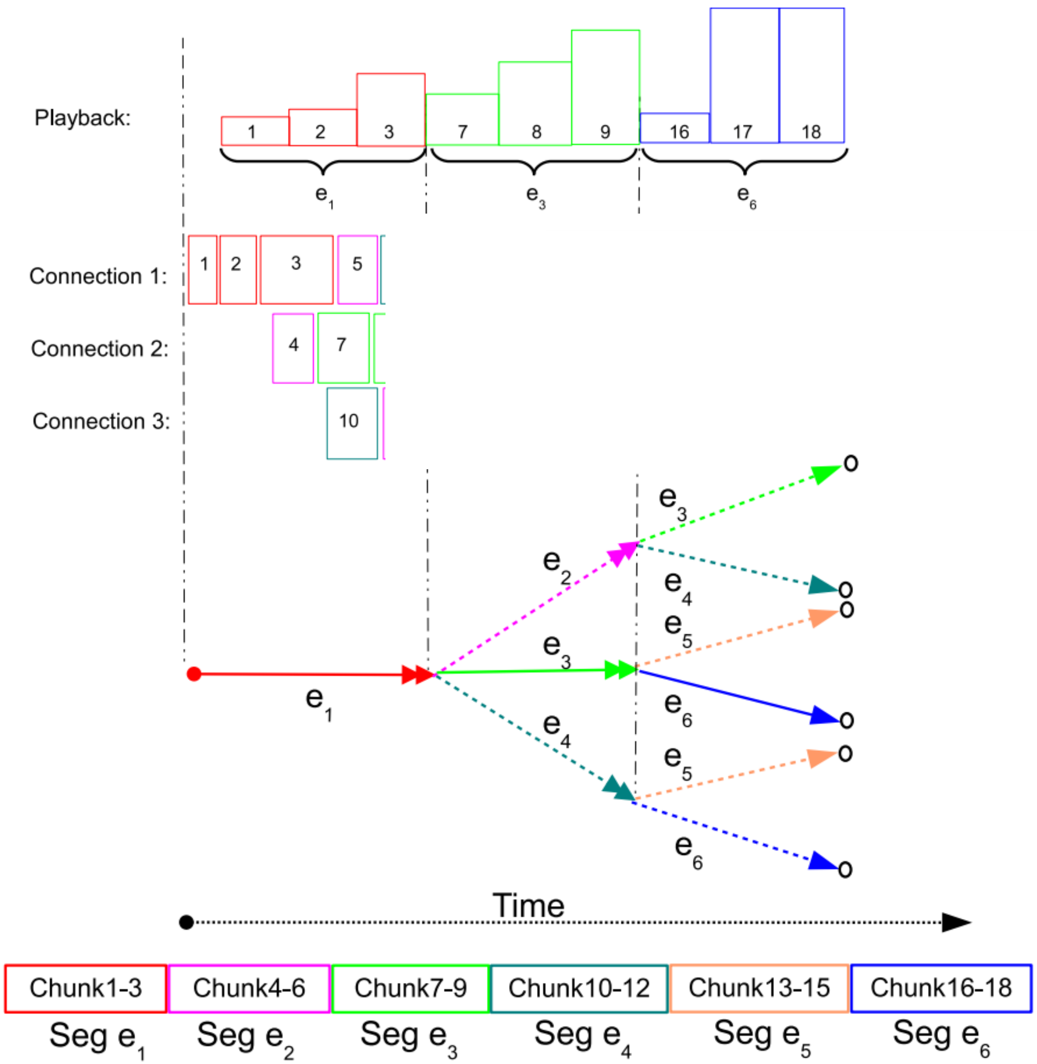
Concurrent Download Example



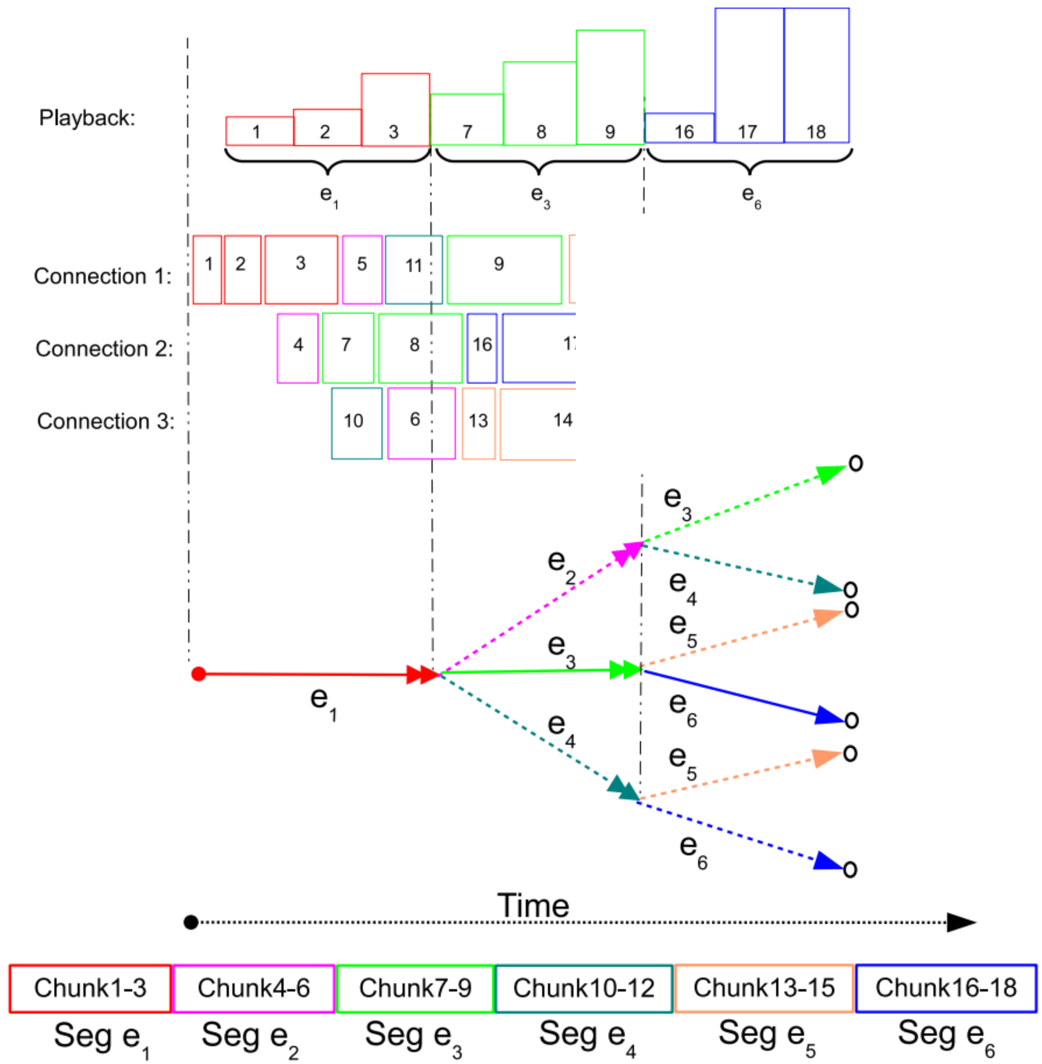
Concurrent Download Example



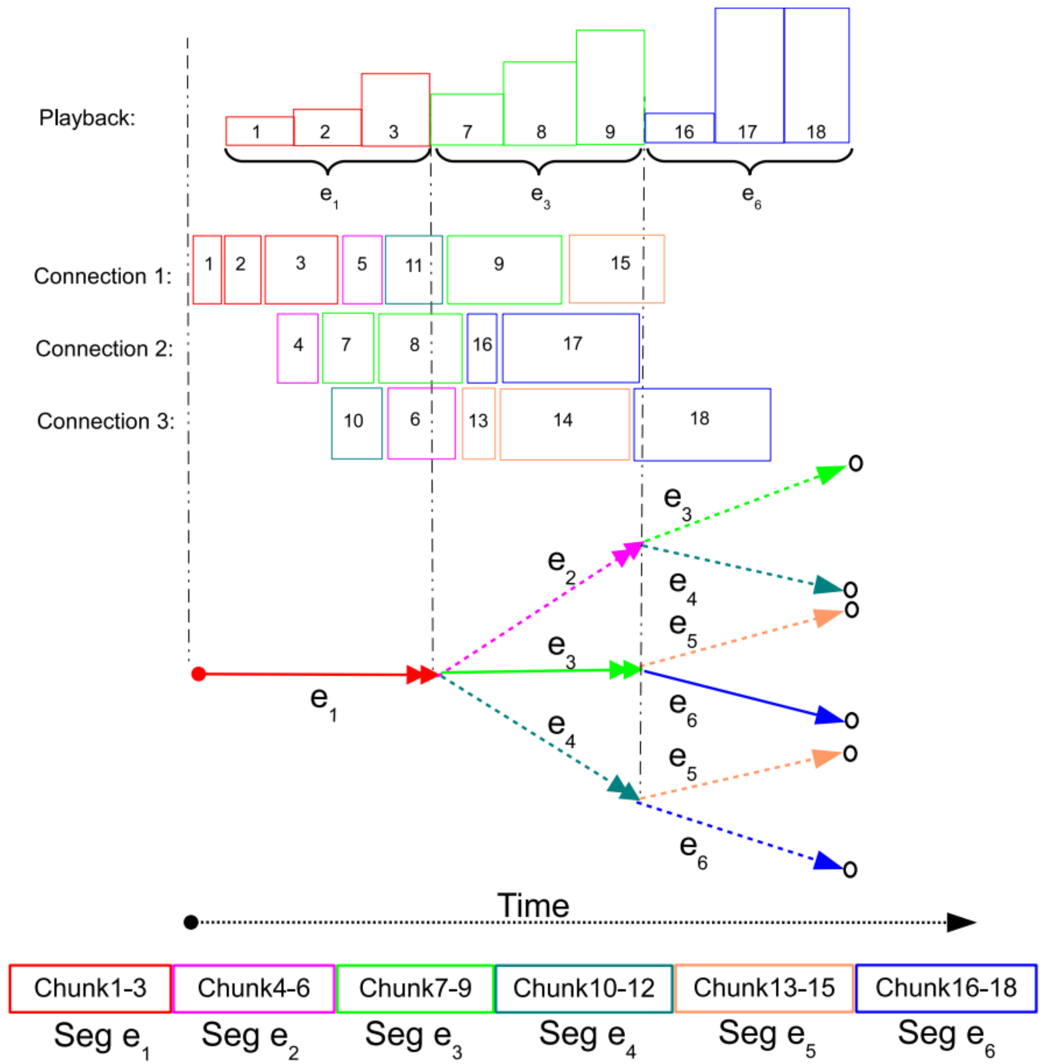
Concurrent Download Example



Concurrent Download Example



Concurrent Download Example



Summary part 1 ...

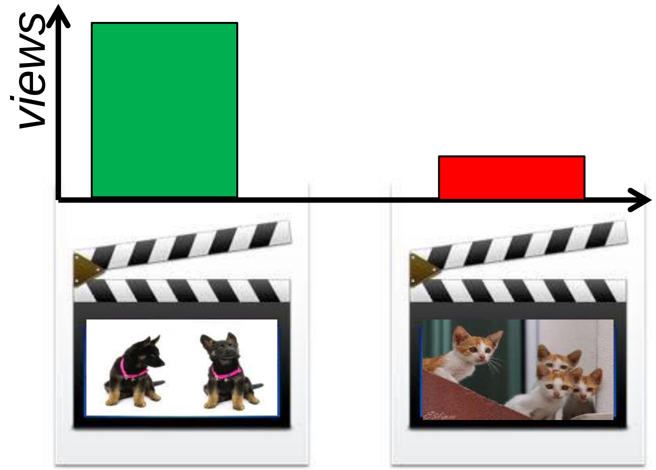
- Designed and implemented branched video player that achieve seamless streaming without playback interruptions

- Designed optimized policies that maximize playback quality while ensuring sufficient workahead to avoid stalls

- Evaluation shows that solution effectively adapt quality levels and number of parallel connections so as to provide best possible video quality, given current conditions

---

- Note: Extensions, generalizations, and variations include “multi-file prefetching for impatient users” [*Proc. ACM Multimedia 2015*]



The Untold Story of the Clones: Content-agnostic Factors that Impact YouTube Video Popularity

Proc. ACM SIGKDD 2012.

Characterizing and Modeling Popularity of User-generated Videos

Proc. IFIP PERFORMANCE 2011.

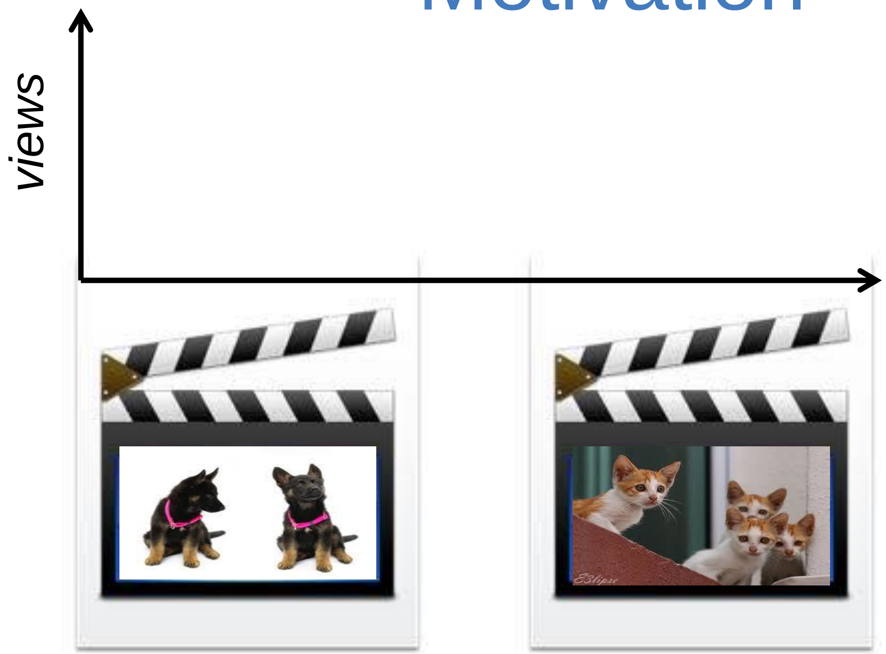
Motivation

- Streaming services responsible for majority of traffic
- Video dissemination (e.g., YouTube) can have wide-spread impacts on opinions, thoughts, and cultures

Motivation

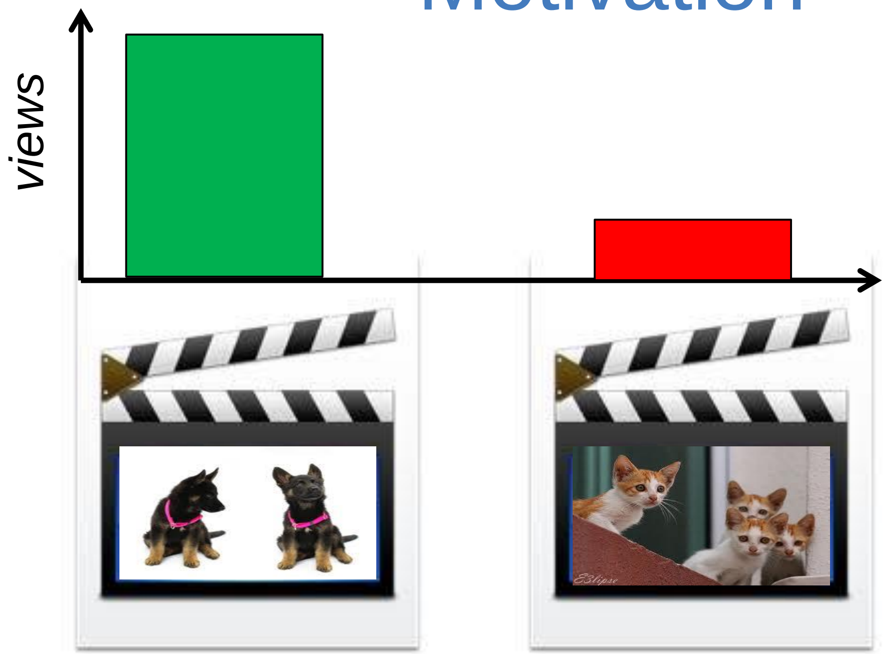
- Not all videos will reach the same popularity and have the same impact

Motivation



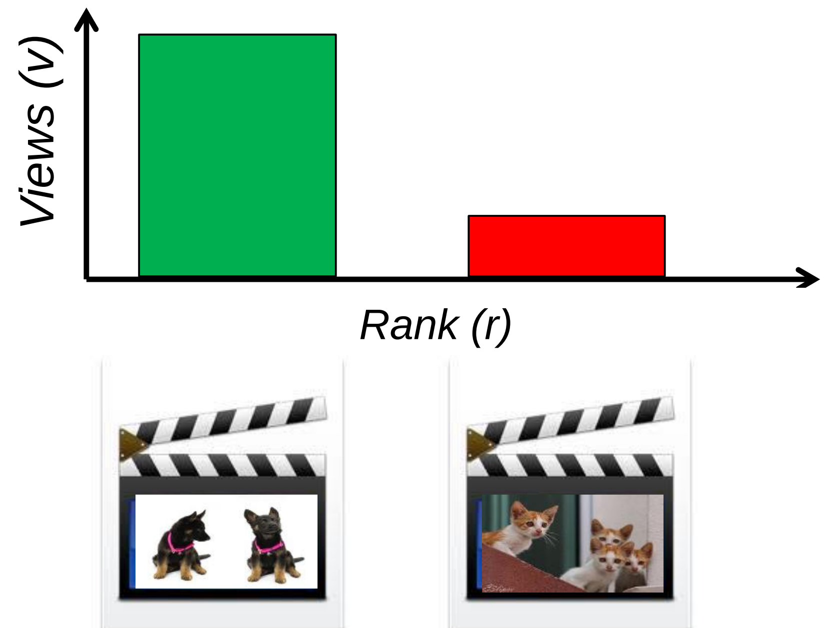
- Not all videos will reach the same popularity and have the same impact

Motivation

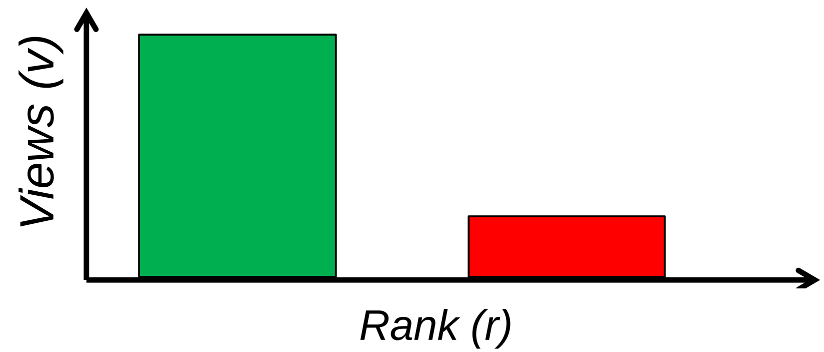


- Not all videos will reach the same popularity and have the same impact

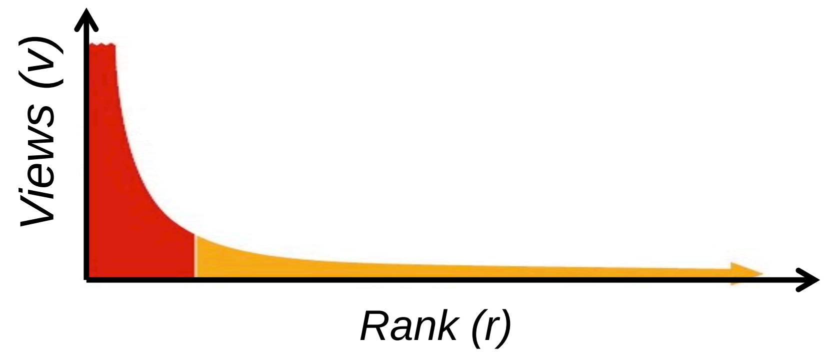
Aside ...
Popularity distribution



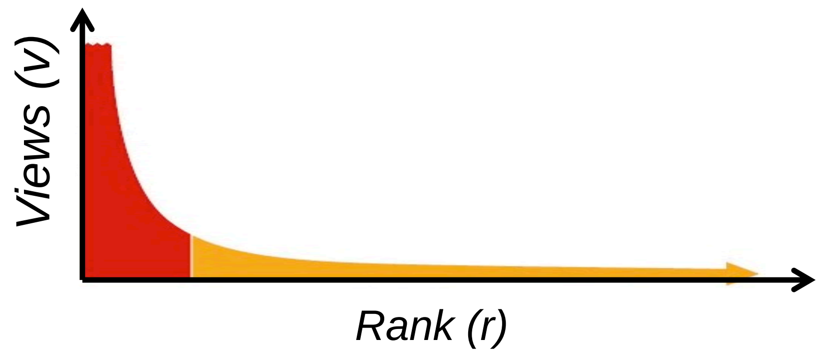
Aside ...
Popularity distribution



Aside ...
Popularity distribution

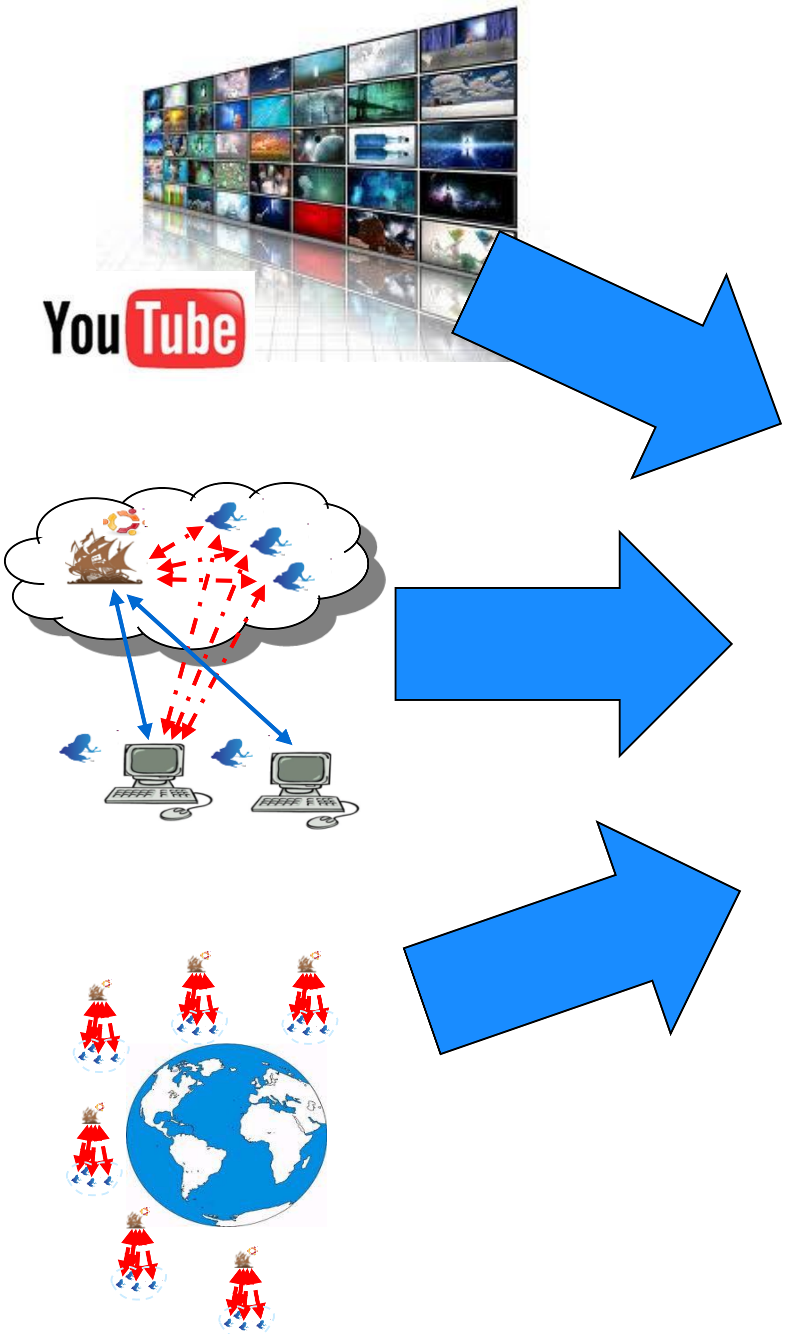


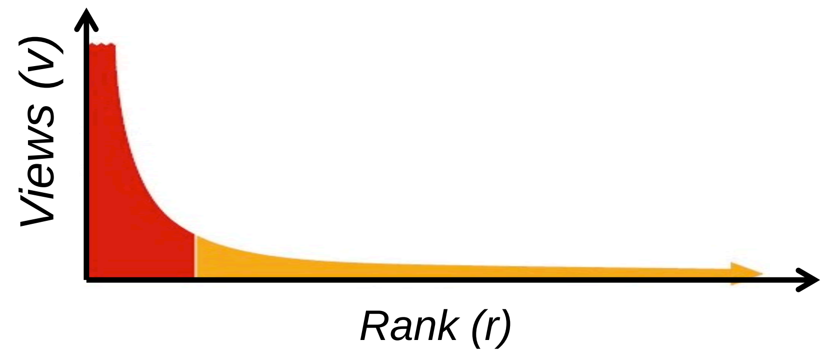
Aside ...
Popularity distribution



IFIP Performance ‘11, IPTPS ‘10

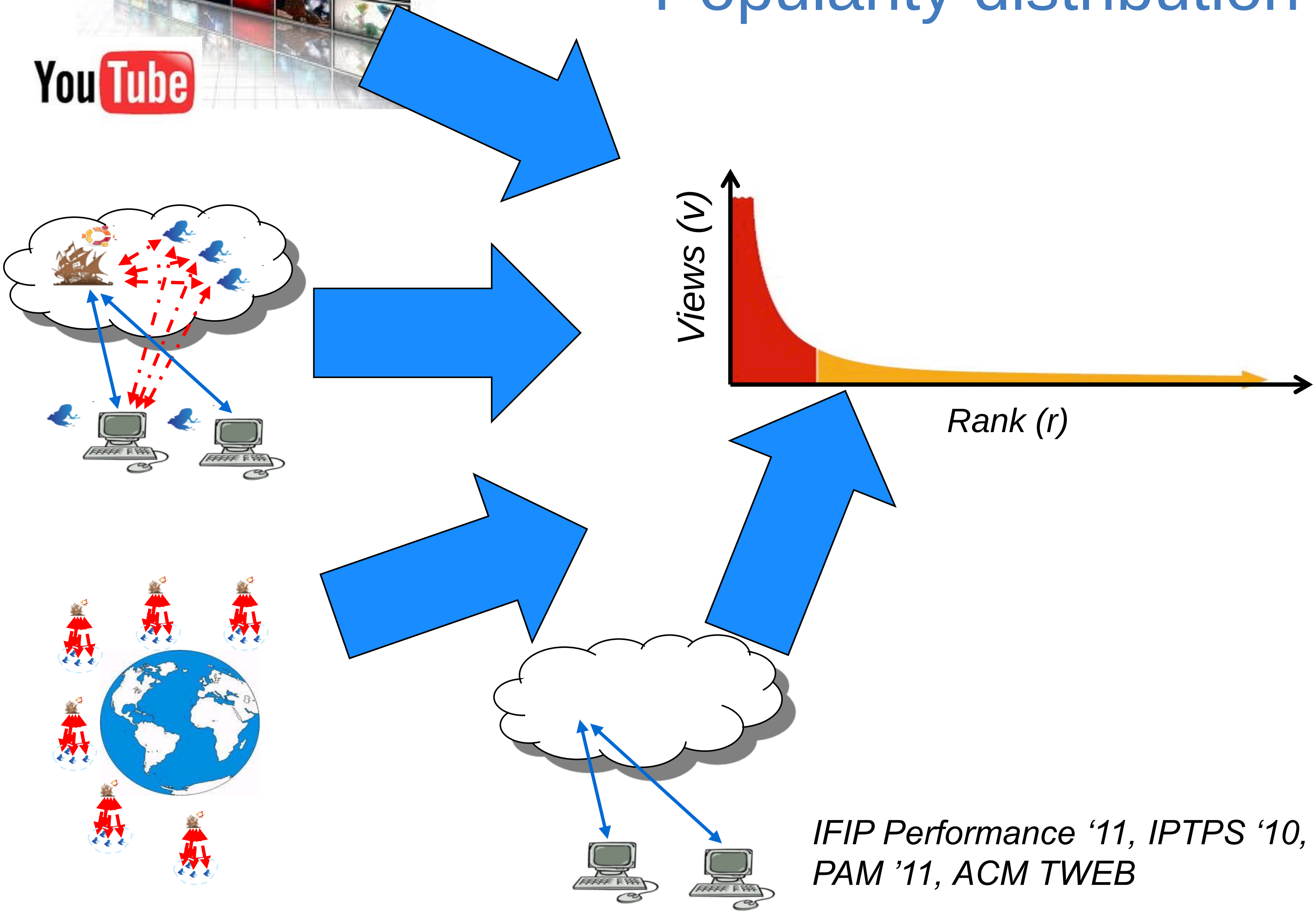
Aside ...
Popularity distribution





IFIP Performance ‘11, IPTPS ‘10, PAM ’11

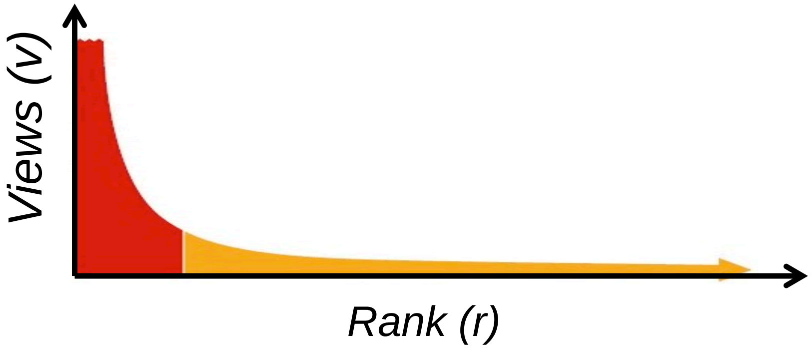
Aside ...
Popularity distribution



IFIP Performance ‘11, IPTPS ‘10, PAM ’11, ACM TWEB

Zipf popularity...

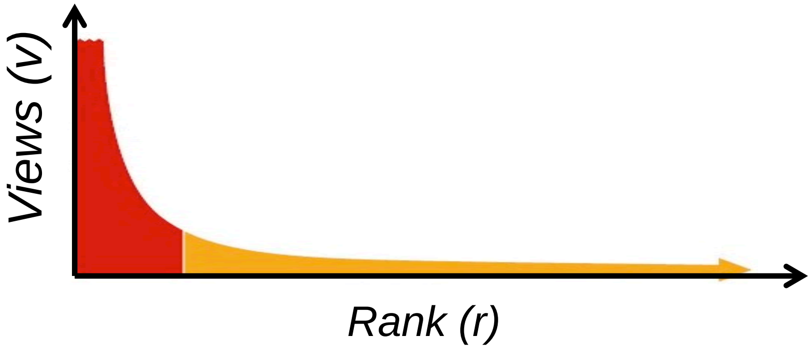
... and long tails

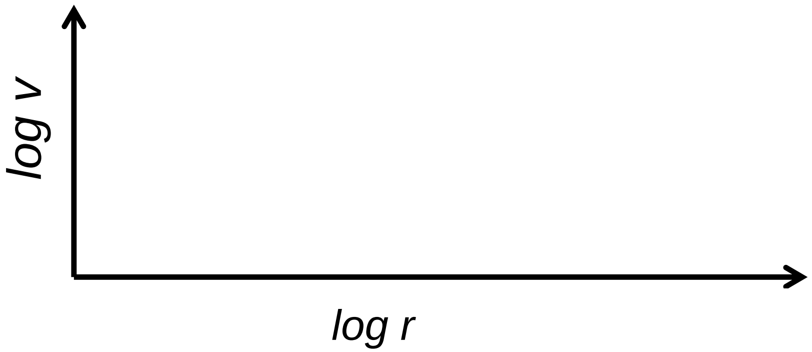


$$v_r \propto r^{-\alpha}$$

Zipf popularity...

... and long tails

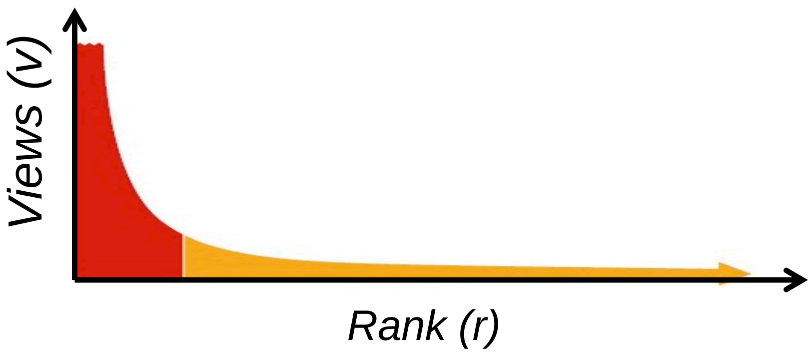


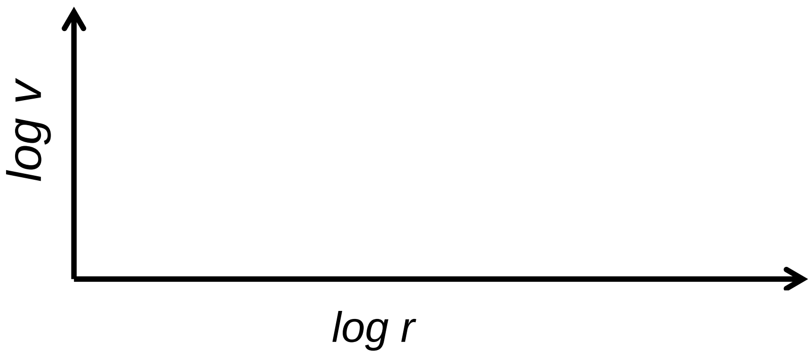


$$v_r \propto r^{-\alpha}$$

Zipf popularity...

... and long tails



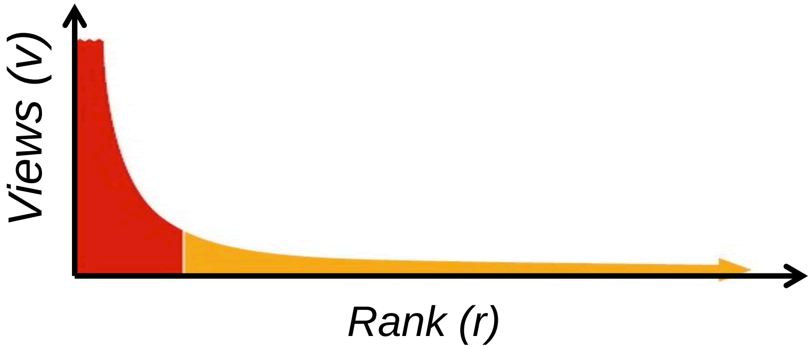


$$v_r \propto r^{-\alpha}$$

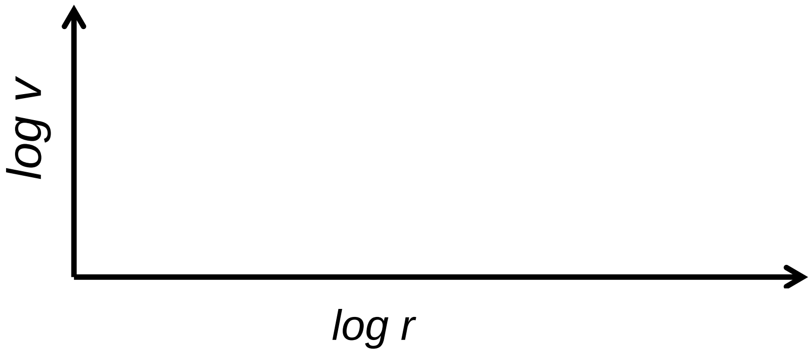
$$\log v_r = \log v_1 - \alpha \log r$$

Zipf popularity...

... and long tails



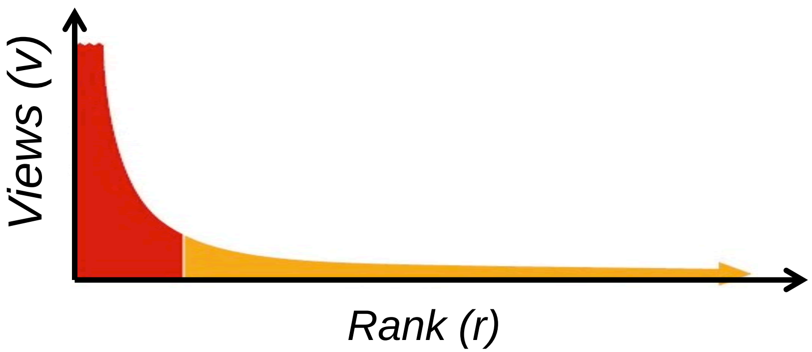
$$v_r \propto r^{-\alpha}$$

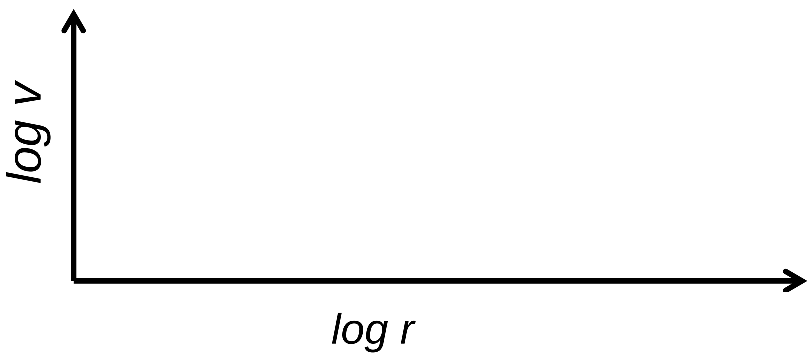


$$\log v_r = \log v_1 - \alpha \log r$$

Zipf popularity...

... and long tails





$$v_r \propto r^{-\alpha}$$

$$\log v_r = \log v_1 - \alpha \log r$$

$y(x)$ = x_0 -- α x

Zipf popularity...

... and long tails

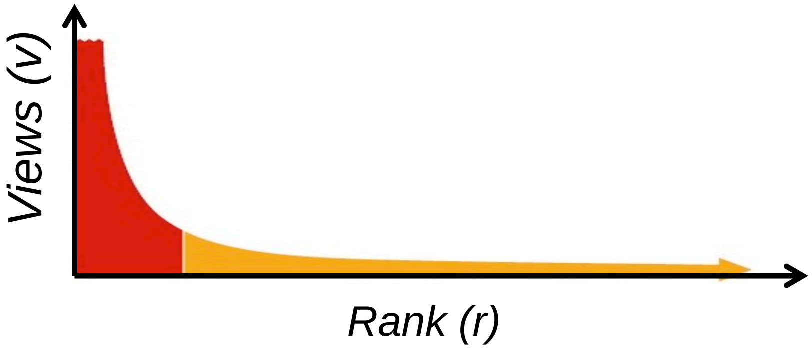
$$v_r \propto r^{-\alpha}$$

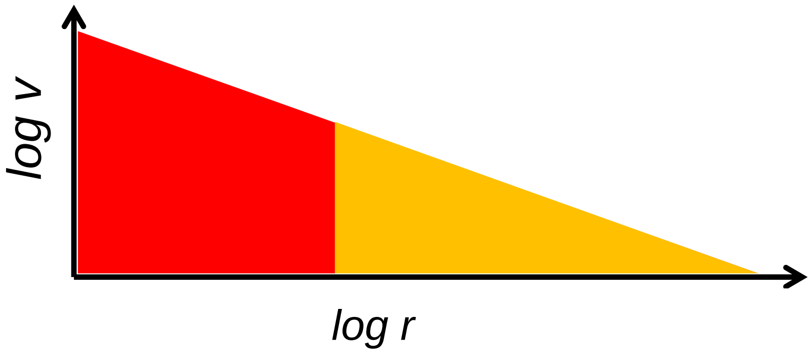
$$\log v_r = \log v_1 - \alpha \log r$$

$y(x)$ = x_0 -- α x

Zipf popularity...

... and long tails

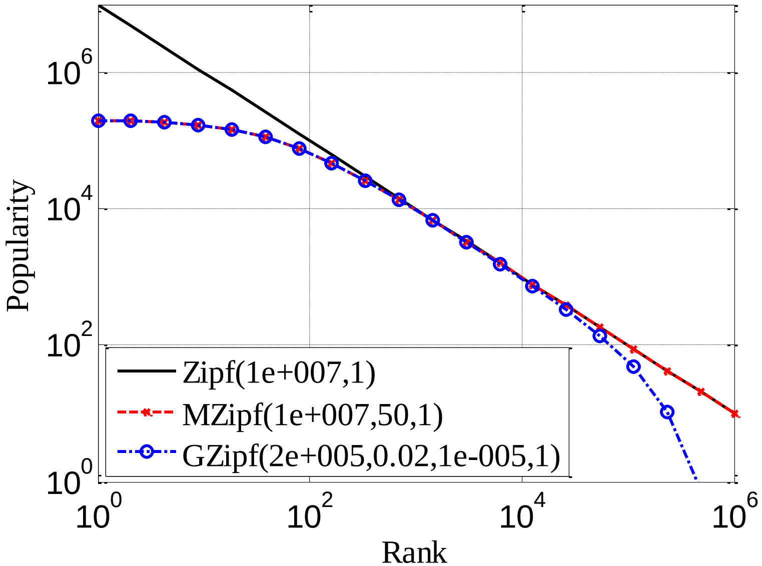


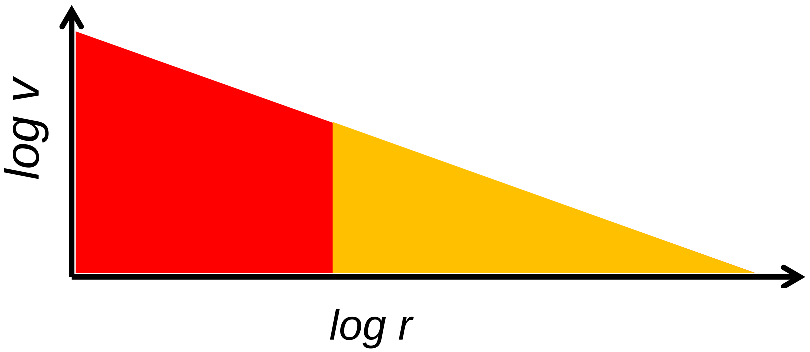


$$v_r \propto r^{-\alpha}$$

$$\log v_r = \log v_1 - \alpha \log r$$

Zipf popularity...
... and long tails

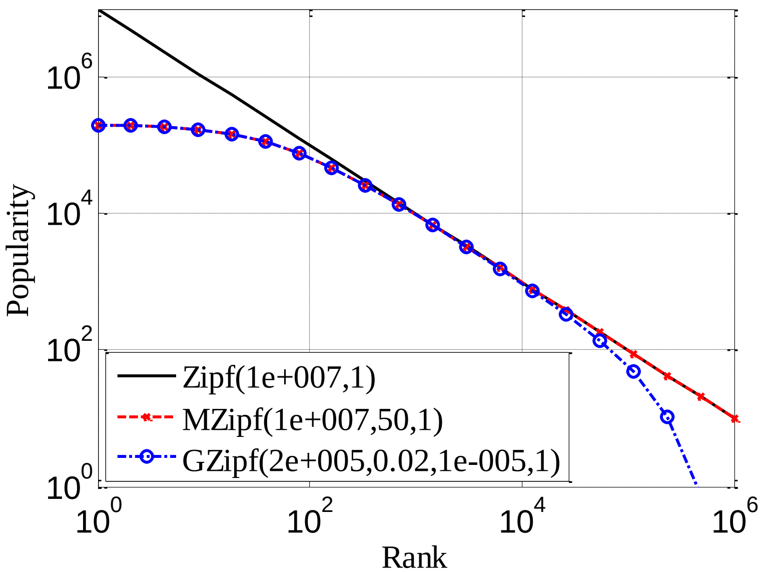




$$\log v_r = \log v_1 - \alpha \log r$$

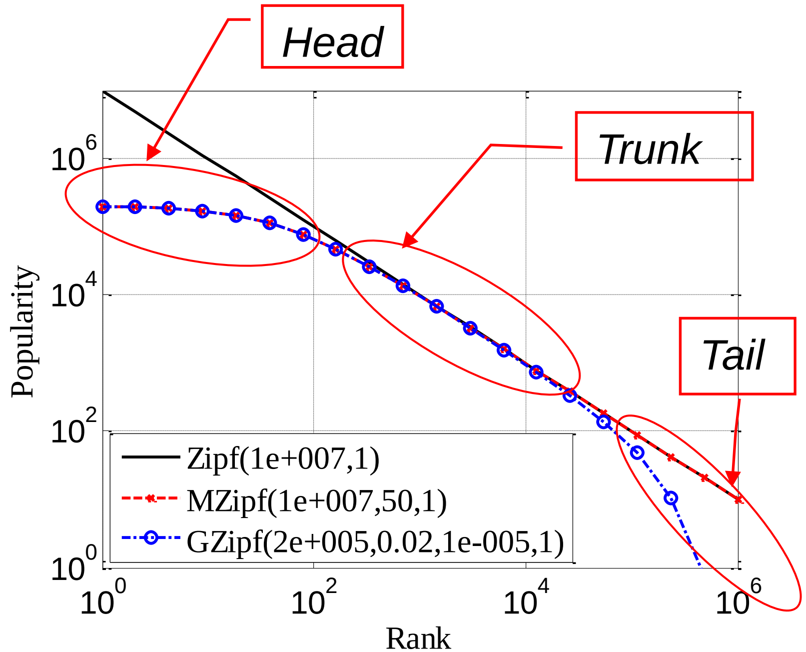
Zipf popularity...

... and long tails



E.g., ACM TWEB, PAM ‘11
IFIP Performance ‘11, IPTPS ‘10

Zipf popularity...
... and long tails

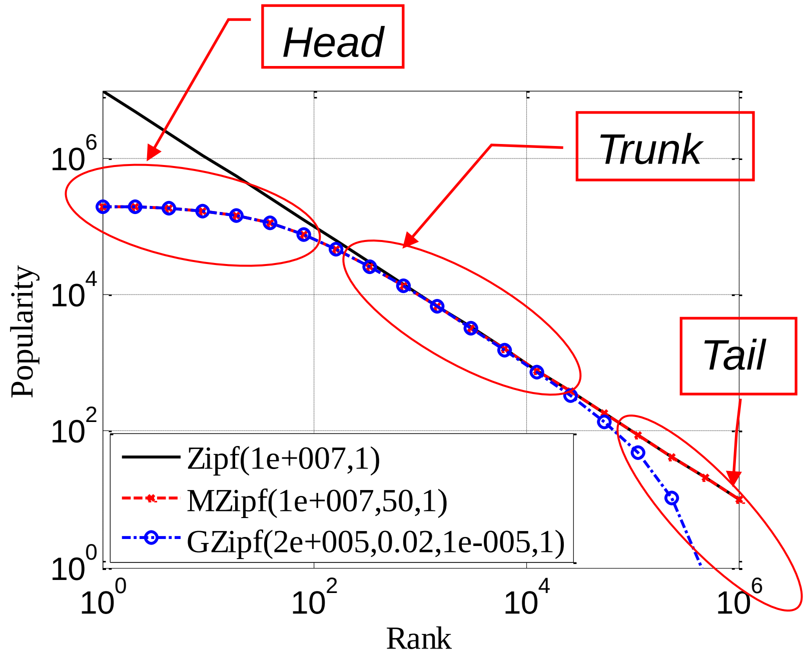


- *Popularity distribution statistics and models*
  - *Across services (impact on system design)*
  - *Lifetime vs current*
  - *Over different time period (churn)*
  - *Different sampling methods*

E.g., ACM TWEB, PAM '11, IFIP Performance '11, IPTPS '10

Zipf popularity...

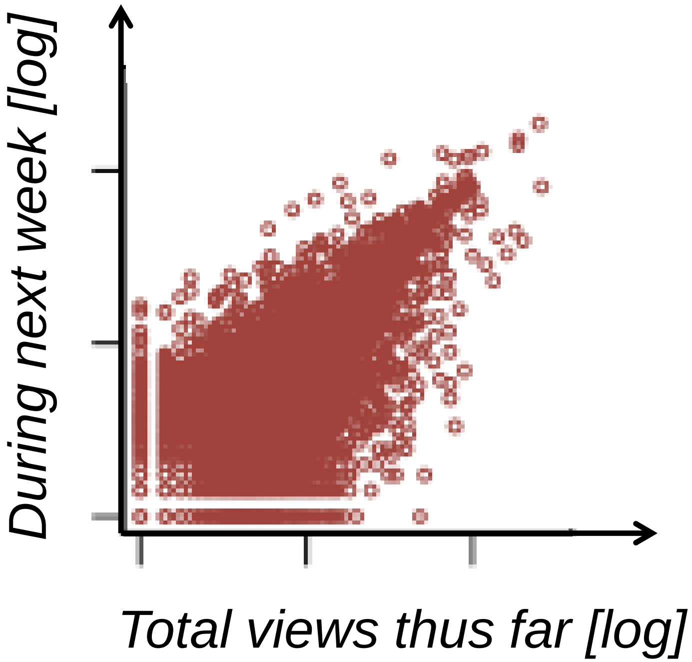
... and long tails



- *Popularity distribution statistics and models*
  - *Across services (impact on system design)*
  - ***Lifetime vs current***
  - ***Over different time period (churn)***
  - *Different sampling methods*

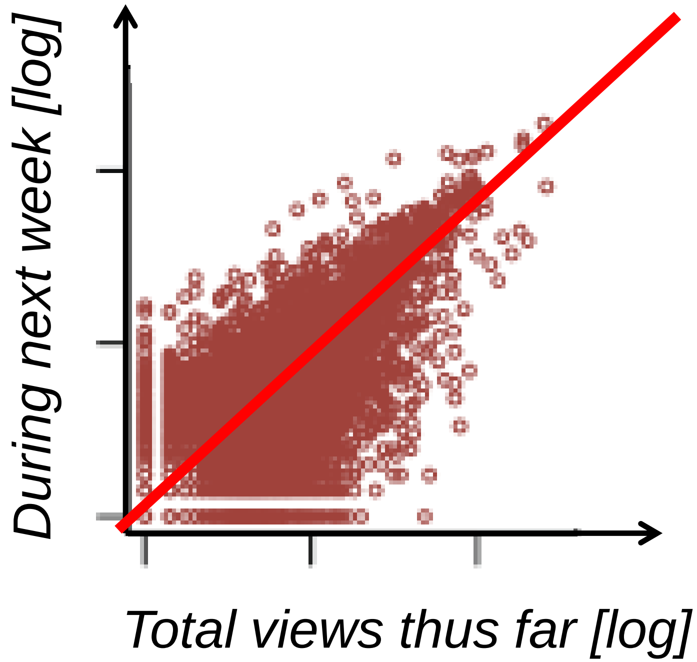
E.g., ACM TWEB, PAM '11, IFIP Performance '11, IPTPS '10

Rich-gets-richer ...
... and churn



E.g., IFIP Performance '11

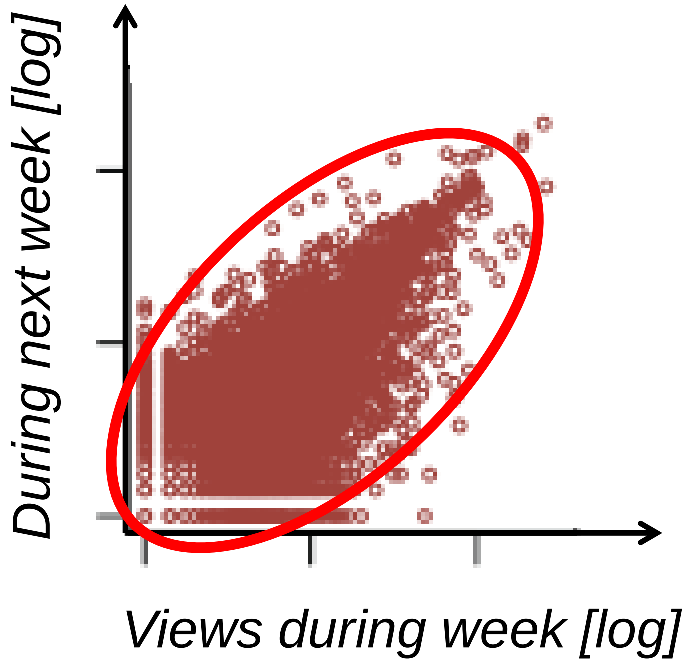
Rich-gets-richer ...
... and churn



- *The more views a video has, the more views it is likely to get in the future*

E.g., IFIP Performance '11

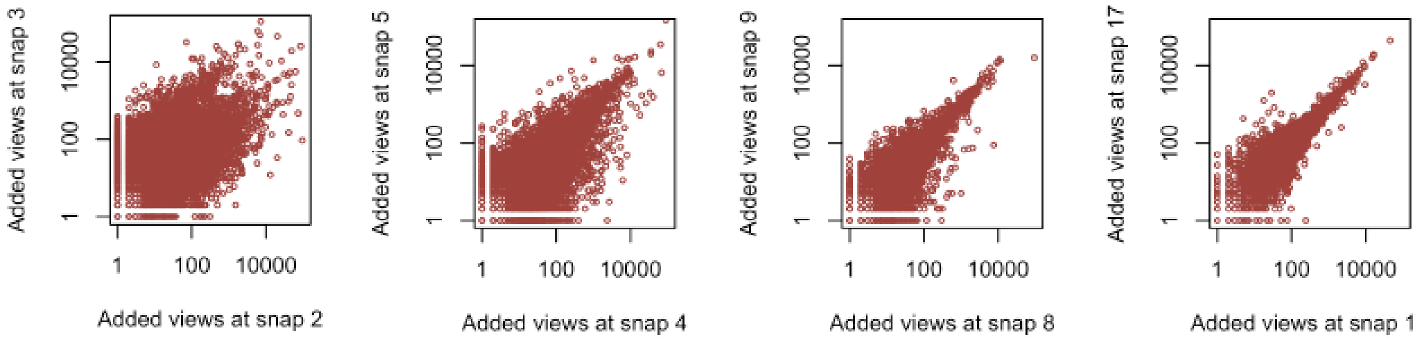
Rich-gets-richer ...
... and churn



- *The more views a video has, the more views it is likely to get in the future*
- *The relative popularity of the individual videos are highly non-stationary*

E.g., IFIP Performance '11

Rich-gets-richer ...
... and churn



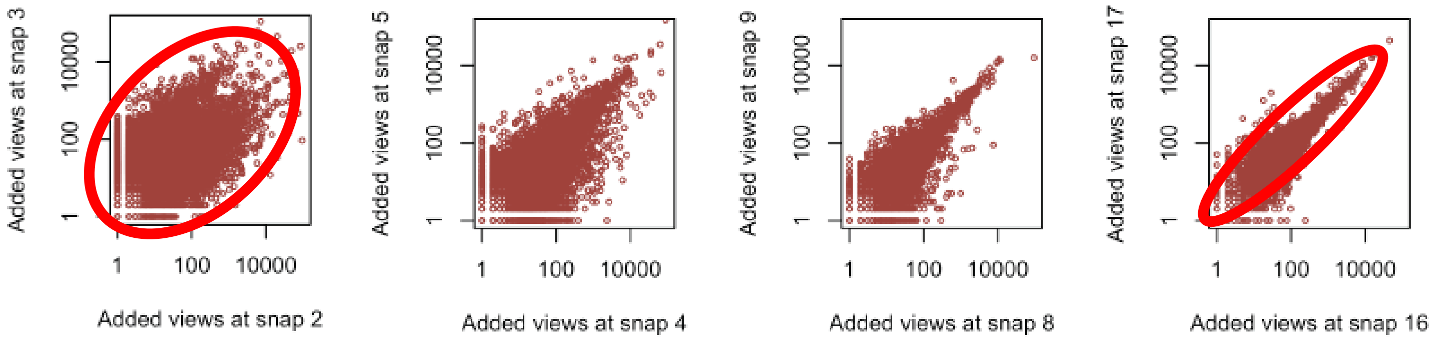
Week 2 *Week4* *Week 8* *Week 16*

Young videos → *Old videos*

- *The more views a video has, the more views it is likely to get in the future*
- *The relative popularity of the individual videos are highly non-stationary*

E.g., IFIP Performance '11

Rich-gets-richer ...
... and churn



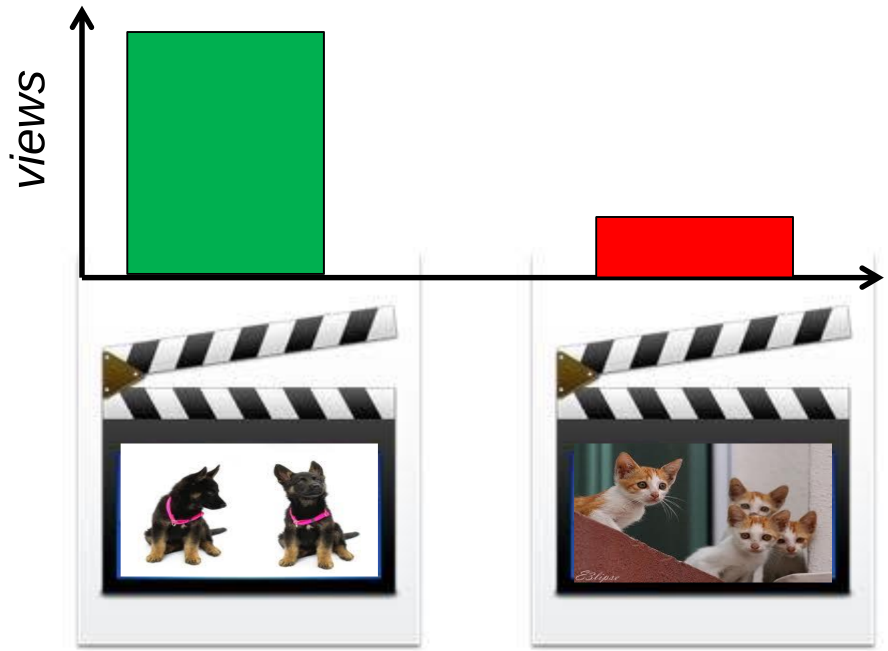
Week 2 *Week4* *Week 8* *Week 16*

Young videos → *Old videos*

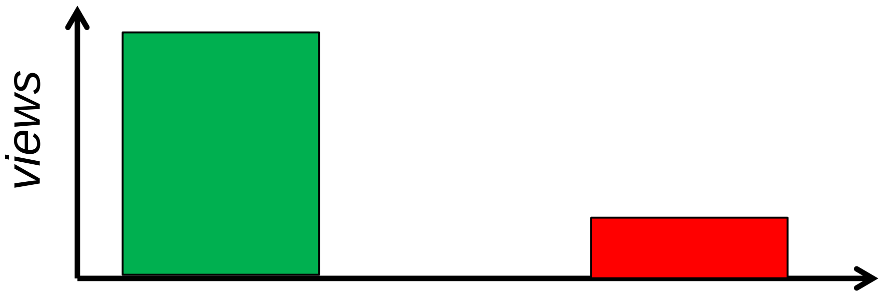
- *The more views a video has, the more views it is likely to get in the future*
- *The relative popularity of the individual videos are highly non-stationary*
- *Some long-term popularity*

E.g., IFIP Performance '11

Motivation



Motivation



- Some popularity differences due to content differences

Motivation

- Some popularity differences due to content differences

Motivation

- Some popularity differences due to content differences
- But also because of other “content-agnostic” factors
  - The latter factors are of considerable interest but it has been difficult to accurately study them

Motivation

- Some popularity differences due to content differences
- But also because of other “content-agnostic” factors
  - The latter factors are of considerable interest but it has been difficult to accurately study them

In general, existing works ***do not*** *take content differences into account .. .(e.g., large number of rich-gets-richer studies)*

Methodology

- Develop and apply a methodology that is able to accurately assess, both qualitatively and quantitatively, the impacts of various content-agnostic factors on video popularity

Methodology

- Develop and apply a methodology that is able to accurately assess, both qualitatively and quantitatively, the impacts of various content-agnostic factors on video popularity

Methodology

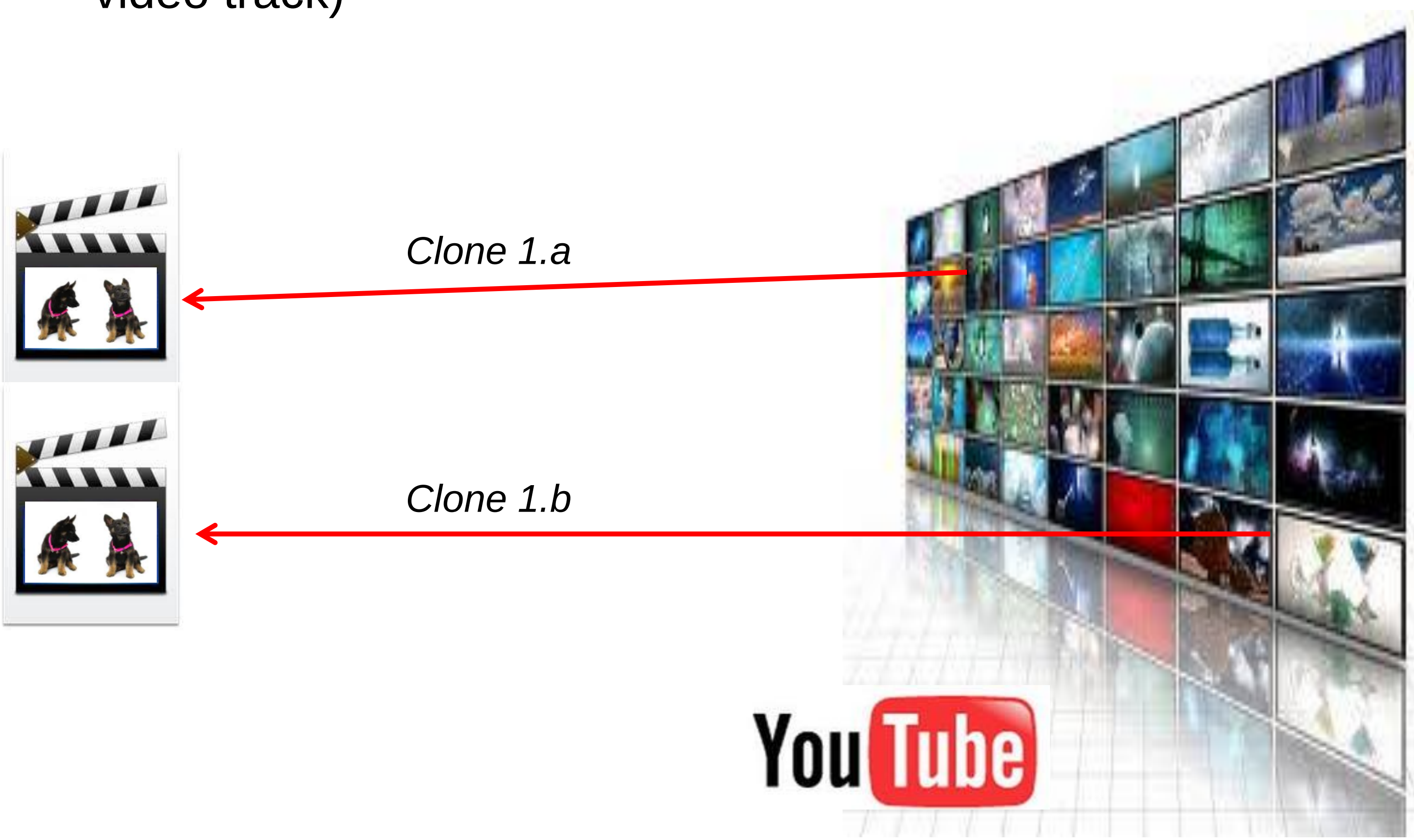
- Clones
  - Videos that have “identical” content (e.g., same audio and video track)

Methodology

- Clones
  - Videos that have “identical” content (e.g., same audio and video track)

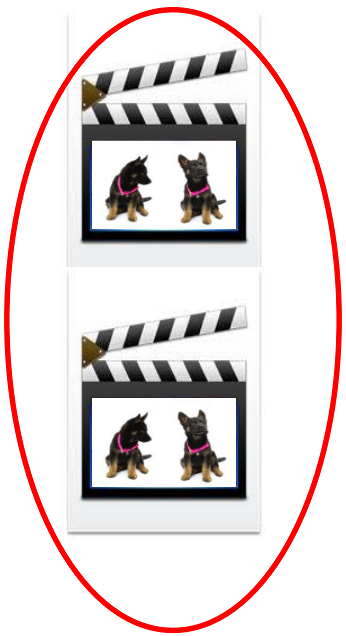
Methodology

- Clones
  - Videos that have “identical” content (e.g., same audio and video track)



Methodology

- Clones
  - Videos that have “identical” content
- Clone set
  - Set of videos that have “identical” content

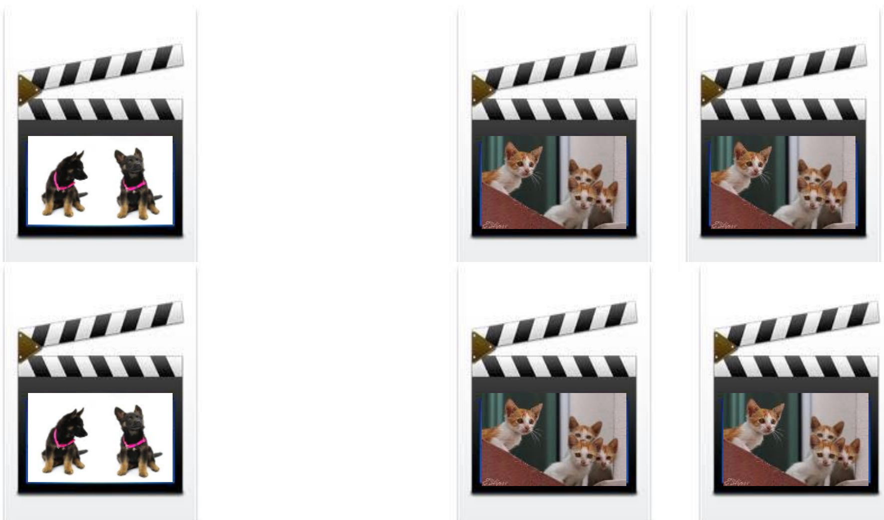


Clone set 1

YouTube

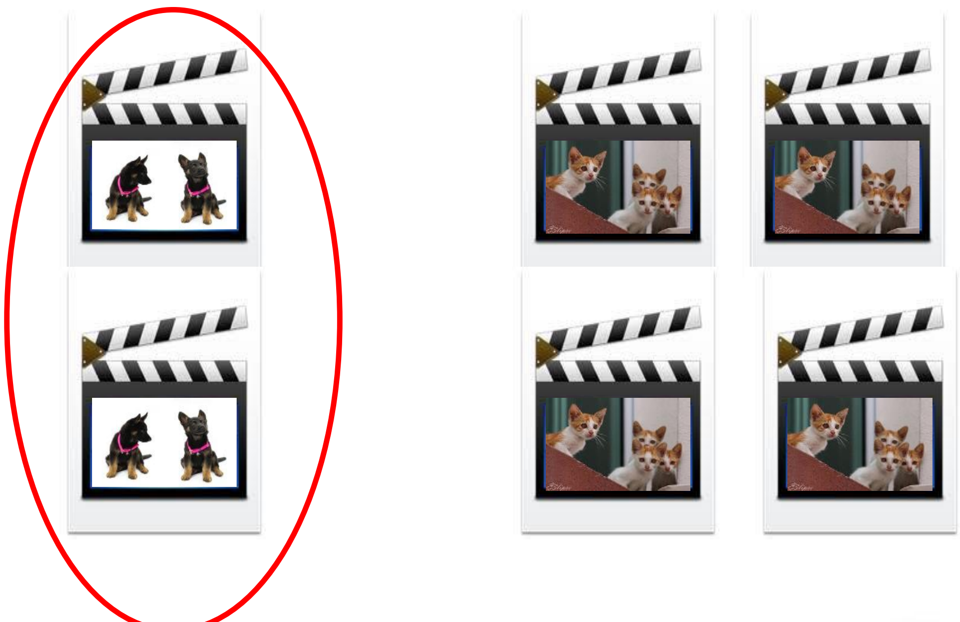
Methodology

- Clones
  - Videos that have “identical” content
- Clone set
  - Set of videos that have “identical” content



Methodology

- Clones
  - Videos that have “identical” content
- Clone set
  - Set of videos that have “identical” content



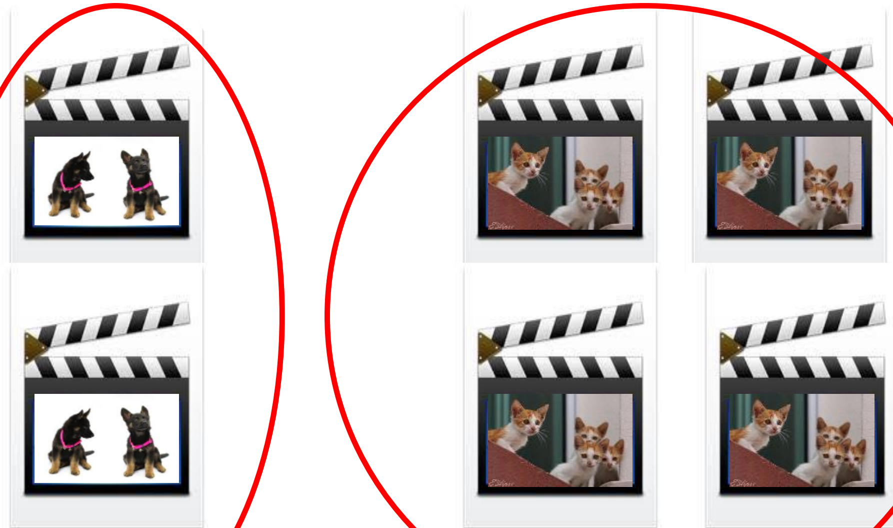
Methodology

- Clones
  - Videos that have “identical” content
- Clone set
  - Set of videos that have “identical” content

Methodology

- Clones
  - Videos that have “i… …ontent
- Clone set
  - Se… …os that have “identical” content

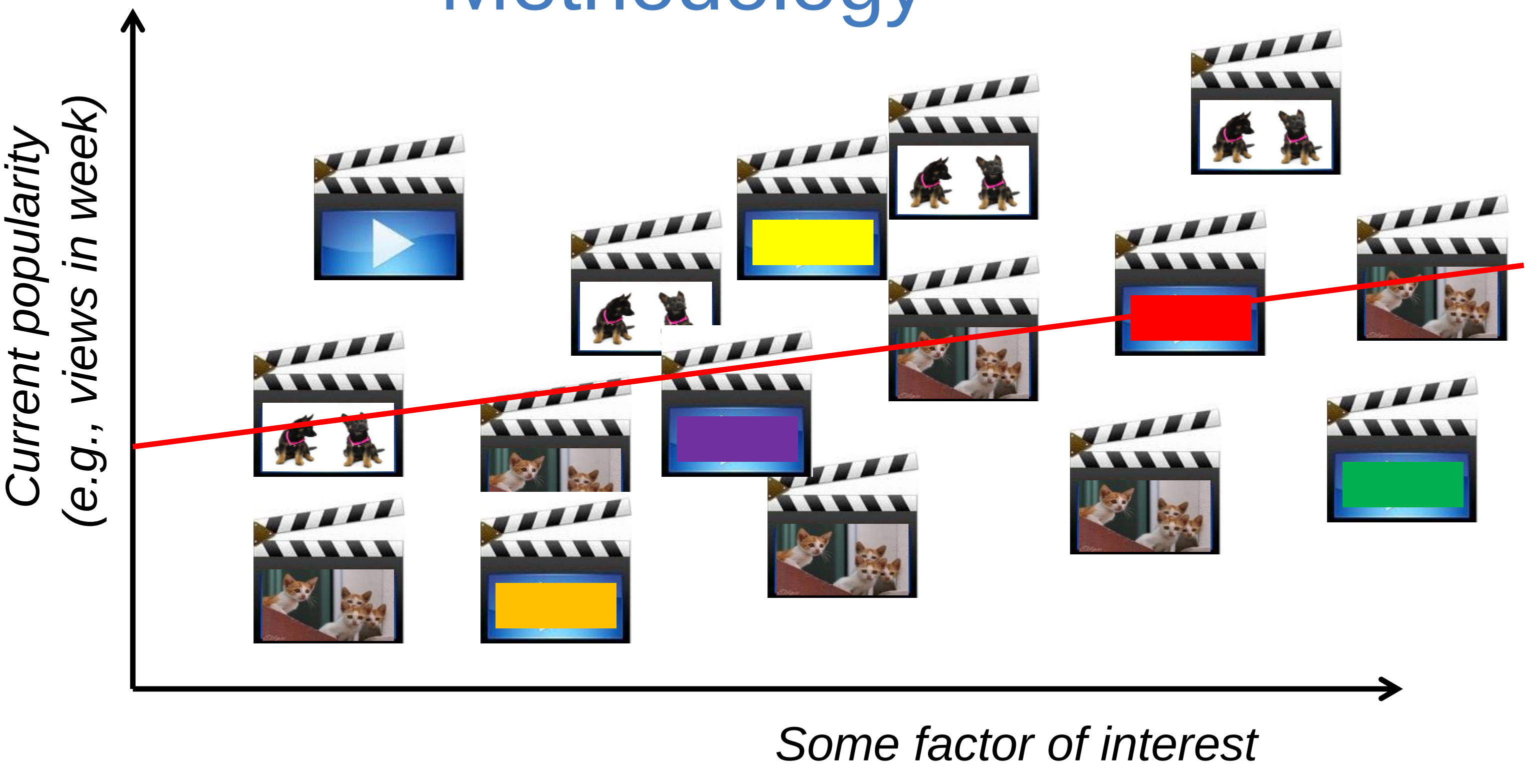
Clone sets allow us to control for content



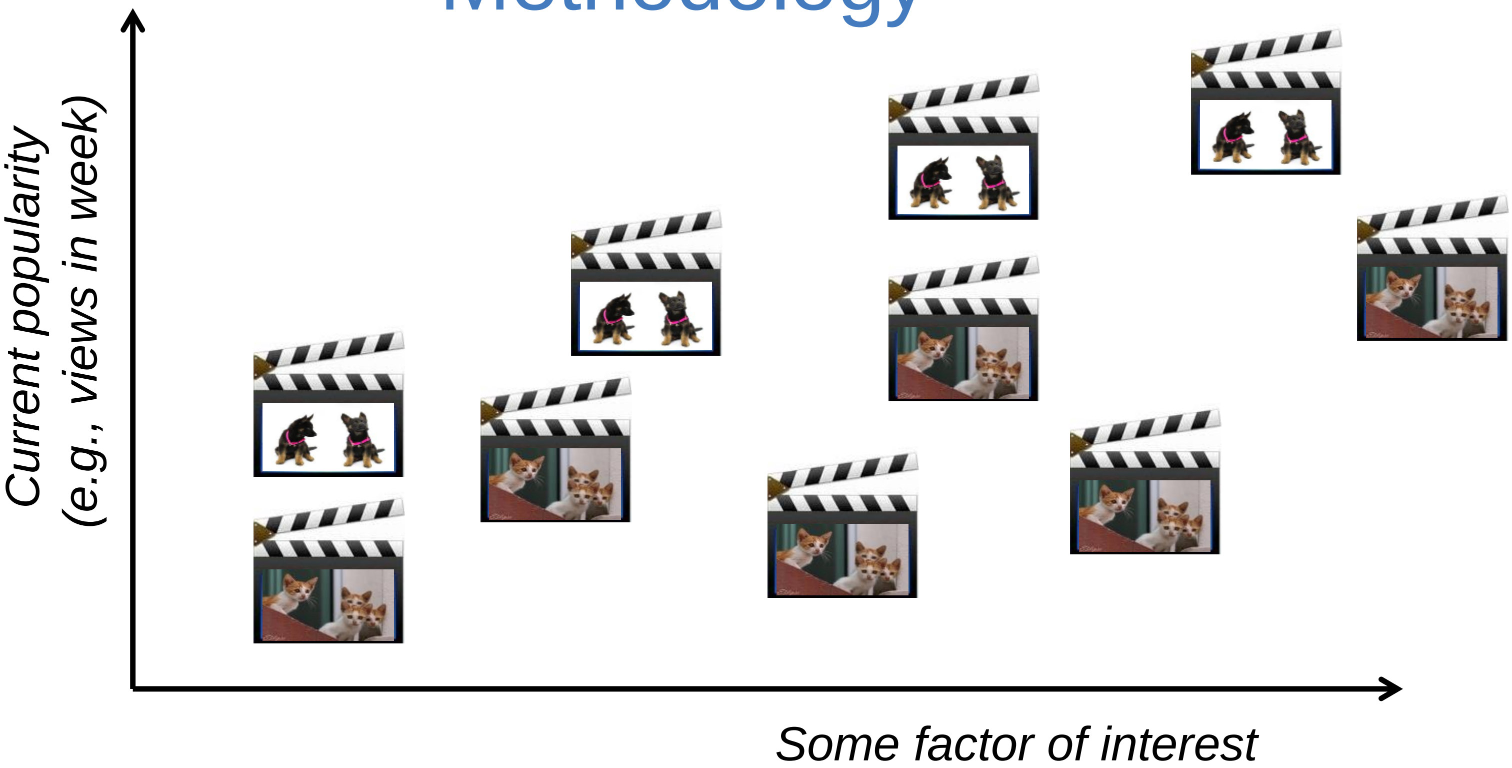
Methodology

Clone sets allow us to control for content

Methodology

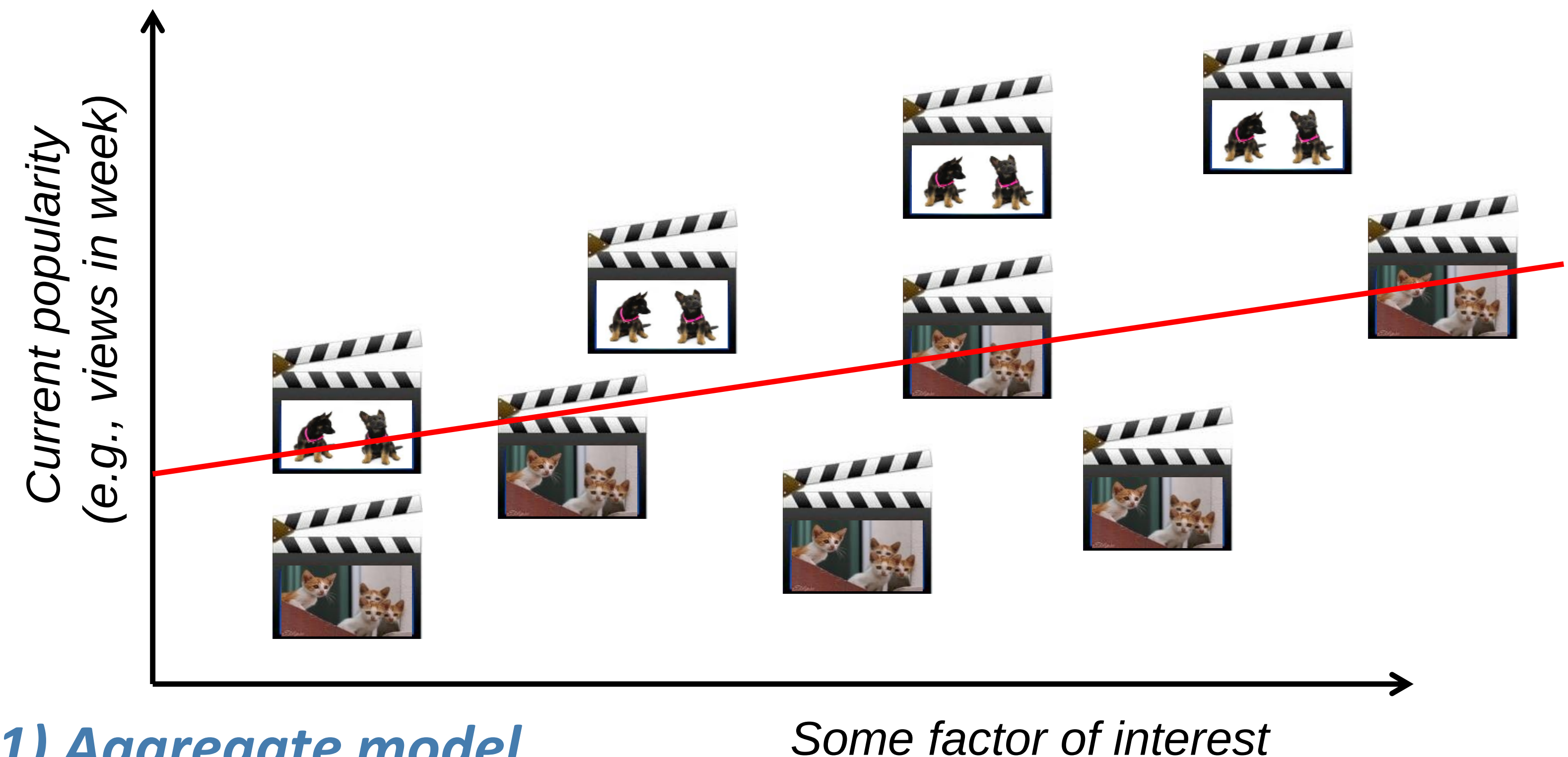


Methodology



- Focus on clone sets

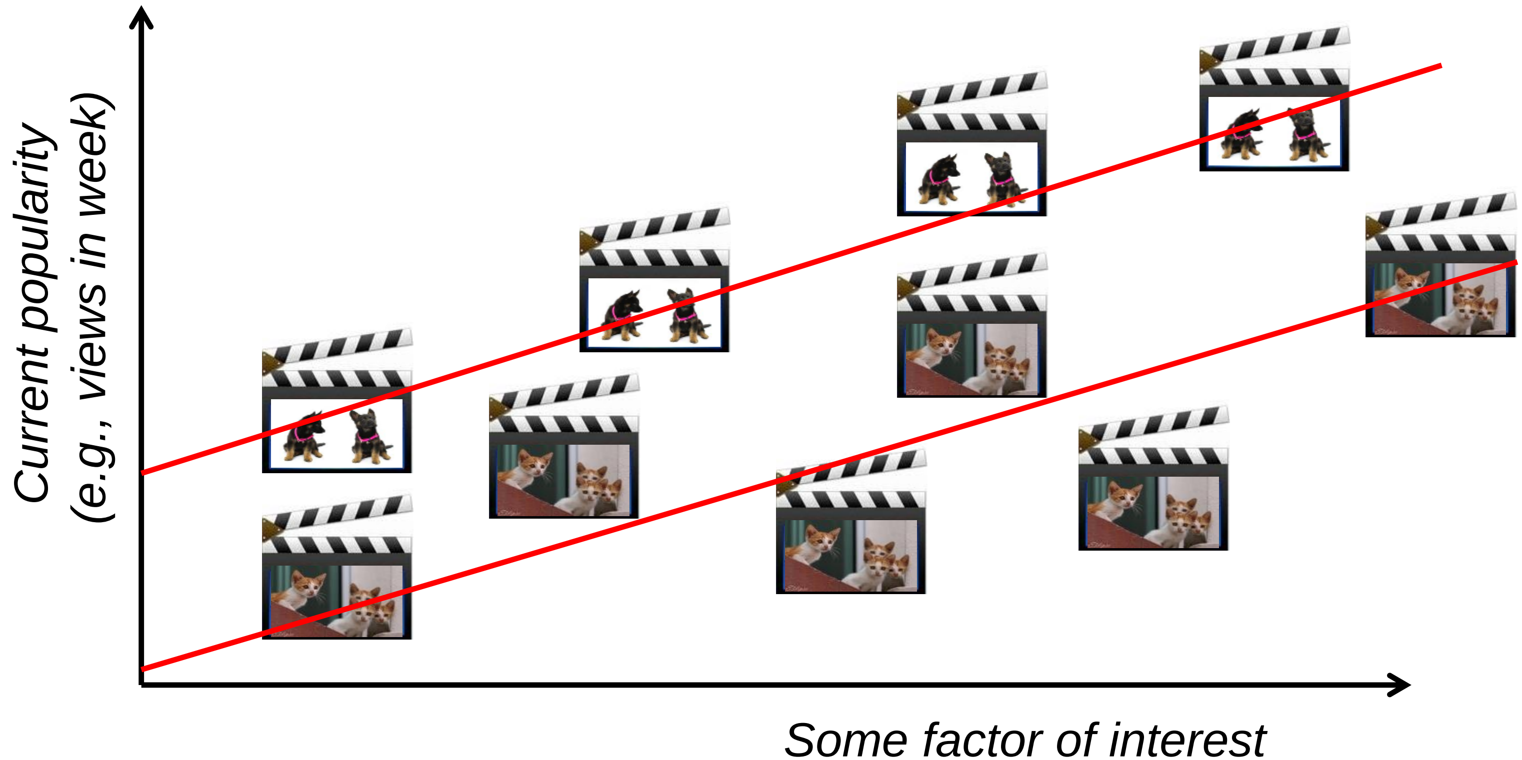
Methodology: Aggregate model



(1) Aggregate model

$$Y_i = \underbrace{\beta_0 + \sum_{p=1}^{P} \beta_p X_{i,p} + \qquad\qquad}_{\text{Predicted value}} \underbrace{+\, \varepsilon_i}_{\text{Error}}$$

Methodology: Content-based model



$$\underbrace{Y_i = \beta_0 + \sum_{p=1}^{P} \beta_p X_{i,p} + \sum_{k=2}^{K} \gamma_k Z_{i,k}}_{\textit{Predicted value}} + \underbrace{\varepsilon_i}_{\textit{Error}}$$

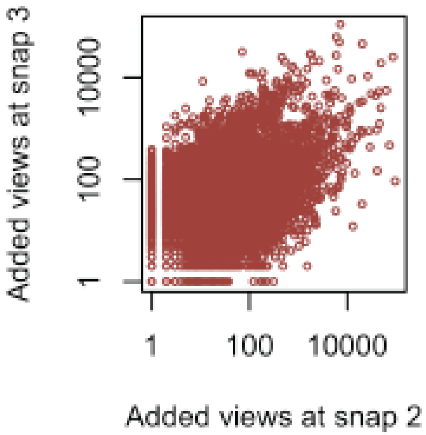
Dynamic Content Allocation for Cloud-assisted Service of Periodic Workloads

Proc. IEEE INFOCOM 2014

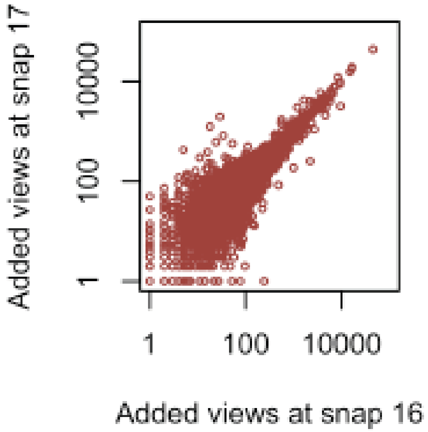
Internet Content Delivery

- Large amounts of data with varying popularity
- Multi-billion market ($8B to $20B, 2012-2015)
  - Goal: Minimize content delivery costs
- Migration to cloud data centers

Internet Content Delivery



Young videos



Old videos

E.g., *Borghol et al., "Characterizing and Modeling Popularity of User-generated Videos", Proc. IFIP Performance, Oct. 2011.*

- Large amounts of data with varying popularity
- Multi-billion market ($8B to $20B, 2012-2015)
  - Goal: Minimize content delivery costs
- Migration to cloud data centers

Internet Content Delivery

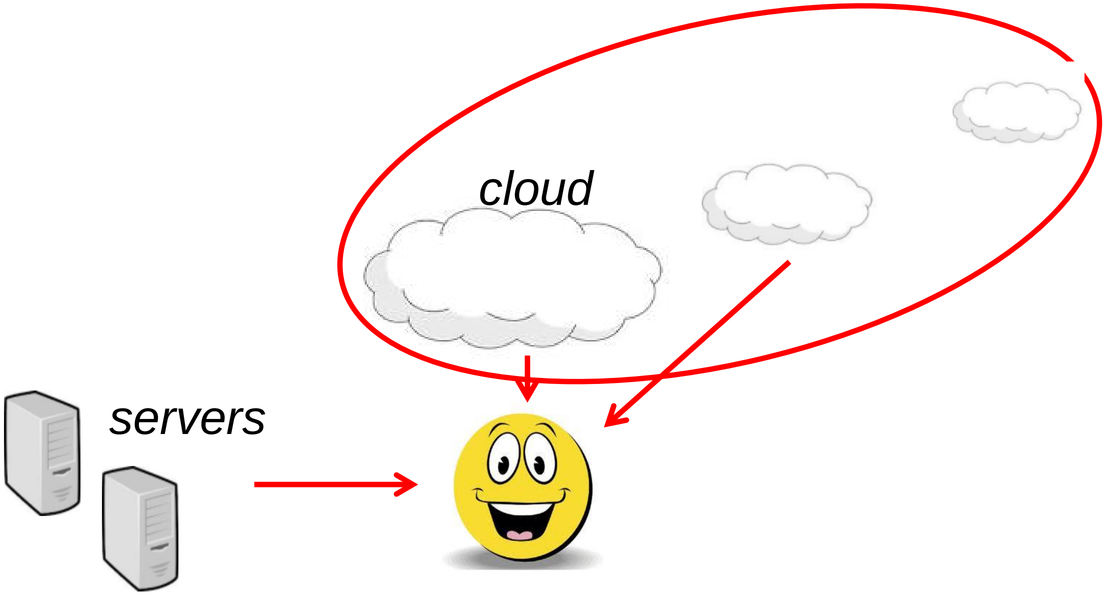
- Large amounts of data with varying popularity
- Multi-billion market ($8B to $20B, 2012-2015)
  - Goal: Minimize content delivery costs
- Migration to cloud data centers

Internet Content Delivery

- Large amounts of data with varying popularity
- Multi-billion market ($8B to $20B, 2012-2015)
  - Goal: Minimize content delivery costs
- Migration to cloud data centers

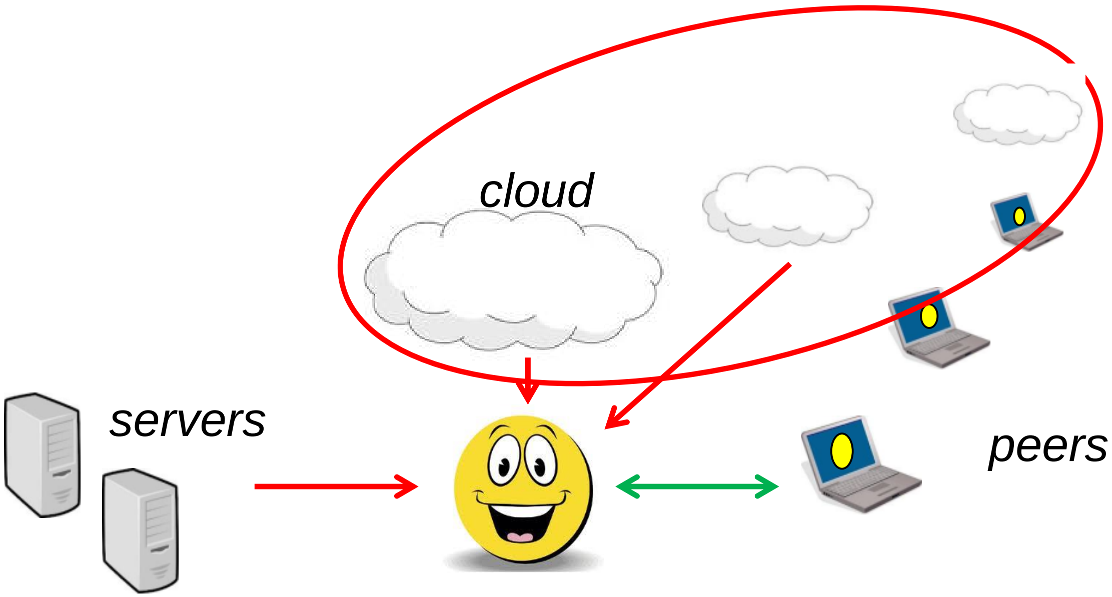
Motivation

- Goal: Minimize content delivery costs
  - Capped servers: fixed bandwidth (and storage) cap
  - Elastic cloud bandwidth: flexible, but pays premium
- Dynamic content allocation: Want to utilize capped bandwidth (and storage) as much as possible

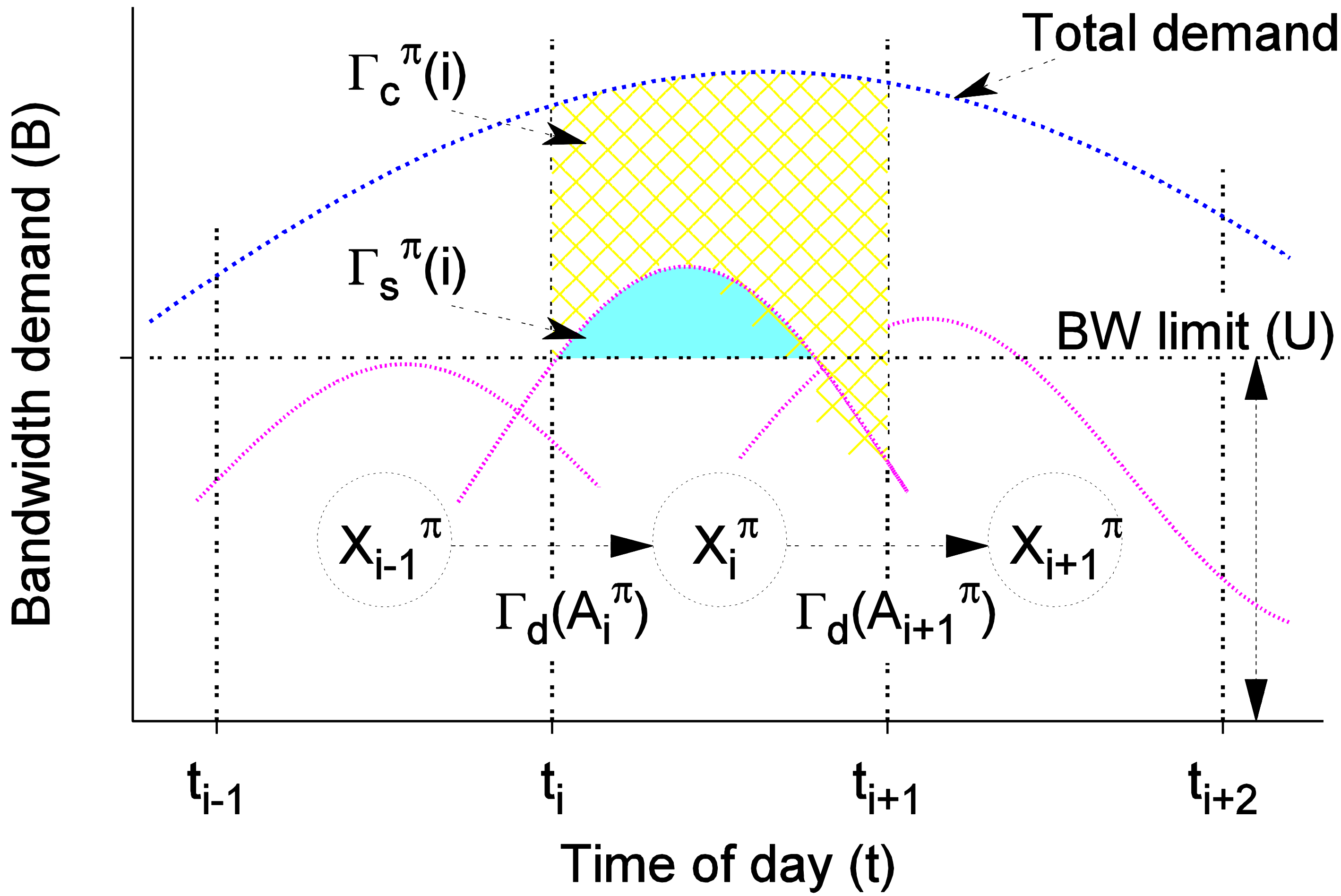


Motivation

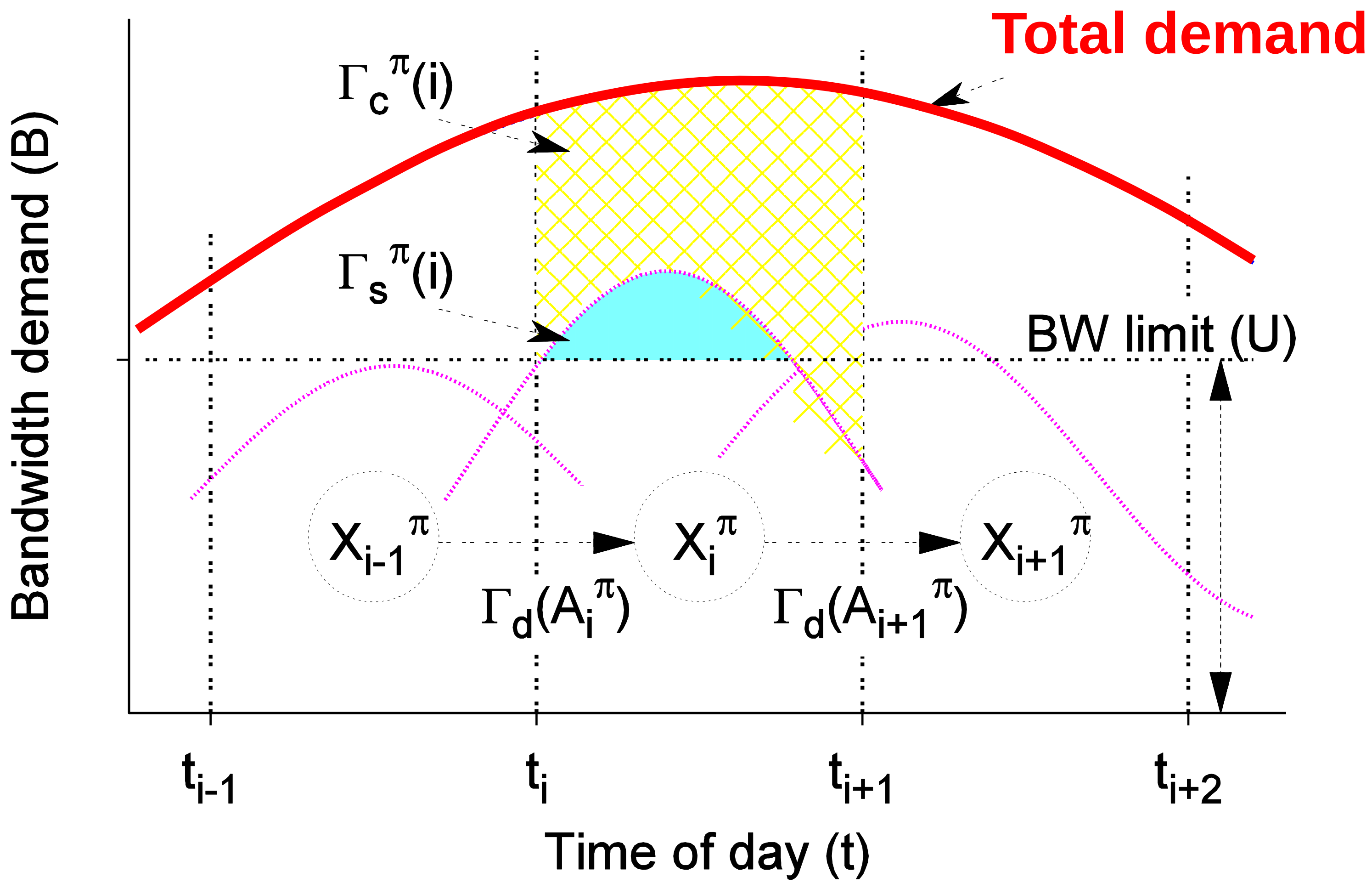
- Goal: Minimize content delivery costs
  - Capped servers: fixed bandwidth (and storage) cap
  - Elastic cloud bandwidth: flexible, but pays premium
- Dynamic content allocation: Want to utilize capped bandwidth (and storage) as much as possible



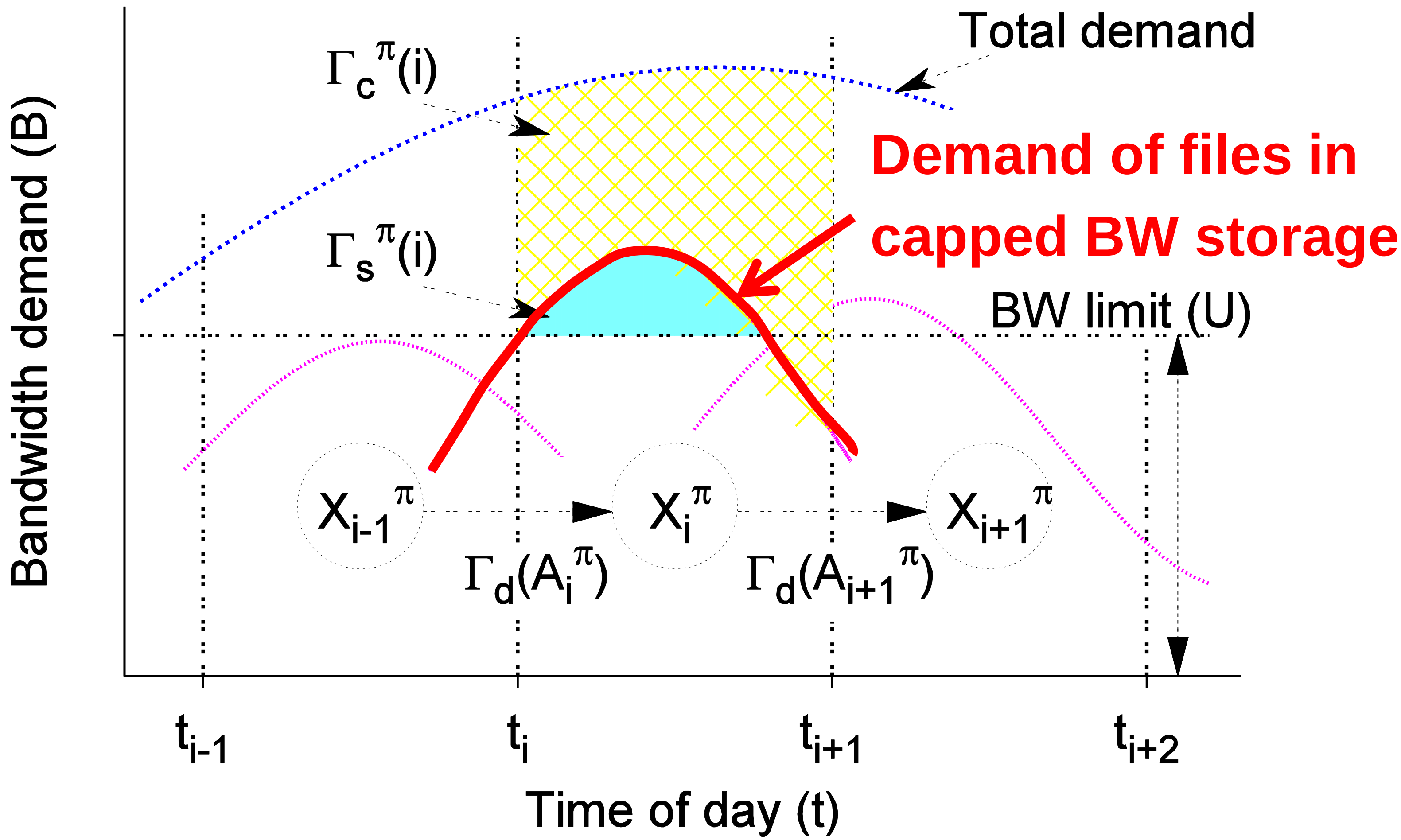
Cost minimization formulation



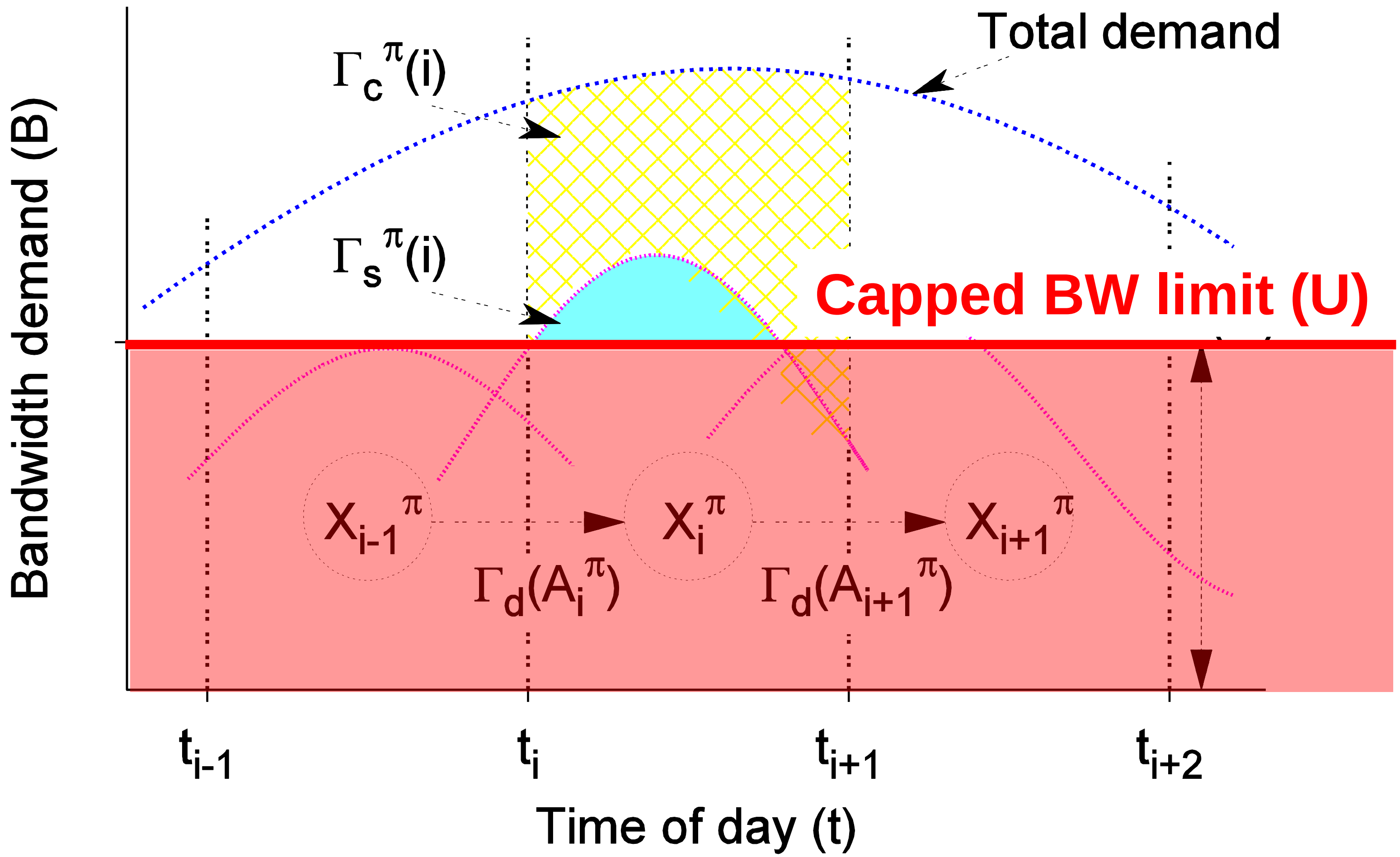
Cost minimization formulation



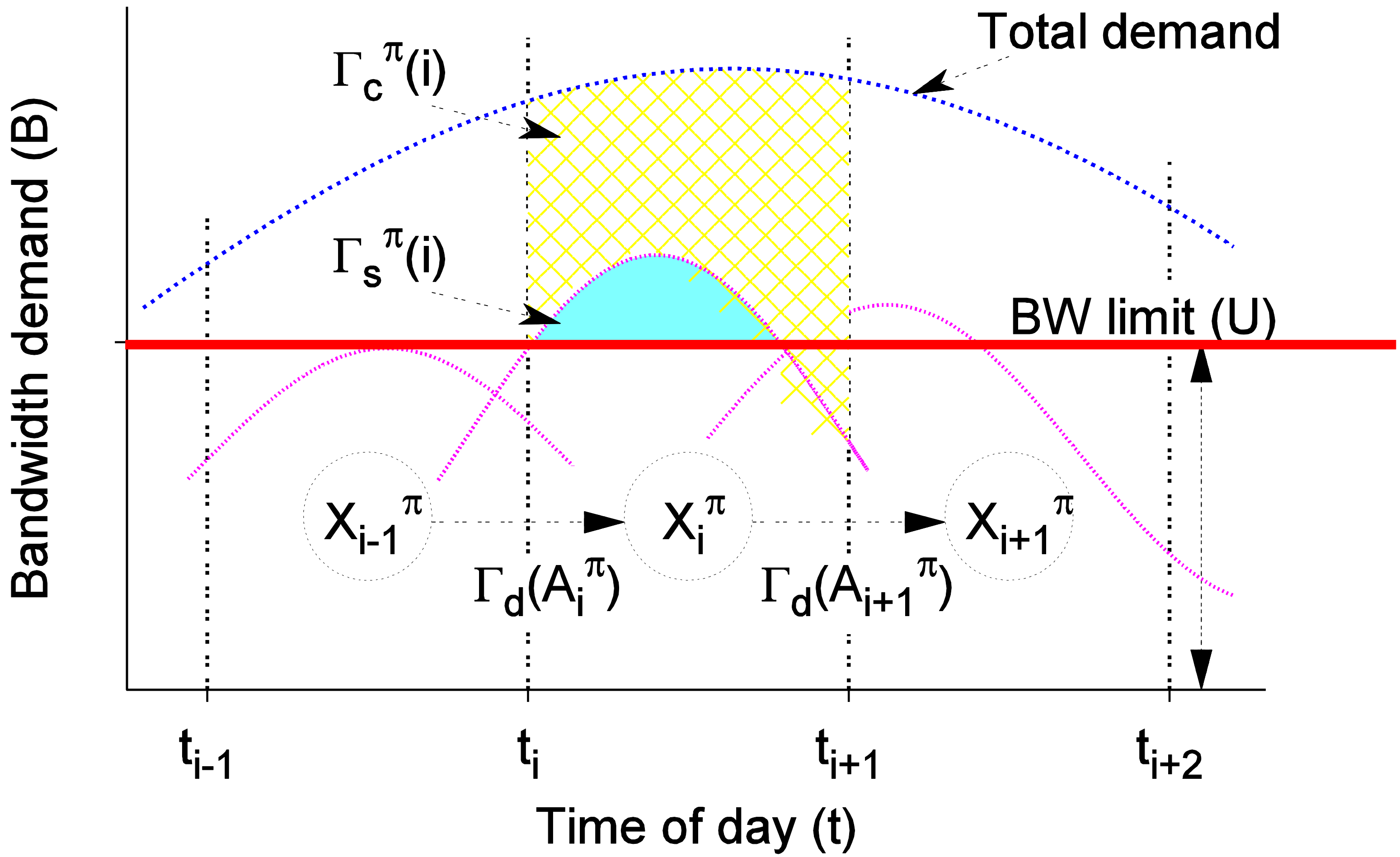
Cost minimization formulation



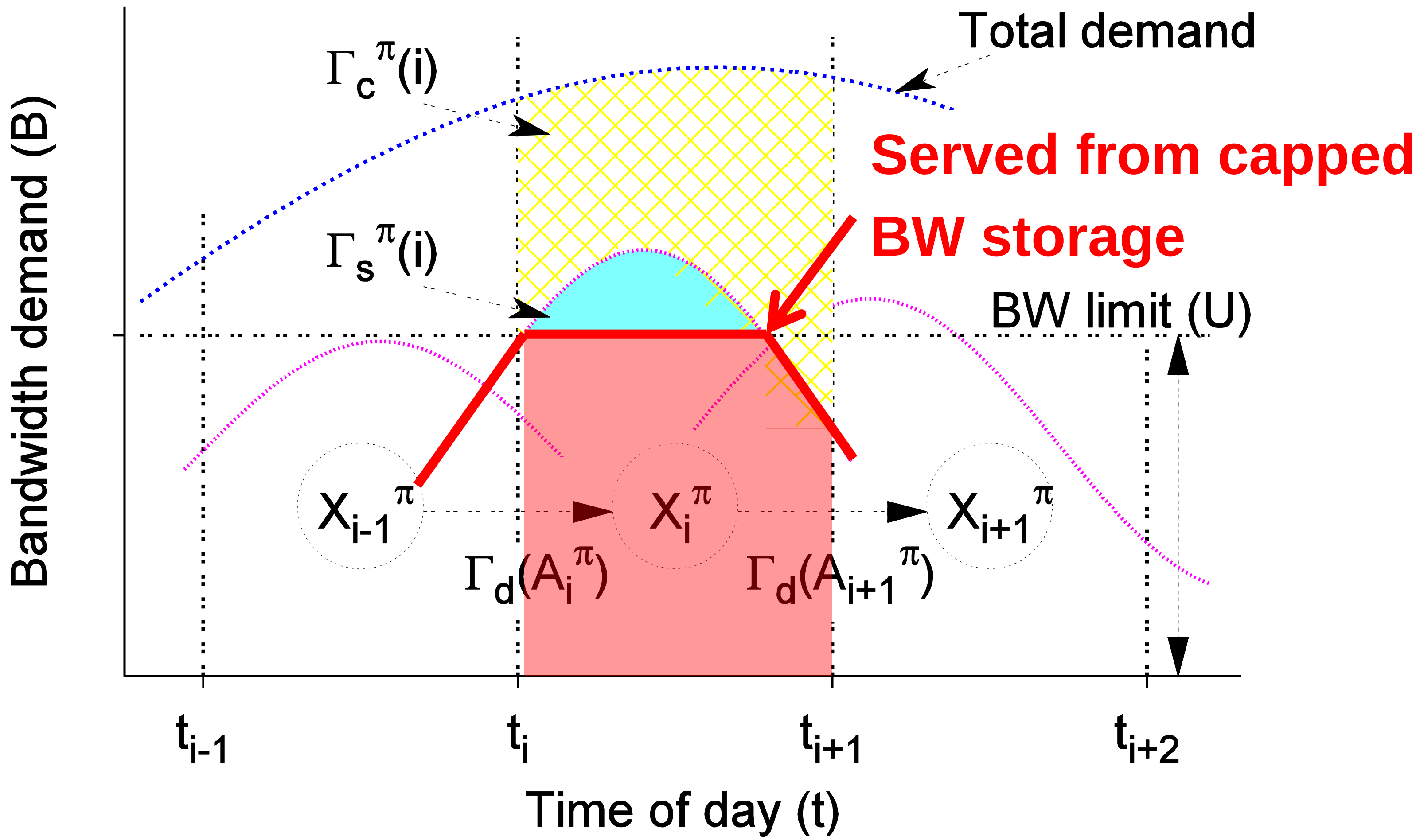
Cost minimization formulation



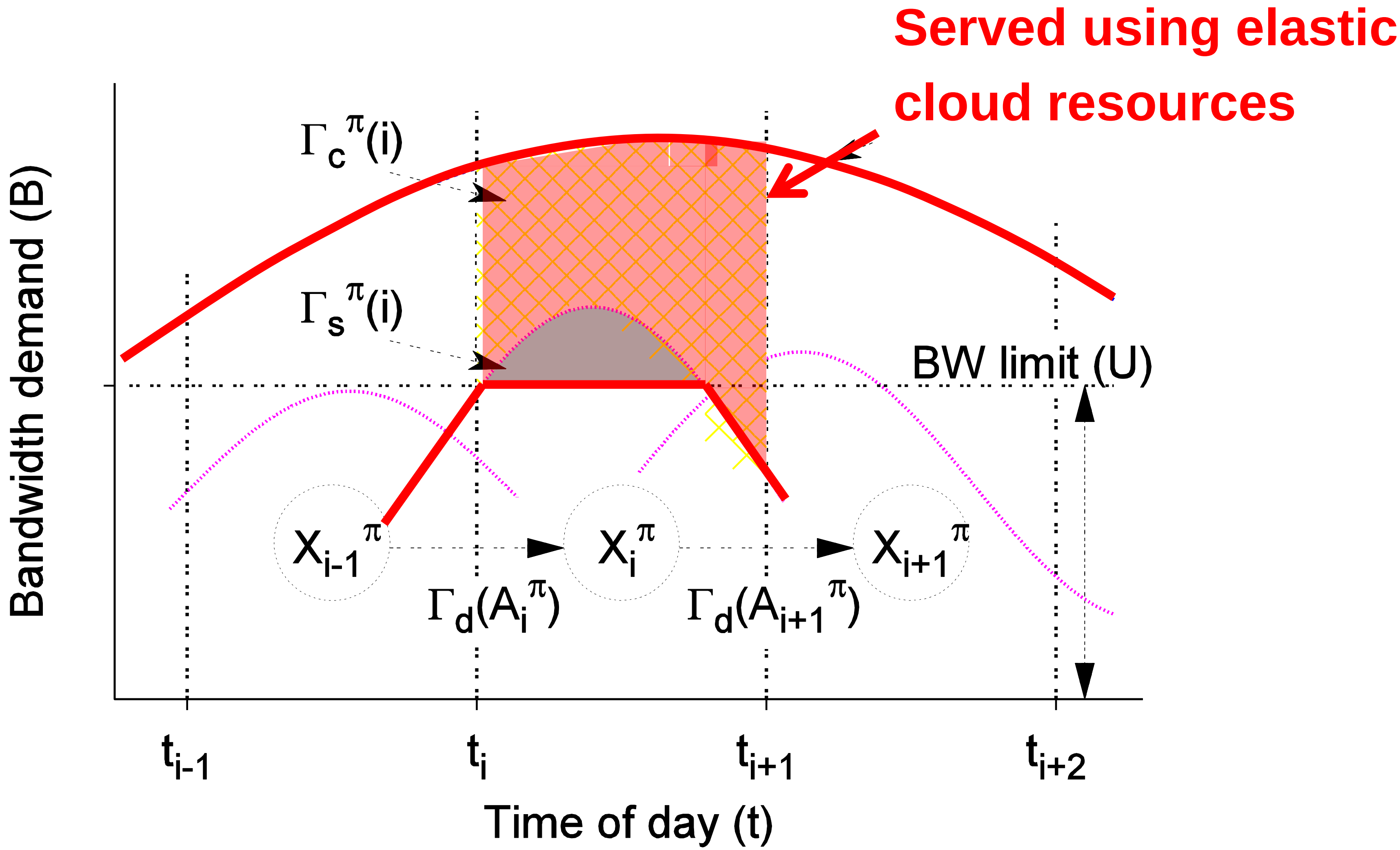
Cost minimization formulation



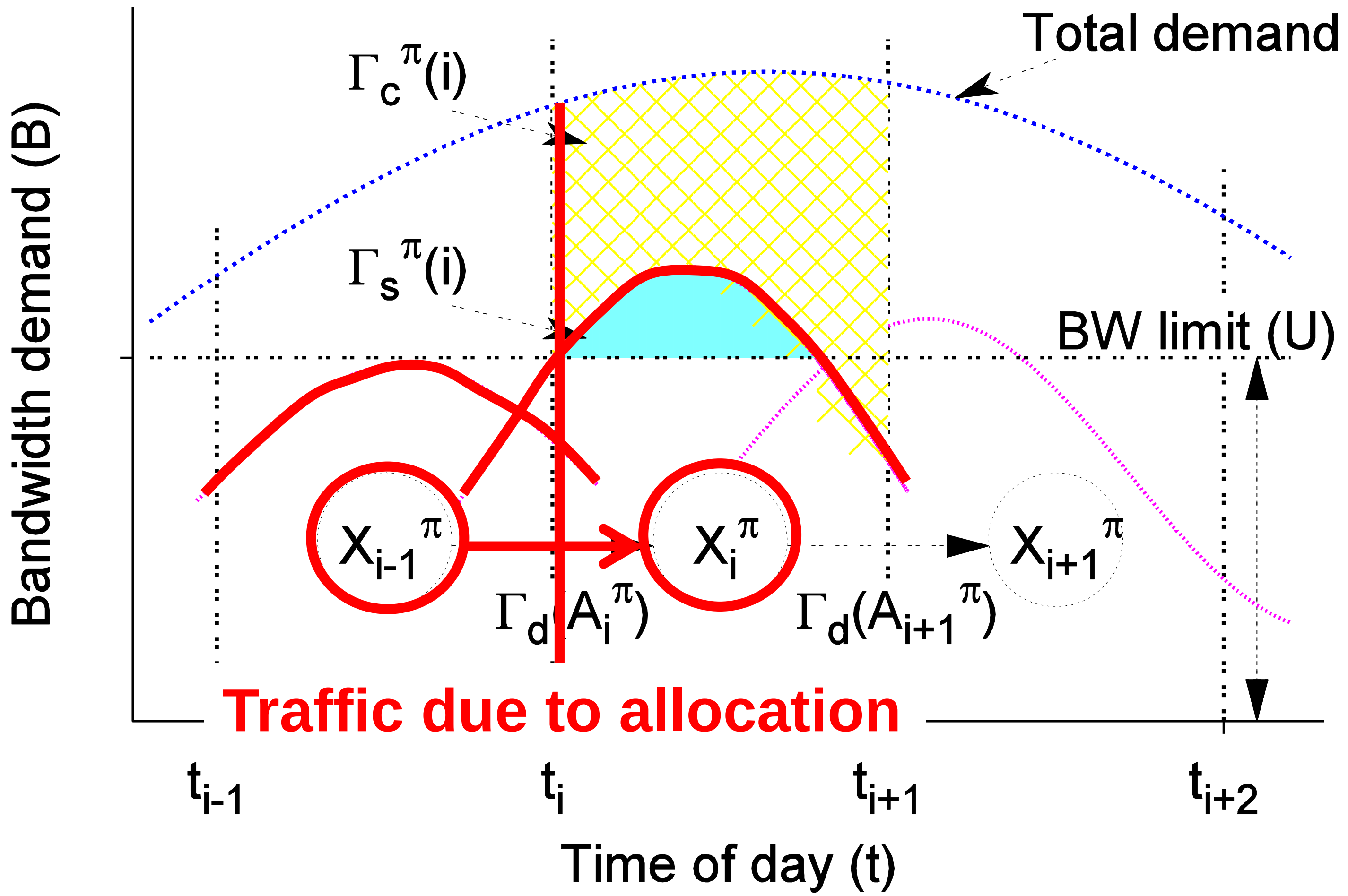
Cost minimization formulation



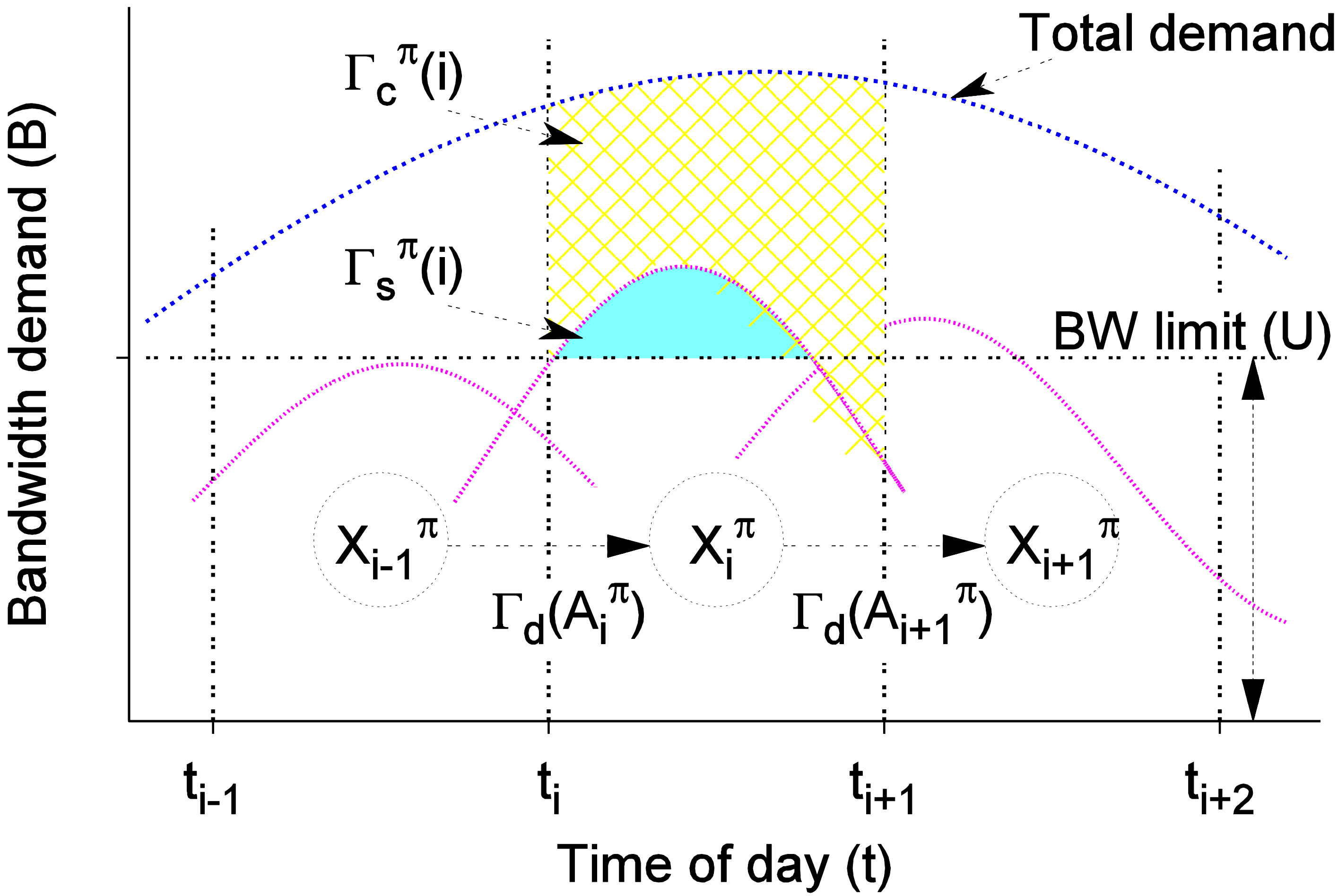
Cost minimization formulation



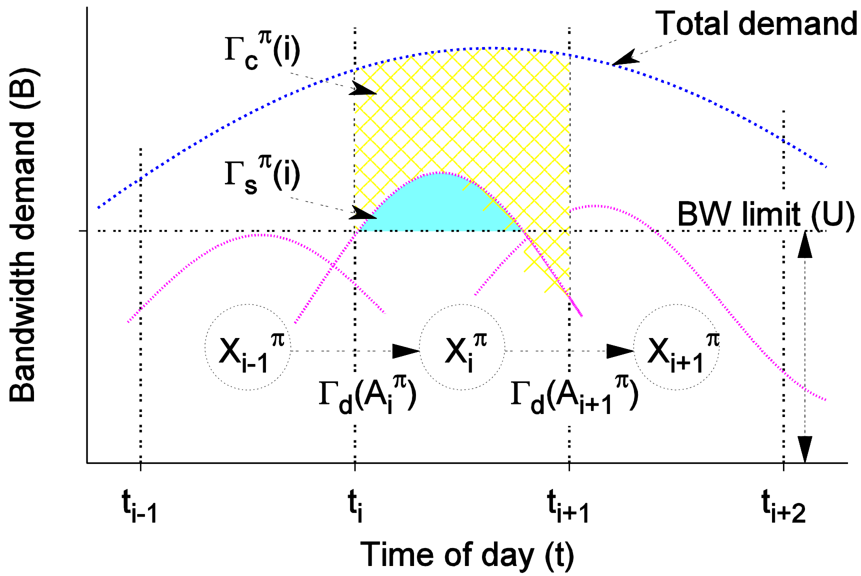
Cost minimization formulation



Cost minimization formulation



Cost minimization formulation



- Traffic of files only in cloud

$$\Gamma_c^\pi(i) = E\left[\int_{t_i^\pi}^{t_{i+1}^\pi} \sum_{f \notin \mathcal{X}_i^\pi} B_f(t)\right]$$

- Spillover traffic

$$\Gamma_s^\pi(i) = E\left[\int_{t_i^\pi}^{t_{i+1}^\pi} \left(\sum_{f \in \mathcal{X}_i^\pi} B_f(t) - U\right)^+ dt\right]$$

- Traffic due to allocation

$$\Gamma_d^\pi(A_i^\pi) = \sum_{f \in A_i^\pi} L_f$$

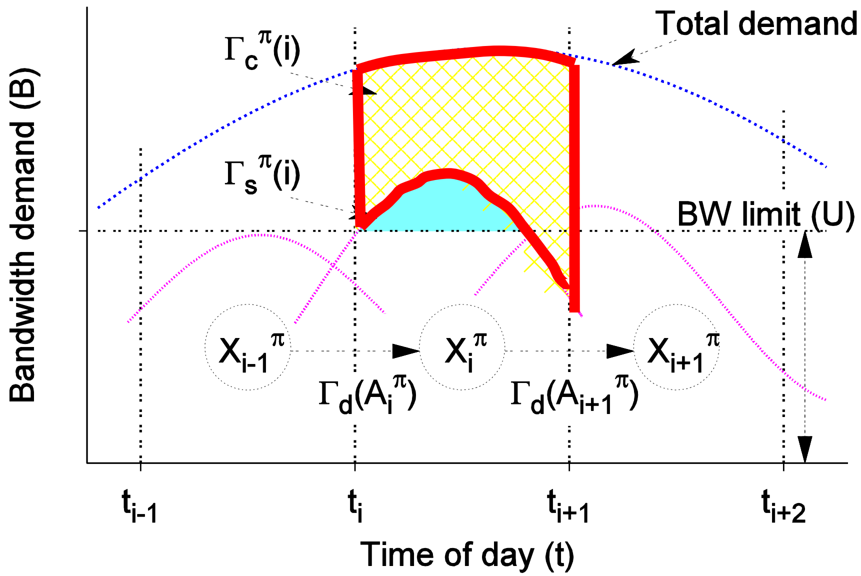
- Total expected cost

$$J^\pi(T, \mathcal{X}_0) = \gamma \times \sum_{i=0}^{I^\pi} \{\Gamma_d^\pi(A_i^\pi) + \Gamma_c^\pi(i) + \Gamma_s^\pi(i)\}$$

- Optimal policy

$$\pi^* = \arg\min_{\pi \in \Pi} J^\pi(T, \mathcal{X}_0)$$

Cost minimization formulation



- Traffic of files only in cloud

$$\Gamma_c^\pi(i) = E\left[\int_{t_i^\pi}^{t_{i+1}^\pi} \sum_{f \notin \mathcal{X}_i^\pi} B_f(t)\right]$$

- Spillover traffic

$$\Gamma_s^\pi(i) = E\left[\int_{t_i^\pi}^{t_{i+1}^\pi} \left(\sum_{f \in \mathcal{X}_i^\pi} B_f(t) - U\right)^+ dt\right]$$

- Traffic due to allocation

$$\Gamma_d^\pi(A_i^\pi) = \sum_{f \in A_i^\pi} L_f$$

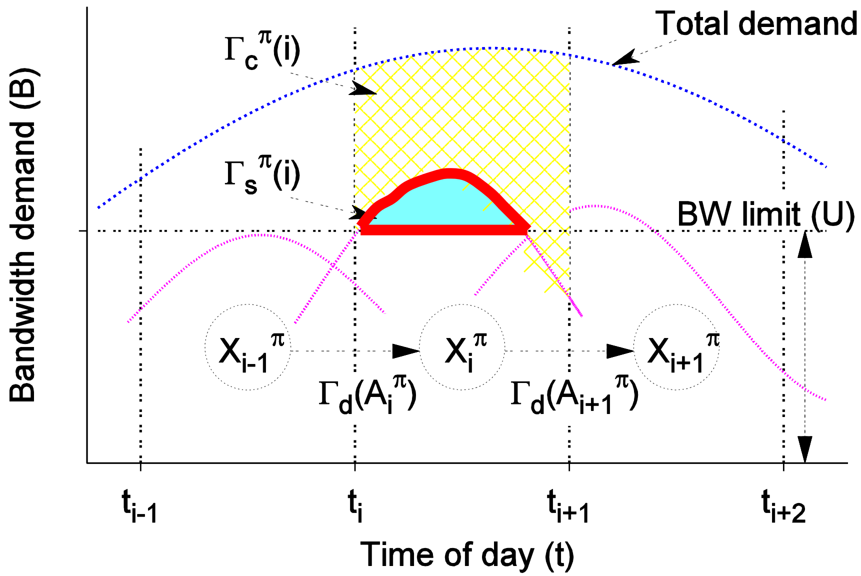
- Total expected cost

$$J^\pi(T, \mathcal{X}_0) = \gamma \times \sum_{i=0}^{I^\pi} \left\{\Gamma_d^\pi(A_i^\pi) + \Gamma_c^\pi(i) + \Gamma_s^\pi(i)\right\}$$

- Optimal policy

$$\pi^* = \arg\min_{\pi \in \Pi} J^\pi(T, \mathcal{X}_0)$$

Cost minimization formulation



- Traffic of files only in cloud

$$\Gamma_c^\pi(i) = E\left[\int_{t_i^\pi}^{t_{i+1}^\pi} \sum_{f \notin \mathcal{X}_i^\pi} B_f(t)\right]$$

- Spillover traffic

$$\Gamma_s^\pi(i) = E\left[\int_{t_i^\pi}^{t_{i+1}^\pi} \left(\sum_{f \in \mathcal{X}_i^\pi} B_f(t) - U\right)^+ dt\right]$$

- Traffic due to allocation

$$\Gamma_d^\pi(A_i^\pi) = \sum_{f \in A_i^\pi} L_f$$

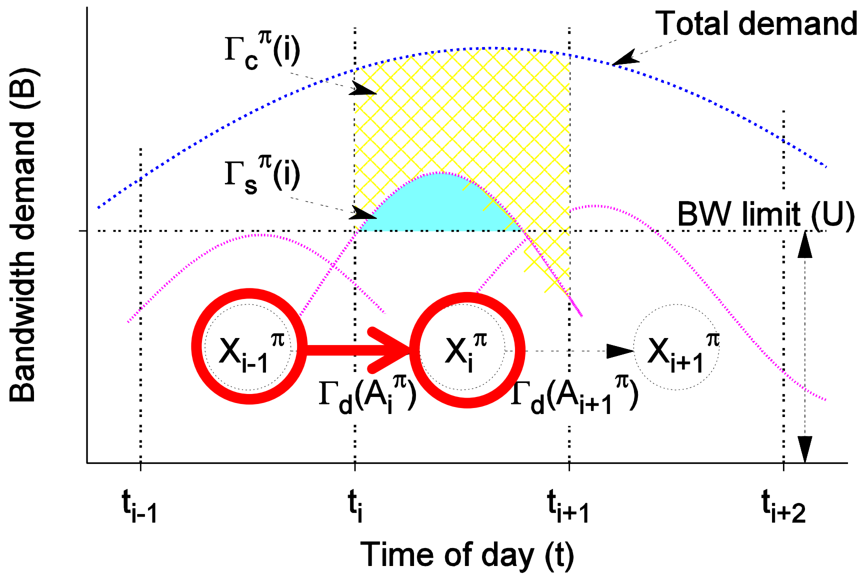
- Total expected cost

$$J^\pi(T, \mathcal{X}_0) = \gamma \times \sum_{i=0}^{I^\pi} \{\Gamma_d^\pi(A_i^\pi) + \Gamma_c^\pi(i) + \Gamma_s^\pi(i)\}$$

- Optimal policy

$$\pi^* = \arg\min_{\pi \in \Pi} J^\pi(T, \mathcal{X}_0)$$

Cost minimization formulation



- Traffic of files only in cloud

$$\Gamma_c^\pi(i) = E\left[\int_{t_i^\pi}^{t_{i+1}^\pi} \sum_{f \notin \mathcal{X}_i^\pi} B_f(t)\right]$$

- Spillover traffic

$$\Gamma_s^\pi(i) = E\left[\int_{t_i^\pi}^{t_{i+1}^\pi} \left(\sum_{f \in \mathcal{X}_i^\pi} B_f(t) - U\right)^+ dt\right]$$

- Traffic due to allocation

$$\Gamma_d^\pi(A_i^\pi) = \sum_{f \in A_i^\pi} L_f$$

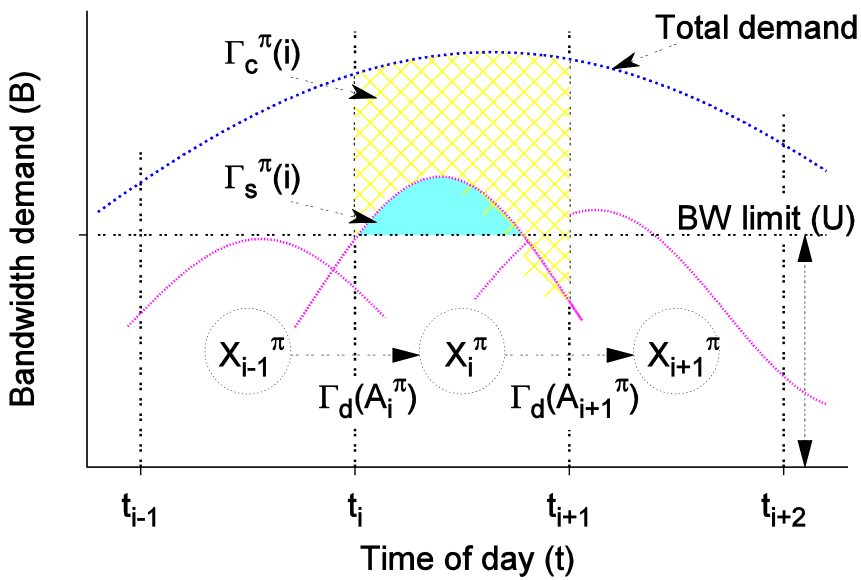
- Total expected cost

$$J^\pi(T, \mathcal{X}_0) = \gamma \times \sum_{i=0}^{I^\pi} \{\Gamma_d^\pi(A_i^\pi) + \Gamma_c^\pi(i) + \Gamma_s^\pi(i)\}$$

- Optimal policy

$$\pi^* = \arg\min_{\pi \in \Pi} J^\pi(T, \mathcal{X}_0)$$

Cost minimization formulation



- Traffic of files only in cloud

$$\Gamma_c^\pi(i) = E\left[\int_{t_i^\pi}^{t_{i+1}^\pi} \sum_{f \notin \mathcal{X}_i^\pi} B_f(t)\right]$$

- Spillover traffic

$$\Gamma_s^\pi(i) = E\left[\int_{t_i^\pi}^{t_{i+1}^\pi} \left(\sum_{f \in \mathcal{X}_i^\pi} B_f(t) - U\right)^+ dt\right]$$

- Traffic due to allocation

$$\Gamma_d^\pi(A_i^\pi) = \sum_{f \in A_i^\pi} L_f$$

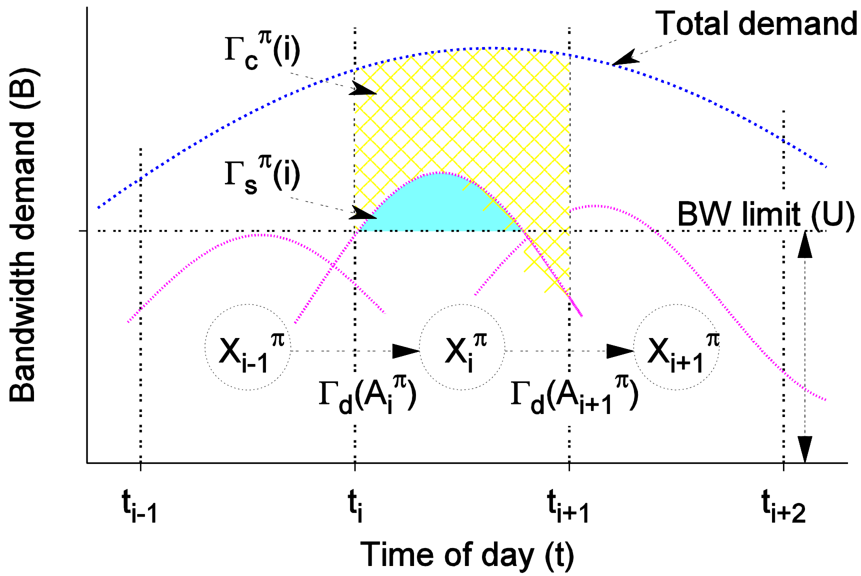
- Total expected cost

$$J^\pi(T, \mathcal{X}_0) = \gamma \times \sum_{i=0}^{I^\pi} \left\{\Gamma_d^\pi(A_i^\pi) + \Gamma_c^\pi(i) + \Gamma_s^\pi(i)\right\}$$

- Optimal policy

$$\pi^* = \arg\min_{\pi \in \Pi} J^\pi(T, \mathcal{X}_0)$$

Cost minimization formulation



- Traffic of files only in cloud

$$\Gamma_c^\pi(i) = E\left[\int_{t_i^\pi}^{t_{i+1}^\pi} \sum_{f \notin \mathcal{X}_i^\pi} B_f(t)\right]$$

- Spillover traffic

$$\Gamma_s^\pi(i) = E\left[\int_{t_i^\pi}^{t_{i+1}^\pi} \left(\sum_{f \in \mathcal{X}_i^\pi} B_f(t) - U\right)^+ dt\right]$$

- Traffic due to allocation

$$\Gamma_d^\pi(A_i^\pi) = \sum_{f \in A_i^\pi} L_f$$

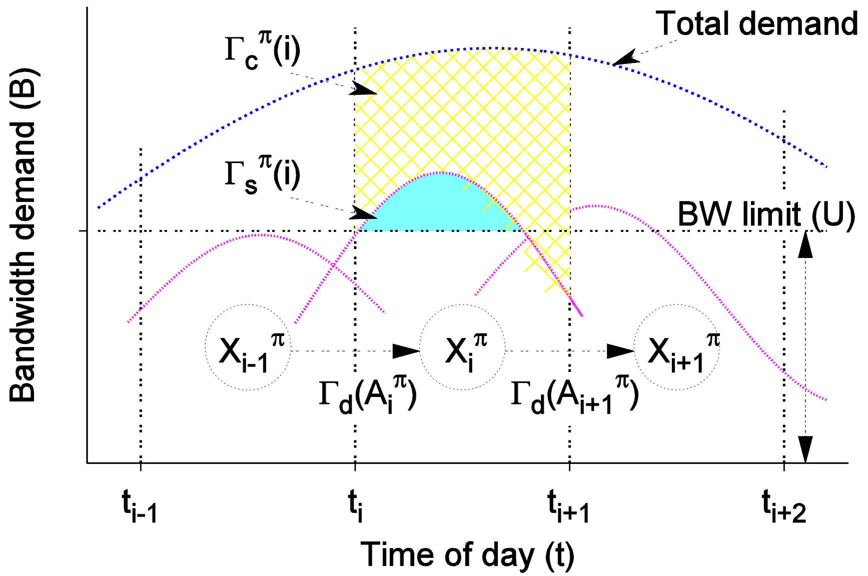
- Total expected cost

$$J^\pi(T, \mathcal{X}_0) = \gamma \times \sum_{i=0}^{I^\pi} \{\Gamma_d^\pi(A_i^\pi) + \Gamma_c^\pi(i) + \Gamma_s^\pi(i)\}$$

- Optimal policy

$$\pi^* = \arg\min_{\pi \in \Pi} J^\pi(T, \mathcal{X}_0)$$

Cost minimization formulation



- Traffic of files only in cloud

$$\Gamma_c^\pi(i) = E\left[\int_{t_i^\pi}^{t_{i+1}^\pi} \sum_{f \notin \mathcal{X}_i^\pi} B_f(t)\right]$$

- Spillover traffic

$$\Gamma_s^\pi(i) = E\left[\int_{t_i^\pi}^{t_{i+1}^\pi} \left(\sum_{f \in \mathcal{X}_i^\pi} B_f(t) - U\right)^+ dt\right]$$

- Traffic due to allocation

$$\Gamma_d^\pi(A_i^\pi) = \sum_{f \in A_i^\pi} L_f$$

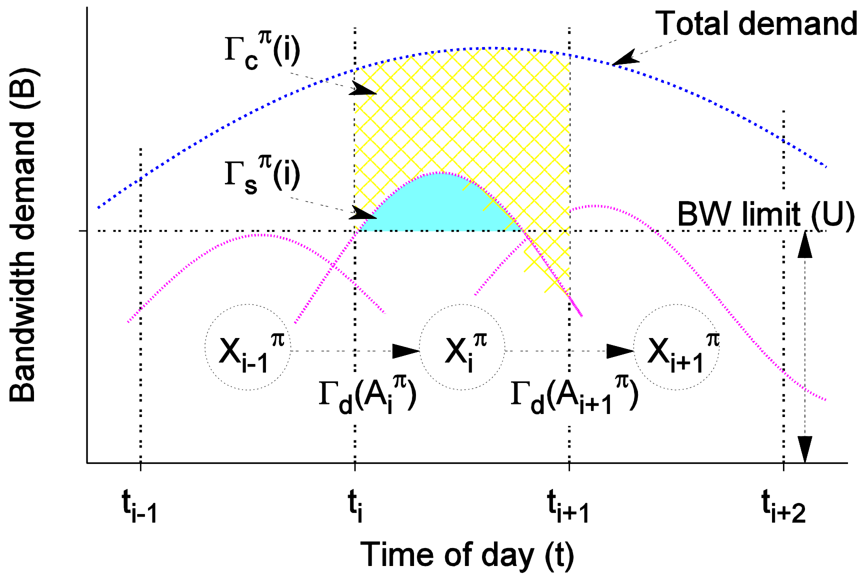
- Total expected cost

$$J^\pi(T, \mathcal{X}_0) = \gamma \times \sum_{i=0}^{I^\pi} \{\Gamma_d^\pi(A_i^\pi) + \Gamma_c^\pi(i) + \Gamma_s^\pi(i)\}$$

- Optimal policy

$$\pi^* = \arg\min_{\pi \in \Pi} J^\pi(T, \mathcal{X}_0)$$

Cost minimization formulation



- Traffic of files only in cloud

$$\Gamma_c^{\pi}(i) = E\left[\int_{t_i^{\pi}}^{t_{i+1}^{\pi}} \sum_{f \notin \mathcal{X}_i^{\pi}} B_f(t)\right]$$

- Spillover traffic

$$\Gamma_s^{\pi}(i) = E\left[\int_{t_i^{\pi}}^{t_{i+1}^{\pi}} \left(\sum_{f \in \mathcal{X}_i^{\pi}} B_f(t) - U\right)^{+} dt\right]$$

- Traffic due to allocation

$$\Gamma_d^{\pi}(A_i^{\pi}) = \sum_{f \in A_i^{\pi}} L_f$$

- Total expected cost

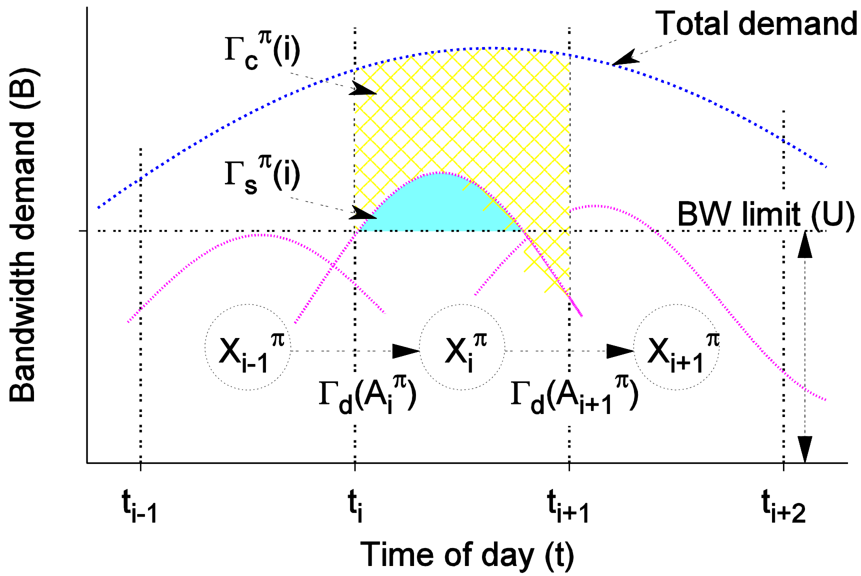
$$J^{\pi}(T, \mathcal{X}_0) = \gamma \times \sum_{i=0}^{I^{\pi}} \left\{\Gamma_d^{\pi}(A_i^{\pi}) + \Gamma_c^{\pi}(i) + \Gamma_s^{\pi}(i)\right\}$$

- Optimal policy

$$\pi^* = \arg\min_{\pi \in \Pi} J^{\pi}(T, \mathcal{X}_0)$$

Utilization maximization

~~Cost minimization formulation~~



- Traffic of files only in cloud

$$\Gamma_c^\pi(i) = E\left[\int_{t_i^\pi}^{t_{i+1}^\pi} \sum_{f \notin \mathcal{X}_i^\pi} B_f(t)\right]$$

- Spillover traffic

$$\Gamma_s^\pi(i) = E\left[\int_{t_i^\pi}^{t_{i+1}^\pi} \left(\sum_{f \in \mathcal{X}_i^\pi} B_f(t) - U\right)^+ dt\right]$$

- Traffic due to allocation

$$\Gamma_d^\pi(A_i^\pi) = \sum_{f \in A_i^\pi} L_f$$

- Total expected cost

$$J^\pi(T, \mathcal{X}_0) = \gamma \times \sum_{i=0}^{I^\pi} \left\{\Gamma_d^\pi(A_i^\pi) + \Gamma_c^\pi(i) + \Gamma_s^\pi(i)\right\}$$

- Optimal policy

$$\pi^* = \arg\min_{\pi \in \Pi} J^\pi(T, \mathcal{X}_0)$$

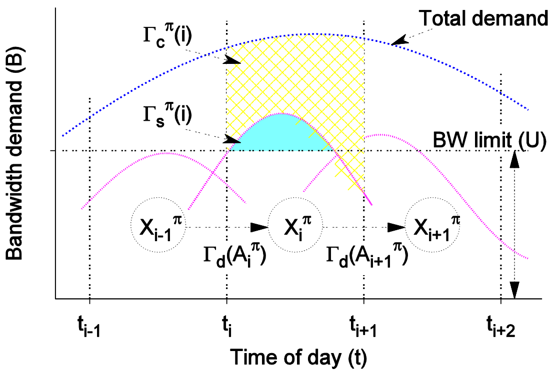
- Equivalent formulation

$$\overline{\Gamma}_s^\pi(i) = E\left[\int_{t_i^\pi}^{t_{i+1}^\pi} \min\left(U, \sum_{f \in \mathcal{X}_i^\pi} B_f(t)\right) dt\right]$$

$$U^\pi(T, \mathcal{X}_0) = \gamma \times \sum_{i=0}^{I^\pi} \left\{\overline{\Gamma}_s^\pi(i) - \Gamma_d^\pi(A_i^\pi)\right\}$$

Optimal policy $\pi^* = \arg\max_{\pi \in \Pi} U^\pi(T, \mathcal{X}_0)$

Dynamic content allocation problem



- Formulate as a finite horizon dynamic decision process problem
- Show discrete time decision process is good approximation
- Exact solution as MILP
- Provide computationally feasible approximations (and prove properties about approximation ratios)
- Validate model and algorithms using traces from Spotify

Caching and Optimized Request Routing in Cloud-based Content Delivery Systems

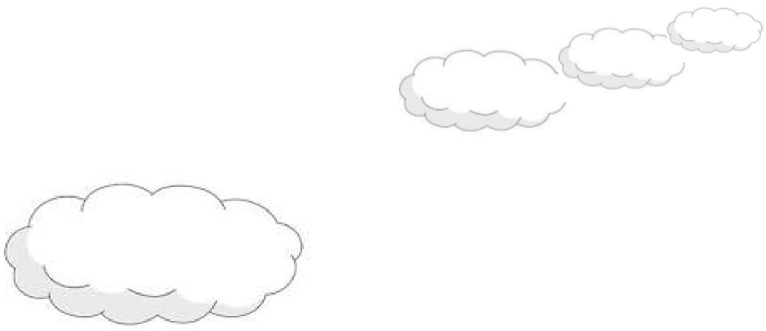
Proc. IFIP PERFORMANCE 2014.

Internet Content Delivery

- Migration to geographically distributed cloud data centers
  - Goal: Minimize content delivery costs

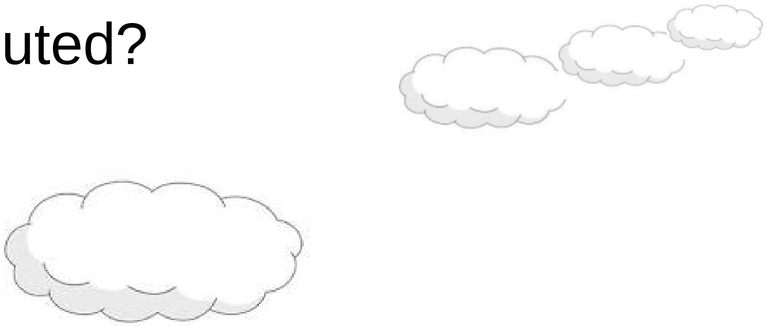
Motivation

- Geographically distributed cloud
  - Elastic cloud bandwidth and storage
  - When sufficiently expensive storage costs, not all contents should be cached at all locations



Motivation

- Geographically distributed cloud
  - Elastic cloud bandwidth and storage
  - When sufficiently expensive storage costs, not all contents should be cached at all locations
- Two policy questions arise
  - What content should be cached where?
  - How should requests be routed?



Motivation

- Geographically distributed cloud
  - Elastic cloud bandwidth and storage
  - When sufficiently expensive storage costs, not all contents should be cached at all locations
- Two policy questions arise
  - What content should be cached where?
  - How should requests be routed?

Motivation

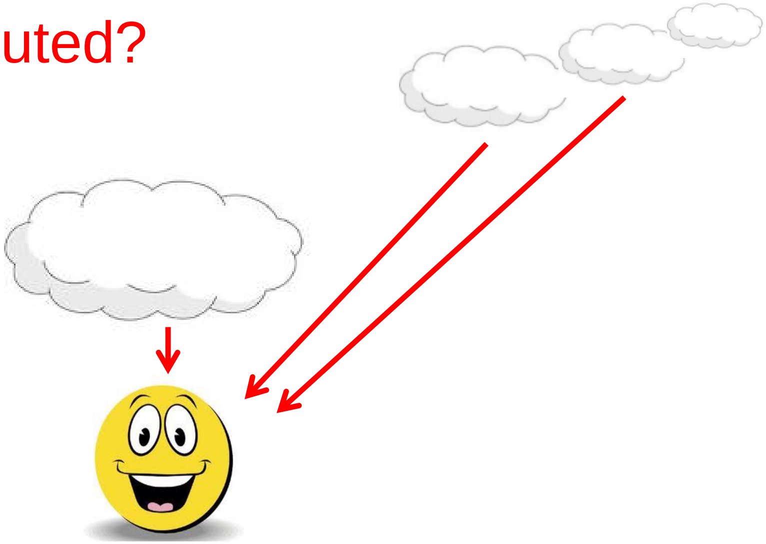
- Geographically distributed cloud
  - Elastic cloud bandwidth and storage
  - When sufficiently expensive storage costs, not all contents should be cached at all locations
- Two policy questions arise
  - What content should be cached where?
  - How should requests be routed?

Motivation

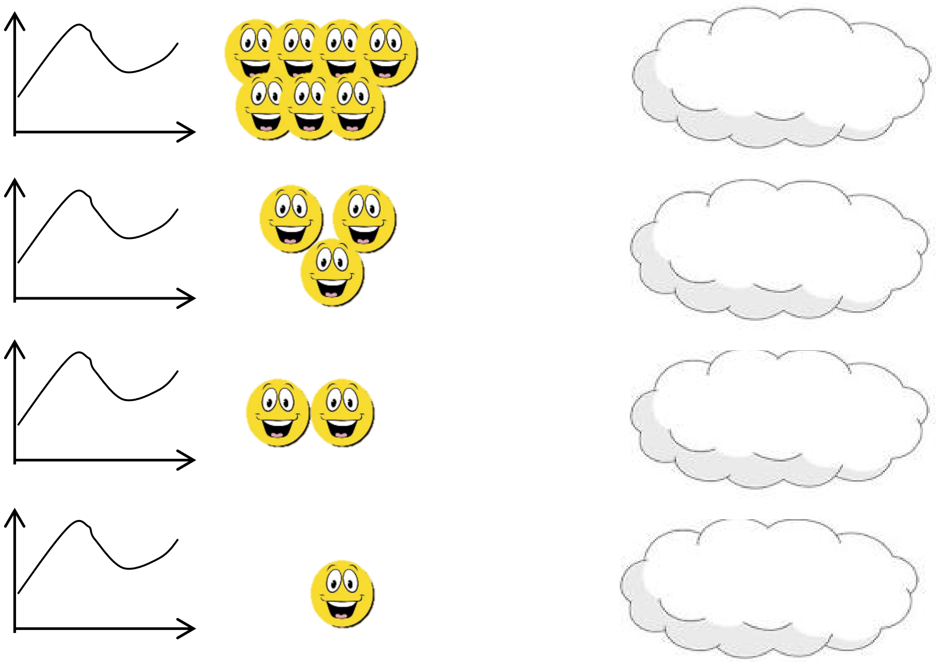
- Geographically distributed cloud
  - Elastic cloud bandwidth and storage
  - When sufficiently expensive storage costs, not all contents should be cached at all locations
- Two policy questions arise
  - What content should be cached where?
  - How should requests be routed?

Motivation

- Geographically distributed cloud
  - Elastic cloud bandwidth and storage
  - When sufficiently expensive storage costs, not all contents should be cached at all locations
- Two policy questions arise
  - What content should be cached where?
  - How should requests be routed?



Dynamic TTL-based approach

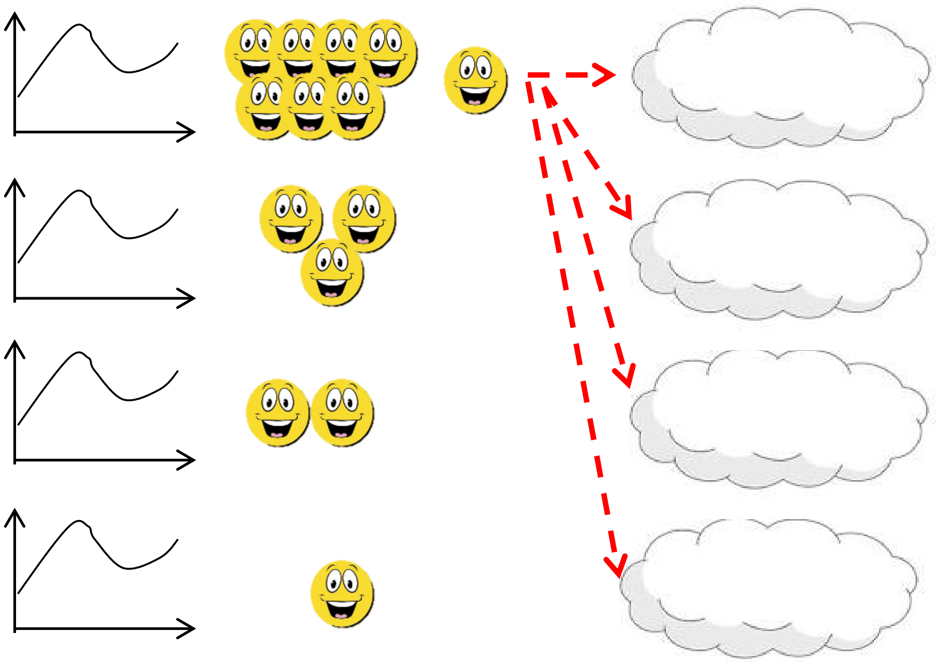


1) Request routing

2) TTL-caching

- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

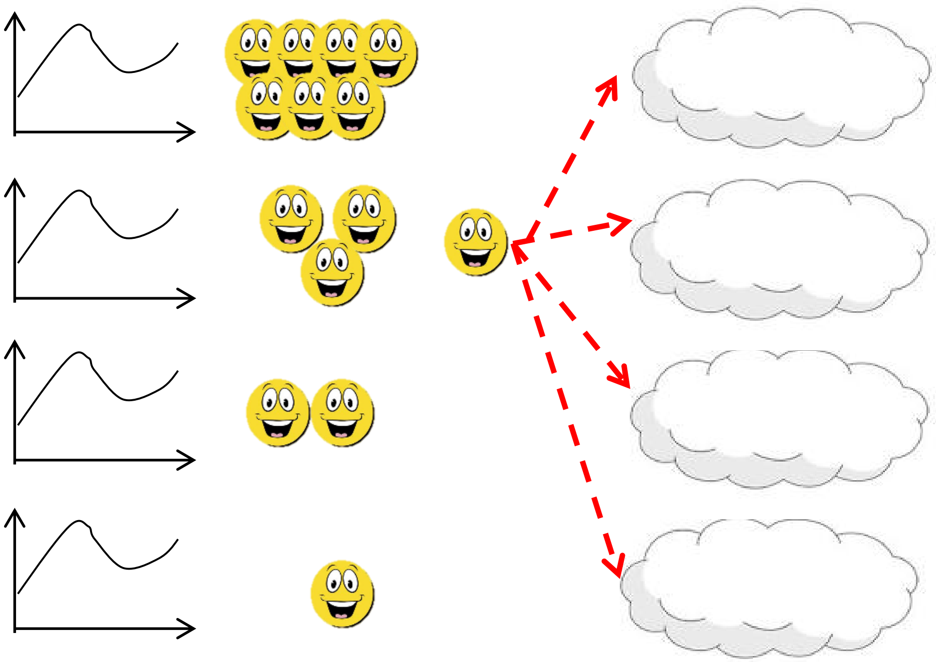
Dynamic TTL-based approach



1) Request routing
2) TTL-caching

- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

Dynamic TTL-based approach

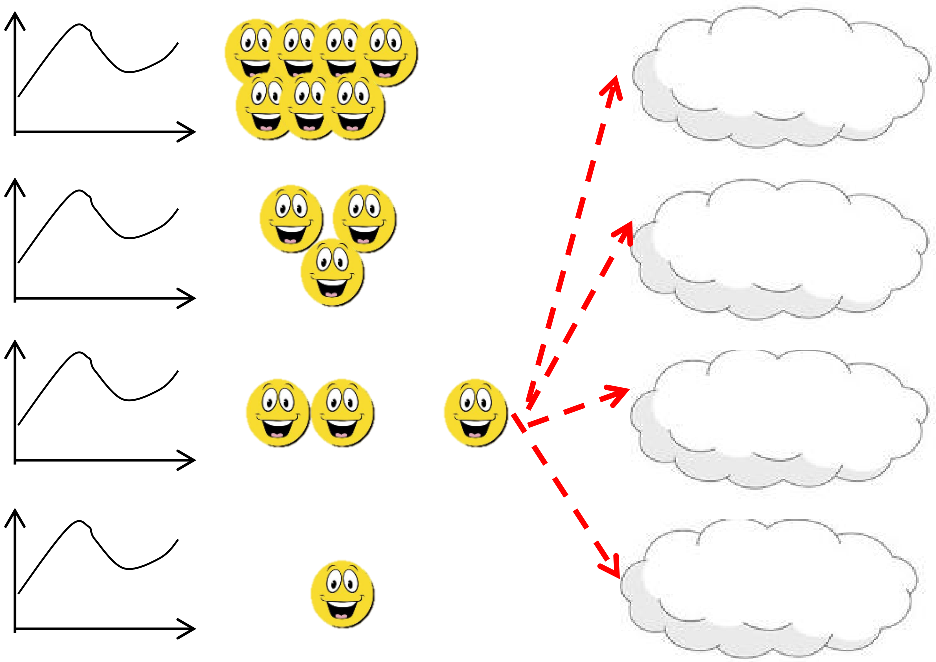


1) Request routing

2) TTL-caching

- Elastic cloud bandwidth and storage
    - TTL T_i used at each server location
- Optimized request routing determines content replication

Dynamic TTL-based approach

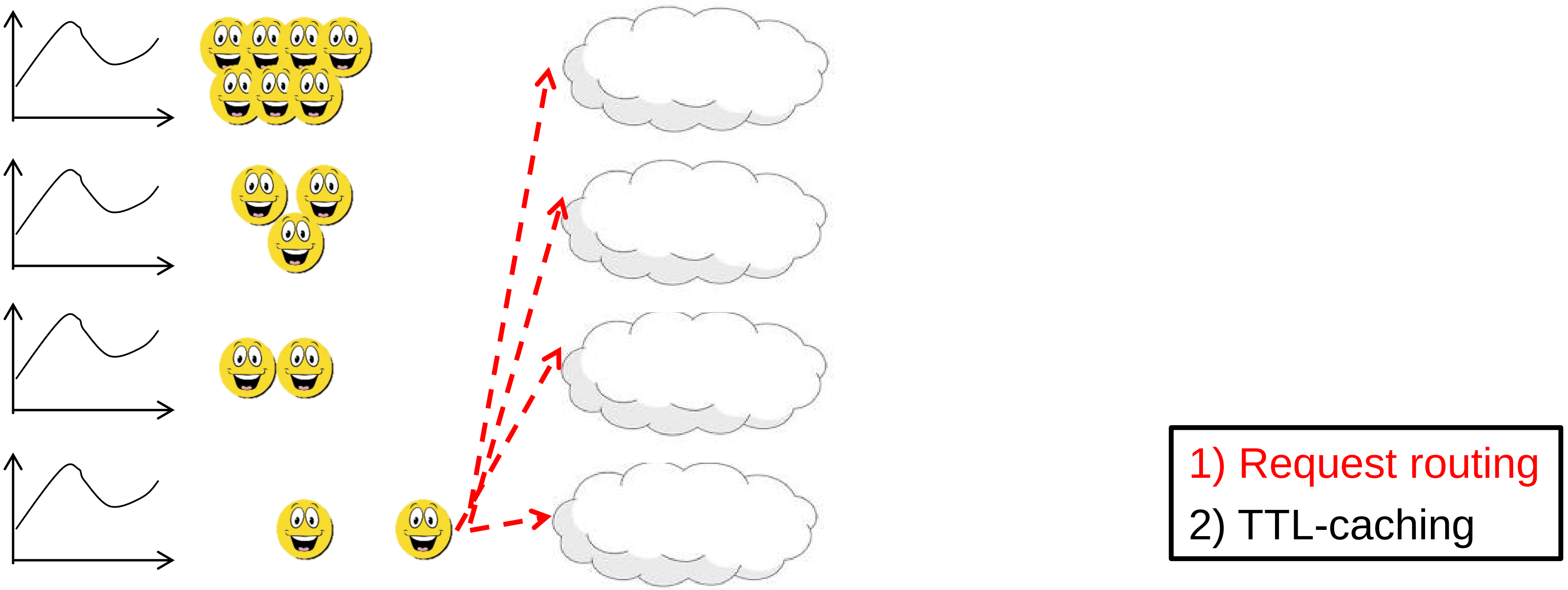


1) Request routing

2) TTL-caching

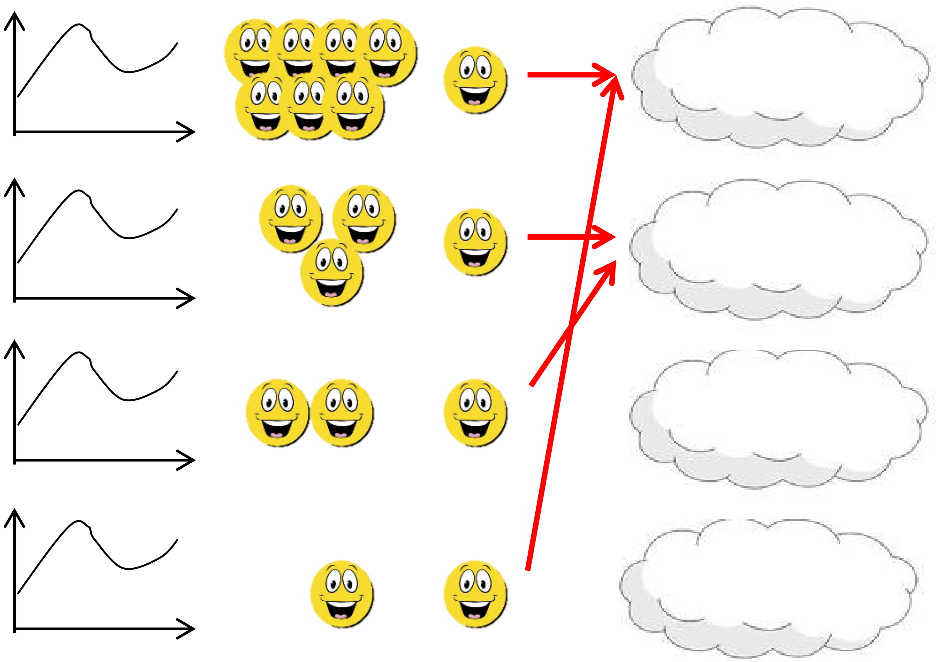
- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

Dynamic TTL-based approach



- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

Dynamic TTL-based approach

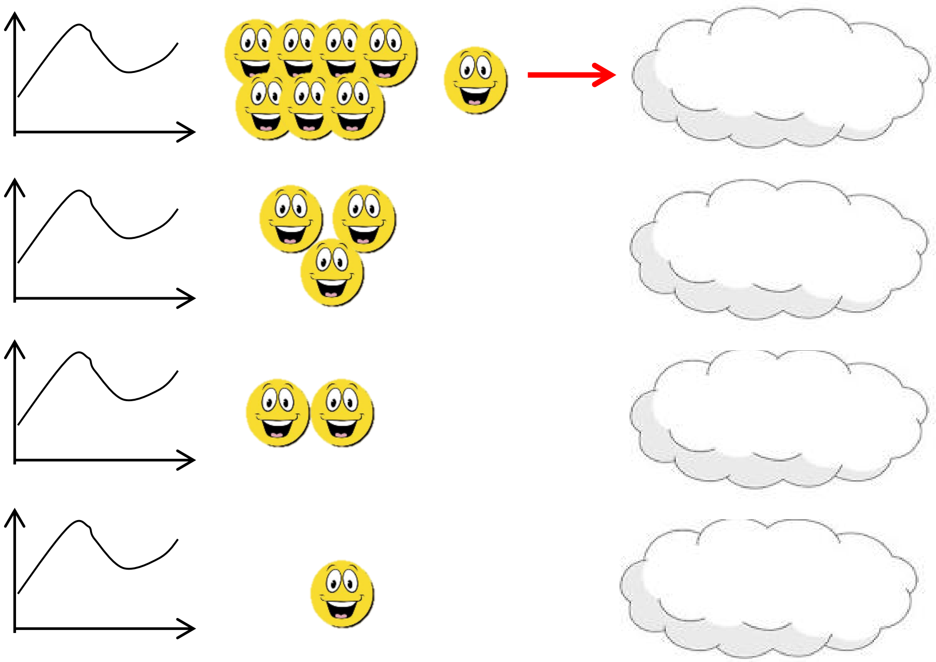


1) Request routing

2) TTL-caching

- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

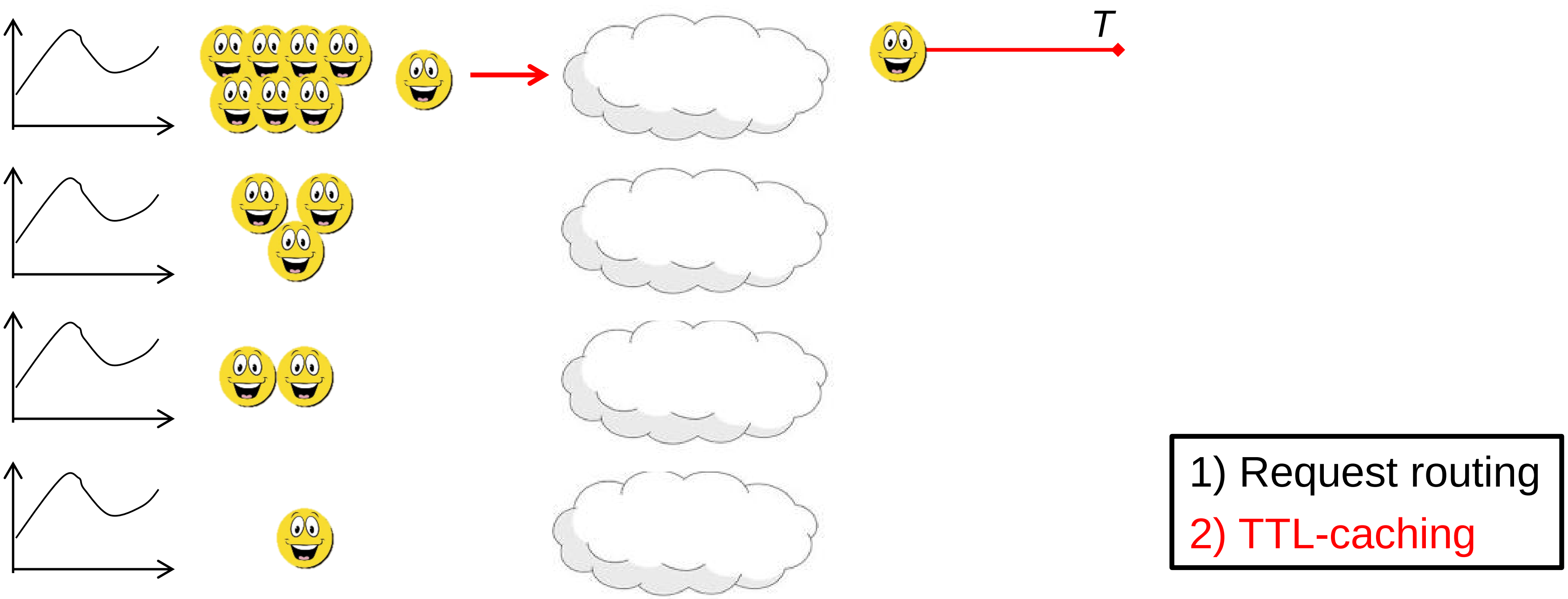
Dynamic TTL-based approach



1) Request routing
2) TTL-caching

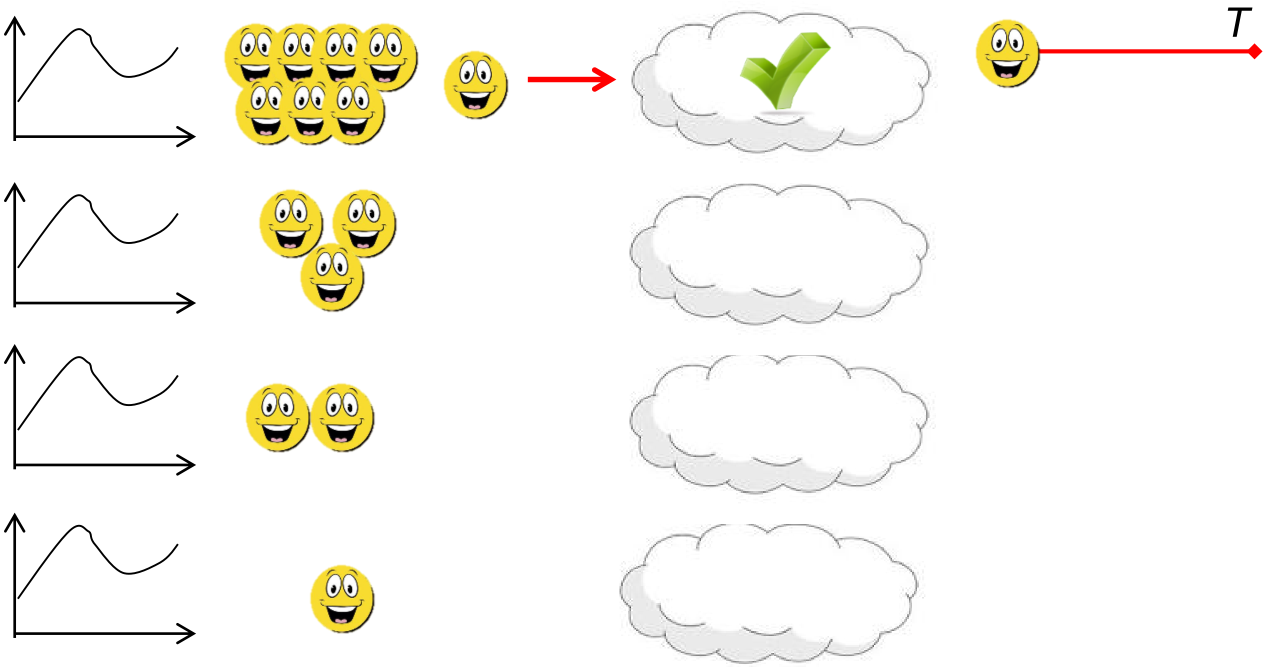
- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

Dynamic TTL-based approach



- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

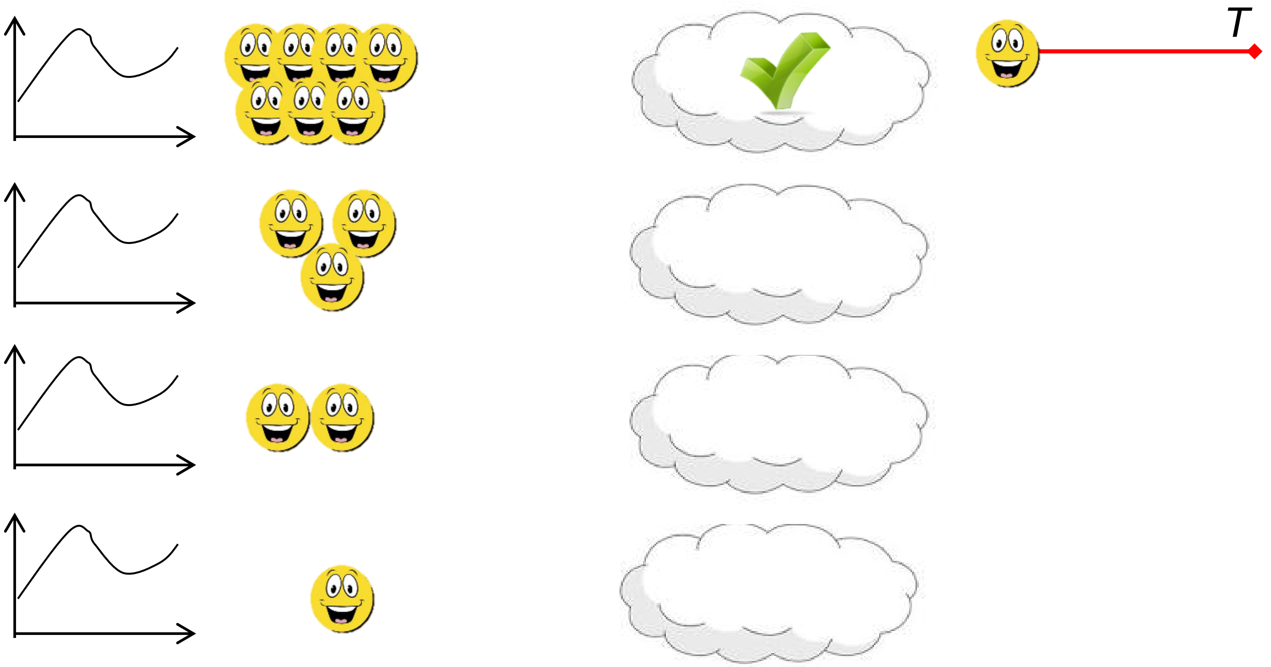
Dynamic TTL-based approach



1) Request routing
2) TTL-caching

- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

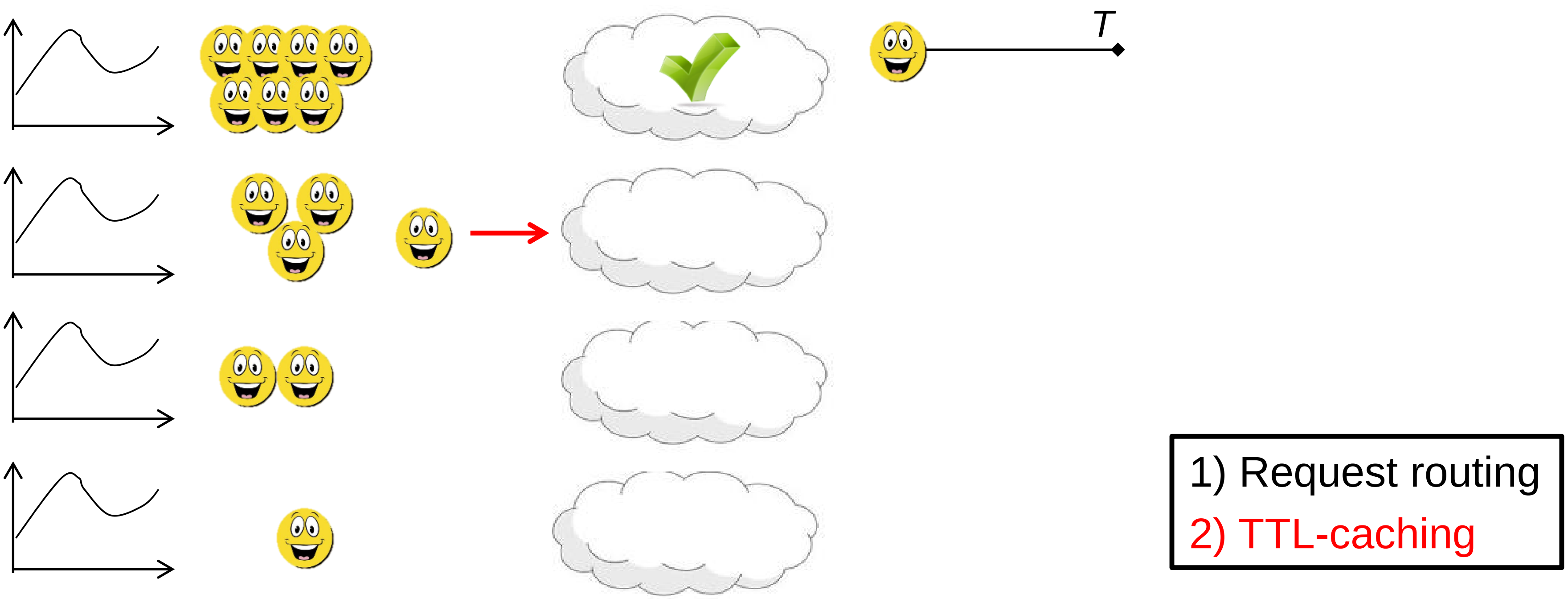
Dynamic TTL-based approach



1) Request routing
2) TTL-caching

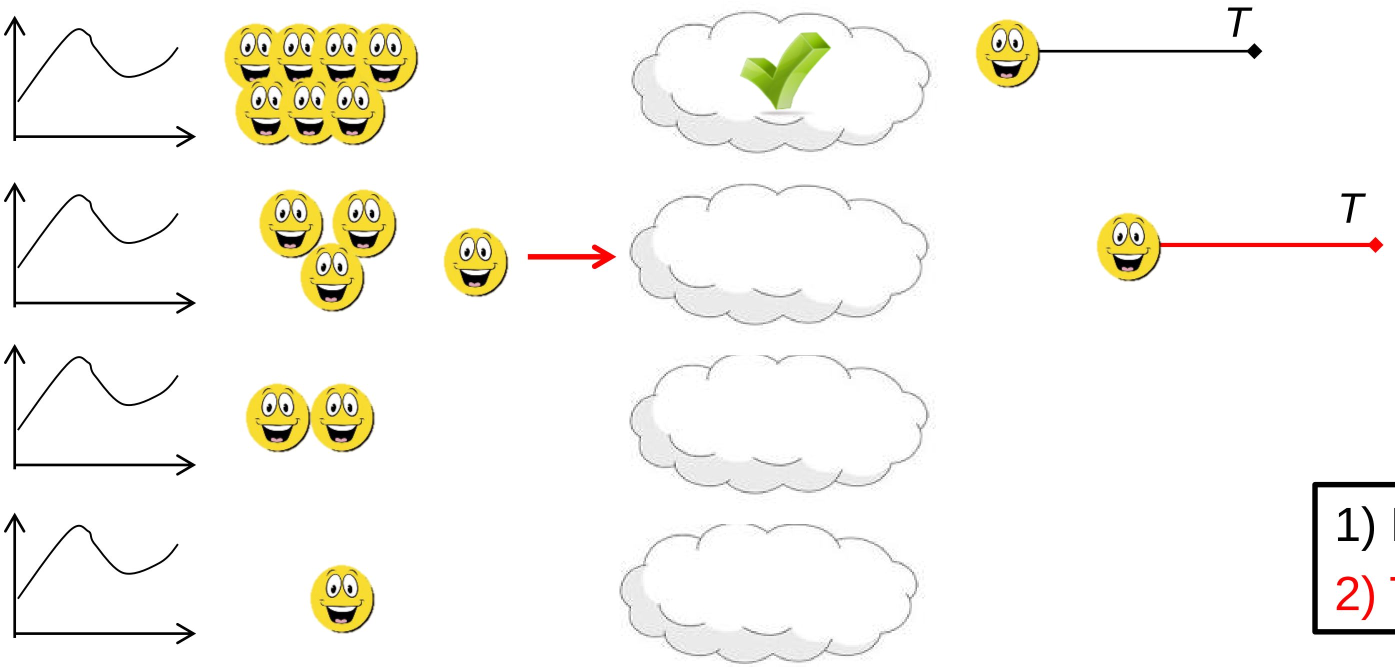
- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

Dynamic TTL-based approach



- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

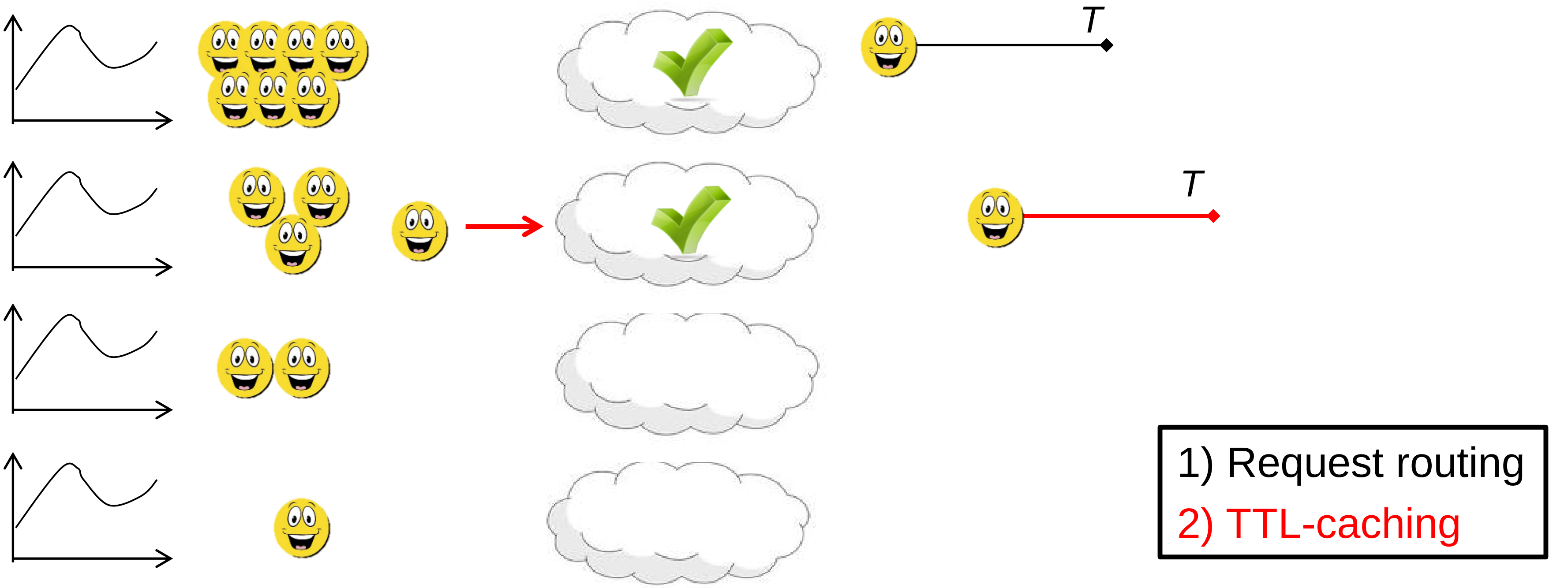
Dynamic TTL-based approach



1) Request routing
2) TTL-caching

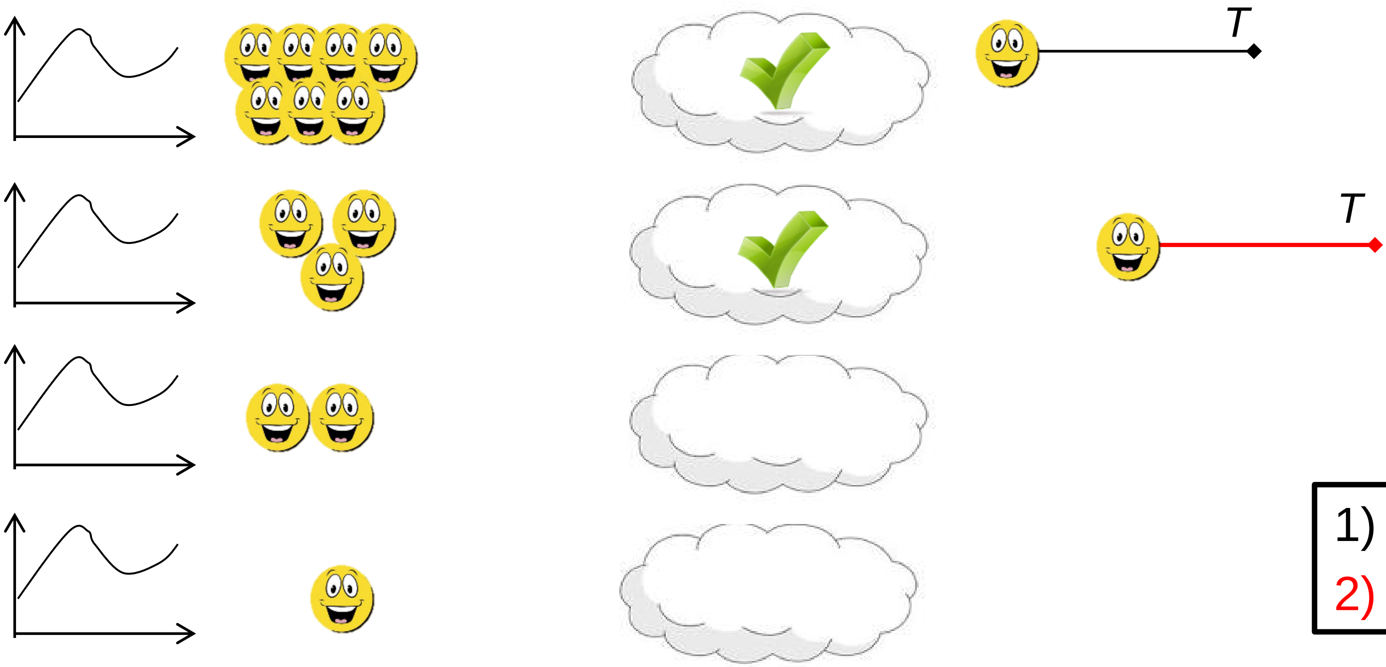
- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

Dynamic TTL-based approach



- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

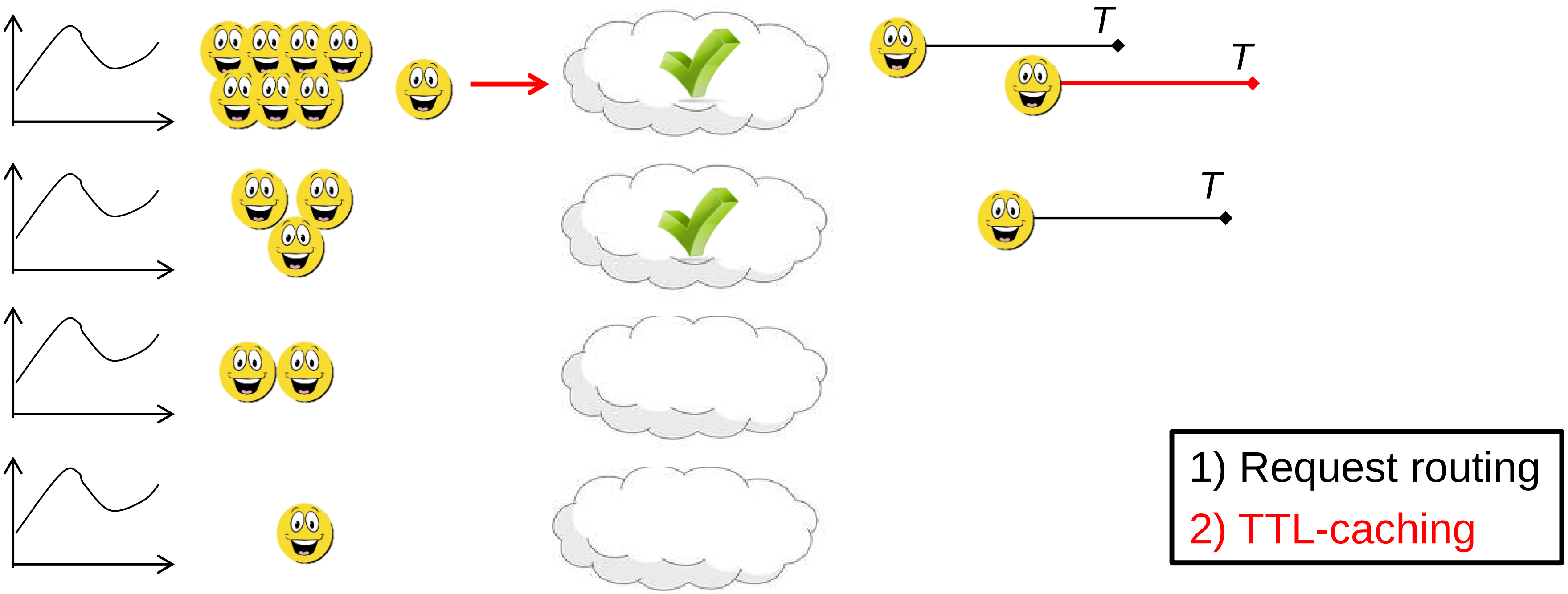
Dynamic TTL-based approach



1) Request routing
2) TTL-caching

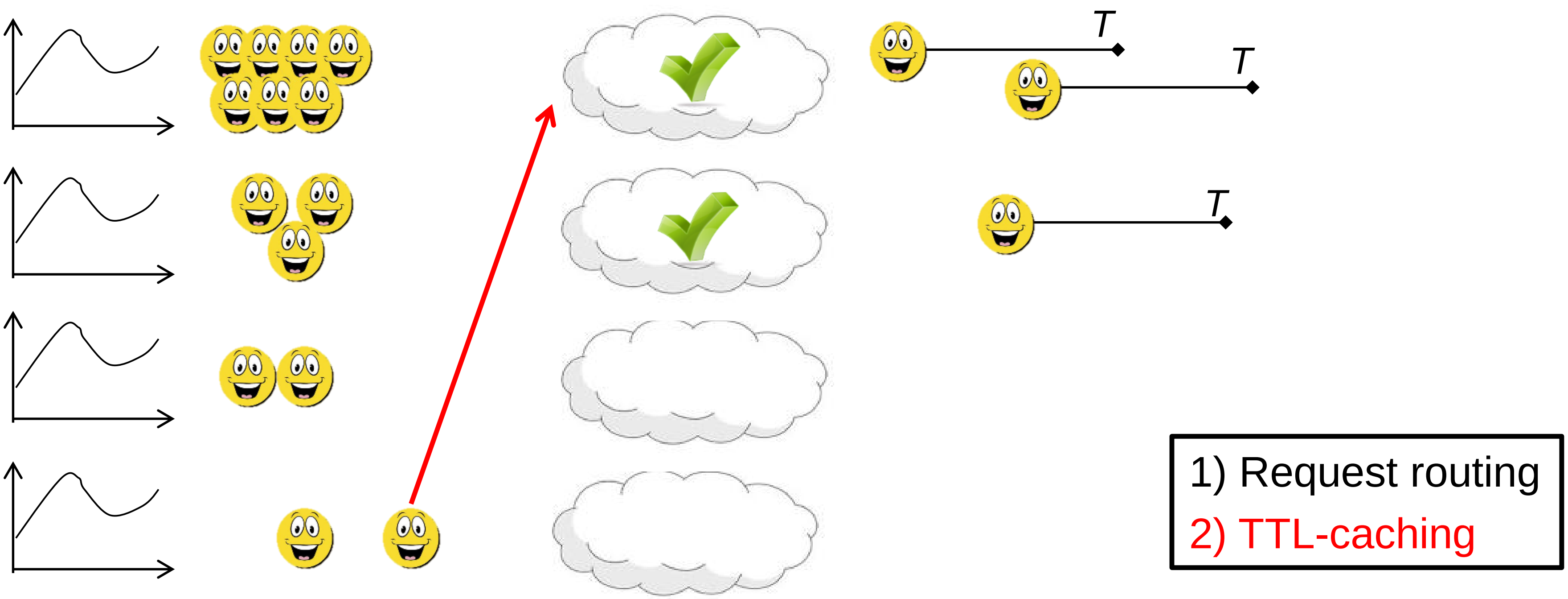
- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

Dynamic TTL-based approach



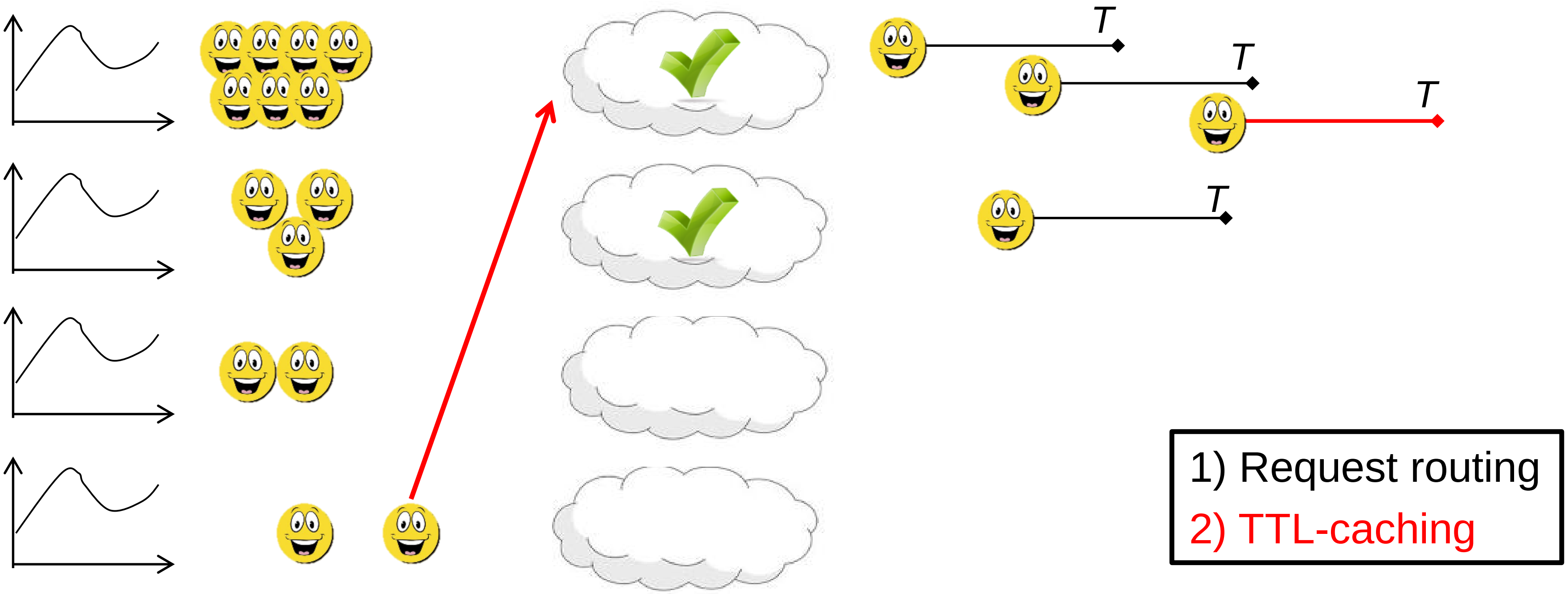
- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

Dynamic TTL-based approach



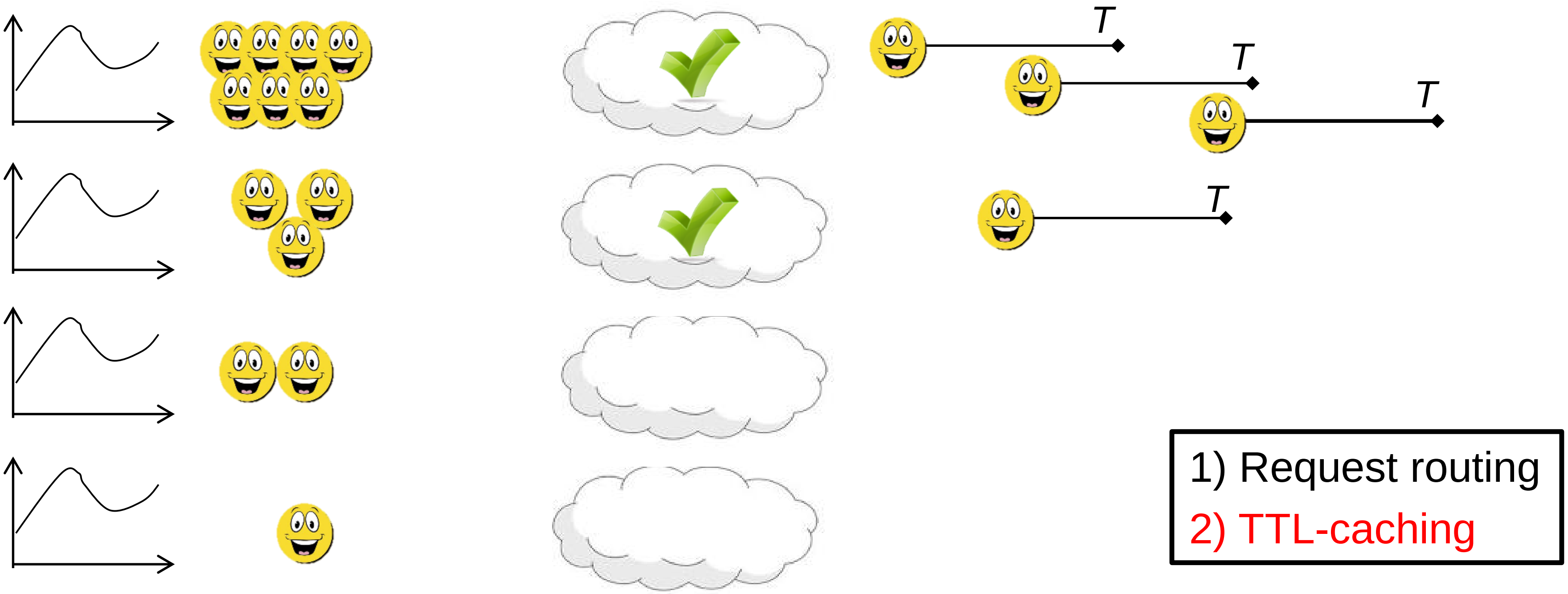
- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

Dynamic TTL-based approach



- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

Dynamic TTL-based approach



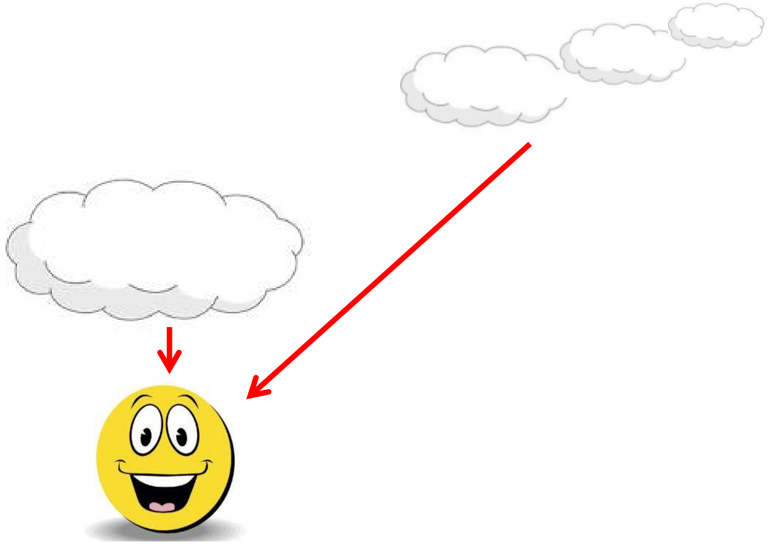
- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

Request routing optimization

Minimize

$$\sum_{i\in\mathcal{N}}\left(\gamma_i e^{-\gamma_i T} + L(1-e^{-\gamma_i T}) + R\sum_{c\in\mathcal{M}:i^*(c)\neq i}\lambda_{c,i}\right), \quad \text{where } \gamma_i = \sum_{c\in\mathcal{M}}\lambda_{c,i}$$

- Minimize content delivery costs
  - Cache miss cost
  - Cache storage cost
  - Remote routing cost



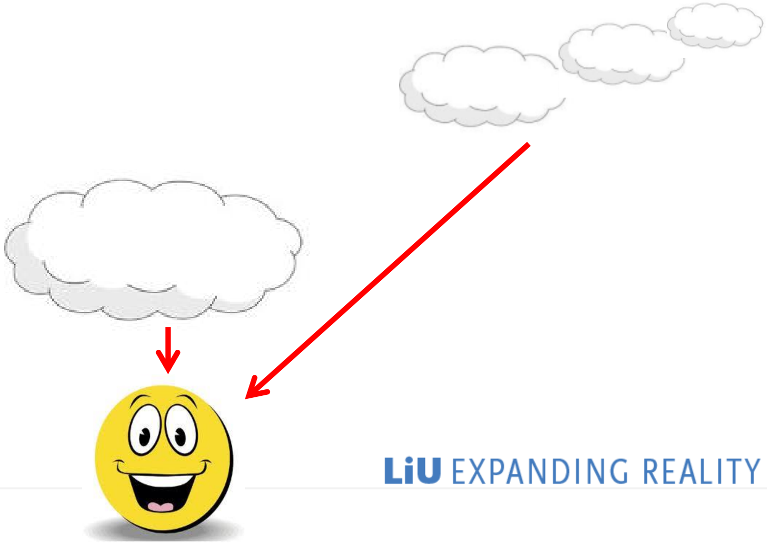
Request routing optimization

Minimize

$$\sum_{i \in N} \left(\gamma_i e^{-\gamma_i T} + L(1 - e^{-\gamma_i T}) + R \sum_{c \in M : i^*(c) \neq i} \lambda_{c,i} \right), \quad \text{where } \boxed{\gamma_i = \sum_{c \in M} \lambda_{c,i}}$$

Aggregate request rate at server location i

- Minimize content delivery costs
    - Cache miss cost
    - Cache storage cost
    - Remote routing cost

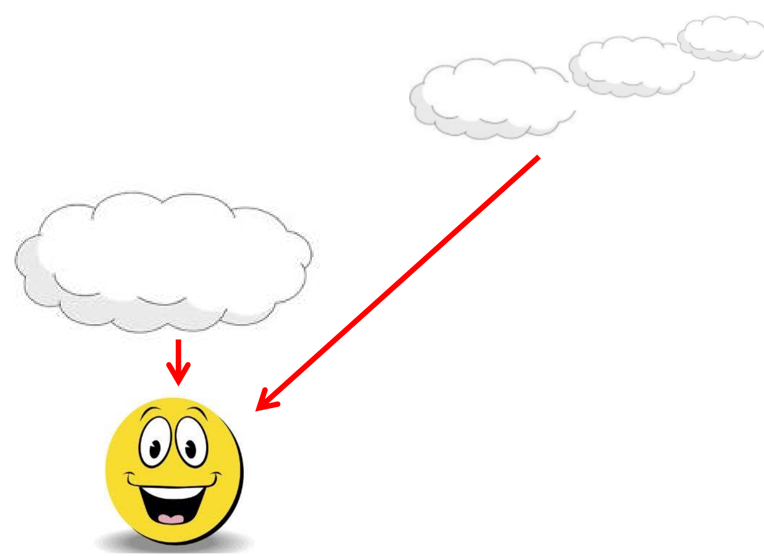


Request routing optimization

Minimize

$$\sum_{i \in N} \left(\underbrace{\boxed{\gamma_i e^{-\gamma_i T}}}_{\text{Cache miss cost}} + \underbrace{\boxed{L(1 - e^{-\gamma_i T})}}_{\text{Cache storage cost}} + \underbrace{\boxed{R \sum_{c \in M: i^*(c) \neq i} \lambda_{c,i}}}_{\text{Remote routing cost}} \right), \quad \text{where } \gamma_i = \sum_{c \in M} \lambda_{c,i}$$

- Minimize content delivery costs
  - Cache miss cost
  - Cache storage cost
  - Remote routing cost



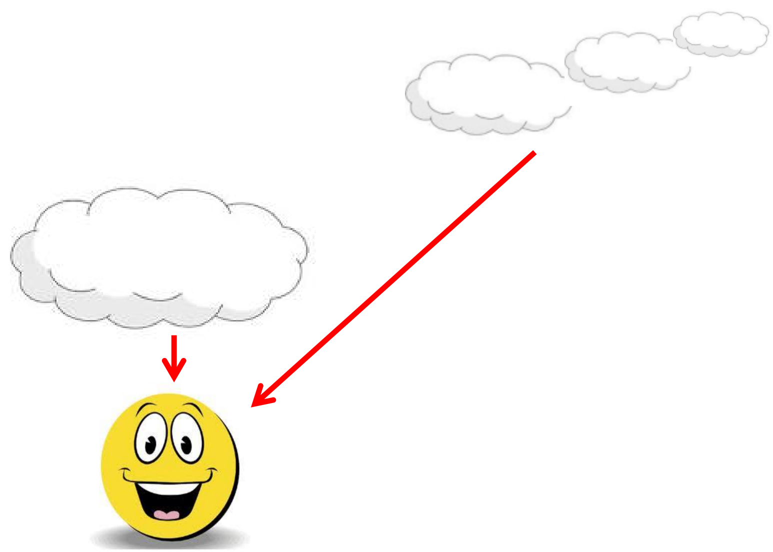
Request routing optimization

Minimize

$$\sum_{i\in \mathcal{N}}\left(\boxed{\gamma_i e^{-\gamma_i T}} + L(1-e^{-\gamma_i T}) + R\sum_{c\in \mathcal{M}: i^*(c)\neq i}\lambda_{c,i}\right), \quad \text{where } \gamma_i = \sum_{c\in \mathcal{M}}\lambda_{c,i}$$

Cache miss cost

- Minimize content delivery costs
  - Cache miss cost
  - Cache storage cost
  - Remote routing cost



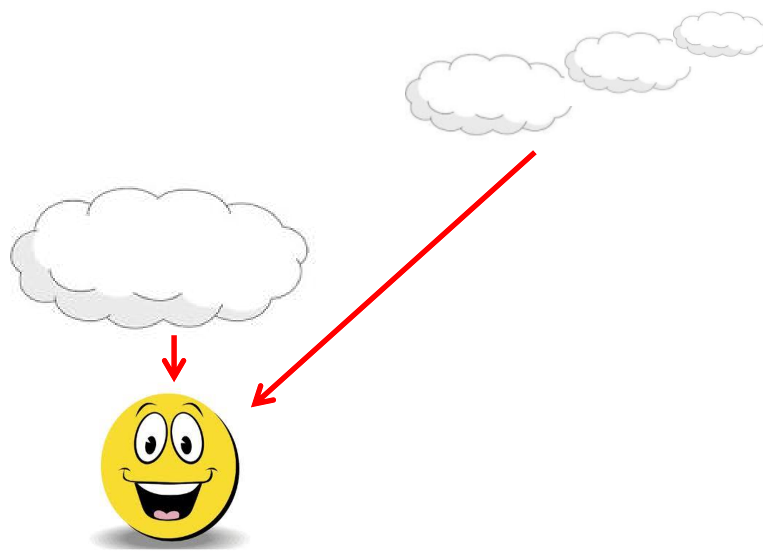
Request routing optimization

Minimize

$$\sum_{i\in \mathcal{N}}\left(\gamma_i e^{-\gamma_i T} + \boxed{L(1-e^{-\gamma_i T})} + R\sum_{c\in \mathcal{M}:i^*(c)\neq i}\lambda_{c,i}\right), \quad \text{where } \gamma_i = \sum_{c\in \mathcal{M}}\lambda_{c,i}$$

Cache storage cost

- Minimize content delivery costs
  - Cache miss cost
  - Cache storage cost
  - Remote routing cost



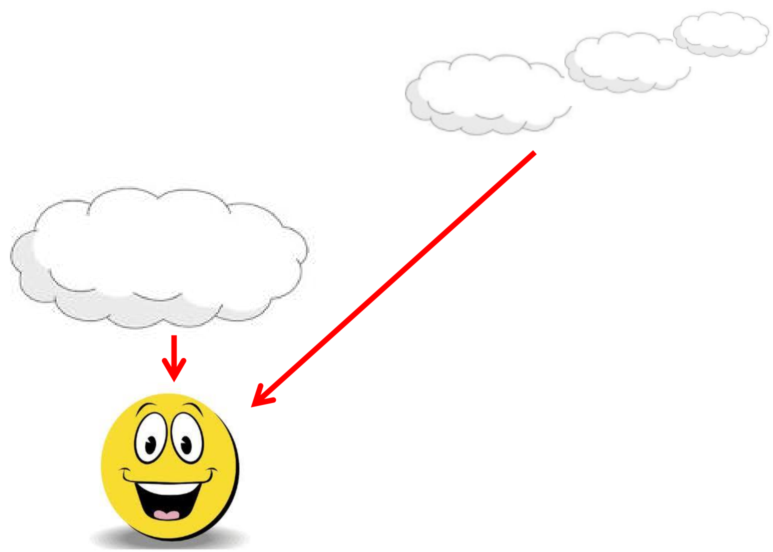
Request routing optimization

Minimize

$$\sum_{i\in \mathcal{N}}\left(\gamma_i e^{-\gamma_i T} + L(1-e^{-\gamma_i T}) + \boxed{R\sum_{c\in \mathcal{M}: i^*(c)\neq i}\lambda_{c,i}}\right), \quad \text{where } \gamma_i = \sum_{c\in \mathcal{M}}\lambda_{c,i}$$

Remote routing cost

- Minimize content delivery costs
  - Cache miss cost
  - Cache storage cost
  - Remote routing cost



Request routing optimization

Minimize

$$\sum_{i\in N}\left(\boxed{\gamma_i e^{-\gamma_i T}} + \boxed{L(1-e^{-\gamma_i T})} + \boxed{R\sum_{c\in M:i^*(c)\neq i}\lambda_{c,i}}\right), \quad \text{where } \boxed{\gamma_i = \sum_{c\in M}\lambda_{c,i}}$$

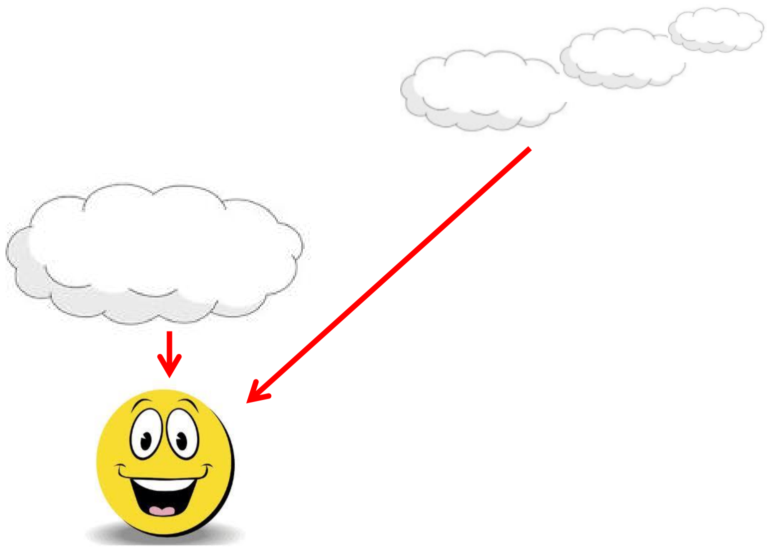
Cache miss cost

Cache storage cost

Remote routing cost

Aggregate request rate at server location i

- Minimize content delivery costs
  - Cache miss cost
  - Cache storage cost
  - Remote routing cost



Request routing optimization

Minimize

$$\sum_{i\in\mathcal{N}} \left(\gamma_i e^{-\gamma_i T} + L(1-e^{-\gamma_i T}) + R\sum_{c\in\mathcal{M}:i^*(c)\neq i}\lambda_{c,i}\right), \quad \text{where } \gamma_i = \sum_{c\in\mathcal{M}}\lambda_{c,i}$$

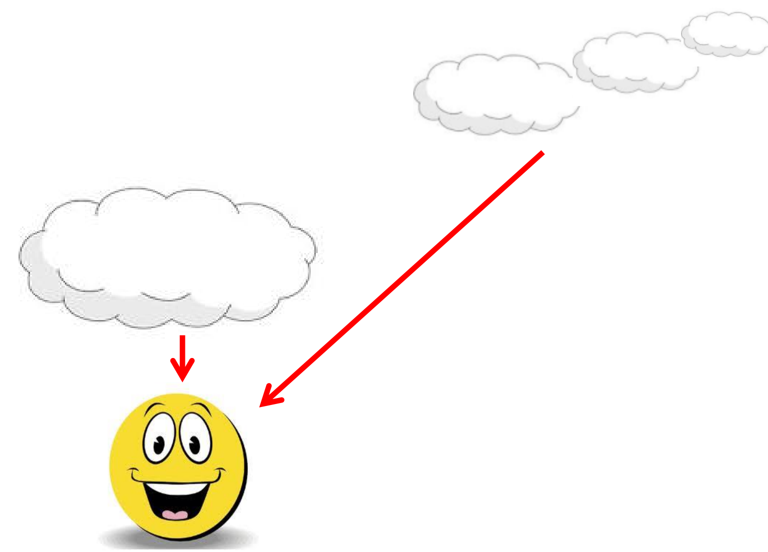
Subject to

$$\sum_{i\in\mathcal{N}}\lambda_{c,i} = \lambda_c, \quad \forall c\in\mathcal{M}$$

$$\lambda_{c,i} \geq 0, \quad \forall i\in\mathcal{N}, \forall c\in\mathcal{M}$$

Conservation constraints

- Minimize content delivery costs
  - Cache miss cost
  - Cache storage cost
  - Remote routing cost



Request routing optimization

Minimize

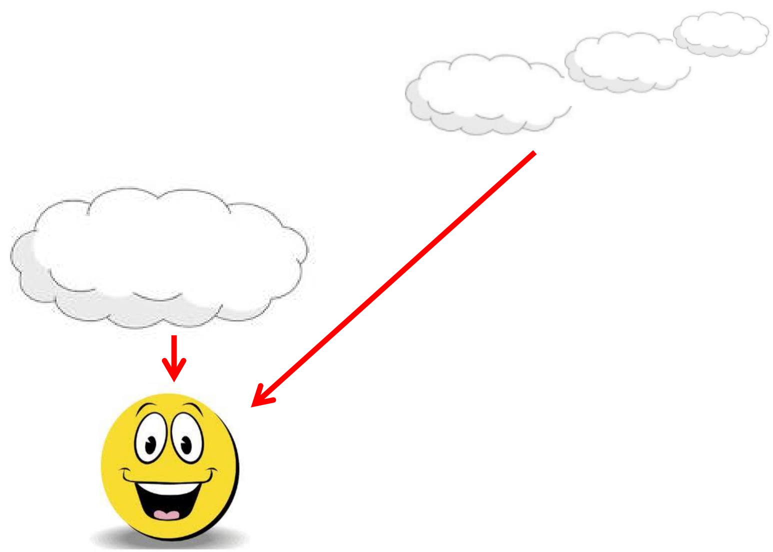
$$\sum_{i \in \mathcal{N}} \left(\gamma_i e^{-\gamma_i T} + L(1 - e^{-\gamma_i T}) + R \sum_{c \in \mathcal{M}: i^*(c) \neq i} \lambda_{c,i} \right), \quad \text{where } \gamma_i = \sum_{c \in \mathcal{M}} \lambda_{c,i}$$

Subject to

$$\sum_{i \in \mathcal{N}} \lambda_{c,i} = \lambda_c, \quad \forall c \in \mathcal{M}$$

$$\lambda_{c,i} \geq 0, \quad \forall i \in \mathcal{N}, \forall c \in \mathcal{M}$$

- Minimize content delivery costs
  - Cache miss cost
  - Cache storage cost
  - Remote routing cost



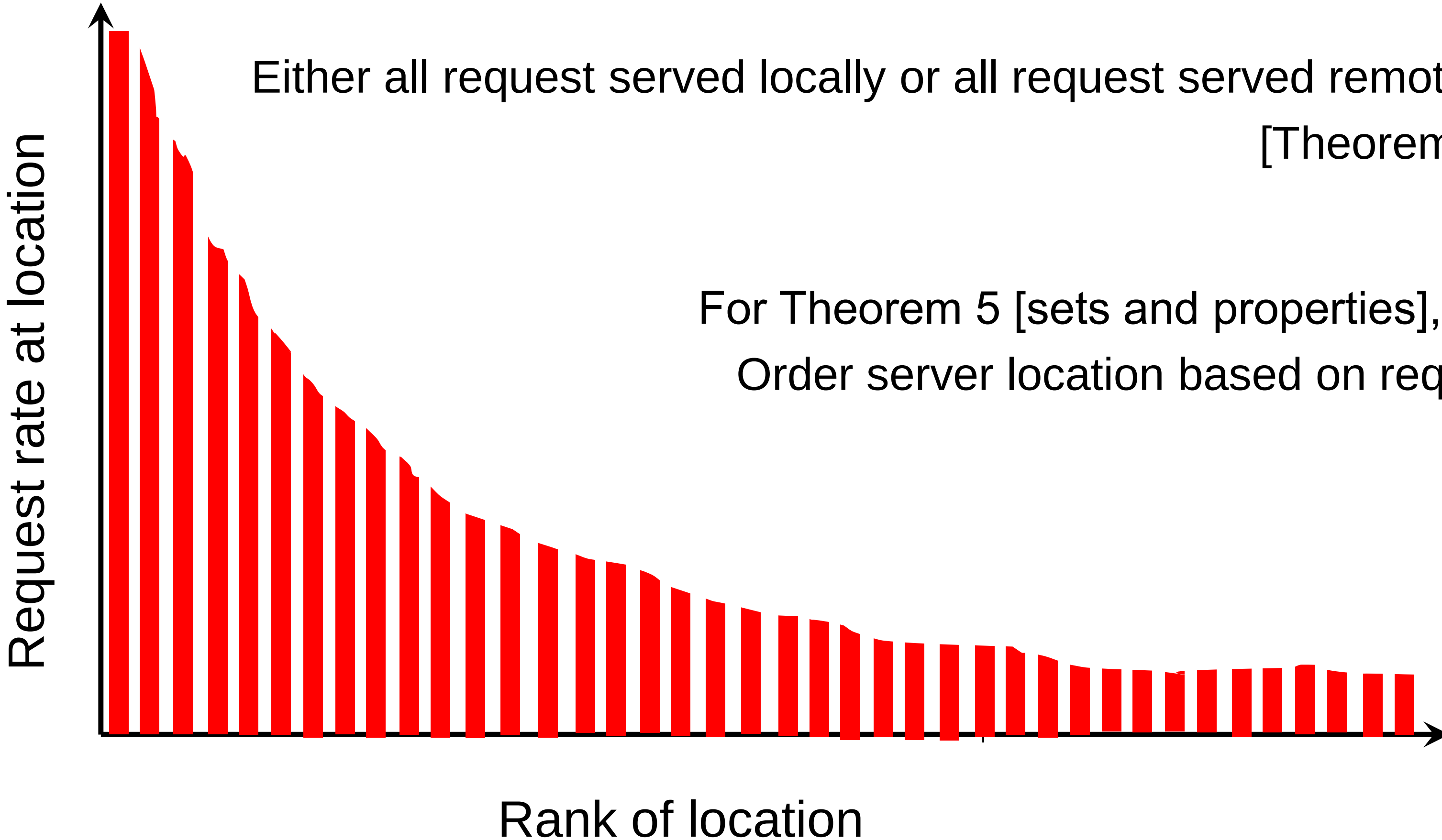
Properties of optimal request routing

Either all request served locally or all request served remotely

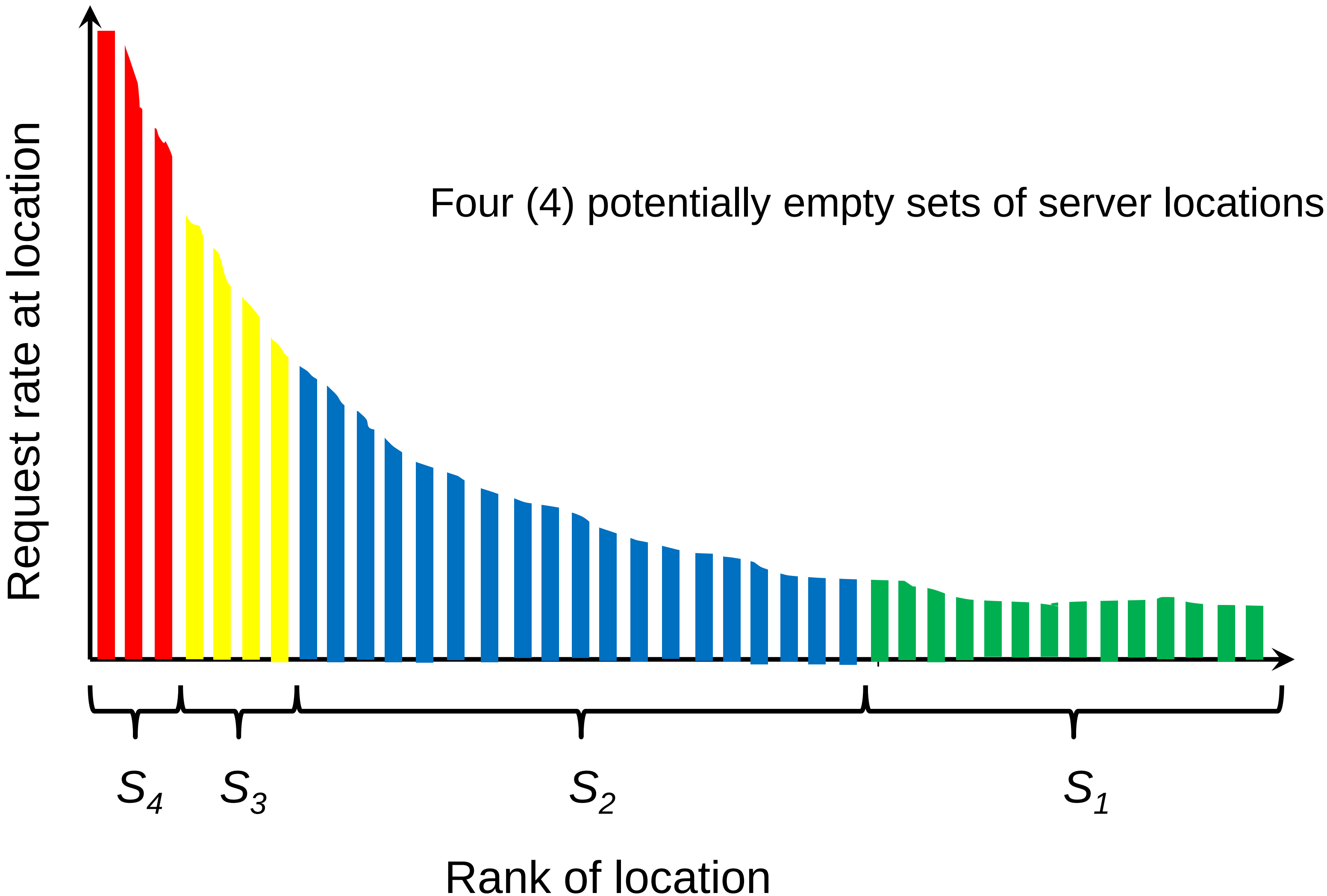
[Theorem 4]

For Theorem 5 [sets and properties], first …

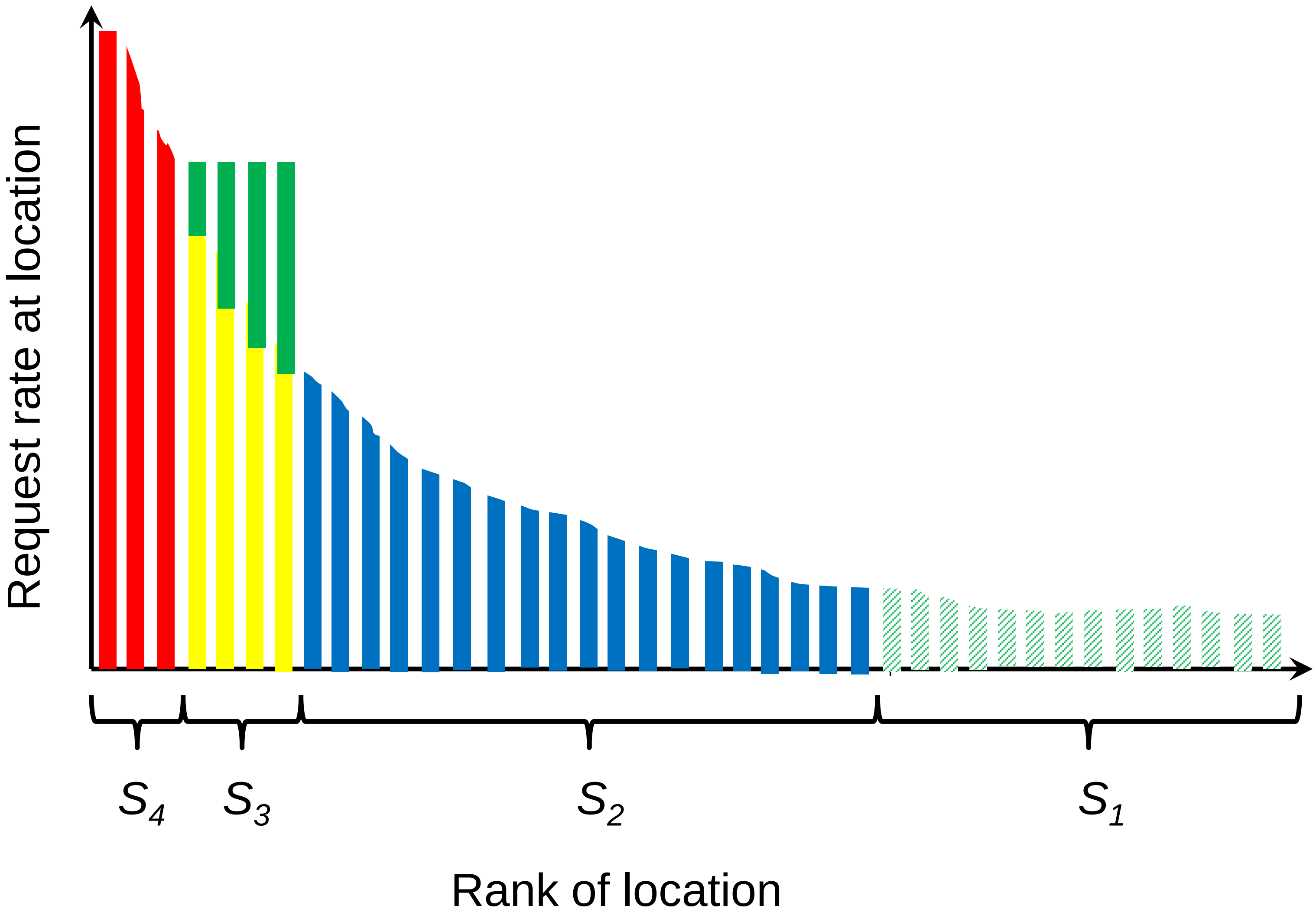
Order server location based on request rate



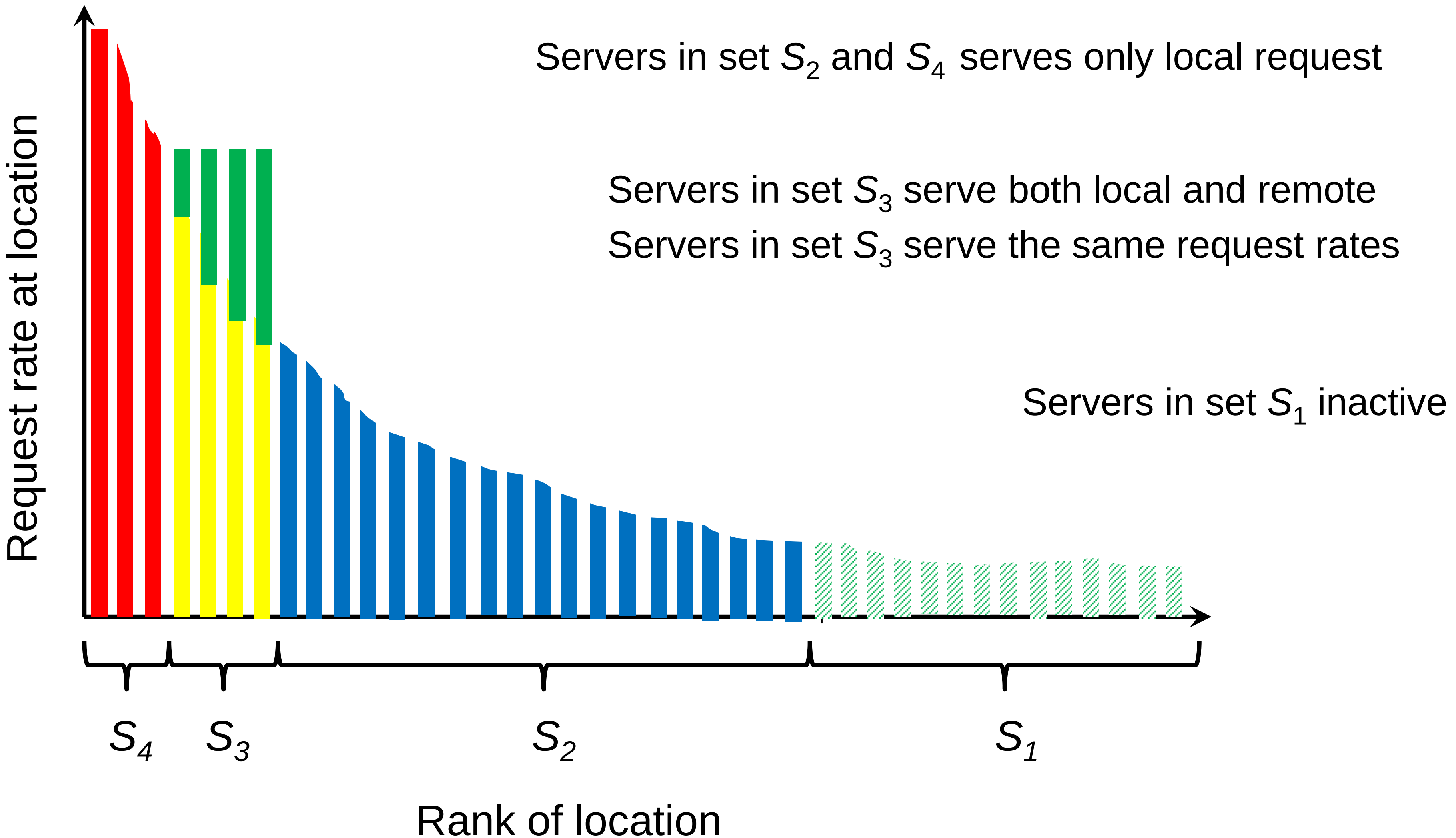
Properties of optimal request routing



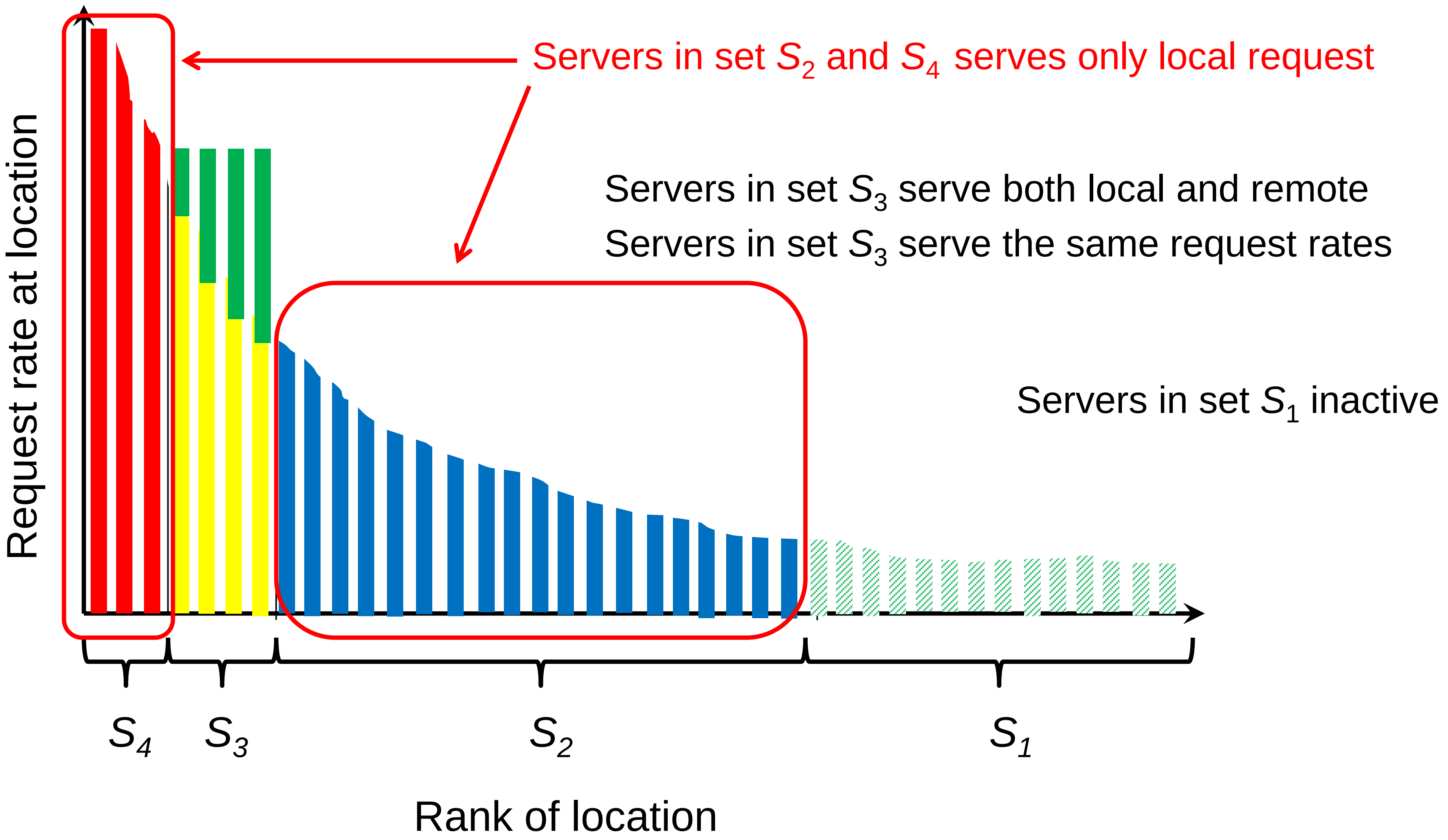
Properties of optimal request routing



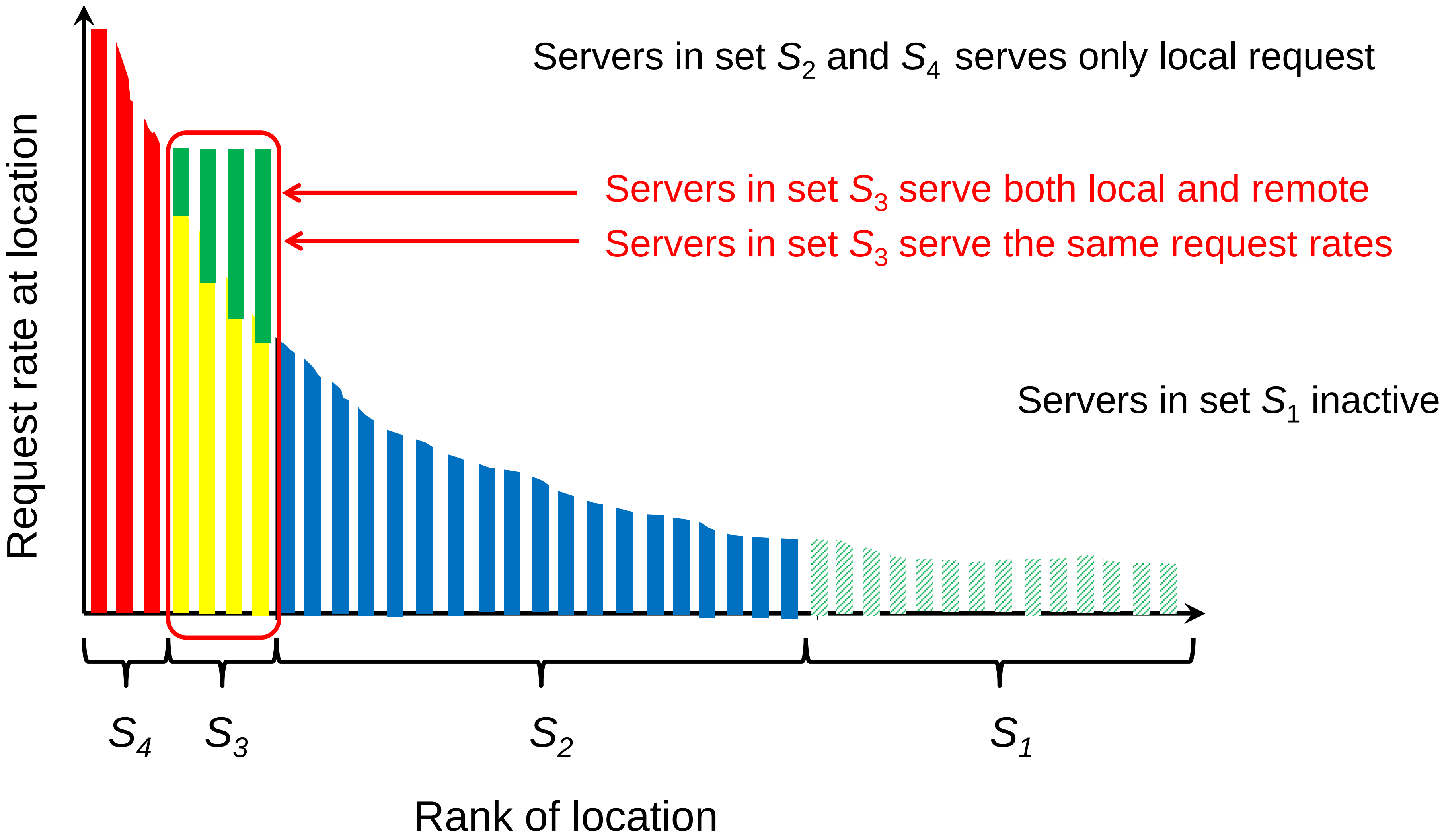
Properties of optimal request routing



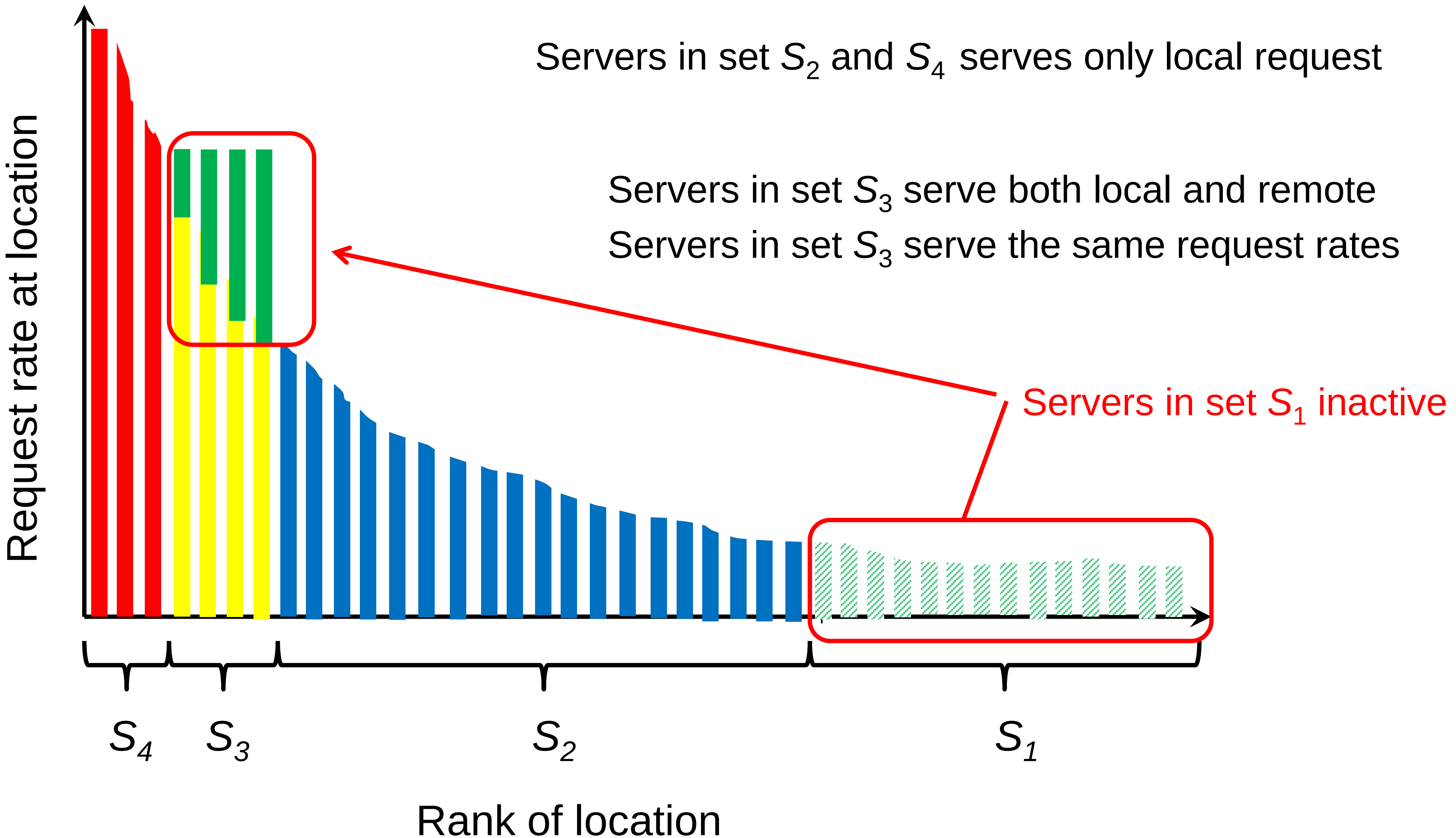
Properties of optimal request routing



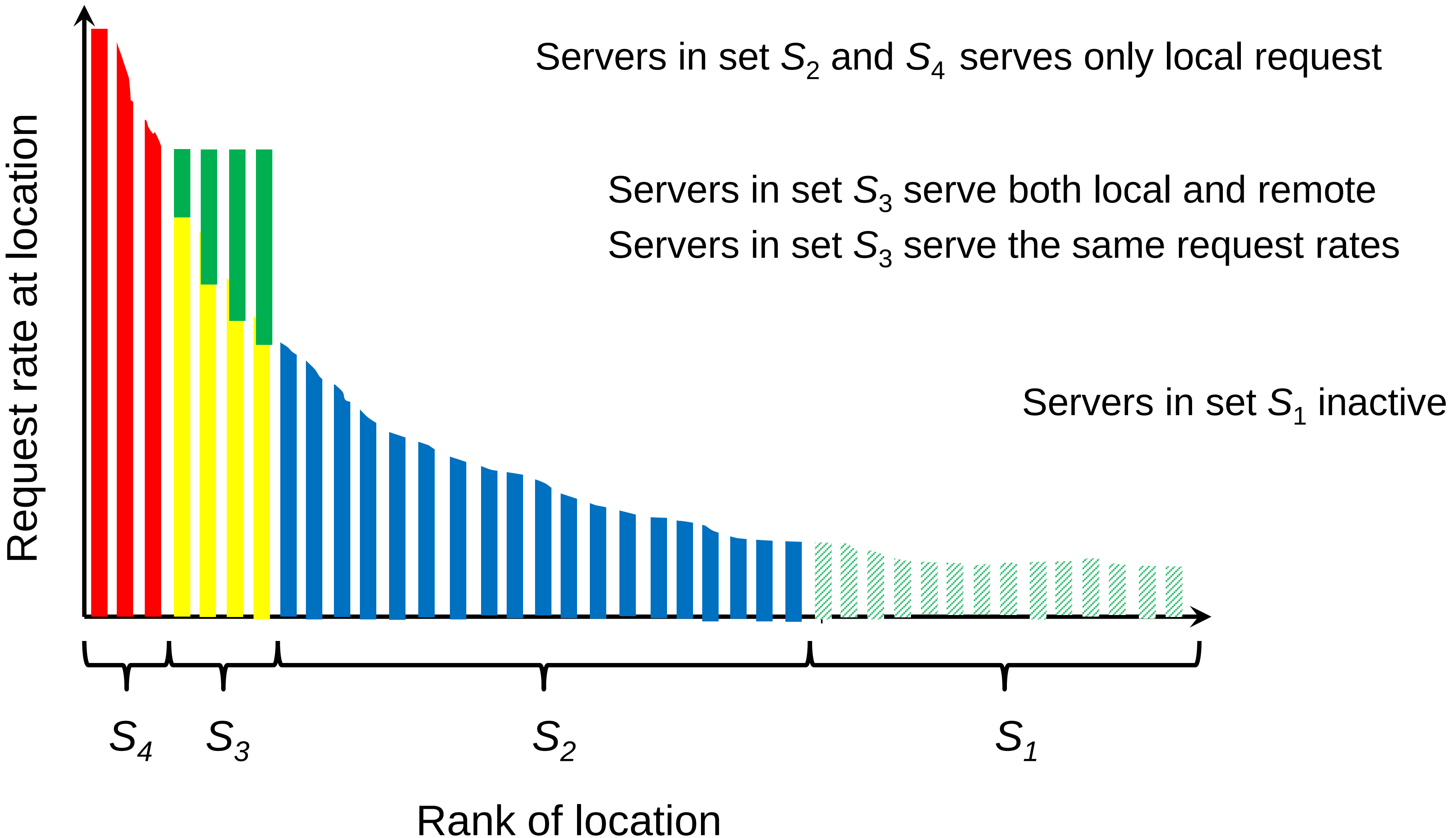
Properties of optimal request routing



Properties of optimal request routing

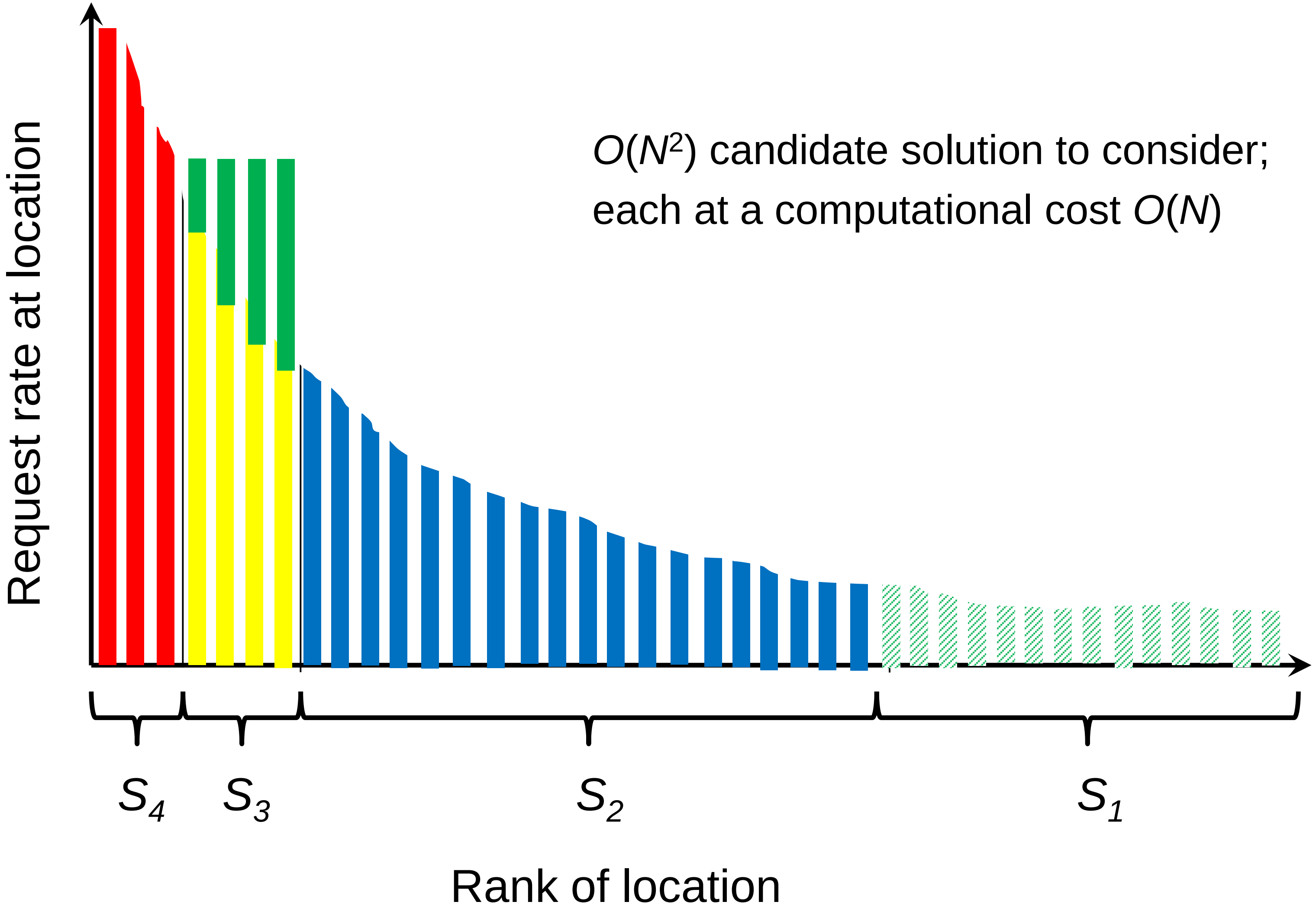


Properties of optimal request routing

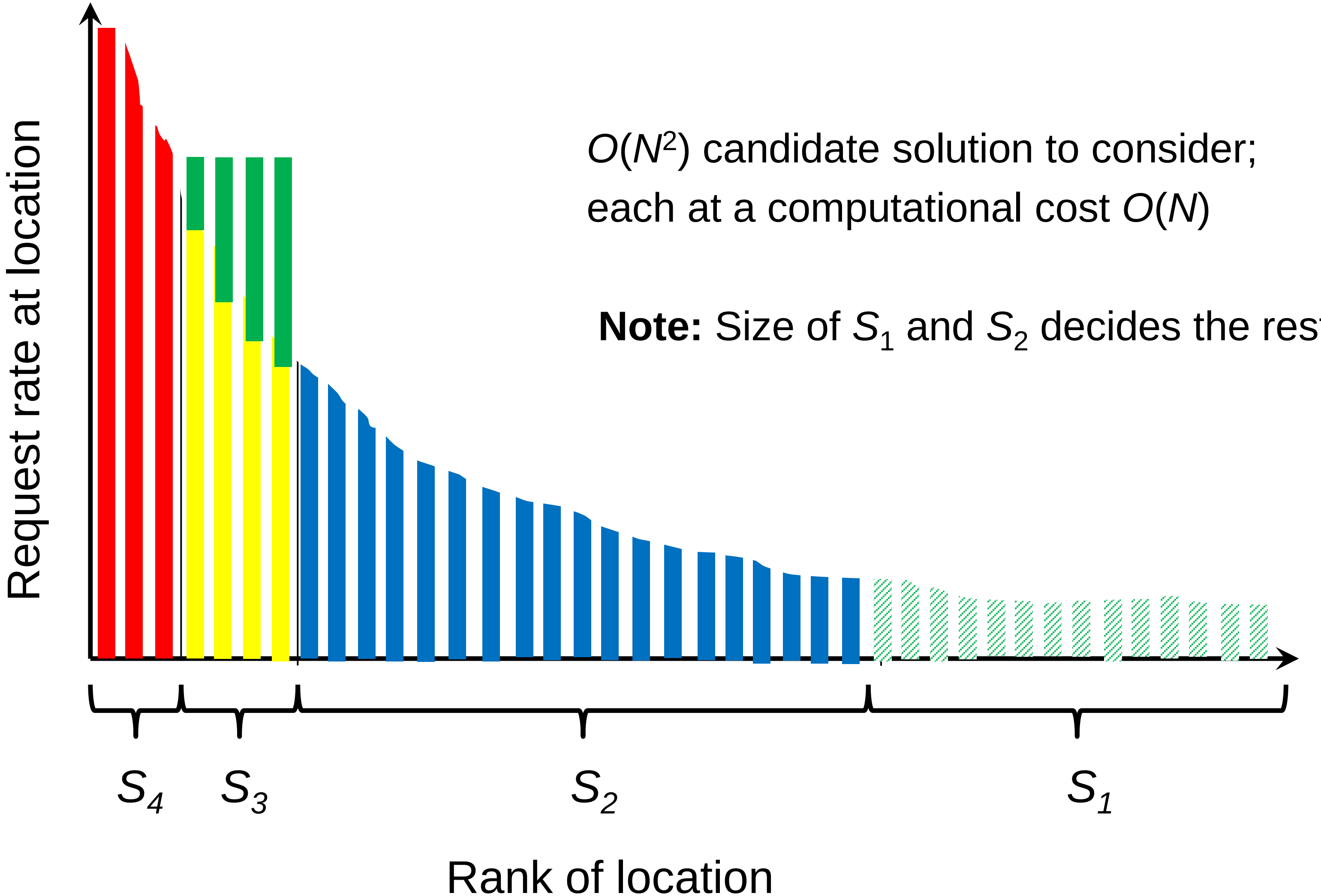


Finding the optimal request routing

$O(N^2)$ candidate solution to consider;
each at a computational cost $O(N)$



Finding the optimal request routing

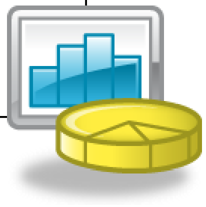


$O(N^2)$ candidate solution to consider;
each at a computational cost $O(N)$

Note: Size of S_1 and S_2 decides the rest

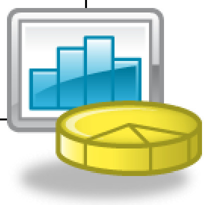
Cost Comparison

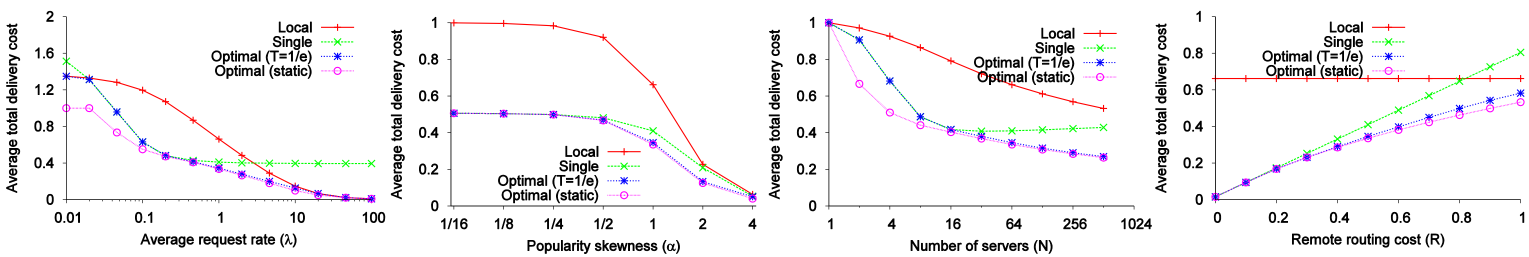
- Compare optimal dynamic policy with baselines
  - Always “local” server
  - Always “single” server
- As well as with optimal “static” placement (any T_i)



Cost Comparison

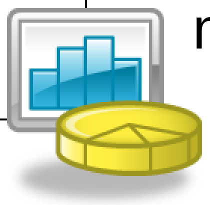
- Compare optimal dynamic policy with baselines
  - Always “local” server
  - Always “single” server
- As well as with optimal “static” placement (any T_i)



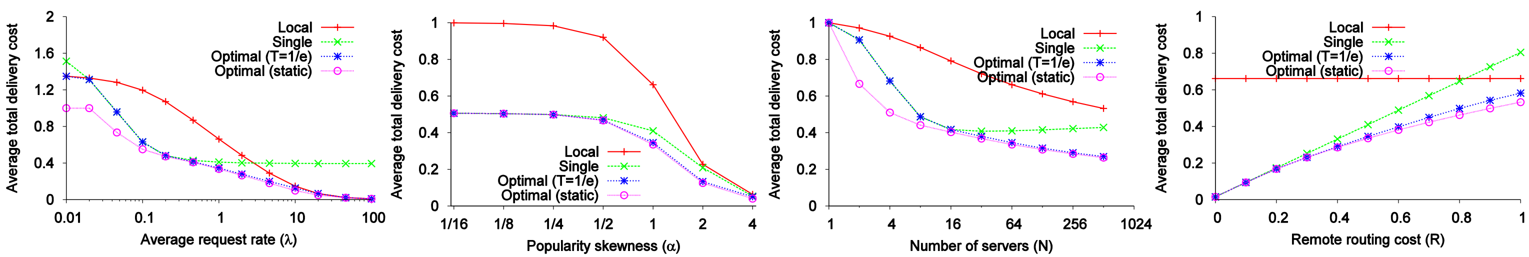


Cost Comparison

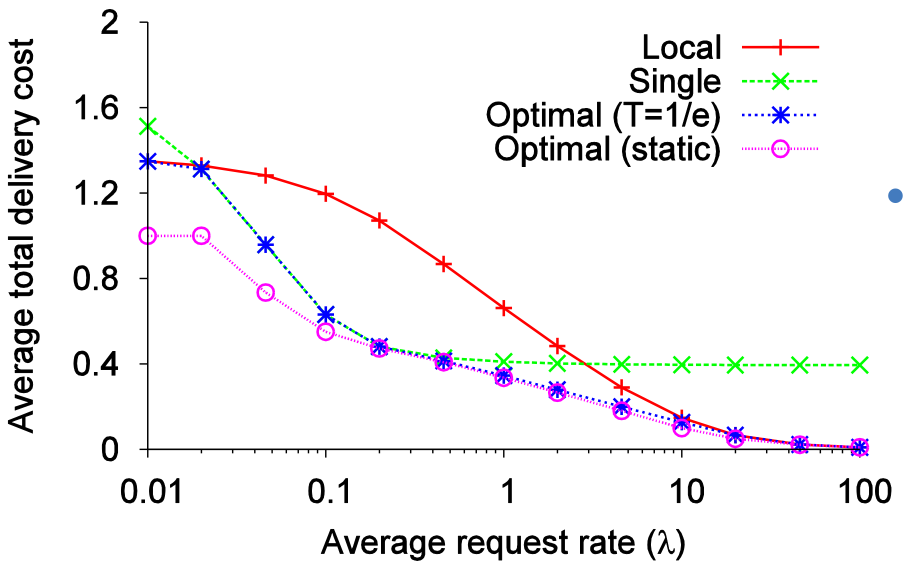
- Compare optimal dynamic policy with baselines
  - Always “local” server
  - Always “single” server
- As well as with optimal “static” placement (any T_i)



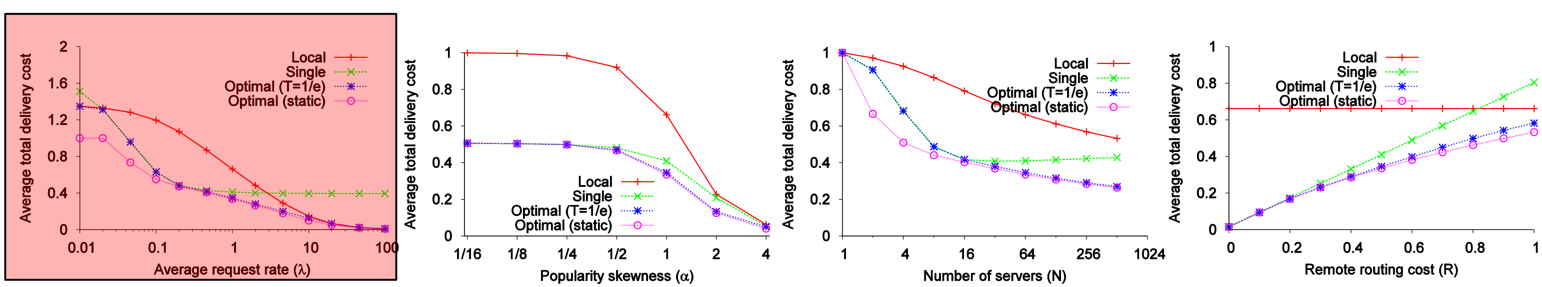
- Significantly outperform baselines (“local” and “single”)
  - Difference can be unbounded
- Even with static load, costs typically close to those with static optimal placement (but much more flexible)



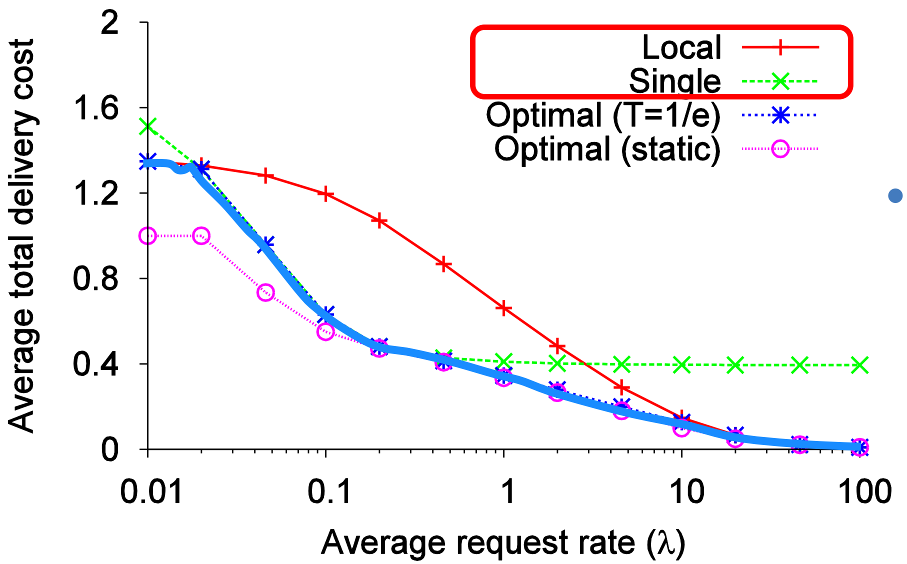
Cost Comparison



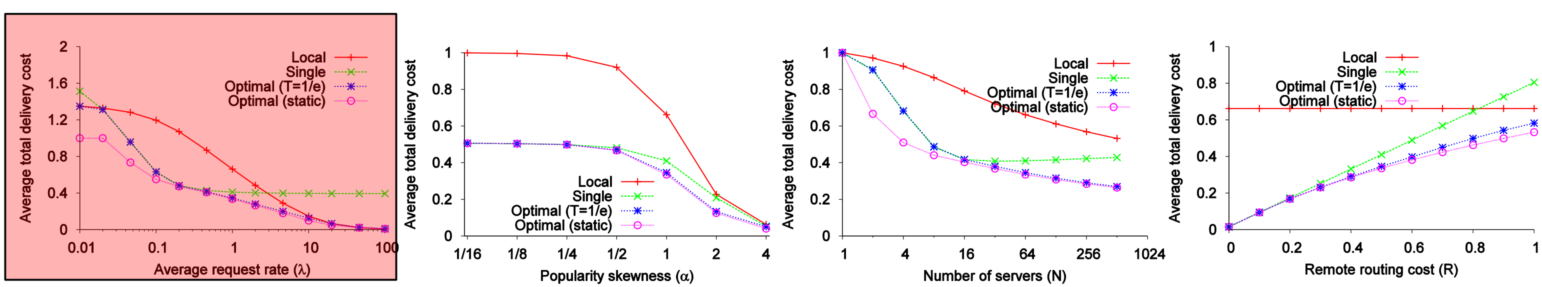
- Significantly outperform baselines (“local” and “single”)
  - Difference can be unbounded
- Even with static load, costs typically close to those with static optimal placement (but much more flexible)



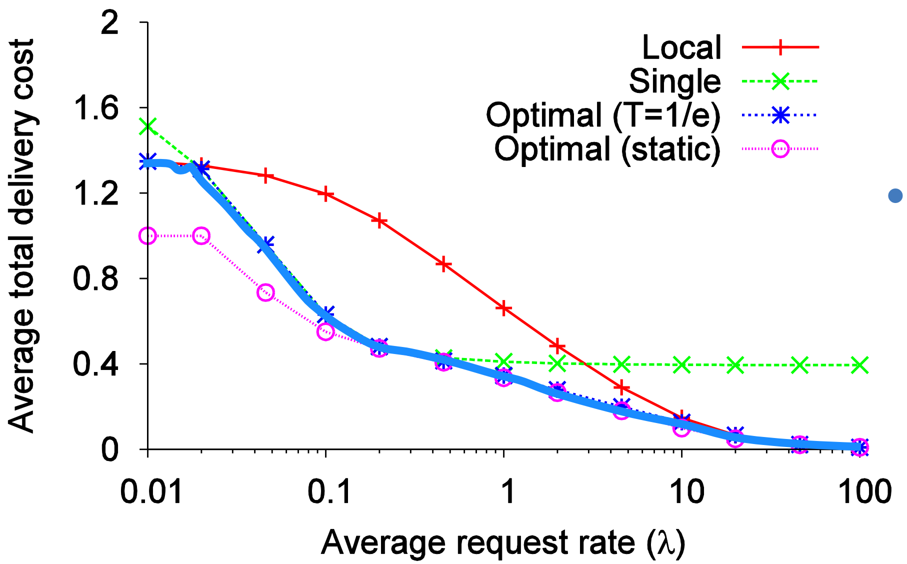
Cost Comparison



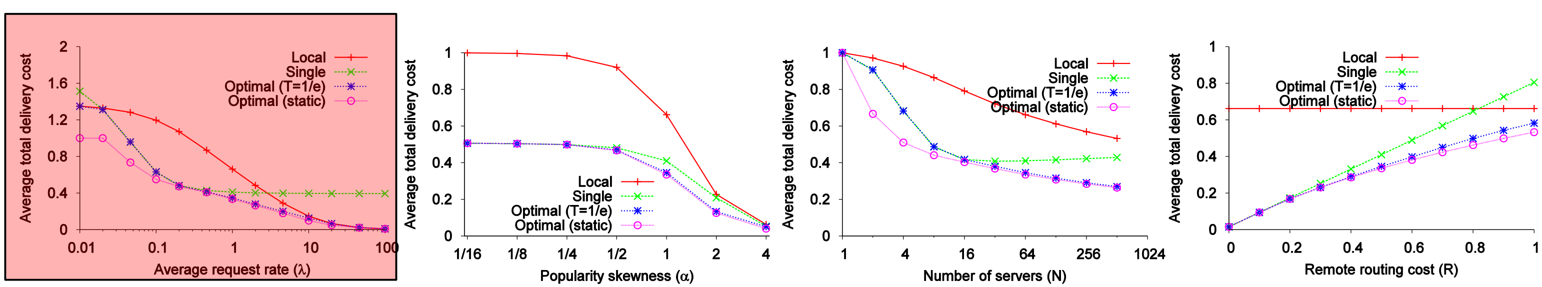
- Significantly outperform baselines (“local” and “single”)
  - Difference can be unbounded
- Even with static load, costs typically close to those with static optimal placement (but much more flexible)



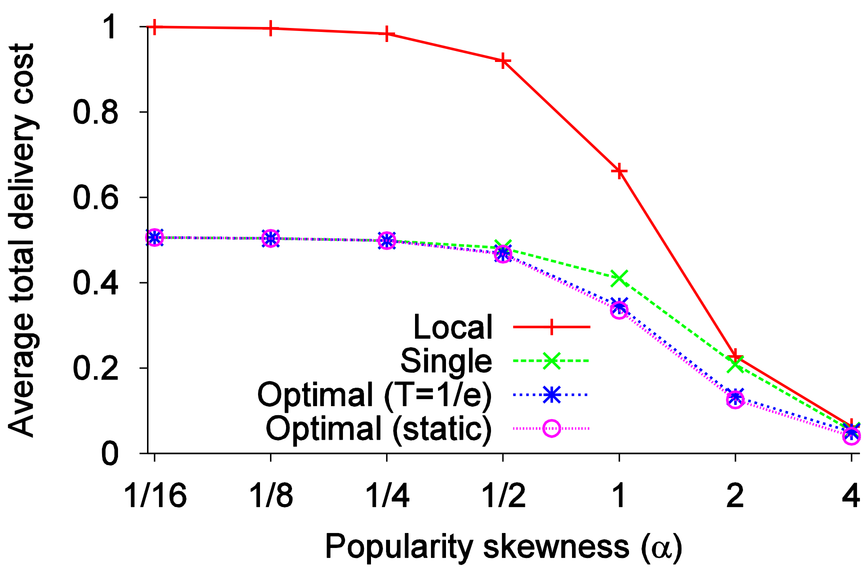
Cost Comparison



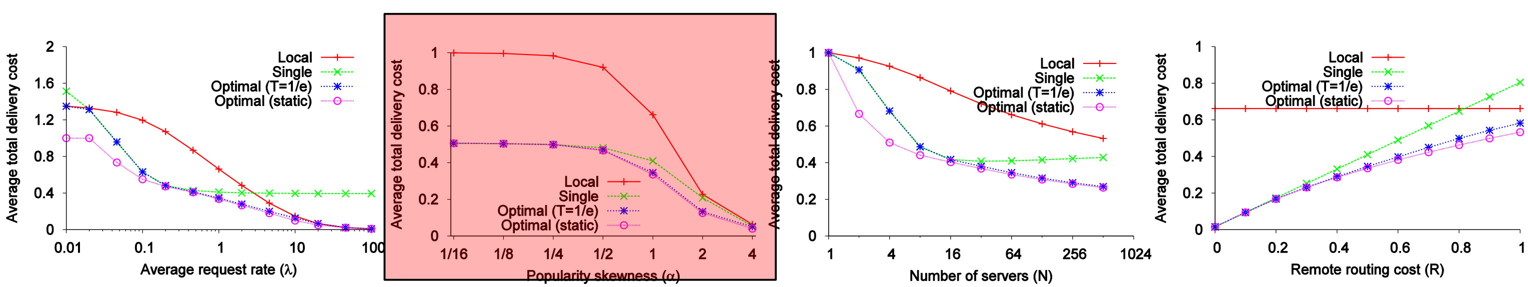
- Significantly outperform baselines (“local” and “single”)
  - Difference can be unbounded
- Even with static load, costs typically close to those with static optimal placement (but much more flexible)



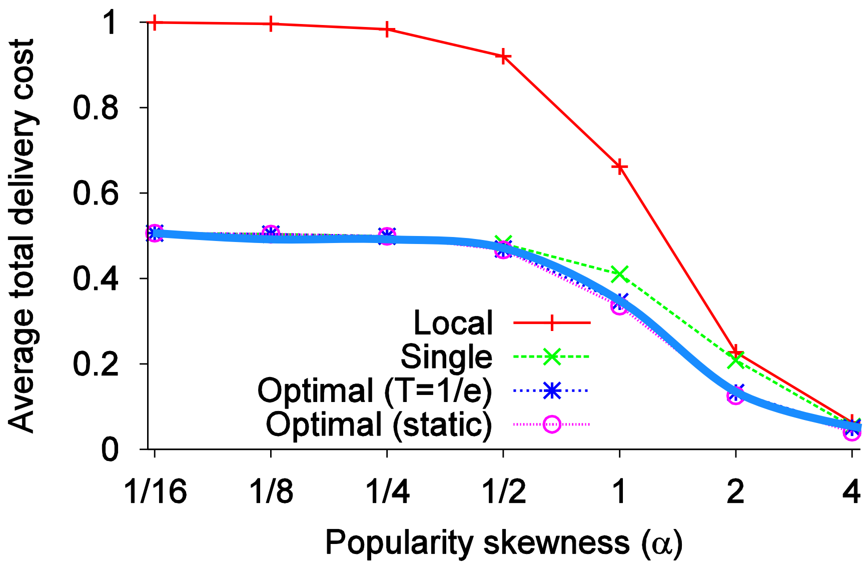
Cost Comparison



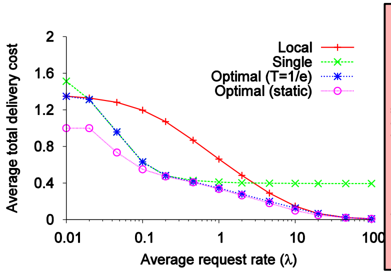
- Significantly outperform baselines (“local” and “single”)
  - Difference can be unbounded
- Even with static load, costs typically close to those with static optimal placement (but much more flexible)

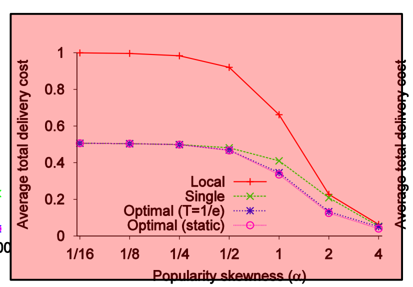


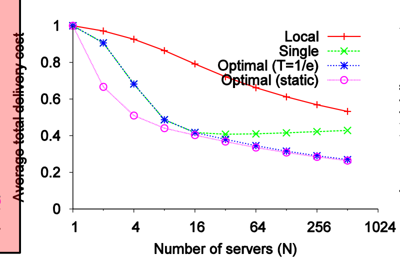
Cost Comparison

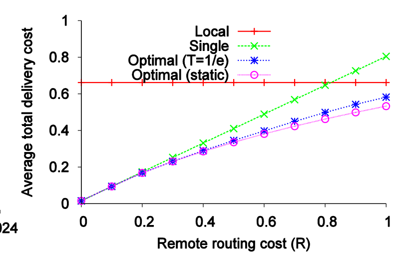


- Significantly outperform baselines (“local” and “single”)
  - Difference can be unbounded
- Even with static load, costs typically close to those with static optimal placement (but much more flexible)

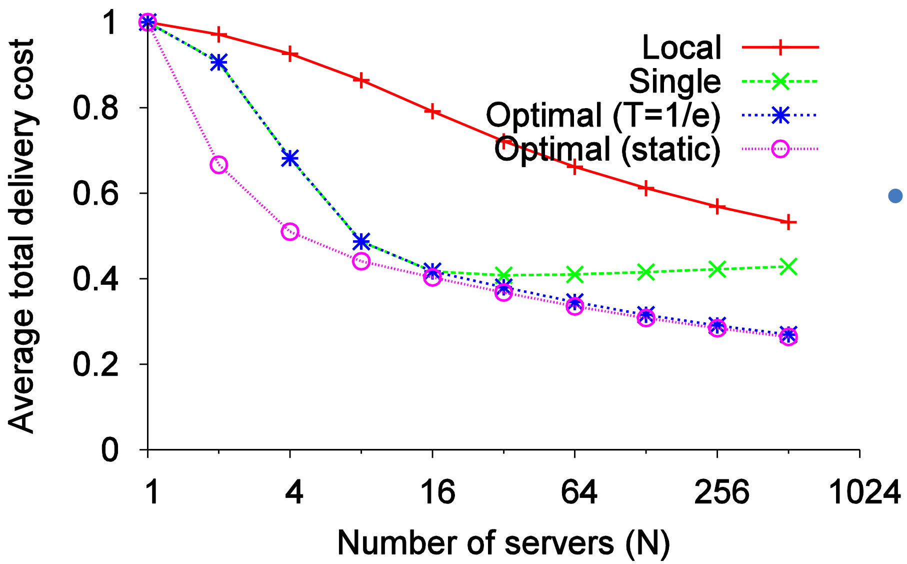




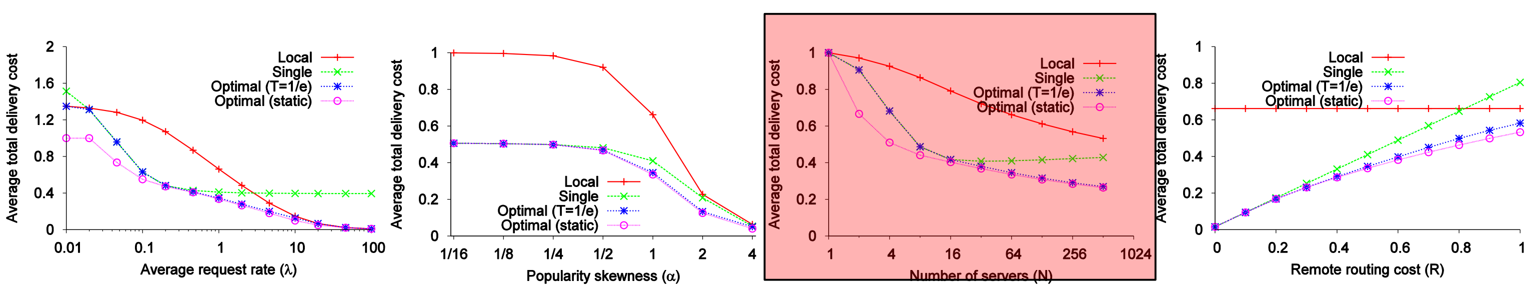




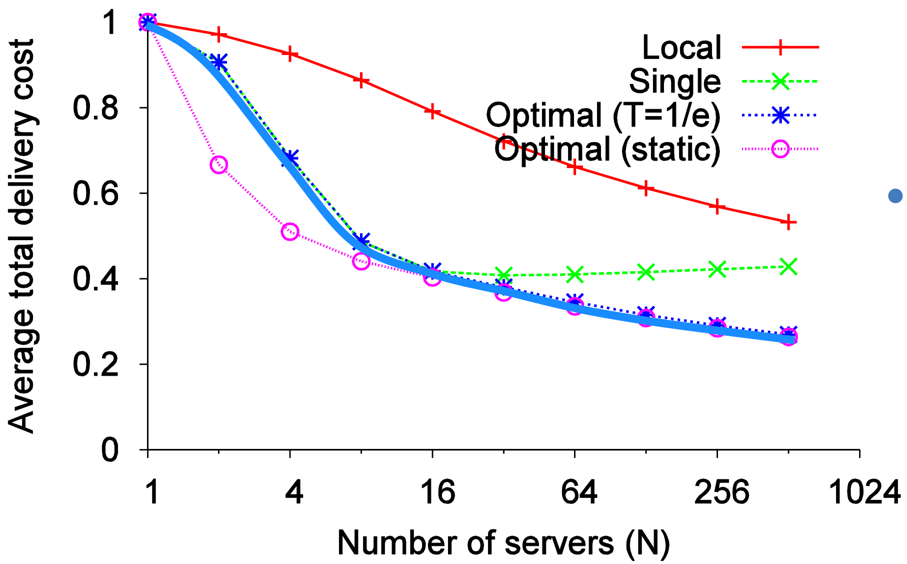
Cost Comparison



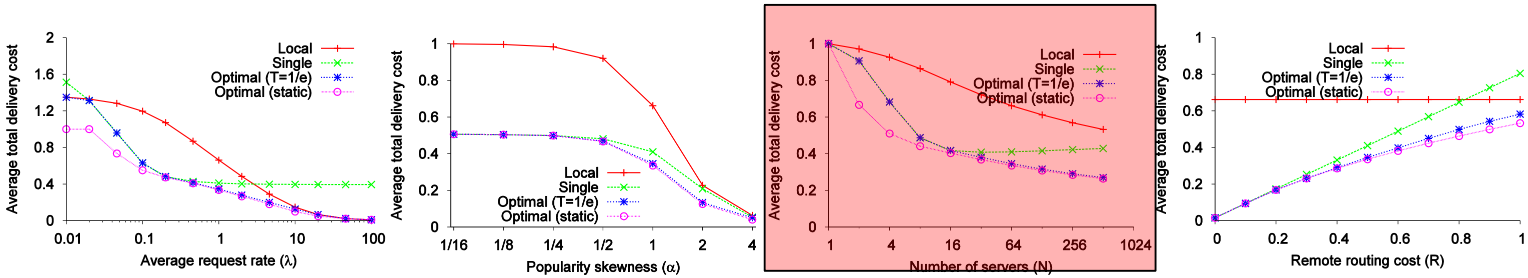
- Significantly outperform baselines (“local” and “single”)
  - Difference can be unbounded
- Even with static load, costs typically close to those with static optimal placement (but much more flexible)



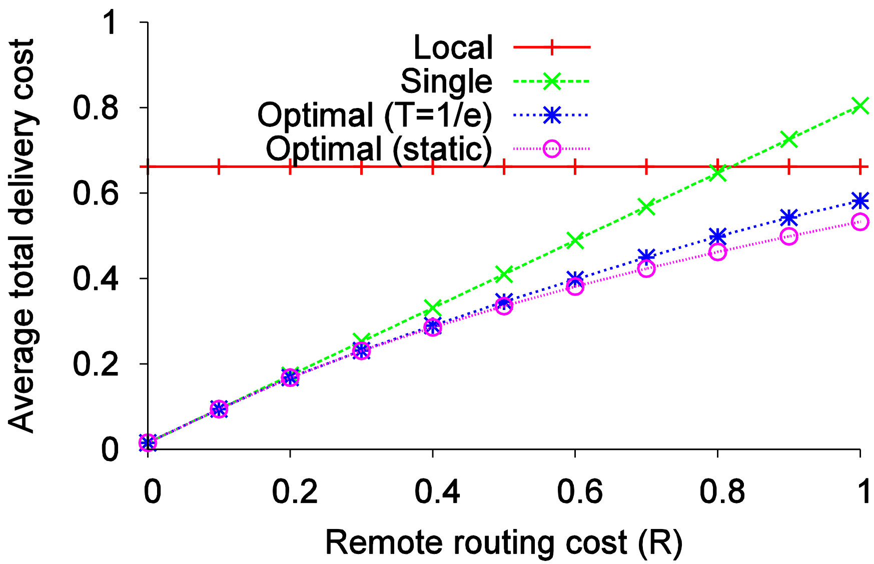
Cost Comparison



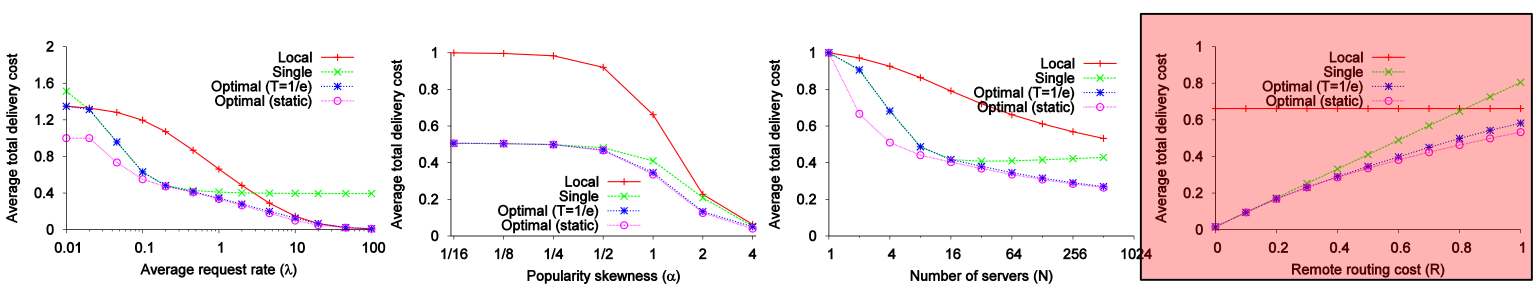
- Significantly outperform baselines (“local” and “single”)
  - Difference can be unbounded
- Even with static load, costs typically close to those with static optimal placement (but much more flexible)



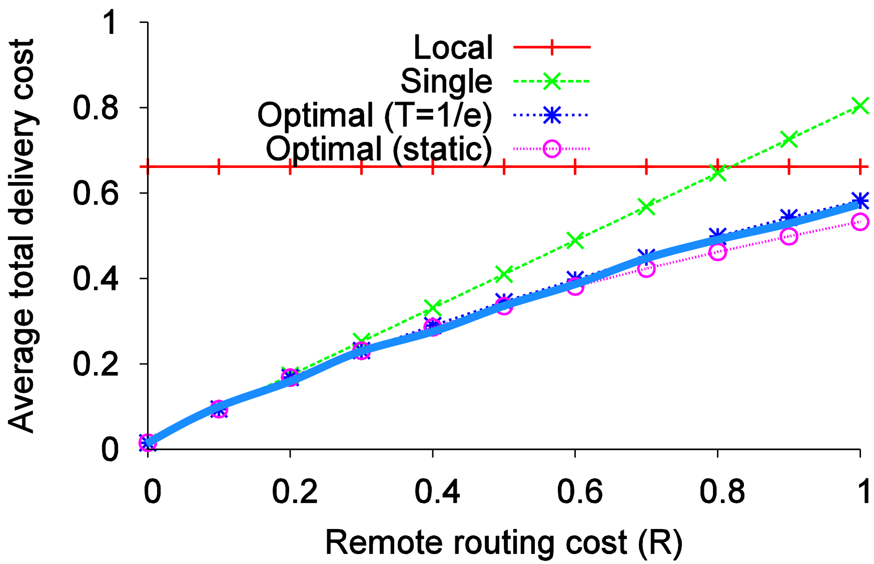
Cost Comparison



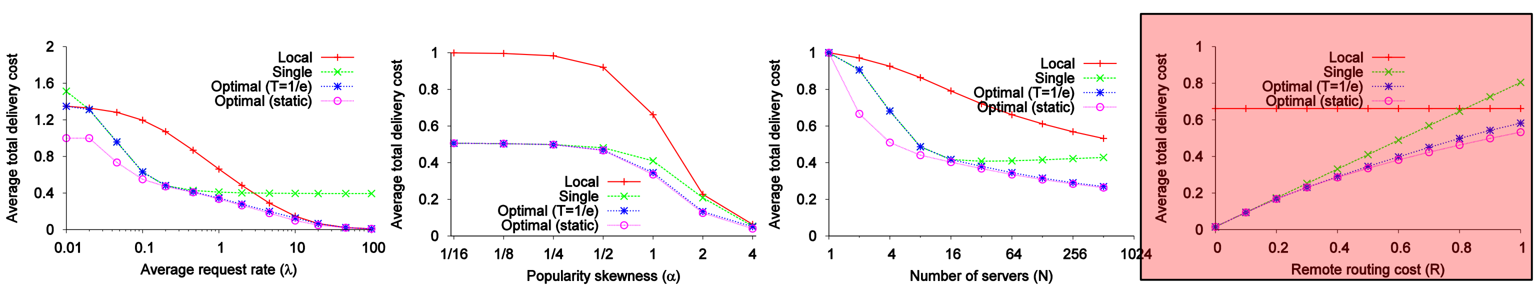
- Significantly outperform baselines ("local" and "single")
  - Difference can be unbounded
- Even with static load, costs typically close to those with static optimal placement (but much more flexible)



Cost Comparison

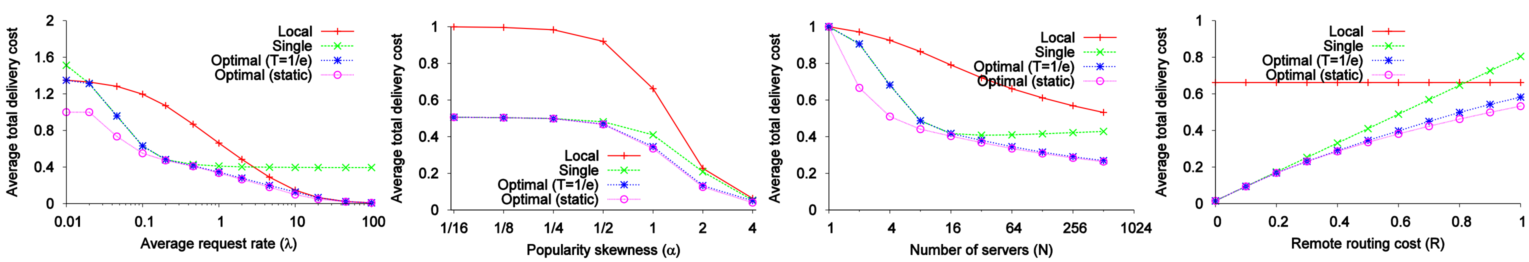


- Significantly outperform baselines ("local" and "single")
  - Difference can be unbounded
- Even with static load, costs typically close to those with static optimal placement (but much more flexible)



Cost Comparison

- Significantly outperform baselines (“local” and “single”)
  - Difference can be unbounded
- Even with static load, costs typically close to those with static optimal placement (but much more flexible)



Contributions

- Propose new delivery approach using distributed clouds
  - Request routing periodically updated
  - Cache content updated dynamically
- Formulate optimization problem
  - Non-convex, so standard techniques not directly applicable
- Identify and prove properties of optimal solution
  - Leverage properties to find optimal solution
- Comparison with optimal static placement and routing, as well as with baseline policies
- Present a lower-cost approximation solution that achieve within 2.5% of optimum

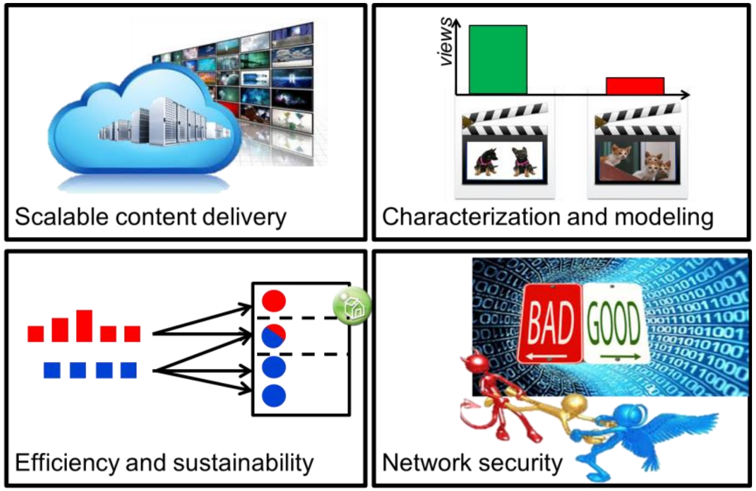
Summary and thank you!

Scalable content delivery

Traffic measurements, analysis, and modeling

Efficient and sustainable ICT

Security and emerging services



LINKÖPINGS UNIVERSITET

Niklas Carlsson (niklas.carlsson@liu.se)
Research overview and pubs: www.ida.liu.se/~nikca/