Towards Demand-Aware Peer Selection With XOR-based Routing

Qingyun Ji, Darya Melnyk, **Arash Pourdamghani**, Stefan Schmid

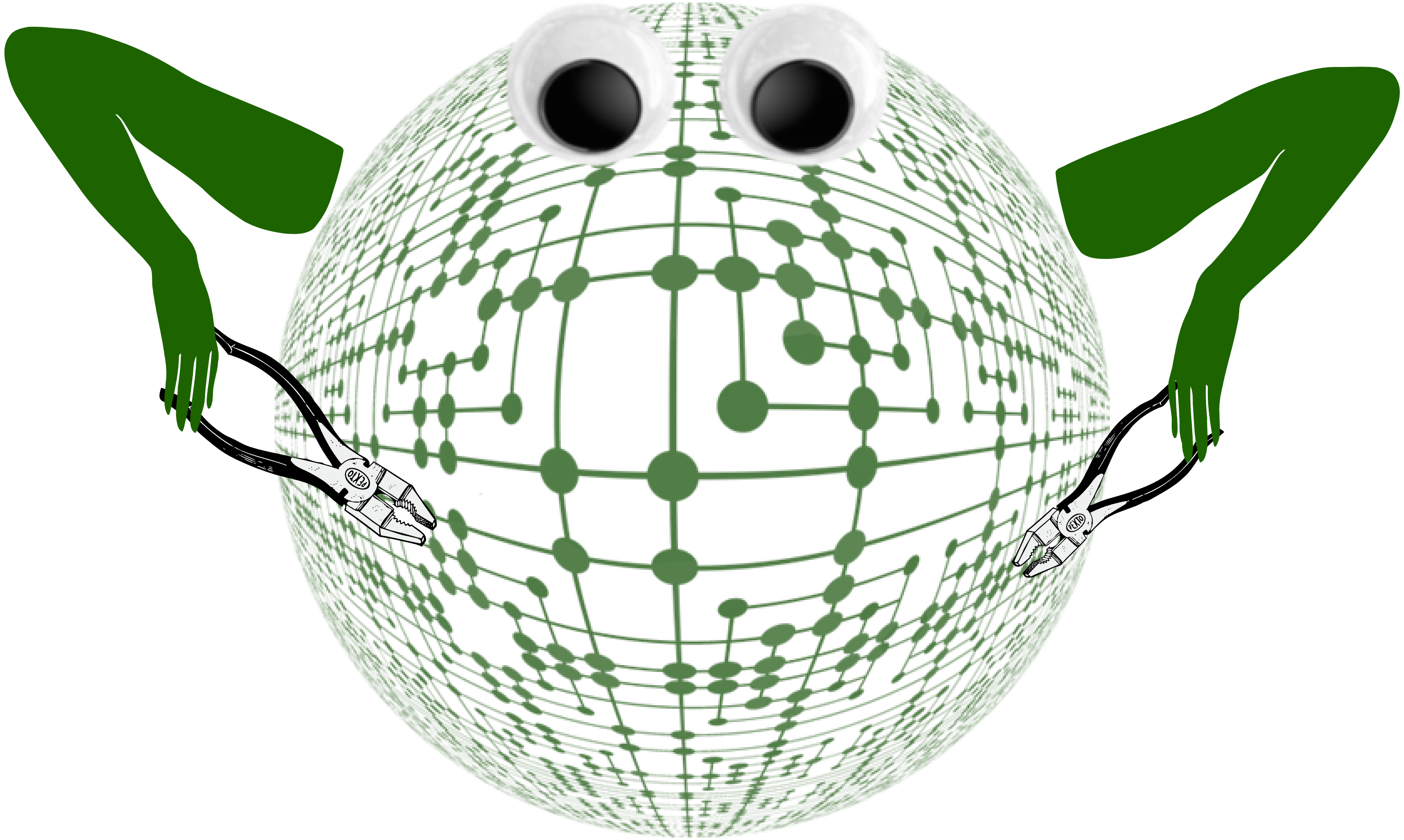
Invited Paper, SSS’25

Towards Demand-Aware Peer Selection With XOR-based Routing

(1) (2) (3)

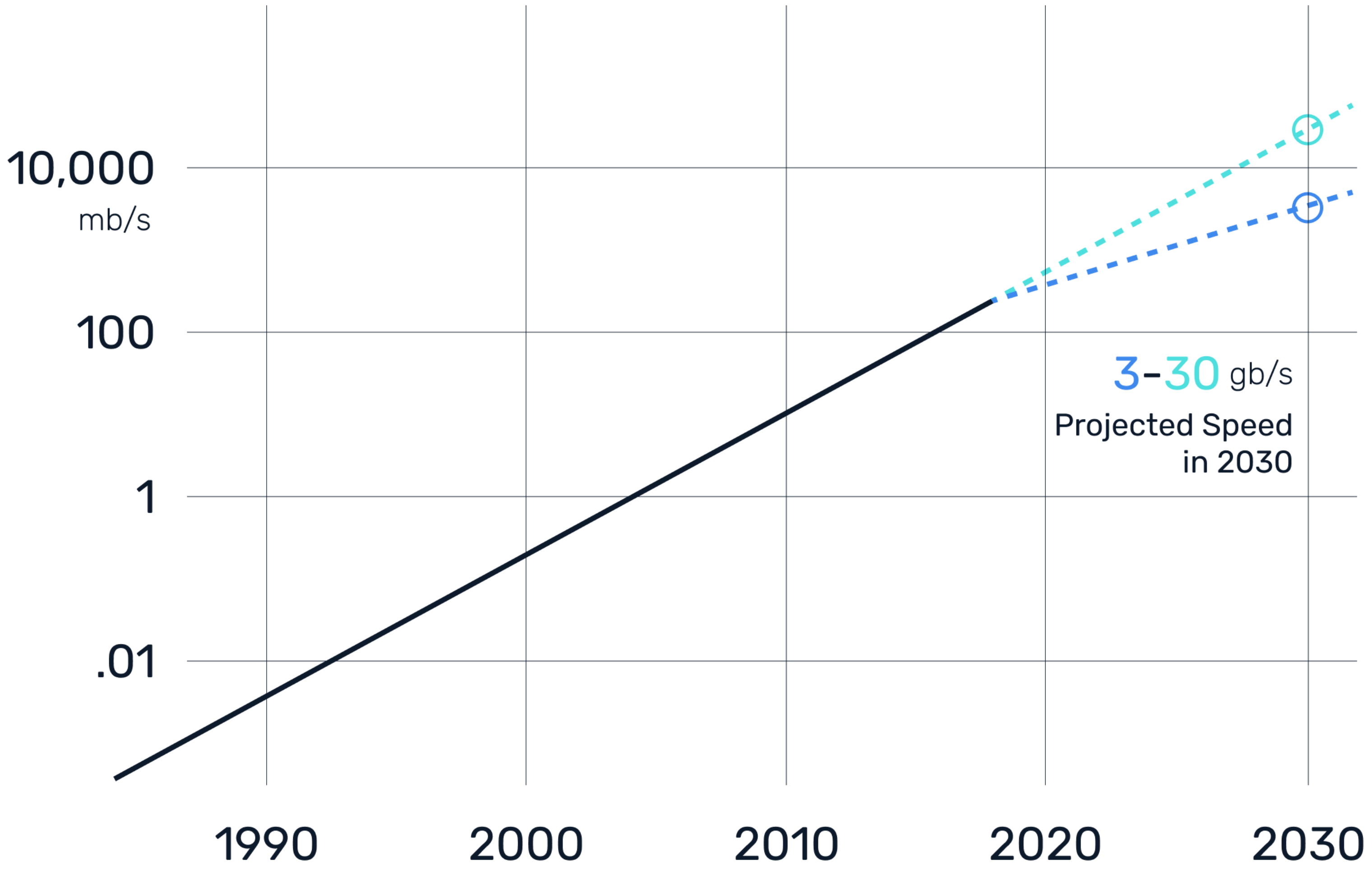
Qingyun Ji, Darya Melnyk, **Arash Pourdamghani**, Stefan Schmid

Invited Paper, SSS’25



An Empirical Step Towards Optimal Demand-aware Networks

Nielsen's Law of User Internet Bandwidth

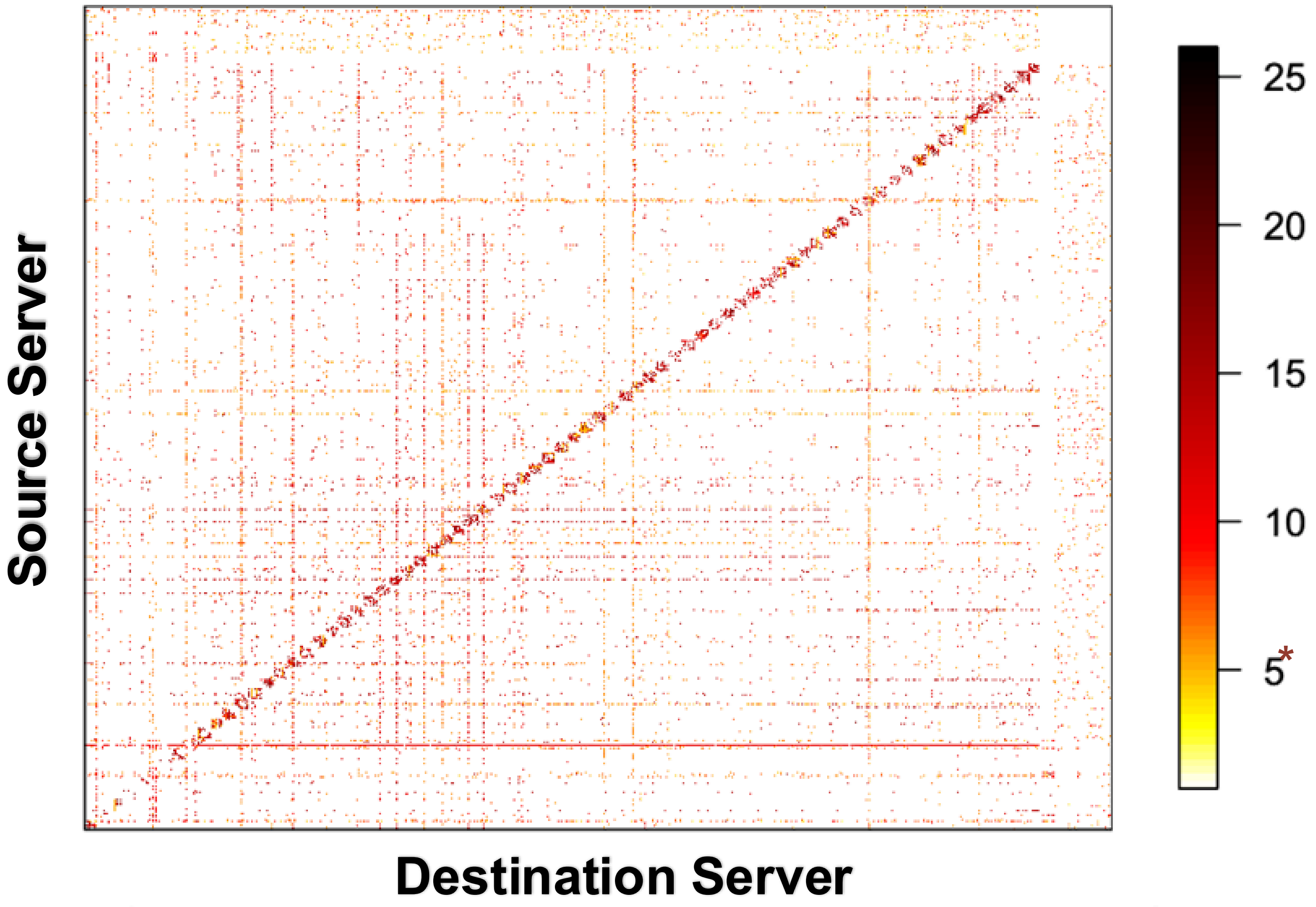


…while we are reaching the end of Moore's law*!

*by most forecasters, including Gordon Moore himself

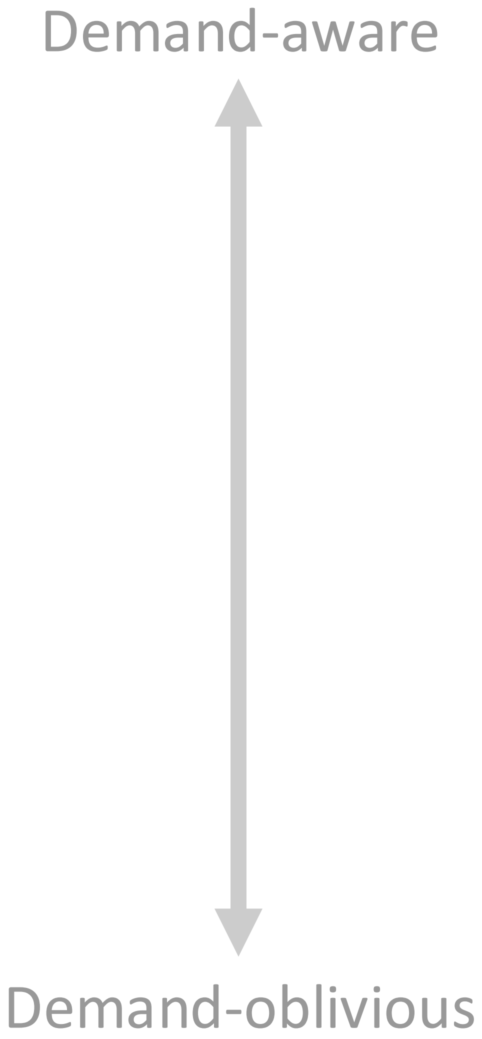
Source of image: Nielsen Norman Group/broadband.money

Structure in The Demand!



*log (Bytes) exchanged between server pairs in a 10s period

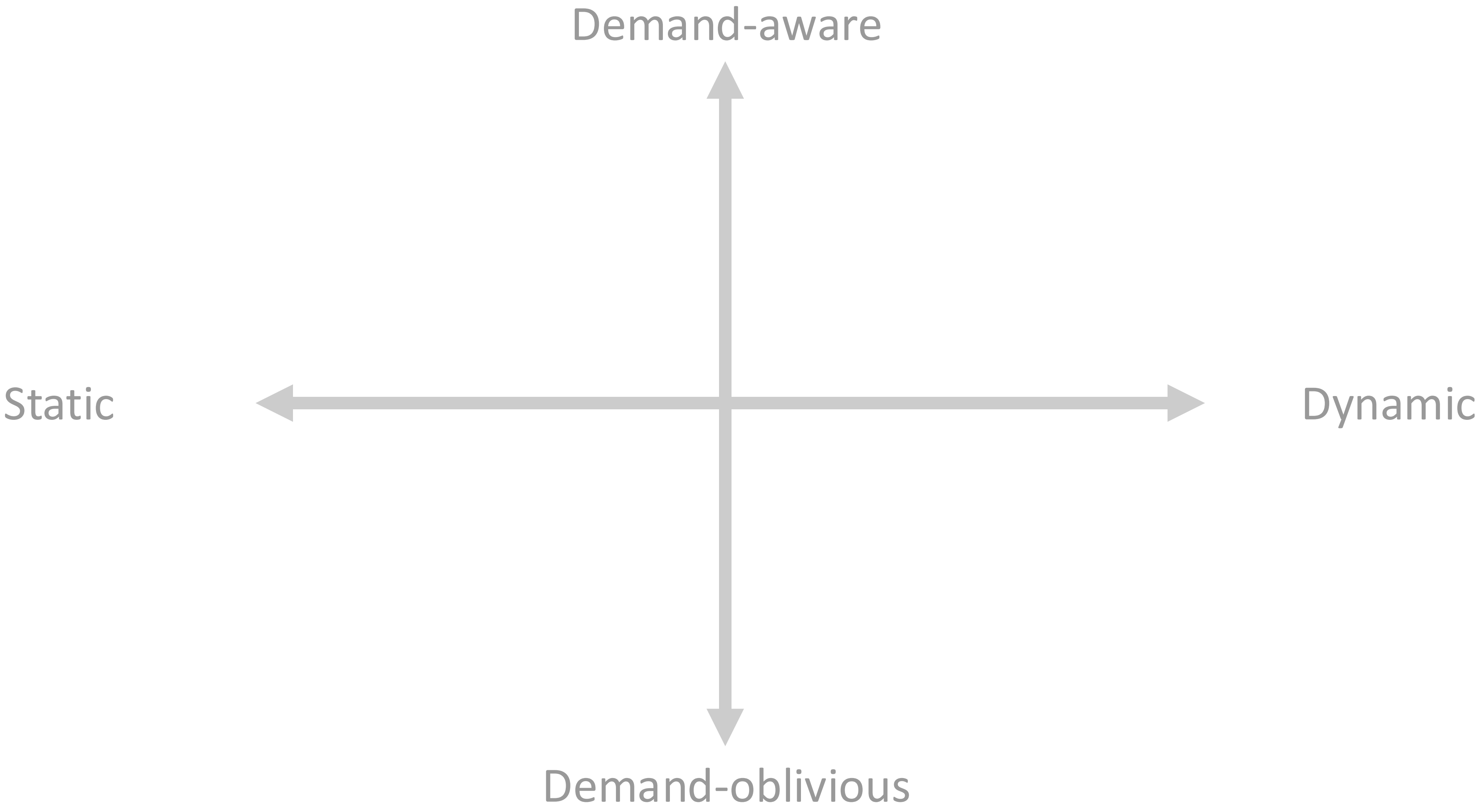
Zooming Out



Demand-aware

Demand-oblivious

Zooming Out



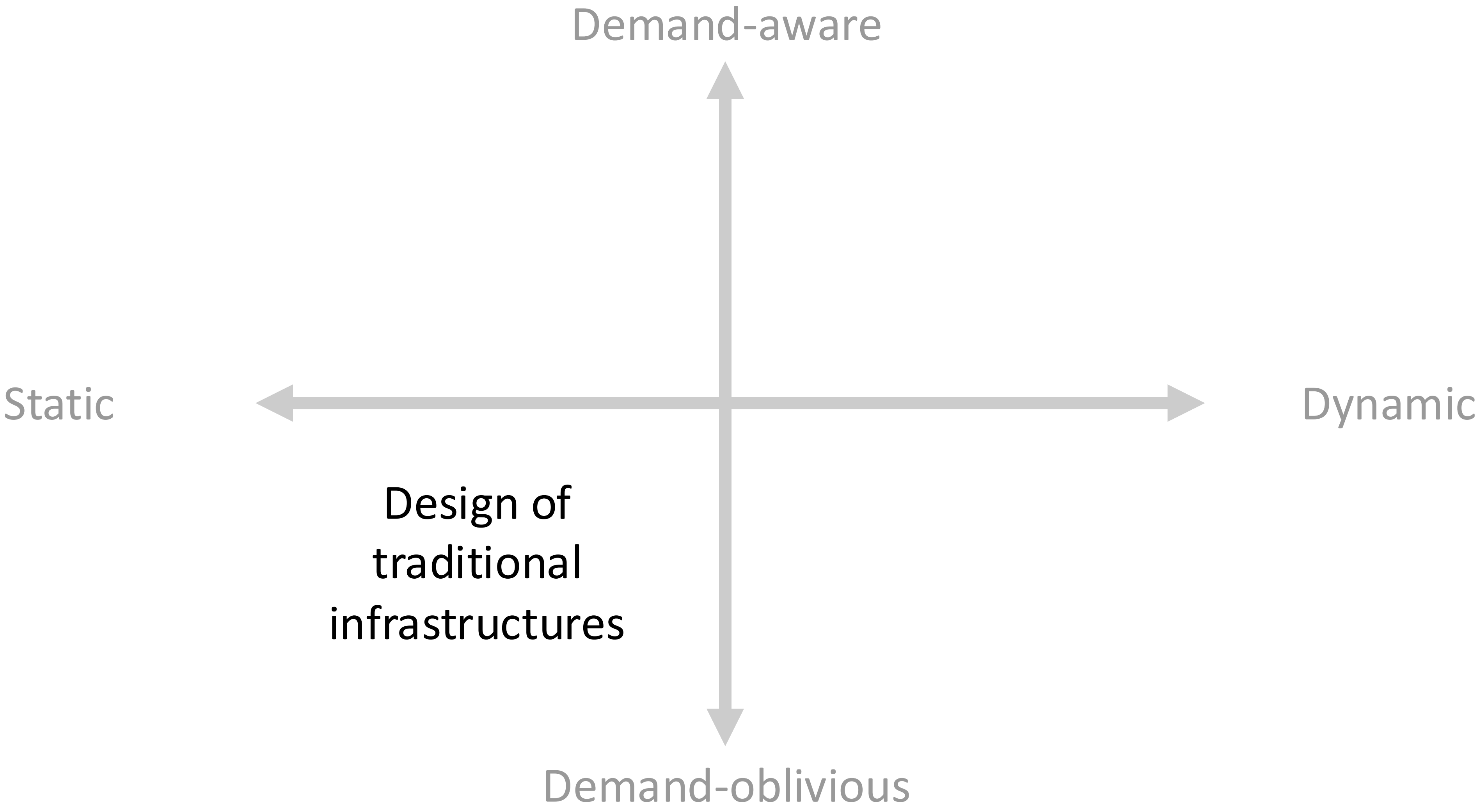
Demand-aware

Static

Dynamic

Demand-oblivious

Zooming Out



Demand-aware

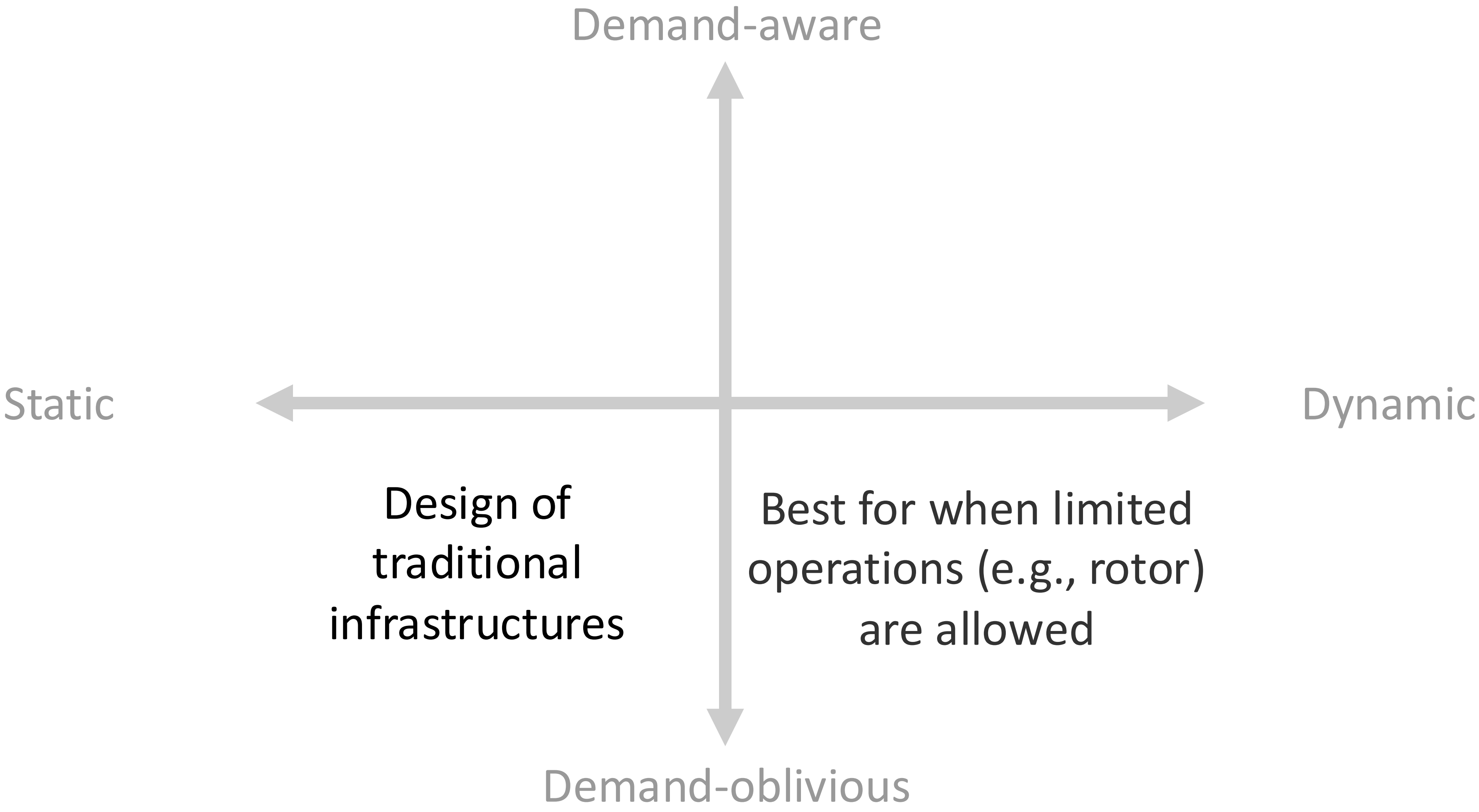
Static

Dynamic

Design of traditional infrastructures

Demand-oblivious

Zooming Out



Demand-aware

Static

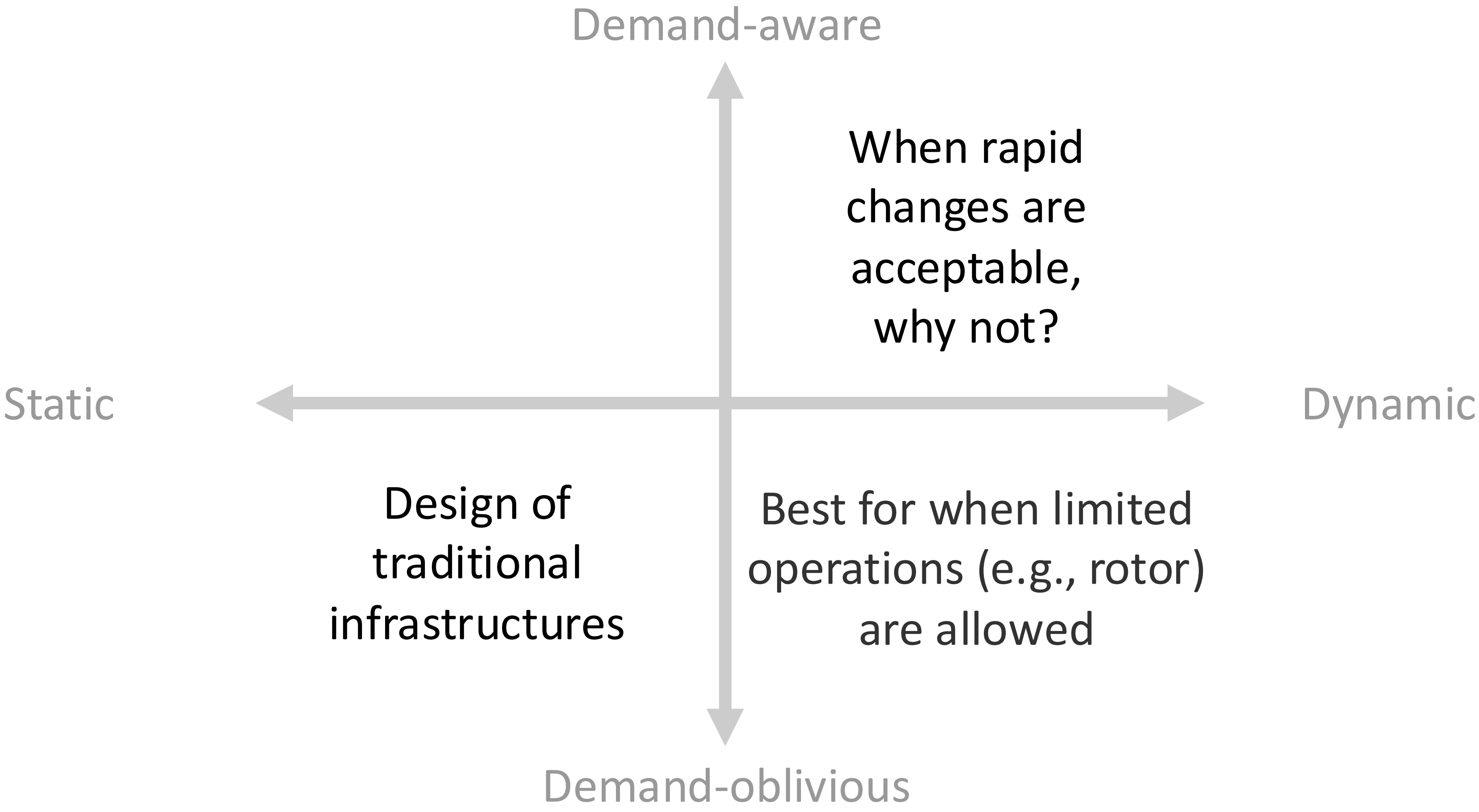
Dynamic

Design of traditional infrastructures

Best for when limited operations (e.g., rotor) are allowed

Demand-oblivious

Zooming Out



Demand-aware

When rapid changes are acceptable, why not?

Static

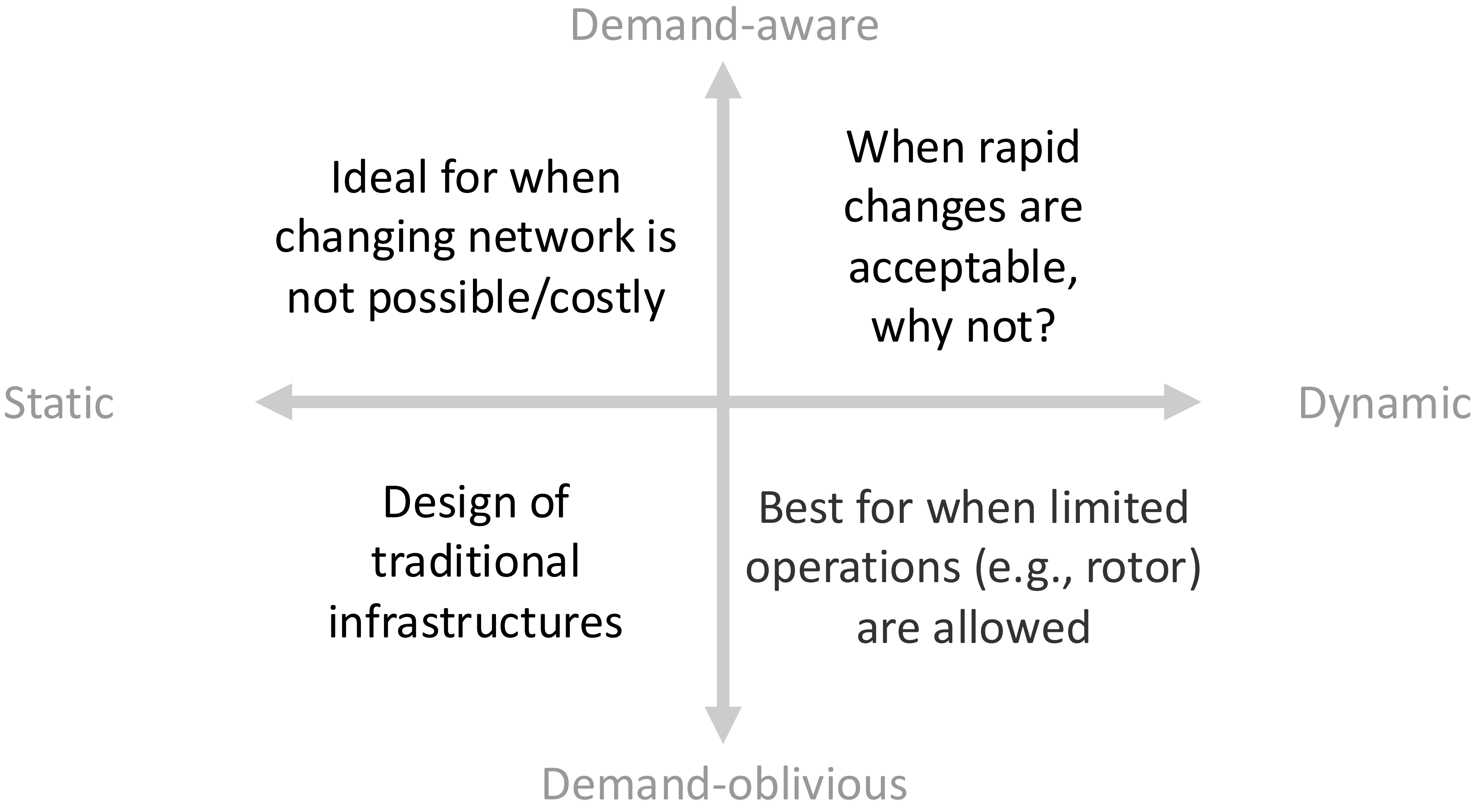
Dynamic

Design of traditional infrastructures

Best for when limited operations (e.g., rotor) are allowed

Demand-oblivious

Zooming Out



Demand-aware

Ideal for when changing network is not possible/costly

When rapid changes are acceptable, why not?

Static

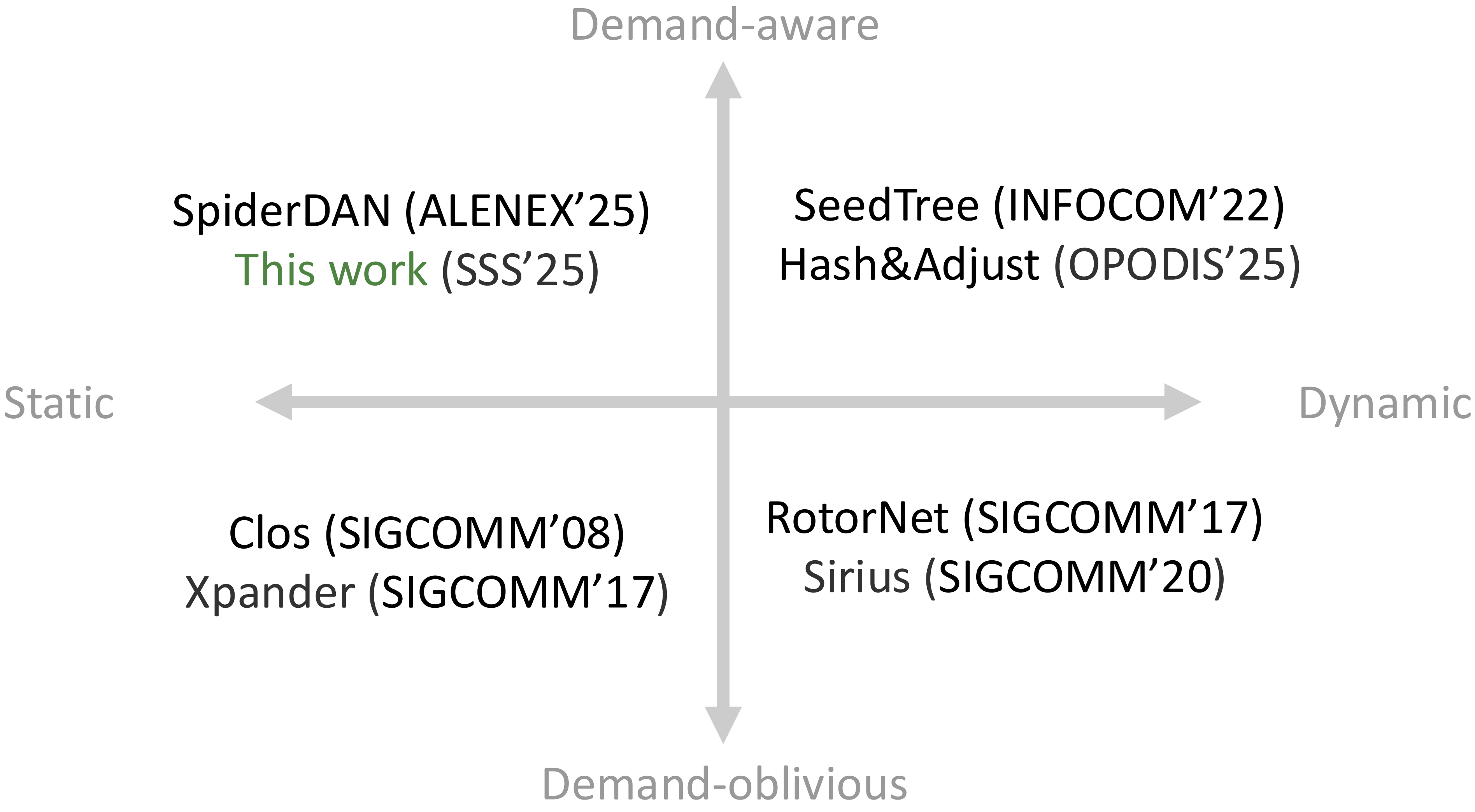
Dynamic

Design of traditional infrastructures

Best for when limited operations (e.g., rotor) are allowed

Demand-oblivious

Zooming Out: Some Prior Works in Networking



Demand-aware

SpiderDAN (ALENEX'25)
This work (SSS'25)

SeedTree (INFOCOM'22)
Hash&Adjust (OPODIS'25)

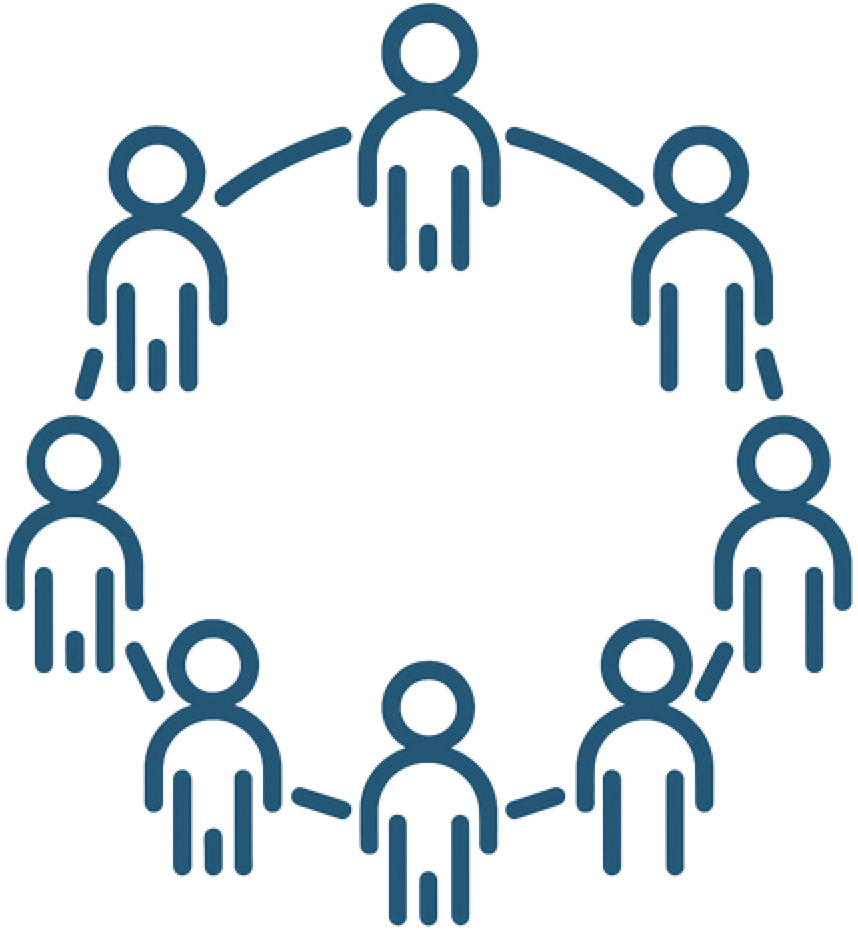
Static

Dynamic

Clos (SIGCOMM'08)
Xpander (SIGCOMM'17)

RotorNet (SIGCOMM'17)
Sirius (SIGCOMM'20)

Demand-oblivious



A Theoretical View At

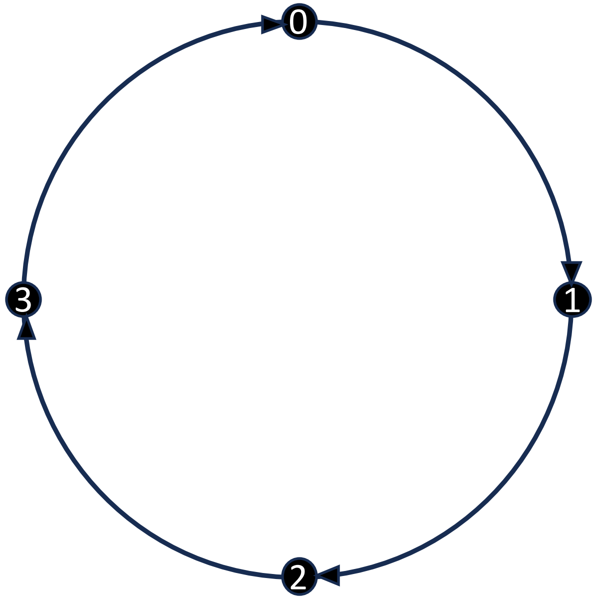
Peer Selection Algorithms

Model

Input:

$\mathcal{D} =$

0	0.15	0.2	0
0.05	0	0.05	0.1
0	0.05	0	0.1
0.1	0.05	0.25	0

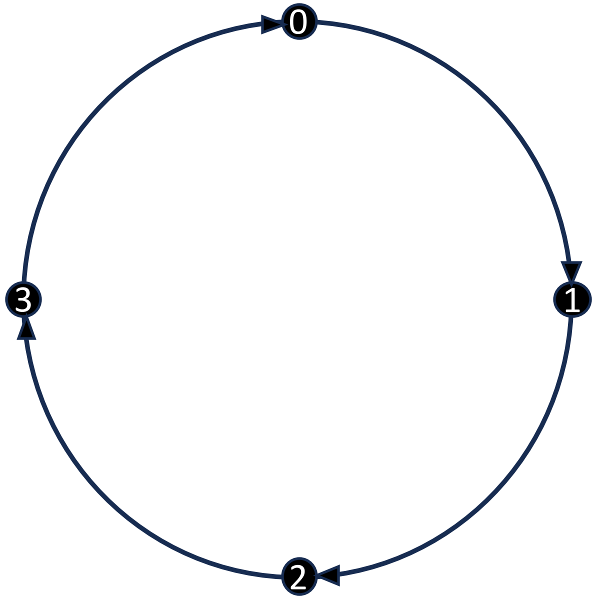


Model

Input:

$\mathcal{D}$ =

0	0.15	0.2	0
0.05	0	0.05	0.1
0	0.05	0	0.1
0.1	0.05	0.25	0



Constraints:

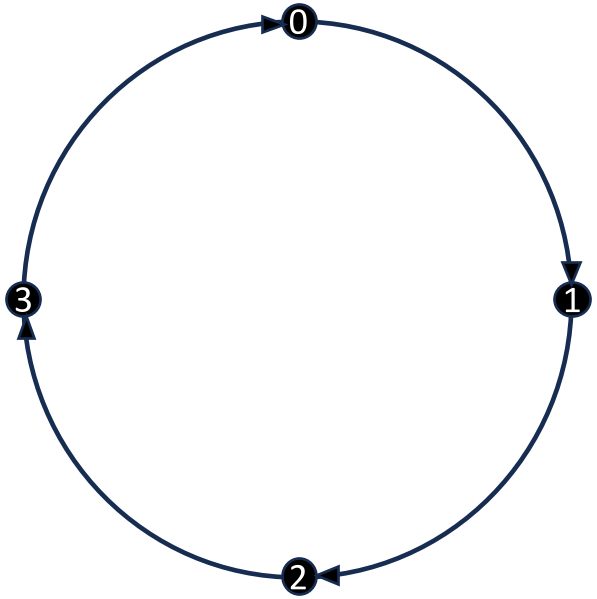
- $\log n$ degree

Model

Input:

$\mathcal{D} =$

0	0.15	0.2	0
0.05	0	0.05	0.1
0	0.05	0	0.1
0.1	0.05	0.25	0



Constraints:

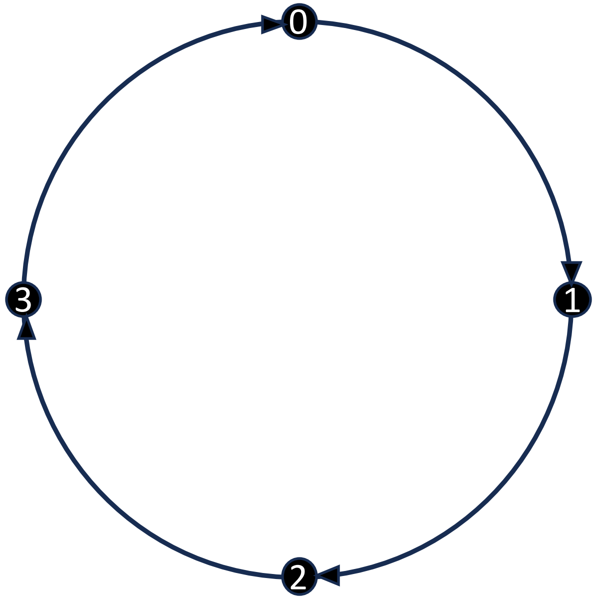
- $\log n$ degree
- Constructed locally

Model

Input:

$\mathcal{D} =$

0	0.15	0.2	0
0.05	0	0.05	0.1
0	0.05	0	0.1
0.1	0.05	0.25	0



Constraints:

- $\log n$ degree
- Constructed locally

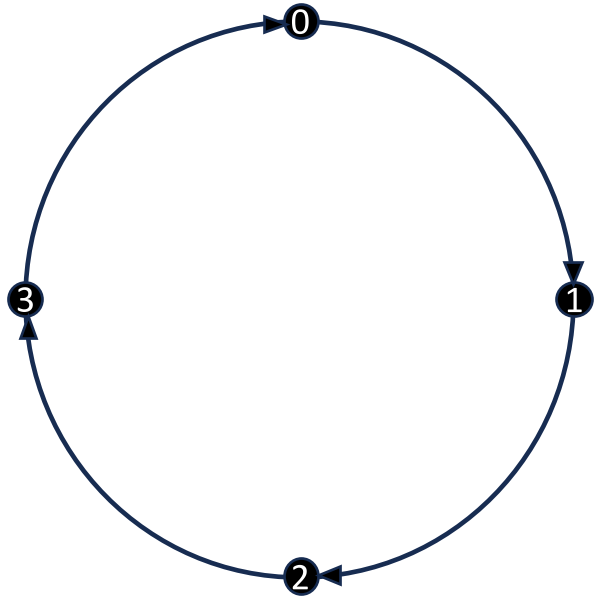
Objective:

$$\min_{E} \sum_{i,j\in\mathcal{V}} \mathit{path\ length}_{ij} \times \mathcal{D}_{ij}$$

Model

Input:

$$\mathcal{D} = \begin{bmatrix} 0 & 0.15 & 0.2 & 0 \\ 0.05 & 0 & 0.05 & 0.1 \\ 0 & 0.05 & 0 & 0.1 \\ 0.1 & 0.05 & 0.25 & 0 \end{bmatrix}$$



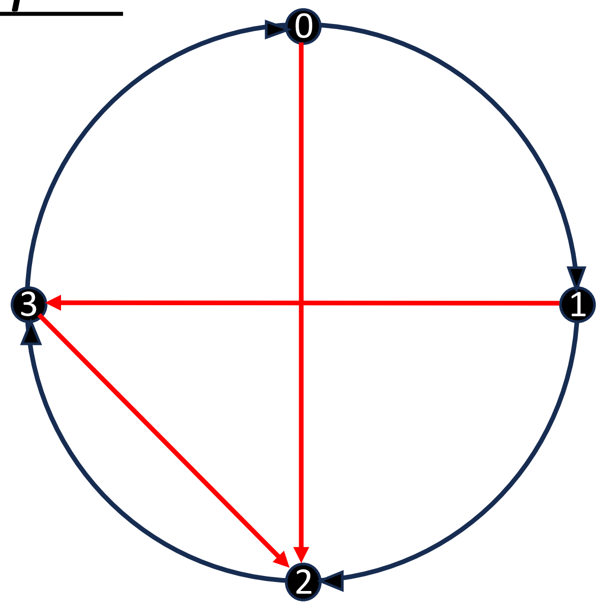
Constraints:

- $\log n$ degree
- Constructed locally

Objective:

$$\min_{E} \sum_{i,j \in \mathcal{V}} path\ length_{ij} \times \mathcal{D}_{ij}$$

Example:



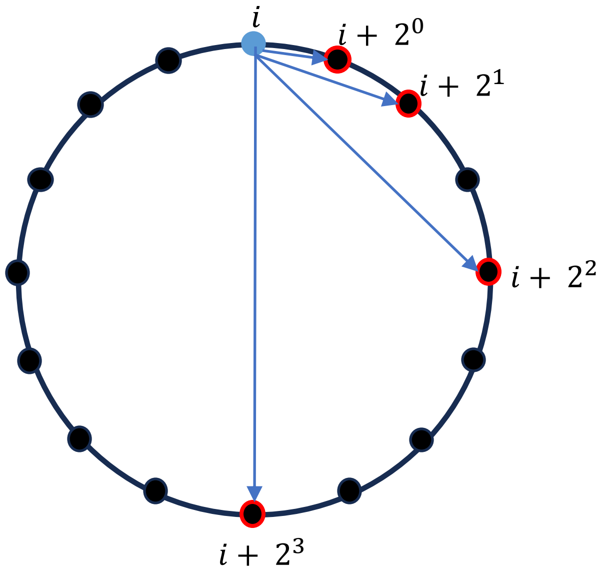
$$\begin{bmatrix} 0 & 1 & 1 & 2 \\ 2 & 0 & 1 & 1 \\ 2 & 3 & 0 & 1 \\ 1 & 2 & 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 0.15 & 0.2 & 0 \\ 0.05 & 0 & 0.05 & 0.1 \\ 0 & 0.05 & 0 & 0.1 \\ 0.1 & 0.05 & 0.25 & 0 \end{bmatrix}$$

$$= \quad 0.35 + 0.25 + 0.25 + 0.45 = 1.3$$

Selected Prior Peer-selection algorithms

Chord

[Stoica et al., 2003]

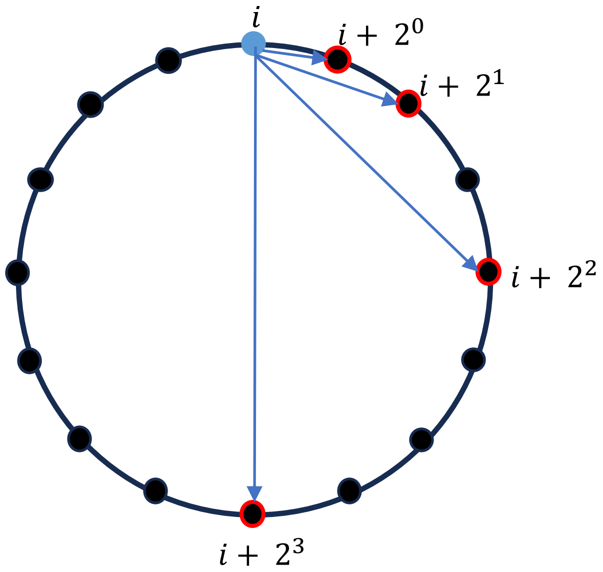


Blind to demand

Selected Prior Peer-selection algorithms

Chord

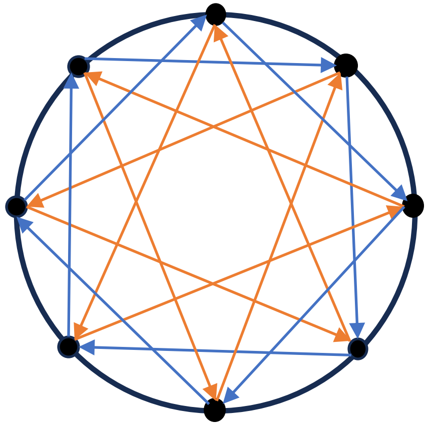
[Stoica et al., 2003]



Blind to demand

Permutations

[W. Wang et al., 2023]



“+2” permutation

“+5” permutation

Coin-change routing

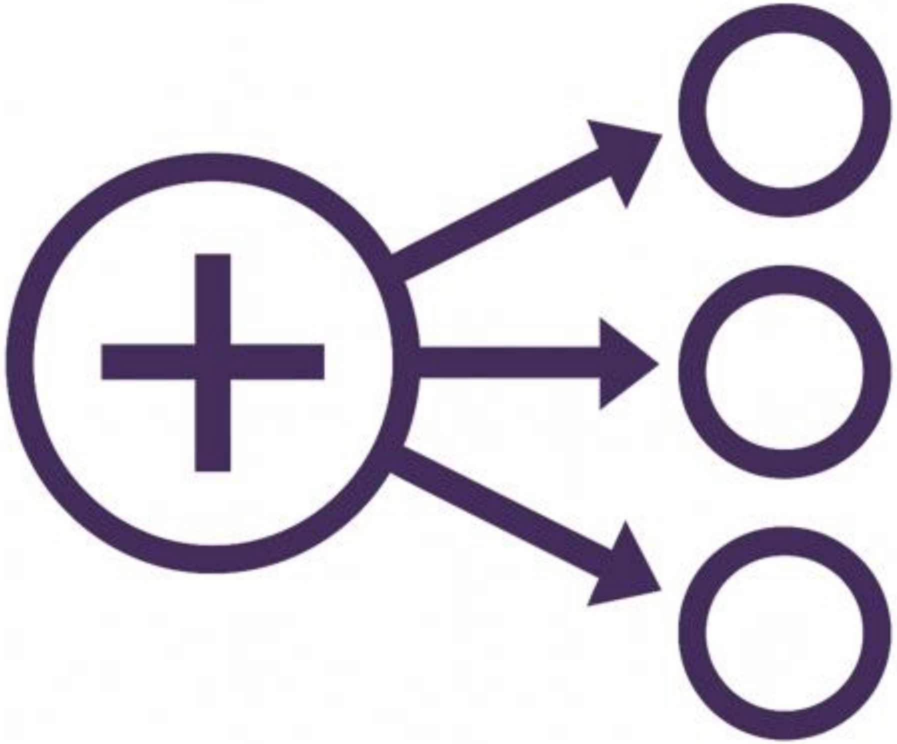
Other relevant related work

Other Peer-selection algorithms:

- Kademlia: randomized peer selection [Maymounkov, Mazieres, 2002]
- Continuous-discrete approach [Naor, Wieder 2003]

Network augmentation for minimizing average shortest path length:

- Small world phenomenon [Kleinberg, STOC 2000] and [Watts and Strogatz, 1998]
- NP-hardness and approximation for adding fixed number of edges [Meyerson and Tagiku, 2009]



Supporting

XOR-Based Routing

Why XOR-Based Routing?

We need a routing mechanism that would be:

- Simple: does not require complex computation
- Local: only depends on information of neighbors
- Greedy: each step bring you closer to destination

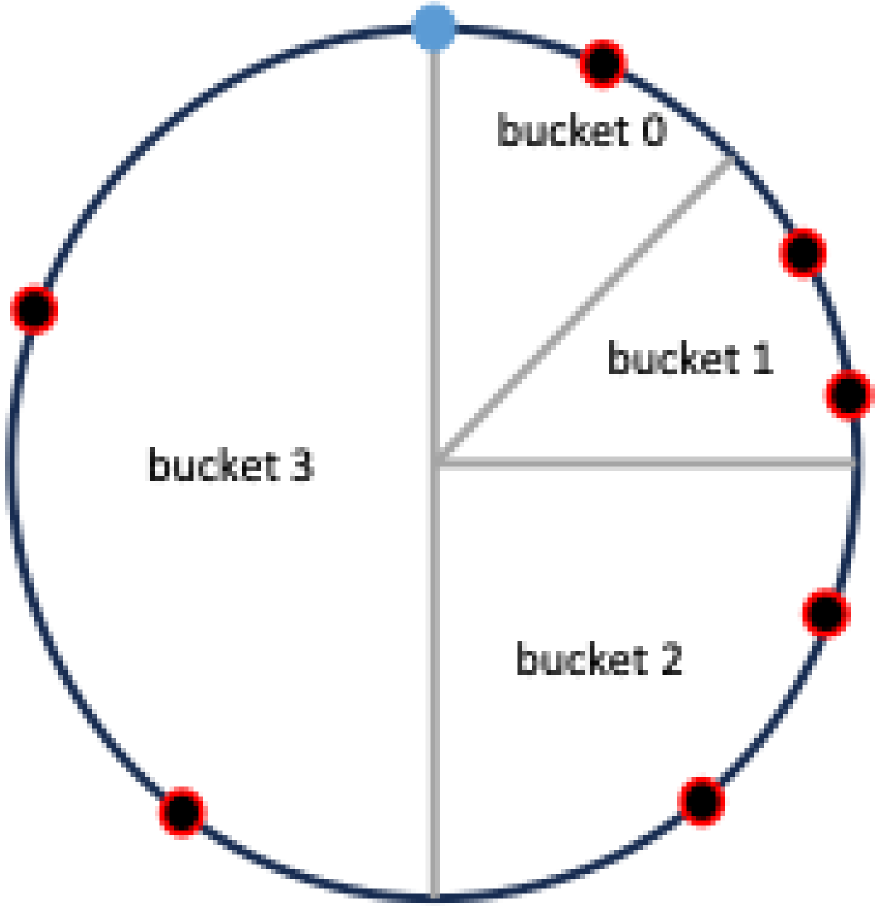
What is XOR-Based Routing?

To route from a source to a destination:

1. Take XOR of the current node and the destination
2. Find the left most “1”, call its index i
3. Go to the node that is different in i-th bit compared to the current ID
4. Go to step 1 and repeat until reaching destination

Our peer selection algorithm supports XOR-based routing by design.

Our Algorithms: Binary Search in Buckets



Our Algorithms: Binary Search in Buckets

Demand from node $0(000)'s$ perspective:

i	0 (src)	1	2	3	4	5	6	7
$demand_{0,i}$	0	0.15	0.05	0.1	0.05	0.15	0.2	0.3

Our Algorithms: Binary Search in Buckets

Demand from node $0(000)'s$ perspective:

i	0 (src)	1	2	3	4	5	6	7
$demand_{0,i}$	0	0.15	0.05	0.1	0.05	0.15	0.2	0.3

demand split on the bucket level

i	1
$demand_{0,i}$	0.15

bucket 2
(001)

i	2	3
$demand_{0,i}$	0.05	0.1

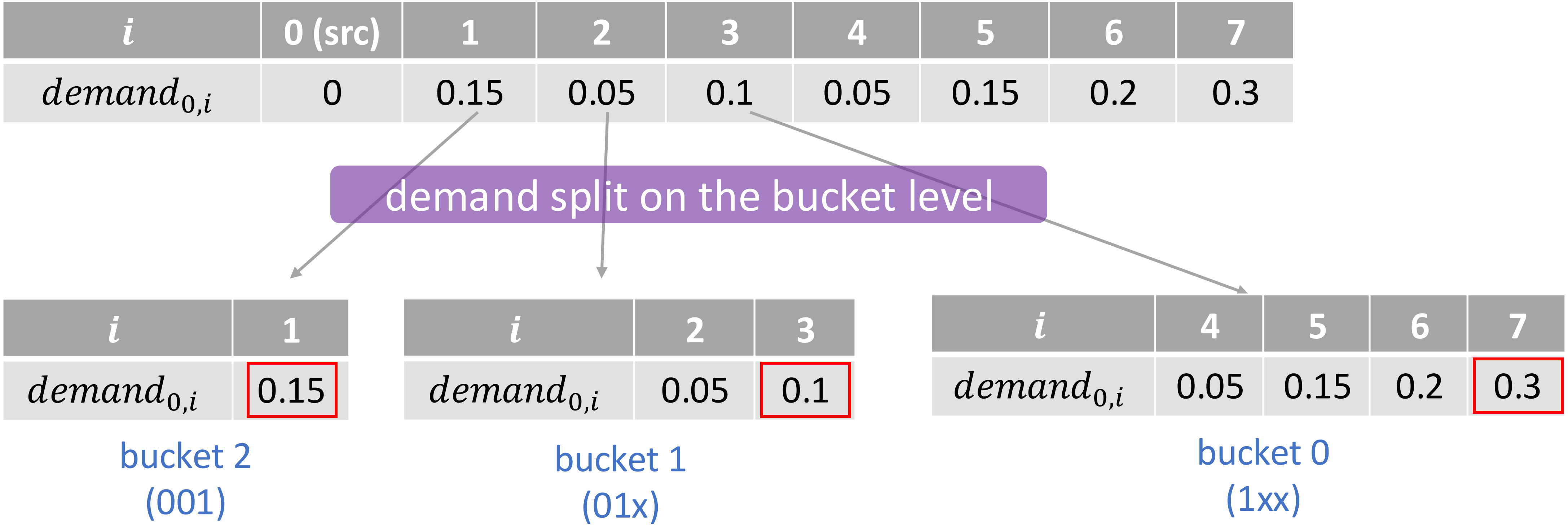
bucket 1
(01x)

i	4	5	6	7
$demand_{0,i}$	0.05	0.15	0.2	0.3

bucket 0
(1xx)

Our Algorithms: Binary Search in Buckets (BSB) - Max Demand

Demand from node $0(000)'s$ perspective:



i	0 (src)	1	2	3	4	5	6	7
$demand_{0,i}$	0	0.15	0.05	0.1	0.05	0.15	0.2	0.3

demand split on the bucket level

i	1
$demand_{0,i}$	0.15

bucket 2
(001)

i	2	3
$demand_{0,i}$	0.05	0.1

bucket 1
(01x)

i	4	5	6	7
$demand_{0,i}$	0.05	0.15	0.2	0.3

bucket 0
(1xx)

Our Algorithms: Binary Search in Buckets (BSB) – Half-split

Demand from node $0(000)'s$ perspective:

i	0 (src)	1	2	3	4	5	6	7
$demand_{0,i}$	0	0.15	0.05	0.1	0.05	0.15	0.2	0.3

demand split on the bucket level

i	1
$demand_{0,i}$	0.15
$\sum_{0}^{i} demand_{0,i}$	$\boxed{0.15}$

bucket 2
(001)

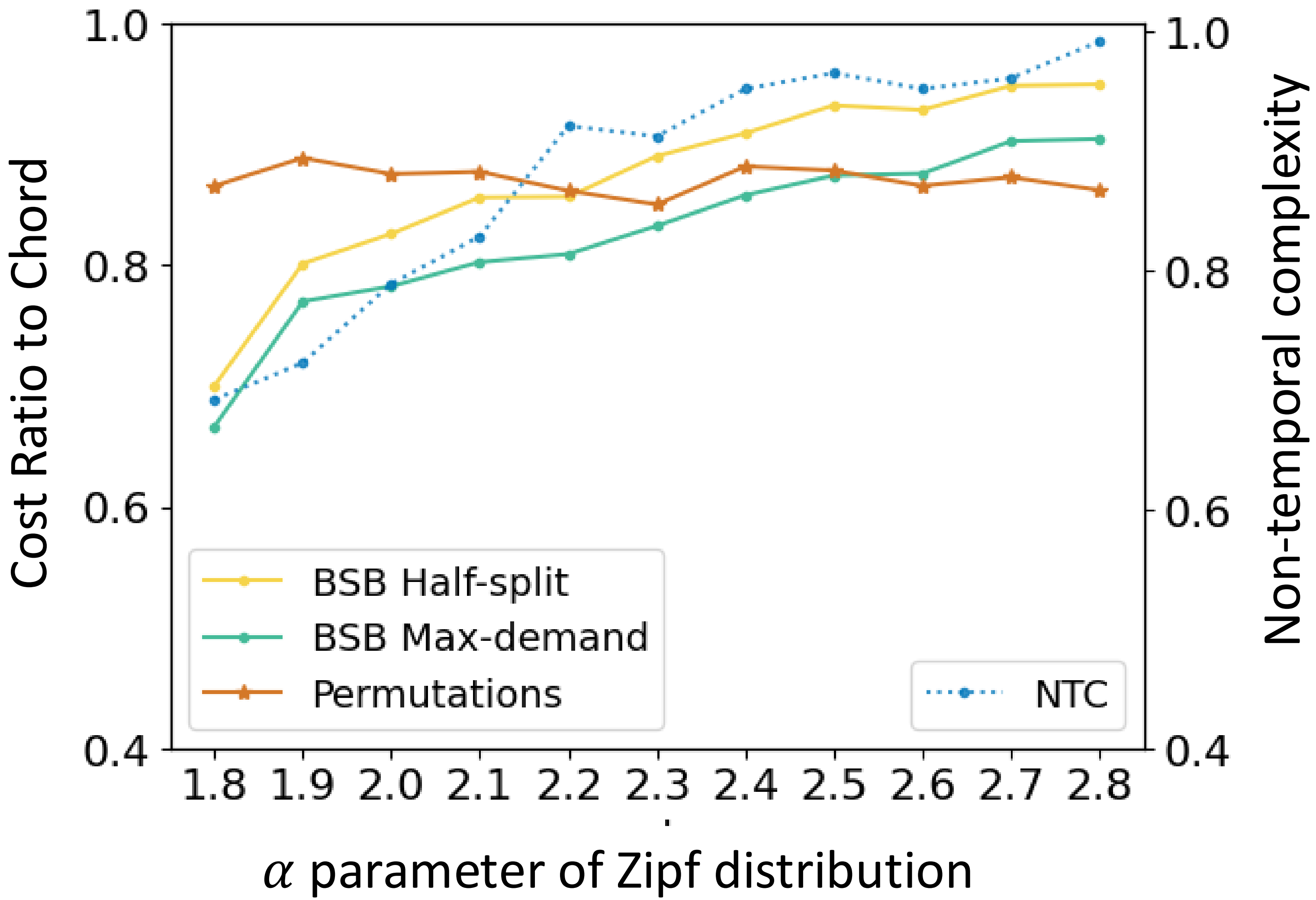
i	2	3
$demand_{0,i}$	0.05	0.1
$\sum_{0}^{i} demand_{0,i}$	0.05	$\boxed{0.15}$

bucket 1
(01x)

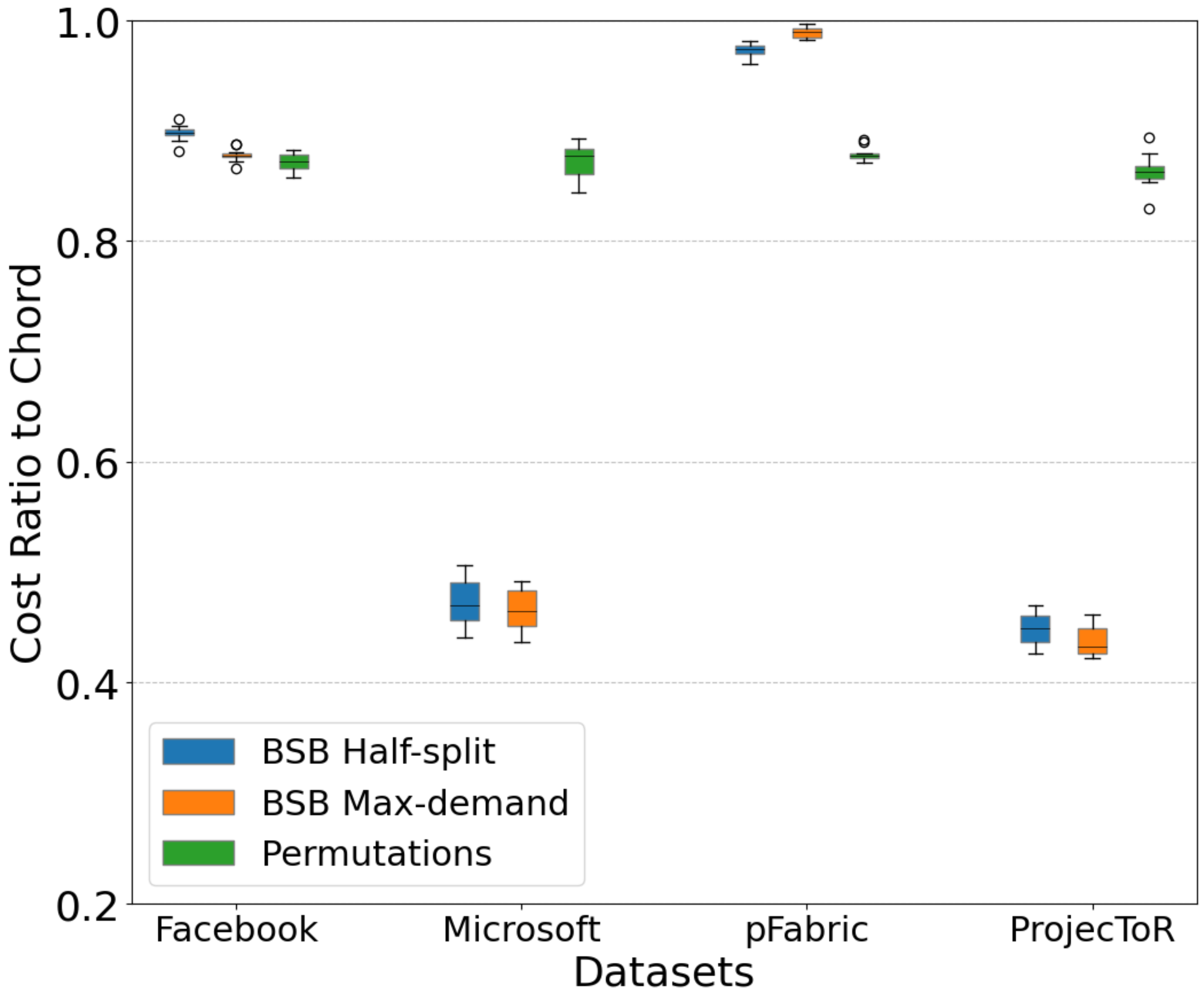
i	4	5	6	7
$demand_{0,i}$	0.05	0.15	0.2	0.3
$\sum_{0}^{i} demand_{0,i}$	0.05	0.2	$\boxed{0.4}$	0.7

bucket 0
(1xx)

Empirical Results: Synthetic Traffic Data



Empirical Results: Real-world Datasets



Conclusion & Future Work

- Conclusion
  - We introduced a demand-aware peer selection algorithm with XOR-based routing
  - With skewed demand, BSB reduced communication cost, by up to 43% compared to SOTA.
- Future work
  - Providing a randomized variant of the algorithm.
  - Deployment in other application areas, e.g. blockchain systems.

Full paper:
https://arxiv.org/pdf/2509.20974

Simulation code:
github.com/inet-tub/BSB

Our group's website:
tu.berlin/en/eninet