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

Niklas Carlsson, *Linköping University, Sweden*

Derek Eager, University of Saskatchewan, Canada

Ajay Gopinathan, Google, USA

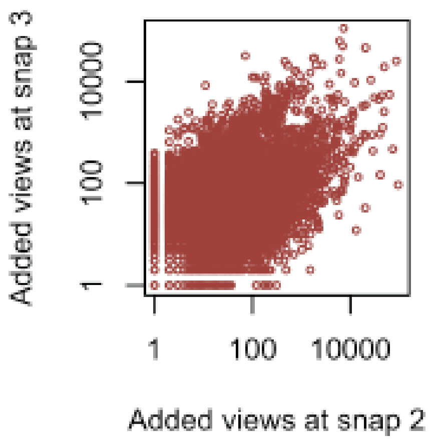
Zongpeng Li, University of Calgary, Canada

@ *IFIP PERFORMANCE 2014*, Torino, Italy, Oct. 2014

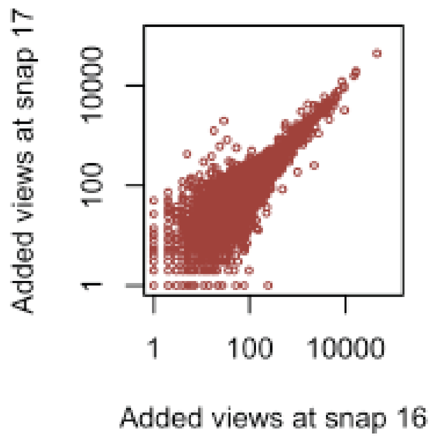
Internet Content Delivery

- 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

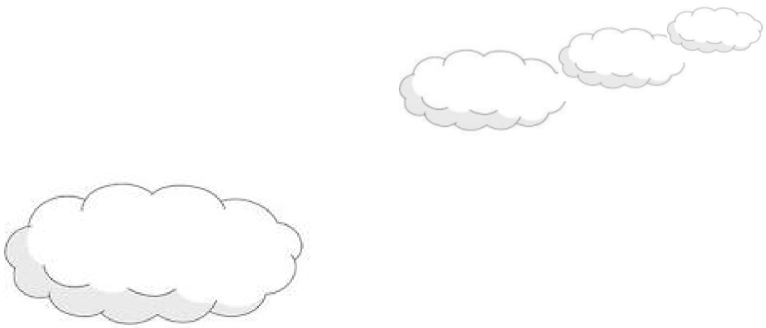
- Large amounts of data with varying popularity
- Multi-billion market ($8B to $20B, 2012-2015)
  - Goal: Minimize content delivery costs
- Migration to cloud data centers

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 geographically distributed cloud data centers

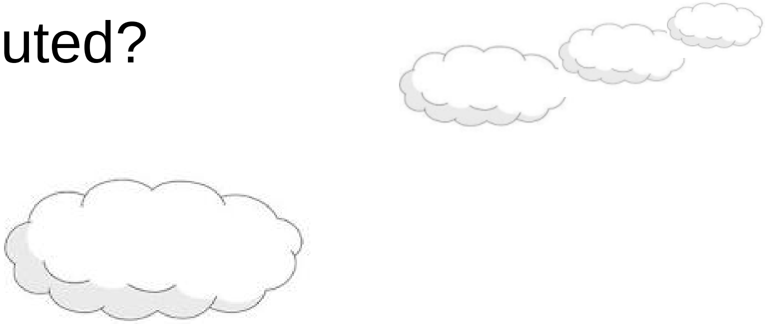
Motivation

- 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

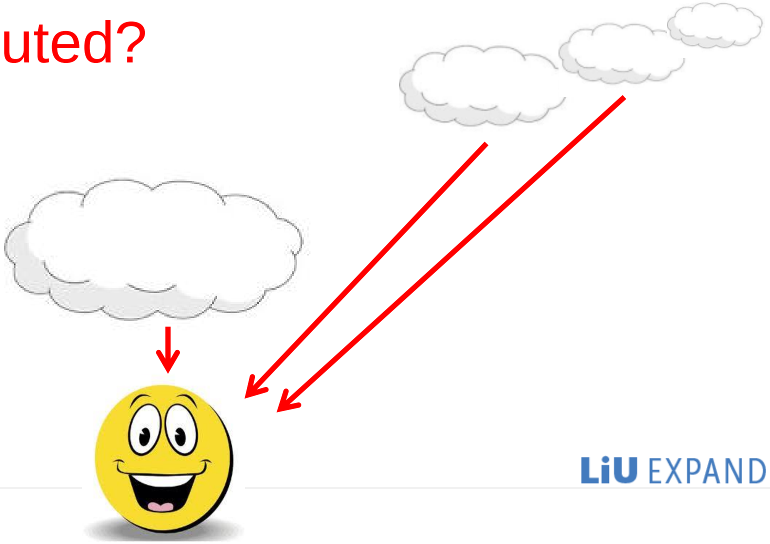
- Geographically distributed cloud
  - Elastic cloud bandwidth and storage
  - When sufficiently expensive storage costs, not all contents should be cached at all locations
- Two policy questions arise
  - What content should be cached where?
  - How should requests be routed?

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?



LiU EXPANDING REALITY

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

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

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

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

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

LiU EXPANDING REALITY

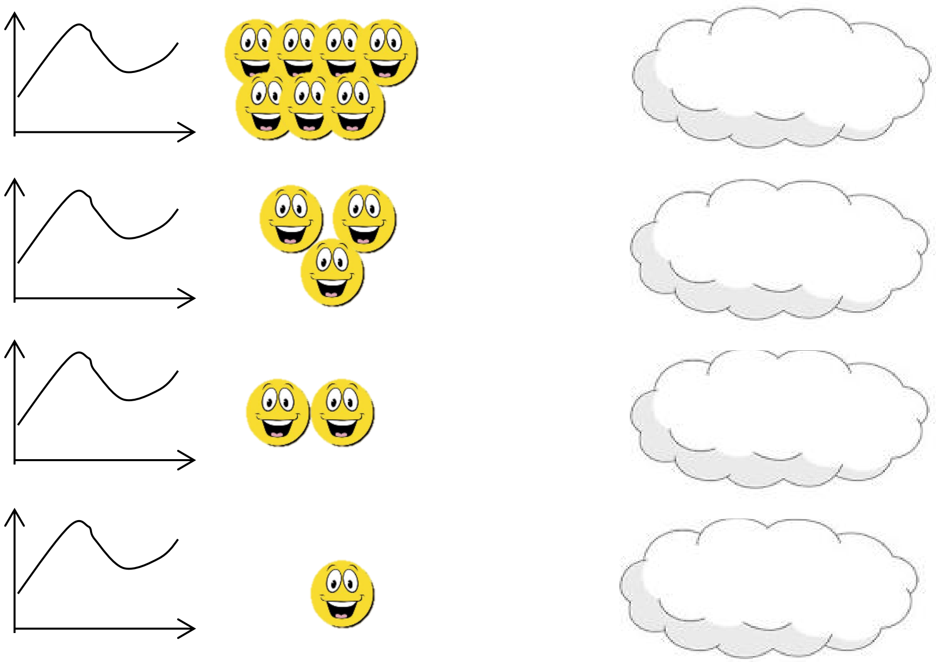
Contributions

- 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

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

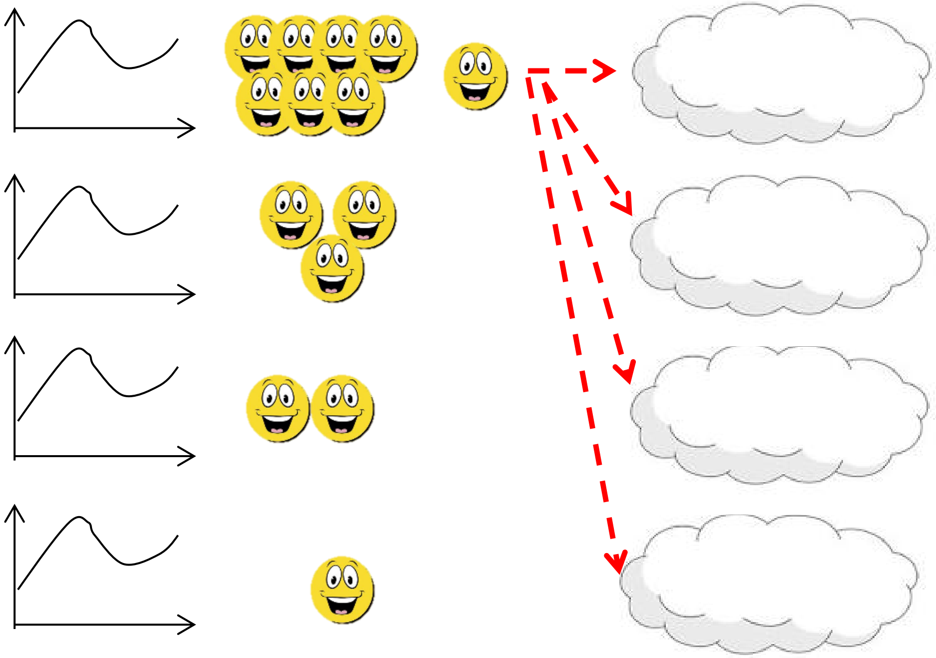
Dynamic TTL-based approach



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

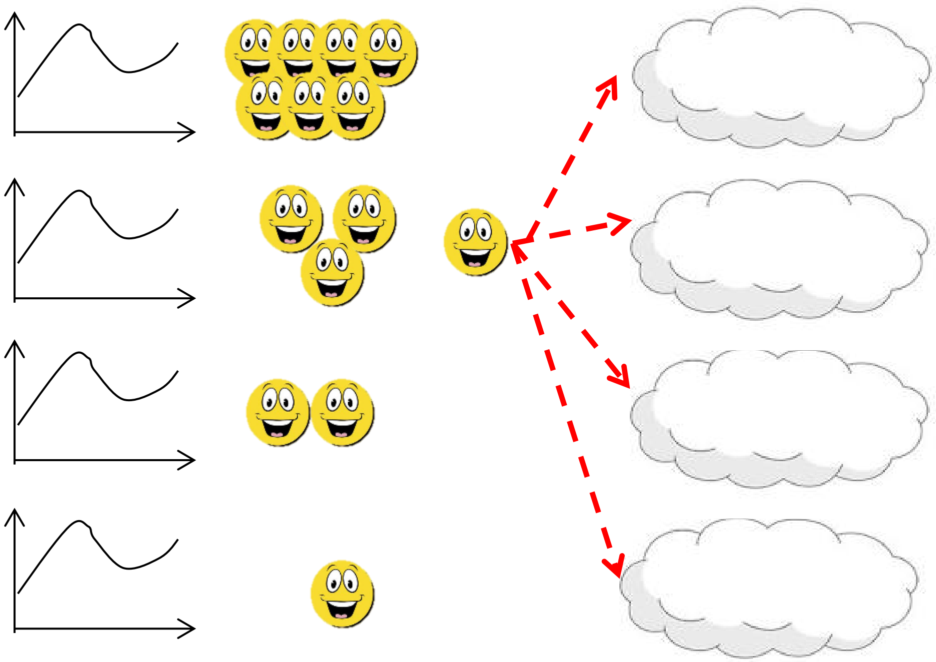
Dynamic TTL-based approach



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

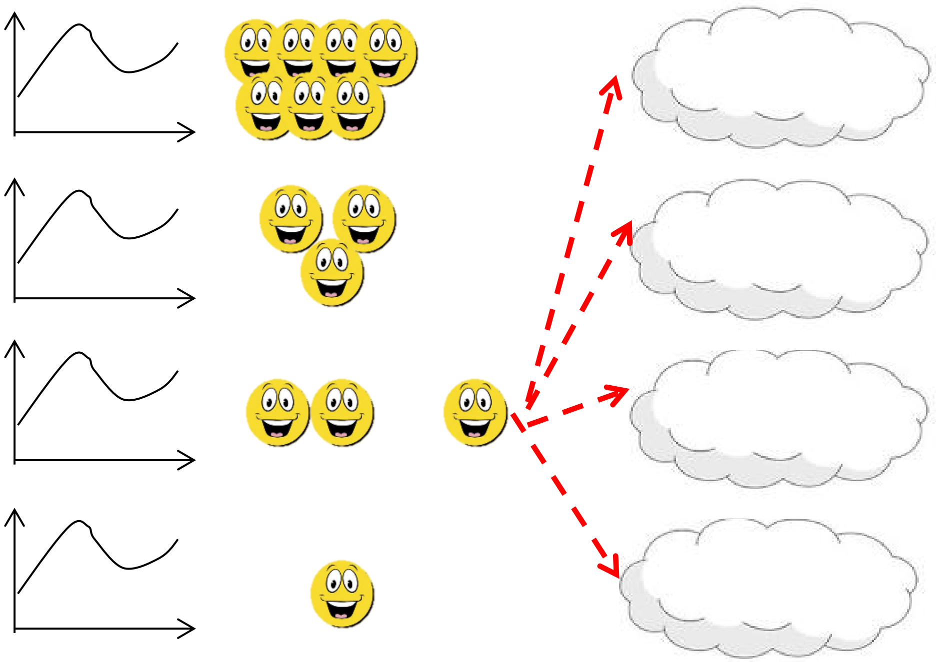
Dynamic TTL-based approach



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

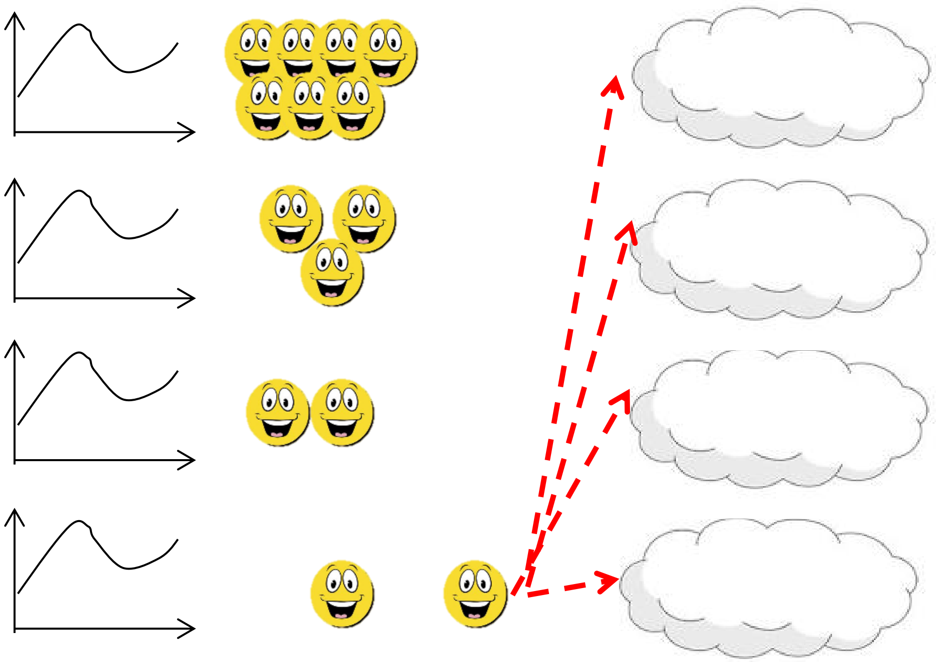
Dynamic TTL-based approach



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

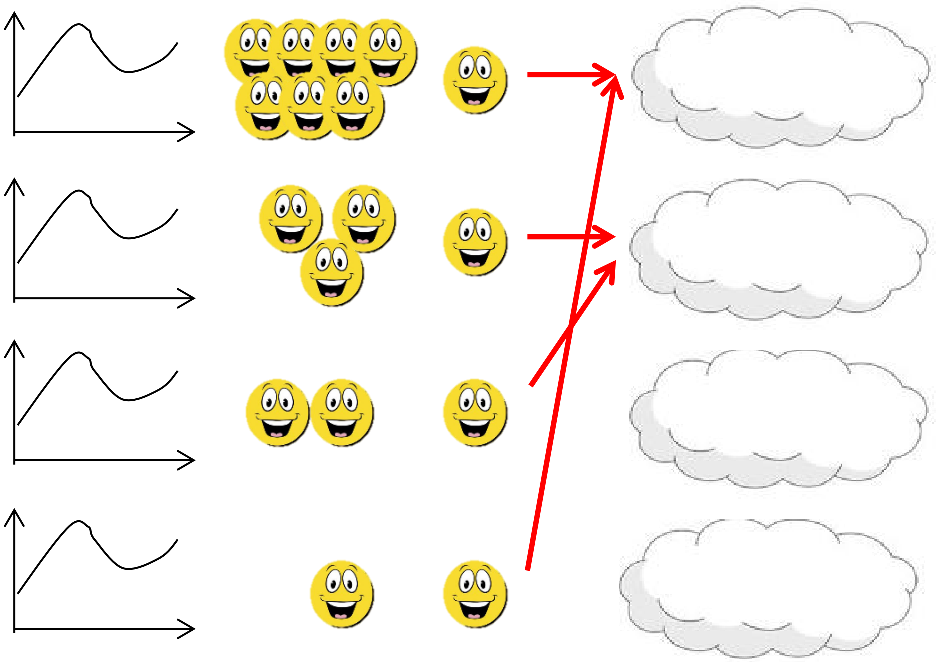
Dynamic TTL-based approach



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

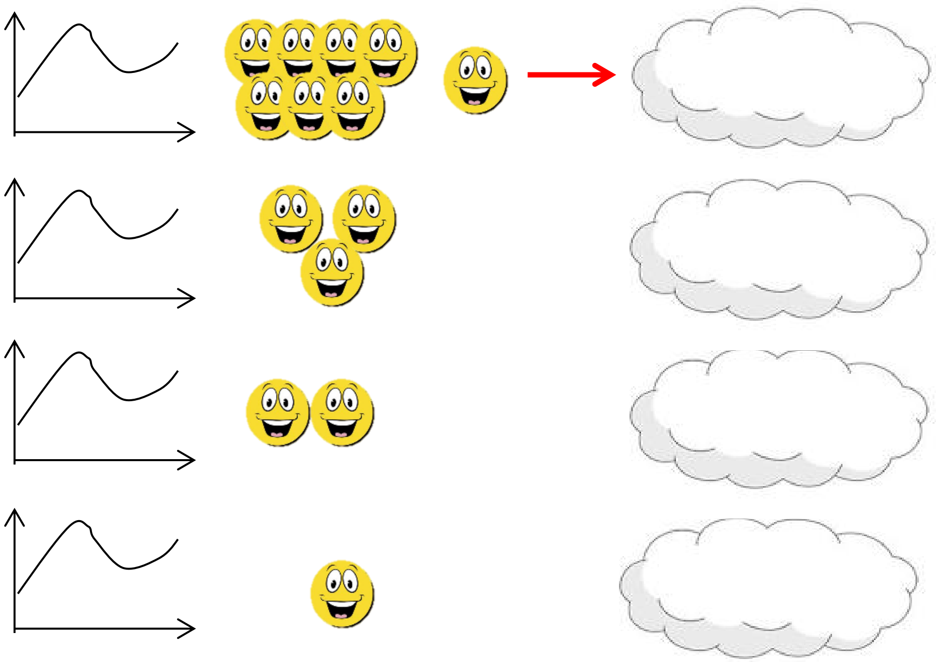


1) Request routing

2) TTL-caching

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

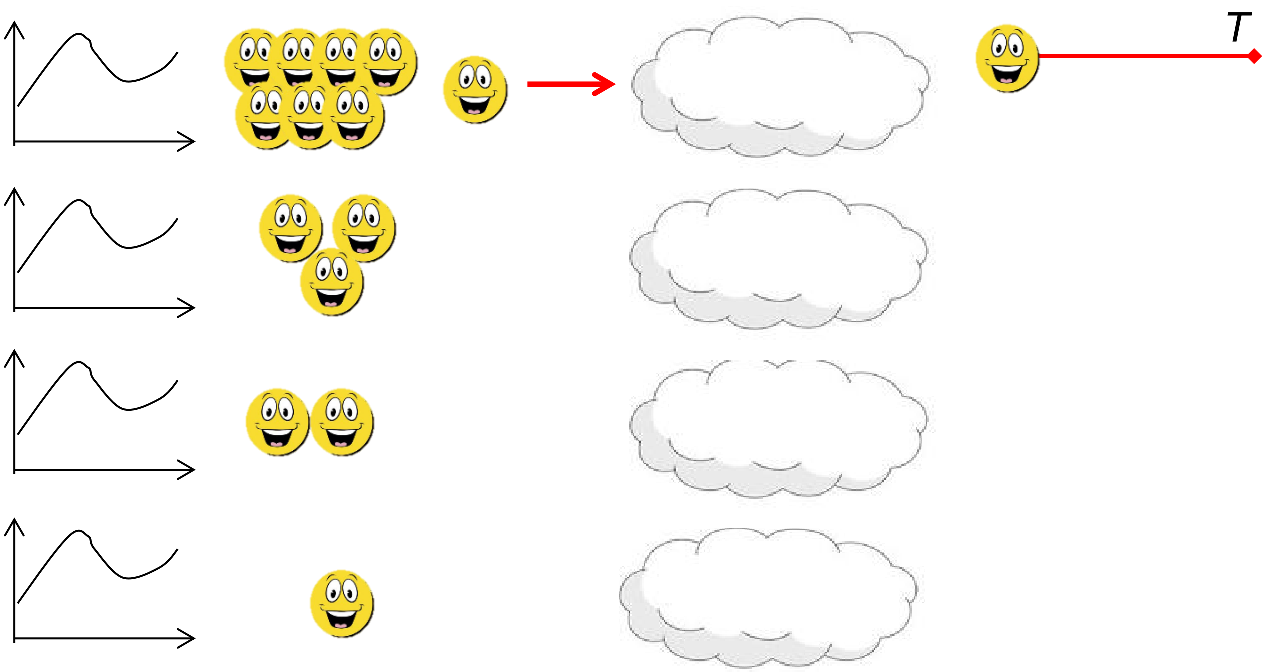
Dynamic TTL-based approach



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

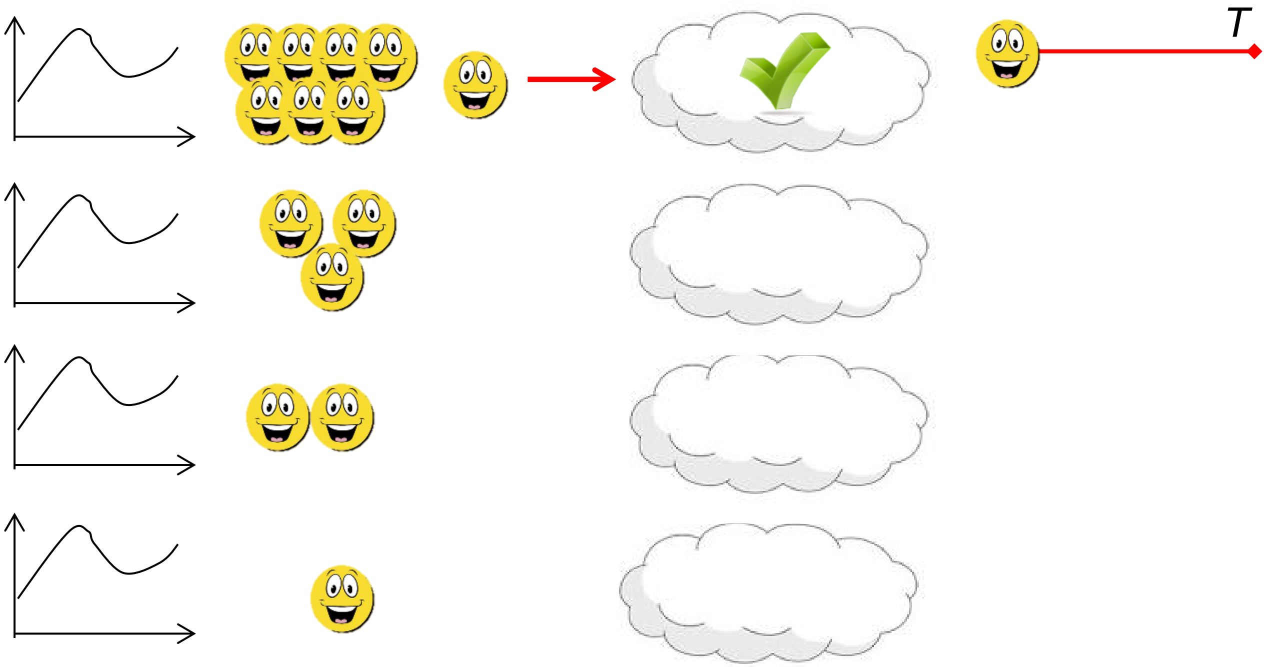
Dynamic TTL-based approach



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

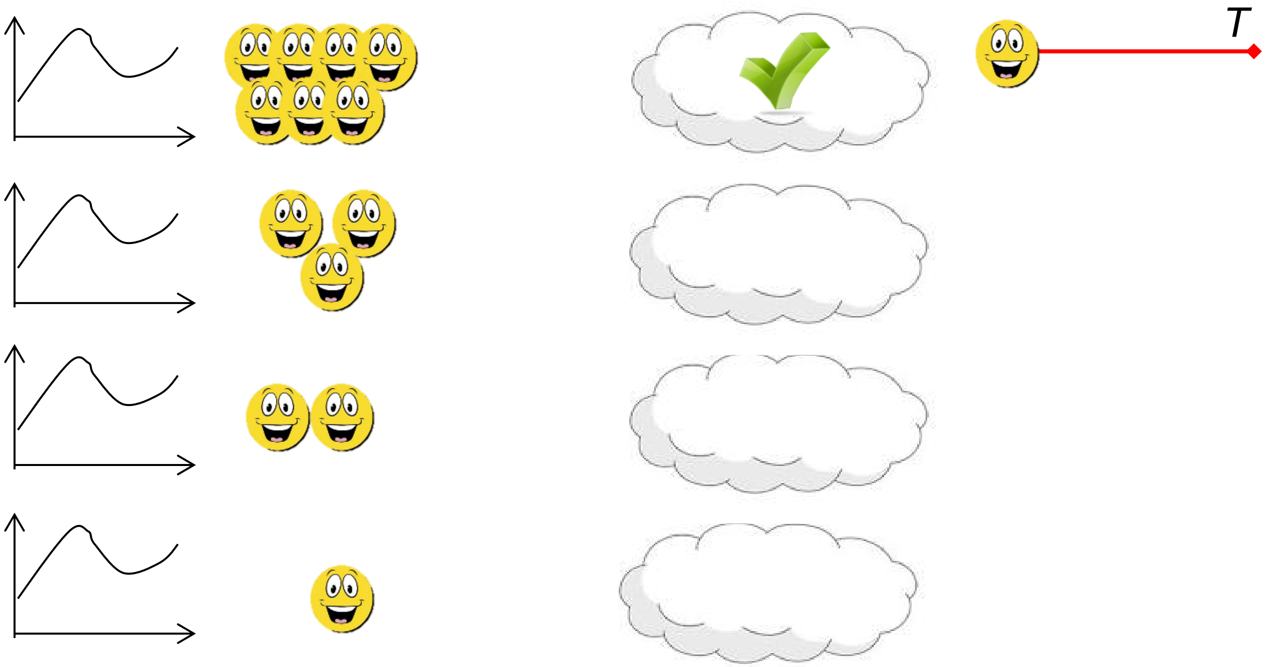
Dynamic TTL-based approach



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

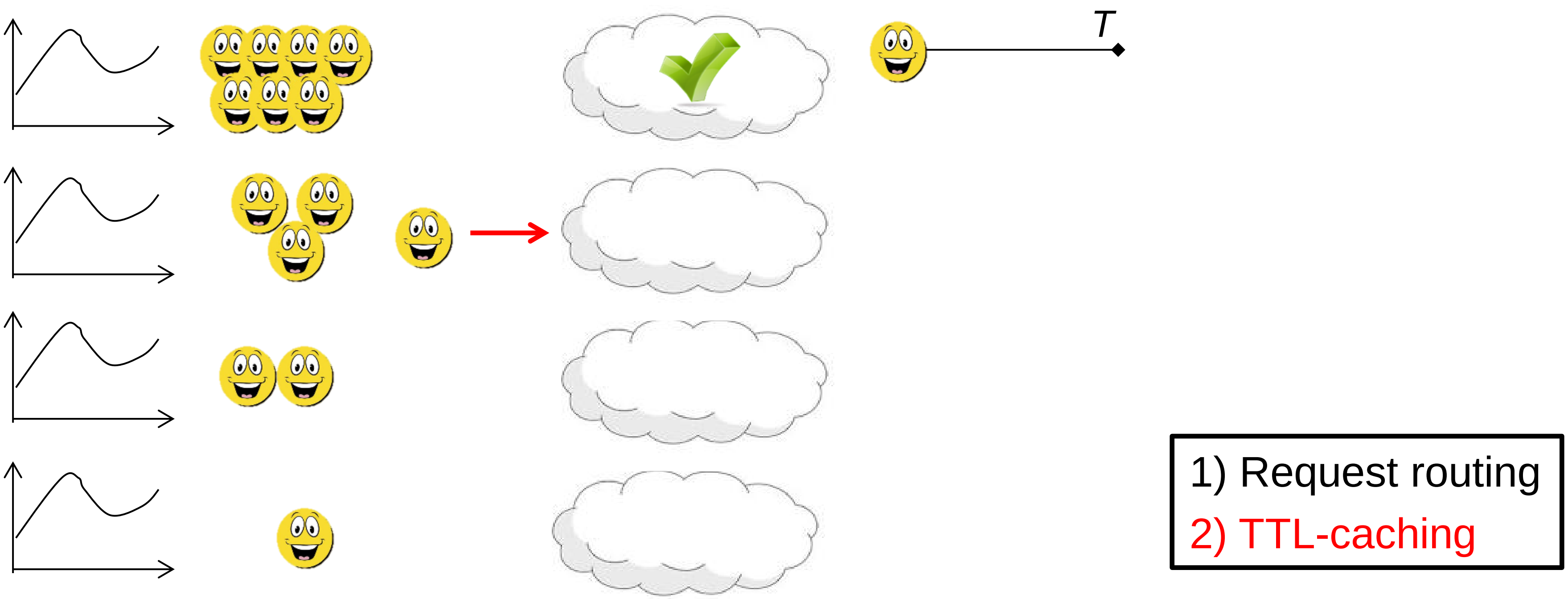
Dynamic TTL-based approach



1) Request routing
2) TTL-caching

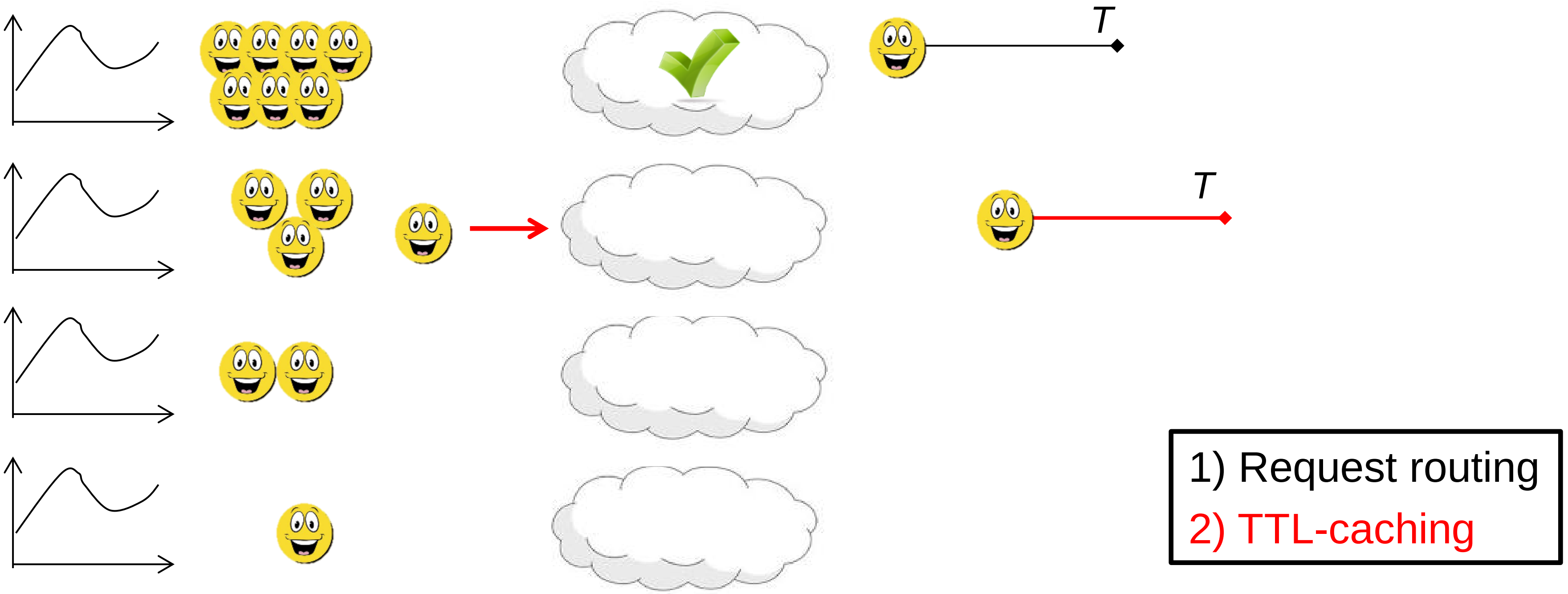
- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

Dynamic TTL-based approach



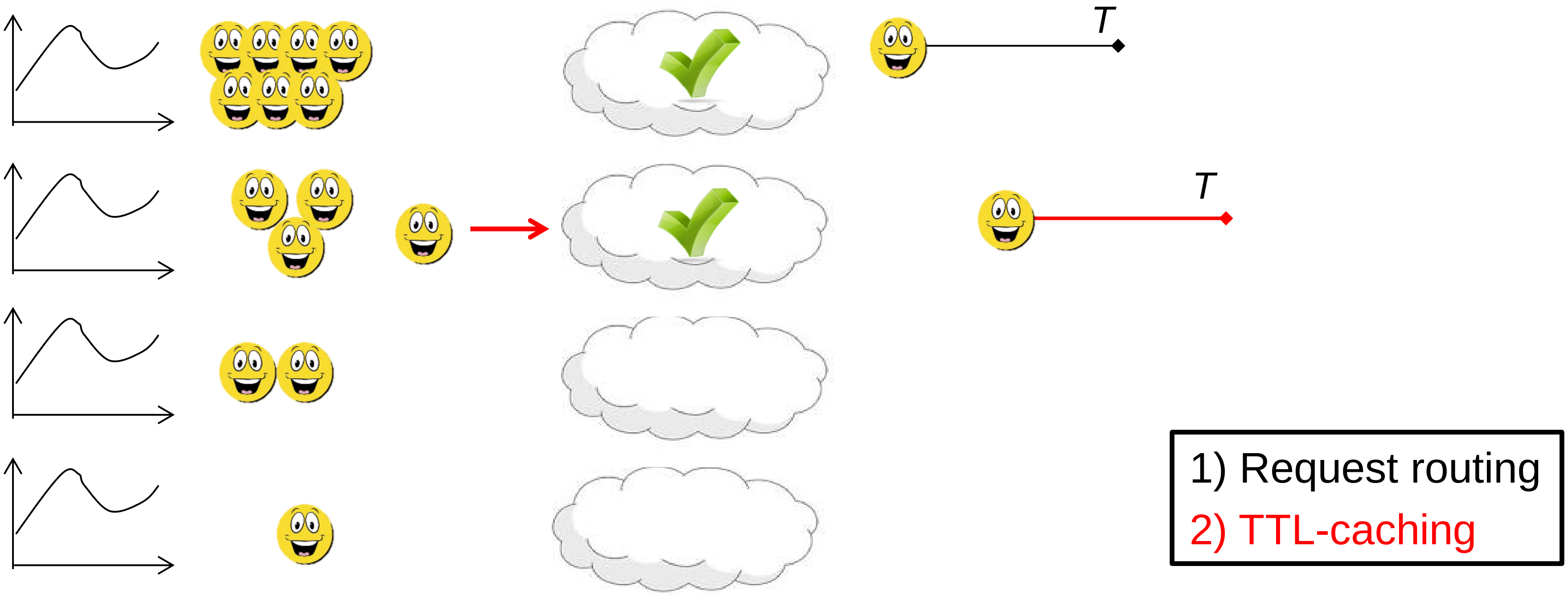
- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

Dynamic TTL-based approach



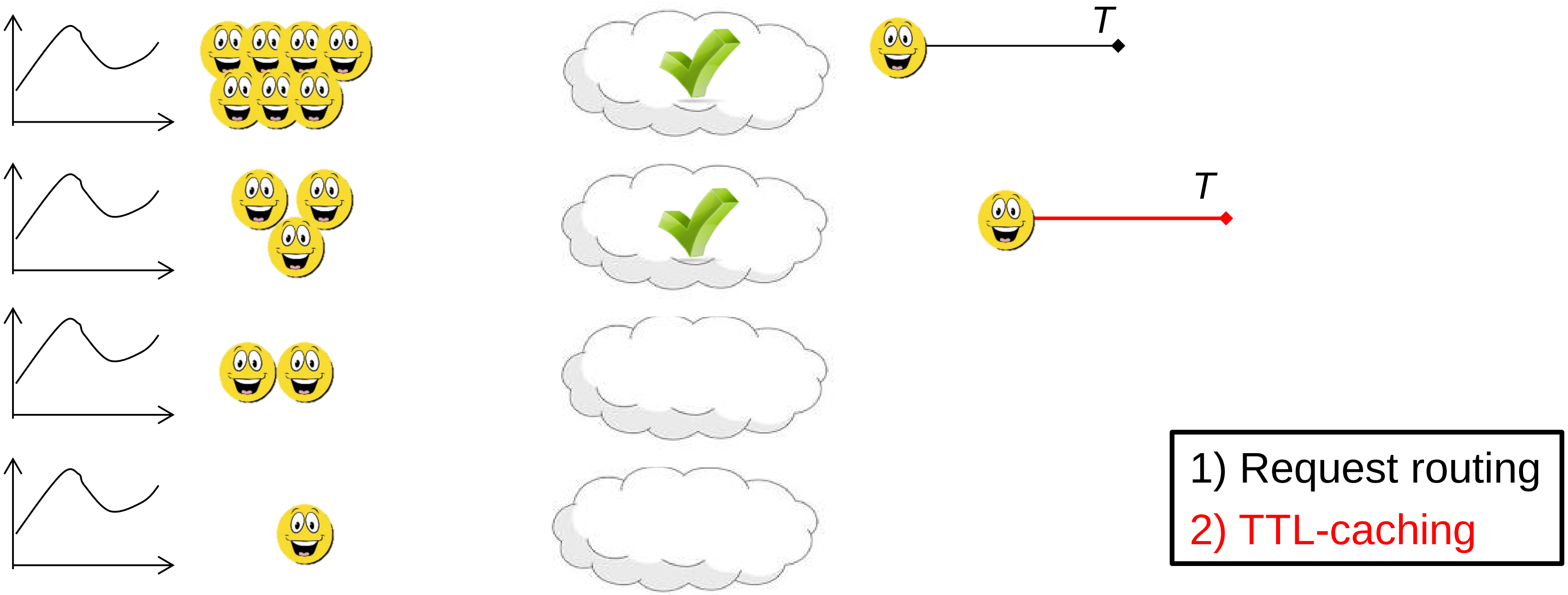
- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

Dynamic TTL-based approach



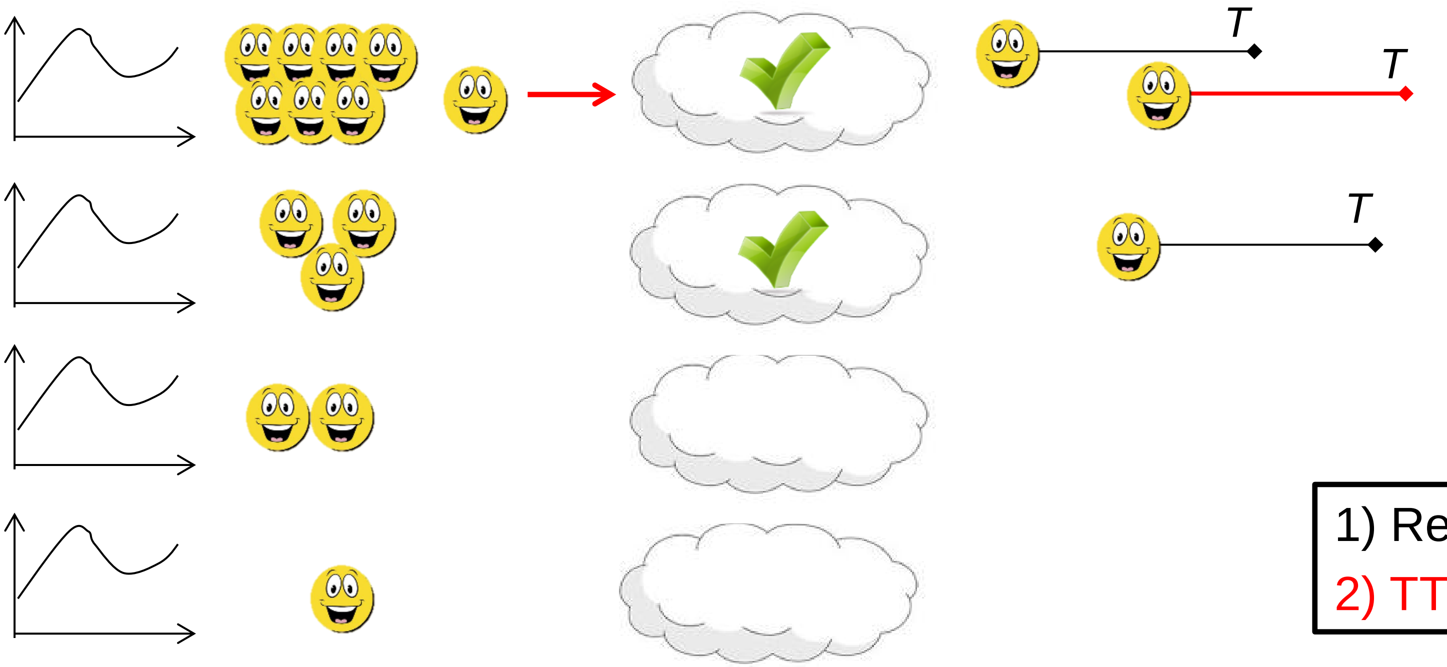
- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

Dynamic TTL-based approach



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

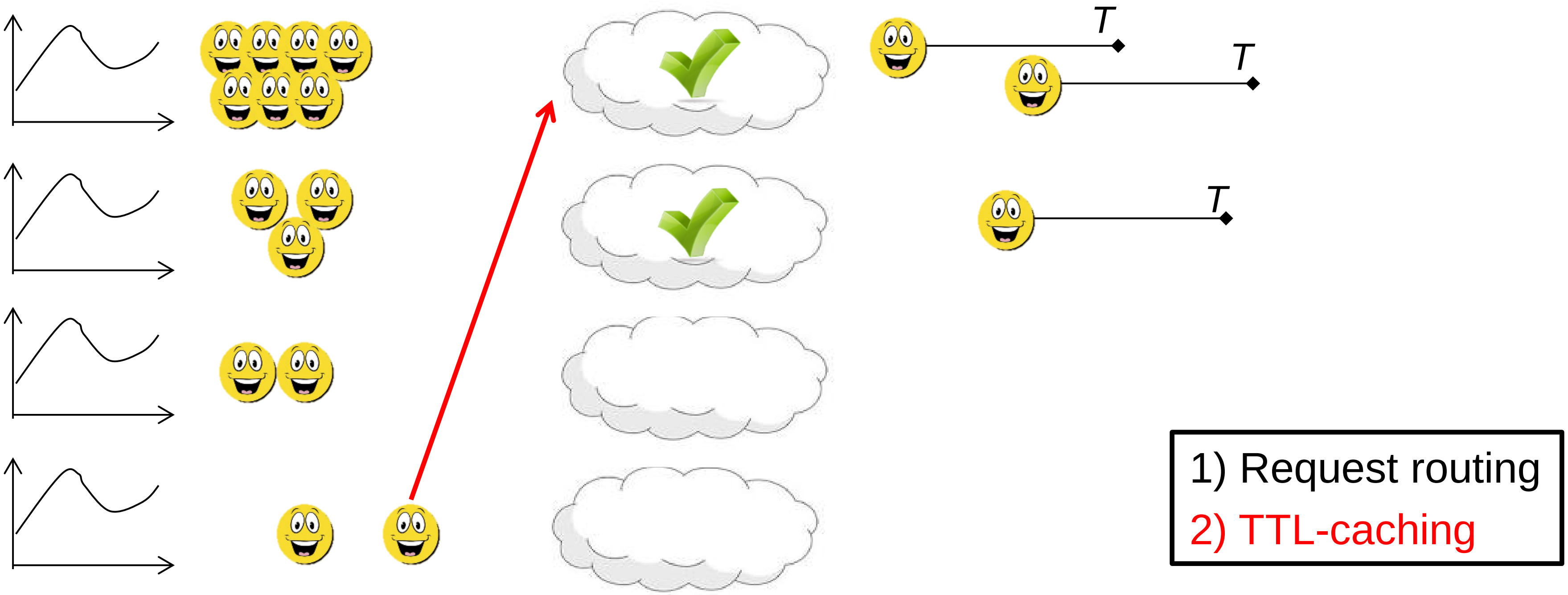
Dynamic TTL-based approach



1) Request routing
2) TTL-caching

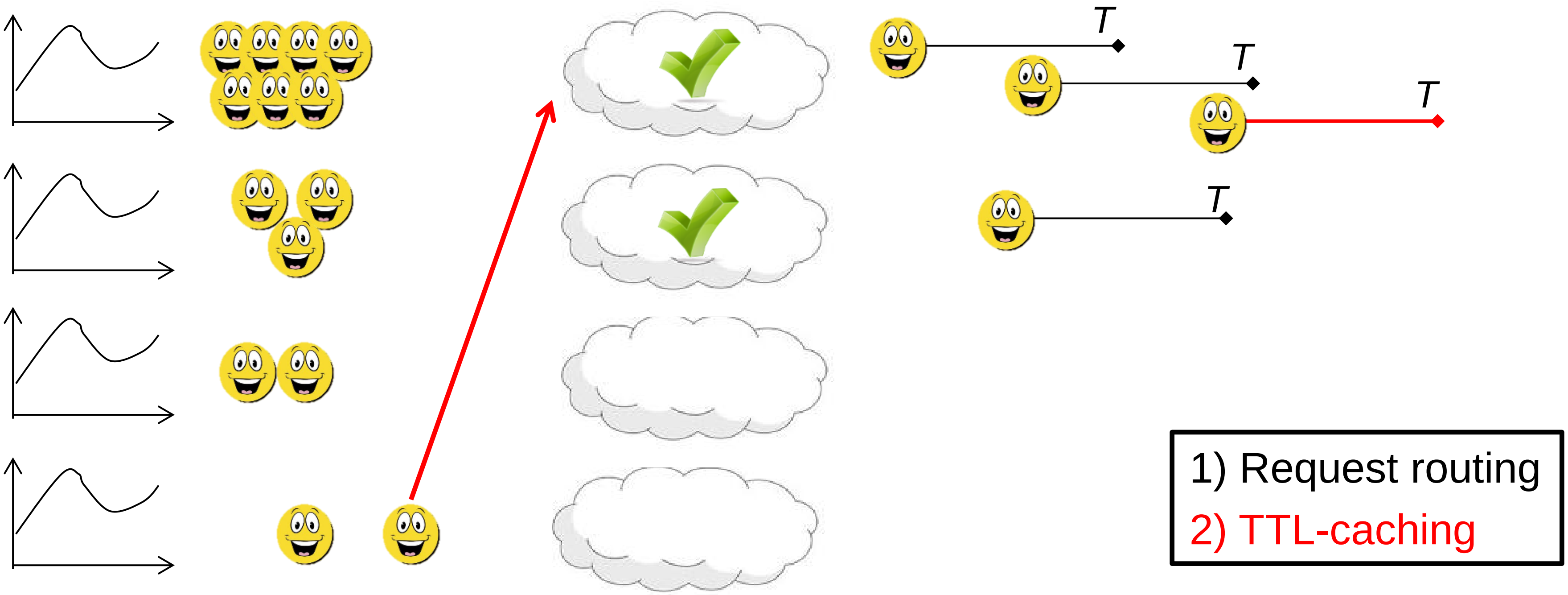
- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

Dynamic TTL-based approach



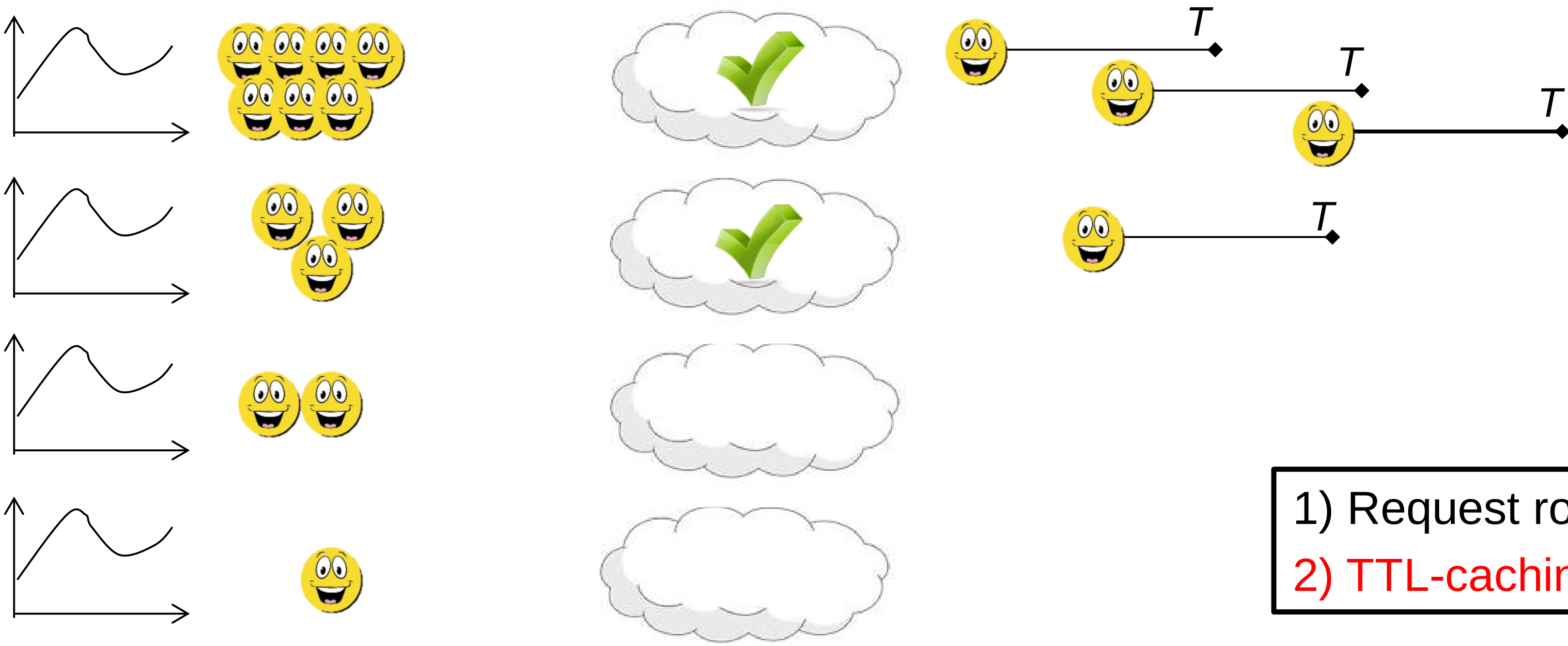
- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

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



1) Request routing
2) TTL-caching

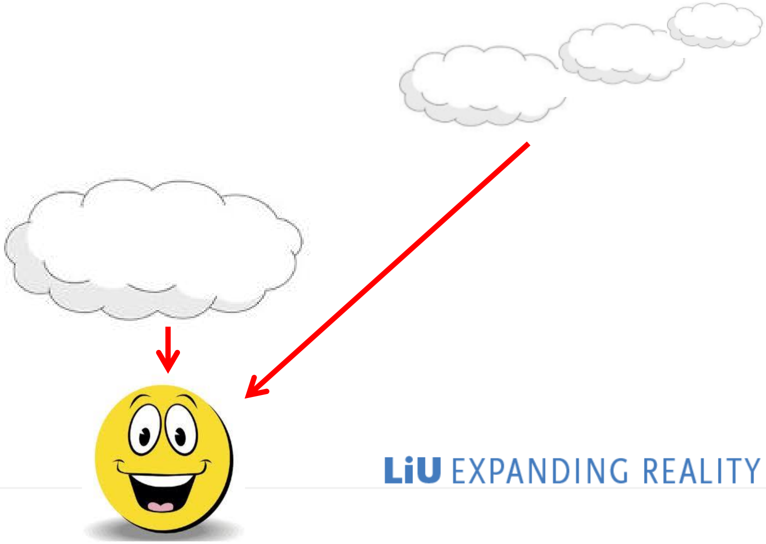
- Elastic cloud bandwidth and storage
  - TTL T_i used at each server location
- Optimized request routing determines content replication

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



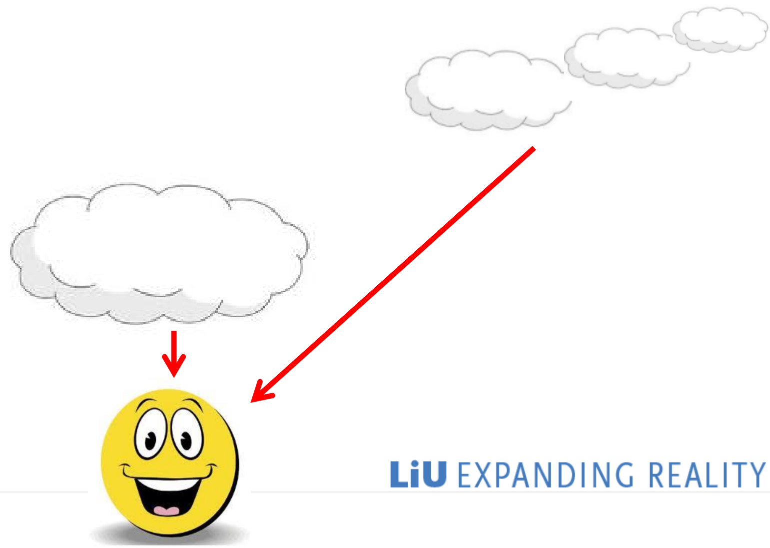
Request routing optimization

Minimize

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

Aggregate request rate at server location i

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



Request routing optimization

Minimize

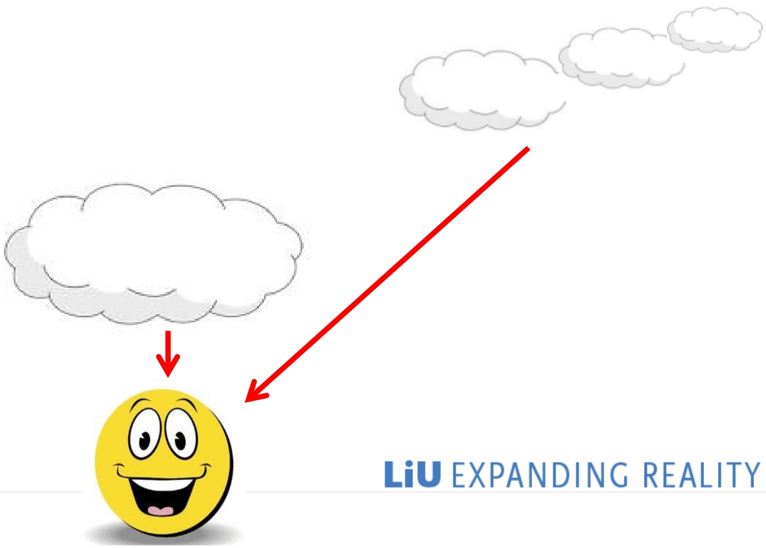
$$\sum_{i\in \mathcal{N}}\left(\gamma_i e^{-\gamma_i T} + L(1-e^{-\gamma_i T}) + R\sum_{c\in \mathcal{M}: i^*(c)\neq i}\lambda_{c,i}\right), \quad \text{where } \gamma_i = \sum_{c\in \mathcal{M}}\lambda_{c,i}$$

Cache miss cost

Cache storage cost

Remote routing cost

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



LiU EXPANDING REALITY

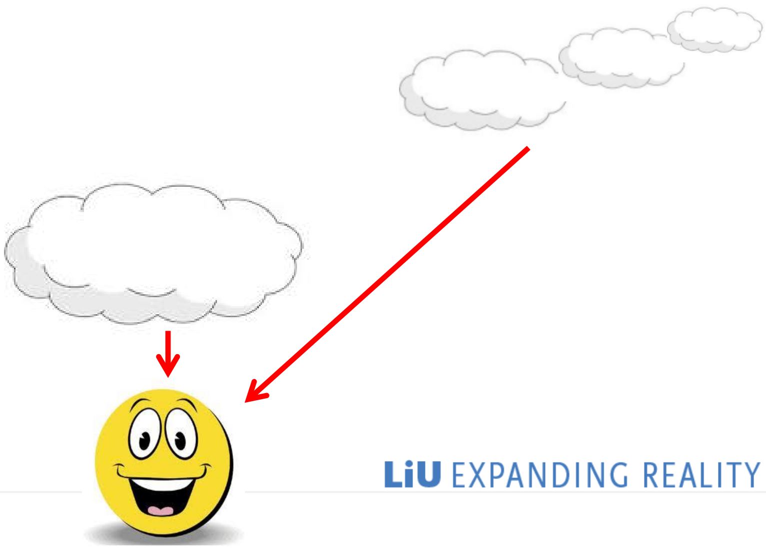
Request routing optimization

Minimize

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

Cache miss cost

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



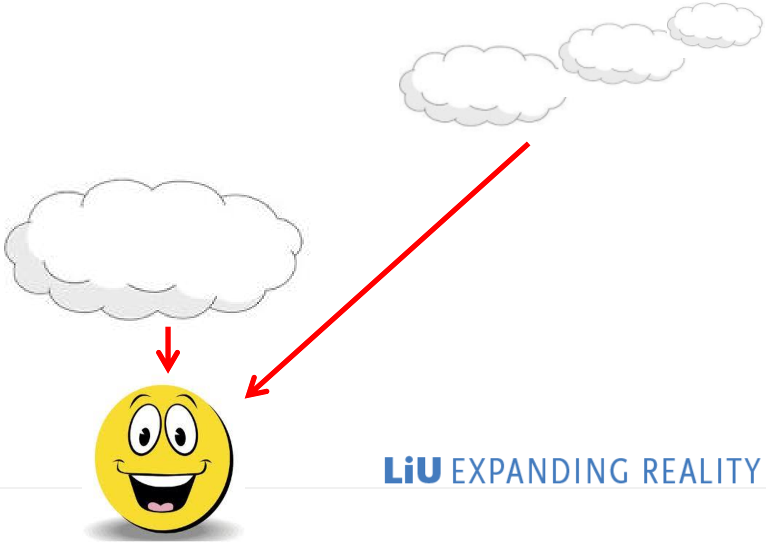
Request routing optimization

Minimize

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

Cache storage cost

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



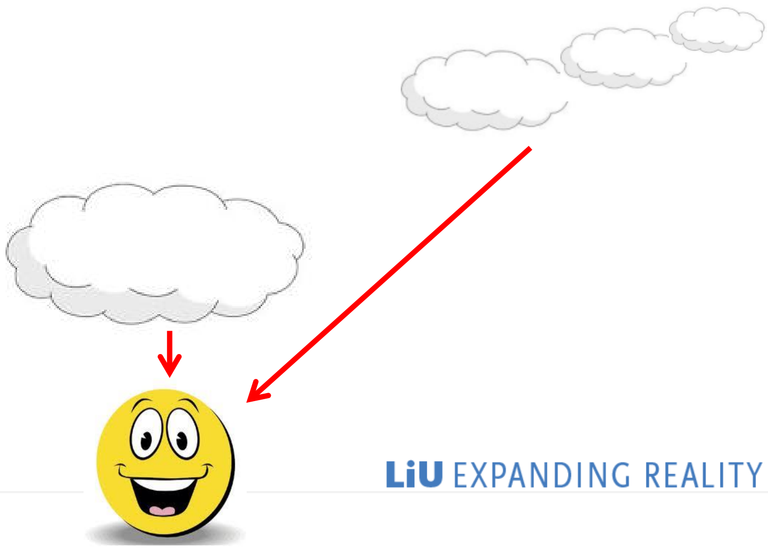
Request routing optimization

Minimize

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

Remote routing cost

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



Request routing optimization

Minimize

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

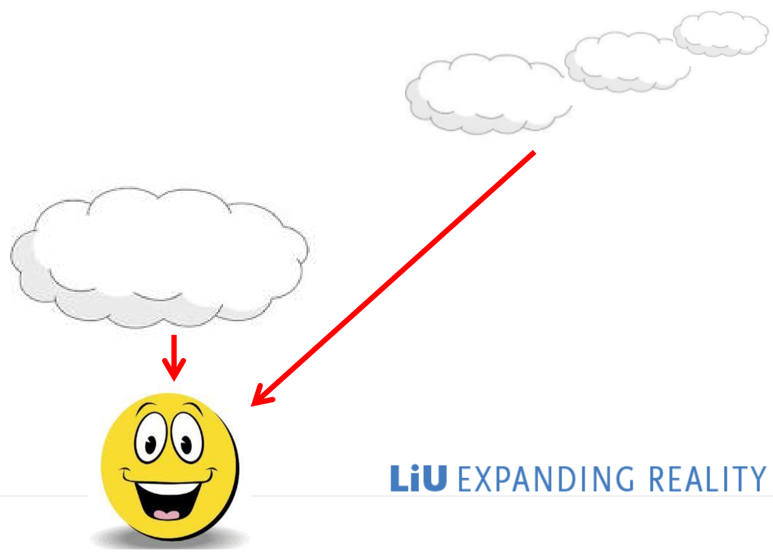
Cache miss cost

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

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

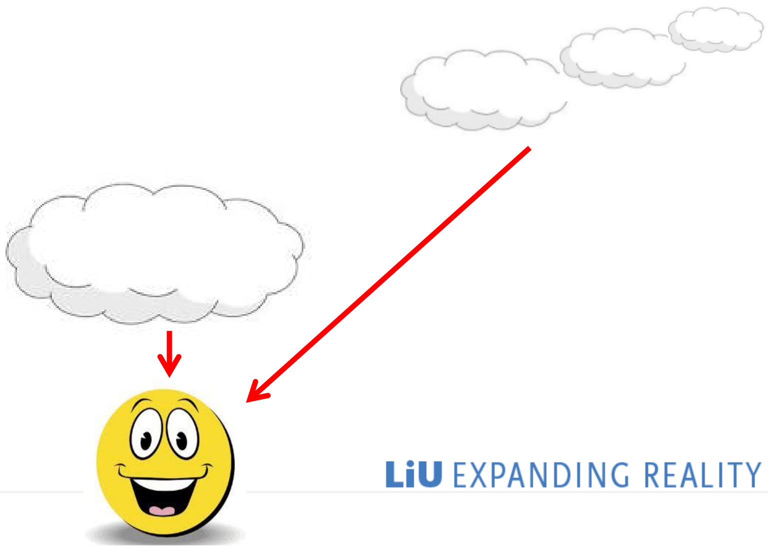
Subject to

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

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

Conservation constraints

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



Request routing optimization

Minimize

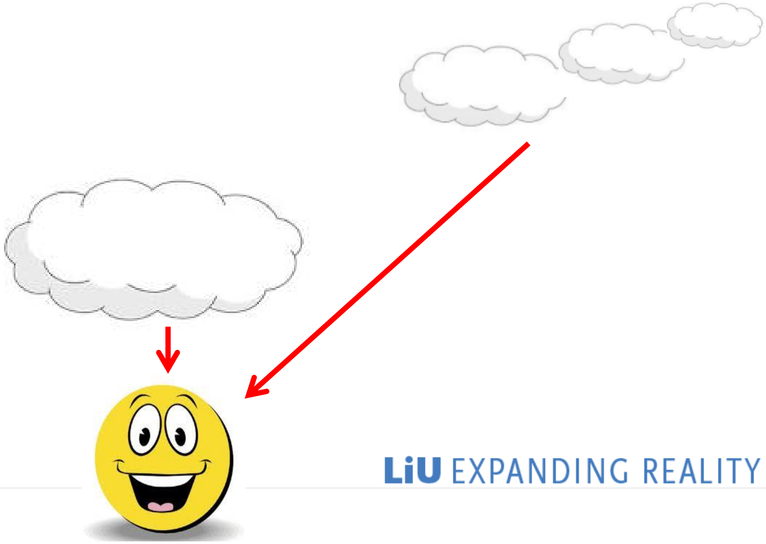
$$\sum_{i\in\mathcal{N}}\left(\gamma_i e^{-\gamma_i T} + L(1-e^{-\gamma_i T}) + R\sum_{c\in\mathcal{M}:i^*(c)\neq i}\lambda_{c,i}\right), \quad \text{where } \gamma_i = \sum_{c\in\mathcal{M}}\lambda_{c,i}$$

Subject to

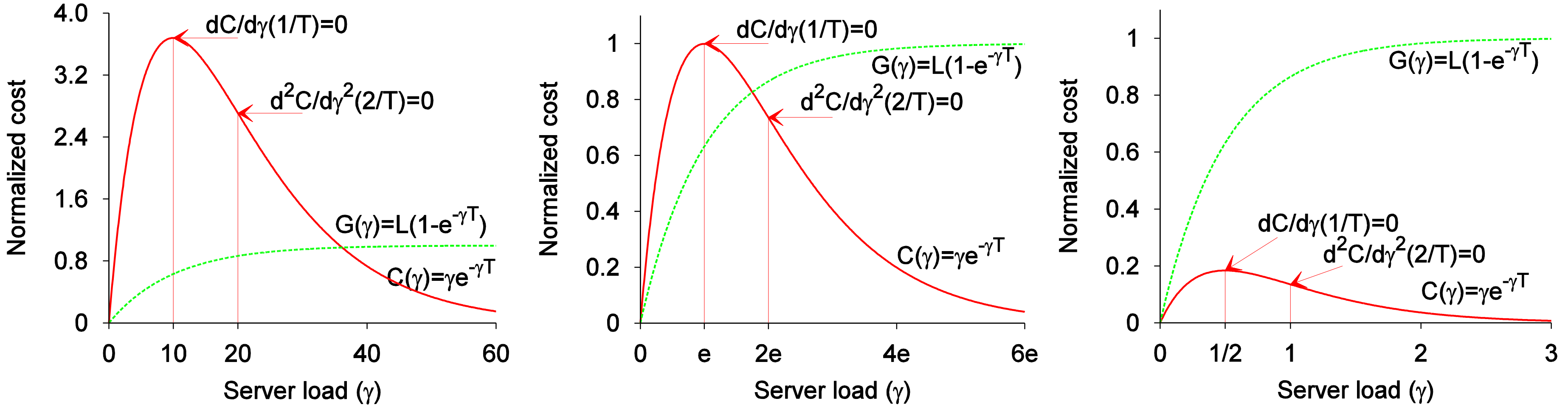
$$\sum_{i\in\mathcal{N}}\lambda_{c,i} = \lambda_c, \quad \forall c \in \mathcal{M}$$

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

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



Cost tradeoff example



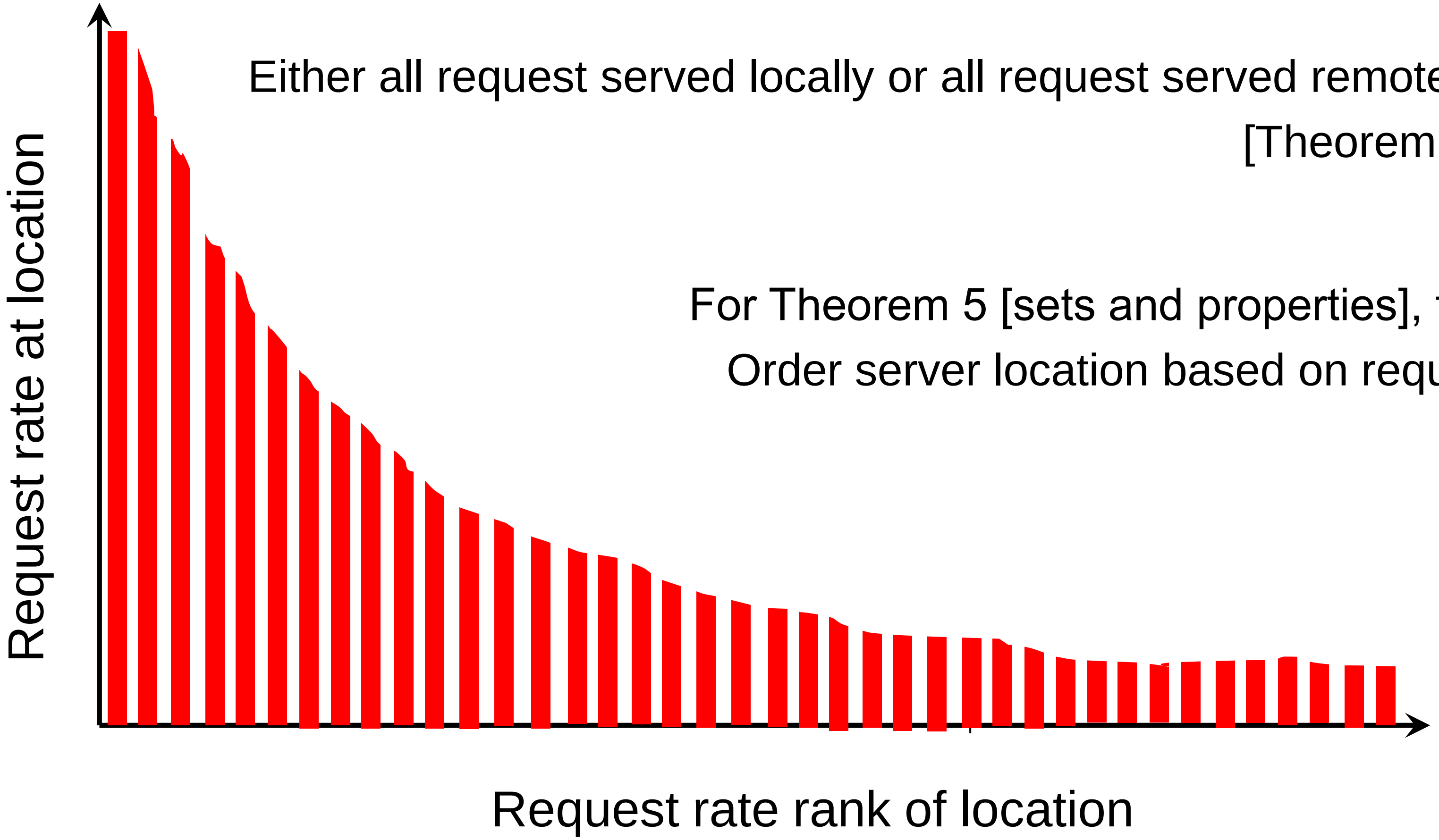
(a) Miss cost dominates (b) Equal peak costs (c) Storage cost dominates

- Rates of incurring cache miss and storage costs
  - Miss cost function has infliction point (red curves)
  - Storage cost function concave (green curves)

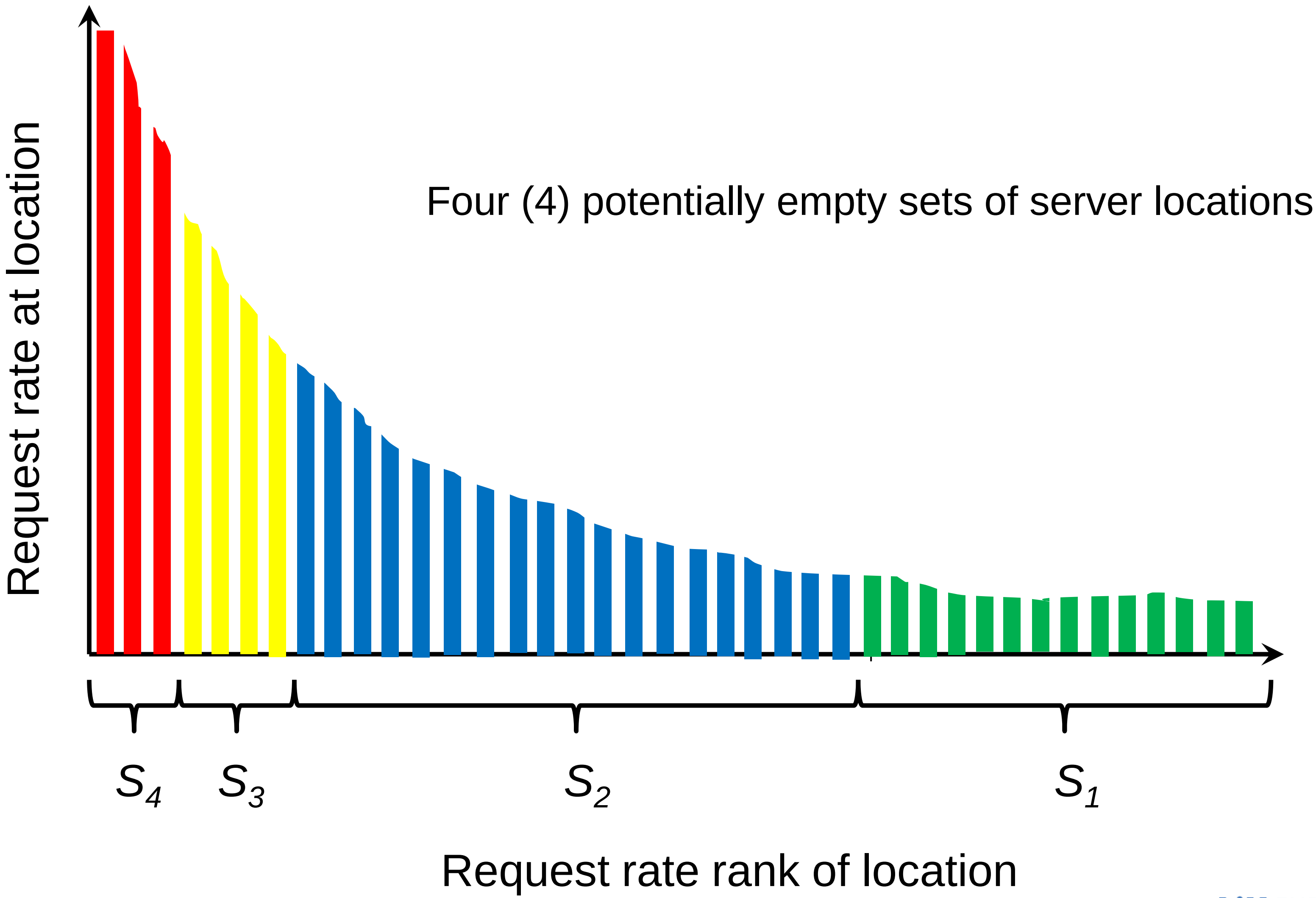
Summary of optimal request routing results

- Special cases
  - R $\rightarrow$ 0 [Theorem 1: single server]
  - R $\rightarrow \infty$ [Theorem 2: always local]
  - Ignoring miss cost [Theorem 3: all remote to single server]
- General case
  - Either all request local or all request remote [Theorem 4]
  - Optimal to split servers in four sets, each with properties that allow solution to be found at calculation cost $O(N^3)$ [Theorem 5]
- Optimal static placement
  - Optimal static routing with heterogeneous T_i thresholds results in static placement with calculation cost $O(N^2)$ [Theorem 6]

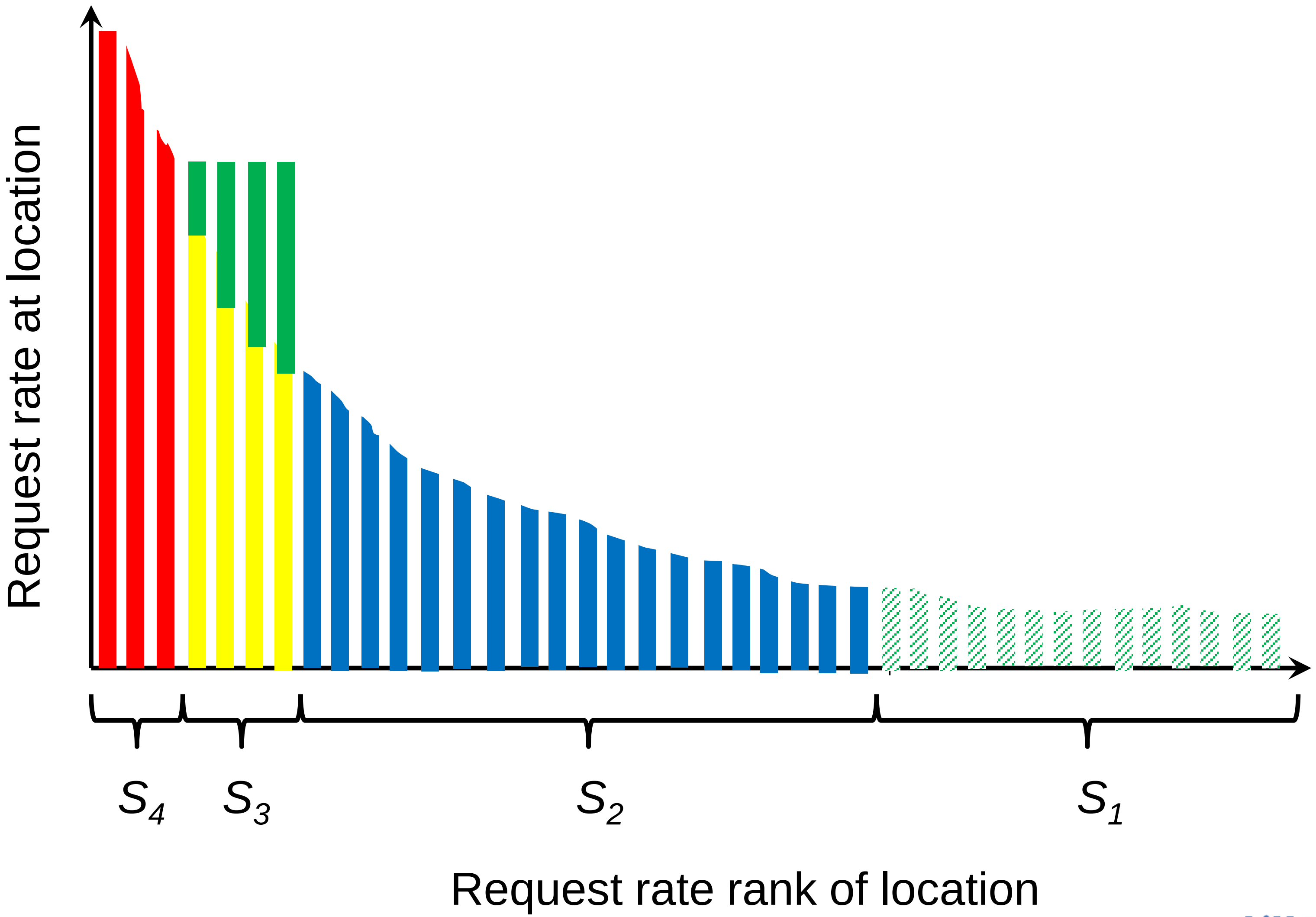
Properties of optimal request routing



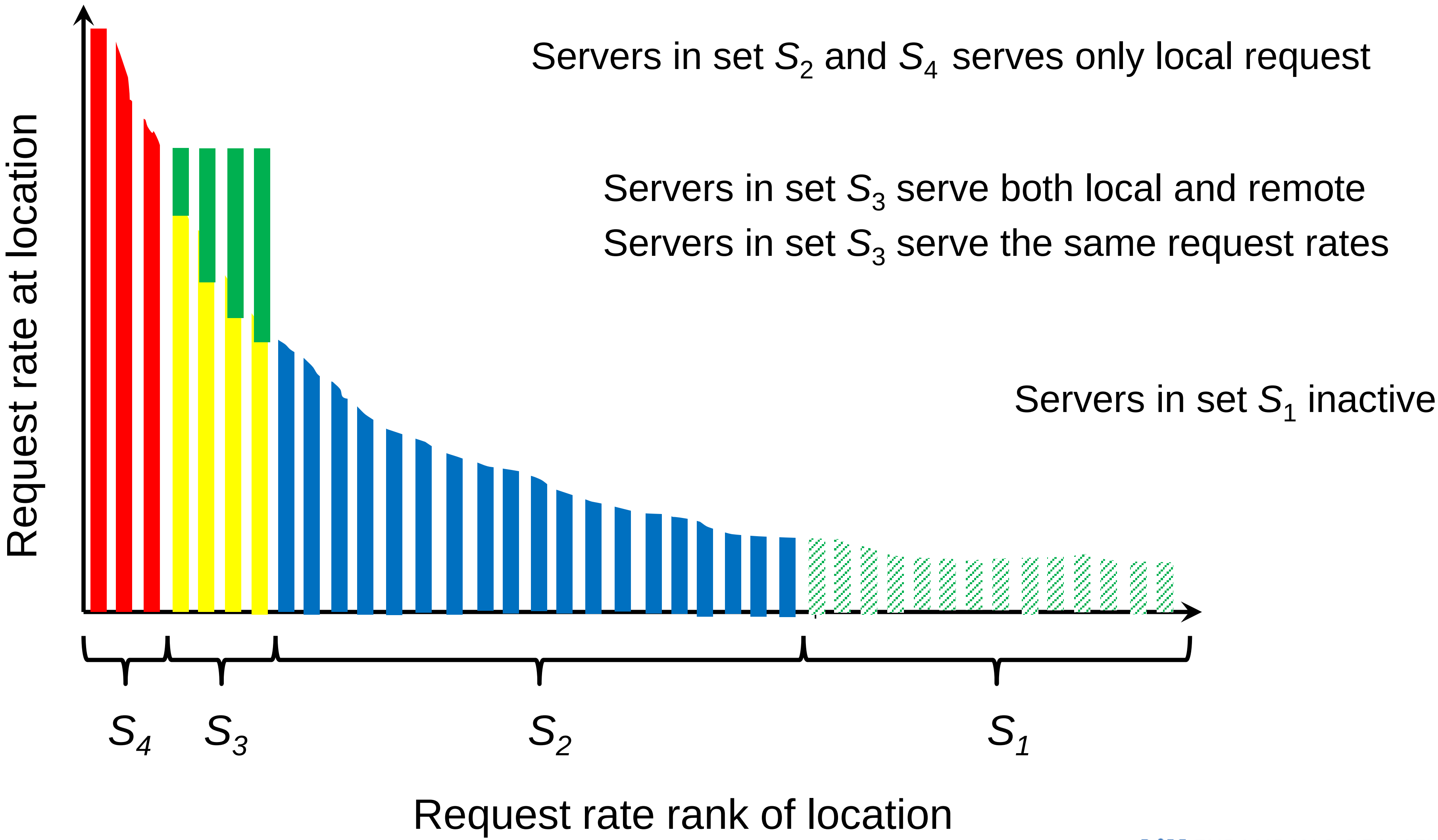
Properties of optimal request routing



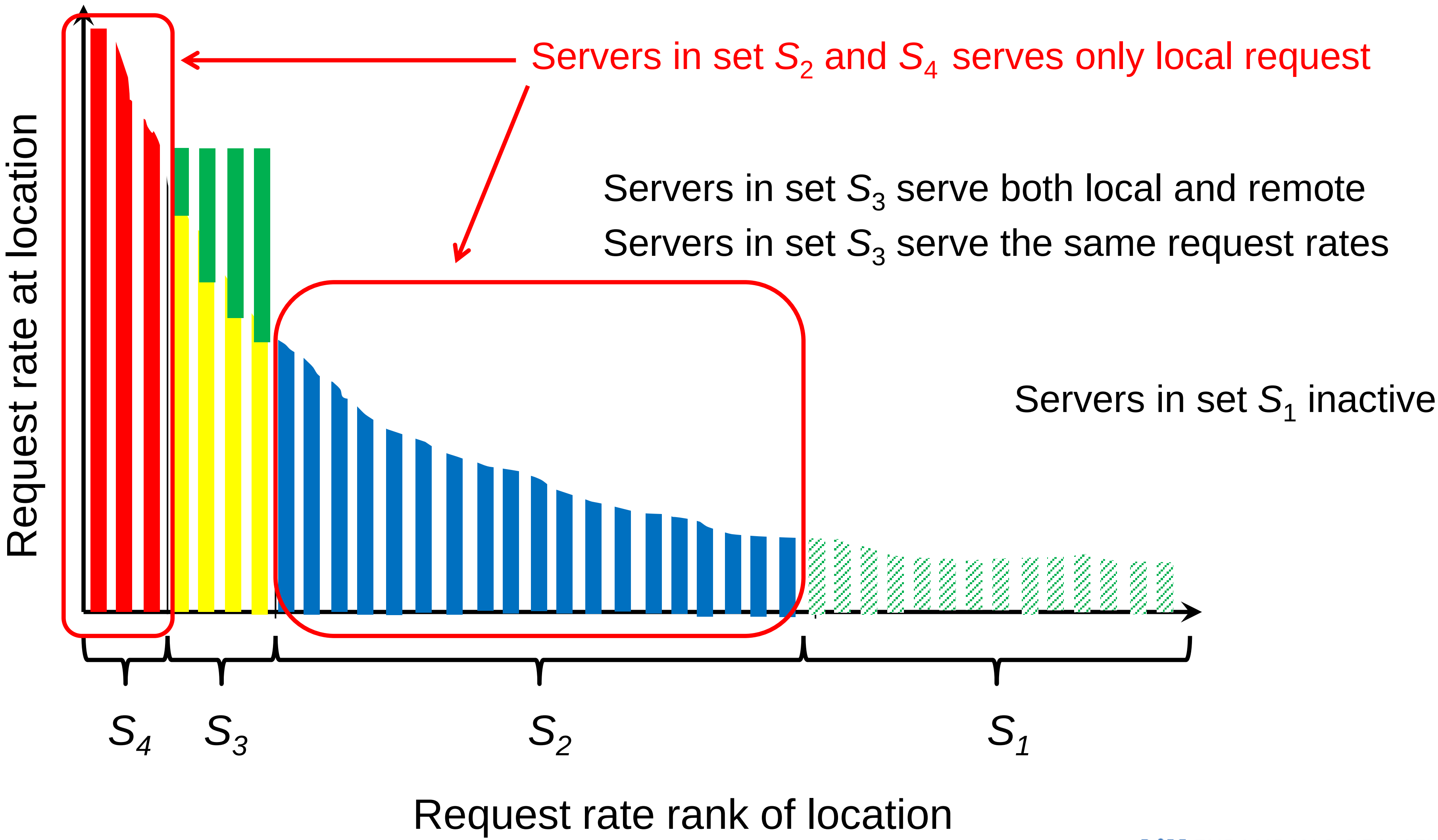
Properties of optimal request routing



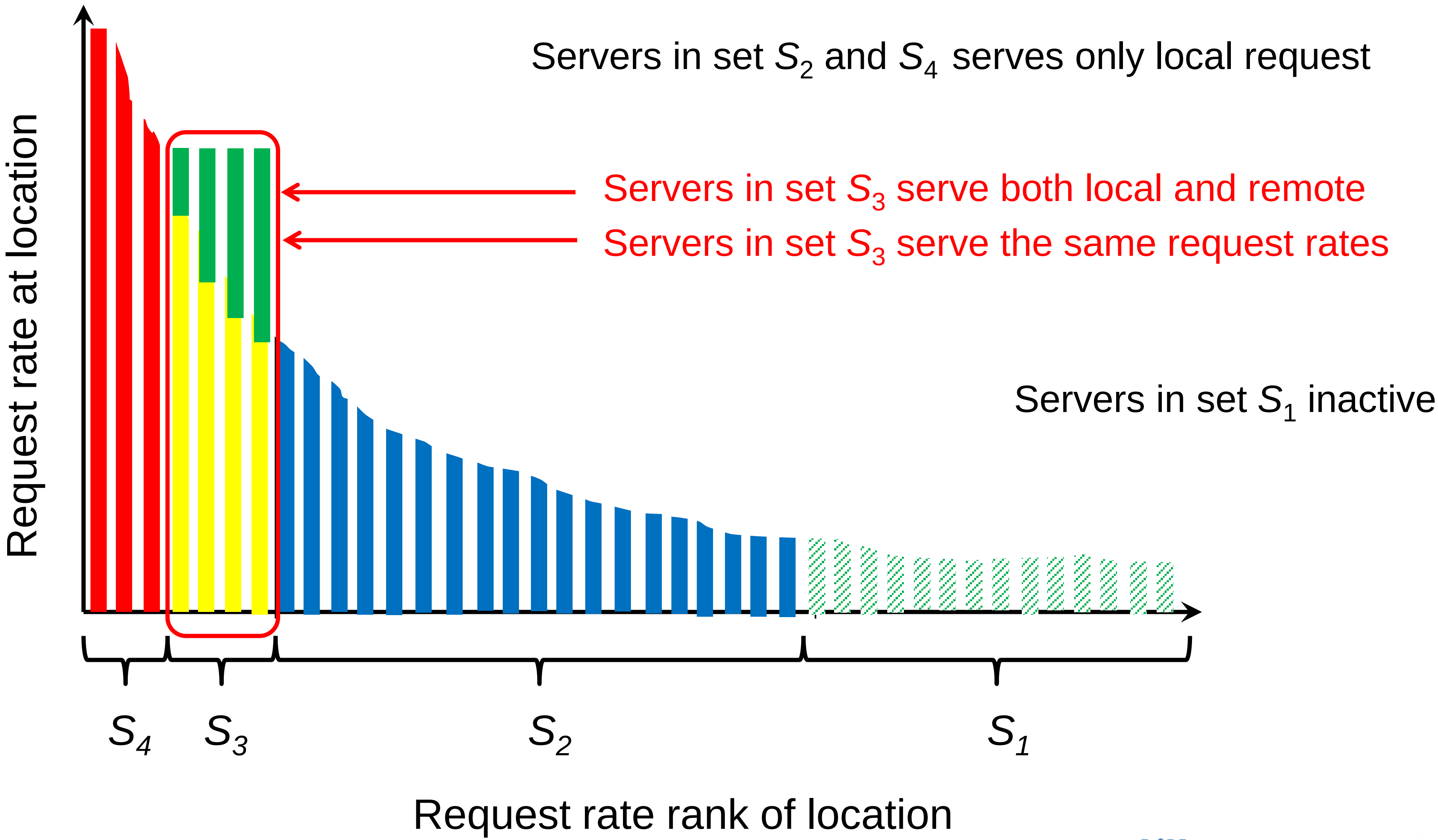
Properties of optimal request routing



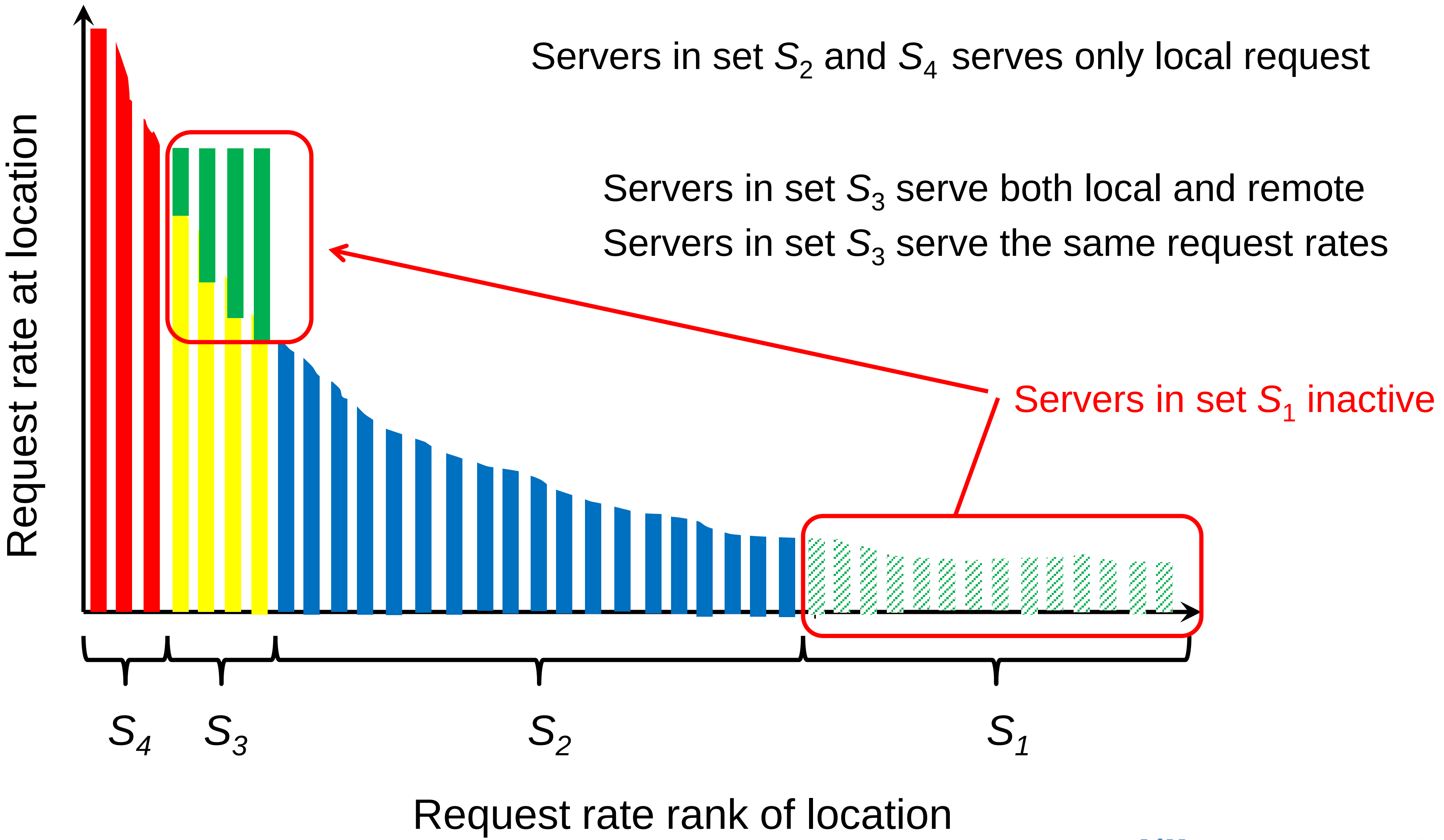
Properties of optimal request routing



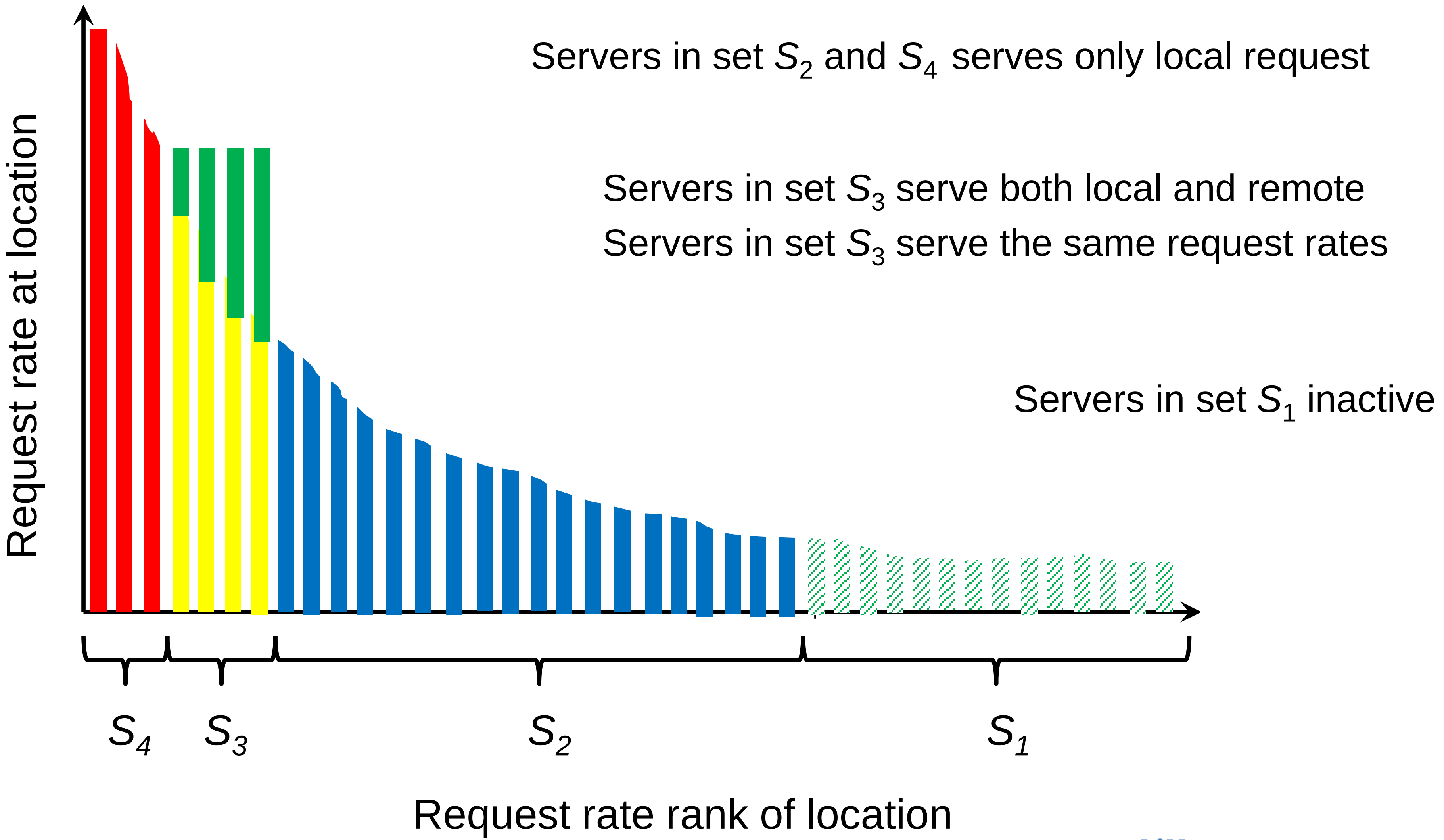
Properties of optimal request routing



Properties of optimal request routing

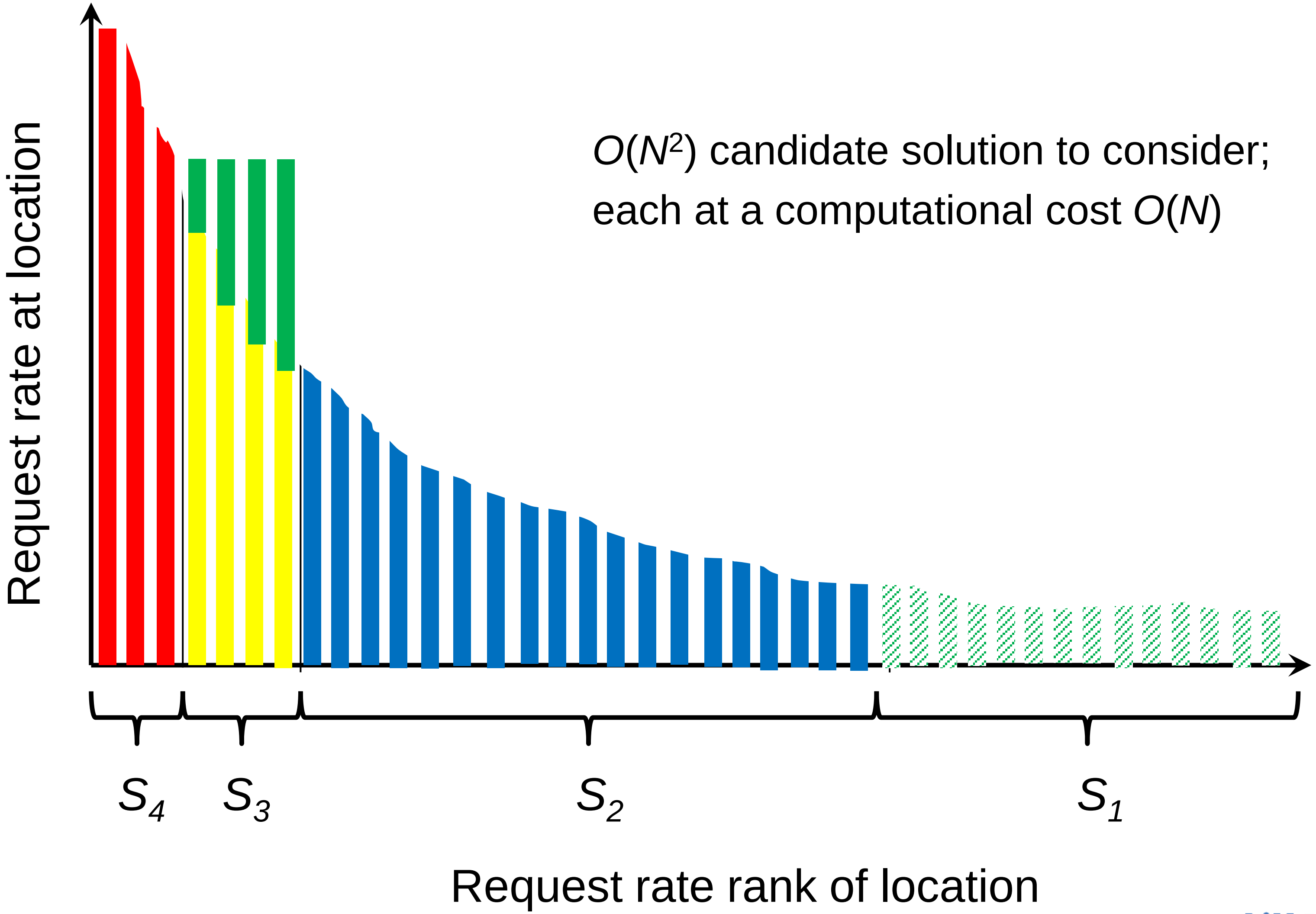


Properties of optimal request routing



Finding the optimal request routing

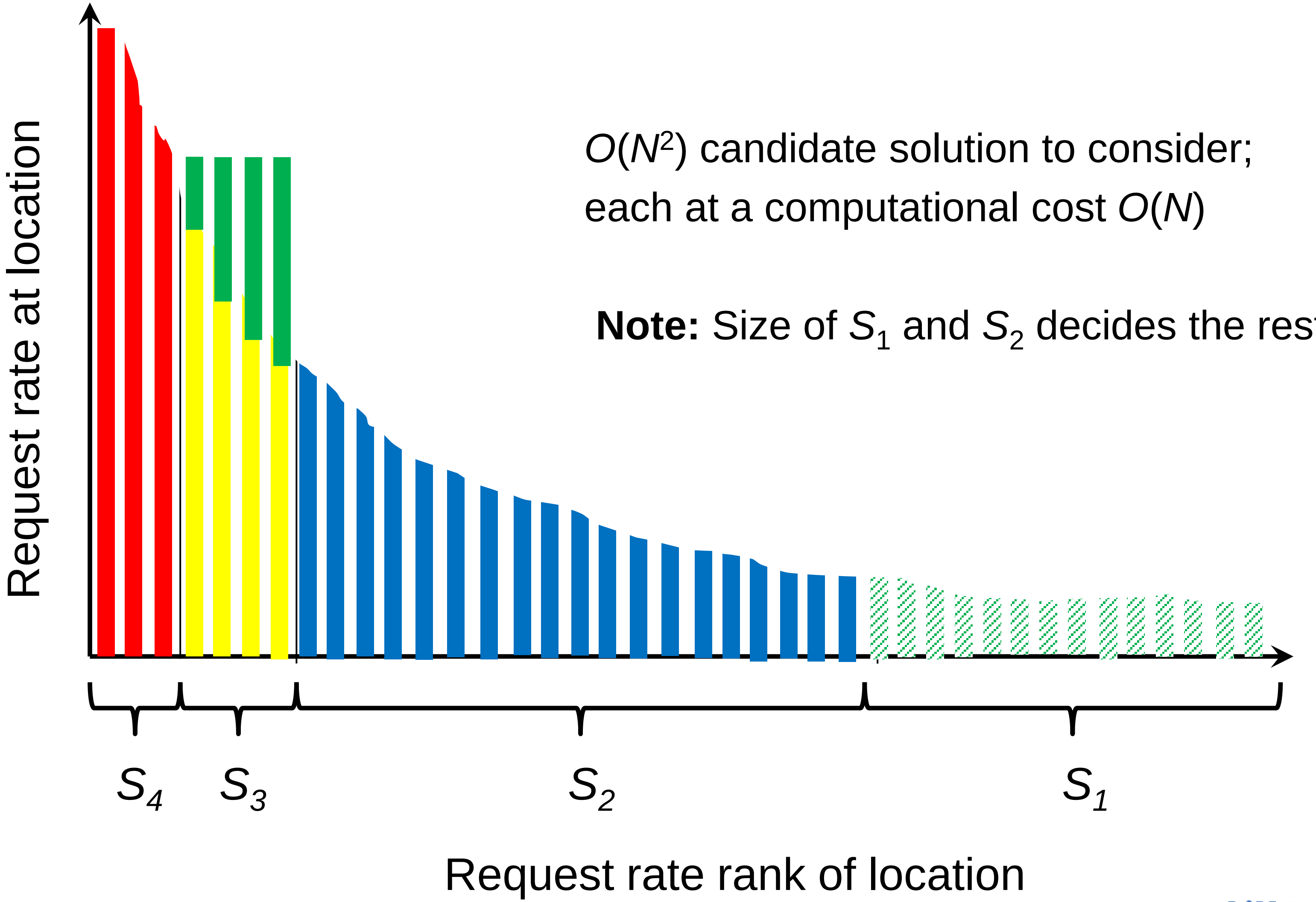
$O(N^2)$ candidate solution to consider; each at a computational cost $O(N)$



Finding the optimal request routing

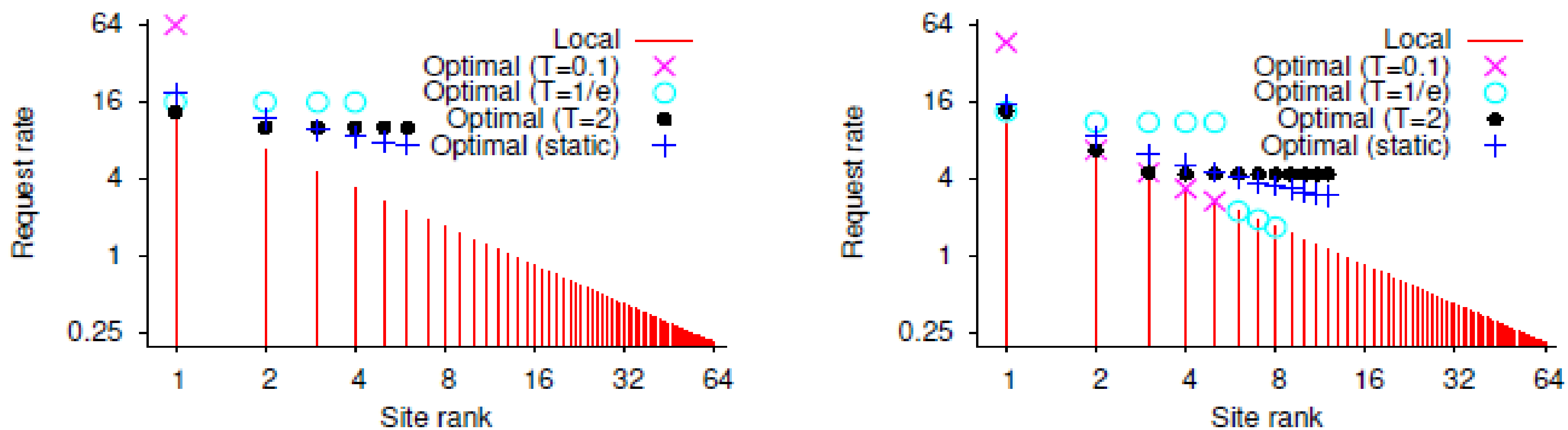
$O(N^2)$ candidate solution to consider; each at a computational cost $O(N)$

Note: Size of S_1 and S_2 decides the rest



Example characteristics

- Properties of optimal solution; e.g.,
  - Shorter T, less servers cache
  - Larger R, more servers cache



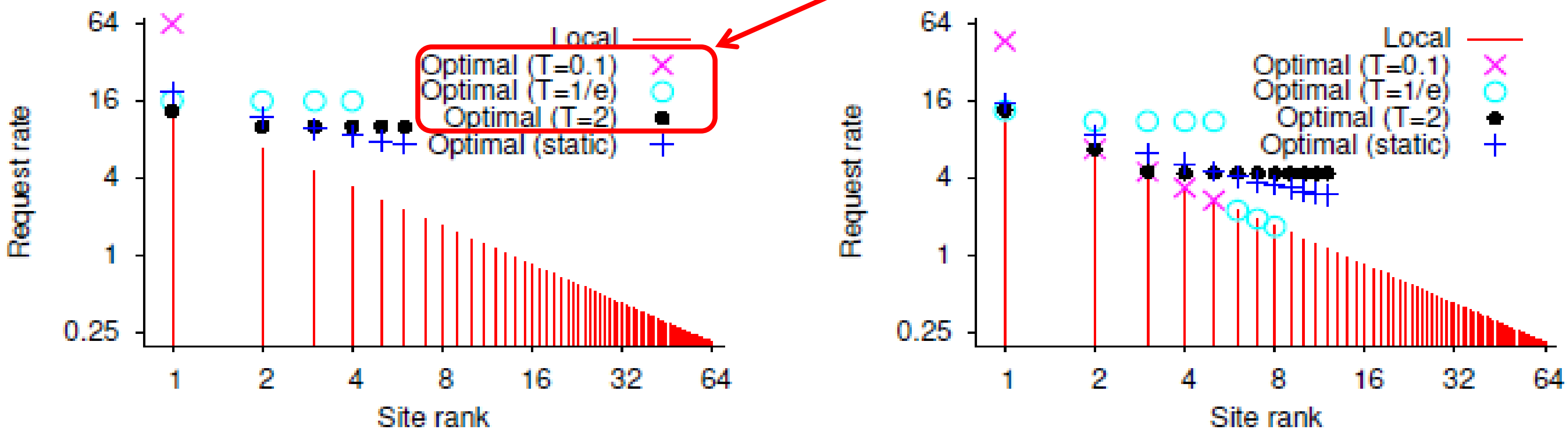
(a) Default ($R = 0.5$) (b) High remote routing cost ($R = 0.9$)

Figure 3: Rates of requests directed to each server with different policies. (Default: $\lambda = 1$, $\alpha = 1$, $N = 64$, $L = 1$.)

Example characteristics

- Properties of optimal solution; e.g.,
  - Shorter T, less servers cache
  - Larger R, more servers cache

T=0.1: 1 active
T=1/e: 4 active
T=2: 6 active



(a) Default ($R = 0.5$)

(b) High remote routing cost ($R = 0.9$)

Figure 3: Rates of requests directed to each server with different policies. (Default: $\lambda = 1$, $\alpha = 1$, $N = 64$, $L = 1$.)

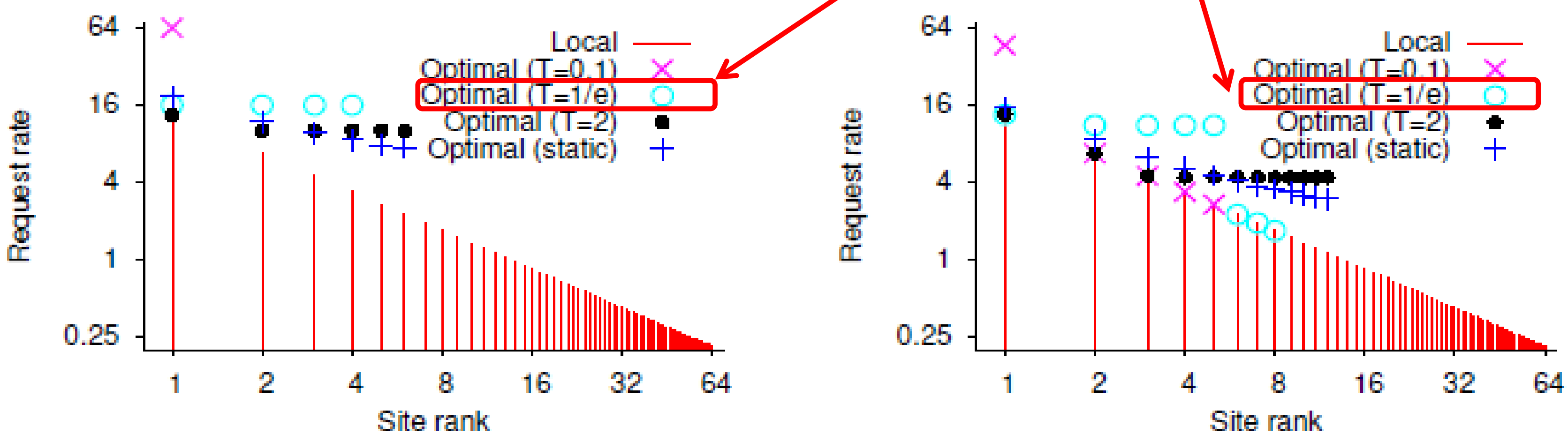
LiU EXPANDING REALITY

Example characteristics

- Properties of optimal solution; e.g.,
  - Shorter T, less servers cache
  - Larger R, more servers cache

R=0.5: 4 active

R=0.9: 8 active



(a) Default ($R = 0.5$)

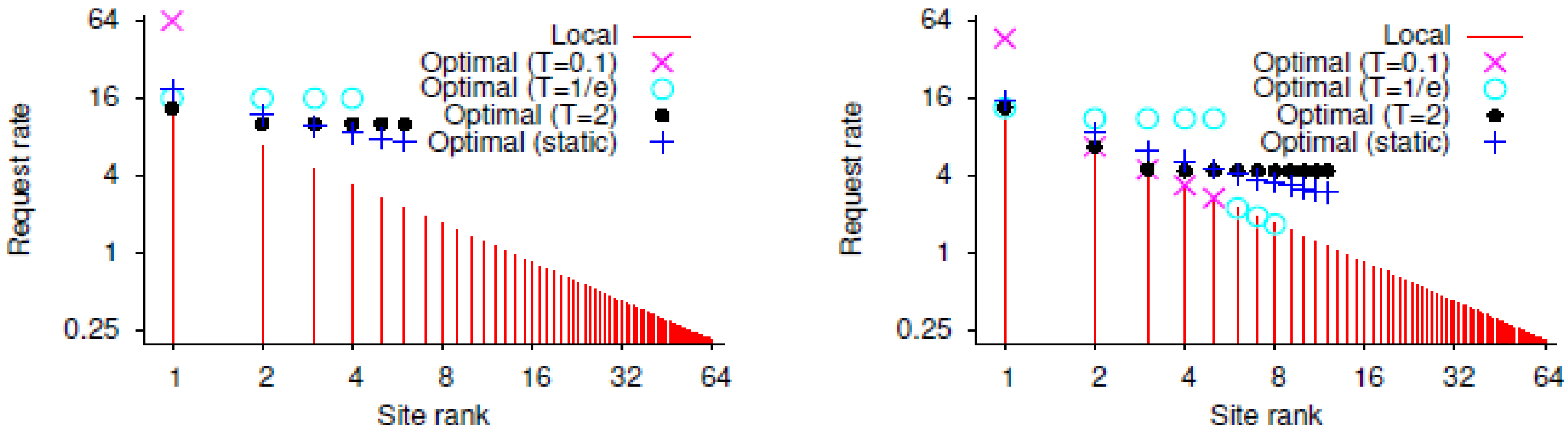
(b) High remote routing cost ($R = 0.9$)

Figure 3: Rates of requests directed to each server with different policies. (Default: $\lambda = 1$, $\alpha = 1$, $N = 64$, $L = 1$.)

LiU EXPANDING REALITY

Example characteristics

- Properties of optimal solution; e.g.,
  - Shorter T, less servers cache
  - Larger R, more servers cache

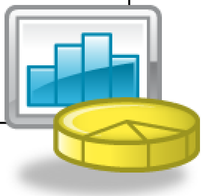


(a) Default ($R = 0.5$) (b) High remote routing cost ($R = 0.9$)

Figure 3: Rates of requests directed to each server with different policies. (Default: $\lambda = 1$, $\alpha = 1$, $N = 64$, $L = 1$.)

Cost Breakdown

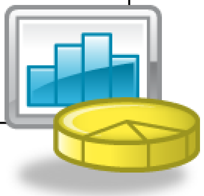
- Breakdown of costs into
  - Cache miss cost
  - Cache storage cost
  - Remote routing cost



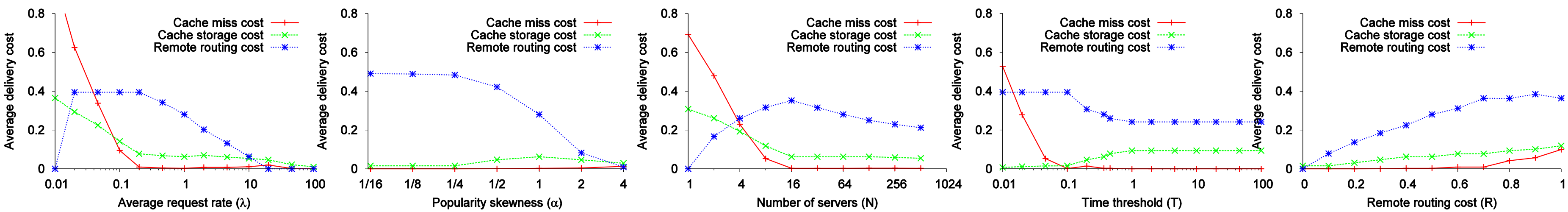
- Characterization when varying
  - request rate
  - load skew
  - number of servers
  - TTL threshold
  - remote routing cost

Cost Breakdown

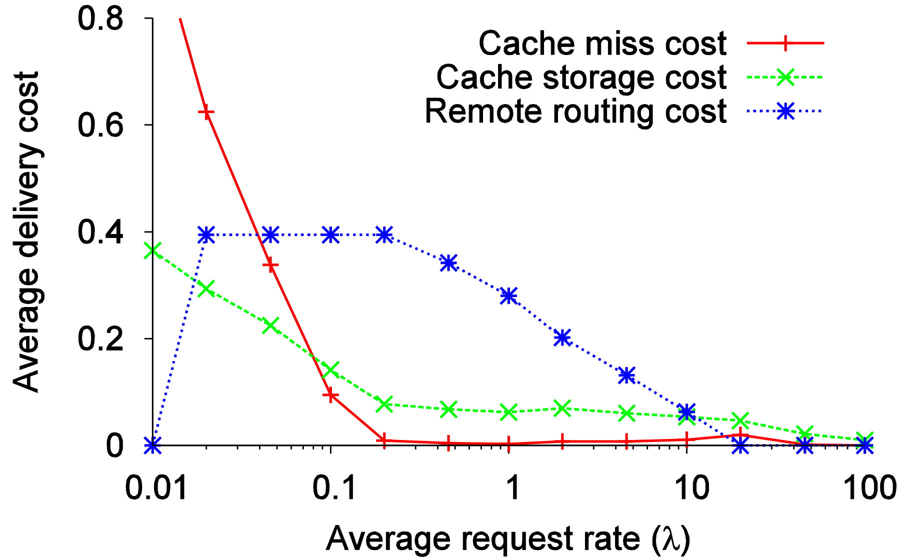
- Breakdown of costs into
  - Cache miss cost
  - Cache storage cost
  - Remote routing cost



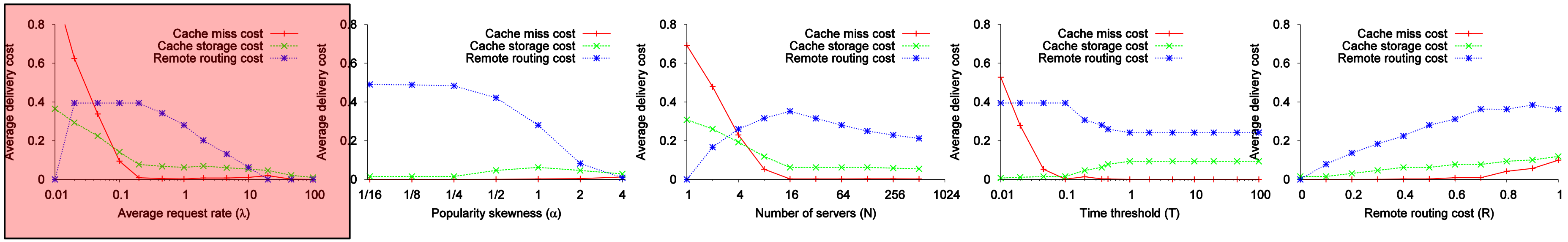
- Characterization when varying
  - request rate
  - load skew
  - number of servers
  - TTL threshold
  - remote routing cost



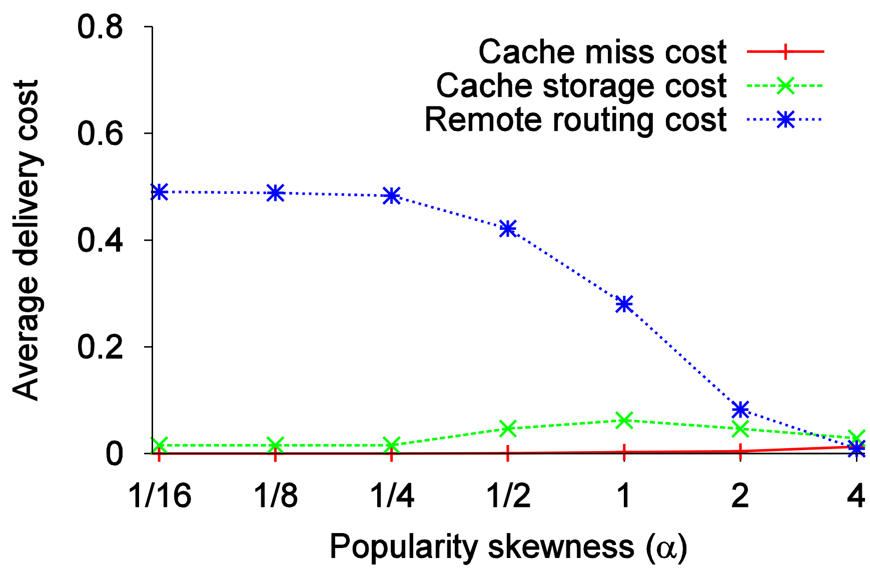
Cost Breakdown



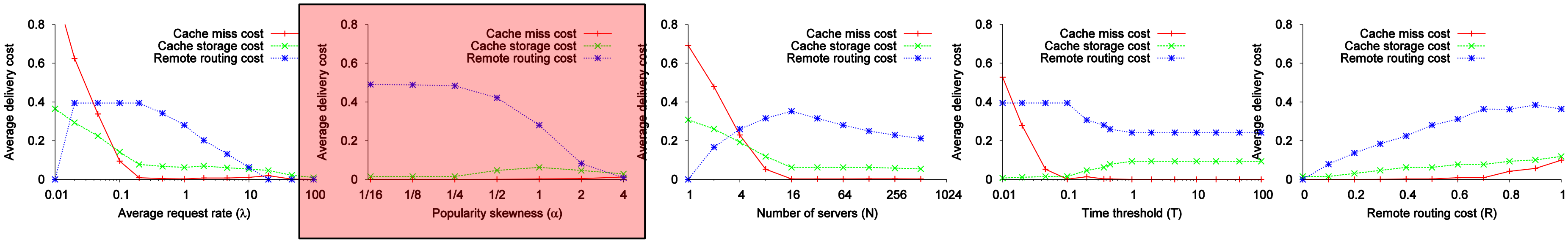
- Characterization when varying
  - request rate
  - load skew
  - number of servers
  - TTL threshold
  - remote routing cost



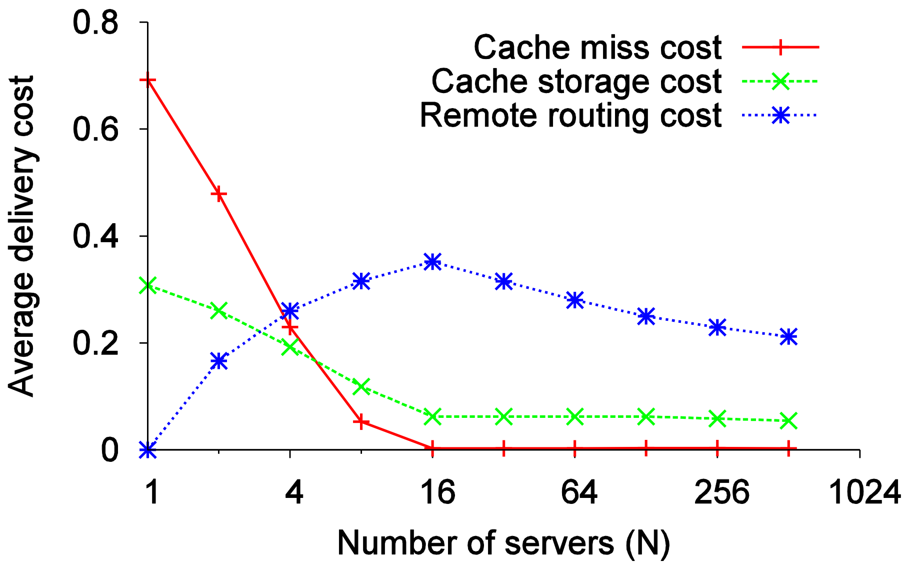
Cost Breakdown



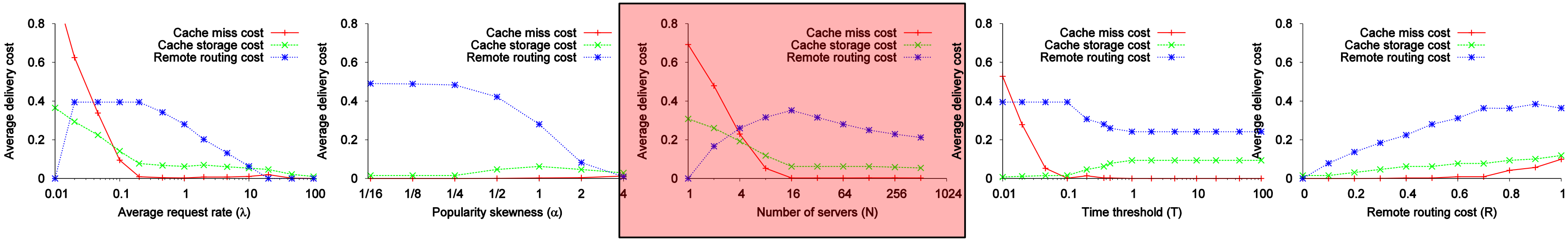
- Characterization when varying
  - request rate
  - load skew
  - number of servers
  - TTL threshold
  - remote routing cost



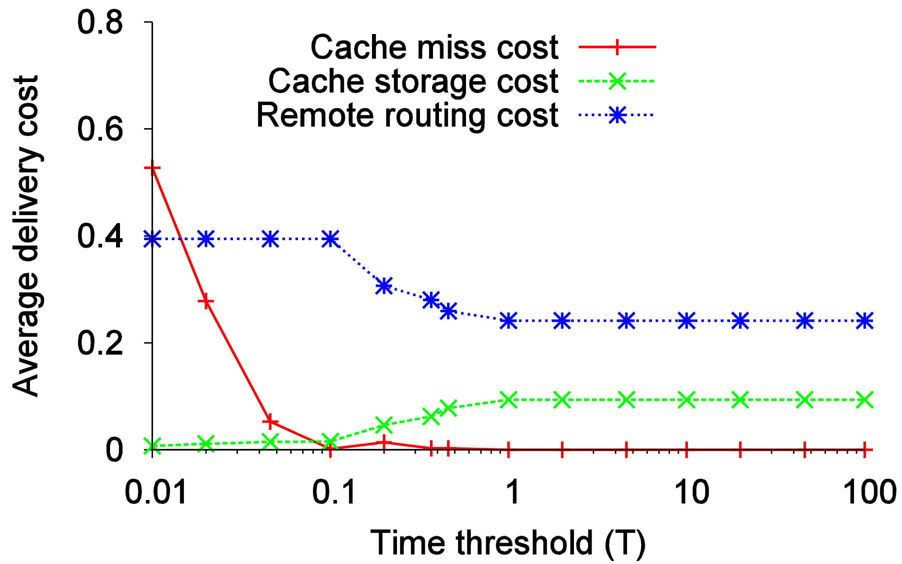
Cost Breakdown



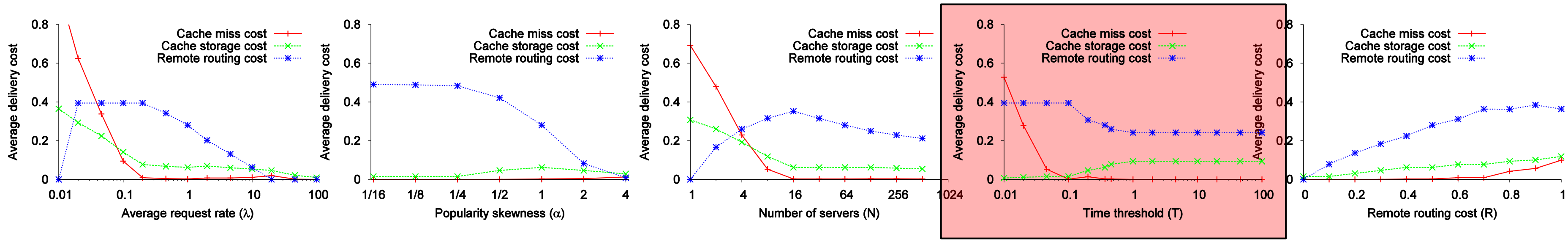
- Characterization when varying
  - request rate
  - load skew
  - number of servers
  - TTL threshold
  - remote routing cost



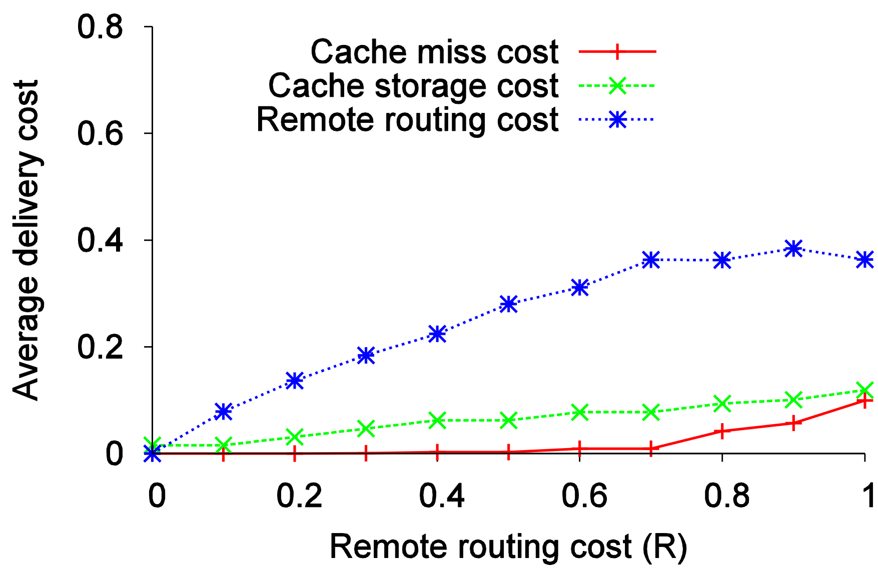
Cost Breakdown



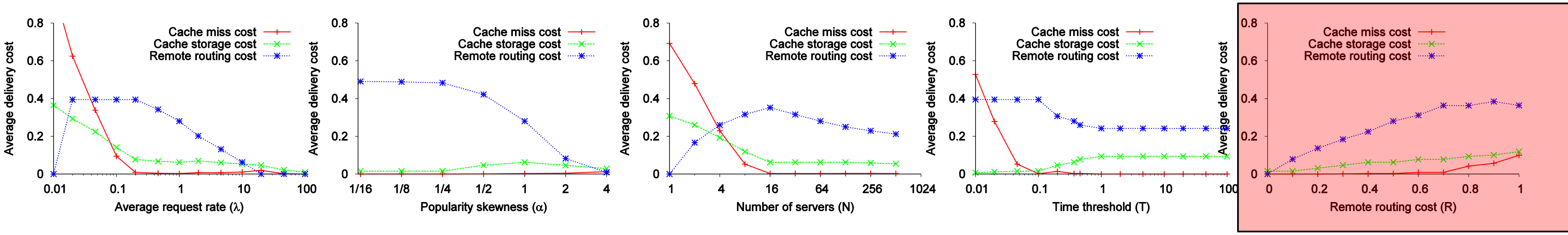
- Characterization when varying
  - request rate
  - load skew
  - number of servers
  - TTL threshold
  - remote routing cost



Cost Breakdown



- Characterization when varying
  - request rate
  - load skew
  - number of servers
  - TTL threshold
  - remote routing cost

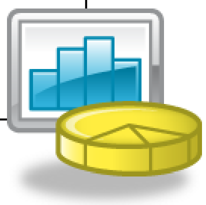


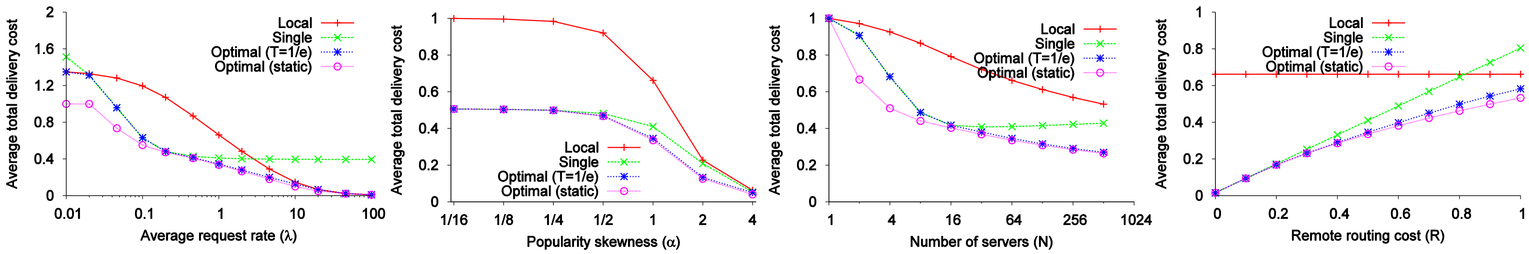
Cost Comparison

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

Cost Comparison

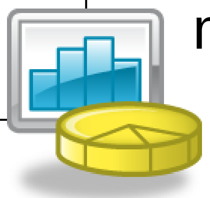
- Compare optimal dynamic policy with baselines
  - Always “local” server
  - Always “single” server
- As well as with optimal “static” placement (any T_i)



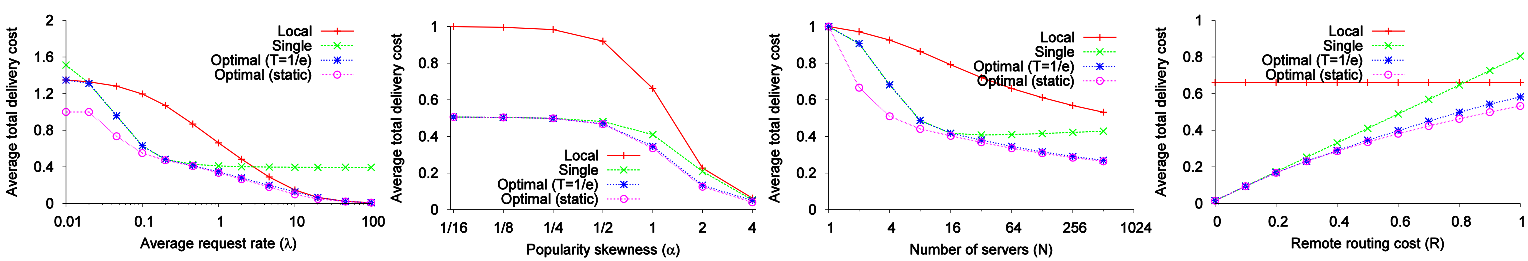


Cost Comparison

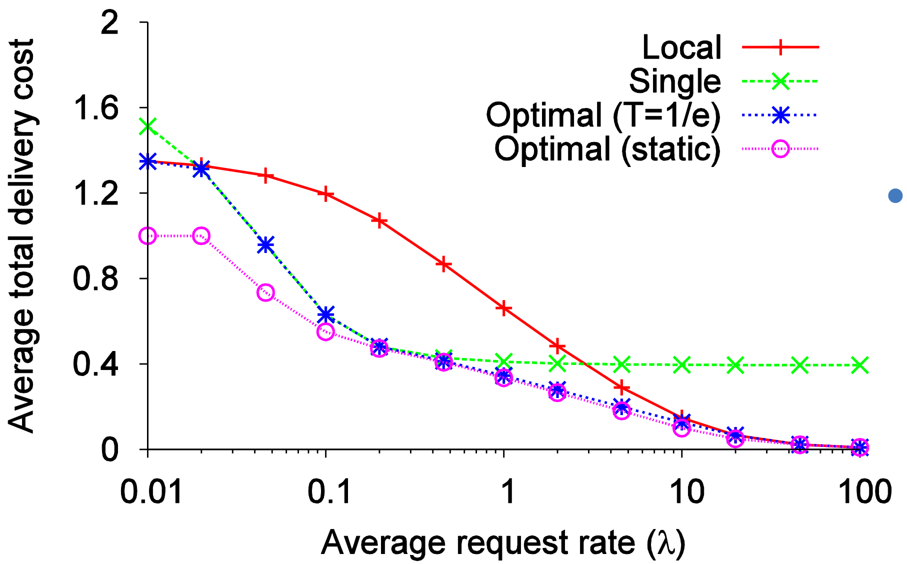
- Compare optimal dynamic policy with baselines
  - Always “local” server
  - Always “single” server
- As well as with optimal “static” placement (any T_i)



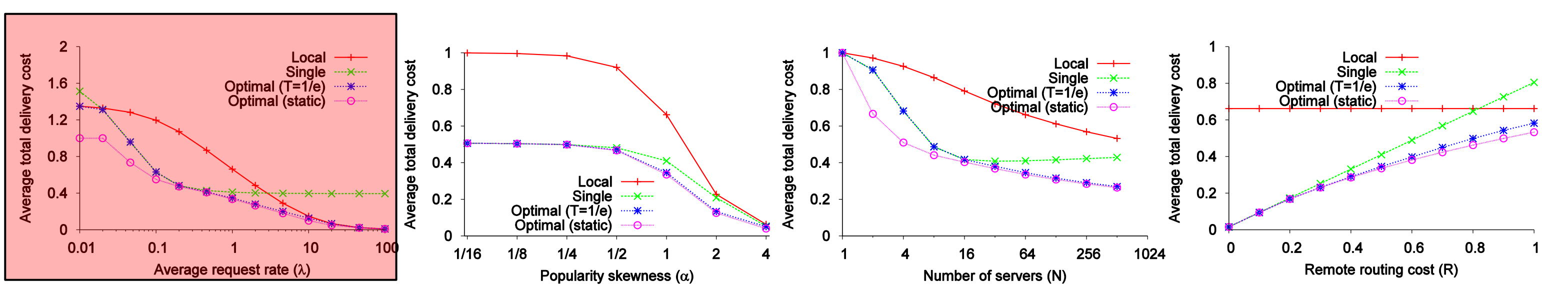
- Significantly outperform baselines (“local” and “single”)
  - Difference can be unbounded
- Even with static load, costs typically close to those with static optimal placement (but much more flexible)



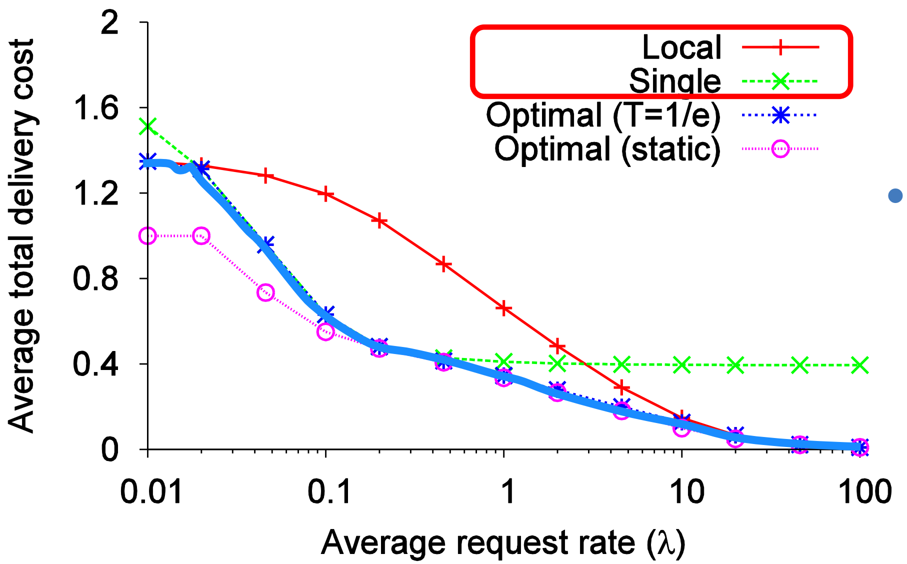
Cost Comparison



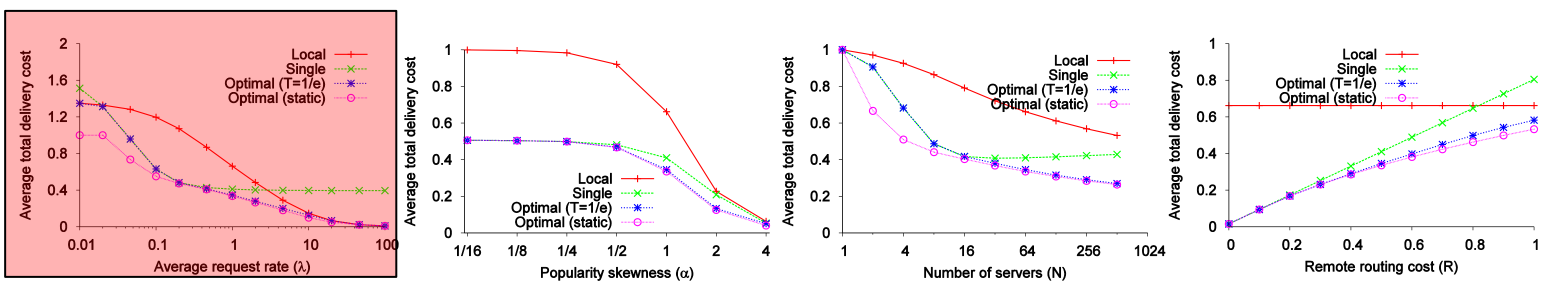
- Significantly outperform baselines (“local” and “single”)
  - Difference can be unbounded
- Even with static load, costs typically close to those with static optimal placement (but much more flexible)



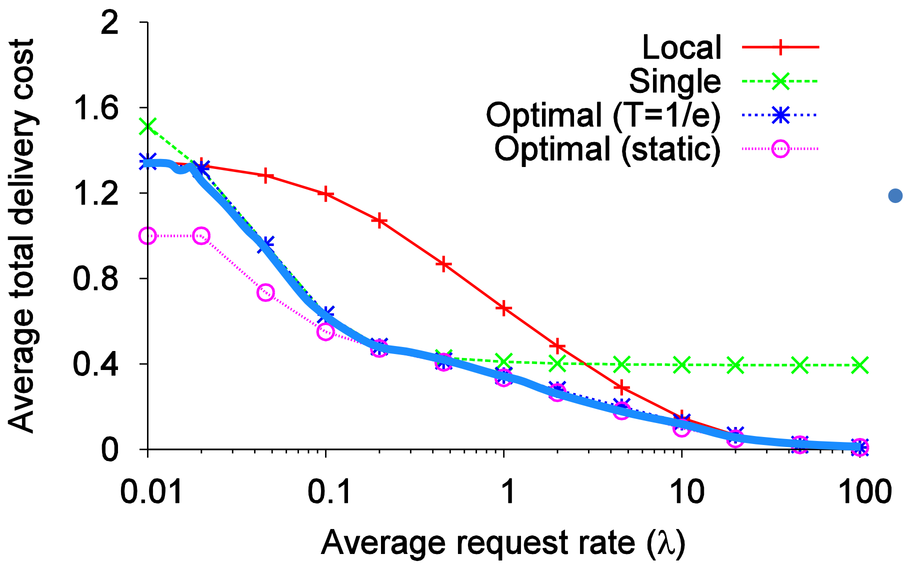
Cost Comparison



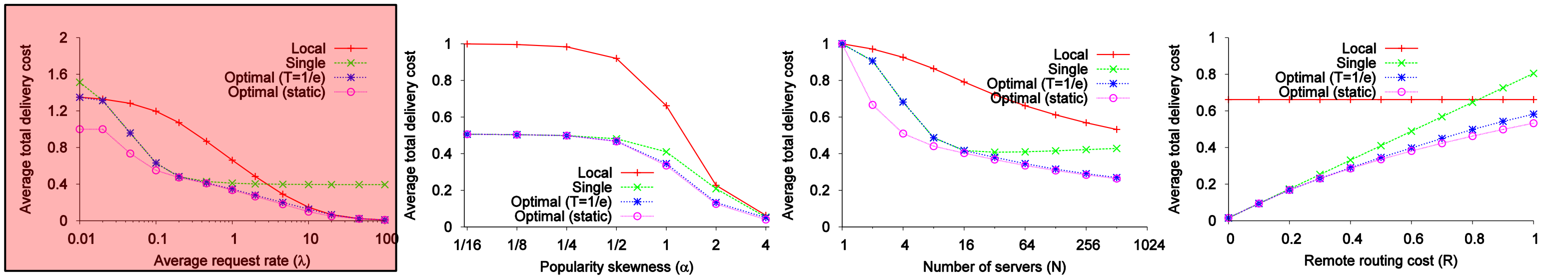
- Significantly outperform baselines ("local" and "single")
  - Difference can be unbounded
- Even with static load, costs typically close to those with static optimal placement (but much more flexible)



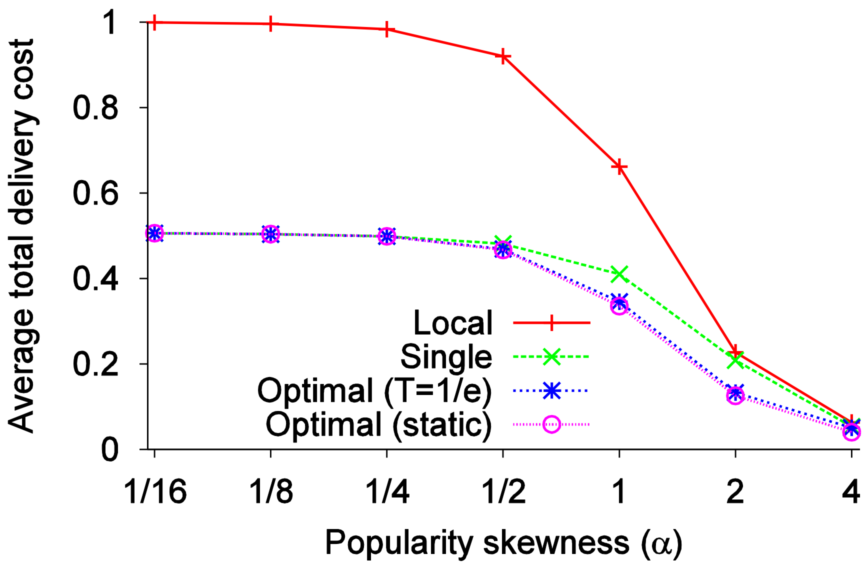
Cost Comparison



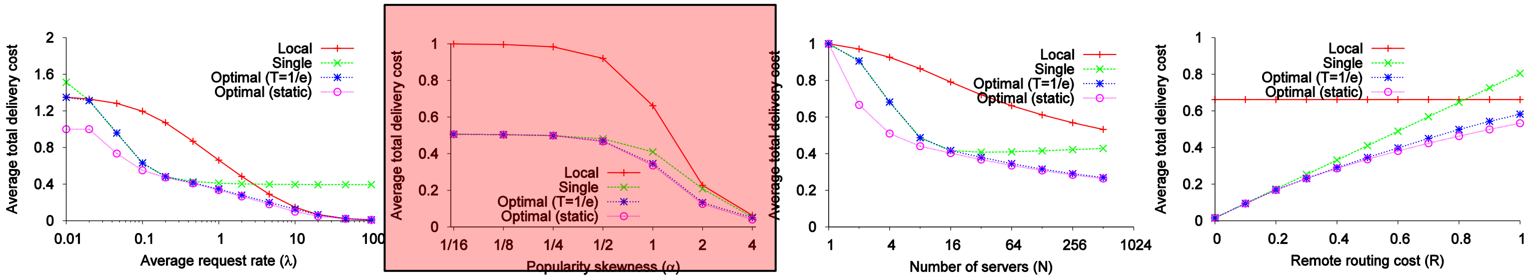
- Significantly outperform baselines ("local" and "single")
  - Difference can be unbounded
- Even with static load, costs typically close to those with static optimal placement (but much more flexible)



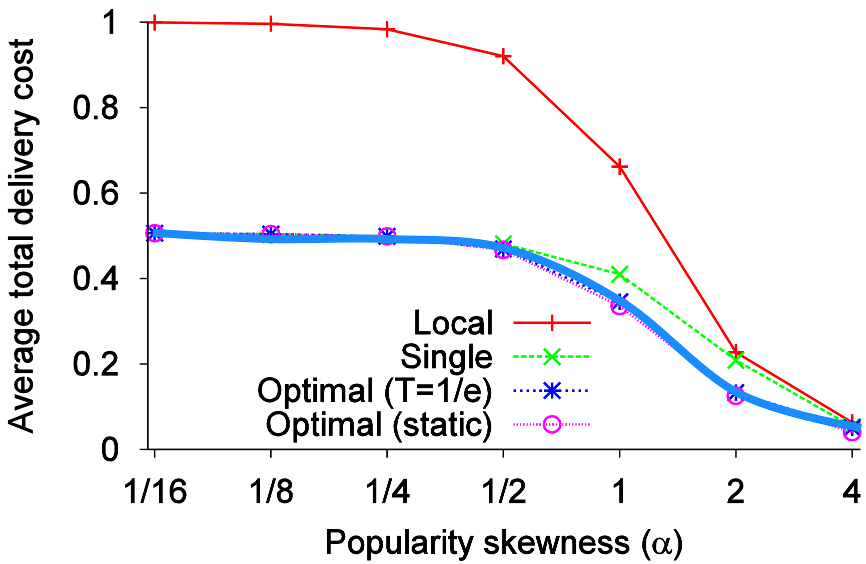
Cost Comparison



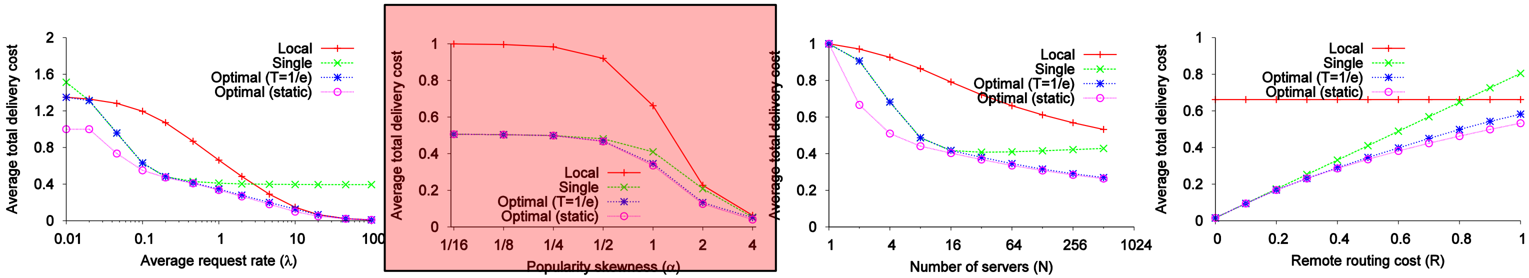
- Significantly outperform baselines (“local” and “single”)
  - Difference can be unbounded
- Even with static load, costs typically close to those with static optimal placement (but much more flexible)



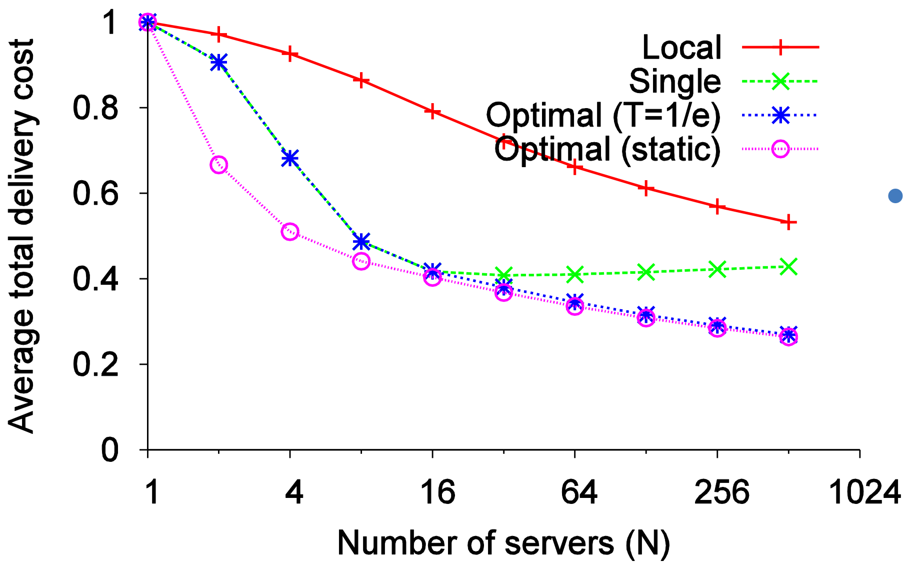
Cost Comparison



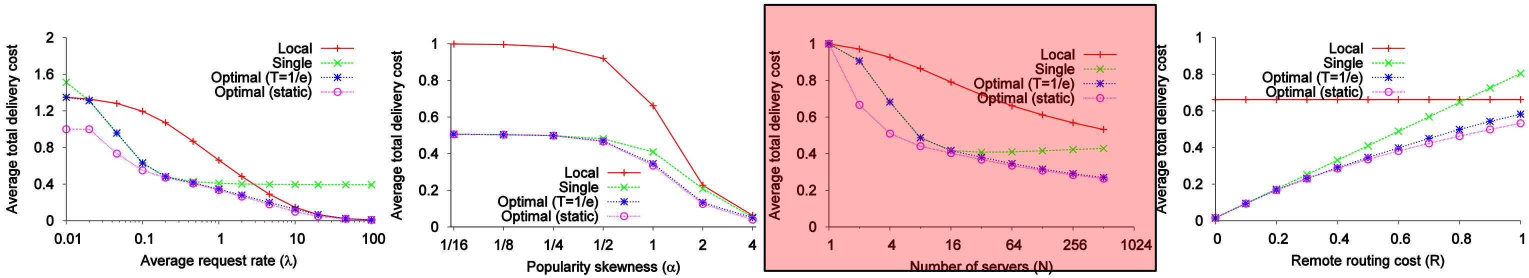
- Significantly outperform baselines ("local" and "single")
  - Difference can be unbounded
- Even with static load, costs typically close to those with static optimal placement (but much more flexible)



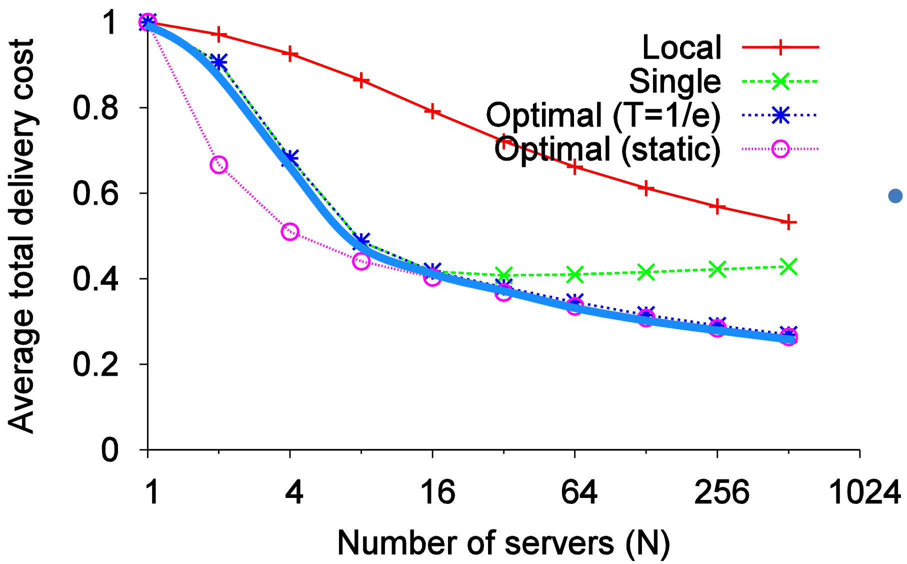
Cost Comparison



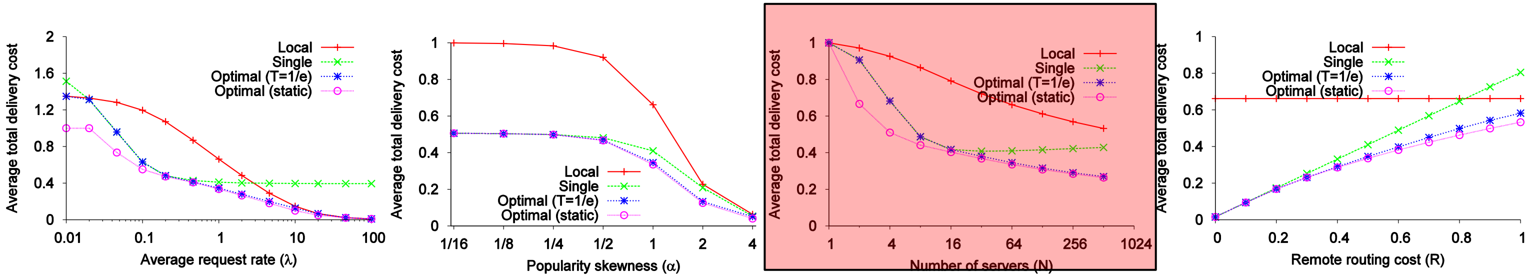
- Significantly outperform baselines ("local" and "single")
  - Difference can be unbounded
- Even with static load, costs typically close to those with static optimal placement (but much more flexible)



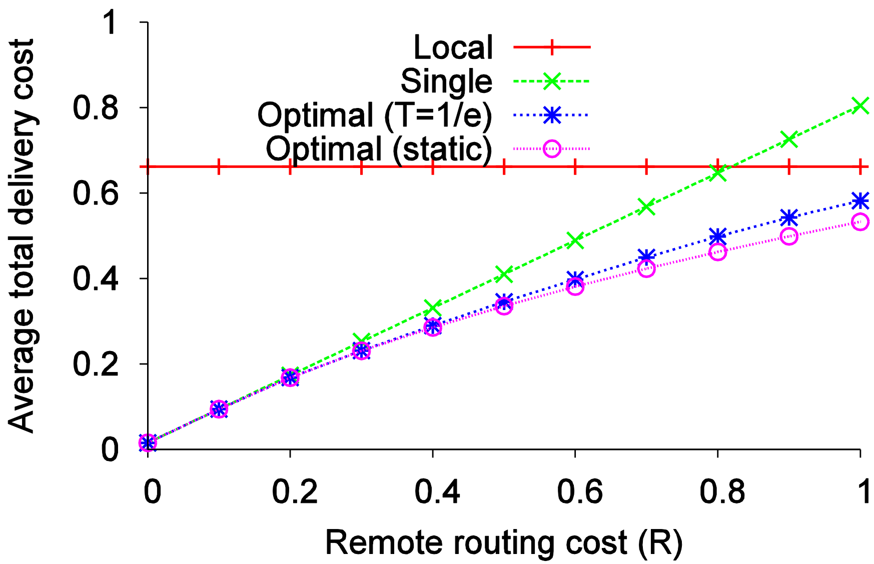
Cost Comparison



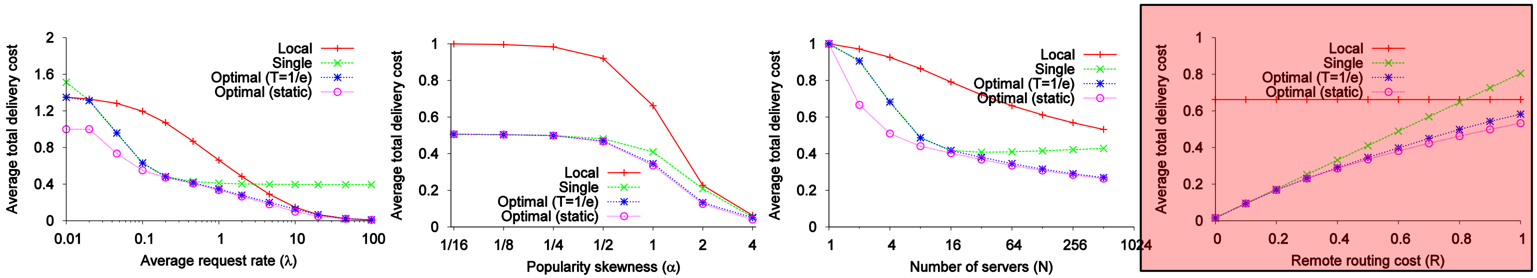
- Significantly outperform baselines (“local” and “single”)
  - Difference can be unbounded
- Even with static load, costs typically close to those with static optimal placement (but much more flexible)



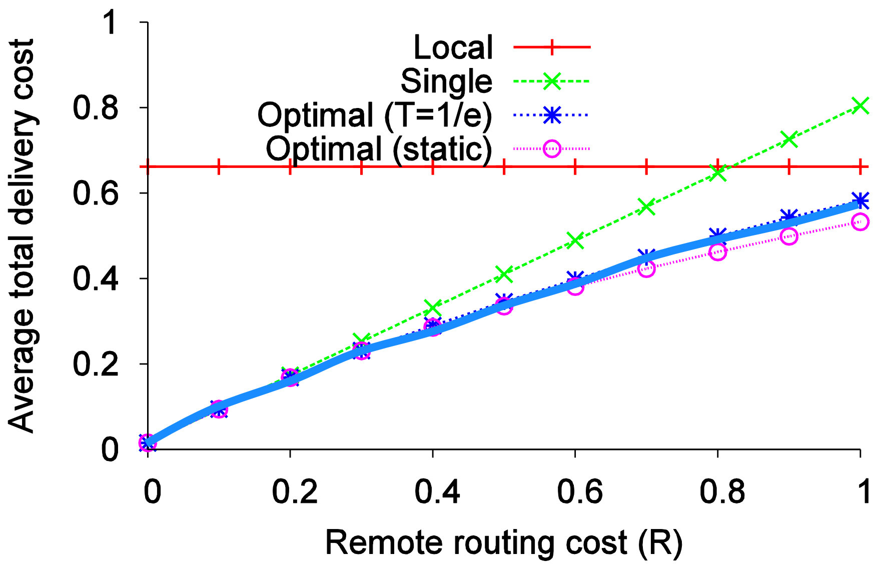
Cost Comparison



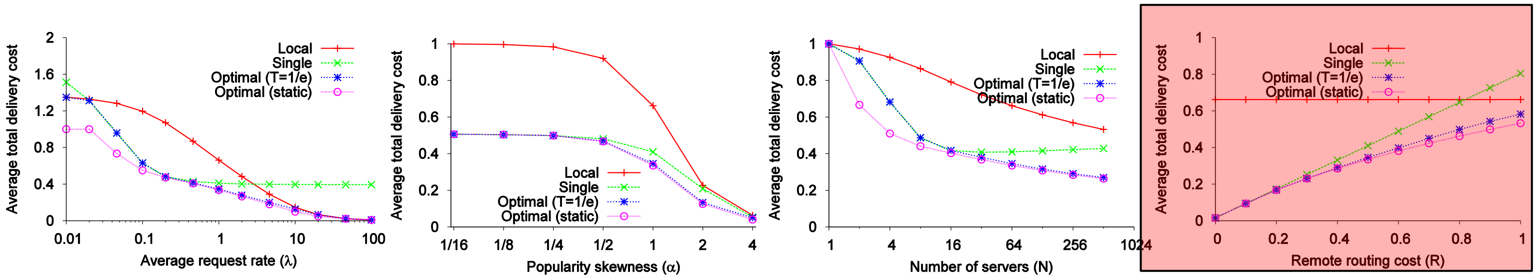
- Significantly outperform baselines (“local” and “single”)
  - Difference can be unbounded
- Even with static load, costs typically close to those with static optimal placement (but much more flexible)



Cost Comparison

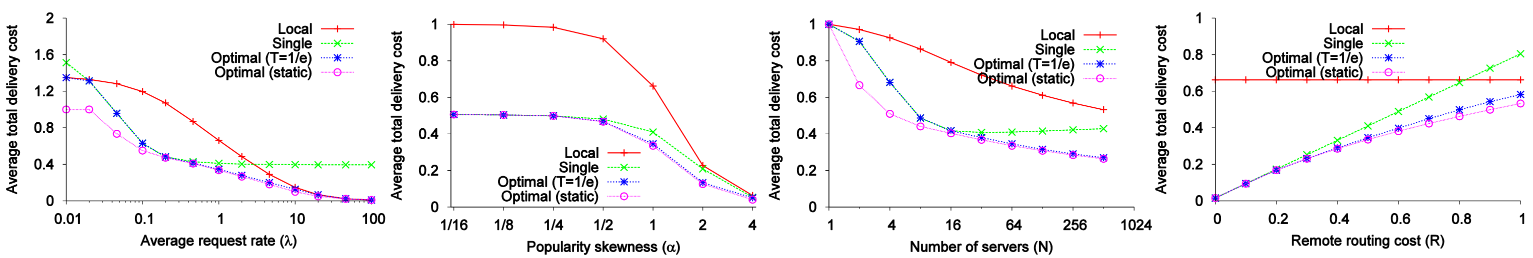


- Significantly outperform baselines ("local" and "single")
  - Difference can be unbounded
- Even with static load, costs typically close to those with static optimal placement (but much more flexible)



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)

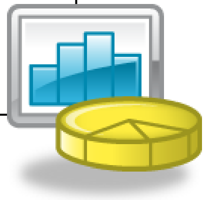


Lower-complexity heuristics

- Two candidate policies
  - Top skewed: Optimal if ignoring miss cost [Theorem 3]
  - Balanced policy: Always assume set S_2 is empty (only three sets to consider)
- Both only need to consider $O(N)$ candidate solutions

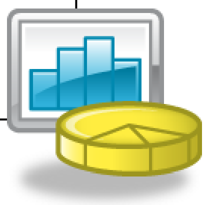
Cost increase comparison

- Calculate increase in costs for
  - Top skewed
  - Balanced
- compared with optimal dynamic policy under different workload settings



Cost increase with top-skewed

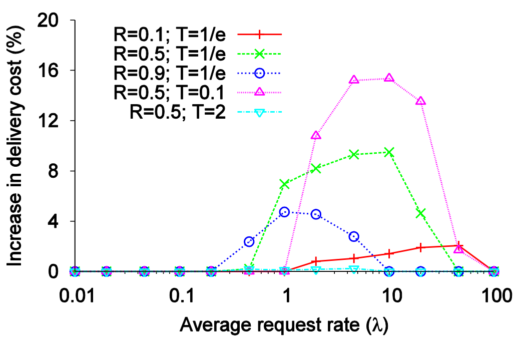
- Calculate increase in costs for
  - Top skewed
  - Balanced
- compared with optimal dynamic policy under different workload settings

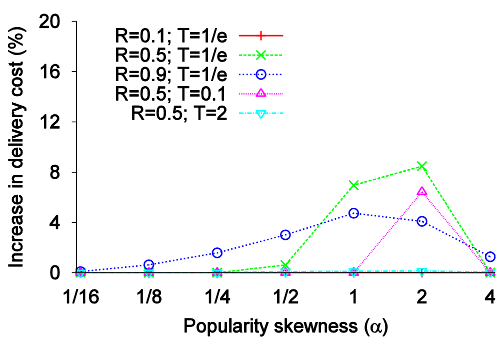


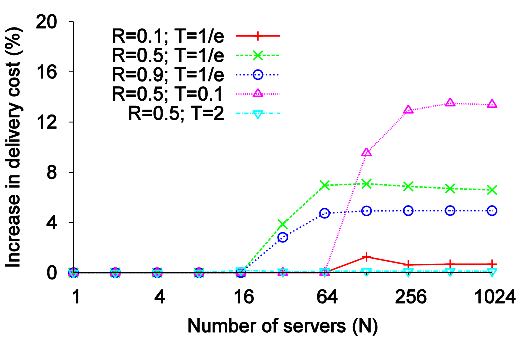
- Up-to 18% increase in cost with “top-skewed”
- Within 2.5% increase in cost with “balanced”

Cost increase with top-skewed

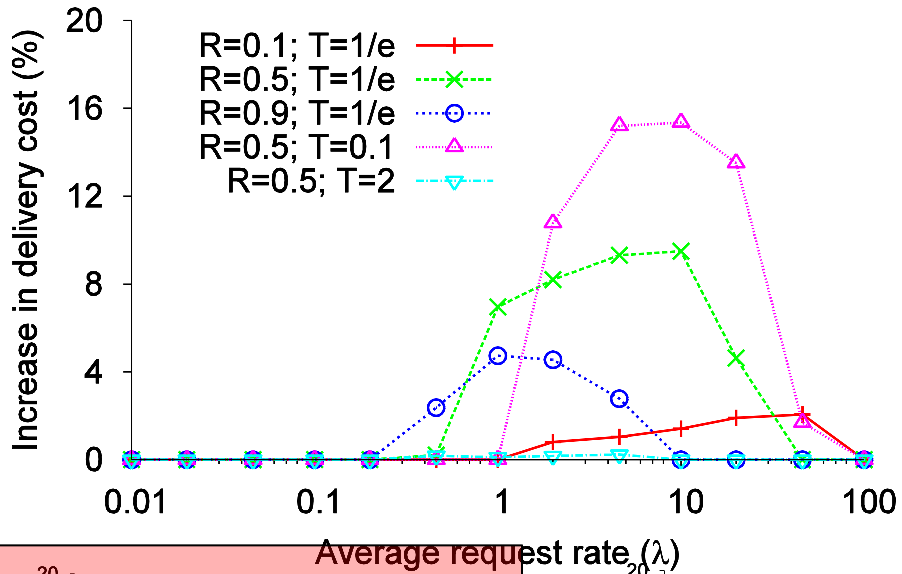
- Up-to 18% increase in cost with “top-skewed”
- Within 2.5% increase in cost with “balanced”



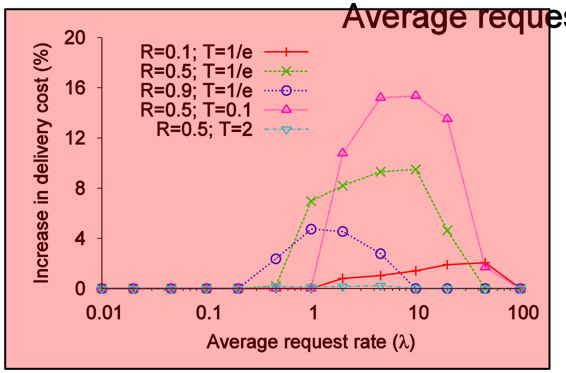


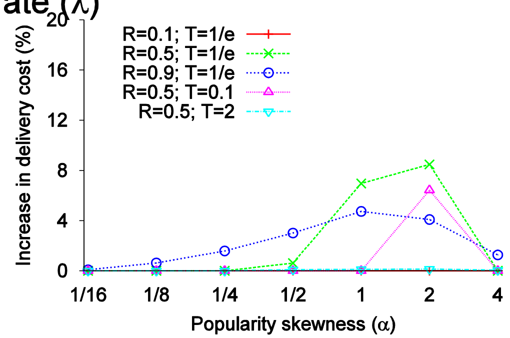


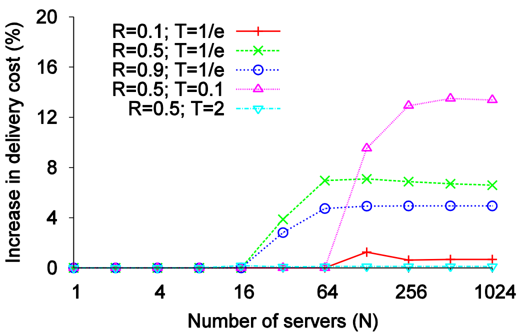
Cost increase with top-skewed



- Up-to 18% increase in cost with "top-skewed"
- Within 2.5% increase in cost with "balanced"

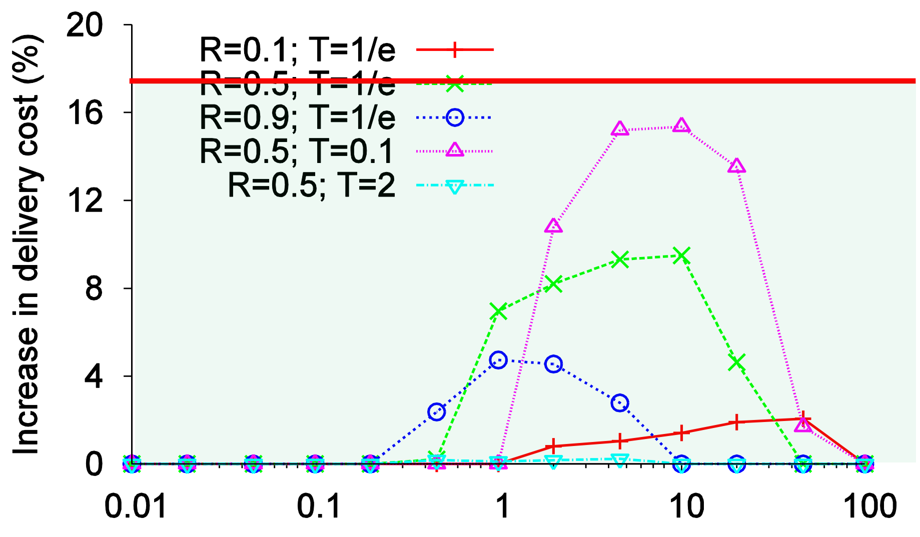


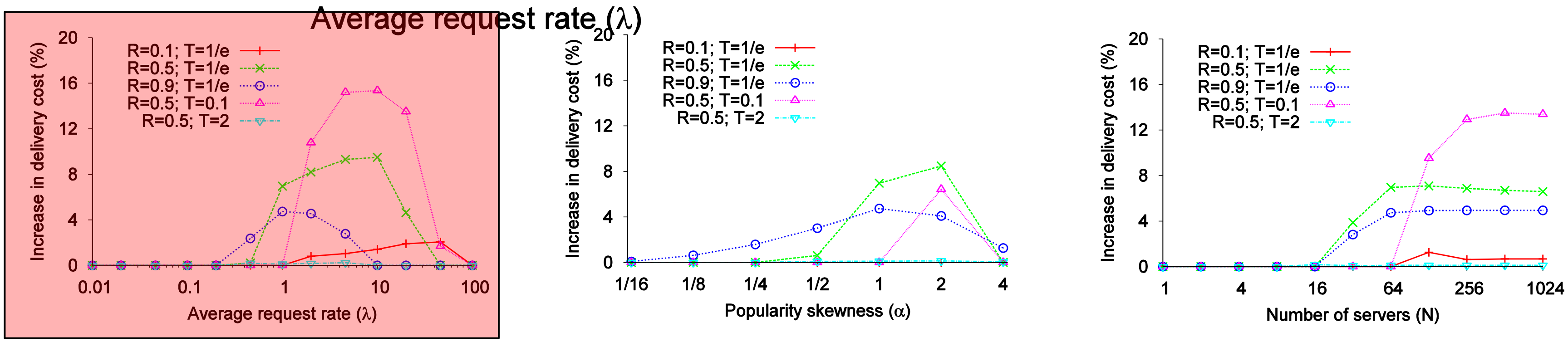




Cost increase with top-skewed

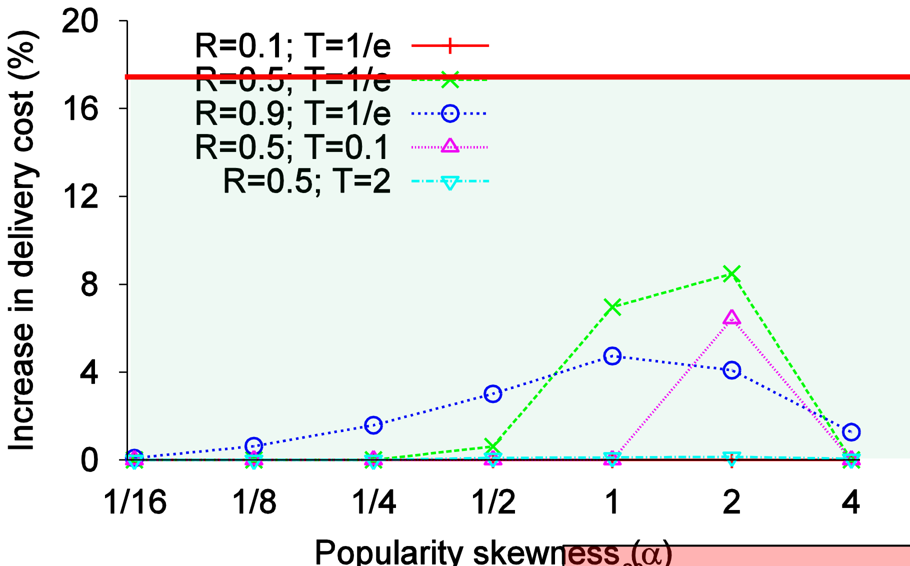
- Up-to 18% increase in cost with "top-skewed"
- Within 2.5% increase in cost with "balanced"

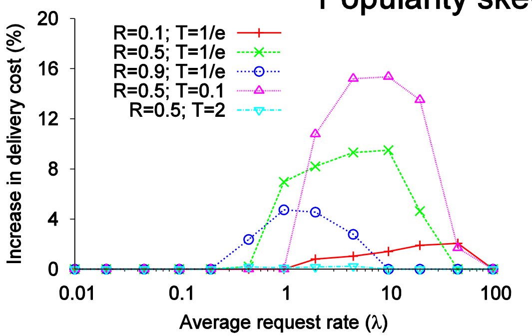


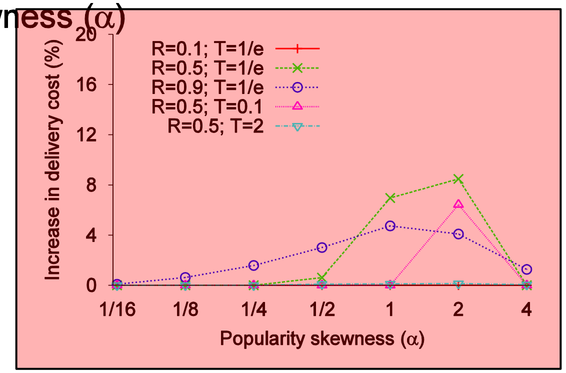


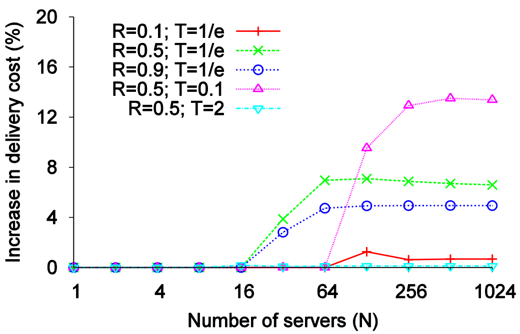
Cost increase with top-skewed

- Up-to 18% increase in cost with "top-skewed"
- Within 2.5% increase in cost with "balanced"



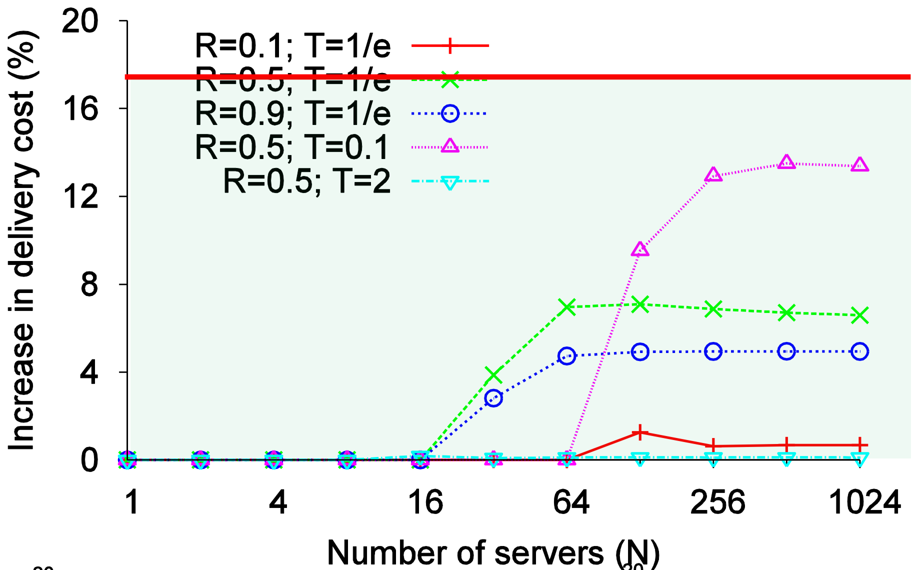


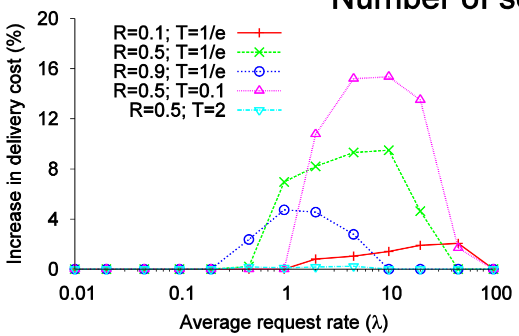


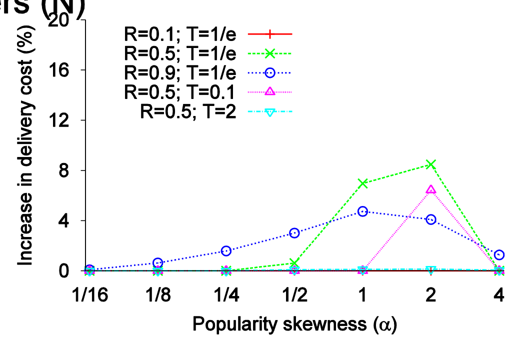


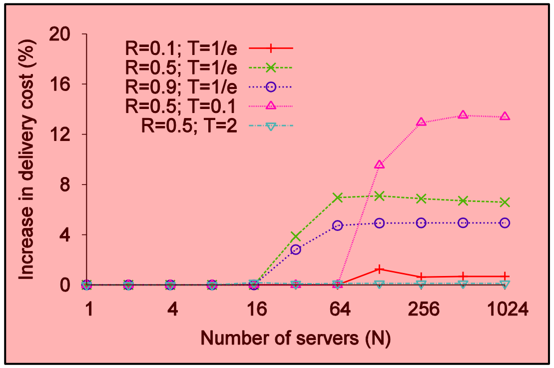
Cost increase with top-skewed

- Up-to 18% increase in cost with “top-skewed”
- Within 2.5% increase in cost with “balanced”



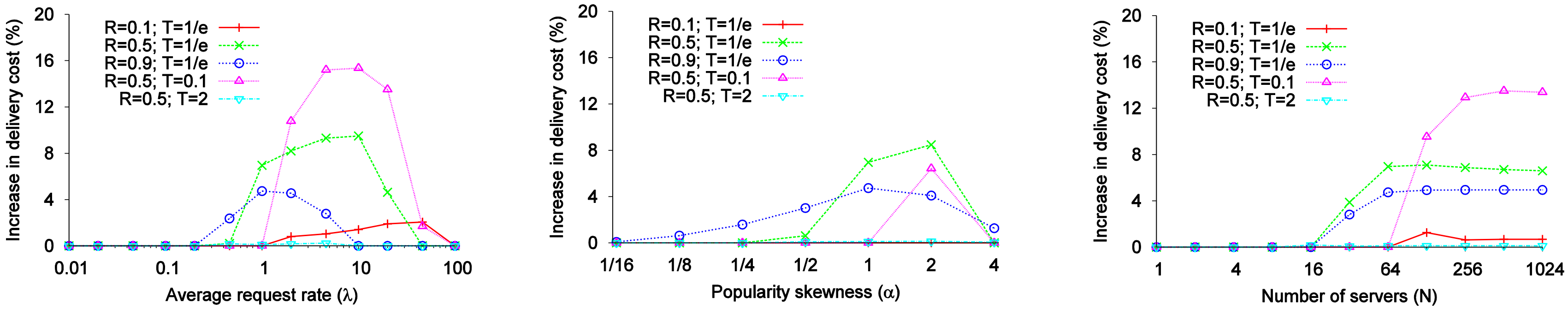






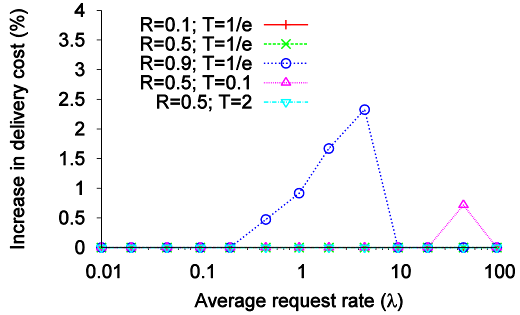
Cost increase with top-skewed

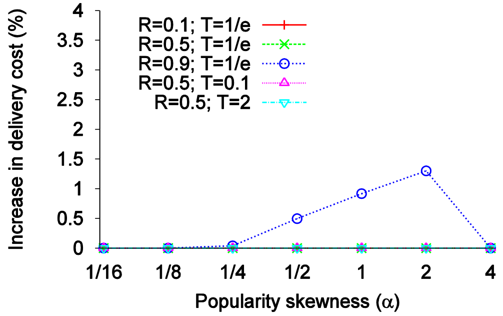
- Up-to 18% increase in cost with "top-skewed"
- Within 2.5% increase in cost with "balanced"

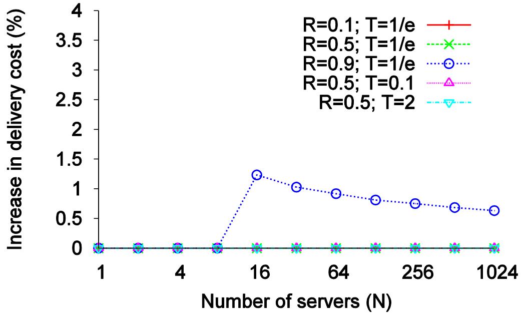


Cost increase with balanced

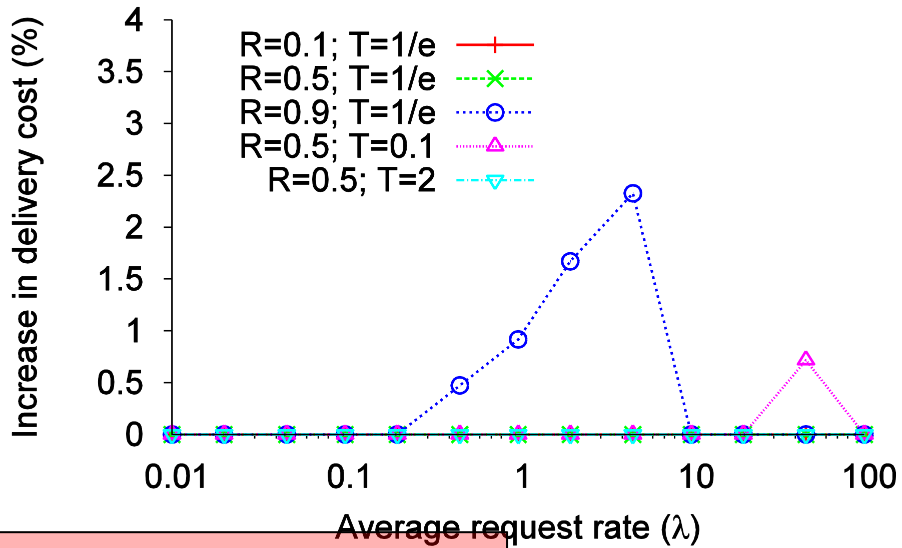
- Up-to 18% increase in cost with “top-skewed”
- Within 2.5% increase in cost with “balanced”



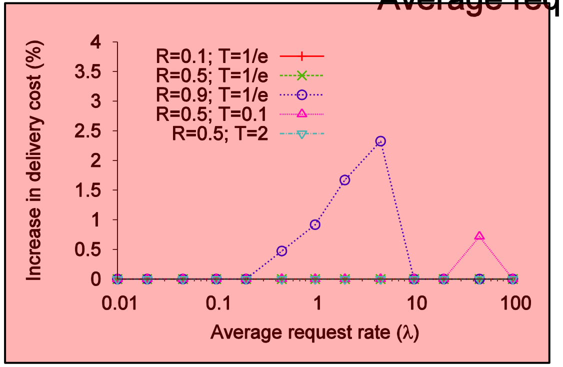


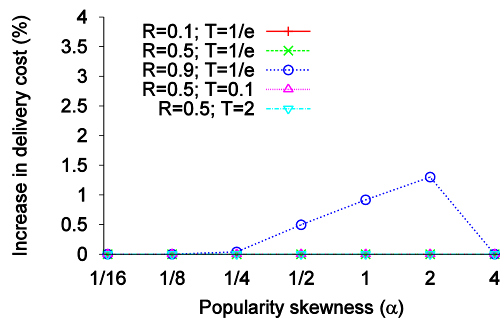


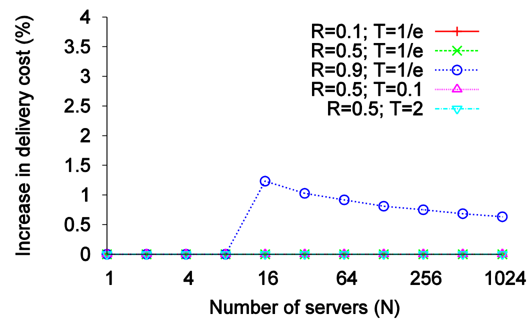
Cost increase with balanced



- Up-to 18% increase in cost with “top-skewed”
- Within 2.5% increase in cost with “balanced”

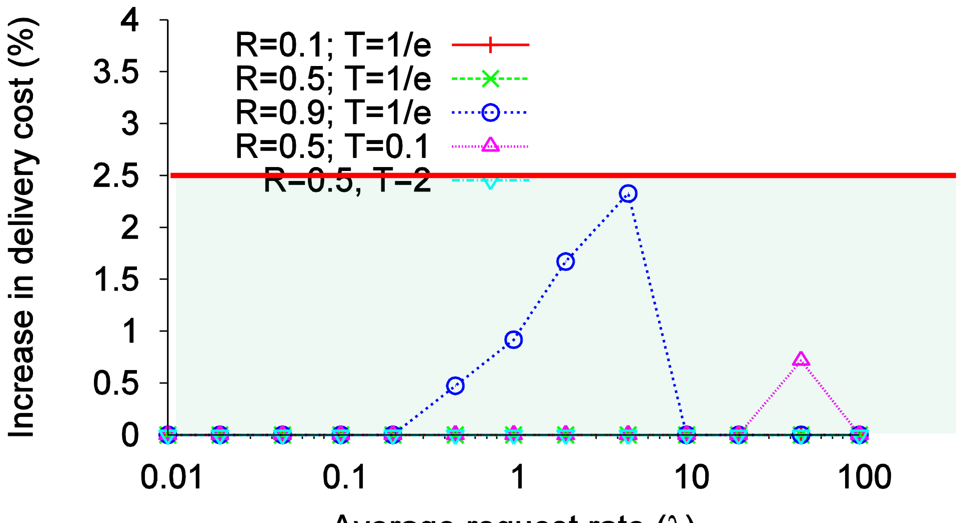


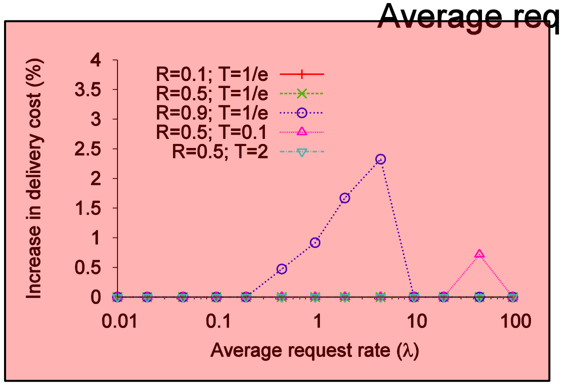


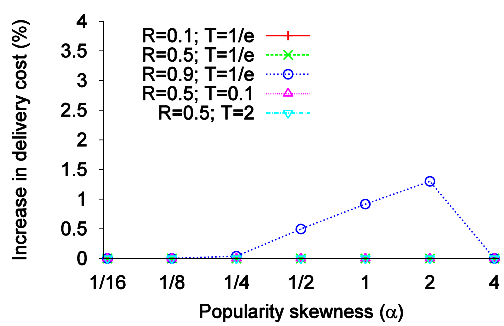


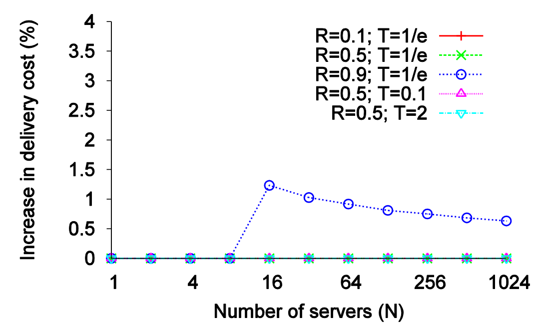
Cost increase with balanced

- Up-to 18% increase in cost with “top-skewed”
- Within 2.5% increase in cost with “balanced”

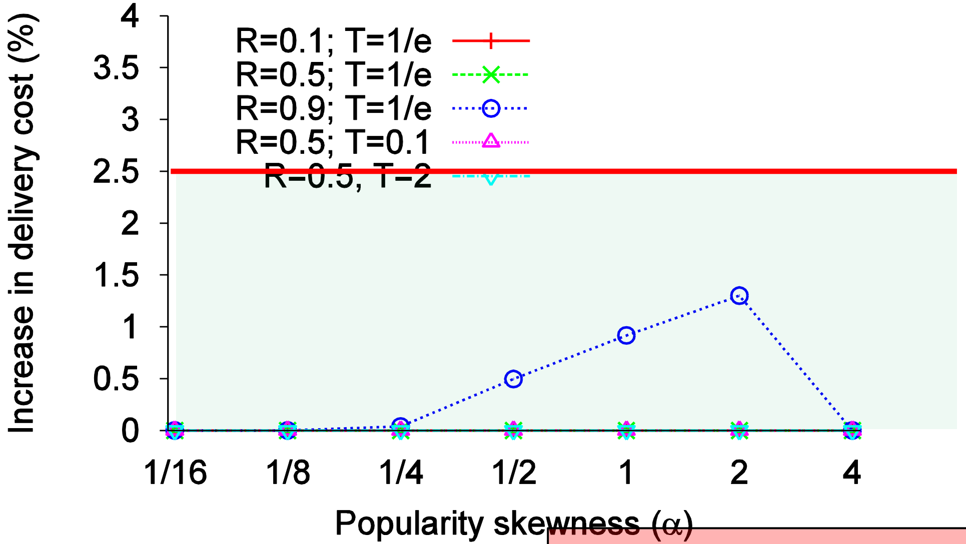




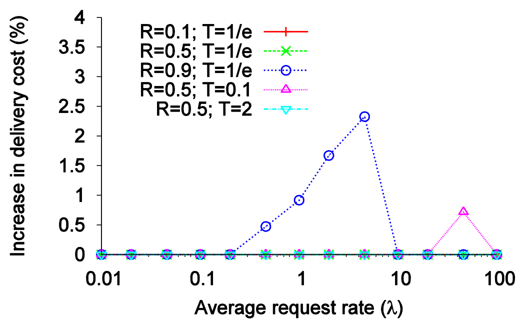


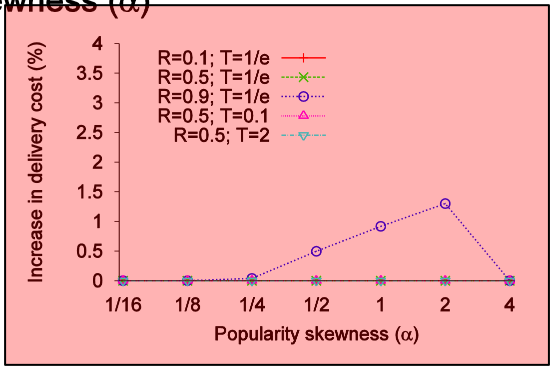


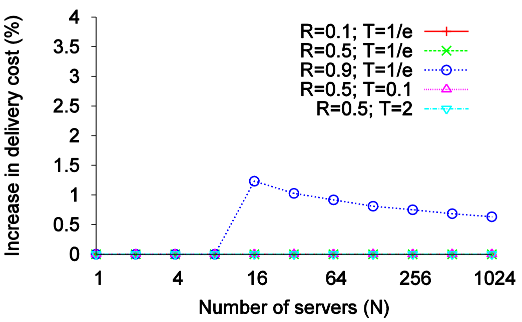
Cost increase with balanced



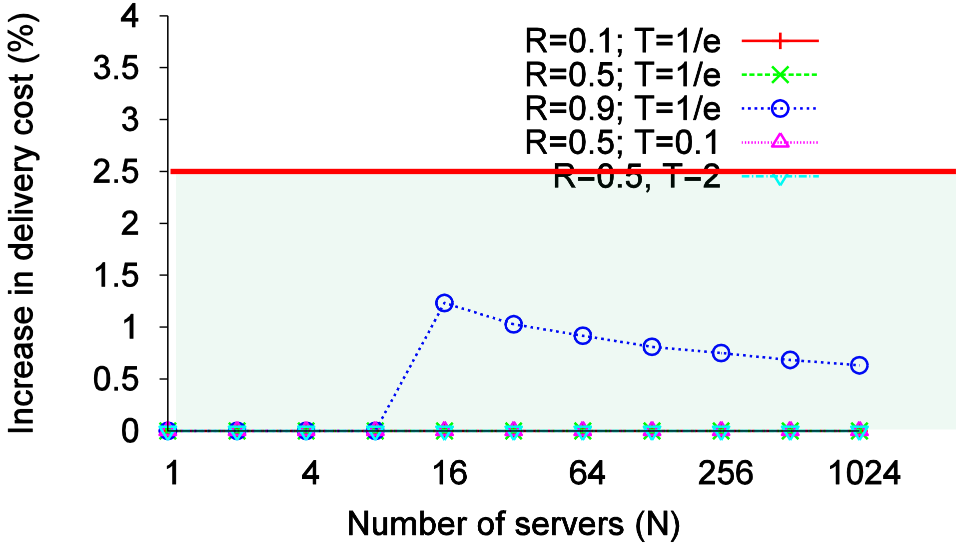
- Up-to 18% increase in cost with “top-skewed”
- Within 2.5% increase in cost with “balanced”



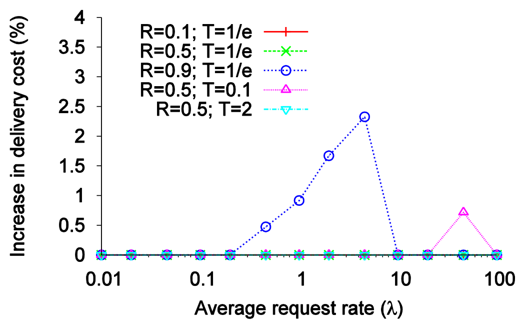


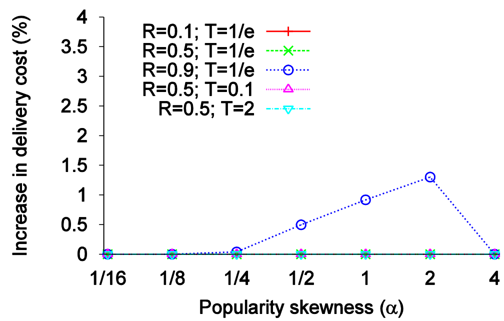


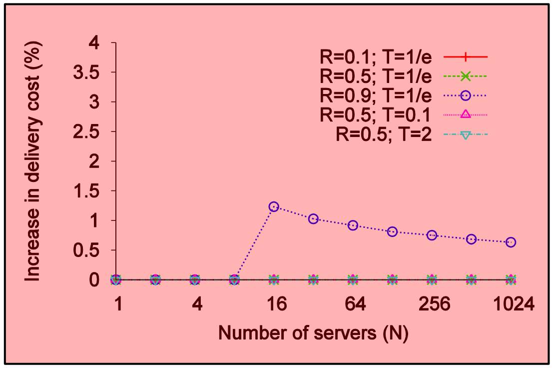
Cost increase with balanced



- Up-to 18% increase in cost with “top-skewed”
- Within 2.5% increase in cost with “balanced”

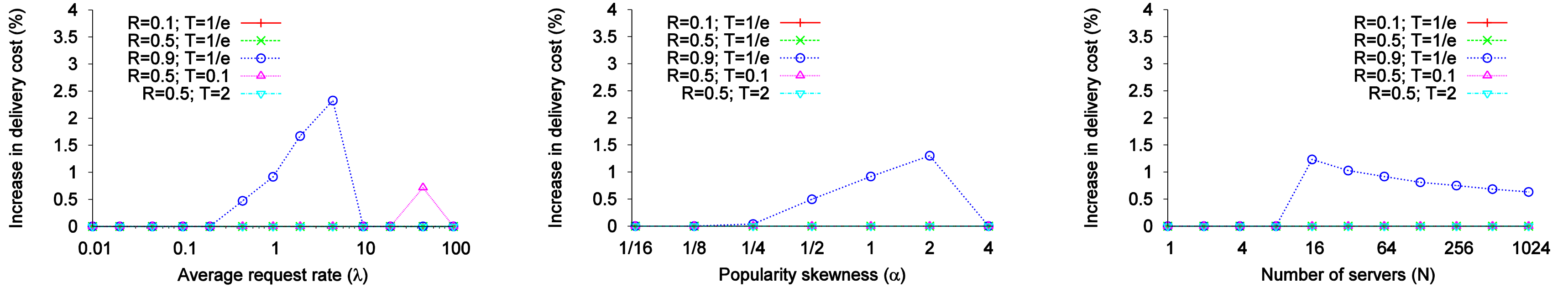






Cost increase with balanced

- Up-to 18% increase in cost with “top-skewed”
- Within 2.5% increase in cost with “balanced”



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

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

Thank you!

***Niklas Carlsson**, Derek Eager, Ajay Gopinathan, and Zongpeng Li*

www.ida.liu.se/~nikca/papers/iperformance14.pdf