Institute for the Wireless Internet of Things at Northeastern University

EDGE Institute

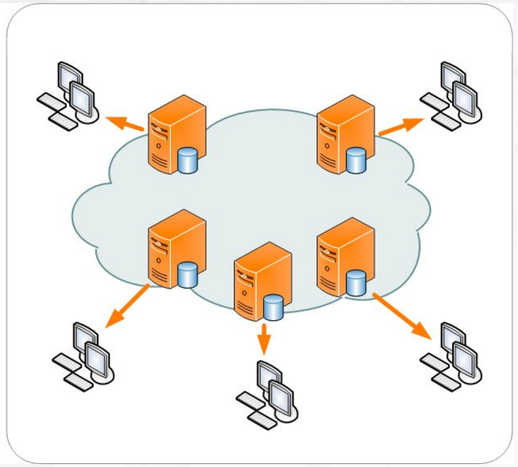
Cache Networks with Optimality Guarantees

Stratis Ioannidis
Department of Electrical and Computer Engineering
Northeastern University

Joint work with
Khashayar Kamran, Yuezhou Liu, Yuanyuan Li, Qian Ma, Milad Mahdian, Armin Moharrer, Tareq Si Salem, Giovanni Neglia, and Edmund Yeh

Motivation

Caching and **object allocation** problems are **ubiquitous**

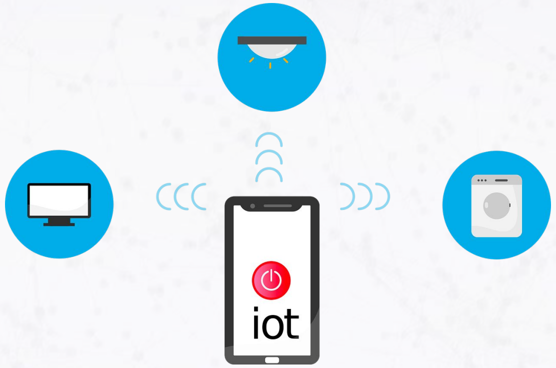


CDNs

[Traverso et al. CCR 2013]
[Leconte et al. ITC 2015]
[Leconte et al. SIGMETRICS 2012]

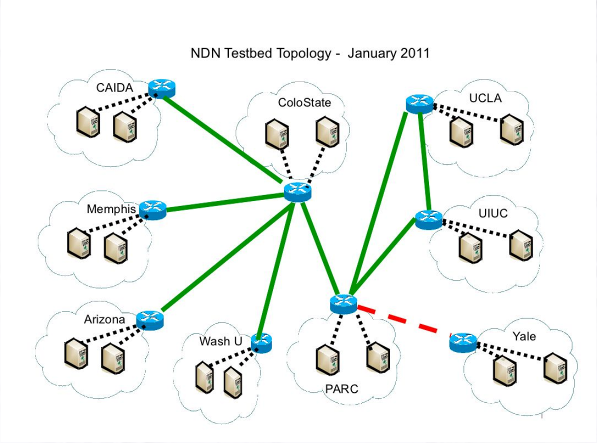
Cloud Computing

[Cara et al. INFOCOM 2019]
[Arteaga et al. FAST 2016]



Edge/Wireless IoT

[Deghan et al. INFOCOM 2015]
[Leconte et al. ITC 2015]
[Leconte et al. SIGMETRICS 2012]

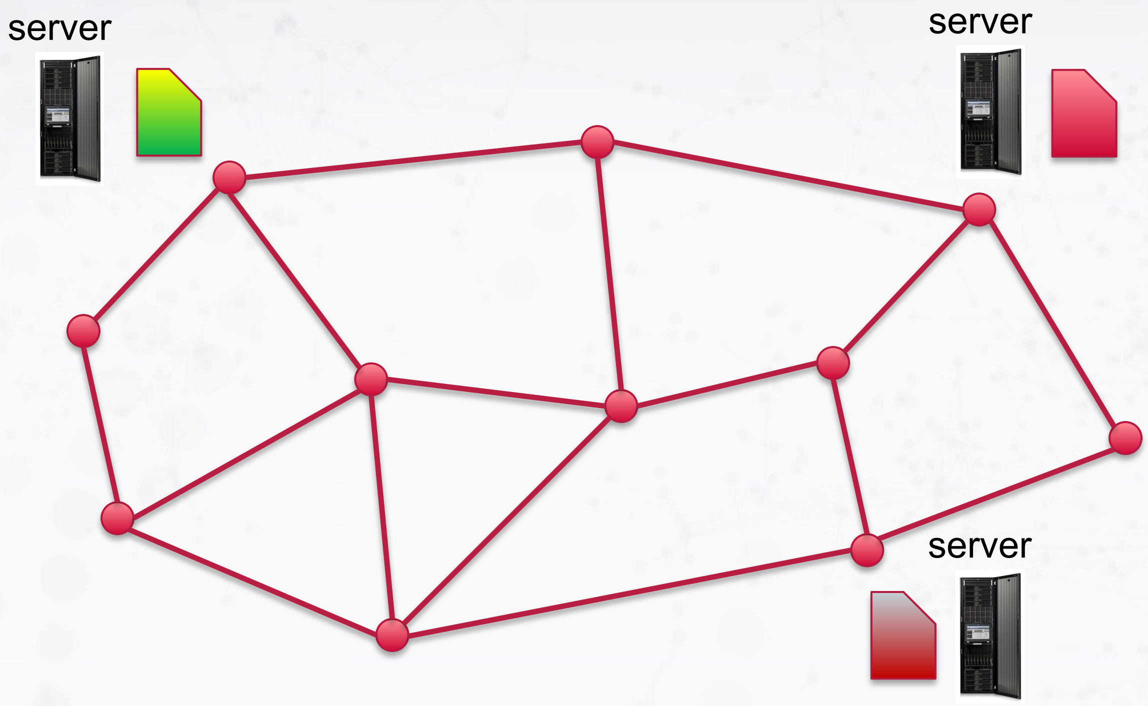


Content-Centric Networking

[Martina et al. INFOCOM 2014]
[Rosenweig et al. INFOCOM 2013]
[Wang et al. ICNP 2013]
[Tyson et al. ICCCN 2012]
[Yeh et al. ICN 2013]
[Jacobson et al. CONEXT 2009]

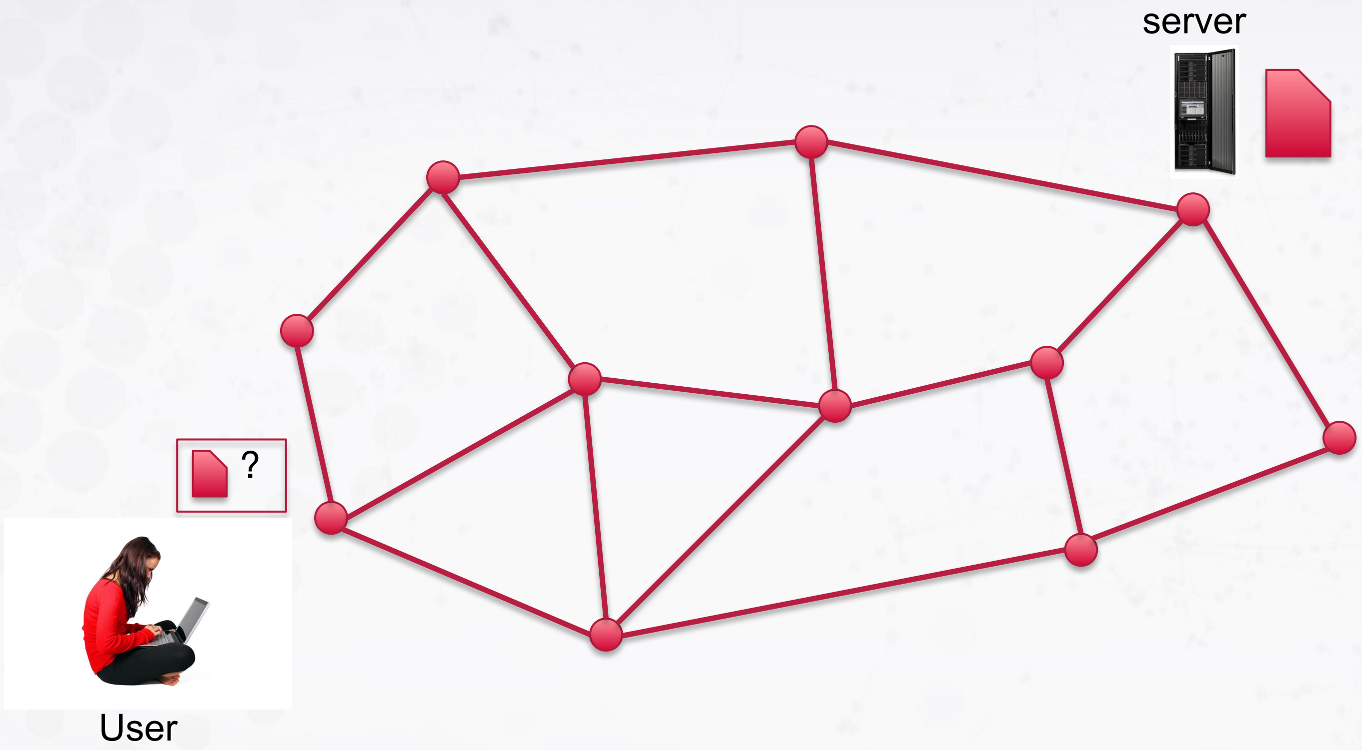
Institute for the Wireless Internet of Things at Northeastern

A Cache Network



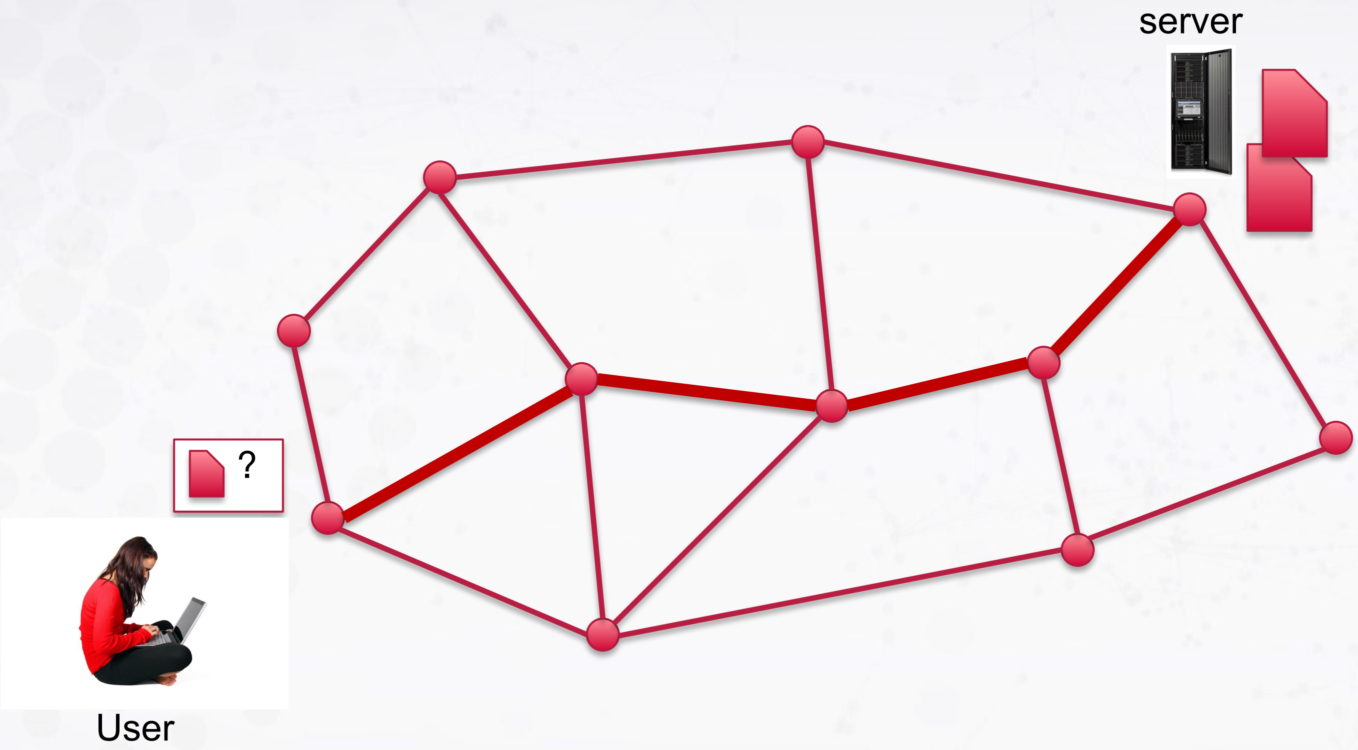
Designated servers in the network store **content items** (e.g., files, file chunks).

A Cache Network



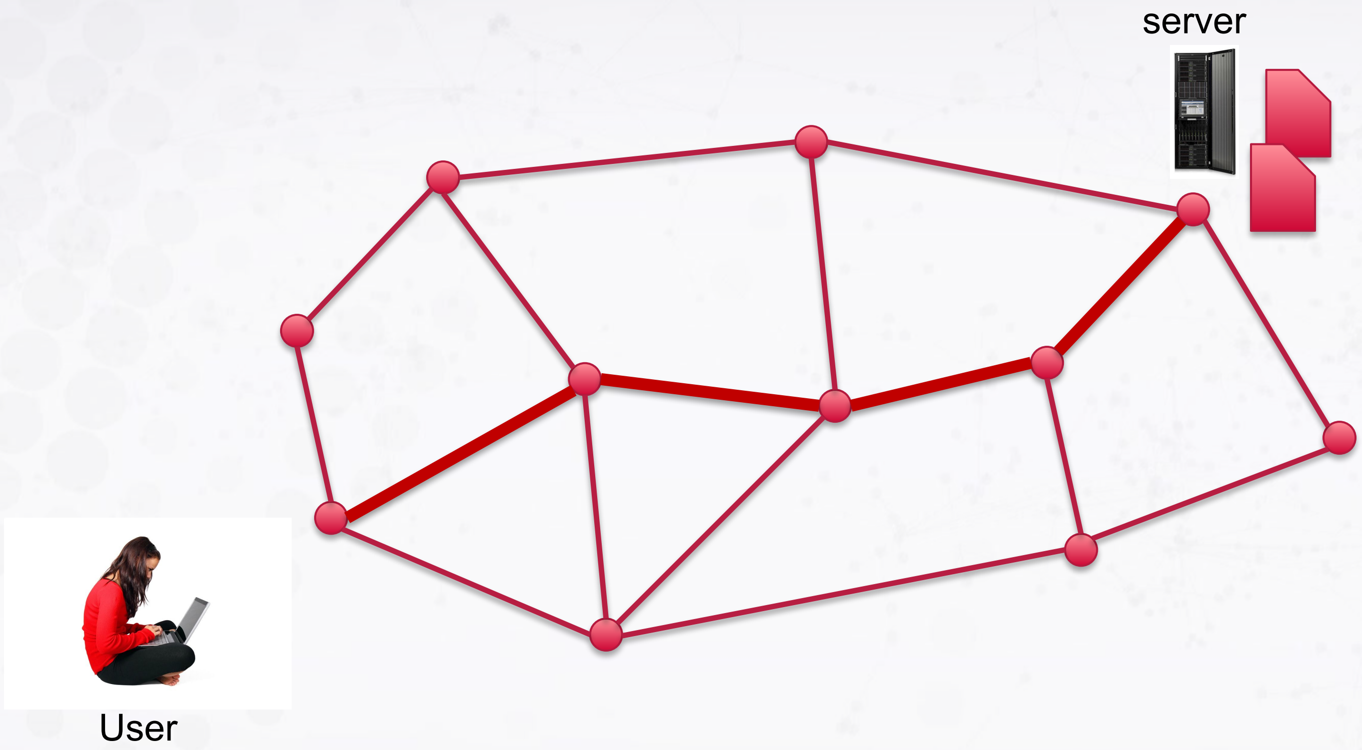
Nodes generate **requests** for content items

A Cache Network



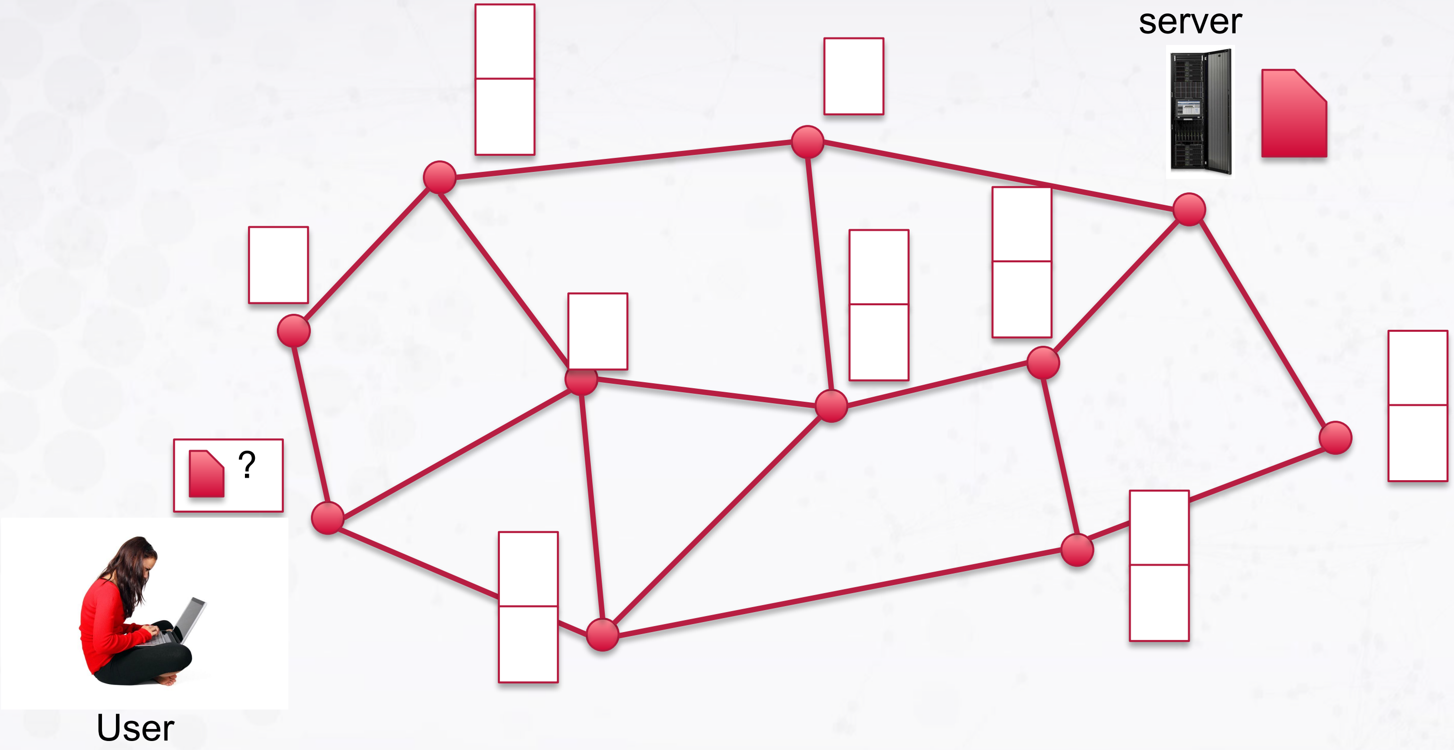
Requests routed towards a designated server

A Cache Network



Responses routed over **reverse** path

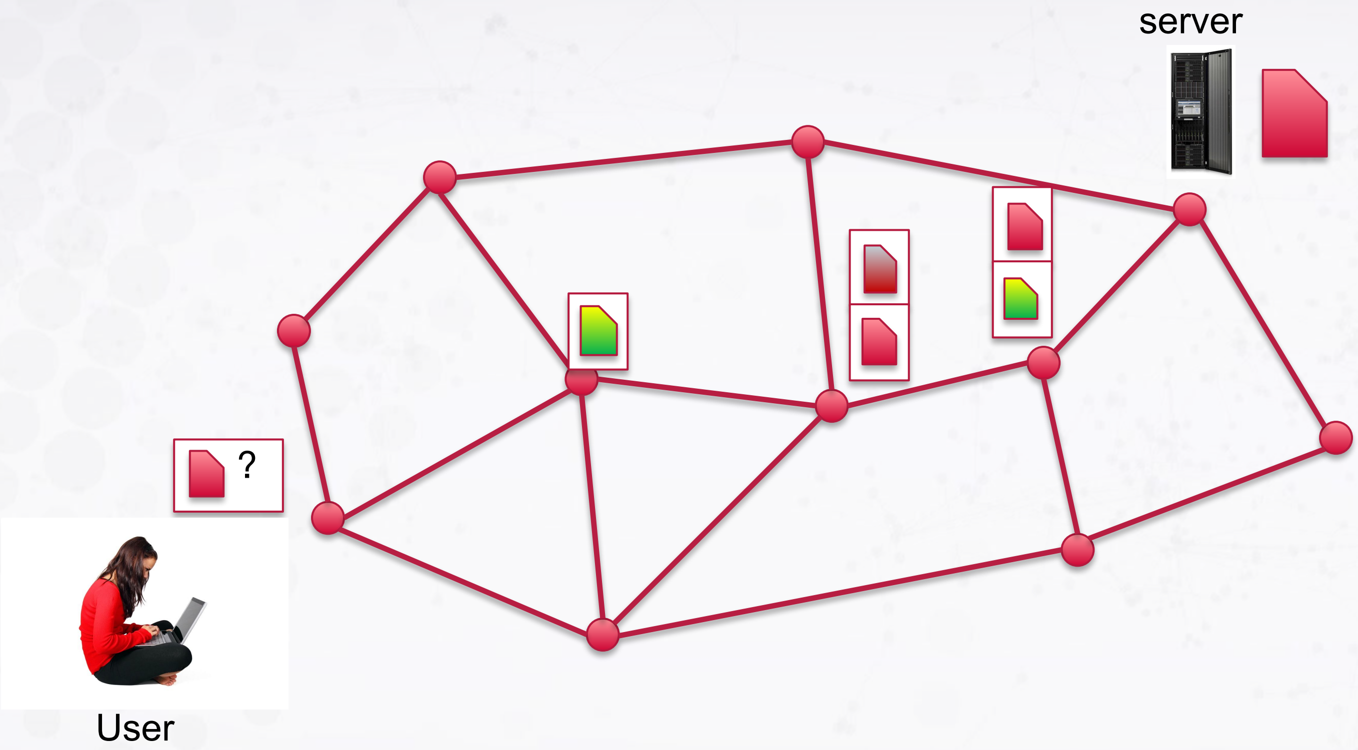
A Cache Network



Nodes have **caches** with finite capacities

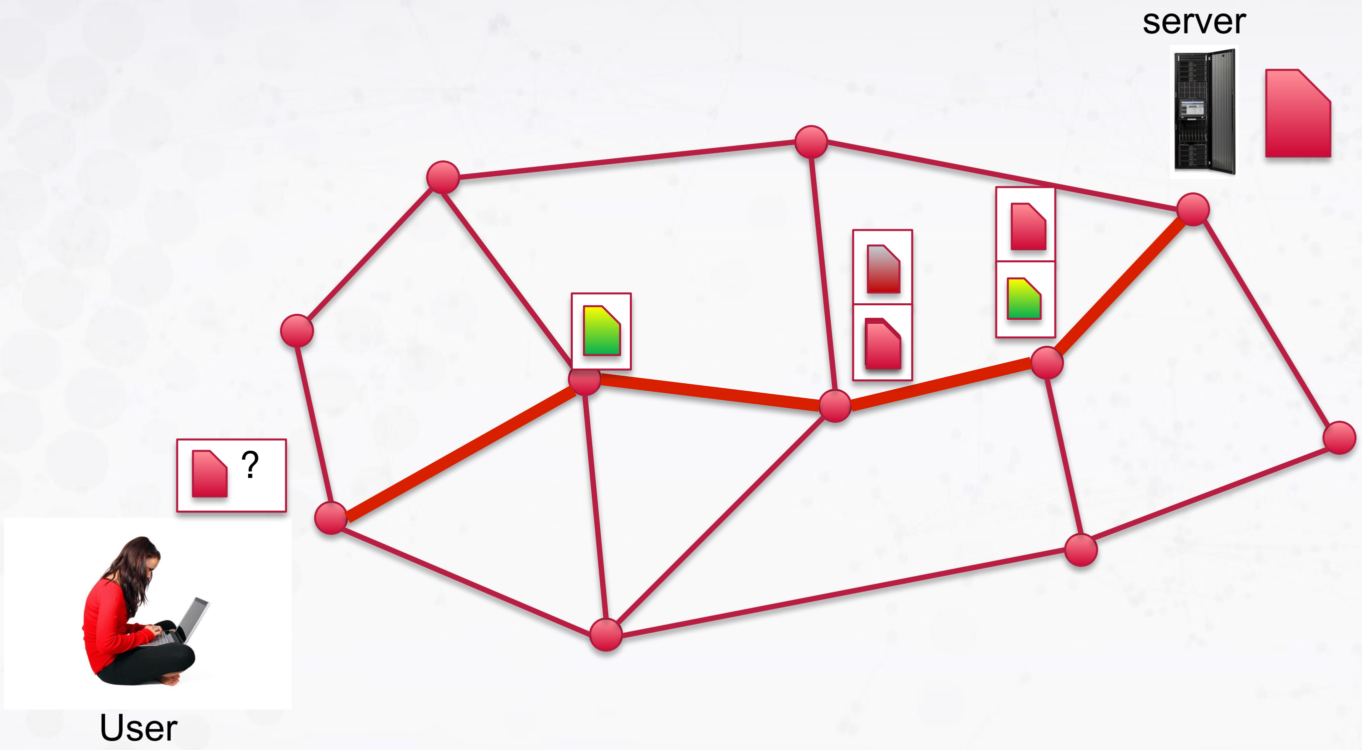
Institute for the Wireless Internet of Things at Northeastern

A Cache Network



Nodes have **caches** with finite capacities

A Cache Network



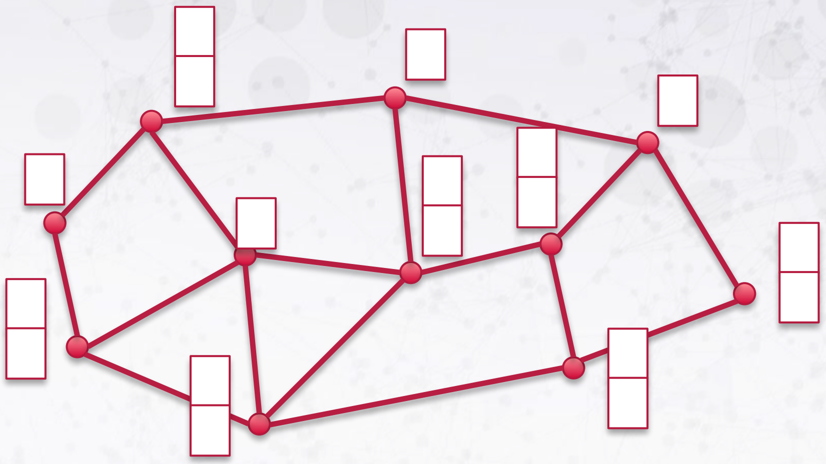
Requests terminate early upon a **cache hit**

Cache Network Problems

- **Cache Networks: nodes** can store **content.**

- Optimize **caching decisions**
- …plus:
  - **Routing**
  - **Scheduling/service allocation**
  - **Admission control**
  - …
- Minimize **delays** or **transfer costs**, maximize **throughput** or **utility**, incorporate **fairness,** study **stability** …

- **Distributed**, **adaptive** algorithms



Much, much **harder**, because **caching is combinatorial**!!!

Our Research Contributions

- Distributed, adaptive, algorithms optimizing **caching** decisions
  - Stochastic requests [I. and Yeh, SIGMETRICS 2016/ToN 2018]
  - Adversarial requests/no-regret setting [Li, Si Salem, Neglia, and I., SIGMETRICS 2022]
- Joint optimization of caching **and** routing [I. and Yeh, ICN 2017/JSAC 2018] [Li, Si Salem, Neglia, and I., SIGMETRICS 2022]
- Queuing Models
  - Kelly cache networks [Mahdian, Moharrer, I., and Yeh, INFOCOM 2019/ToN 2020]
  - Cache networks with counting queues [Li and I., INFOCOM 2020/ToN 2021]
  - Stability/admission control [Kamran, Moharrer, I., and Yeh, INFOCOM 2021]
- Fair caching networks [Liu, Li, I., and Yeh, Performance 2020]

Overview

- ❑ Cache network optimization
- ❑ Jointly optimizing caching and routing
- ❑ Introducing queues

Overview

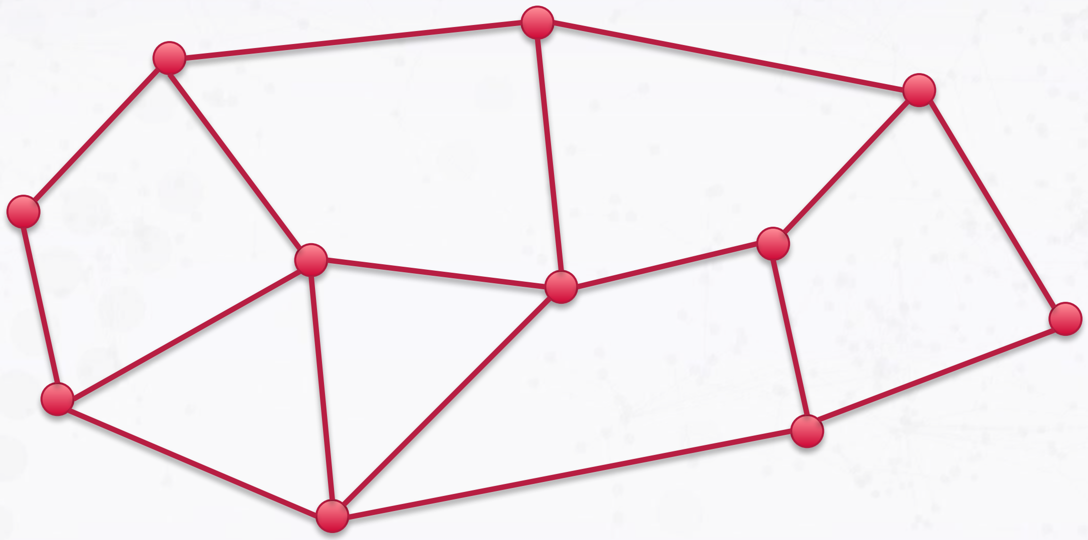
- ❑ Cache network optimization
- ❑ Jointly optimizing caching and routing
- ❑ Introducing queues

Institute for the Wireless Internet of Things at Northeastern

Model: Network

[I. and Yeh, SIGMETRICS 2016/ToN 2018]

$G(V, E)$



Network represented as a directed, bi-directional graph $G(V, E)$

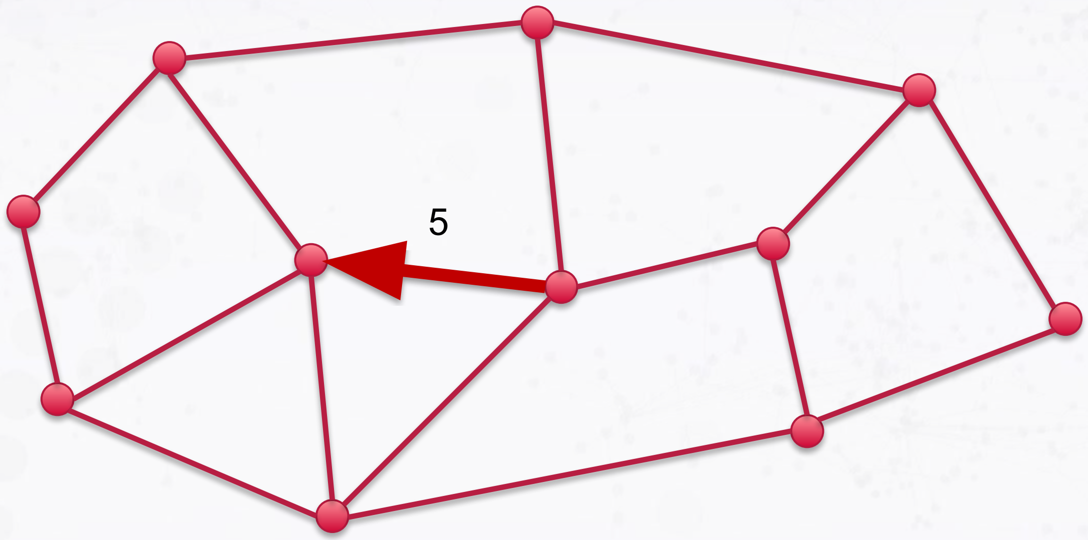
Institute for the Wireless Internet of Things at Northeastern

Model: Network

[I. and Yeh, SIGMETRICS 2016/ToN 2018]

$G(V, E)$

Edge costs: $w_{uv}, (u, v) \in E$



Each edge $(u, v) \in E$ has a cost/weight w_{uv}

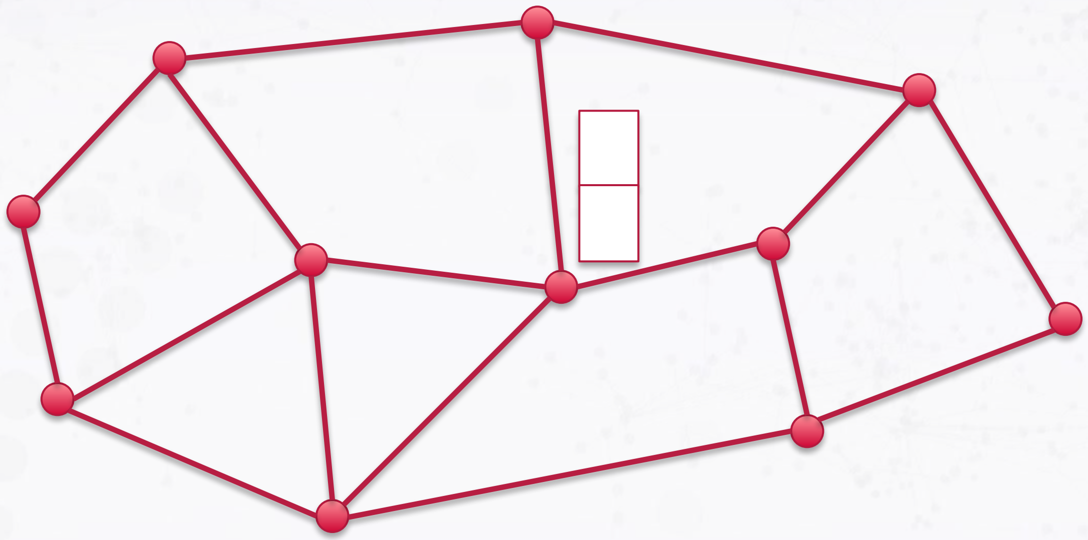
Model: Network

[I. and Yeh, SIGMETRICS 2016/ToN 2018]

$G(V, E)$

Edge costs: $w_{uv}, (u,v) \in E$

Node capacities: $c_v, v \in V$



Node $v \in V$ has a cache with capacity $c_v \in \mathbb{N}$

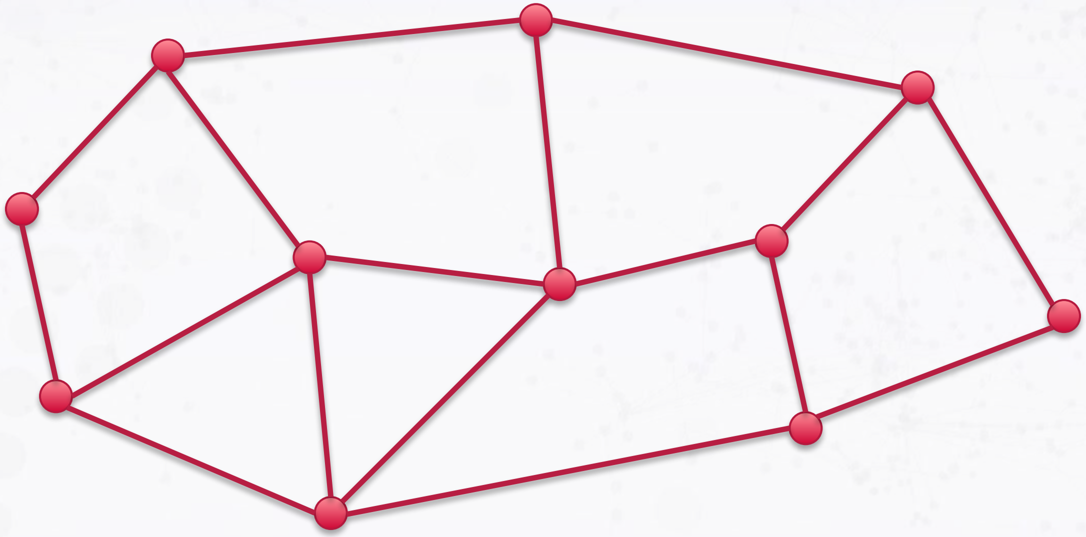
Institute for the Wireless Internet of Things at Northeastern

Model: Network

[I. and Yeh, SIGMETRICS 2016/ToN 2018]

$G(V, E)$

$\mathcal{C} = \{$  $\}$



Edge costs: $w_{uv}, (u, v) \in E$

Node capacities: $c_v, v \in V$

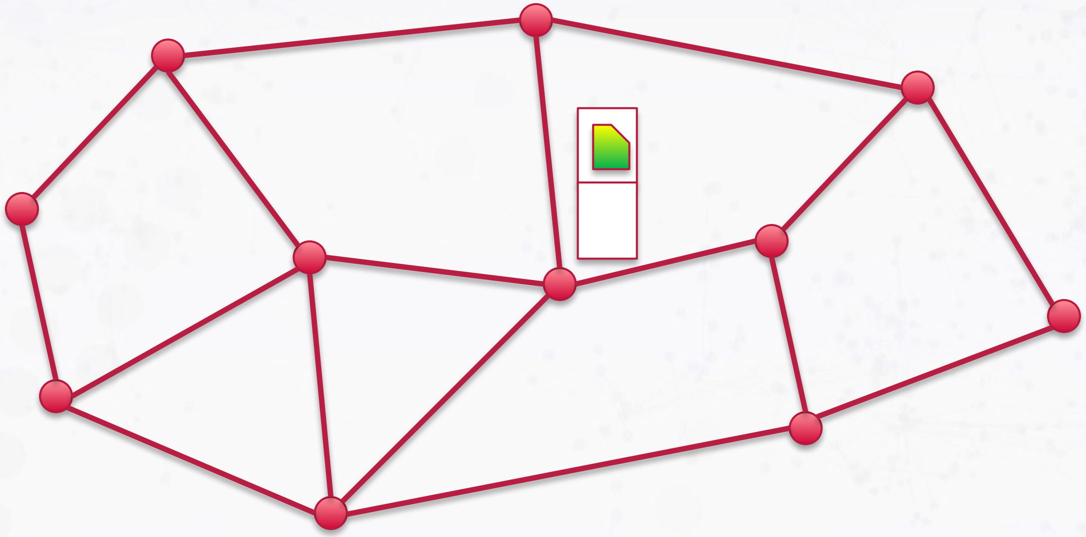
Items stored and requested form the **item catalog** $\mathcal{C}$

Model: Network

[I. and Yeh, SIGMETRICS 2016/ToN 2018]

$G(V, E)$

$\mathcal{C} = \{$  $\}$



Edge costs: $w_{uv}, (u, v) \in E$

Node capacities: $c_v, v \in V$

For $v \in V$ and $i \in \mathcal{C}$, let $x_{vi} = \begin{cases} 1, & \text{if } v \text{ stores } i \\ 0, & \text{o.w.} \end{cases}$

Then, for all $v \in V$, $\sum_{i \in \mathcal{C}} x_{vi} \leq c_v$

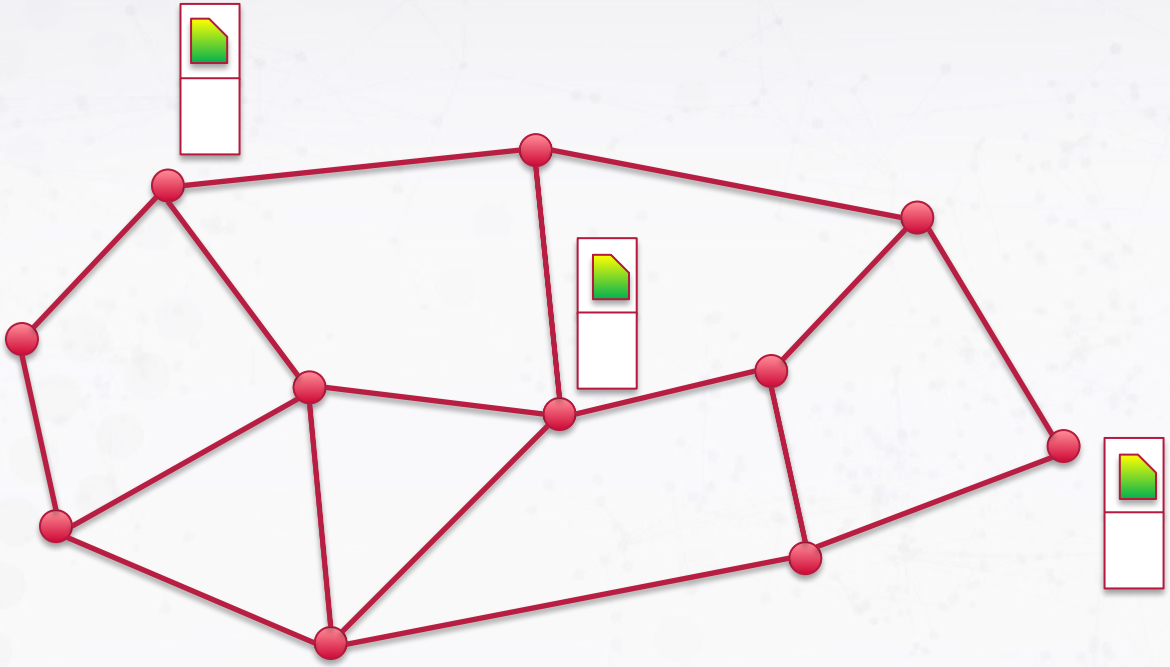
Institute for the Wireless Internet of Things at Northeastern

Model: Designated/Permanent Servers

[I. and Yeh, SIGMETRICS 2016/ToN 2018]

$G(V, E)$

$\mathcal{C} = \{$  $\}$



Edge costs: $w_{uv}, (u, v) \in E$

Node capacities: $c_v, v \in V$

$\sum_{i \in \mathcal{C}} x_{vi} \leq c_v$, for all $v \in V$

For each and $i \in \mathcal{C}$, there exists a set of nodes $S_i \subset V$ (the **designated servers** of i) that **permanently store** i.

I.e., if $v \in S_i$ then $x_{vi} = 1$

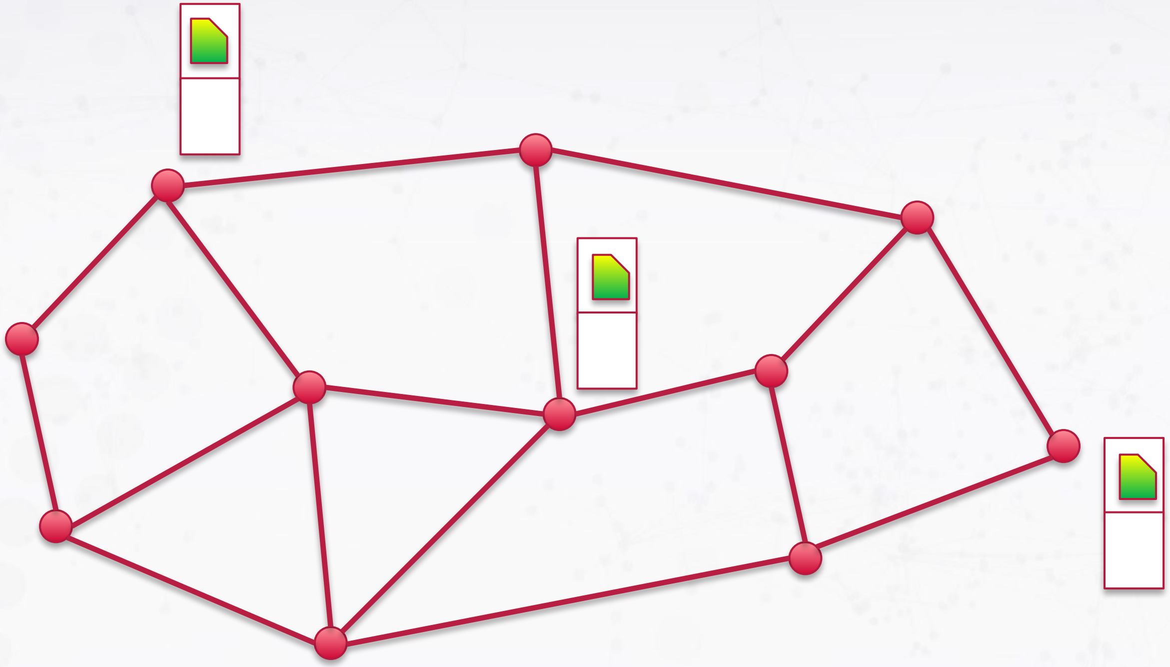
Institute for the Wireless Internet of Things at Northeastern

Model: Designated/Permanent Servers

[I. and Yeh, SIGMETRICS 2016/ToN 2018]

$G(V, E)$

$\mathcal{C} = \{$  $\}$



Edge costs: $w_{uv}, (u,v) \in E$

Node capacities: $c_v, v \in V$

$\sum_{i \in \mathcal{C}} x_{vi} \leq c_v$, for all $v \in V$

For each and $i \in \mathcal{C}$, there exists a set of nodes $S_i \subset V$ (the **designated servers** of i) that **permanently store** i.

I.e., if $v \in S_i$ then $x_{vi} = 1$

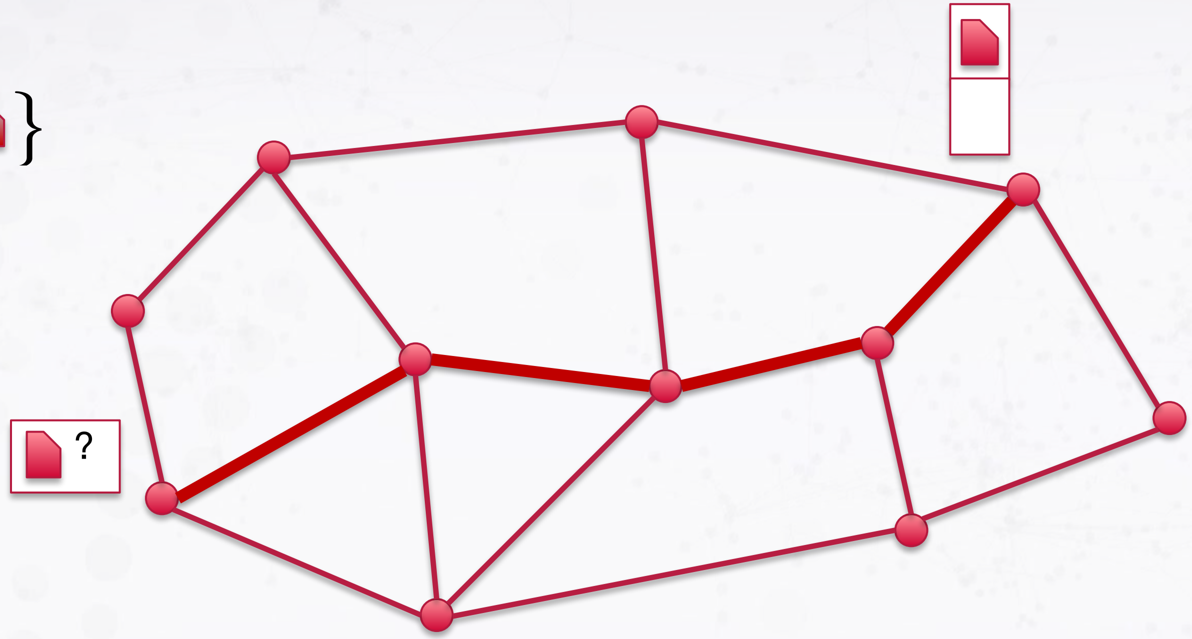
Institute for the Wireless Internet of Things at Northeastern

Model: Demand

[I. and Yeh, SIGMETRICS 2016/ToN 2018]

$G(V, E)$

$\mathcal{C} = \{$  $\}$



Edge costs: $w_{uv}, (u, v) \in E$

Node capacities: $c_v, v \in V$

$\sum_{i \in \mathcal{C}} x_{vi} \leq c_v$, for all $v \in V$

Requests are always satisfied!

A **request** is a pair (i, p) such that:

- ❑ i is an item in $\mathcal{C}$
- ❑ $p = \{p_1, \ldots, p_K\}$ is a simple path in G such that $p_K \in S_i$.

Institute for the Wireless Internet of Things at Northeastern

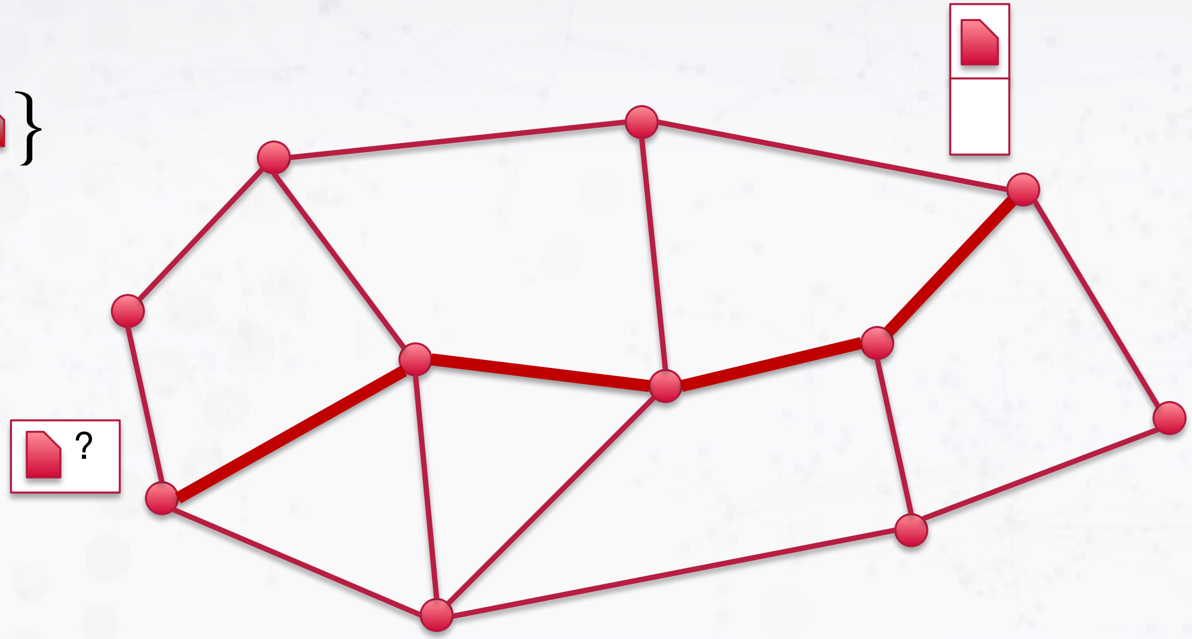
Model: Demand

[I. and Yeh, SIGMETRICS 2016/ToN 2018]

$G(V, E)$

$\mathcal{C} = \{ \quad \}$

$\mathcal{R}$: demand



Edge costs: $w_{uv}, (u,v) \in E$

Node capacities: $c_v, v \in V$

$\sum_{i \in \mathcal{C}} x_{vi} \leq c_v$, for all $v \in V$

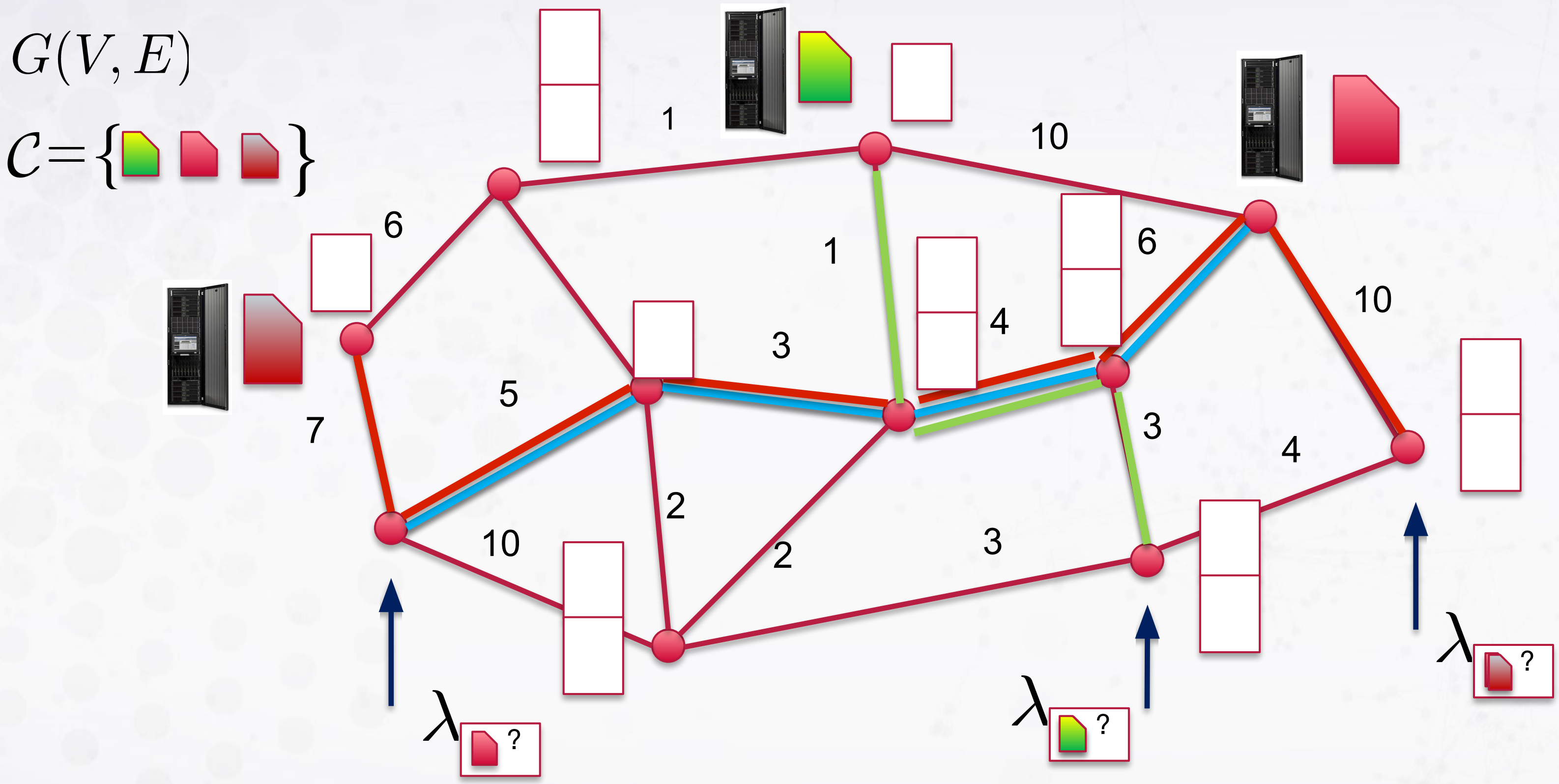
Request rates: $\lambda_{(i,p)}, (i,p) \in \mathcal{R}$

Demand $\mathcal{R}$: set of all requests (i, p)

Request arrival process is Poisson with rate $\lambda_{(i,p)}$

Institute for the Wireless Internet of Things at Northeastern

Model: Goal



Edge costs: $w_{uv}, (u,v) \in E$

Node capacities: $c_v, v \in V$

$\sum_{i \in \mathcal{C}} x_{vi} \leq c_v$, for all $v \in V$

Request rates: $\lambda_{(i,p)}, (i,p) \in \mathcal{R}$

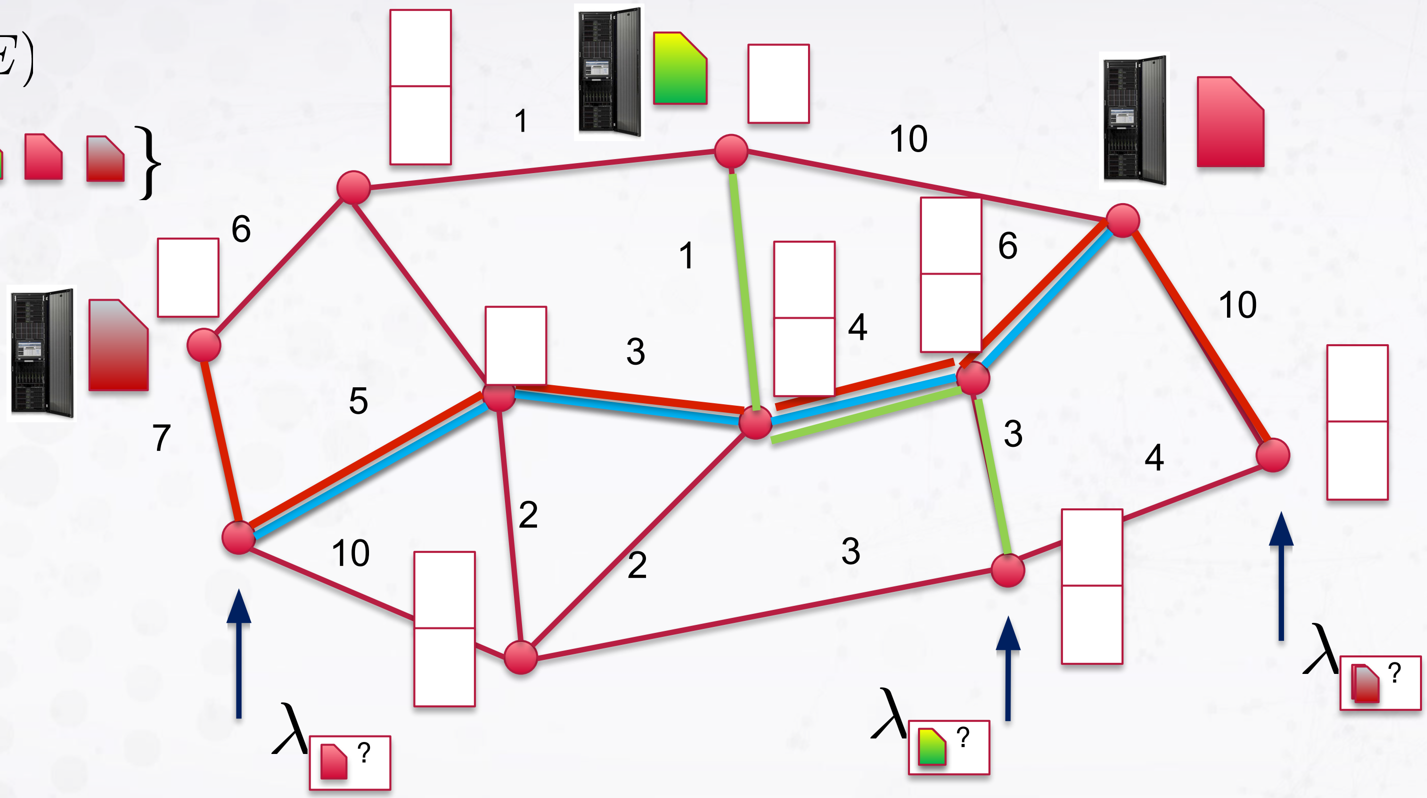
Design content allocation so that expected **transfer costs** are **minimized.**

Institute for the Wireless Internet of Things at Northeastern

Model: Goal

$G(V, E)$

$\mathcal{C} = \{$ ... $\}$



Edge costs: $w_{uv}, (u,v) \in E$

Node capacities: $c_v, v \in V$

$\sum_{i \in \mathcal{C}} x_{vi} \leq c_v$, for all $v \in V$

Request rates: $\lambda_{(i,p)}, (i,p) \in \mathcal{R}$

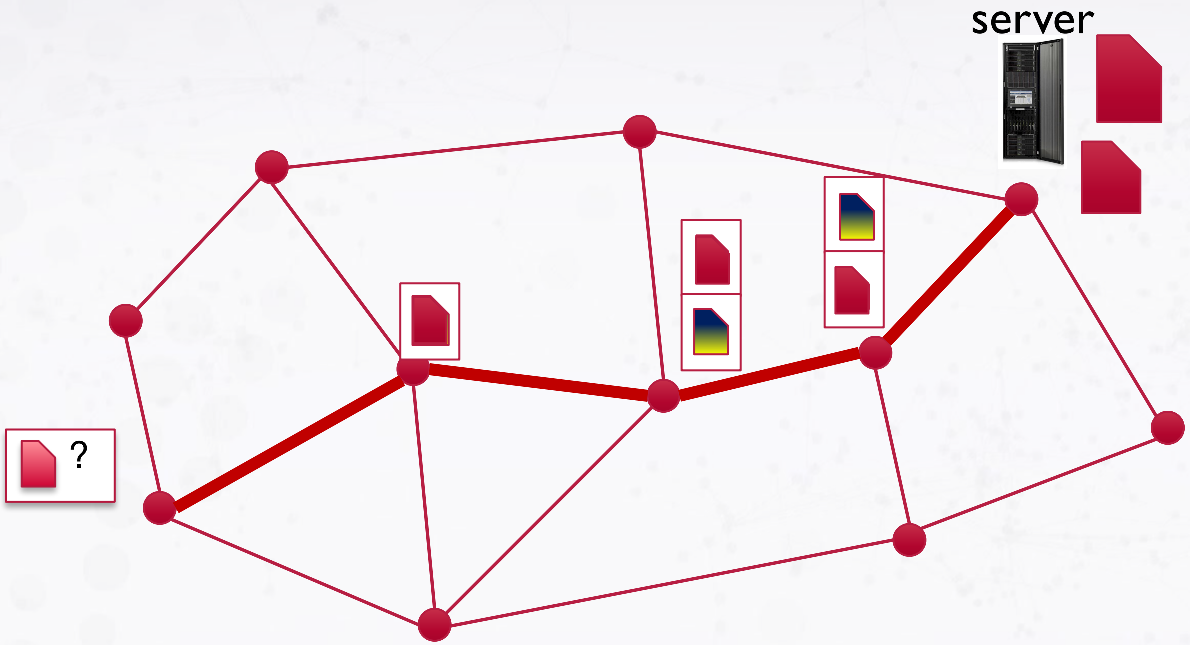
Challenge: Caching algorithm should be

- ❑ **adaptive**, and
- ❑ **distributed.**

Institute for the Wireless Internet of Things at Northeastern

A Simple Algorithm: Path Replication + LRU

[Cohen and Shenker 2002]
[Jacobson et al. 2009]



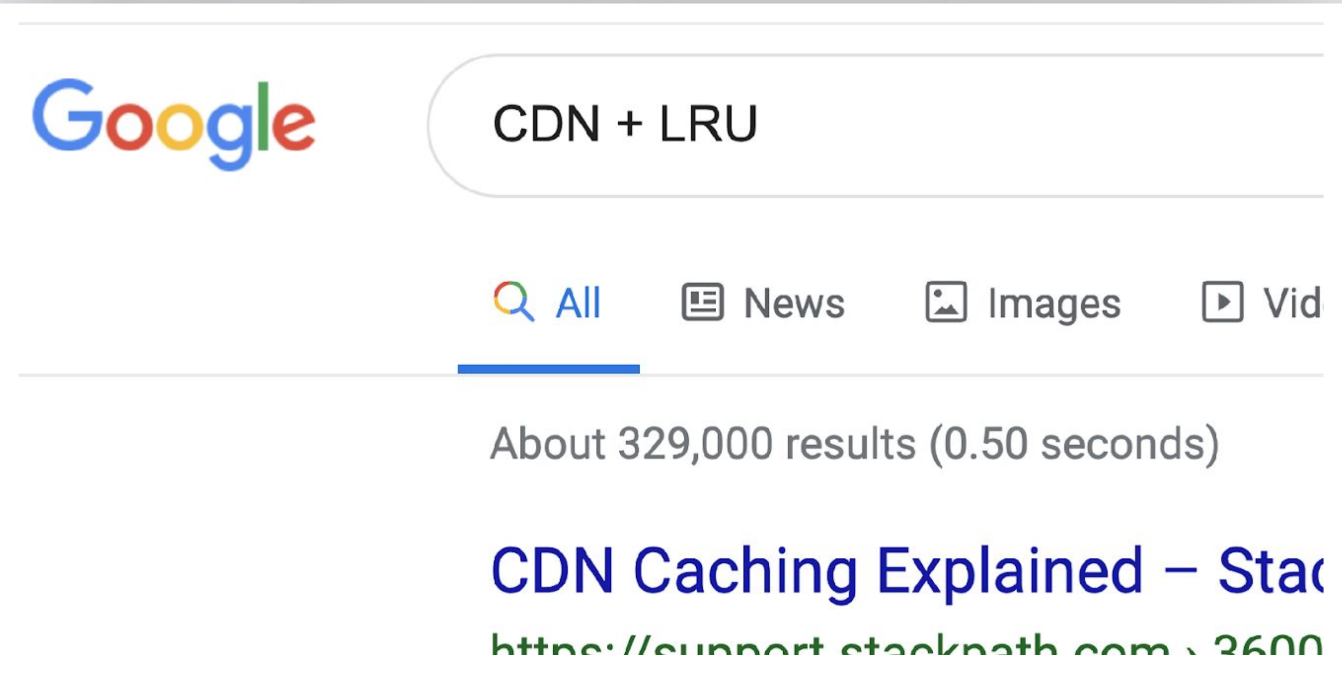
✔ Distributed
✔ Adaptive
✔ Extremely Popular

- ❑ Cache item on every node in the reverse path
- ❑ Evict using a simple policy, e.g., LRU, LFU, FIFO etc.

- ❑ Many variants: Move-Copy-Down (MCD), Leave-Copy-Down (LCD)…

A Simple Algorithm: Path Replication + LRU

[Cohen and Shenker 2002]
[Jacobson et al. 2009]



✔ Distributed
✔ Adaptive
✔ Extremely Popular

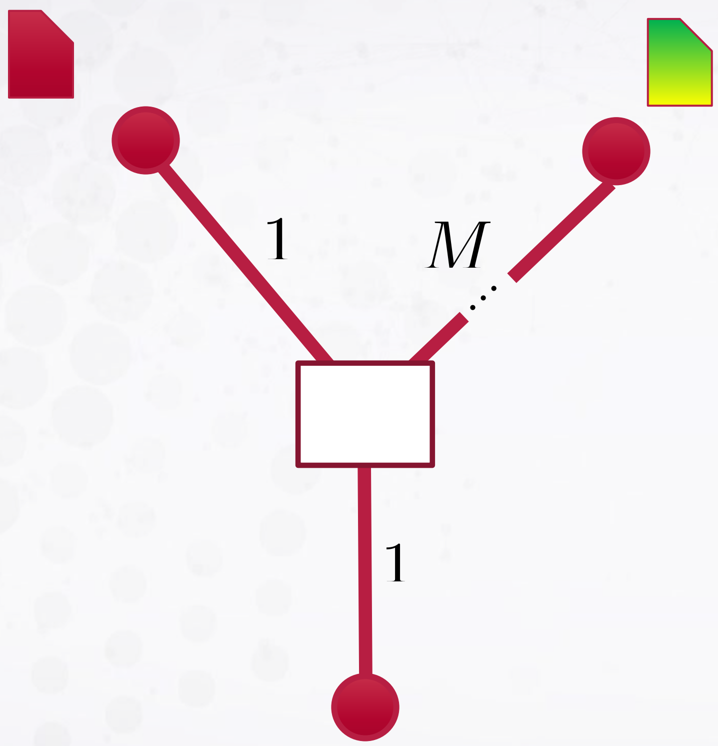
❑ Cache item on every node in the reverse path
❑ Evict using a simple policy, e.g., LRU, LFU, FIFO etc.

❑ Many variants: Move-Copy-Down (MCD), Leave-Copy-Down (LCD)…

But…

Path Replication + LRU is **arbitrarily suboptimal.**

Path Replication + LRU is Arbitrarily Suboptimal



$\lambda_{?} = \lambda_{?} = 0.5$ requests per sec

Cost when caching

$0.5 \times 1 + 0.5 \times 2 = \mathbf{1.5}$

Cost of PR+LRU:

$$0.25\times(M+1) + 0.25 \times 1 + \\ + 0.25 \times 2 + 0.25 \times 1 = \mathbf{0.25}M\mathbf{+1.25}$$

- When M is large, PR+LRU is **arbitrarily suboptimal!**
- True for any strategy (LRU,LFU,FIFO,RR+LCD,MCD) that **ignores upstream costs!!**

[I. and Yeh, SIGMETRICS 2016/ToN 2018]

Institute for the Wireless Internet of Things at Northeastern

Model: Routing Costs & Caching Gain

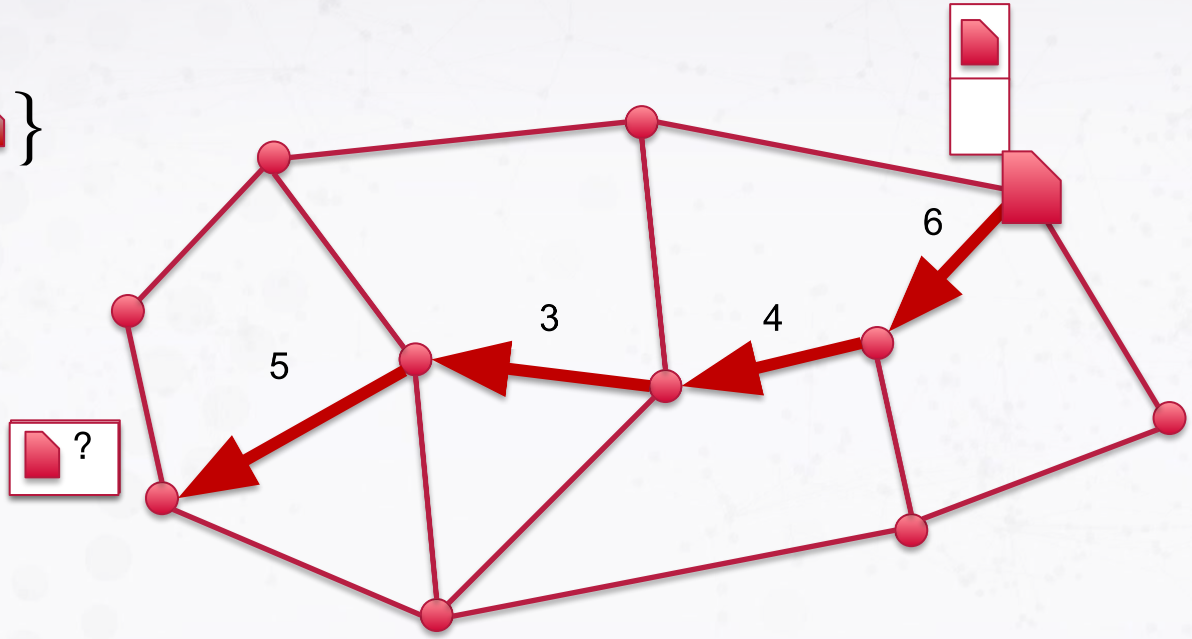
[I. and Yeh, SIGMETRICS 2016/ToN 2018]

$G(V, E)$

$\mathcal{C} = \{$  $\}$

$\mathcal{R}$: demand

Request (i, p)



Edge costs: $w_{uv}, (u, v) \in E$

Node capacities: $c_v, v \in V$

$\sum_{i \in \mathcal{C}} x_{vi} \leq c_v$, for all $v \in V$

Request rates: $\lambda_{(i,p)}, (i, p) \in \mathcal{R}$

Worst case routing cost: **18**

Institute for the Wireless Internet of Things at Northeastern

Model: Routing Costs & Caching Gain

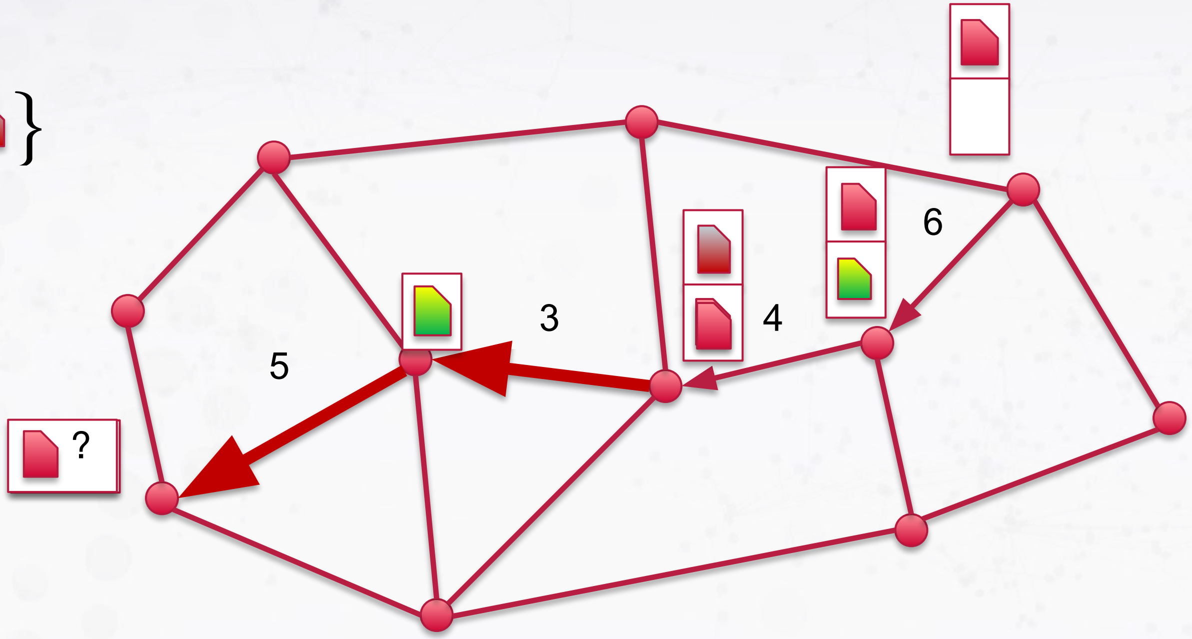
[I. and Yeh, SIGMETRICS 2016/ToN 2018]

$G(V, E)$

$\mathcal{C} = \{$  $\}$

$\mathcal{R}$: demand

Request (i, p)



Edge costs: $w_{uv}, (u, v) \in E$

Node capacities: $c_v, v \in V$

$\sum_{i \in \mathcal{C}} x_{vi} \leq c_v$, for all $v \in V$

Request rates: $\lambda_{(i,p)}, (i, p) \in \mathcal{R}$

Worst case routing cost: **18**

Cost due to intermediate caching: **8**

Institute for the Wireless Internet of Things at Northeastern

Model: Routing Costs & Caching Gain

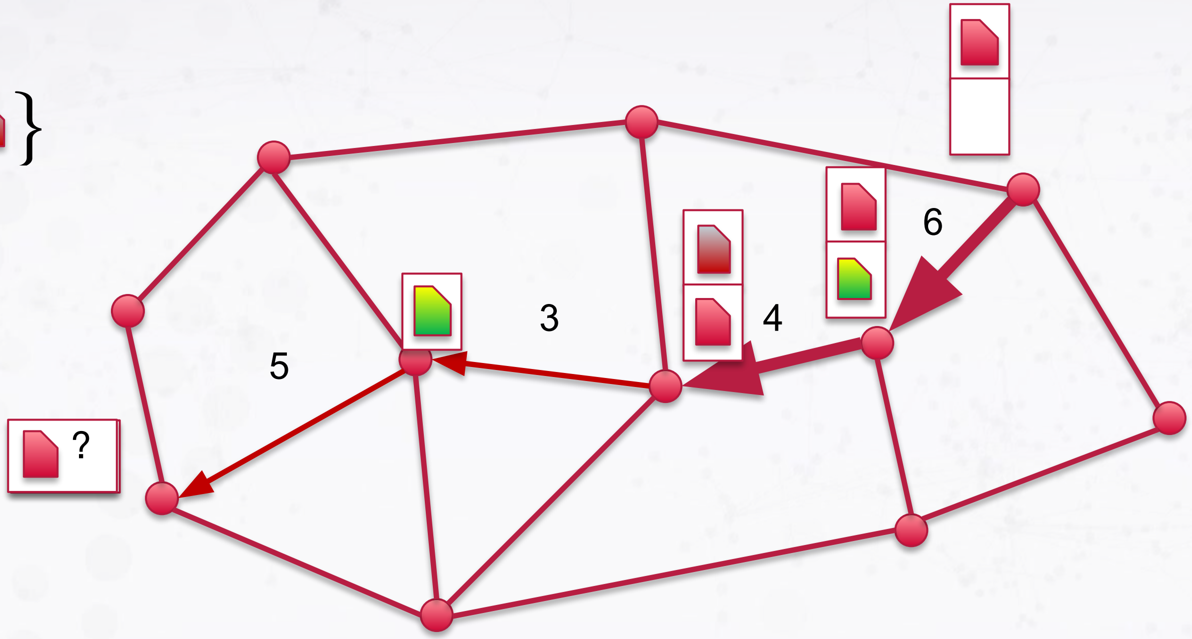
[I. and Yeh, SIGMETRICS 2016/ToN 2018]

$G(V,E)$

$\mathcal{C}=\{$ ▪ ▪ ▪ $\}$

$\mathcal{R}$: demand

Request (i,p)



Edge costs: $w_{uv}, (u,v) \in E$

Node capacities: $c_v, v \in V$

$\sum_{i \in \mathcal{C}} x_{vi} \leq c_v$, for all $v \in V$

Request rates: $\lambda_{(i,p)}, (i,p) \in \mathcal{R}$

Worst case routing cost: **18**

Cost due to intermediate caching: **8**

Caching Gain: **18-8 = 10**

Institute for the Wireless Internet of Things at Northeastern

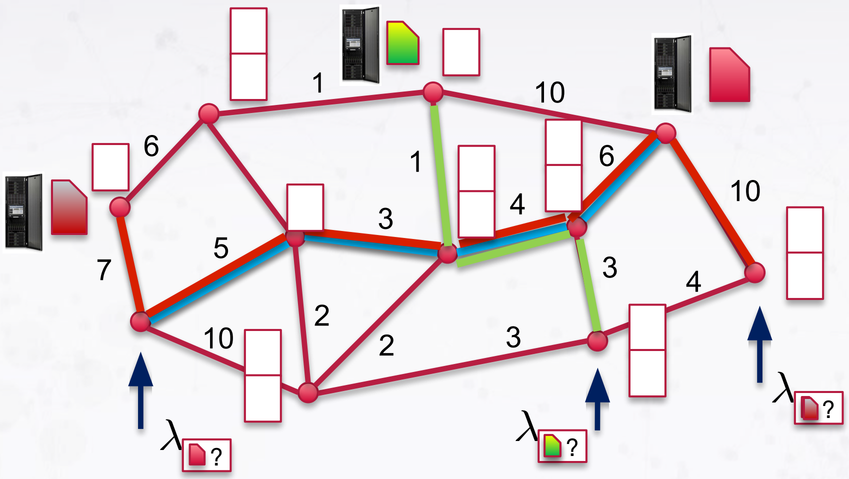
Objective: Maximizing Caching Gain

[I. and Yeh, SIGMETRICS 2016/ToN 2018]

$G(V, E)$

$\mathcal{C} = \{$ ▪ ▪ ▪ $\}$

$\mathcal{R}$: demand



Edge costs: $w_{uv}, (u,v) \in E$

Node capacities: $c_v, v \in V$

$\sum_{i \in \mathcal{C}} x_{vi} \leq c_v$, for all $v \in V$

Request rates: $\lambda_{(i,p)}, (i,p) \in \mathcal{R}$

MaxCG

Maximize: $$F(X) = \sum_{(i,p)\in\mathcal{R}} \lambda_{(i,p)} \sum_{k=1}^{|p|-1} w_{p_{k+1}p_k} \left(1 - \prod_{k'=1}^{k}(1 - x_{p_{k'}i})\right)$$

Subject to:

$\sum_{i \in \mathcal{C}} x_{vi} = c_v,$ for all $v \in V$

$x_{vi} = 1,$ for all $i \in \mathcal{C}$ and $v \in S_i$

$x_{vi} \in \{0,1\},$ for all $v \in V$ and $i \in \mathcal{C}$

Institute for the Wireless Internet of Things at Northeastern

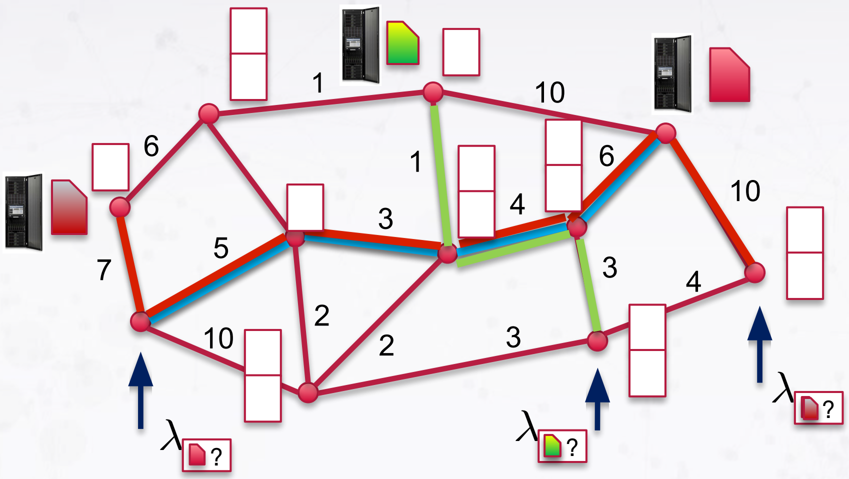
Objective: Maximizing Caching Gain

[I. and Yeh, SIGMETRICS 2016/ToN 2018]

$G(V,E)$

$\mathcal{C}=\{\ \}$

$\mathcal{R}$: demand



Edge costs: $w_{uv}, (u,v)\in E$

Node capacities: $c_v, v\in V$

$\sum_{i\in\mathcal{C}} x_{vi} \leq c_v$, for all $v\in V$

Request rates: $\lambda_{(i,p)}, (i,p)\in\mathcal{R}$

MaxCG

Maximize: $F(X) = \sum_{(i,p)\in\mathcal{R}} \lambda_{(i,p)} \sum_{k=1}^{|p|-1} w_{p_{k+1}p_k}\left(1-\prod_{k'=1}^{k}(1-x_{p_{k'}i})\right)$

Subject to:

$\sum_{i\in\mathcal{C}} x_{vi} = c_v$ for all $v\in V$

$x_{vi}=1,$ for all $i\in\mathcal{C}$ and $v\in S_i$

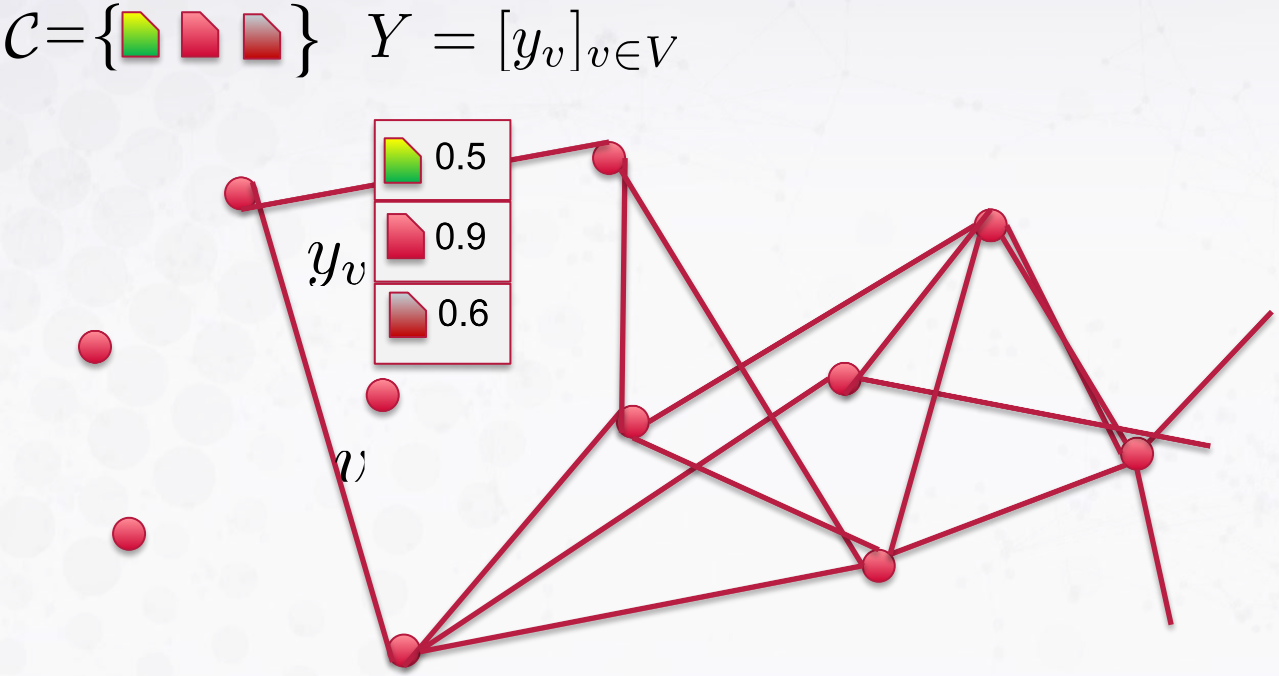
$x_{vi}\in\{0,1\},$ for all $v\in V$ and $i\in\mathcal{C}$

- **NP-hard** but…
- .. **Submodular maximization** under **matroid** constraints
- **1-1/e polytime approximation** algorithm

Institute for the Wireless Internet of Things at Northeastern

Distributed, Adaptive Algorithm

- Each cache maintains **state**
- **State = probability** of caching item

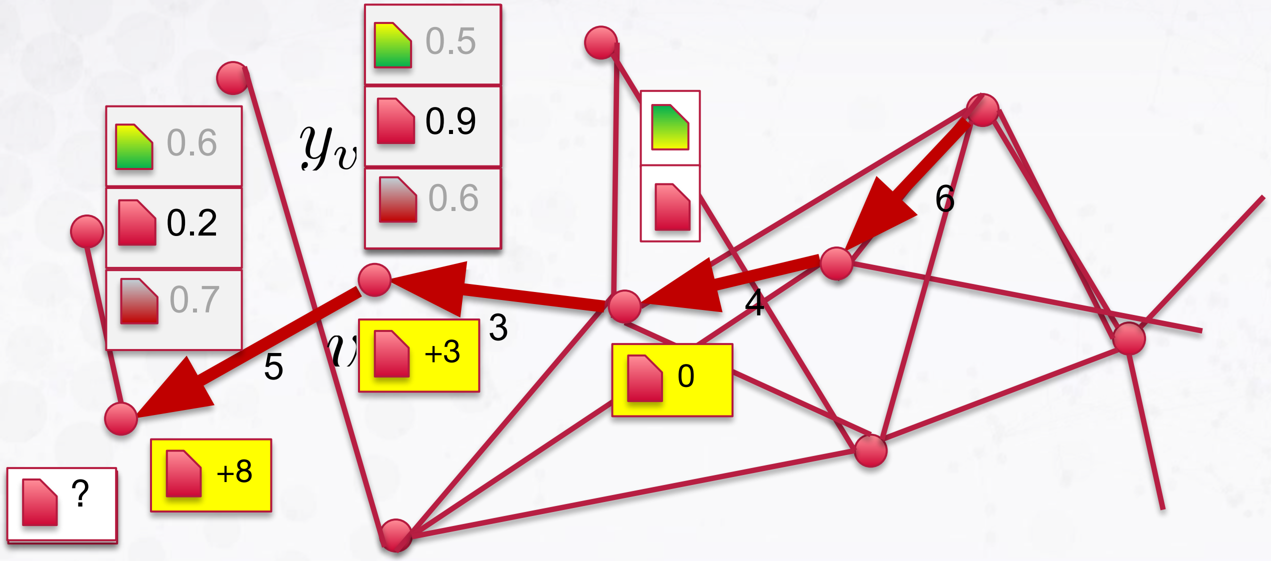


$Y = [y_v]_{v \in V}$

[I. and Yeh, SIGMETRICS 2016/ToN 2018]

Distributed, Adaptive Algorithm

$\mathcal{C}=\{$ ▲ ▲ ▲ $\}$ $Y = [y_v]_{v\in V}$



Theorem: The proposed algorithm leads to an allocation X_k such that

$$\lim_{k\to\infty} \mathbb{E}[F(X_k)] \geq (1-\frac{1}{e})F(X^*)$$

where X^* an optimal solution to the (NP-hard) offline problem.

- Each cache maintains **state**
- **State = probability** of caching item
- Upon request, control message collects information about **upstream costs**

  = gradient of concave relaxation of objective (in expectation)

- During slot of length T, **average** upstream costs
- At **end of slot**, **adapt state** and **refresh** contents by **randomly sampling** from distribution/state, independently across nodes.
- **"value"** of item ▲ is

$\lambda_{?} \times \mathbb{E}[$upstream cost upon ▲ miss$]$

[I. and Yeh, SIGMETRICS 2016/ToN 2018]

Institute for the Wireless Internet of Things at Northeastern

No-Regret Algorithms

[Li, Si Salem, Neglia, and I., SIGMETRICS 2022]

- Theorem assumes:
  - Stationary, stochastic request arrivals
  - Negligible costs for updates

No-Regret Algorithms

[Li, Si Salem, Neglia, and I., SIGMETRICS 2022]

- ❑ **Arbitrary**, **adversarial** request arrivals per time-slot
- ❑ Account for **update costs**

Theorem: A **distributed**, **online** algorithm that attains regret

$$R(T) = (1 - \frac{1}{e}) \sum_{t=1}^{T} F_t(X^*) - \left(\sum_{t=1}^{T} F_t(X_t) - \sum_{t=1}^{T} \mathrm{UC}(X_t, X_{t-1}) \right) = O(\sqrt{T})$$

optimal offline static policy

caching gain of online policy

penalty for update costs

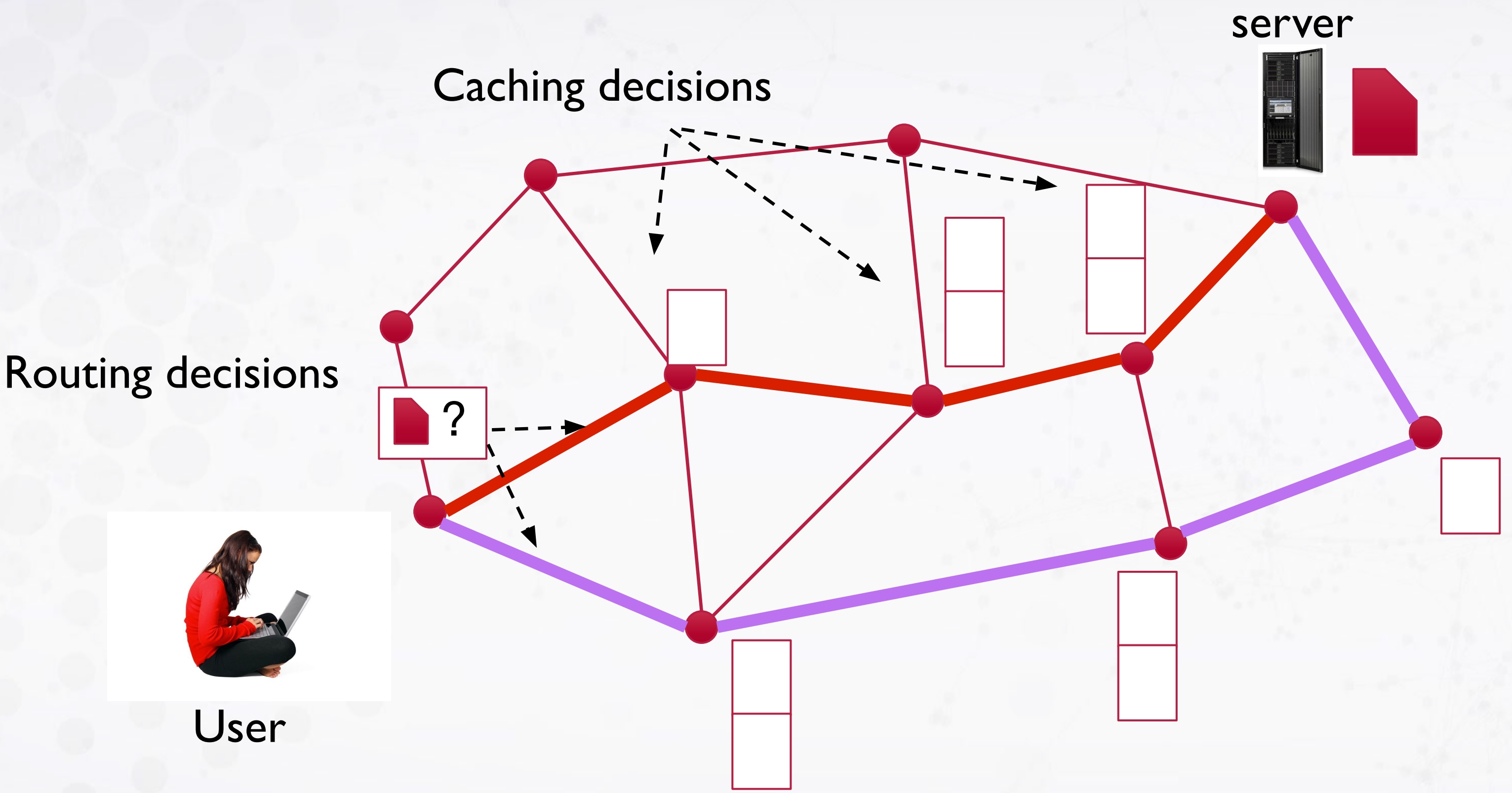
Institute for the Wireless Internet of Things at Northeastern

Overview

- Cache network optimization
- Jointly optimizing caching and routing
- Introducing queues

Joint Optimization

[I. and Yeh, ICN 2017/JSAC 2018]

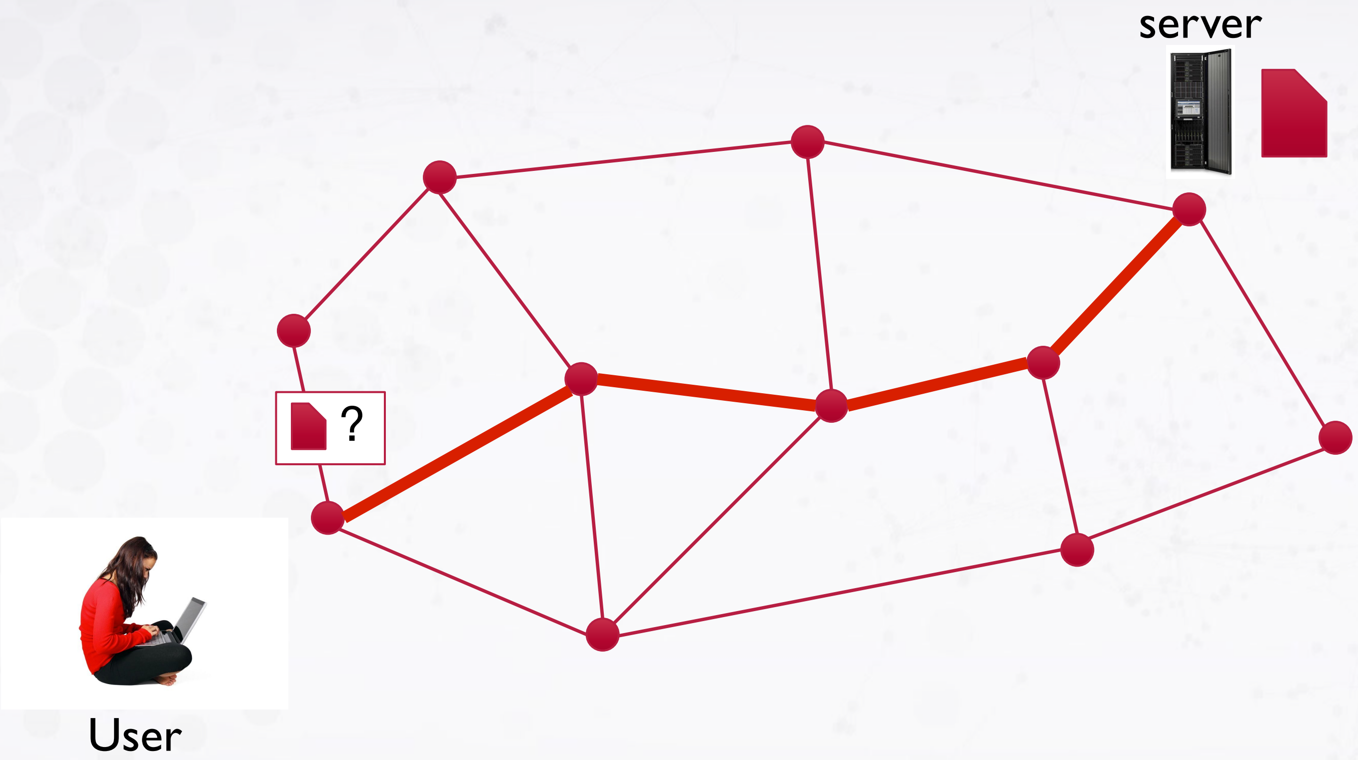


❑ Both **caching and routing** decisions are part of optimization

Institute for the Wireless Internet of Things at Northeastern

Is Joint Optimization Really Necessary?

[I. and Yeh, ICN 2017/JSAC 2018]

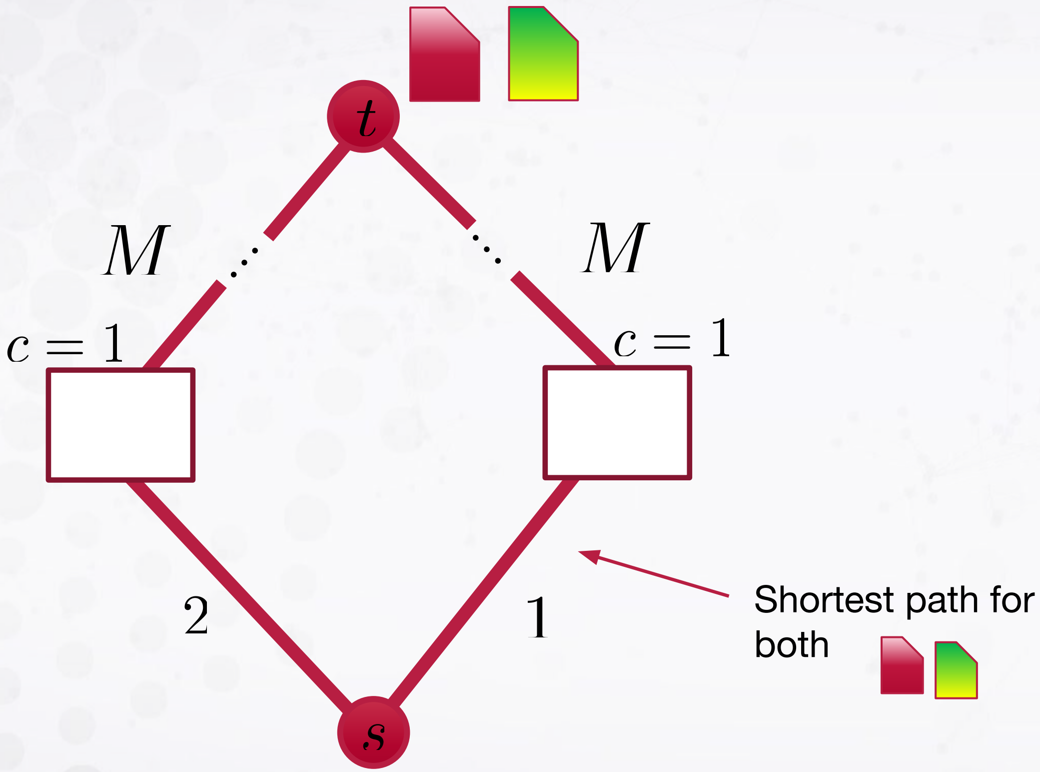


Shortest Weight Path

- ❑ Why not just use **shortest weight path** routing towards **nearest designated server**?

Shortest Path Routing is Arbitrarily Suboptimal

[I. and Yeh, ICN 2017/JSAC 2018]

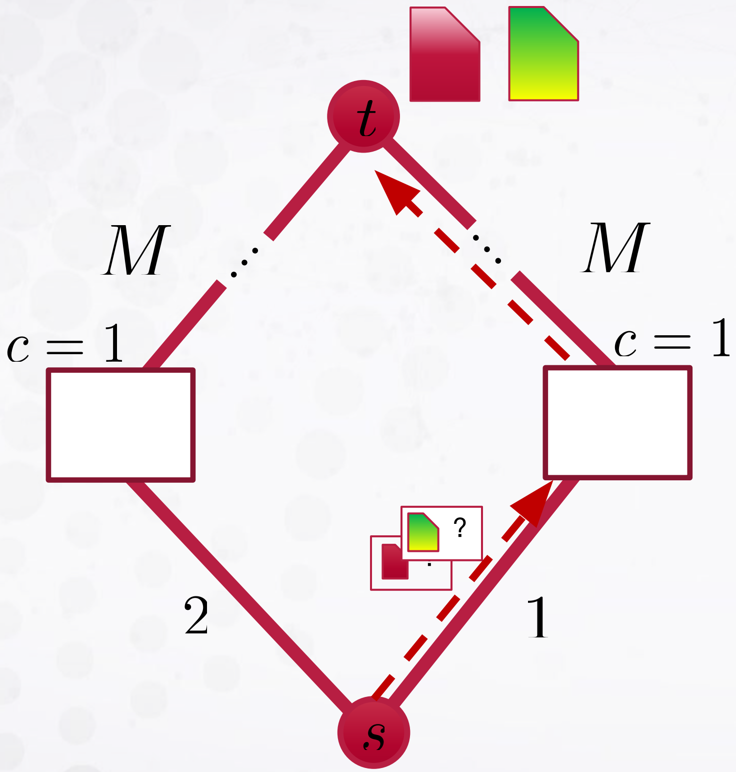


$\lambda_{?} = \lambda_{?} = 0.5$ requests per sec

Institute for the Wireless Internet of Things at Northeastern

Shortest Path Routing is Arbitrarily Suboptimal

[I. and Yeh, ICN 2017/JSAC 2018]

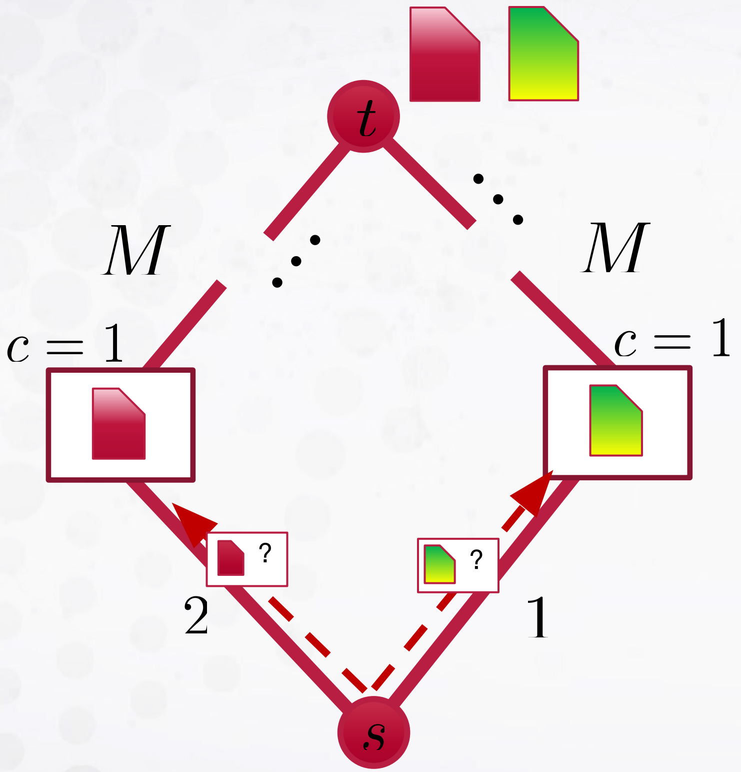


$\lambda = \lambda = 0.5$ requests per sec

Irrespective of caching algorithm used, cost under shortest path routing is $\Theta(M)$

Shortest Path Routing is Arbitrarily Suboptimal

[I. and Yeh, ICN 2017/JSAC 2018]



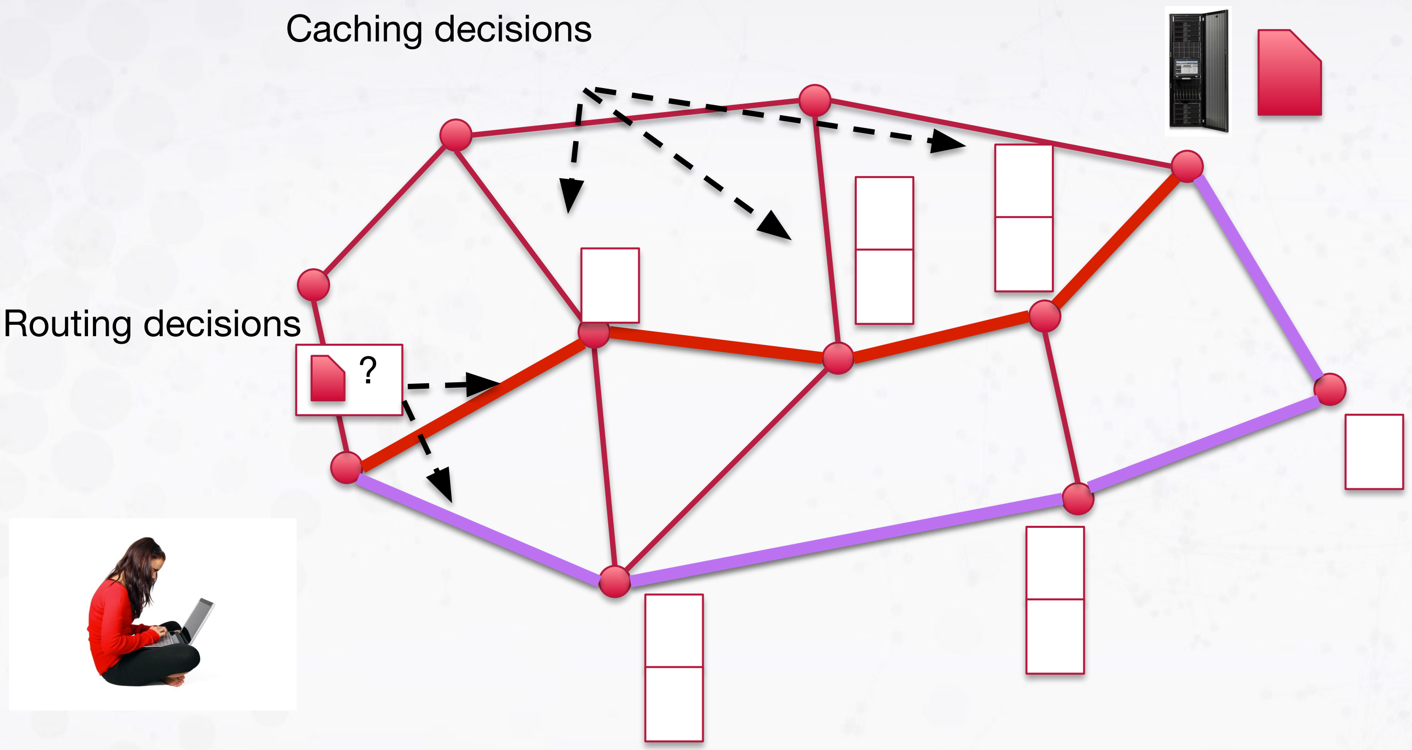
$\lambda = \lambda = 0.5$ requests per sec

Irrespective of caching algorithm used, cost under shortest path routing is $\Theta(M)$

Cost under "split" routing strategy is $O(1)$.

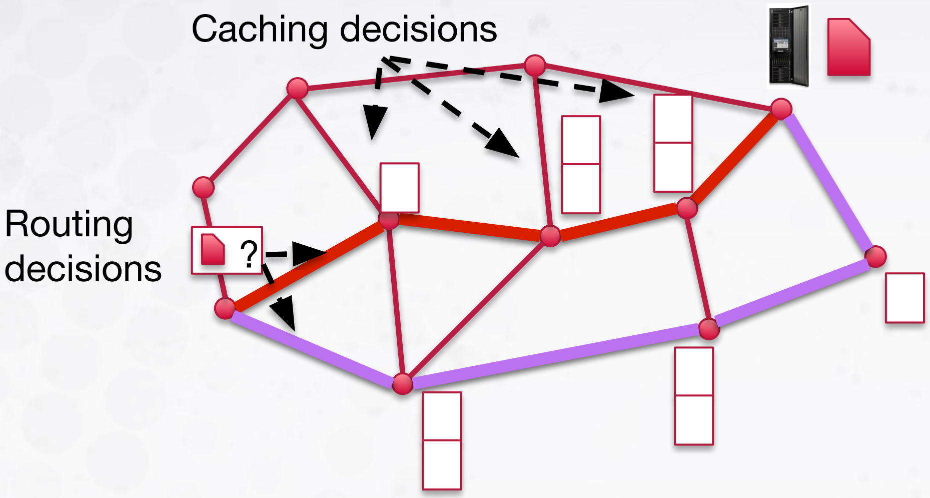
Shortest path routing to nearest server **is arbitrarily suboptimal.**

Key Intuition



Increasing **path diversity** creates
more caching opportunities.

Algorithms with Guarantees



Joint optimization of caching **and** routing

- ❑ **Stochastic** requests [I. and Yeh, ICN 2017/JSAC 2018]
  - **Distributed**, **adaptive** algorithm within 1-1/e from the optimal
- ❑ **Adversarial** requests [Li, Si Salem, Neglia, and I., SIGMETRICS 2022]
  - **Distributed**, **online** algorithm with $O(\sqrt{T})$ regret w.r.t. 1-1/e from the optimal offline solution

Institute for the Wireless Internet of Things at Northeastern

Experiments

[I. and Yeh, ICN 2017/JSAC 2018]

Graph Topologies

Graph	$\|V\|$	$\|E\|$	$\|\mathcal{C}\|$	$\|\mathcal{R}\|$	c_v	$\|\mathcal{P}_{(i,s)}\|$
cycle	30	60	10	100	2	2
grid-2d	100	360	300	1K	3	30
hypercube	128	896	300	1K	3	30
expander	100	716	300	1K	3	30
erdos-renyi	100	1042	300	1K	3	30
regular	100	300	300	1K	3	30
watts-strogatz	100	400	300	1K	3	2
small-world	100	491	300	1K	3	30
barabasi-albert	100	768	300	1K	3	30
geant	22	66	10	100	2	10
abilene	9	26	10	90	2	10
dtelekom	68	546	300	1K	3	30

Routing Algorithms

- ❑ Shortest Path Routing
- ❑ Uniform
- ❑ Dynamic routing: PGA on L for routes alone

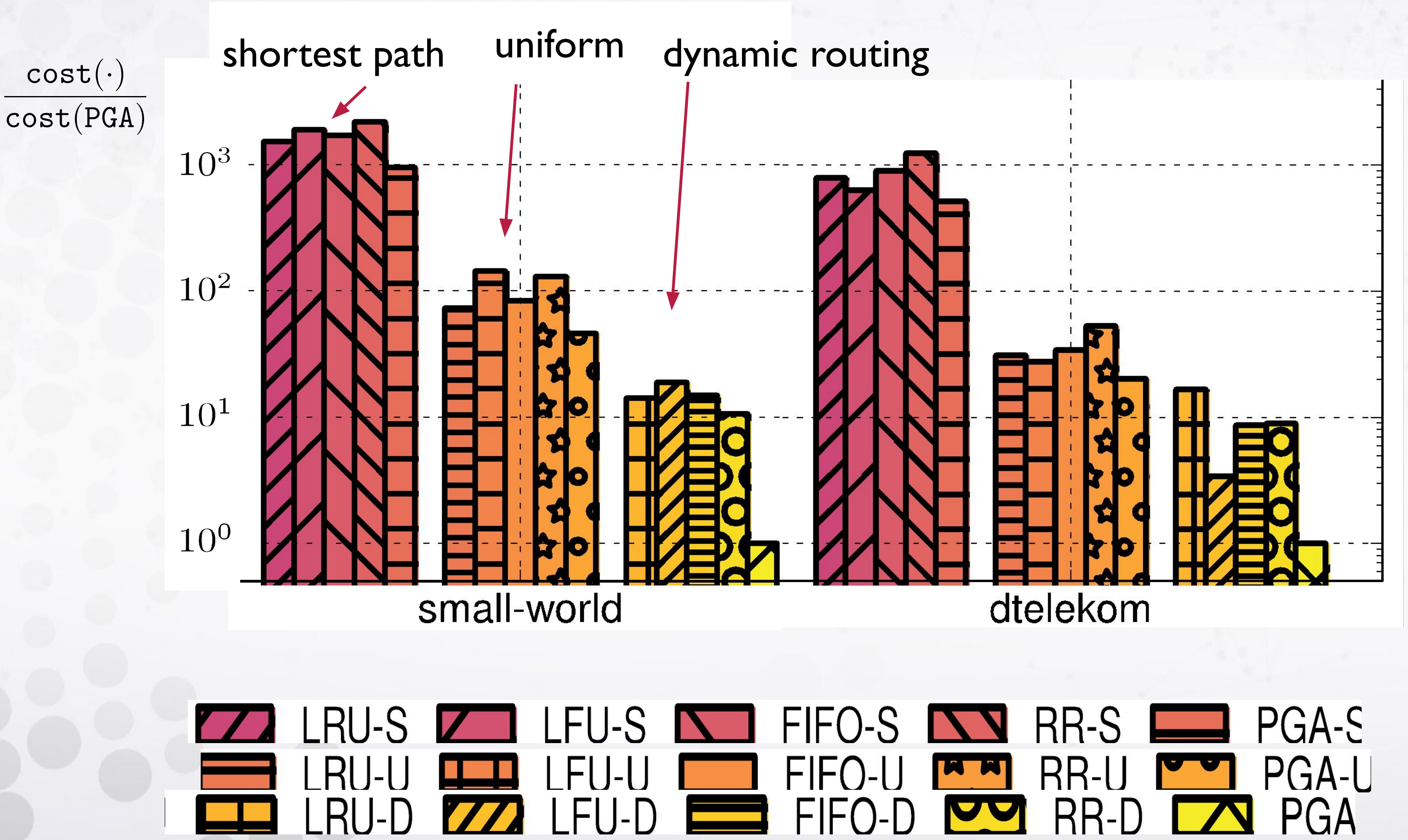
Caching Algorithms

- ❑ LRU
- ❑ LFU
- ❑ FIFO
- ❑ RR
- ❑ PGA on L

Performance Comparison

[I. and Yeh, ICN 2017/JSAC 2018]

Ratio of expected routing cost to routing cost under our algorithm

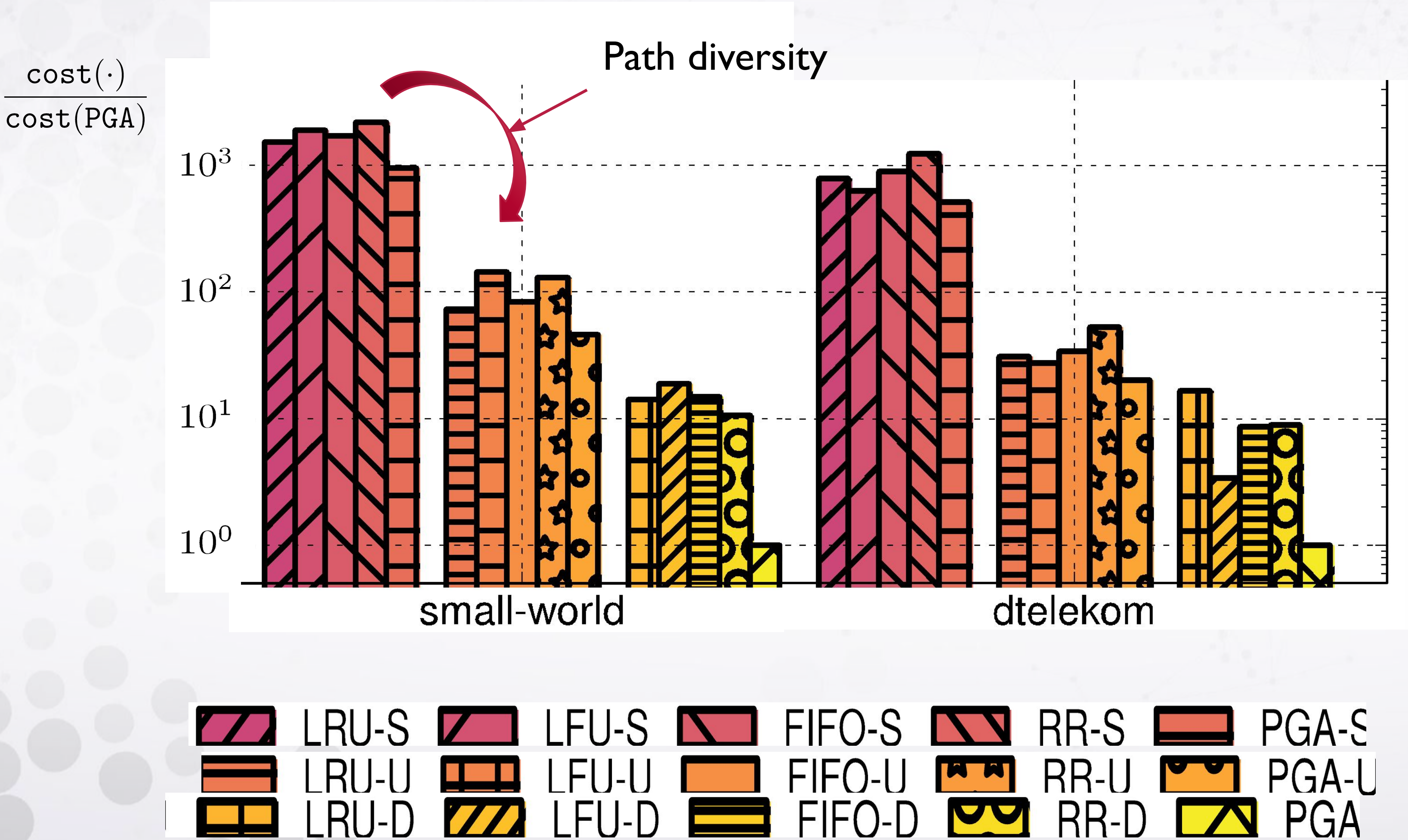


Institute for the Wireless Internet of Things at Northeastern

Performance Comparison

[I. and Yeh, ICN 2017/JSAC 2018]

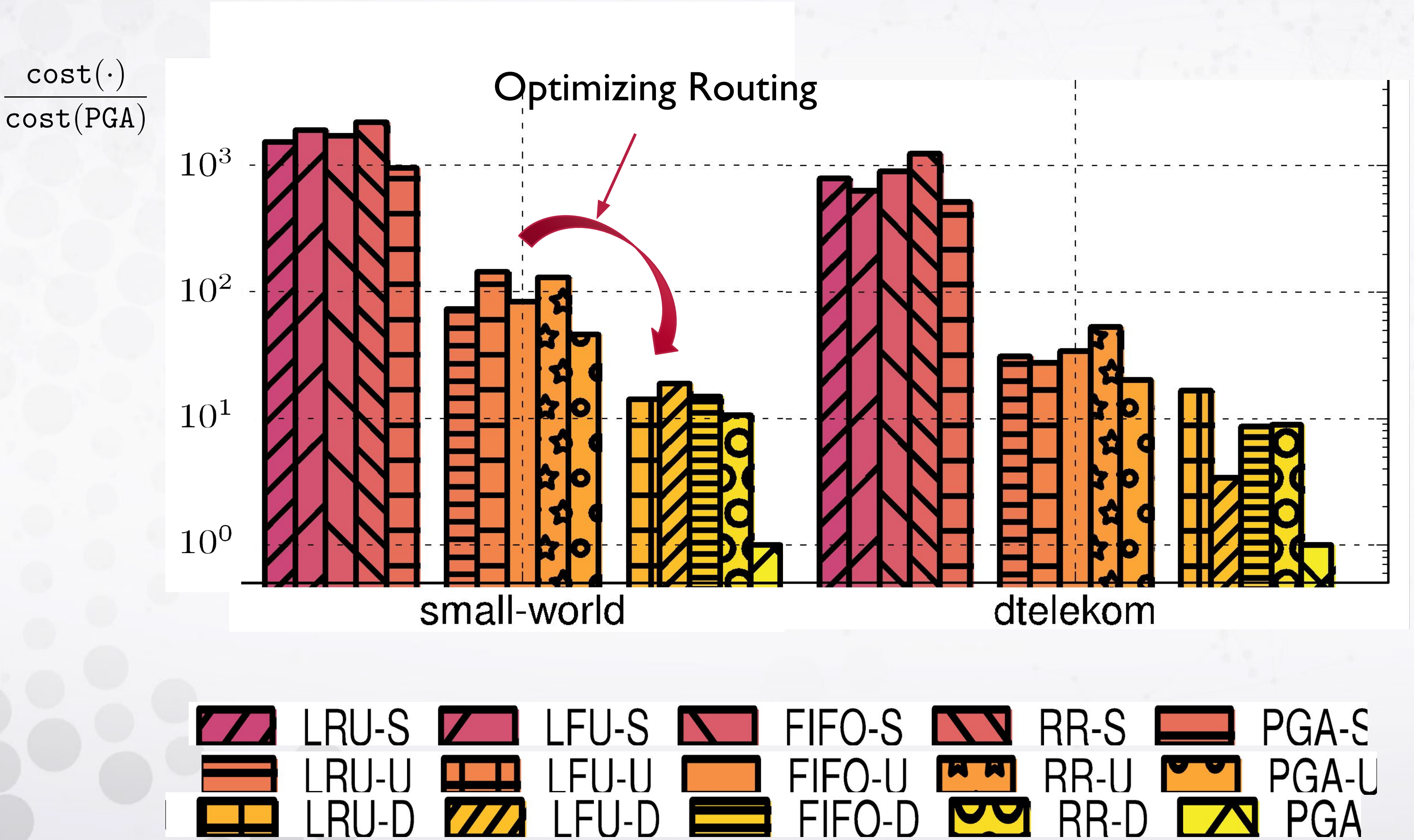
Ratio of expected routing cost to routing cost under our algorithm



Performance Comparison

[I. and Yeh, ICN 2017/JSAC 2018]

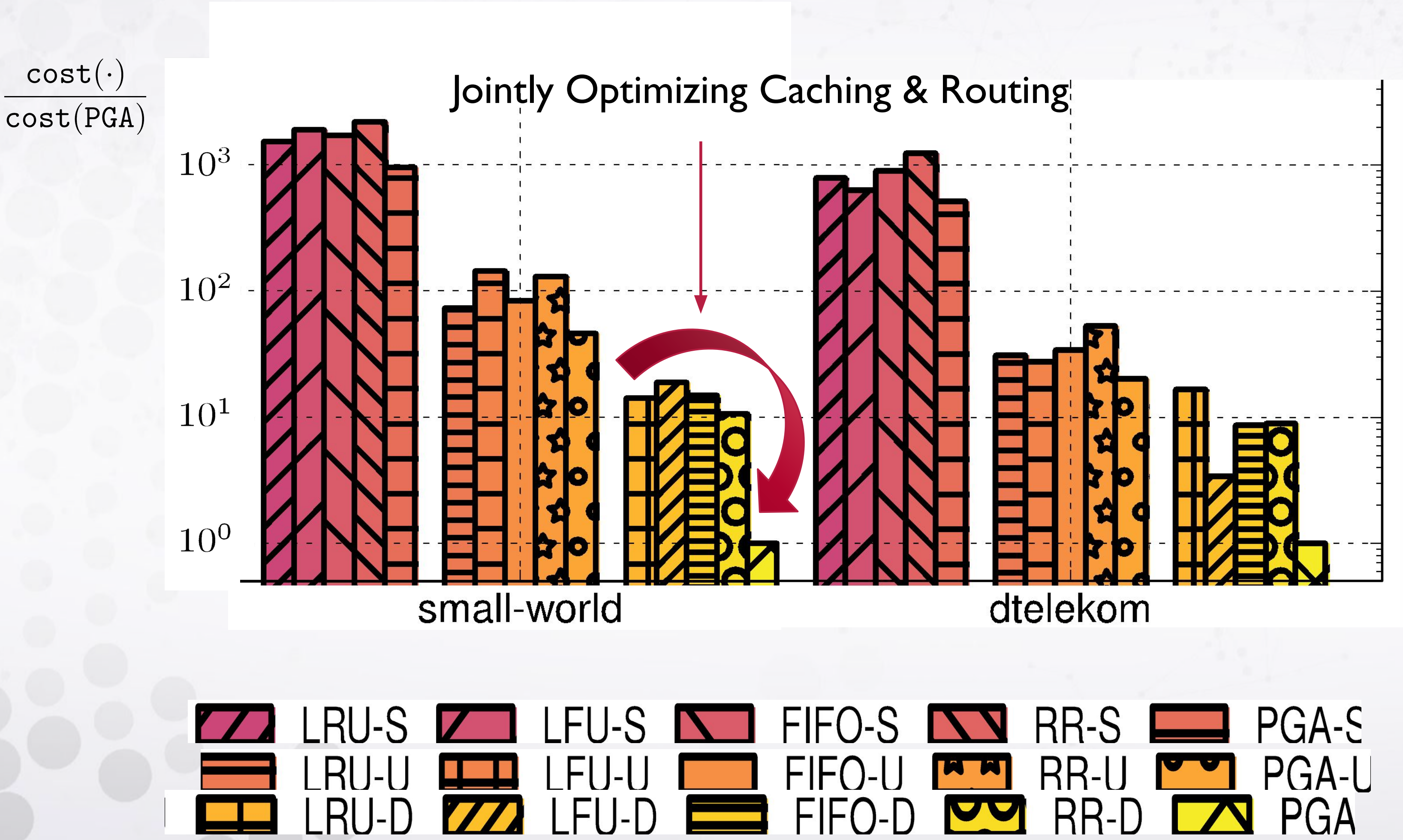
Ratio of expected routing cost to routing cost under our algorithm



Institute for the Wireless Internet of Things at Northeastern

Performance Comparison

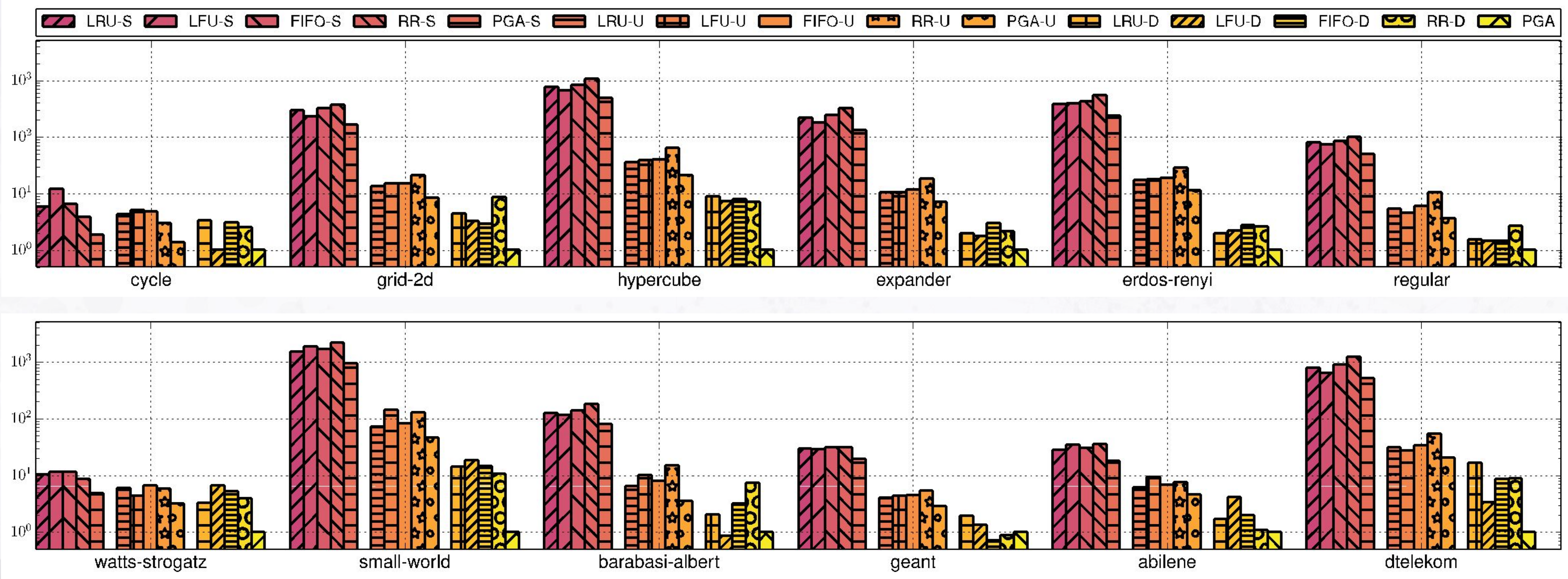
[I. and Yeh, ICN 2017/JSAC 2018]



Institute for the Wireless Internet of Things at Northeastern

Performance Comparison

[I. and Yeh, ICN 2017/JSAC 2018]



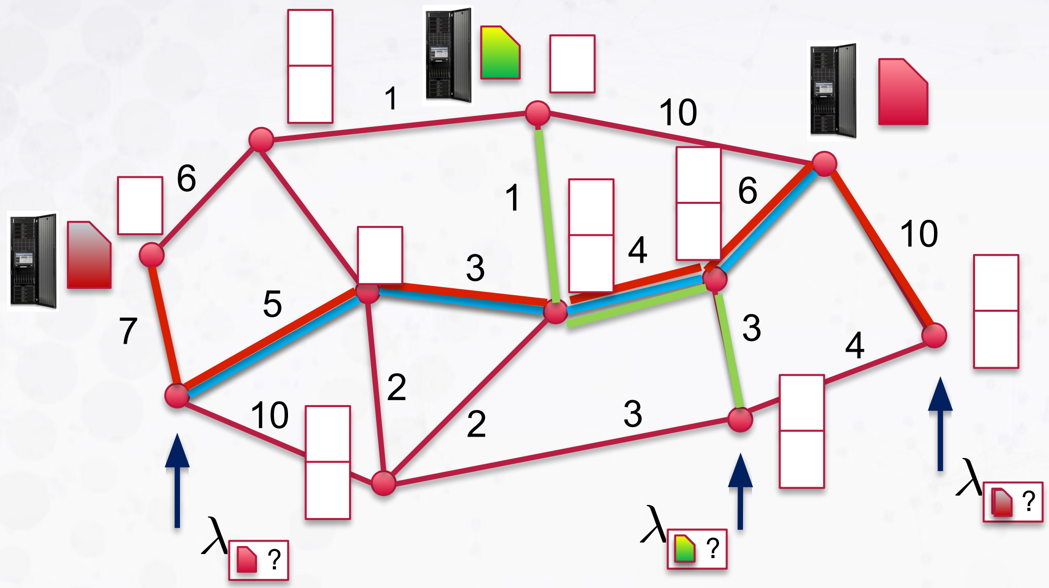
Institute for the Wireless Internet of Things at Northeastern

Overview

- ❑ Cache network optimization
- ❑ Jointly optimizing caching and routing
- ❑ Introducing queues

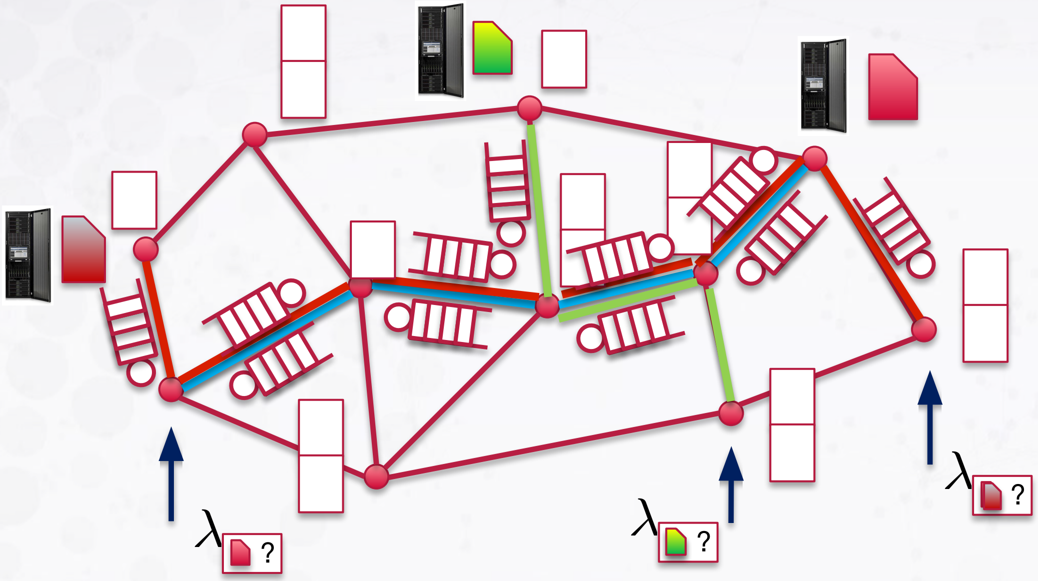
Introducing Queues

[Mahdian, Moharrer, I., and Yeh, INFOCOM 2019/ToN 2020]



Introducing Queues

[Mahdian, Moharrer, I., and Yeh, INFOCOM 2019/ToN 2020]



- Downward edges are associated with **M/M/1 queues**
- Determine **cache contents** so that steady state **queuing costs** are **minimized**
- Size of queue at edge e: $n_e \in \mathbb{N}$
- Cost: $c_e(n_e)$ where $c_e : \mathbb{R}_+ \to \mathbb{R}_+$ is non-decreasing
  - Queue size, its moments, queuing delay, occupancy probability...
- Aggregate expected cost:

$$C(\mathbf{x}, \boldsymbol{\lambda}) \equiv \sum_{e \in E} \mathbb{E}_{\mathbf{x},\boldsymbol{\lambda}}[c(n_e)]$$

- **Caching gain:**

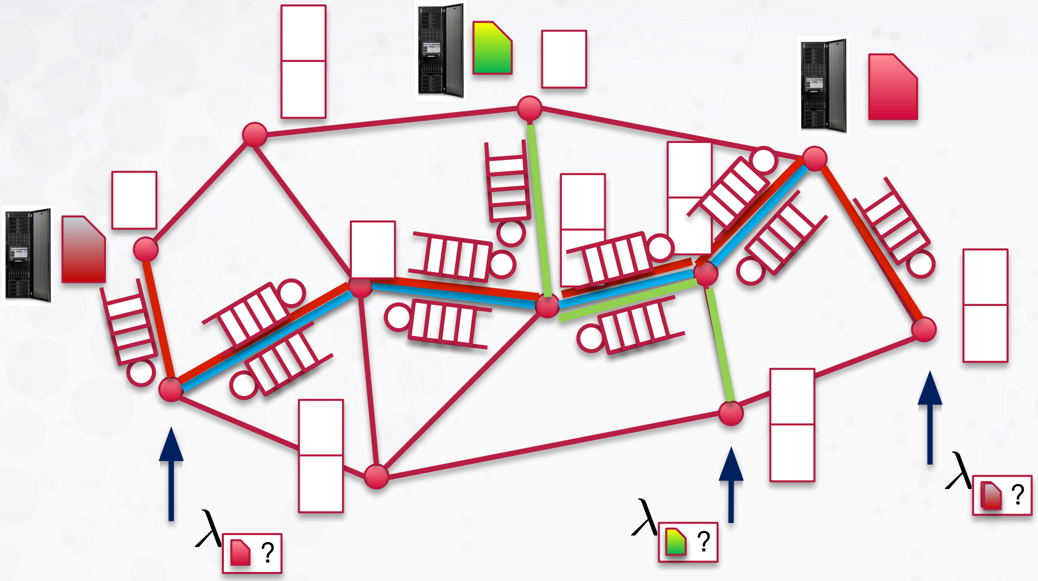
$$F(\mathbf{x}, \boldsymbol{\lambda}) = C(\mathbf{x}_0, \boldsymbol{\lambda}) - C(\mathbf{x}, \boldsymbol{\lambda})$$

caching allocation under which system is stable

Theorem: Maximizing caching gain is a **submodular maximization** problem subject to **matroid constraints**.

Institute for the Wireless Internet of Things at Northeastern

Stability



- Caching gain:

$$F(\mathbf{x}, \boldsymbol{\lambda}) = C\left(\mathbf{x}_0, \boldsymbol{\lambda}\right) - C(\mathbf{x}, \boldsymbol{\lambda})$$

caching allocation under
which system is stable

- **How does one find this?**

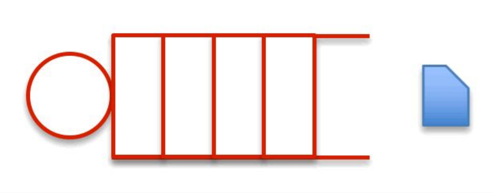
- Optimize caching strategy $(\mathbf{x})$ **and jointly do admission control $(\boldsymbol{\lambda})$** subject to stability constraints.
  - Much **weaker** optimality guarantees.

[Kamran, Moharrer, I., and Yeh, INFOCOM 2021]

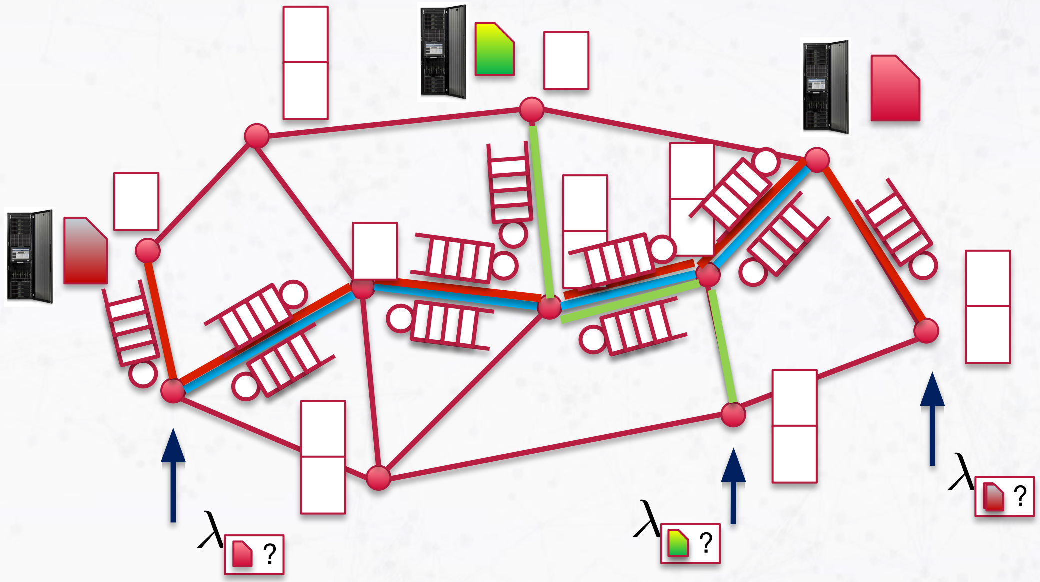
A More Elegant Solution: Counting Queues

[Li and I., INFOCOM 2020/ToN 2021]

Catalog $\mathcal{C}=\{$ ▮ ▮ ▮ $\}$ is finite!

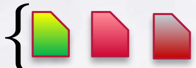


M/M/1 queue

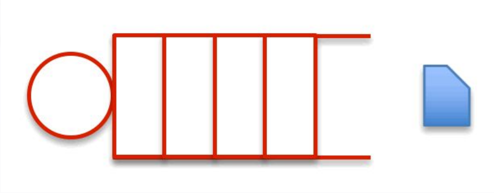


A More Elegant Solution: Counting Queues

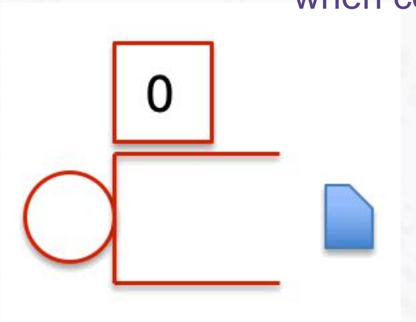
[Li and I., INFOCOM 2020/ToN 2021]

$\mathcal{C}=\{$  $\}$

Identical responses merge when collocated



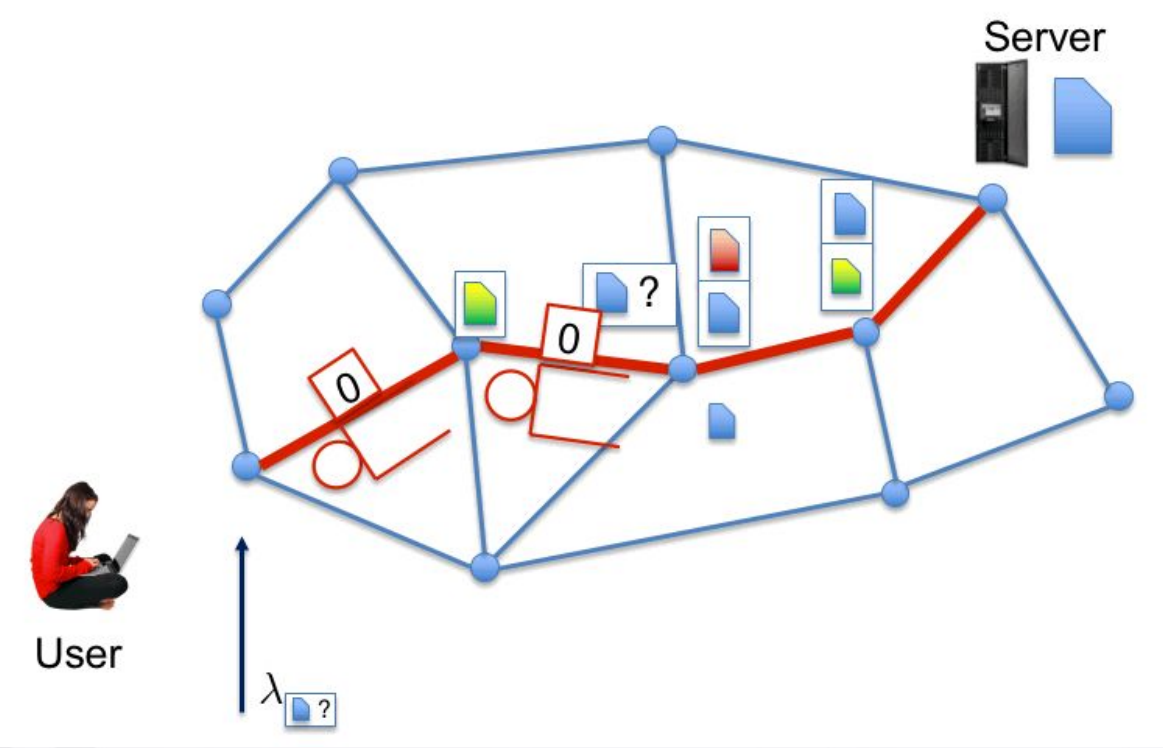
M/M/1 queue



M/M/1c queue

A More Elegant Solution: Counting Queues

[Li and I., INFOCOM 2020/ToN 2021]



- Network with **counting queues**
- Not reversible**, steady-state** queue distribution has **no closed form**
- **Well-approximated** by **M/M/∞** queues
- **Theorem:** Under this approximation, there exists an algorithm **jointly** optimizing of **caching and service rate** allocations within 1-1/e of the optimal.

Institute for the Wireless Internet of Things at Northeastern

Open Directions

- ❑ No-regret algorithms
- ❑ Merging **requests/queries**, not responses
- ❑ Joint optimization tasks
  - ❑ Caching
  - ❑ Routing
  - ❑ Service assignment
  - ❑ Admission control
  - ❑ …

→ Departure from submodularity
→ Distributed algorithms

Institute for the Wireless Internet of Things at Northeastern

Institute for the Wireless Internet of Things
at Northeastern University

EDGE Institute

Adaptive Caching Networks with Optimality Guarantees
S. Ioannidis and E. Yeh, SIGMETRICS 2016/ToN 2018.

Jointly Optimal Routing and Caching for Arbitrary Network Topologies
S. Ioannidis and E. Yeh, ICN 2017/JSAC 2018.

Kelly Cache Networks
M. Mahdian, A. Moharrer, S. Ioannidis, and E. Yeh, INFOCOM 2019/ToN 2020.

Cache Networks with Counting Queues,
Y. Li and S. Ioannidis, INFOCOM 2020/ToN 2021.

Online Caching Networks with Adversarial Guarantees
Y. Li, T. Si Salem, G. Neglia, and S. Ioannidis, SIGMETRICS/PERFORMANCE 2022.

Thank You!

CNS-1718355
CNS-2107062
CNS-2112471