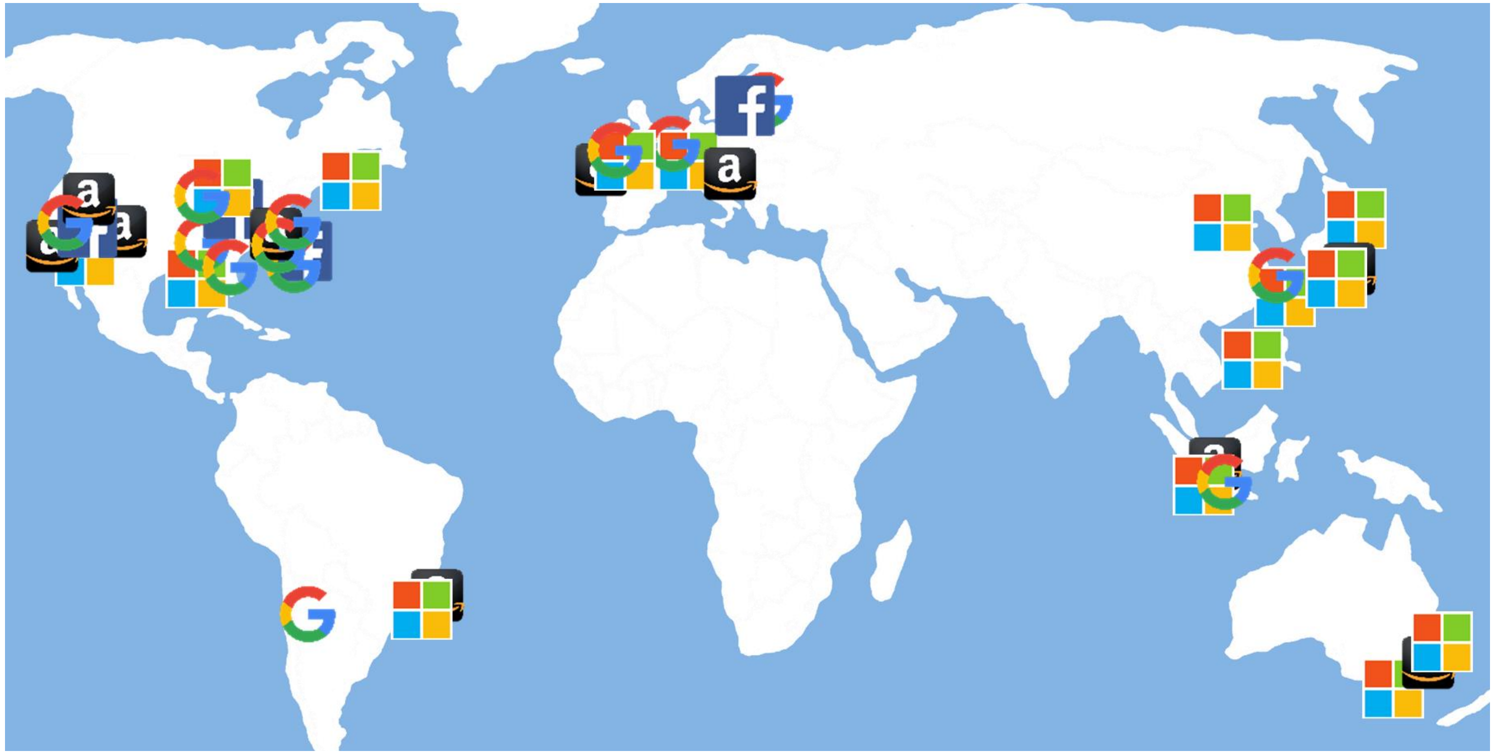
# Minimizing Electricity Cost for Geo-Distributed Interactive Services with Tail Latency Constraint

Mohammad A. Islam, Anshul Gandhi, and Shaolei Ren

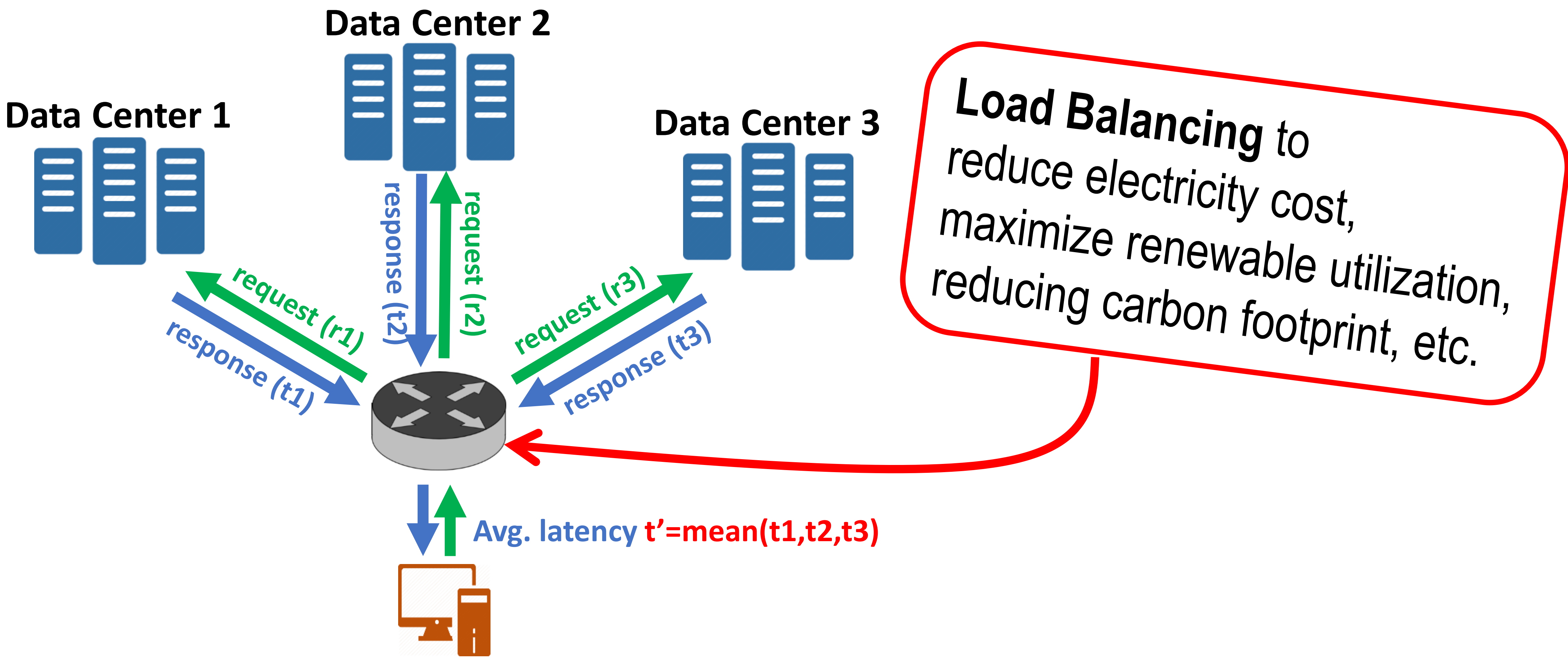
This work was supported in part by the U.S. National Science Foundation under grants CNS-1622832, CNS-1464151, CNS-1551661, CNS-1565474, and ECCS-1610471

# Data centers

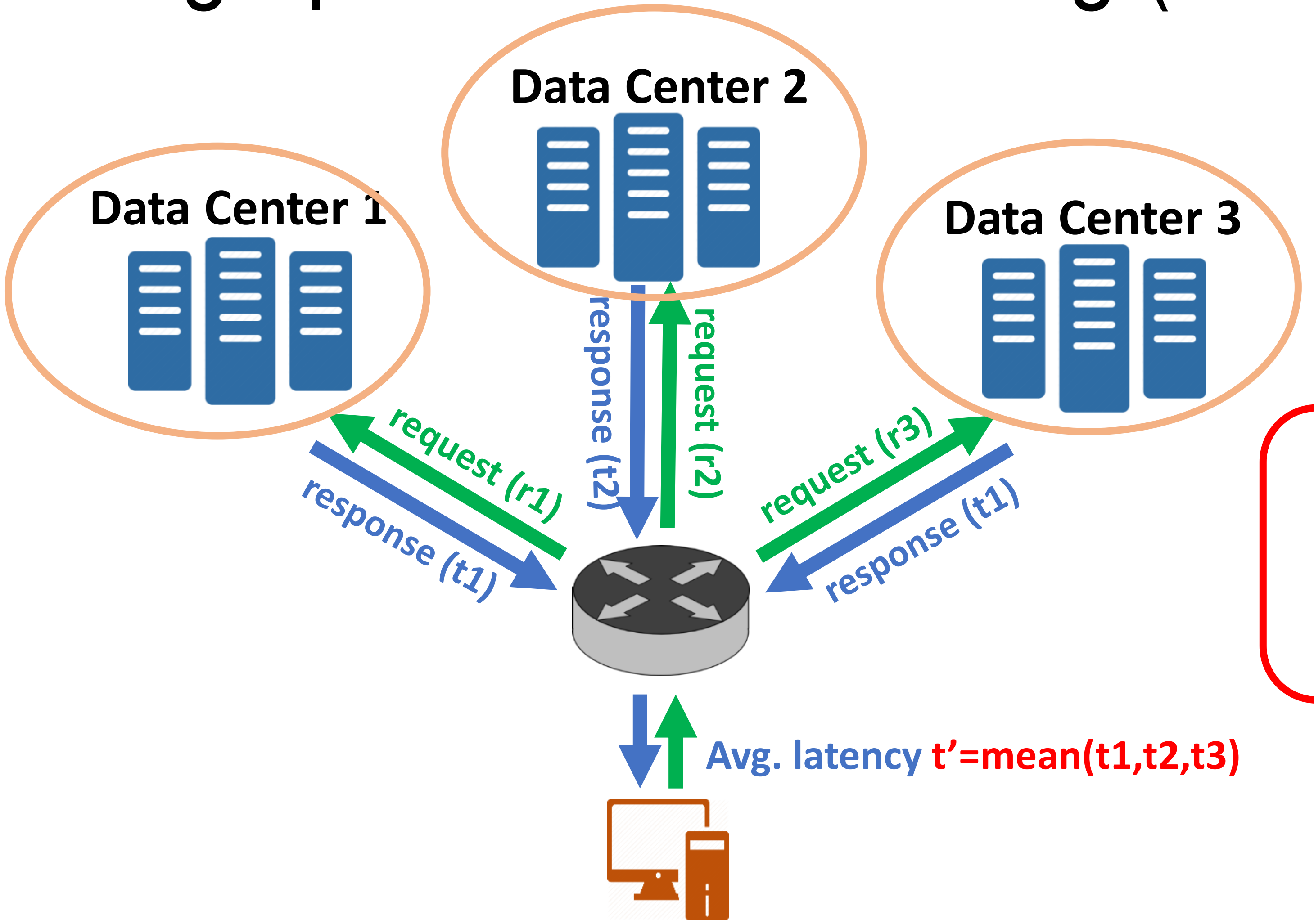
- Large IT companies have data centers all over the world
- Can exploit spatial diversity using *Geographical Load Balancing* (***GLB***)



# Geographical load balancing (GLB)

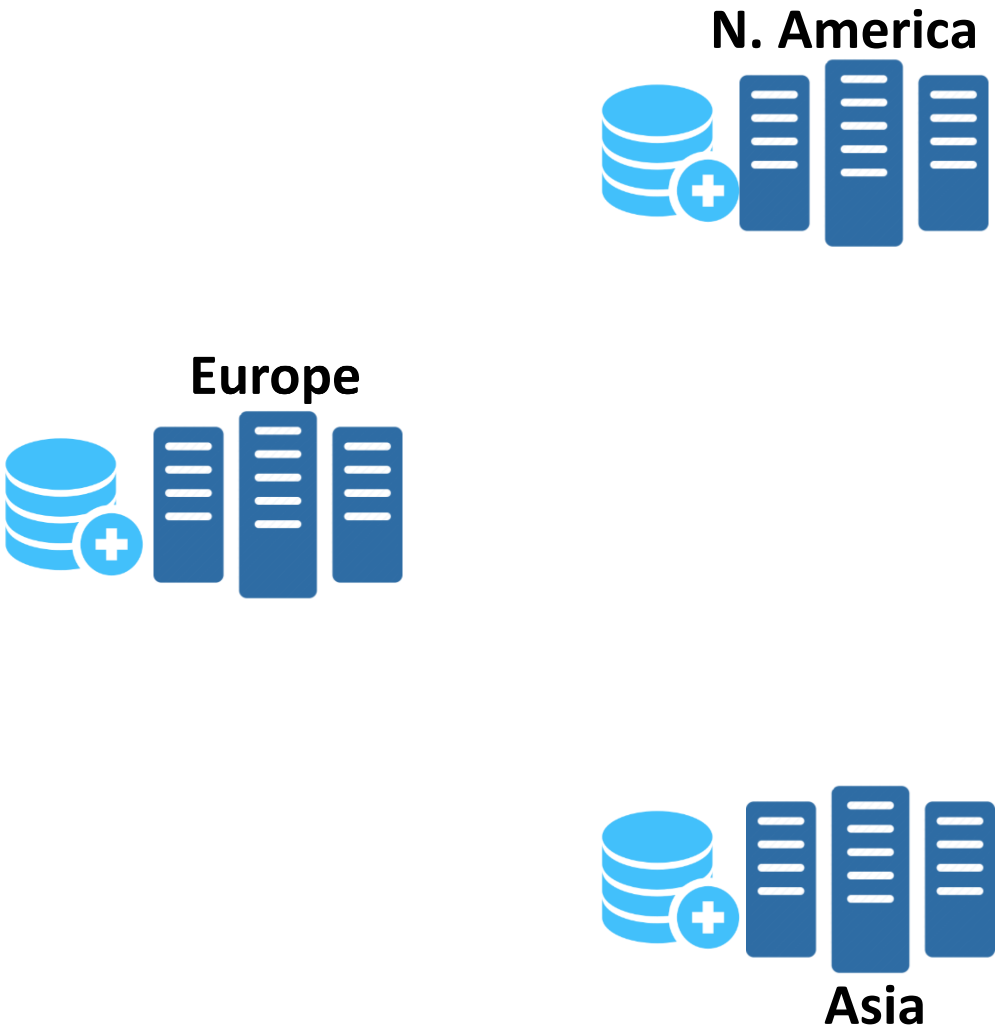


# Geographical load balancing (GLB)



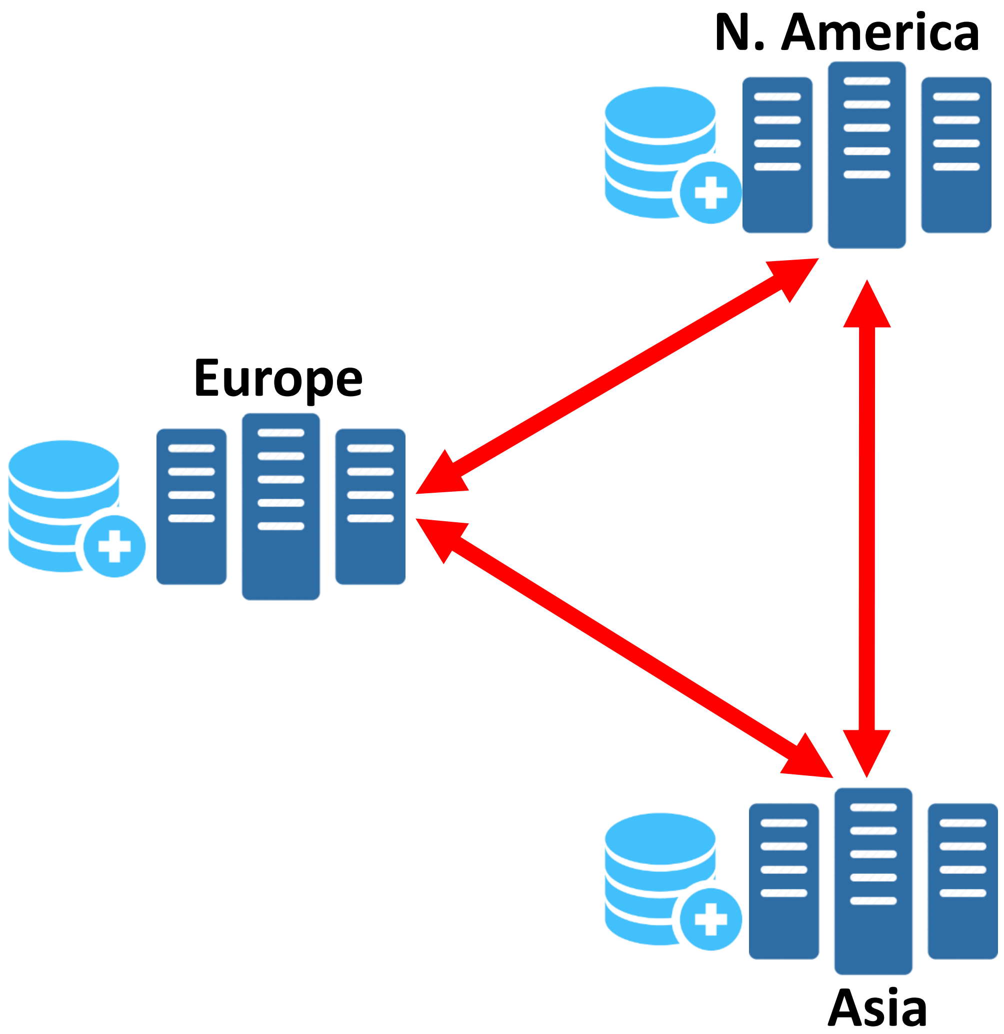

Assuming data required is **centrally** managed, and **replicated** over all the sites

# GLB is facing new challenges



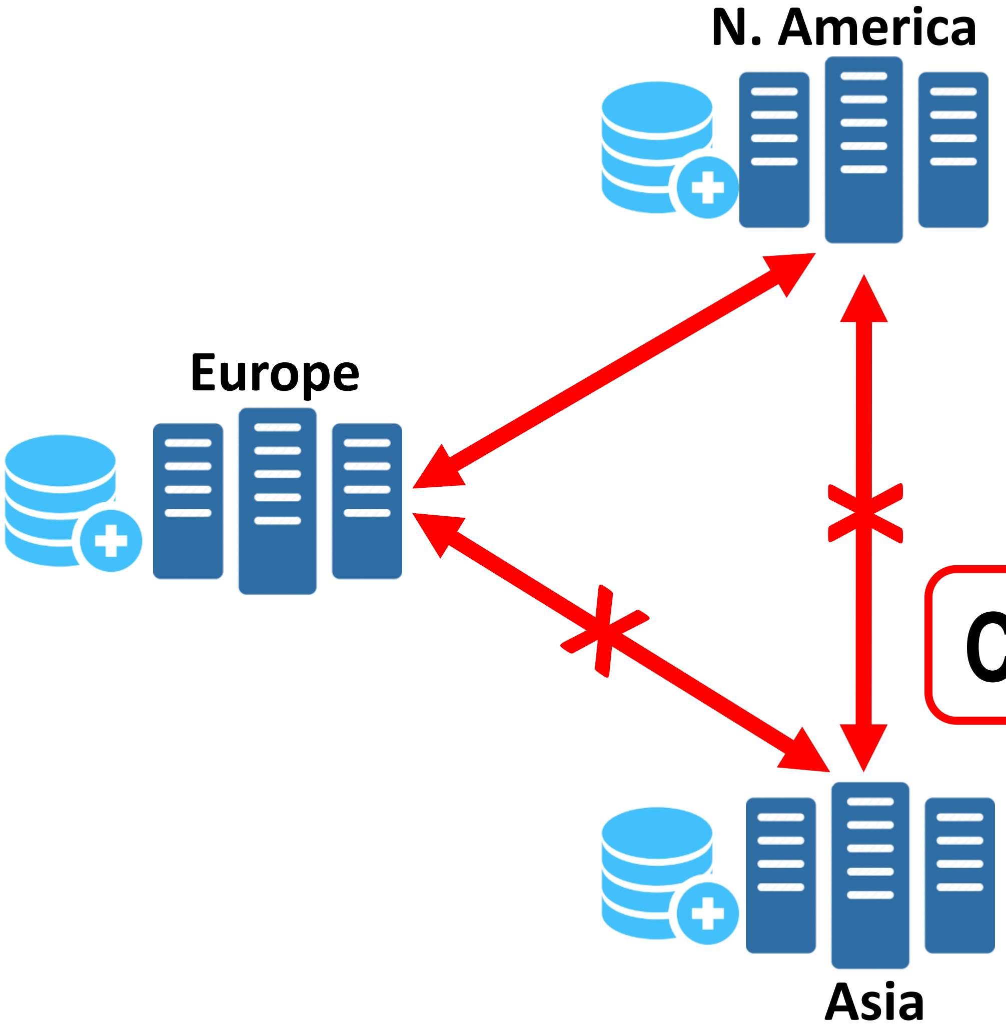

- Tons of locally generated data
  - Smart home, IoT, edge computing

# GLB is facing new challenges




- Tons of locally generated data
  - Smart home, IoT, edge computing
- Limited BW for large data transfer

# GLB is facing new challenges

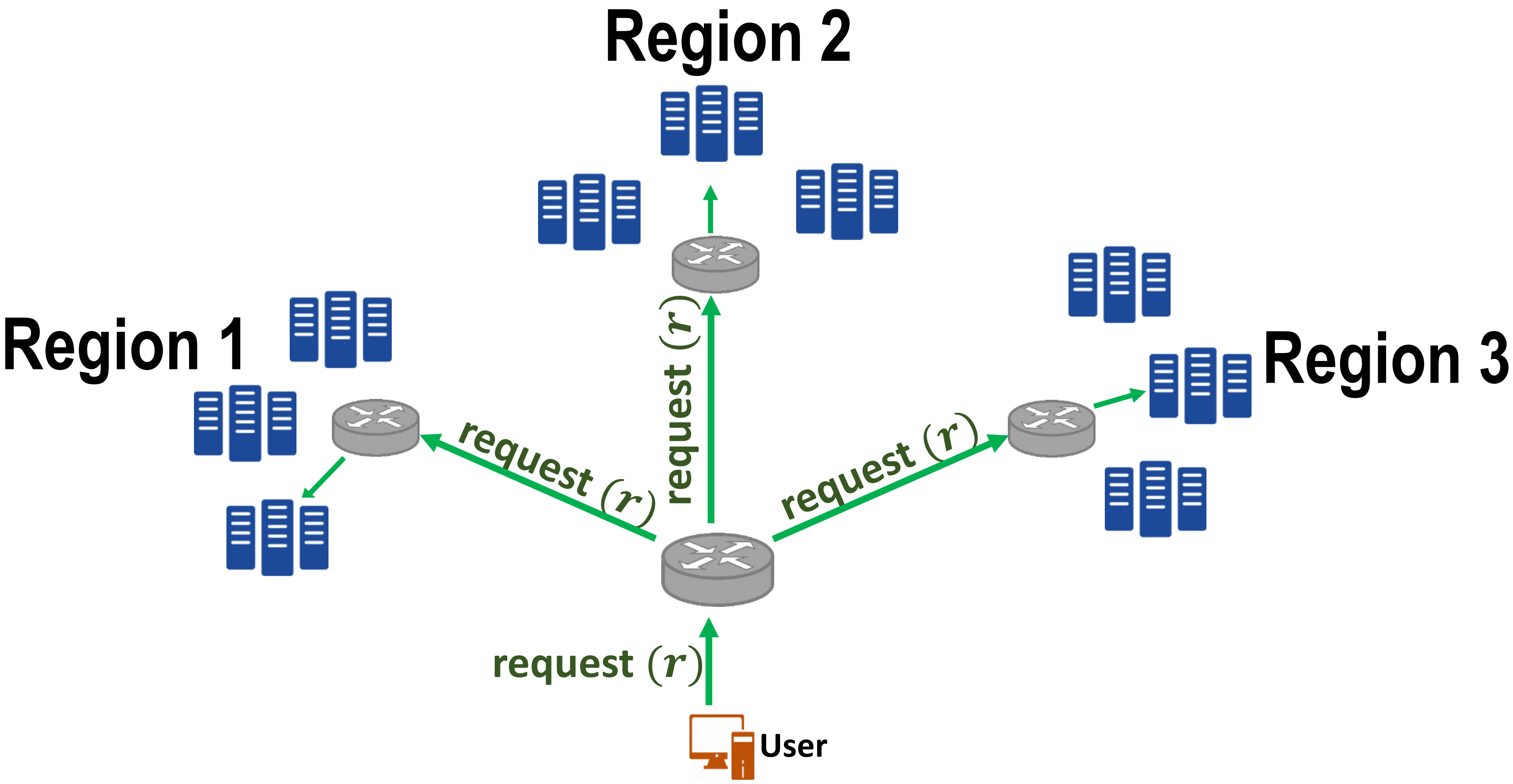



- Tons of locally generated data
    - Smart home, IoT, edge computing
- Limited BW for large data transfer
- Government restriction due to data sovereignty and privacy concerns

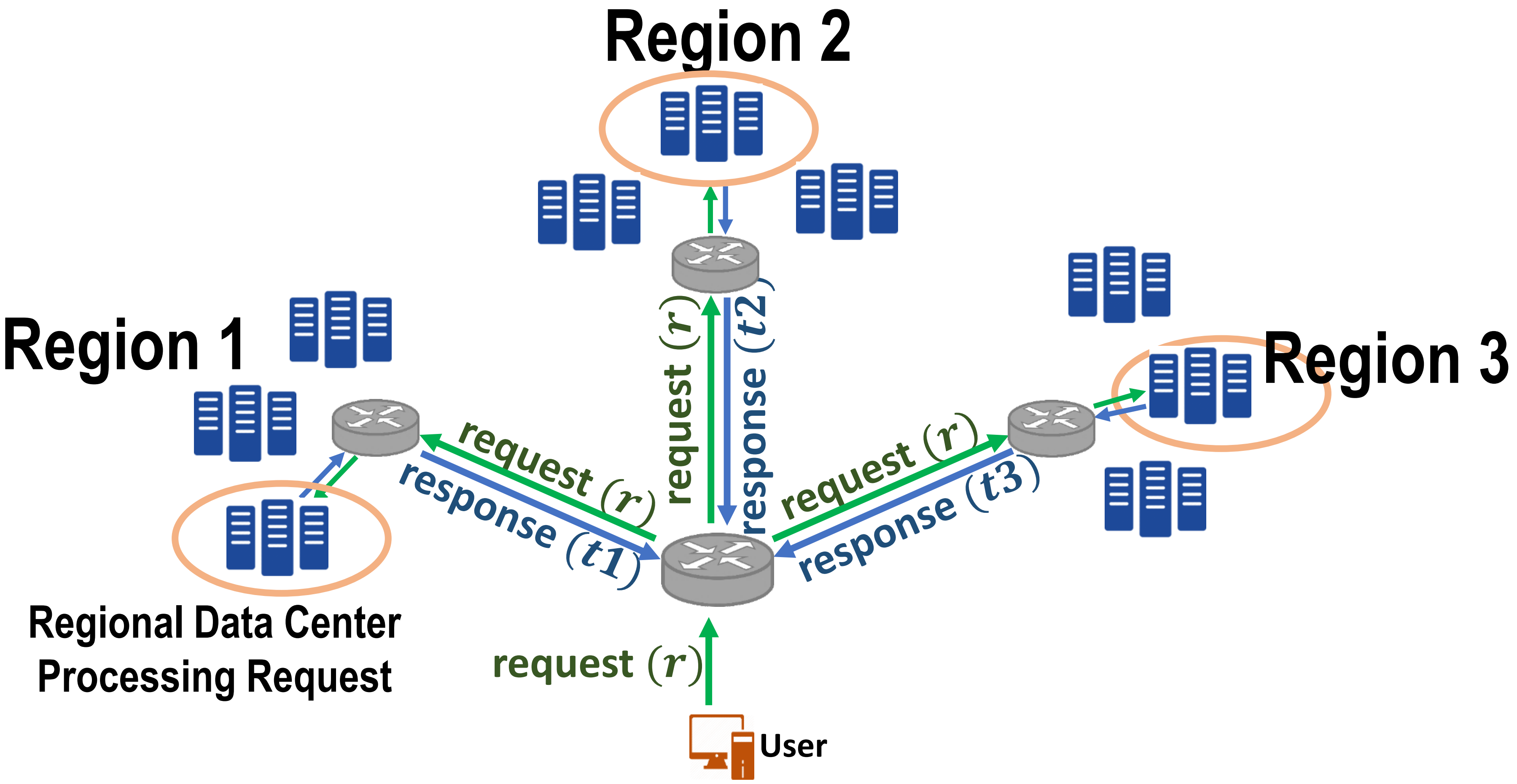
**Centralized processing is not practical**

**Geo-distributed processing is emerging**

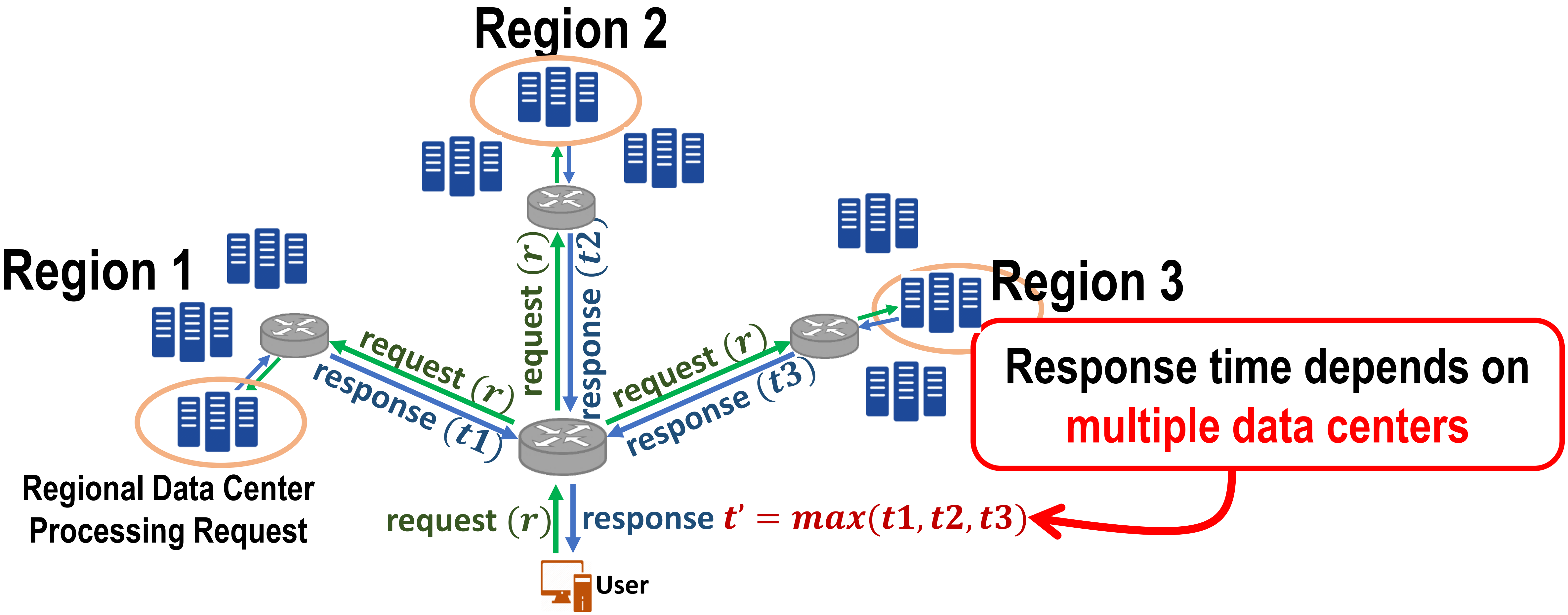
# Geo-distributed processing



# Geo-distributed processing



# Geo-distributed processing



# Tail latency based SLO

- Service providers prefer tail latency (i.e., response time) based SLO
- Two parameters
  - Percentile value (e.g., 95% or p95)
  - Latency threshold
- Example
  - SLO of ***p95 and 100ms***, means ***95% of the response times should be less than 100ms***
- Existing research on GLB mostly focuses on average latency
  - Zhenhua Liu [Sigmetrics'11], Darshan S. Palasamudram [SoCC'12], Kien Li [IGCC'10, SC'11], Yanwei Zhang [Middleware'11]…

# Challenges of geo-distributed processing

- How to characterize the tail latency?
  - Response time depends on multiple paths for each request
  - Includes large network latency
  - Simple queueing models like M/M/1 for average latency cannot be used
- How to optimize load distribution among data centers?

**McTail: a novel GLB algorithm with data driven profiling of tail latency**

# Problem formulation

Total electricity cost

- General formulation with $N$ data centers and $S$ traffic sources

$$\text{minimize}_{\vec{a}} \sum_{j=1}^{N} q_j \cdot e_j(a_j)$$

Tail latency constraint

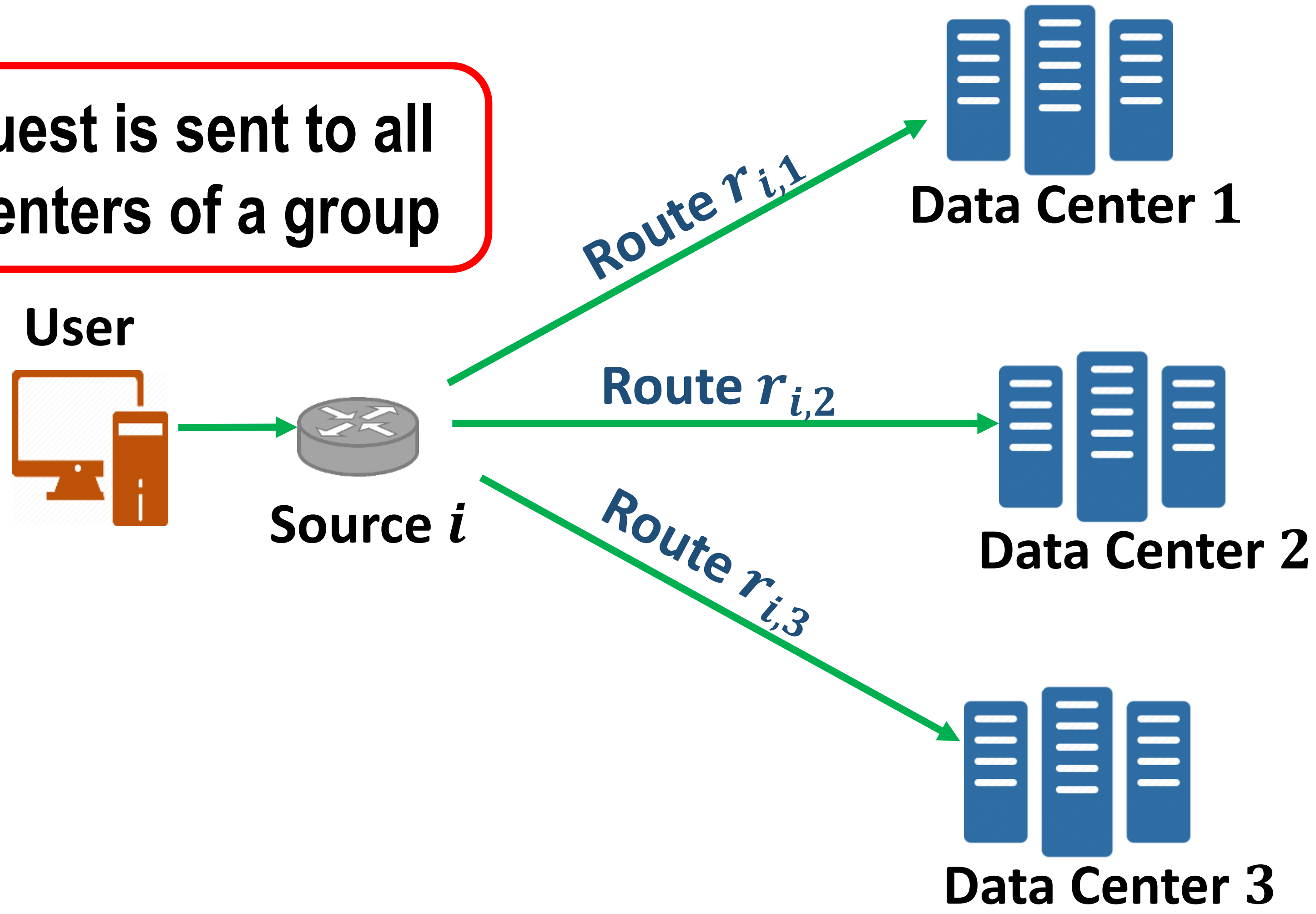
$$\text{subject to, } p_i(\vec{a}, \vec{r}) \geq P_i^{SLO}, \forall i = 1,2,\cdots,S$$

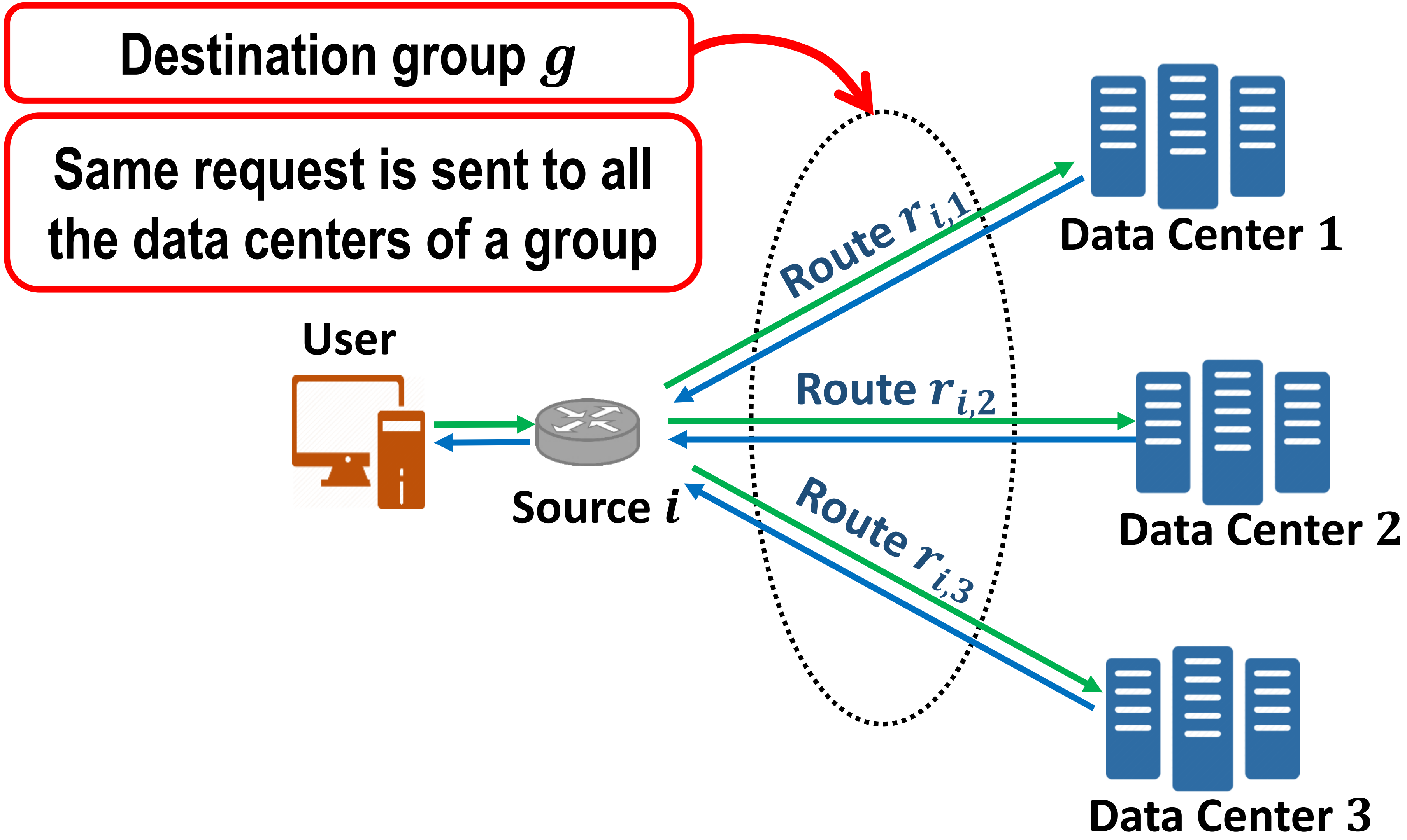
- $\vec{a} = \{a_1, a_2, \cdots a_N\}$ is workload (request processed) at different data centers
- $\vec{r_i}$ is the network paths from source $i$ to all the data centers
- $p_i$ is $\Pr(d_i \leq D_i)$, where $d_i$ is end-to-end response time at traffic source $i$, and $D_i$ is delay target (e.g., 100ms) for tail latency

**How to determine $p_i(\vec{a}, \vec{r}_i)$?**


$p_{i,j}^{route}(a_j, r_{i,j})$ is the probability that response time of $r_{i,j}$ is less than $D_i$

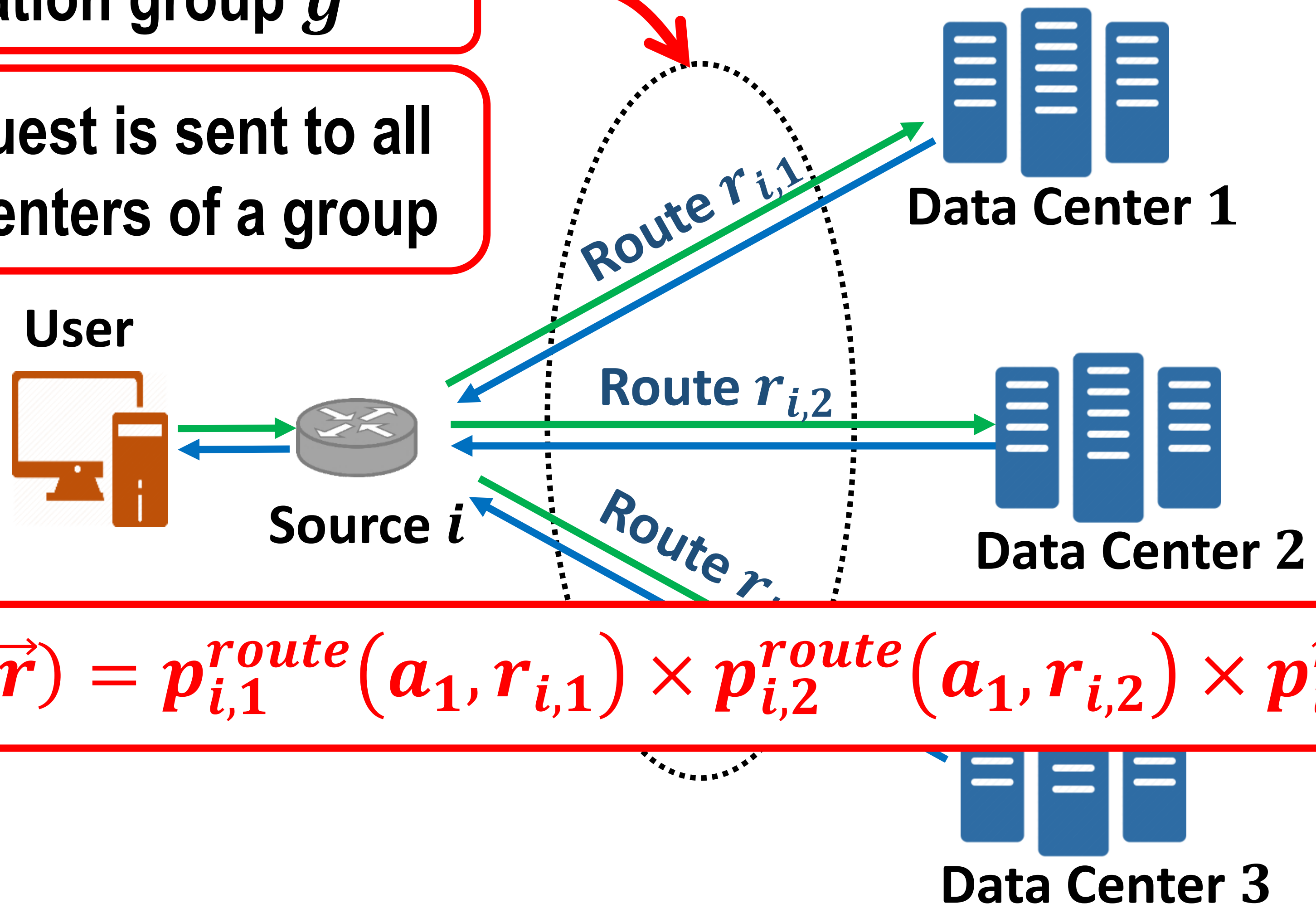
Same request is sent to all the data centers of a group




Destination group $g$
Same request is sent to all the data centers of a group
User
Source $i$
Route $r_{i,1}$
Route $r_{i,2}$
Route $r_{i,3}$
Data Center 1
Data Center 2
Data Center 3

Destination group $g$

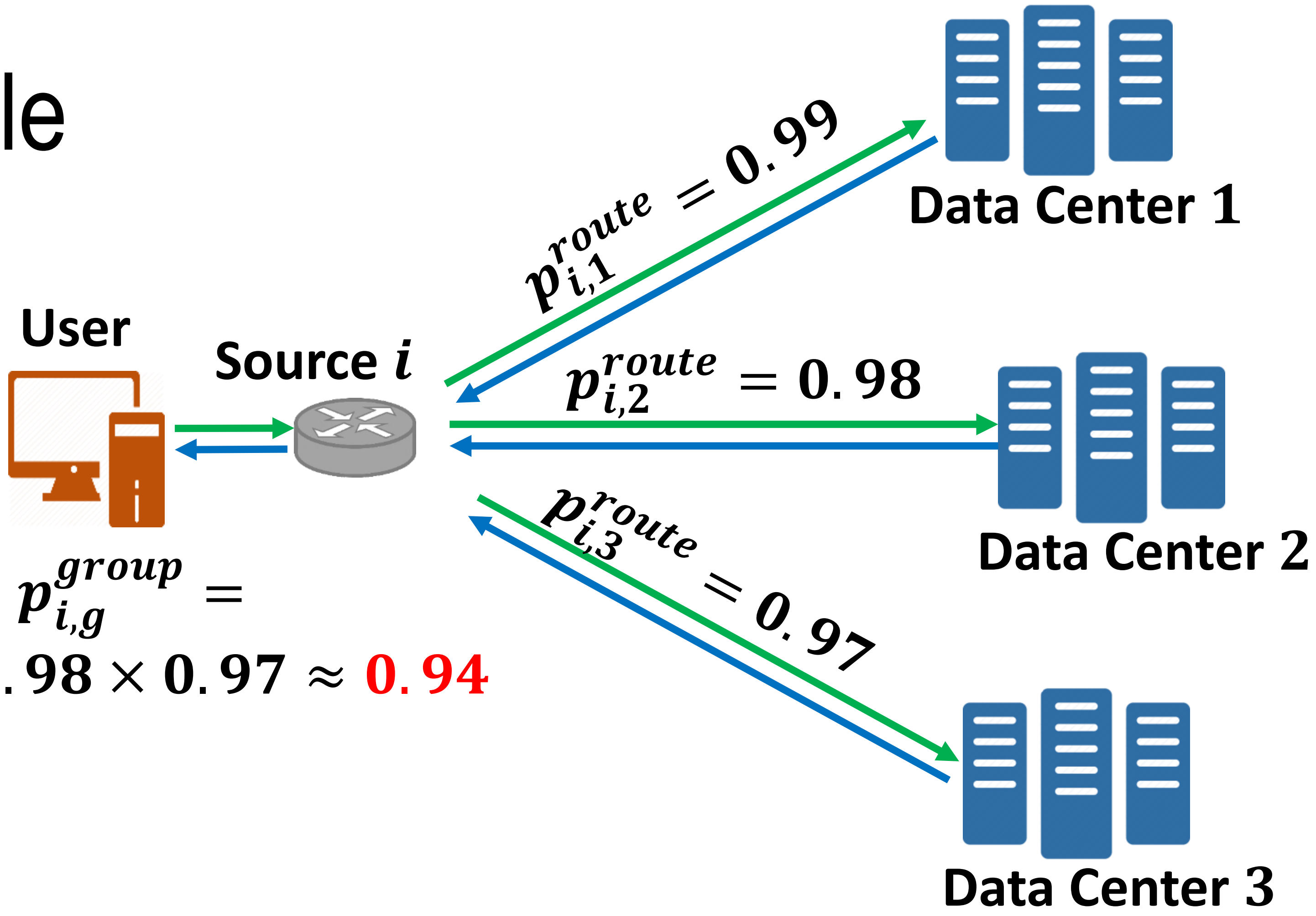
Same request is sent to all the data centers of a group




$$p_{i,g}^{group}(\vec{a},\vec{r}) = p_{i,1}^{route}(a_1,r_{i,1}) \times p_{i,2}^{route}(a_1,r_{i,2}) \times p_{i,1}^{route}(a_3,r_{i,3})$$

Because of differences in data sets, random performance interference etc., response time over different routes can be considered un-correlated

# Example




$p_{i,g}^{group} =$
$0.99 \times 0.98 \times 0.97 \approx 0.94$

For requests sent to this group of data centers,
94% of the response times are less than $D_i$

# Response time probability for a source

- $G = N_1 \times N_2 \times \cdots \times N_M$ possible destination groups
  - Where $N_m$ is the number of data center in region $m$
- Response time probability at source $i$ is

$$p_i(\lambda) = p_i(\vec{a}, \vec{r}) = \frac{1}{\Lambda_i} \sum_{g=1}^{G} \lambda_{i,g} \cdot p_{i,g}^{group}(\vec{a}, \vec{r})$$

  - $\lambda_{i,g}$ is the workload sent to destination group $g$
  - $\Lambda_i = \sum_{g=1}^{G} \lambda_{i,g}$ is the total workload from source $i$

**Weighted average over all the groups**

# Updated problem formulation

$$\text{minimize}_{\vec{a}} \sum_{j=1}^{N} q_j \cdot e_j(a_j)$$

**Objective same as before, minimizing electricity cost**

$$\text{subject to} \; \frac{1}{\Lambda_i} \sum_{g=1}^{G} \lambda_{i,g} \cdot p_{i,g}^{group}(\vec{a}, \vec{r}) \geq P_i^{SLA},$$

**Tail latency decomposed into route-wise latencies**

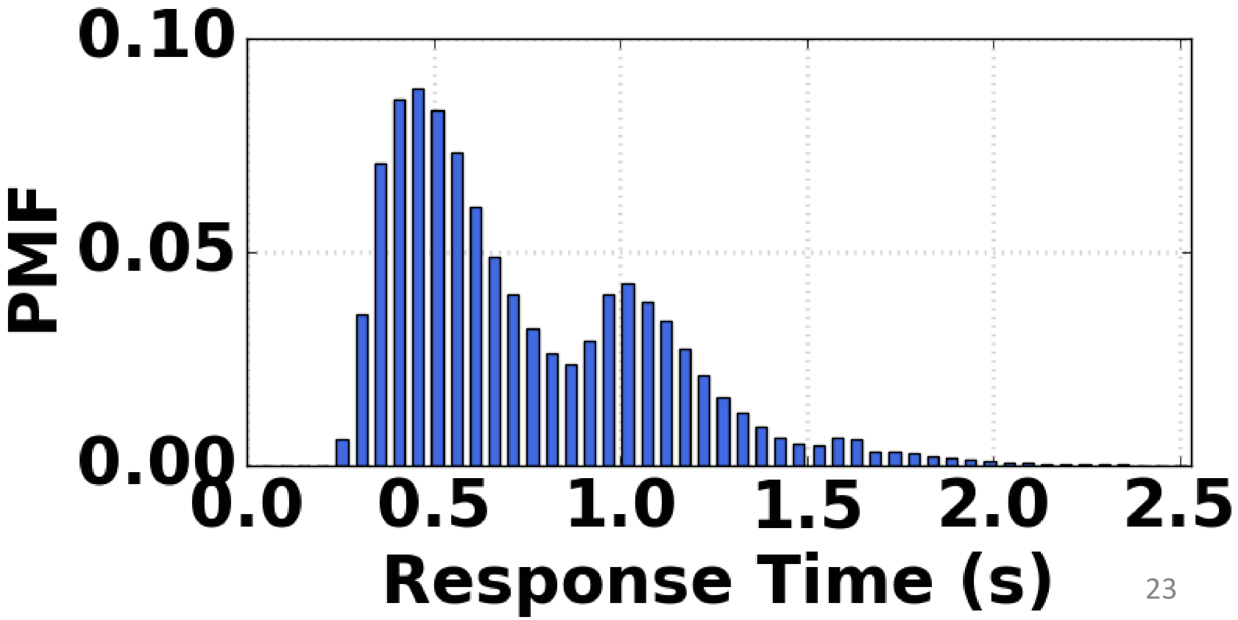
$$\sum_{g=1}^{G} \lambda_{i,g} = \Lambda_i, \forall i = 1,2,\cdots,S$$

**Workload constraint**

**Need to determine $p_{i,j}^{route}(a_j, r_{i,j})$ for all routes**

# Profiling response time probability of a route

- We need tail latency
  - Hard to model for arbitrary workload distributions
- **Data driven approach - profile the response time statistics (find the probability distribution) from observed data**
- Example
  - Response profile for 100K request



# Challenges of data driven approach

- Response time profile of a route depends on amount of data center workload
  - We set $W$ discrete levels of workload for each data center
- $S \times N$ network paths between $S$ sources and $N$ data centers
- **Total $S \times W \times N$ number of profiles**
- Need to update if network latency distribution, data center configuration, or workload composition changes

**Slow and repeated profiling**

# Profiling response statistics for one route

- $F_{i,j}^{N}$ is network latency distribution
- $F_j^D(x)$ is data center latency distribution with load $x$
- End-to-end latency distribution of route $r_{i,j}$ is

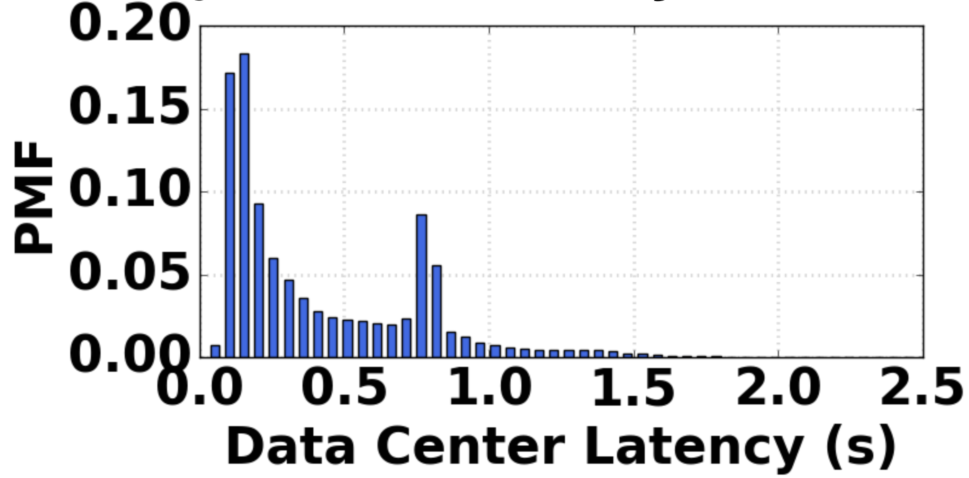
$$\boldsymbol{F_{i,j}^{\mathbf{R}} = F_{i,j}^{N} * F_j^D(x)}$$

  - where " $*$ " is the convolution operator

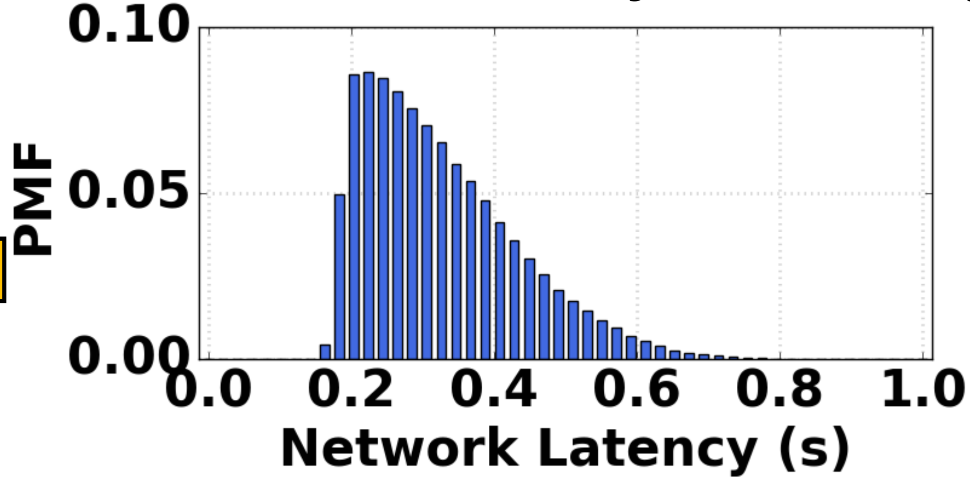
**Key idea: profile $\boldsymbol{F_{i,j}^{N}}$ and $\boldsymbol{F_j^D(x)}$ seperately**

# Example

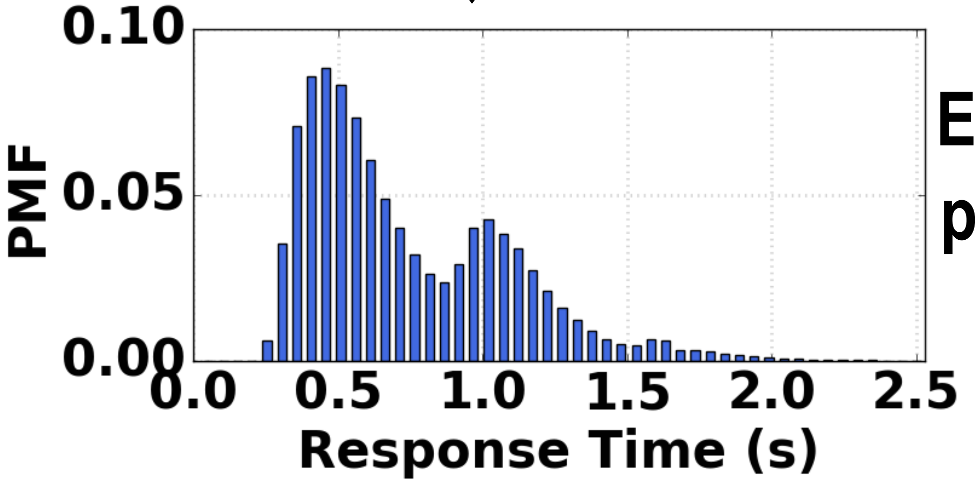
Latency of data center $j$ with load $x$




Network latency of route $r_{i,j}$




Convolution

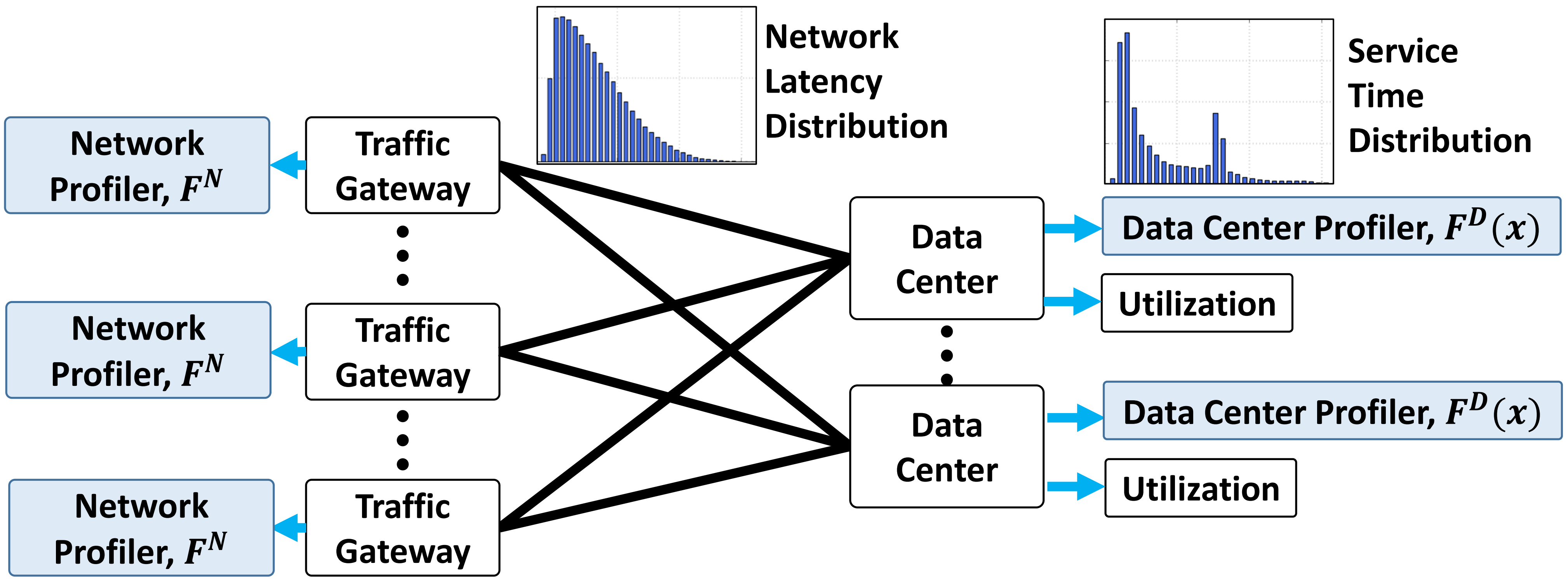



End-to-end response profile of a route, $F^{\mathrm{R}}_{i,j}$

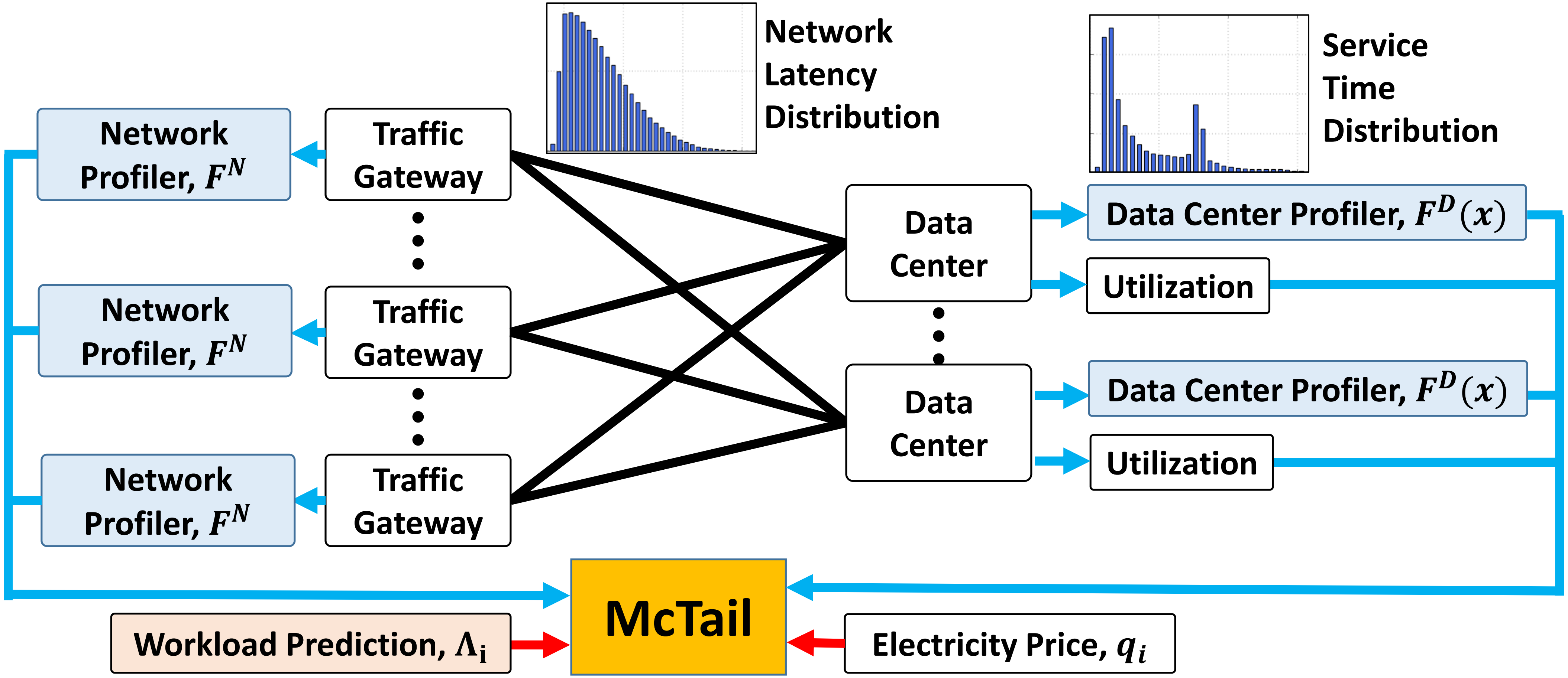
# Profiling response time statistics in McTail

- $S \times N$ network routes profiles
- $N \times W$ data centers profiles
- Total $\boldsymbol{(S + W)} \times \boldsymbol{N}$ profiles versus $\boldsymbol{S \times W} \times \boldsymbol{N}$ profiles before
- Profiling overhead
  - Only data center profiles need updating when workload composition and/or data center configuration is changed
    - Infrequent event
  - Network latency distribution may change more frequently
    - Already monitored by service providers
  - **Data overhead comparable to existing GLB studies**

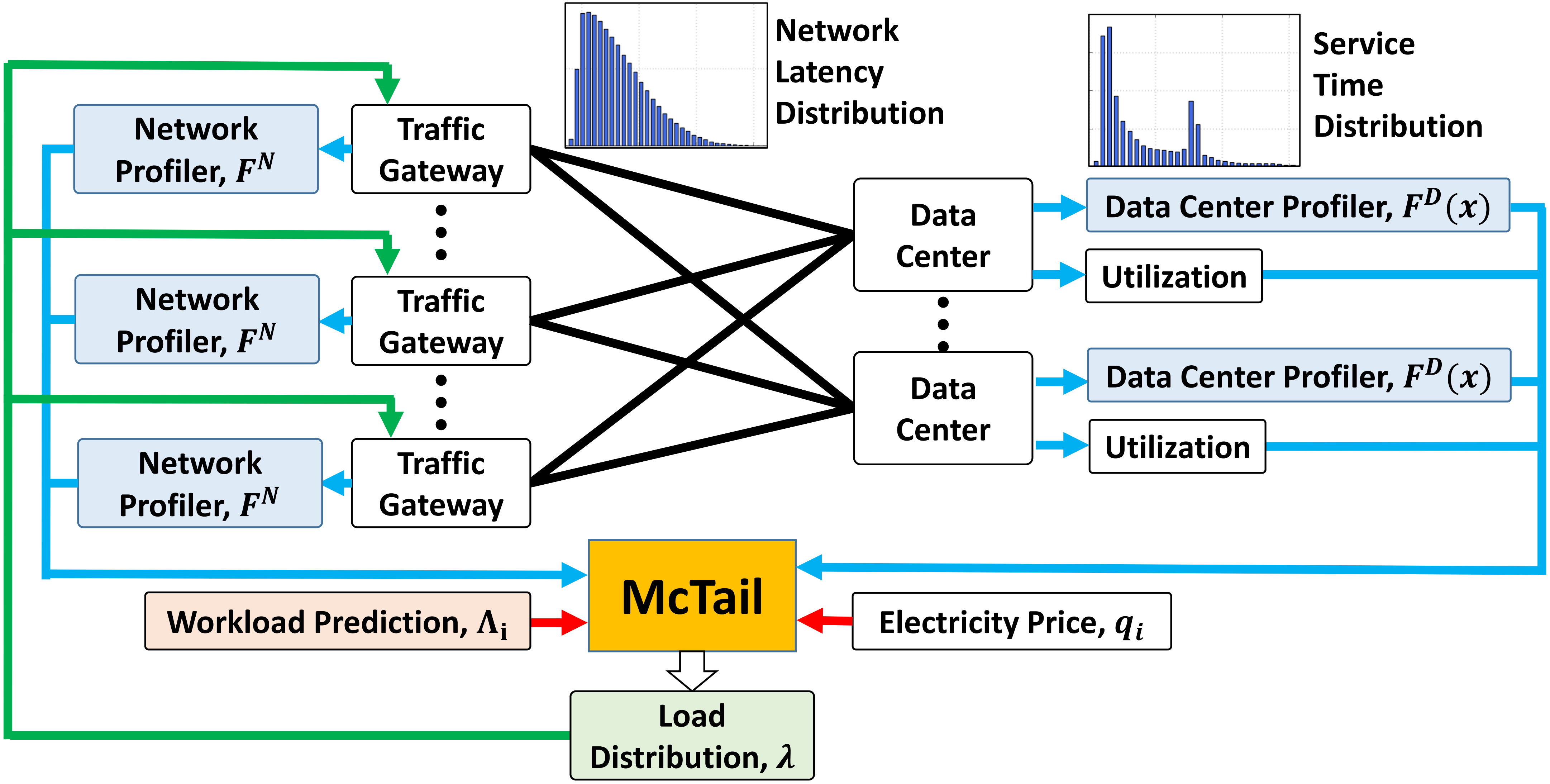
# McTail system diagram



# McTail system diagram



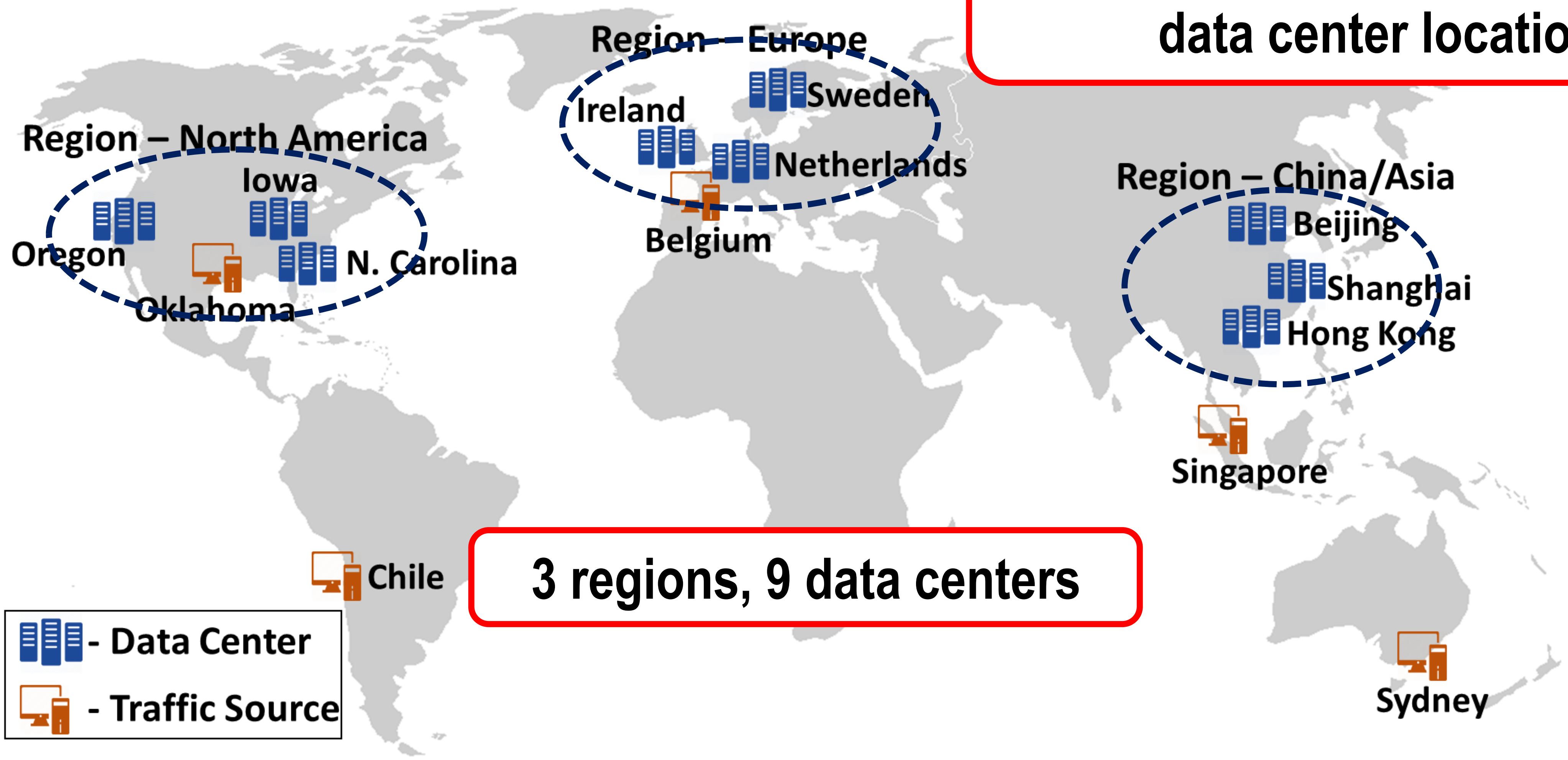
# McTail system diagram



# Evaluation

# Evaluation setup

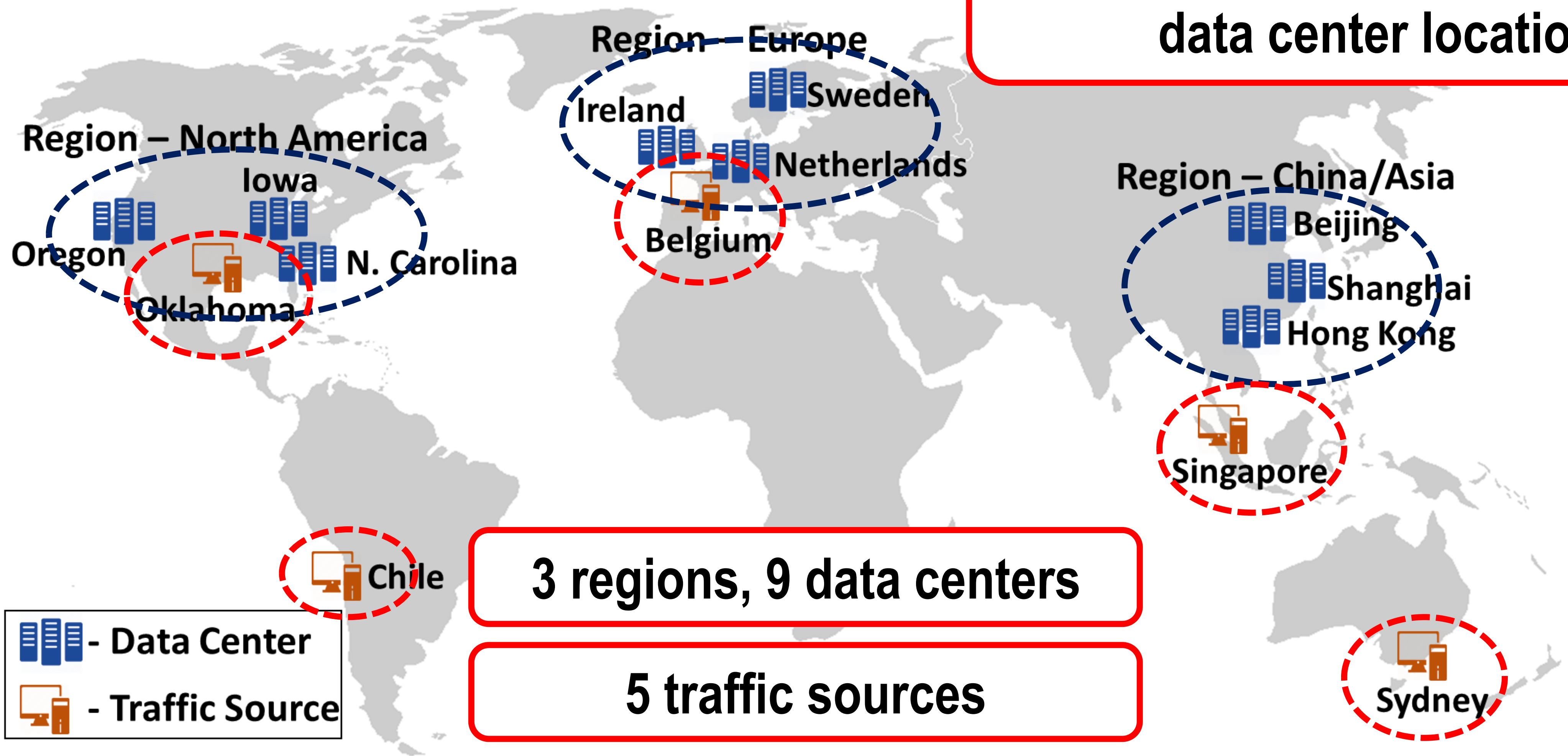
**Based on Google and Facebook data center locations**




**3 regions, 9 data centers**

# Evaluation setup

**Based on Google and Facebook data center locations**



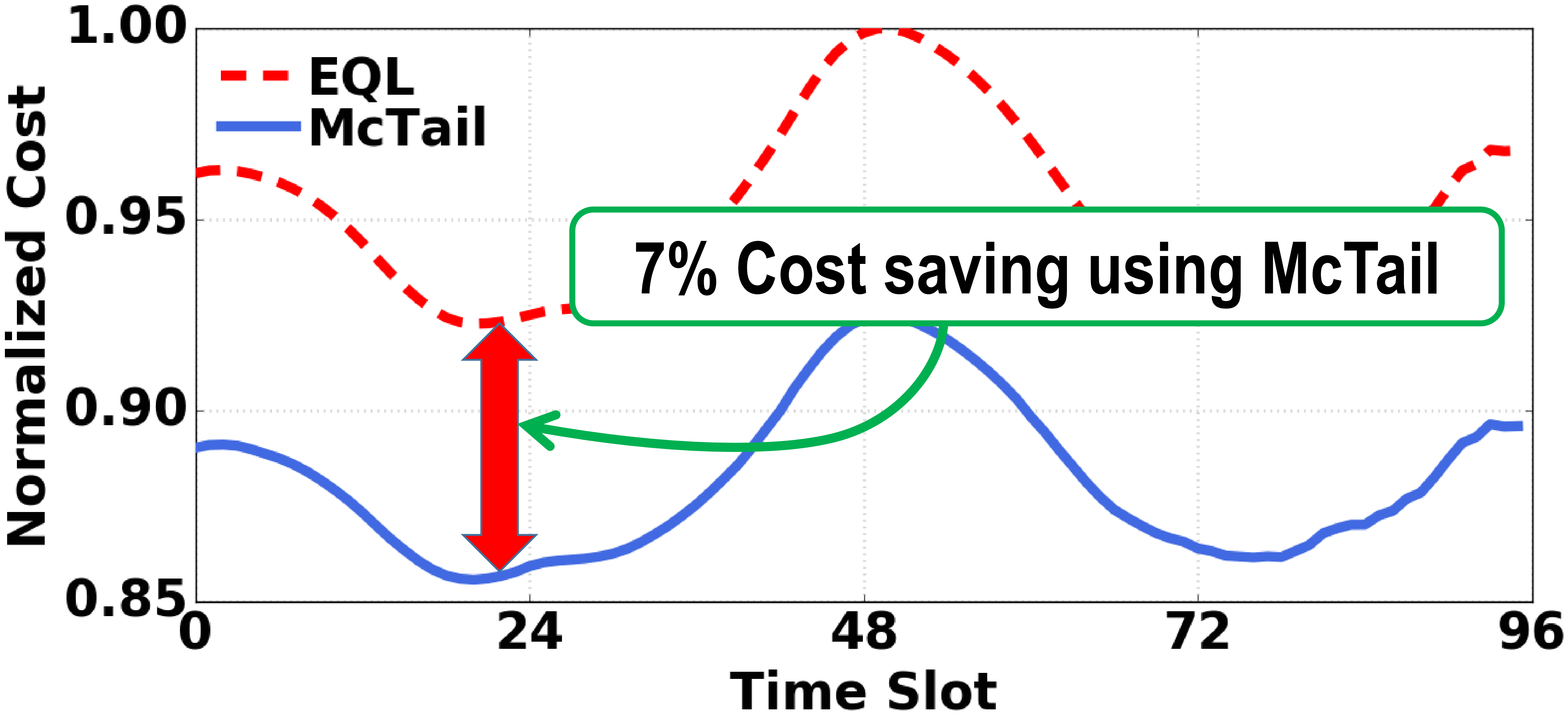

**3 regions, 9 data centers**

**5 traffic sources**

# Evaluation setup

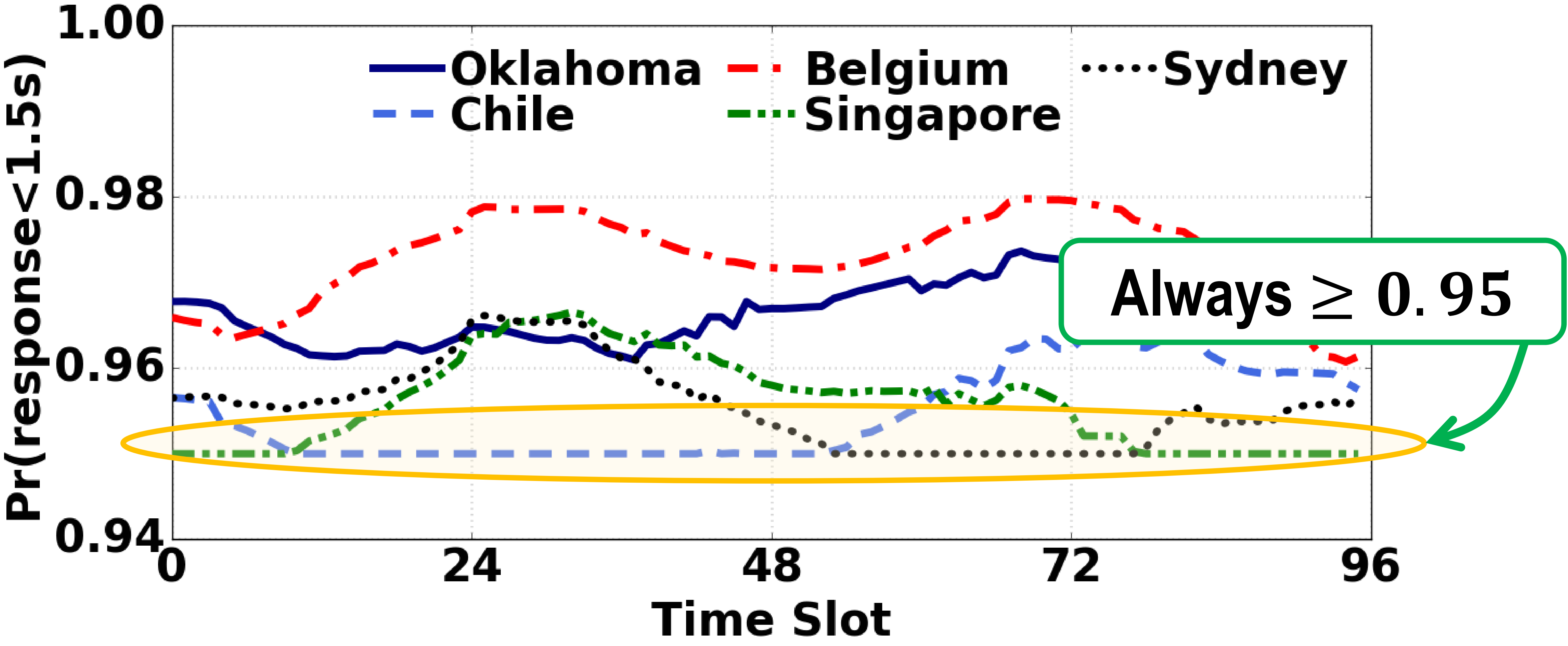
- Discrete event simulation using SimEvents from Mathworks
- Half-normal network latency distribution based on route length
- Real world traces from Google and Microsoft
- Location wise electricity prices
- SLO set to p95 response time of 1.5 seconds
- 24 hour simulation with load distribution updated every 15 minutes
- Homogenous data center setting to ease the simulation

# Cost saving

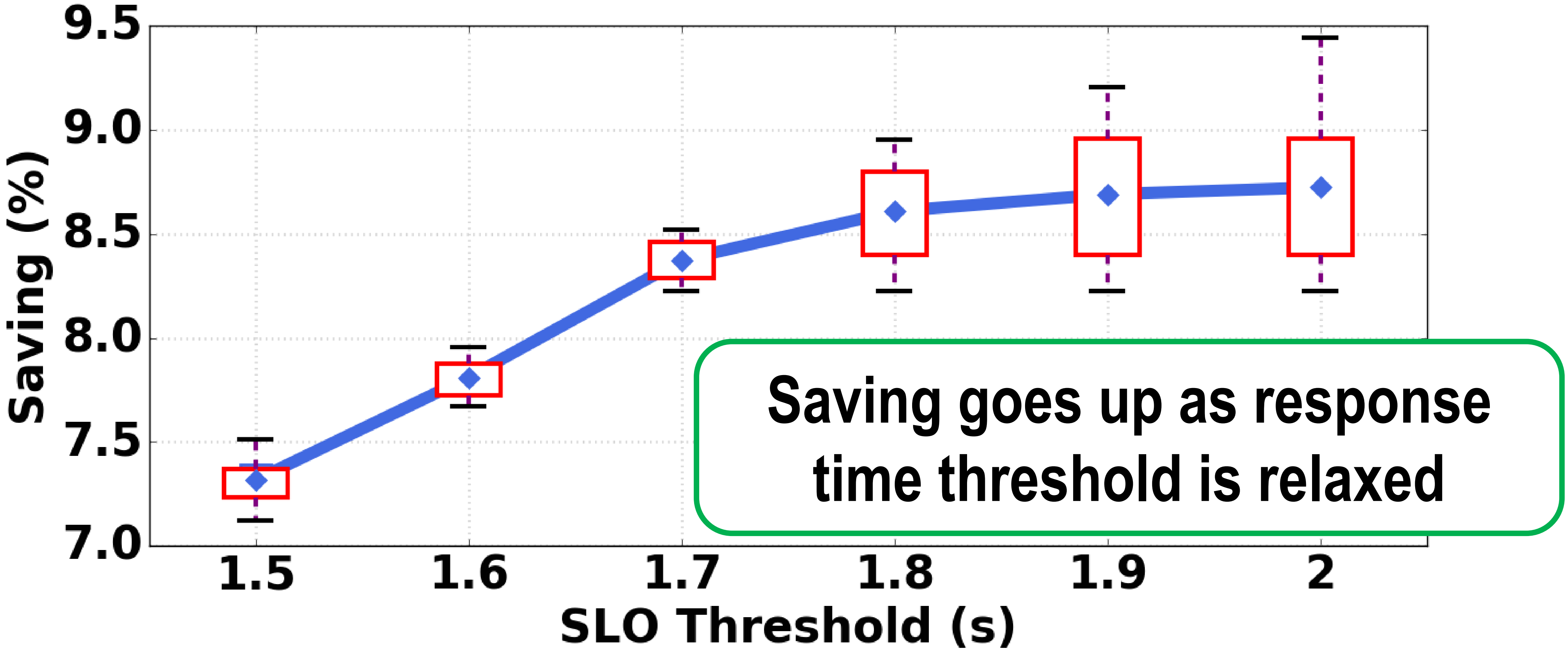


1.00
0.95
0.90
0.85
Normalized Cost
EQL
McTail
7% Cost saving using McTail
0
24
48
72
96
Time Slot

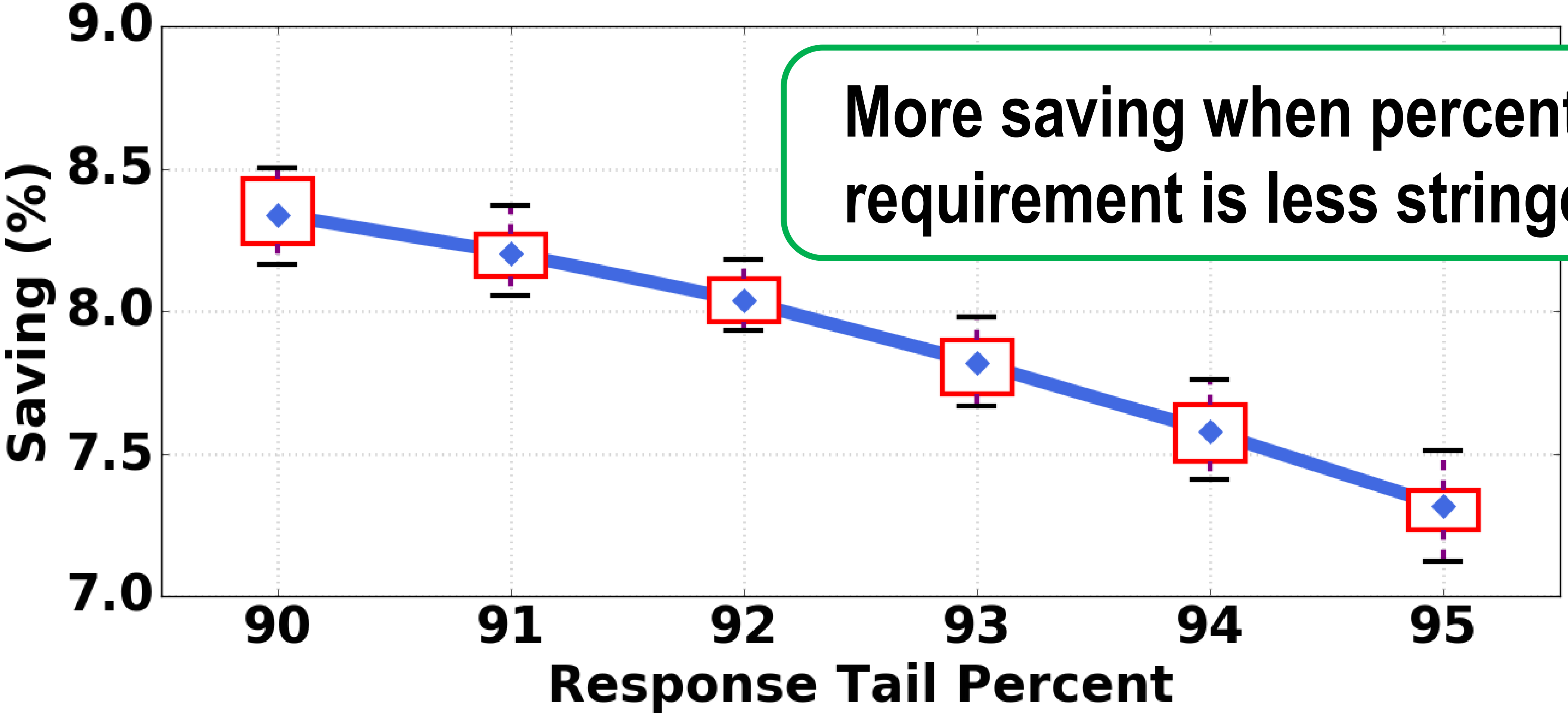
# Performance



# Impact of SLO change



# Impact of SLO change




More saving when percentile requirement is less stringent

# McTail

- A novel GLB algorithm for geo-distributed interactive services
  - Data-driven approach to characterize the tail latency
  - Negligible extra profiling overhead

**Practical and efficient**