Self-Adjusting Topologies

Stefan Schmid

YuriFest

Happy birthday Yuri Gurevich!

Self-Adjusting Topologies

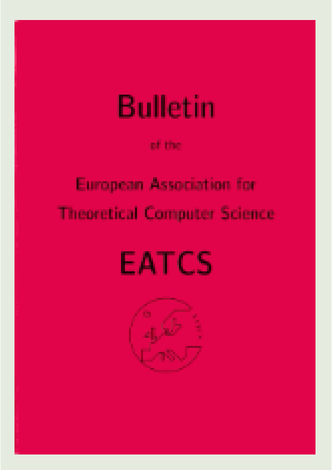
Stefan Schmid

YuriFest

Happy birthday Yuri Gurevich!

Collaboration:
B. EATCS

Since 2015



Bulletin

of the

European Association for

Theoretical Computer Science

EATCS

Self-Adjusting Topologies

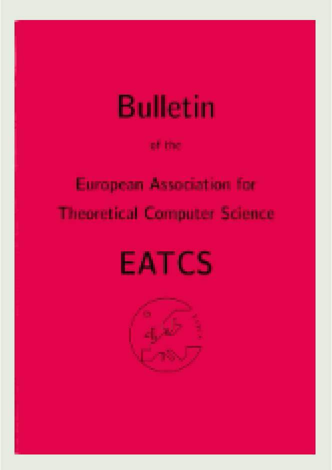
Stefan Schmid

YuriFest

Happy birthday Yuri Gurevich!

Collaboration: B. EATCS

Since 2015



Bulletin

of the

European Association for

Theoretical Computer Science

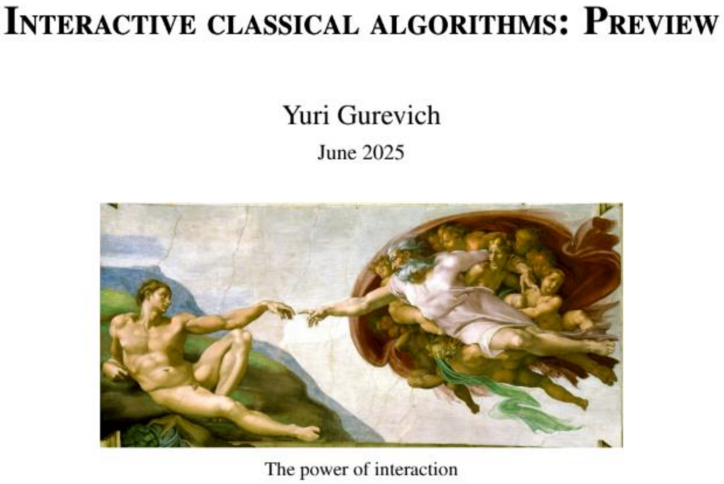
EATCS

Visit in Berlin

2022

Self-Adjusting Topologies

Stefan Schmid



INTERACTIVE CLASSICAL ALGORITHMS: PREVIEW

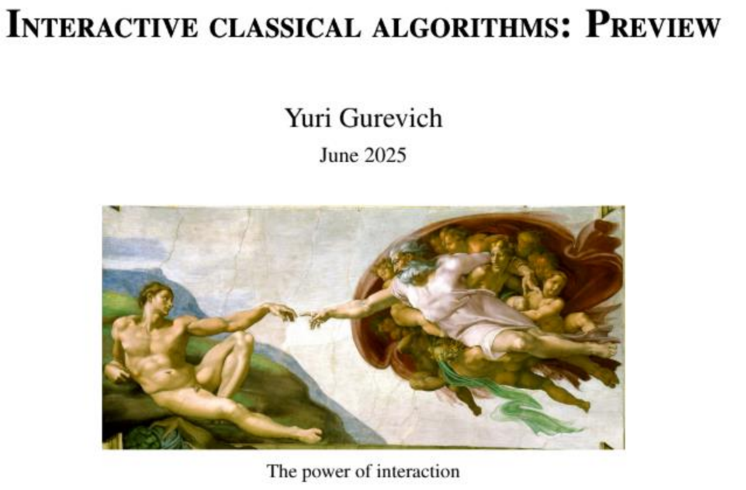
Yuri Gurevich

June 2025

The power of interaction

Self-Adjusting Topologies

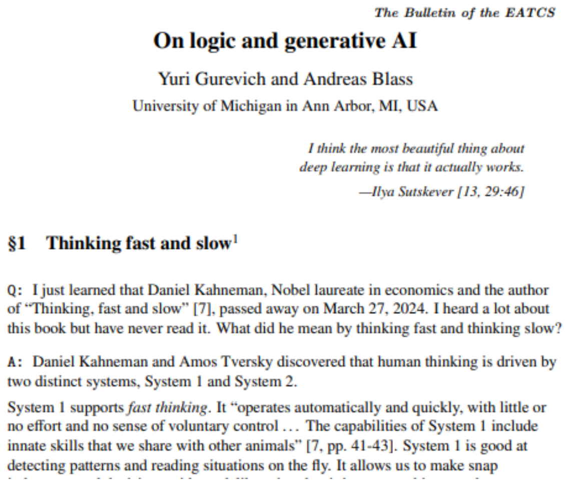
Stefan Schmid



INTERACTIVE CLASSICAL ALGORITHMS: PREVIEW

Yuri Gurevich
June 2025

The power of interaction



The Bulletin of the EATCS

On logic and generative AI

Yuri Gurevich and Andreas Blass
University of Michigan in Ann Arbor, MI, USA

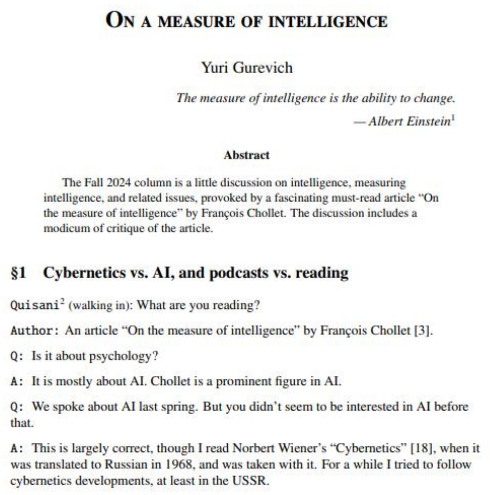
> *I think the most beautiful thing about deep learning is that it actually works.*
> —*Ilya Sutskever [13, 29:46]*

§1 Thinking fast and slow[1]

`Q:` I just learned that Daniel Kahneman, Nobel laureate in economics and the author of "Thinking, fast and slow" [7], passed away on March 27, 2024. I heard a lot about this book but have never read it. What did he mean by thinking fast and thinking slow?

`A:` Daniel Kahneman and Amos Tversky discovered that human thinking is driven by two distinct systems, System 1 and System 2.

System 1 supports *fast thinking*. It "operates automatically and quickly, with little or no effort and no sense of voluntary control . . . The capabilities of System 1 include innate skills that we share with other animals" [7, pp. 41-43]. System 1 is good at detecting patterns and reading situations on the fly. It allows us to make snap



ON A MEASURE OF INTELLIGENCE

Yuri Gurevich

> *The measure of intelligence is the ability to change.*
> —*Albert Einstein*[1]

Abstract

The Fall 2024 column is a little discussion on intelligence, measuring intelligence, and related issues, provoked by a fascinating must-read article "On the measure of intelligence" by François Chollet. The discussion includes a modicum of critique of the article.

§1 Cybernetics vs. AI, and podcasts vs. reading

`Quisani`[2] (walking in): What are you reading?

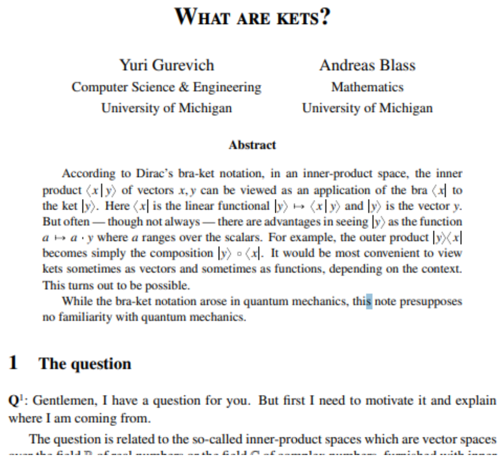
`Author:` An article "On the measure of intelligence" by François Chollet [3].

`Q:` Is it about psychology?

`A:` It is mostly about AI. Chollet is a prominent figure in AI.

`Q:` We spoke about AI last spring. But you didn't seem to be interested in AI before that.

`A:` This is largely correct, though I read Norbert Wiener's "Cybernetics" [18], when it was translated to Russian in 1968, and was taken with it. For a while I tried to follow cybernetics developments, at least in the USSR.



WHAT ARE KETS?

Yuri Gurevich	Andreas Blass
Computer Science & Engineering	Mathematics
University of Michigan	University of Michigan

Abstract

According to Dirac's bra-ket notation, in an inner-product space, the inner product $\langle x | y \rangle$ of vectors x, y can be viewed as an application of the bra $\langle x|$ to the ket $|y\rangle$. Here $\langle x|$ is the linear functional $|y\rangle \mapsto \langle x | y \rangle$ and $|y\rangle$ is the vector y. But often — though not always — there are advantages in seeing $|y\rangle$ as the function $a \mapsto a \cdot y$ where a ranges over the scalars. For example, the outer product $|y\rangle\langle x|$ becomes simply the composition $|y\rangle \circ \langle x|$. It would be most convenient to view kets sometimes as vectors and sometimes as functions, depending on the context. This turns out to be possible.

While the bra-ket notation arose in quantum mechanics, this note presupposes no familiarity with quantum mechanics.

1 The question

Q[1]: Gentlemen, I have a question for you. But first I need to motivate it and explain where I am coming from.

The question is related to the so-called inner-product spaces which are vector spaces

Self-Adjusting Topologies

Stefan Schmid

Many more at:

European Association for Theoretical Computer Science

Issues

Since 2003 all issues of the EATCS Bulletin have been produced entirely electronically with colour pictures and made available on the web for members.

From Number 101 and on, the Bulletin of EATCS will be published electronically using the open journal system

Number 145, February 2025, 195pp

- Access the table of contents and individual contributions via the Open Journal System
- Read the whole issue. (pdf 9,2 Mb)

Number 144, October 2024, 146pp

- Access the table of contents and individual contributions via the Open Journal System
- Read the whole issue. (pdf 2,9 Mb)

Number 143, June 2024, 117pp

- Access the table of contents and individual contributions via the Open Journal System
- Read the whole issue. (pdf 4,6 Mb)

Number 142, February 2024, 99pp

- Access the table of contents and individual contributions via the Open Journal System
- Read the whole issue. (pdf 29,2 Mb)

Number 141, October 2023, 190pp

- Access the table of contents and individual contributions via the Open Journal System
- Read the whole issue. (pdf 11,5 Mb)

Number 140, June 2023, 213pp

- Access the table of contents and individual contributions via the Open Journal System
- Read the whole issue. (pdf 11,5 Mb)

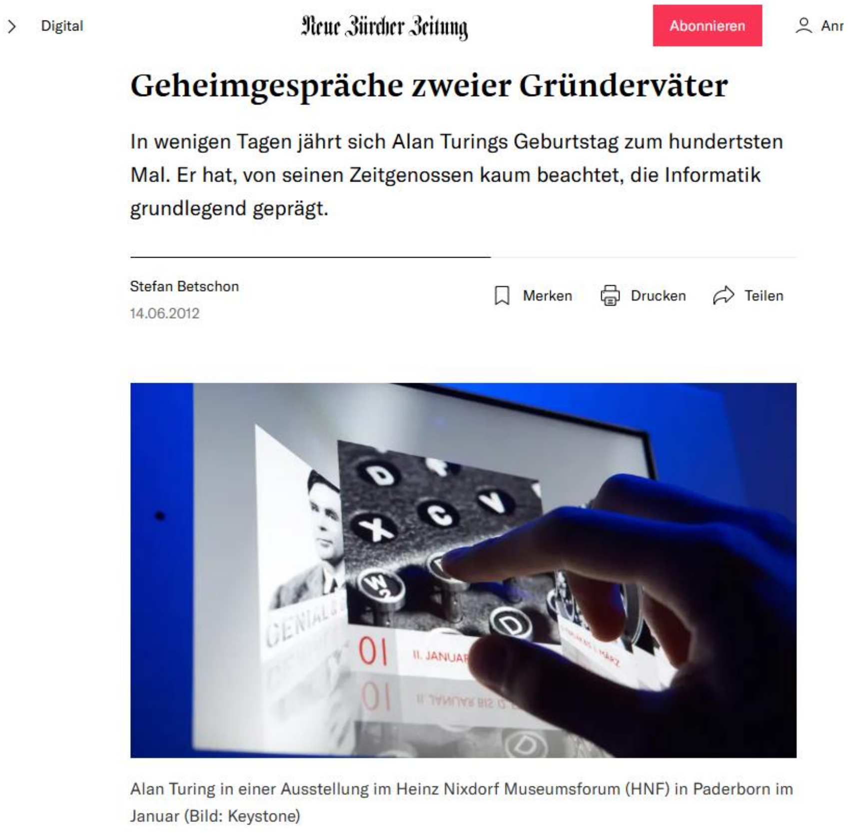
Number 139, February 2023, 136pp

- Access the table of contents and individual contributions via the Open Journal System
- Read the whole issue. (pdf 4,2 Mb)

Self-Adjusting Topologies

Stefan Schmid

Yuri on Alan Turing's 100th birthday:



Digital

Neue Zürcher Zeitung

Abonnieren

Anr

Geheimgespräche zweier Gründerväter

In wenigen Tagen jährt sich Alan Turings Geburtstag zum hundertsten Mal. Er hat, von seinen Zeitgenossen kaum beachtet, die Informatik grundlegend geprägt.

Stefan Betschon
14.06.2012

Merken Drucken Teilen

Alan Turing in einer Ausstellung im Heinz Nixdorf Museumsforum (HNF) in Paderborn im Januar (Bild: Keystone)

Self-Adjusting Topologies

Stefan Schmid

“We cannot direct the wind,
but we can adjust the sails.”
(Folklore)

Self-Adjusting Topologies

Stefan Schmid

“We cannot direct the wind,
but we can adjust the sails.”
(Folklore)

Not just my research, e.g. also Microsoft Research!

Self-Adjusting Topologies

Stefan Schmid

“We cannot direct the wind,
but we can adjust the sails.”
(Folklore)

Not just my research, e.g. also Microsoft Research!

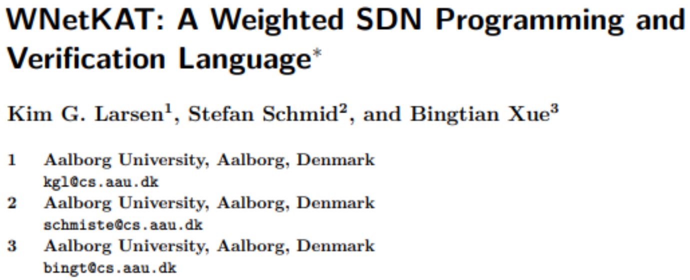
Acknowledgements:

German-Israeli Foundation for Scientific Research and Development

European Research Council
Established by the European Commission

Self-Adjusting Topologies

Stefan Schmid



WNetKAT: A Weighted SDN Programming and Verification Language*

Kim G. Larsen[1], Stefan Schmid[2], and Bingtian Xue[3]

1 Aalborg University, Aalborg, Denmark
kgl@cs.aau.dk
2 Aalborg University, Aalborg, Denmark
schmiste@cs.aau.dk
3 Aalborg University, Aalborg, Denmark
bingt@cs.aau.dk

AalWiNes: A Fast and Quantitative What-If Analysis Tool for MPLS Networks

Peter Gjøl Jensen
Aalborg University
Denmark

Dan Kristiansen
Aalborg University
Denmark

Stefan Schmid
Faculty of Computer Science
University of Vienna
Austria

Morten Konggaard Schou
Aalborg University
Denmark

Bernhard Clemens Schrenk
Faculty of Computer Science
University of Vienna

Jiří Srba
Aalborg University
Denmark

Acknowledgements:

GIF German-Israeli Foundation for Scientific Research and Development

erc European Research Council Established by the European Commission

CACM’25 (with Chen Avin)

Check out our **YouTube interviews**
on Reconfigurable Datacenter Networks:

Prof. Chen Avin
(BGU, Israel)

Prof. Stefan Schmid
(TU Berlin, Germany)

https://tinyurl.com/53c79jav

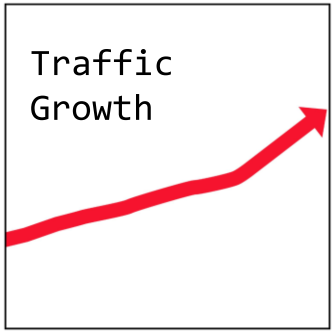
[Revolutionizing Datacenter Networks via Reconfigurable Topologies]()
Chen Avin and Stefan Schmid.
Communications of the ACM (**CACM**), 2025.
Watch here: https://www.youtube.com/@self-adjusting-networks-course

A bit of context

Data-Centric Applications

Datacenters ("hyper-scale")

Interconnecting networks:
a **critical infrastructure**
of our digital society.



Source: Facebook

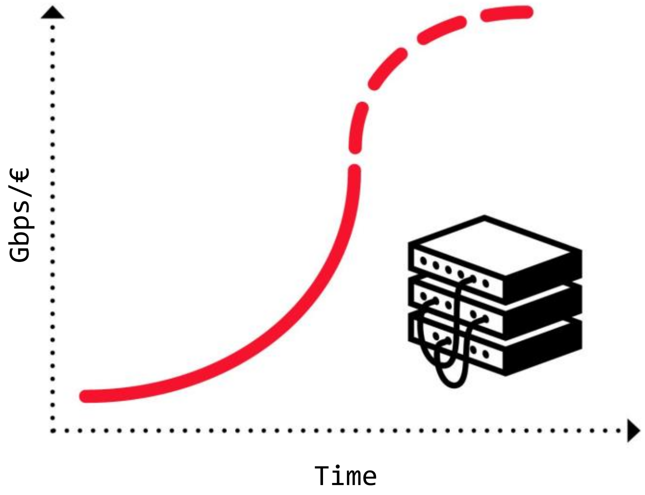
The Problem

Huge Infrastructure, Inefficient Use

- Network equipment reaching capacity limits
  - ⇀ Transistor density rates stalling
  - ⇀ “End of **Moore‘s Law** in networking” [1]

- Hence: more equipment, larger networks

- Resource intensive and: **inefficient**



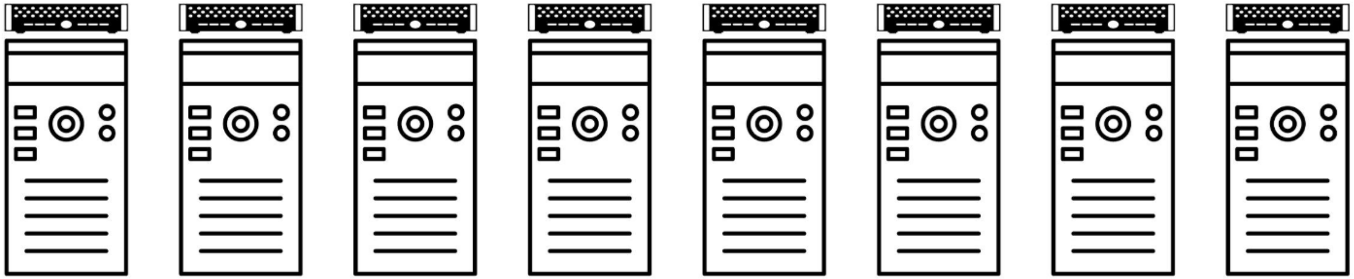
[1] Source: Microsoft, 2019

Annoying for companies,
opportunity for researchers

Root Cause

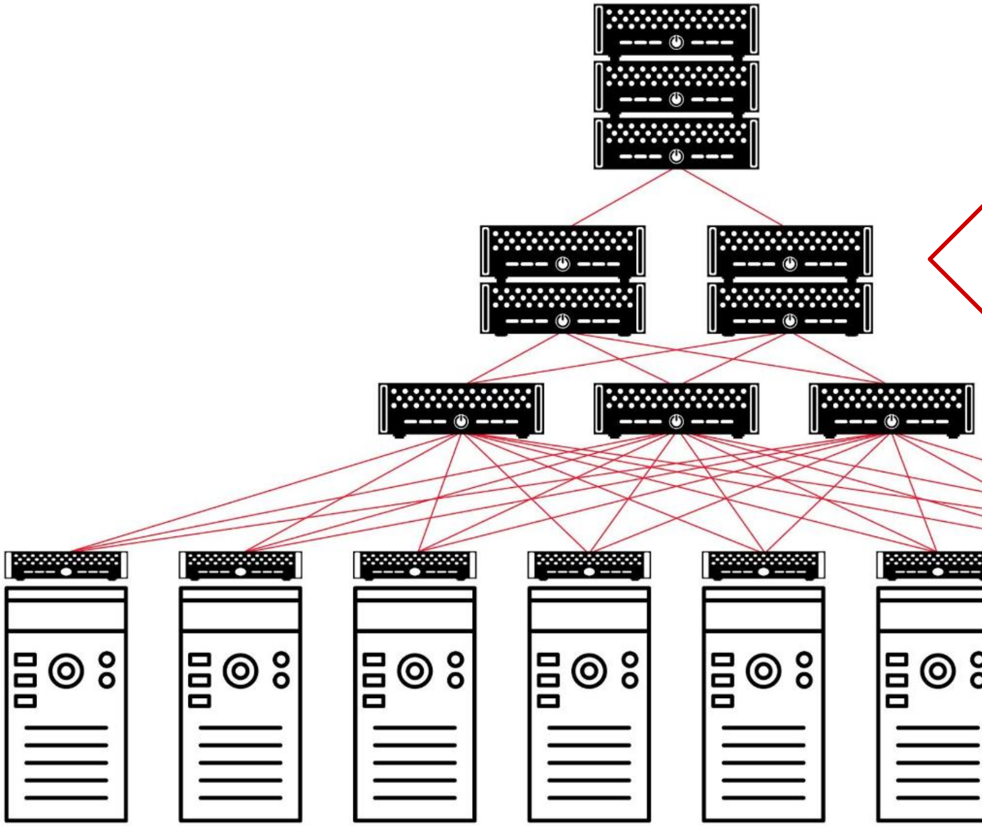
Fixed and Demand-Oblivious Topology

How to interconnect?



Root Cause

Fixed and Demand-Oblivious Topology

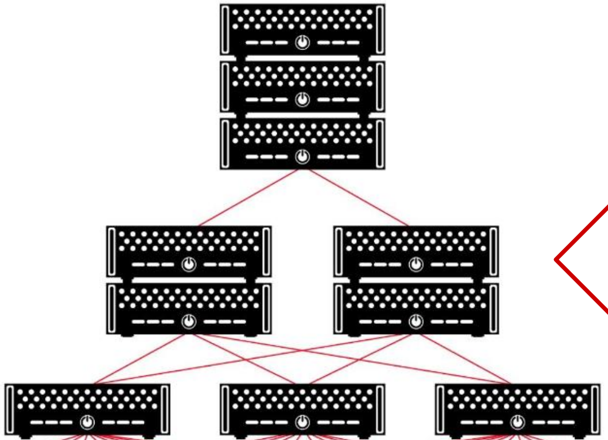


Many flavors,
but in common:
fixed and
oblivious to
actual demand.

Root Cause

Fixed and Demand-Oblivious Topology

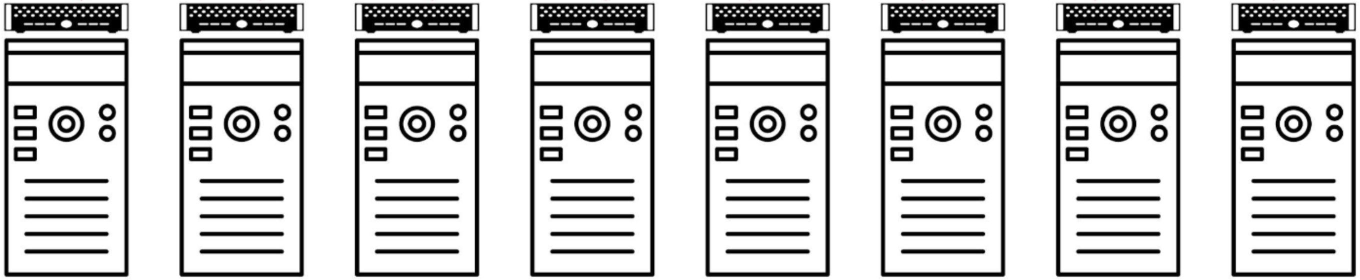
Highway which ignores actual traffic: **frustrating!**



Many flavors, but in common: fixed and **oblivious** to actual demand.

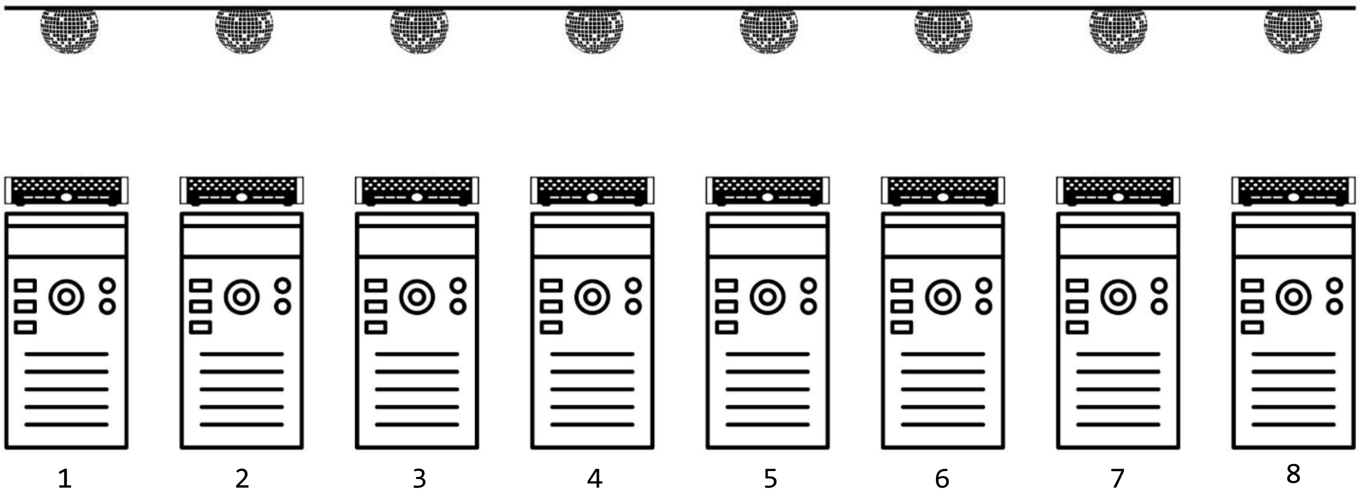
Our Vision

Flexible and Demand-Aware Topologies



Our Vision

Flexible and Demand-Aware Topologies



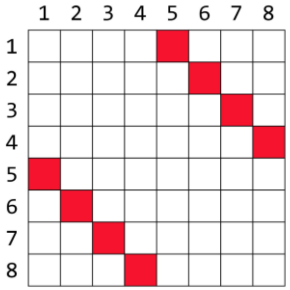
e.g., mirrors

new flexible interconnect

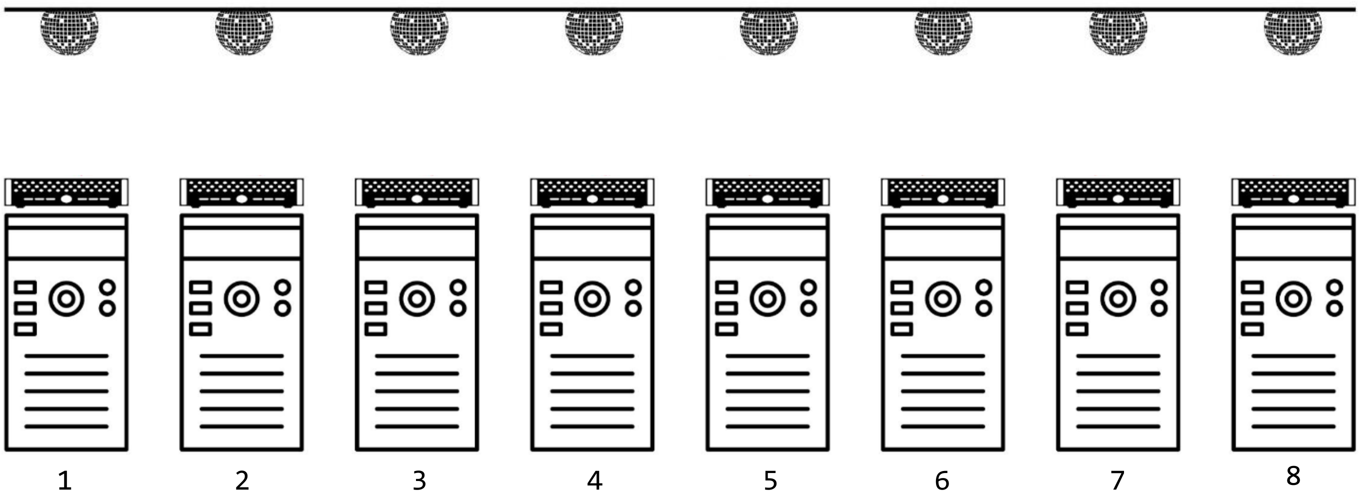
Our Vision

Flexible and Demand-Aware Topologies

demand matrix:



	1	2	3	4	5	6	7	8
1					🟥			
2						🟥		
3							🟥	
4								🟥
5	🟥							
6		🟥						
7			🟥					
8				🟥				



e.g., mirrors

new flexible interconnect

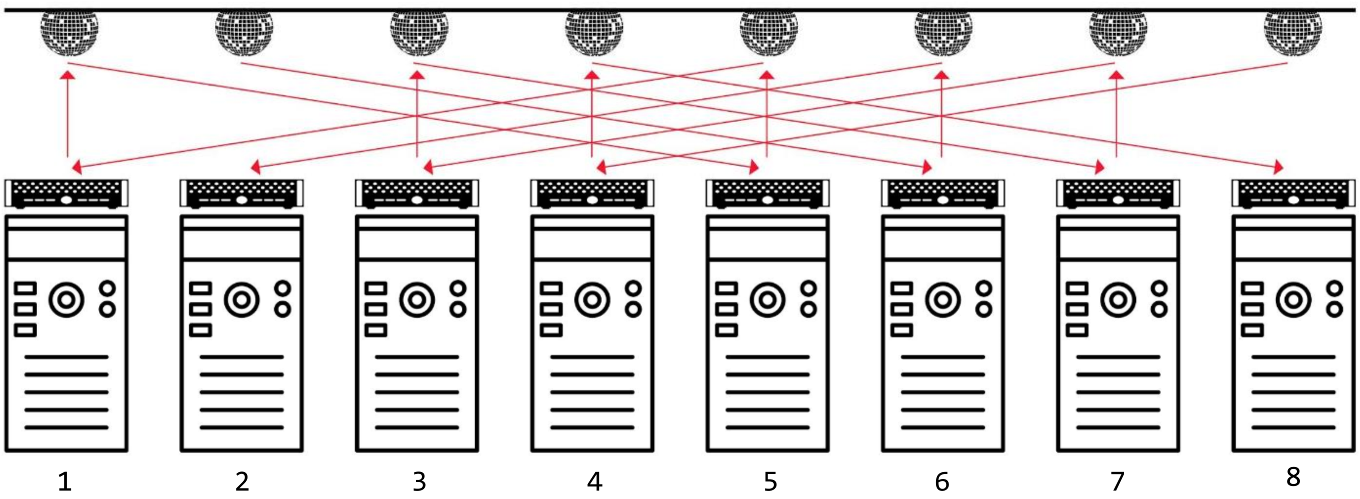
Our Vision

Flexible and Demand-Aware Topologies

Matches demand

demand matrix:

	1	2	3	4	5	6	7	8
1					🟥			
2						🟥		
3							🟥	
4								🟥
5	🟥							
6		🟥						
7			🟥					
8				🟥				



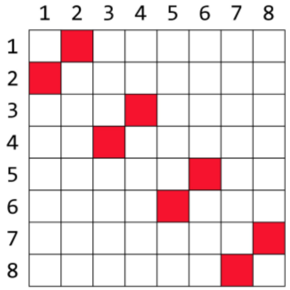
e.g., mirrors

new flexible interconnect

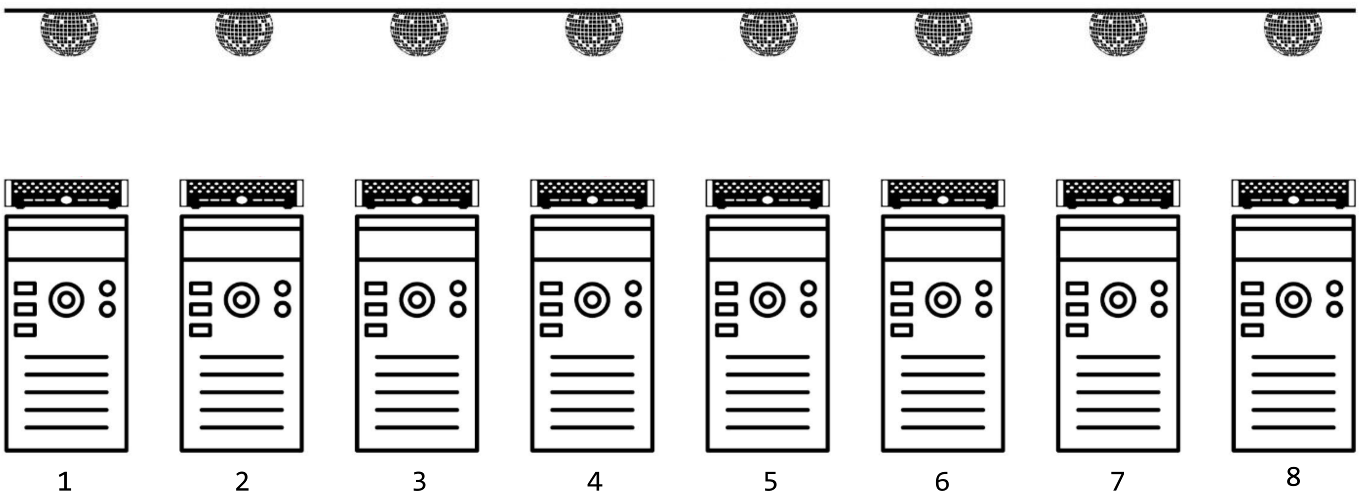
Our Vision

Flexible and Demand-Aware Topologies

new demand:



	1	2	3	4	5	6	7	8
1		🟥						
2	🟥							
3				🟥				
4			🟥					
5						🟥		
6					🟥			
7								🟥
8							🟥	



e.g., mirrors

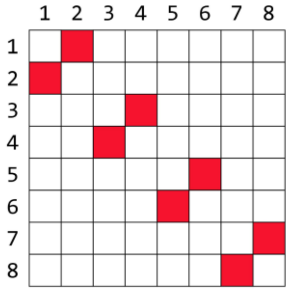
new flexible interconnect

Our Vision

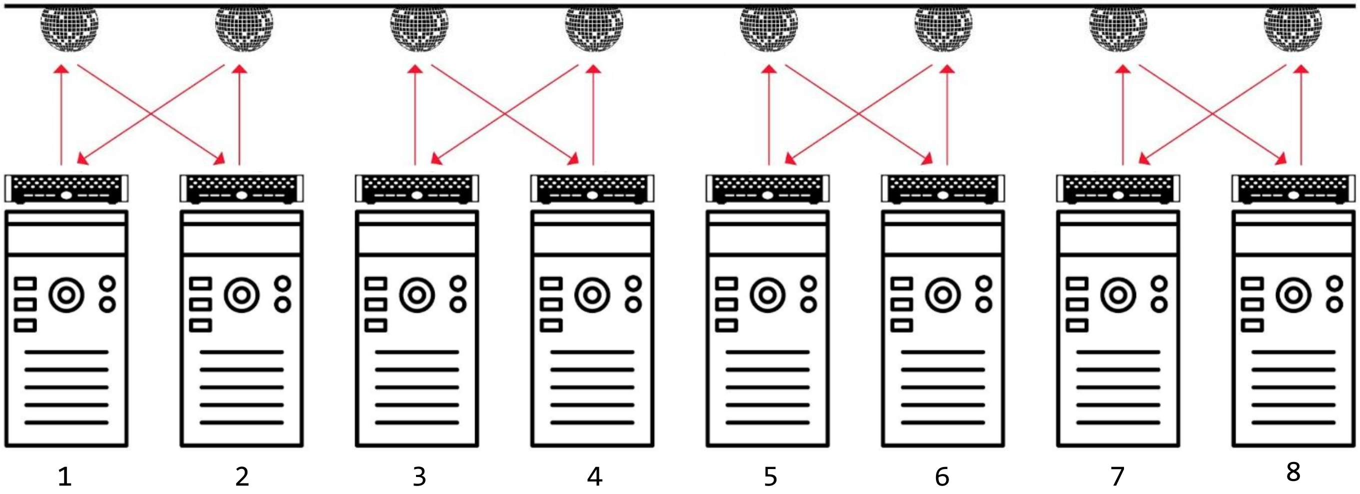
Flexible and Demand-Aware Topologies

Matches demand

new demand:



	1	2	3	4	5	6	7	8
1		🟥						
2	🟥							
3				🟥				
4			🟥					
5						🟥		
6					🟥			
7								🟥
8							🟥	



e.g.,
mirrors

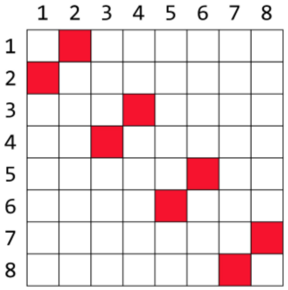
new flexible
interconnect

Our Vision

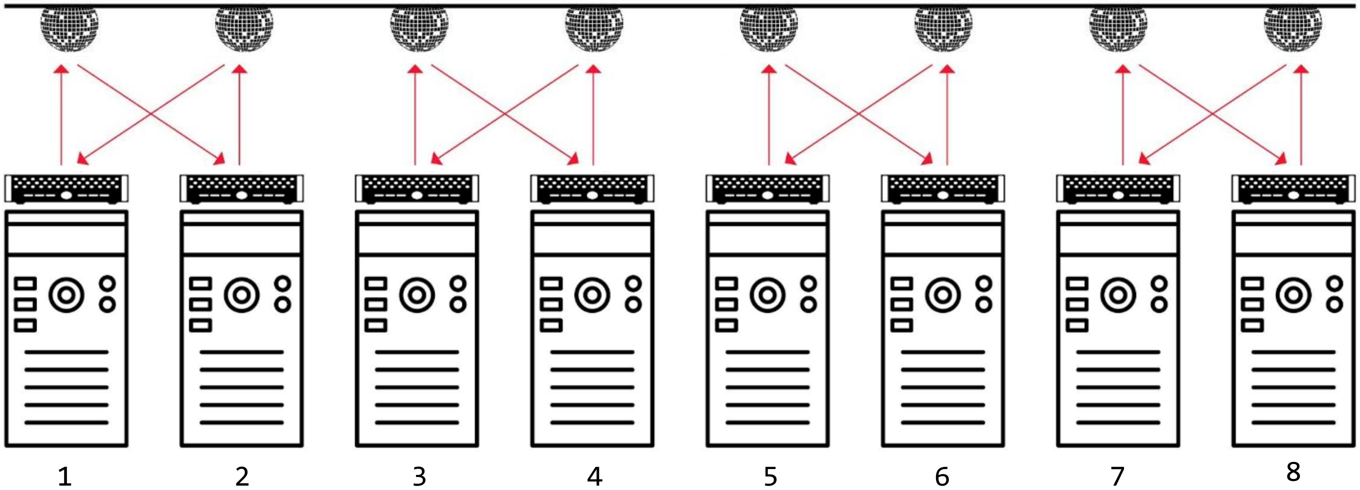
Flexible and Demand-Aware Topologies

Self-Adjusting Networks

new demand:



	1	2	3	4	5	6	7	8
1		🟥						
2	🟥							
3				🟥				
4			🟥					
5						🟥		
6					🟥			
7								🟥
8							🟥	



e.g., mirrors

new flexible interconnect

Our Motivation

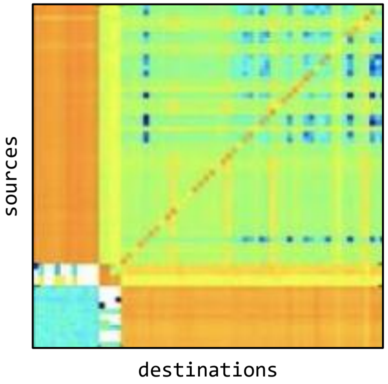
Much Structure in the Demand

Empirical studies:

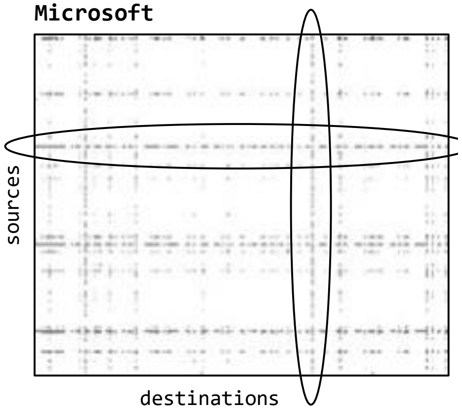
traffic matrices sparse and skewed

traffic bursty over time

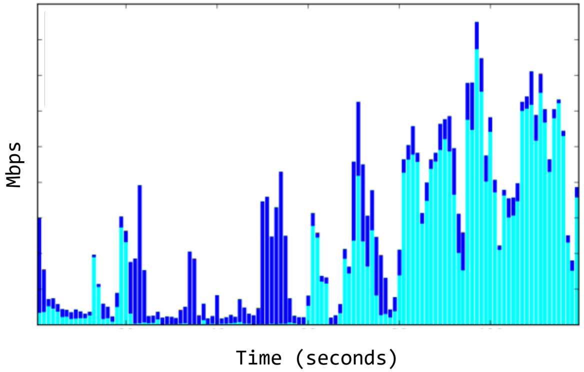
Facebook



Microsoft



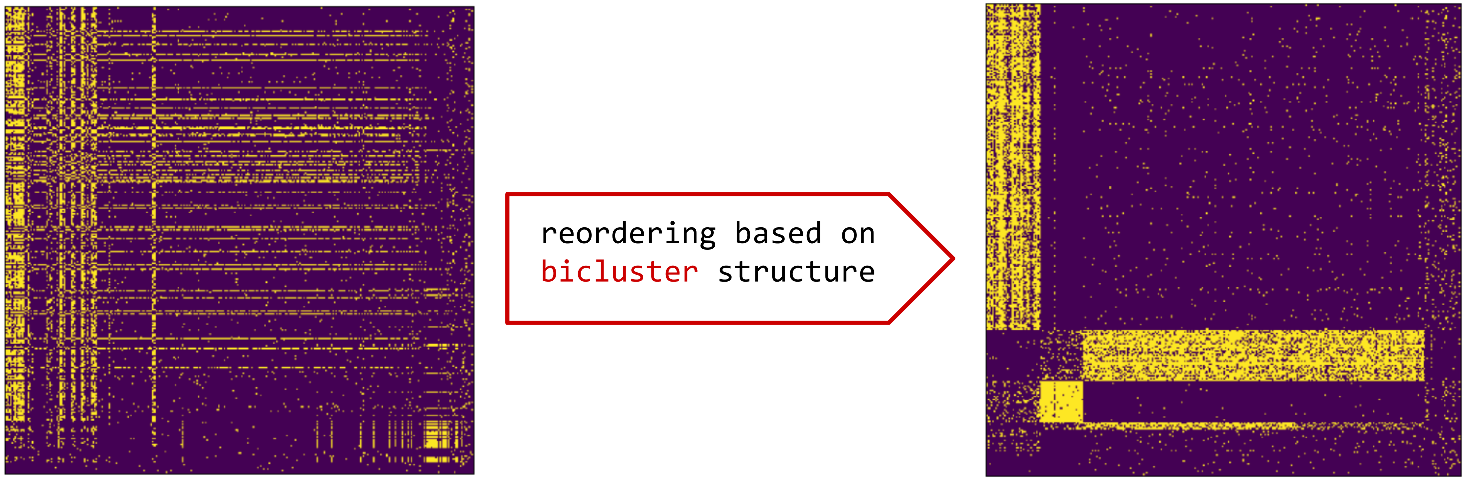
Facebook



Our hypothesis: can be exploited.

Traffic is also clustered:

Small Stable Clusters

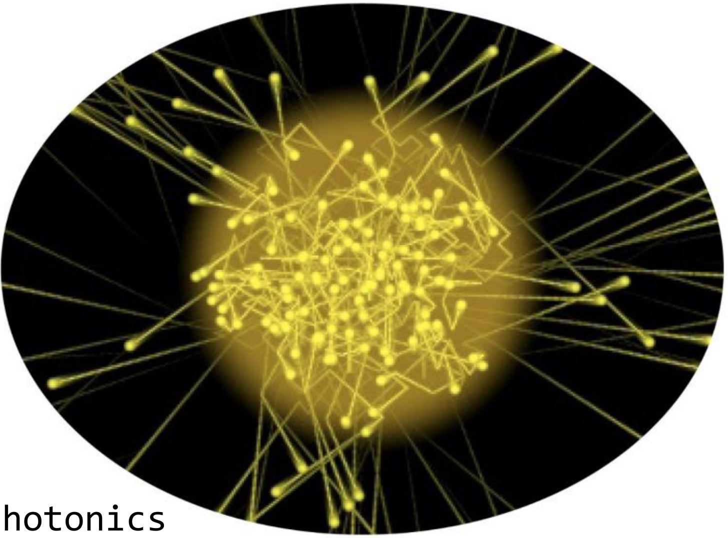


reordering based on bicluster structure

Opportunity: *exploit* with little reconfigurations!

Förster et al., Analyzing the Communication Clusters in Datacenters. WWW 2023

Sounds Crazy?
Emerging Enabling Technology.



Photonics

H2020:

“Photonics one of only five key enabling technologies for future prosperity.”

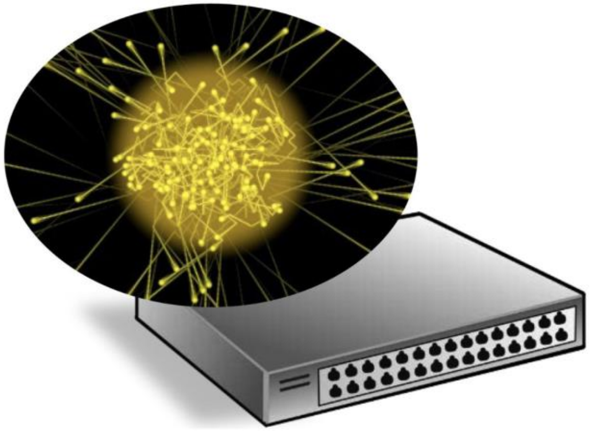
US National Research Council:

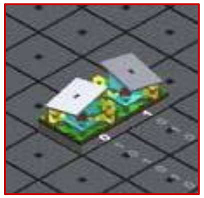
“Photons are the new Electrons.”

Enabler

Novel Reconfigurable Optical Switches

- **Spectrum** of prototypes
  - Different sizes, different reconfiguration times
  - From our last year’s ACM **SIGCOMM** workshop OptSys





Prototype 1

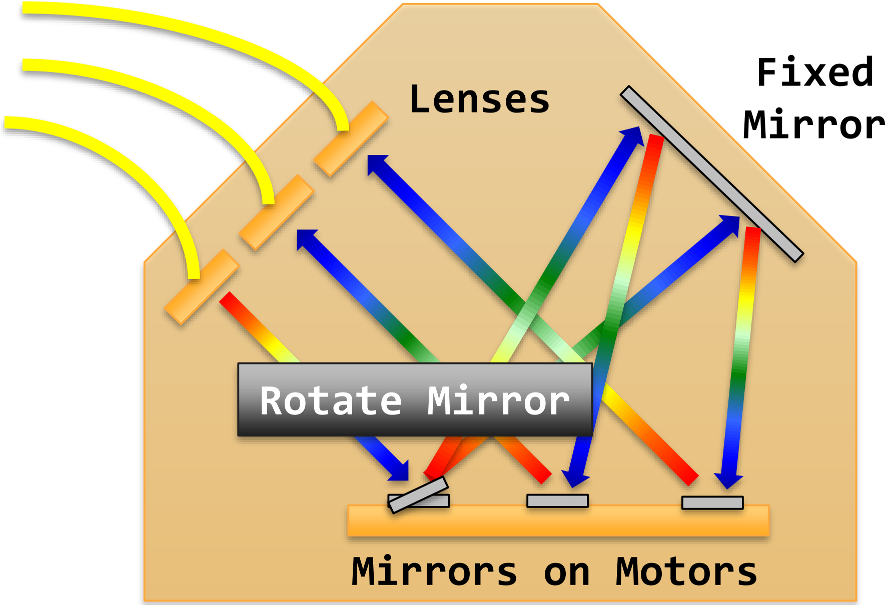
Prototype 2

Prototype 3

Example

Optical Circuit Switch

- ⇢ Optical Circuit Switch rapid adaption of physical layer
  - ⇀ Based on rotating mirrors



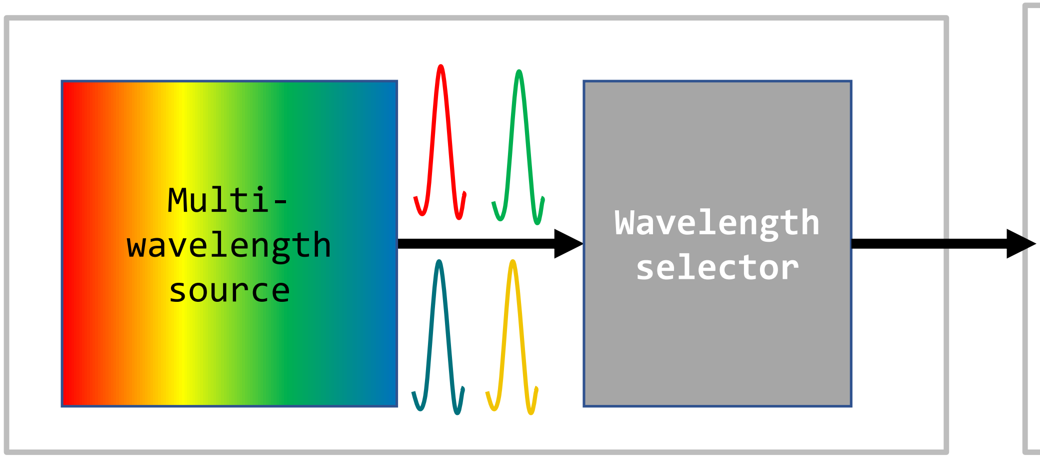
Optical Circuit Switch

By Nathan Farrington, SIGCOMM 2010

Another Example

Tunable Lasers

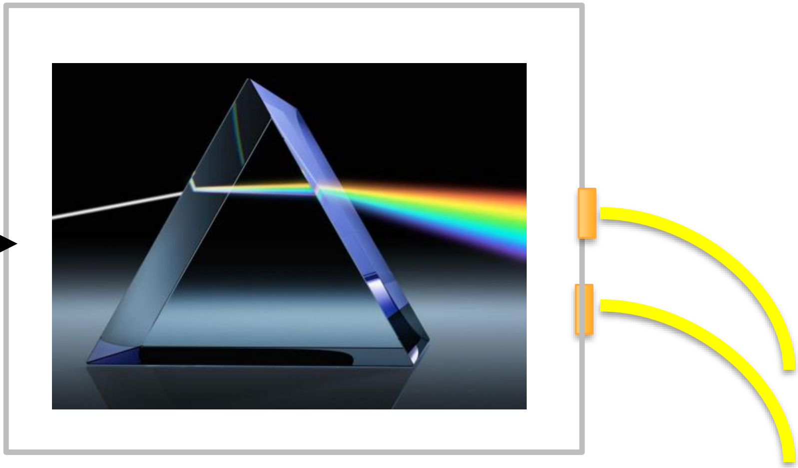
- ⇢ Depending on wavelength, forwarded differently
- ⇢ Optical switch is passive



Multi-wavelength source

Wavelength selector

Electrical switch with tunable laser



Optical switch

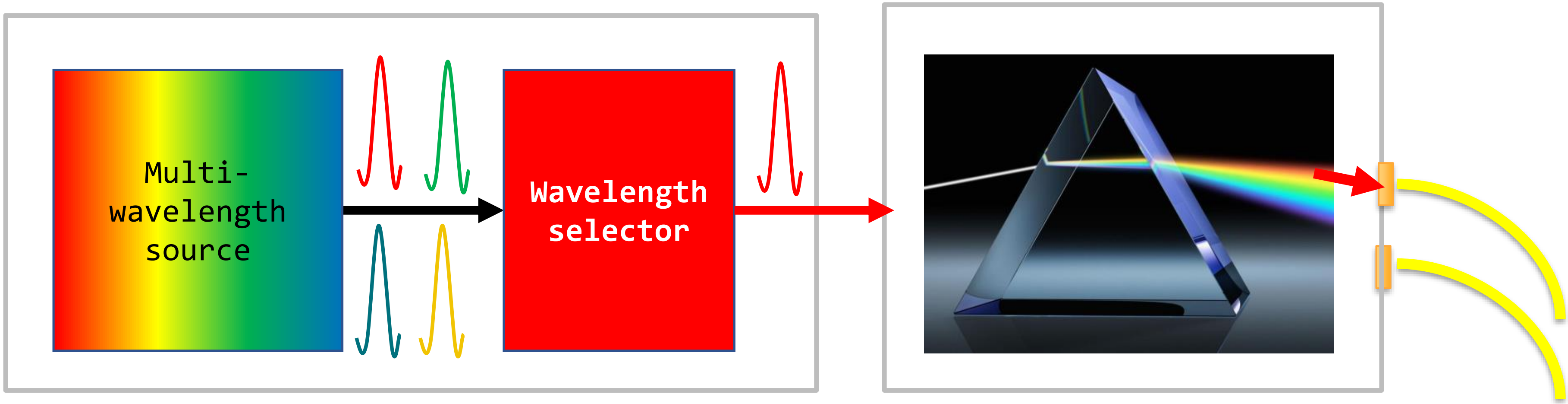
Passive

Ballani et al., Sirius, ACM SIGCOMM 2020.

Another Example

Tunable Lasers

- ⇢ Depending on wavelength, forwarded differently
- ⇢ Optical switch is passive



Electrical switch
with tunable laser

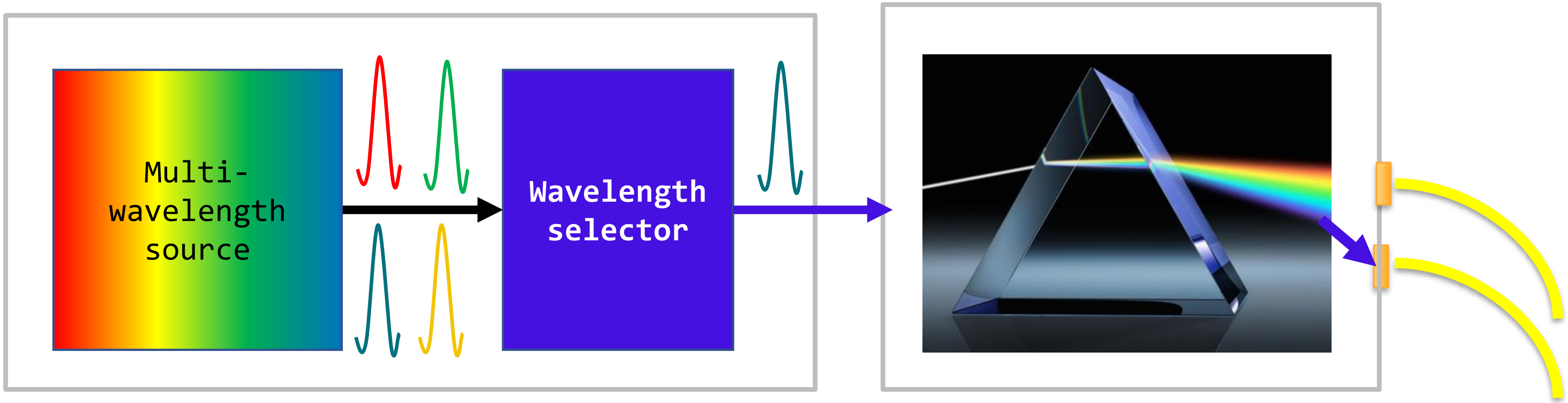
Optical switch
Passive

Ballani et al., Sirius, ACM SIGCOMM 2020.

Another Example

Tunable Lasers

- ⇢ Depending on wavelength, forwarded differently
- ⇢ Optical switch is passive

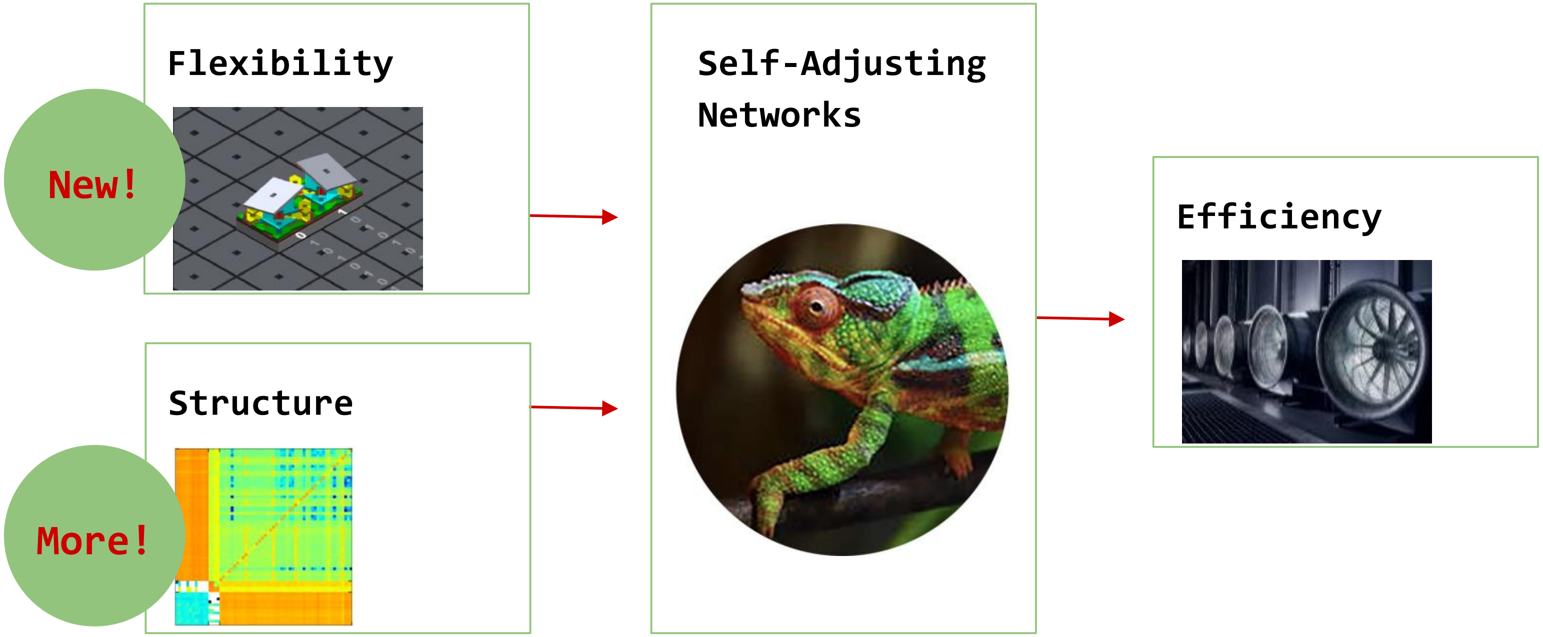


Electrical switch with tunable laser

Optical switch *Passive*

Ballani et al., Sirius, ACM SIGCOMM 2020.

The Big Picture



Flexibility

New!

Structure

More!

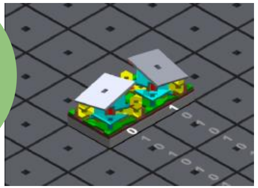
Self-Adjusting Networks

Efficiency

Now is the time!

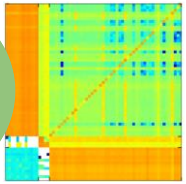
The Big Picture

Flexibility



New!

Structure



More!

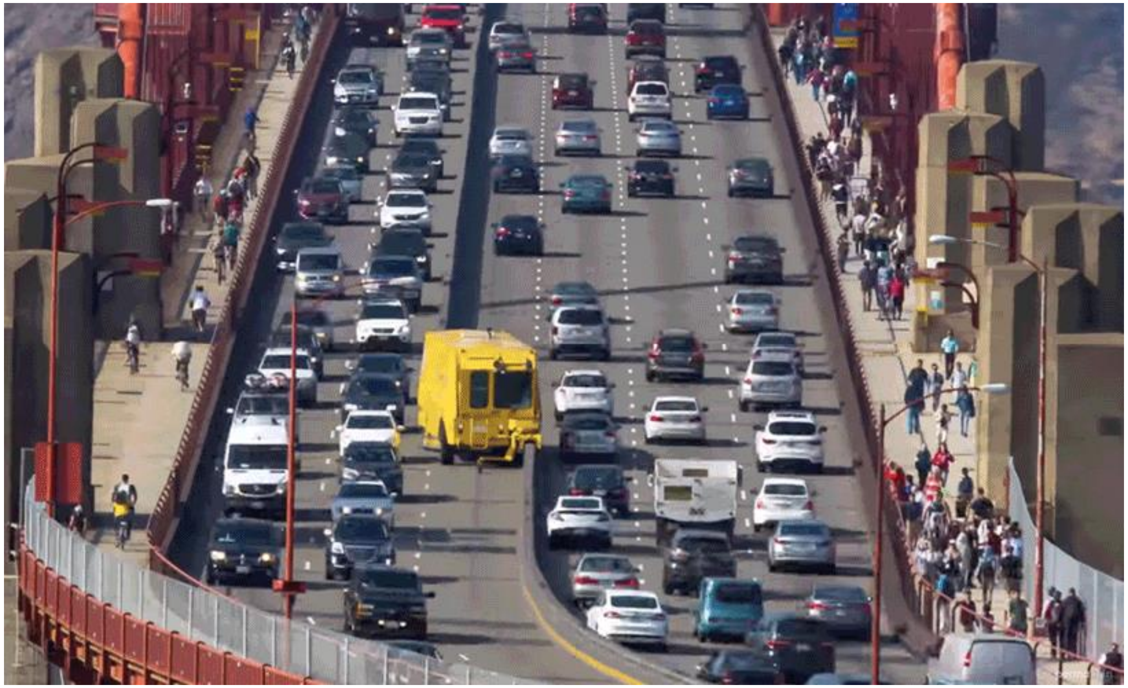
Self-Adjusting Networks

Efficiency

Now is the time!

Our goal: Develop the theoretical **foundations** of demand-aware, self-adjusting networks.

Analogy



Golden Gate Zipper

Unique Position

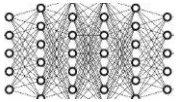
Demand-Aware, Self-Adjusting Systems

Everywhere, but mainly
in software

Algorithmic trading

NETFLIX

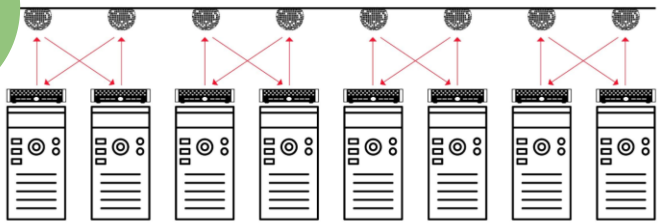
Recommender systems



Neural networks

VS

**Our focus:
in hardware**



Question 1:

How to Quantify such “Structure” in the Demand?

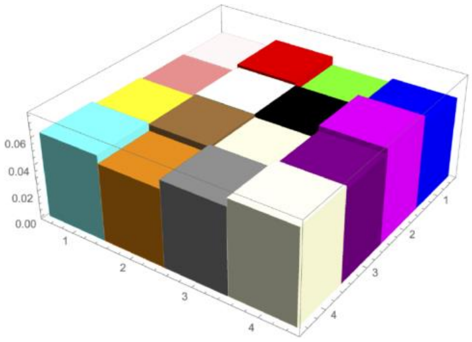
Intuition

Which demand has more structure?

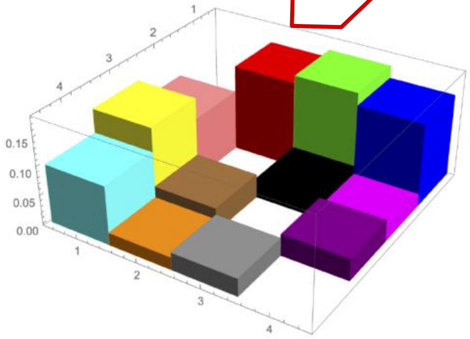
⇢ Traffic matrices of two different distributed ML applications

  ⇀ GPU-to-GPU

Color = communication pair



VS

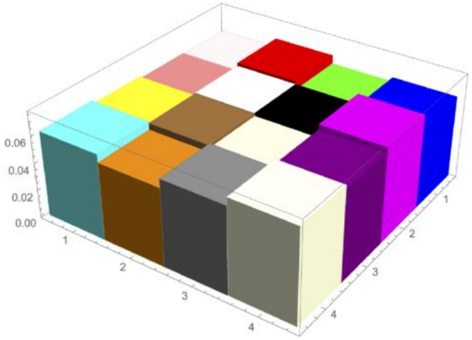


Intuition

Which demand has more structure?

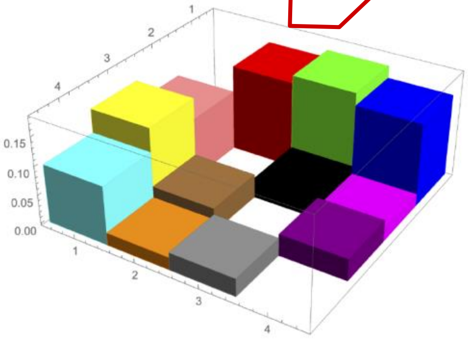
- ⇢ Traffic matrices of two different distributed ML applications
  - → GPU-to-GPU

Color = communication pair



More uniform

VS



More structure

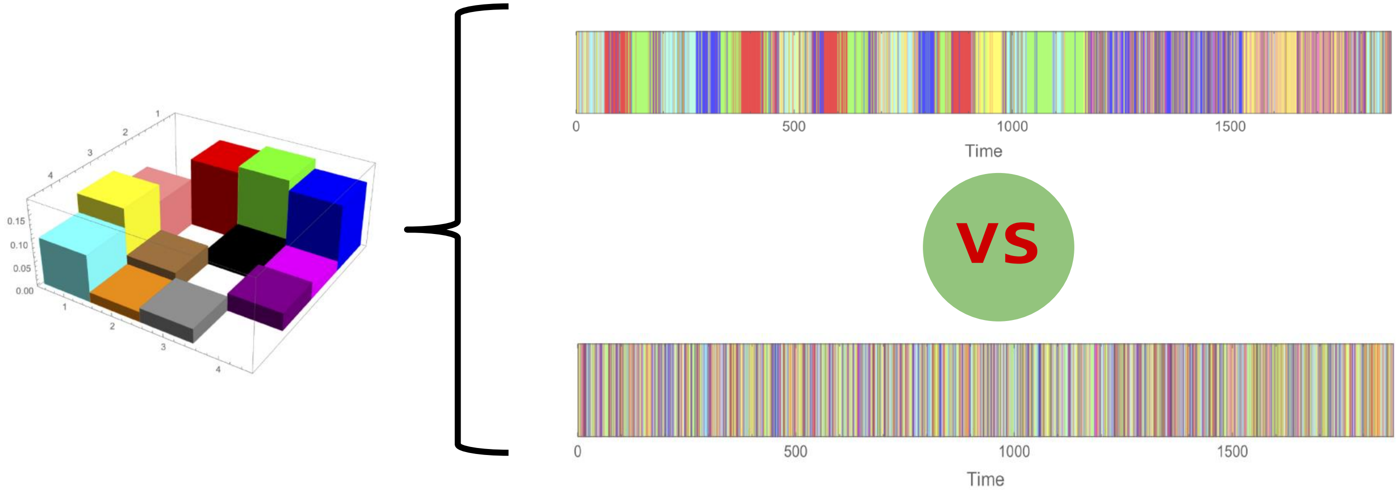
Intuition

Spatial vs temporal structure

⇢ Two different ways to generate same traffic matrix:

⇀ Same non-temporal structure

⇢ Which one has more structure?

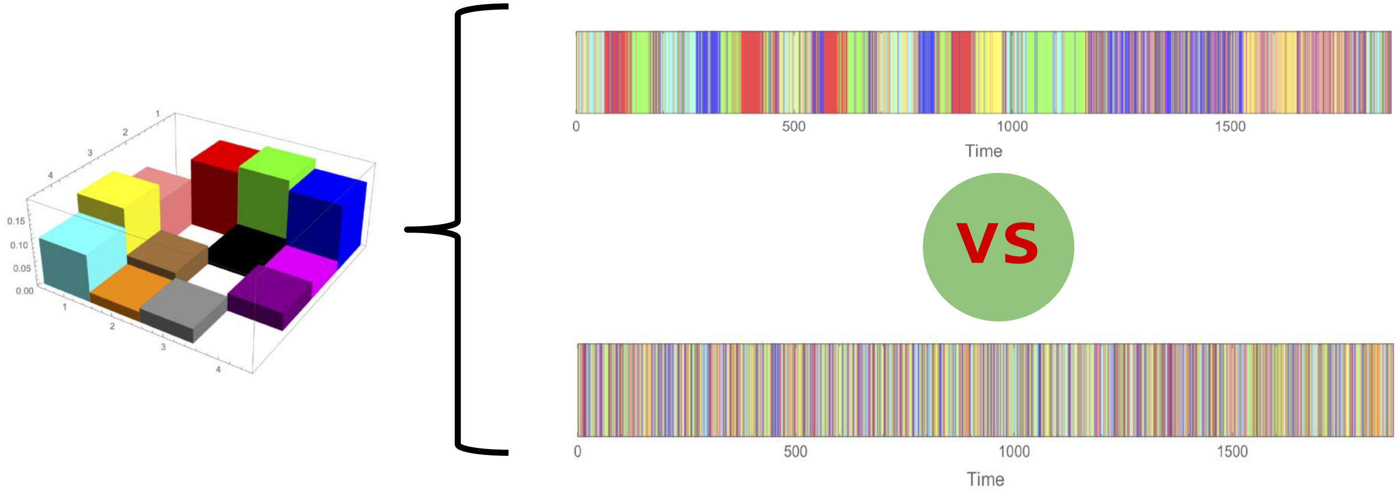


Intuition

Spatial vs temporal structure

- Two different ways to generate same traffic matrix:
    - Same non-temporal structure

- Which one has more structure?

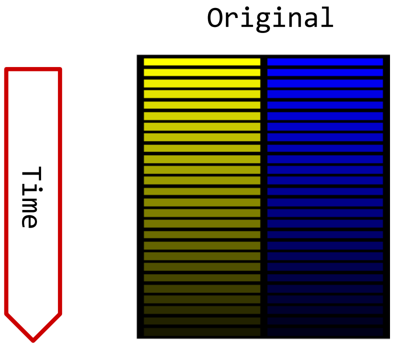


Systematically?

Trace Complexity

Information-Theoretic Approach

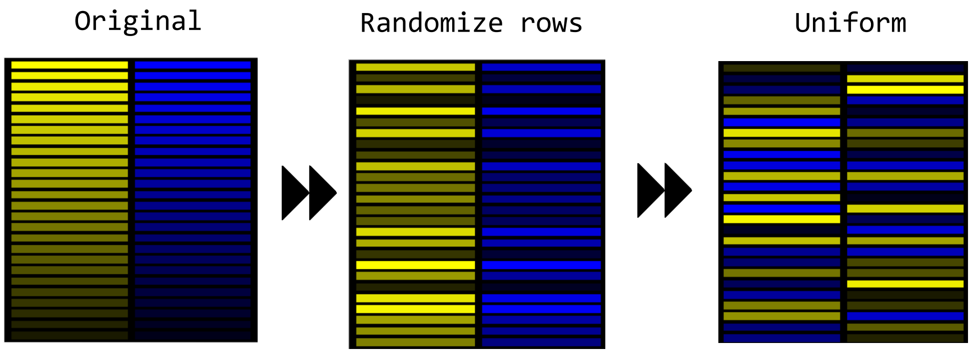
“Shuffle&Compress”



Trace Complexity

Information-Theoretic Approach

“Shuffle&Compress”



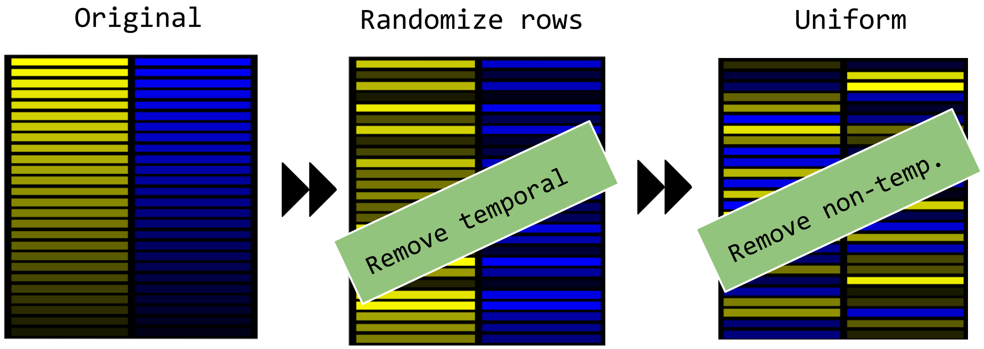
Increasing complexity (systematically randomized)

More structure (compresses better)

Trace Complexity

Information-Theoretic Approach

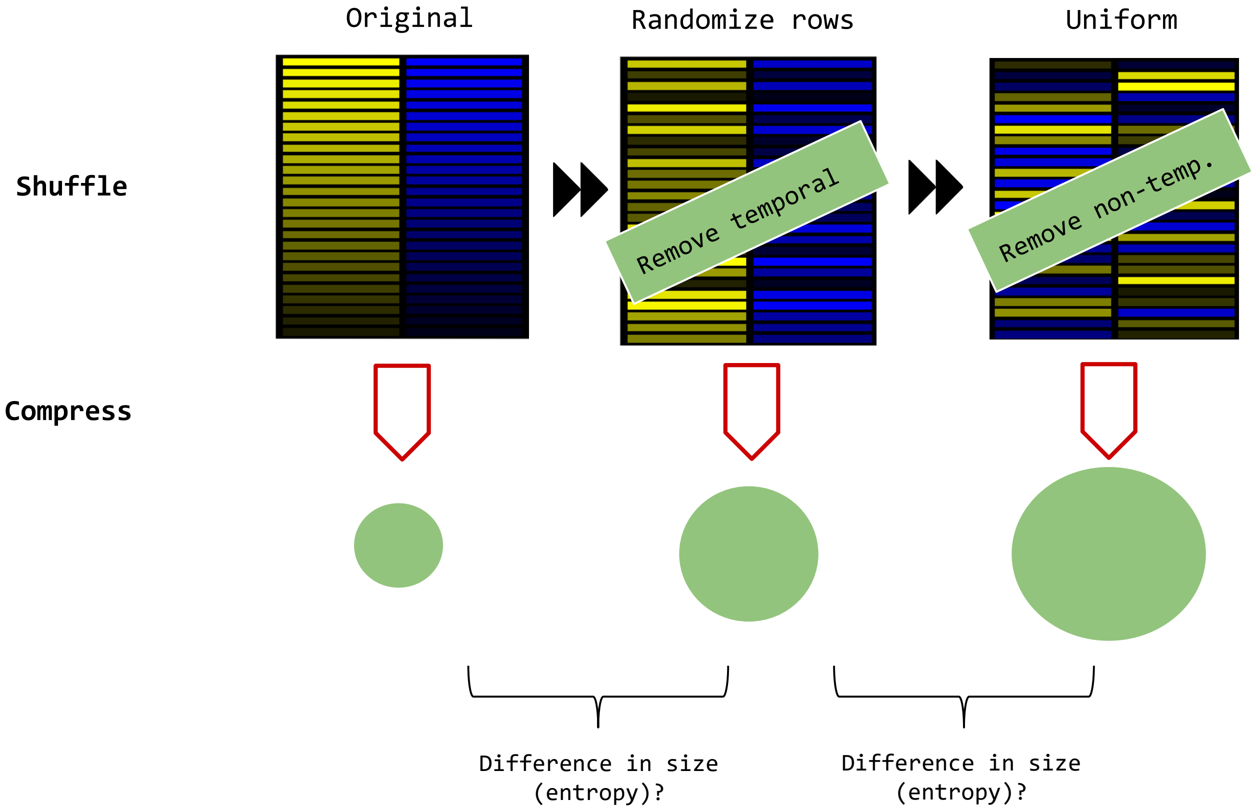
“Shuffle&Compress”



Trace Complexity

Information-Theoretic Approach

“Shuffle&Compress”



Original

Randomize rows

Uniform

Shuffle

Remove temporal

Remove non-temp.

Compress

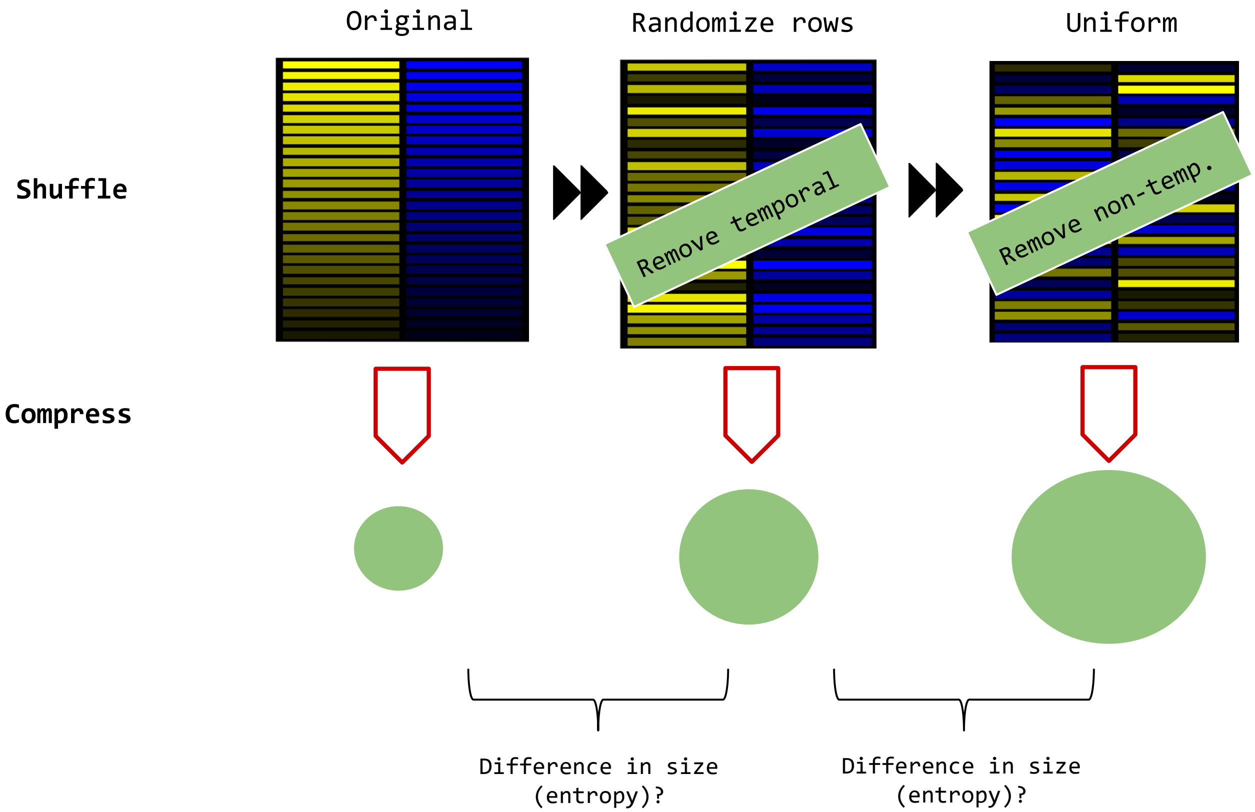
Difference in size (entropy)?

Difference in size (entropy)?

Trace Complexity

Information-Theoretic Approach

“Shuffle&Compress”



Original

Randomize rows

Uniform

Shuffle

Remove temporal

Remove non-temp.

Compress

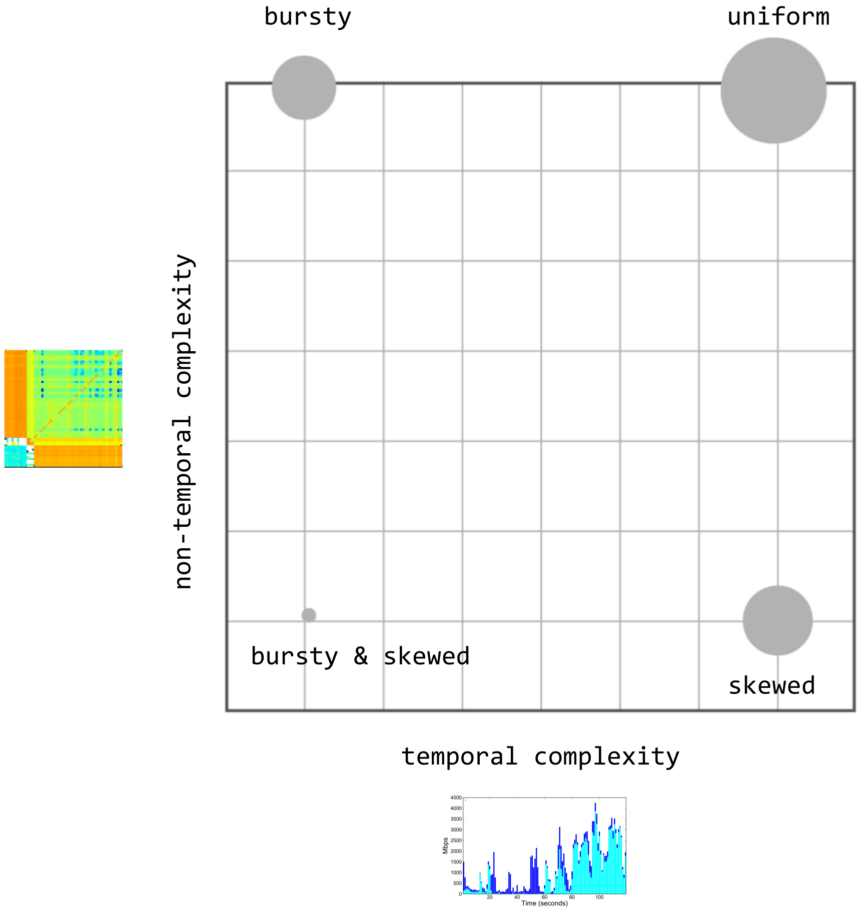
Difference in size (entropy)?

Difference in size (entropy)?

Can be used to define 2-dimensional complexity map!

Our Methodology

Complexity Map

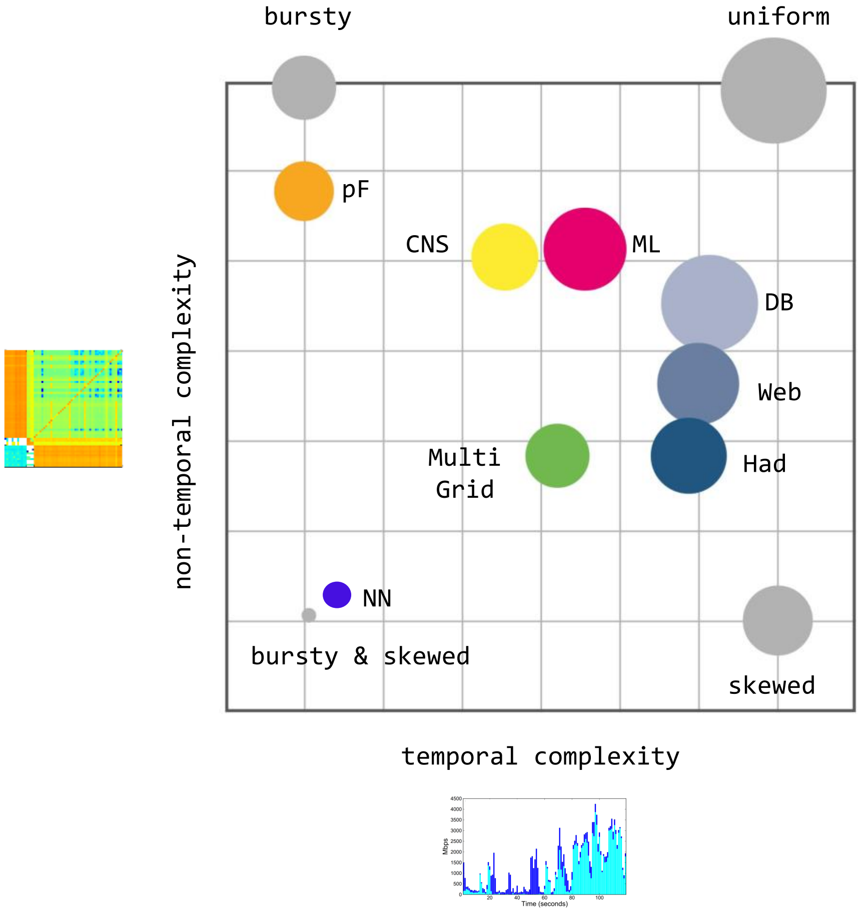


No structure

Our **approach:** iterative **randomization and compression** of trace to identify dimensions of structure.

Our Methodology

Complexity Map



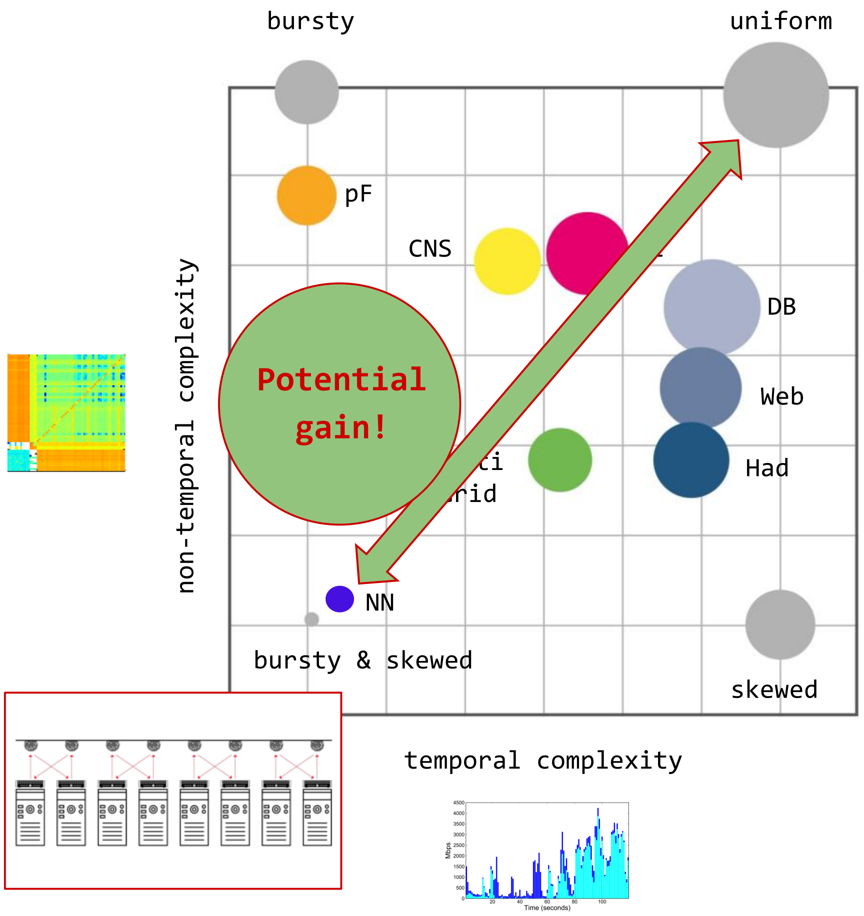
No structure

Our **approach**: iterative **randomization and compression** of trace to identify dimensions of structure.

Different structures!

Our Methodology

Complexity Map

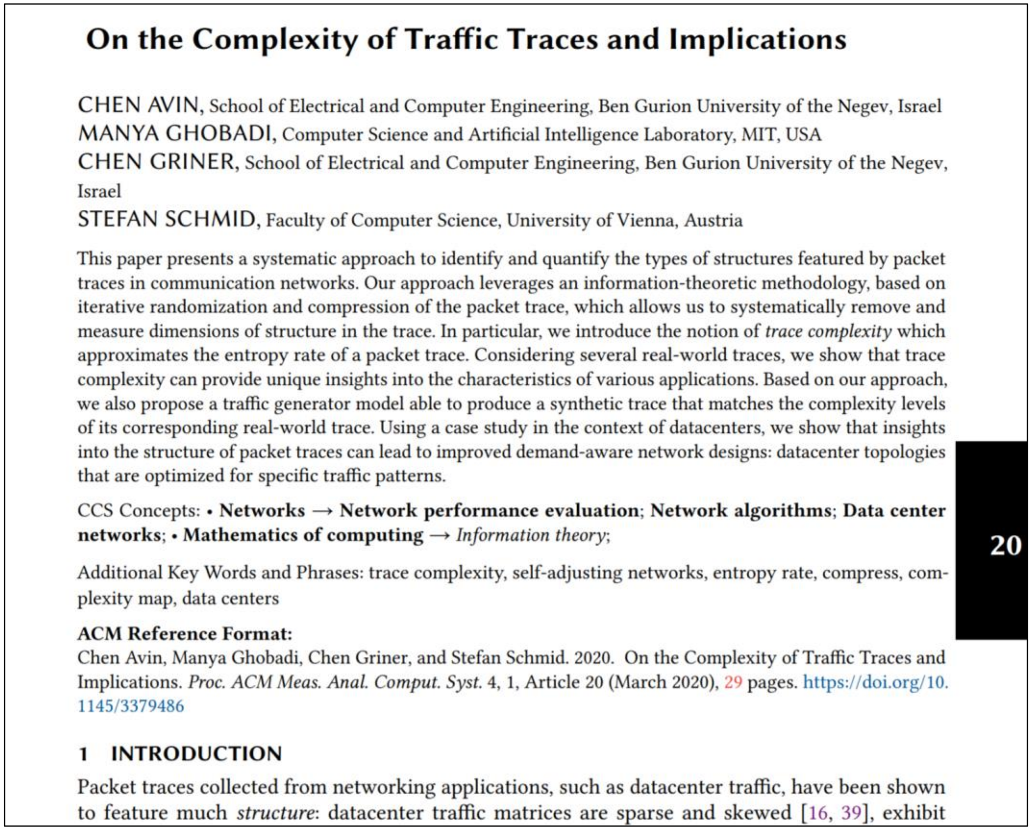


Our **approach**: iterative **randomization and compression** of trace to identify dimensions of structure.

Different structures!

Further Reading

ACM SIGMETRICS 2020



On the Complexity of Traffic Traces and Implications

CHEN AVIN, School of Electrical and Computer Engineering, Ben Gurion University of the Negev, Israel
MANYA GHOBADI, Computer Science and Artificial Intelligence Laboratory, MIT, USA
CHEN GRINER, School of Electrical and Computer Engineering, Ben Gurion University of the Negev, Israel
STEFAN SCHMID, Faculty of Computer Science, University of Vienna, Austria

This paper presents a systematic approach to identify and quantify the types of structures featured by packet traces in communication networks. Our approach leverages an information-theoretic methodology, based on iterative randomization and compression of the packet trace, which allows us to systematically remove and measure dimensions of structure in the trace. In particular, we introduce the notion of *trace complexity* which approximates the entropy rate of a packet trace. Considering several real-world traces, we show that trace complexity can provide unique insights into the characteristics of various applications. Based on our approach, we also propose a traffic generator model able to produce a synthetic trace that matches the complexity levels of its corresponding real-world trace. Using a case study in the context of datacenters, we show that insights into the structure of packet traces can lead to improved demand-aware network designs: datacenter topologies that are optimized for specific traffic patterns.

CCS Concepts: • **Networks** → **Network performance evaluation**; **Network algorithms**; **Data center networks**; • **Mathematics of computing** → *Information theory*;

Additional Key Words and Phrases: trace complexity, self-adjusting networks, entropy rate, compress, complexity map, data centers

ACM Reference Format:
Chen Avin, Manya Ghobadi, Chen Griner, and Stefan Schmid. 2020. On the Complexity of Traffic Traces and Implications. *Proc. ACM Meas. Anal. Comput. Syst.* 4, 1, Article 20 (March 2020), 29 pages. https://doi.org/10.1145/3379486

1 INTRODUCTION

Packet traces collected from networking applications, such as datacenter traffic, have been shown to feature much *structure*: datacenter traffic matrices are sparse and skewed [16, 39], exhibit

Question 2:

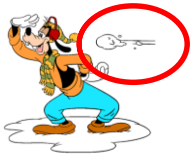
Given This Structure, What Can Be Achieved? Metrics and Algorithms?

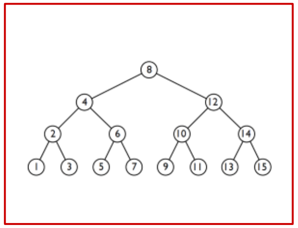
A first insight: entropy of the demand.

Insight:

Connection to Datastructures

Traditional BST

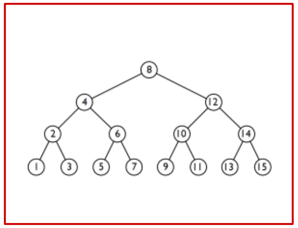




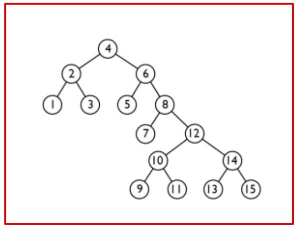
Insight:

Connection to Datastructures

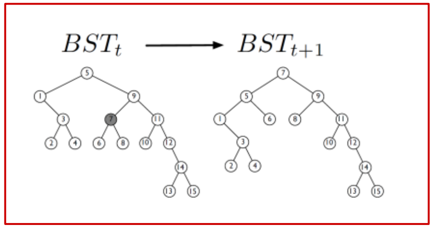
Traditional BST



Demand-aware BST



Self-adjusting BST

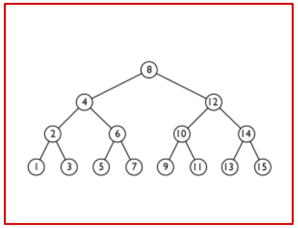


More structure: improved **access cost**

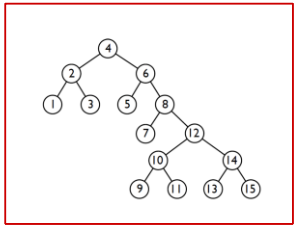
Insight:

Connection to Datastructures & Coding

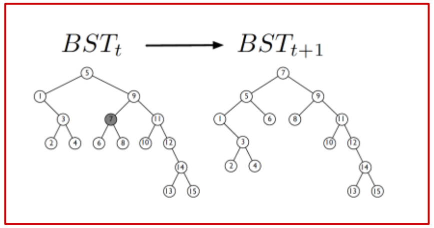
Traditional BST
(Worst-case coding)



Demand-aware BST
(Huffman coding)



Self-adjusting BST
(Dynamic Huffman coding)

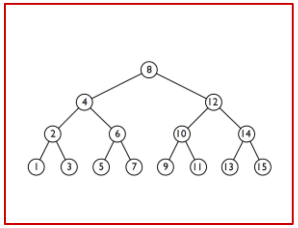


More structure: improved **access cost** / shorter **codes**

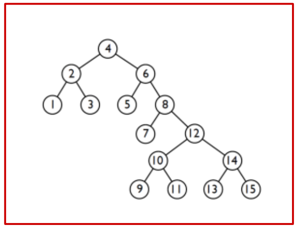
Insight:

Connection to Datastructures & Coding

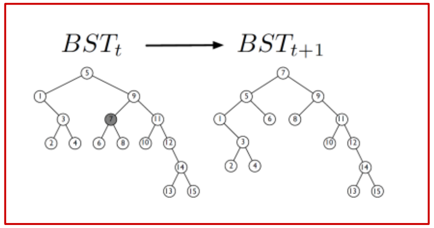
Traditional BST
(Worst-case coding)



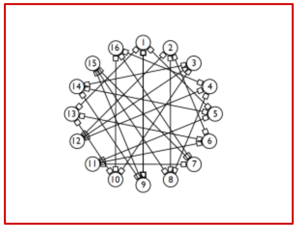
Demand-aware BST
(Huffman coding)

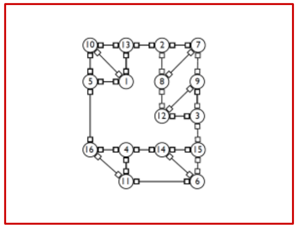


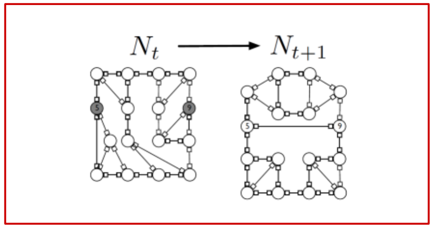
Self-adjusting BST
(Dynamic Huffman coding)



More structure: improved **access cost** / shorter **codes**







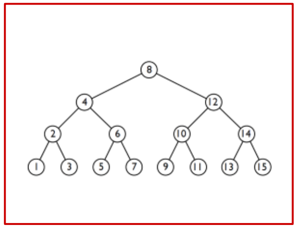
Similar **benefits**?

Insight:

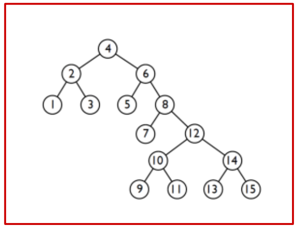
Connection to Datastructures & Coding

More than
an analogy!

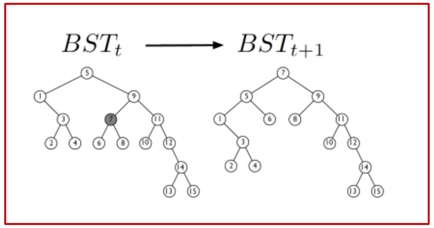
Traditional BST
(Worst-case coding)



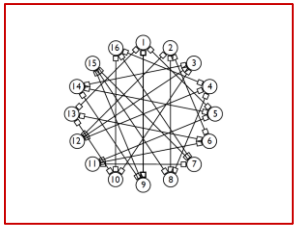
Demand-aware BST
(Huffman coding)

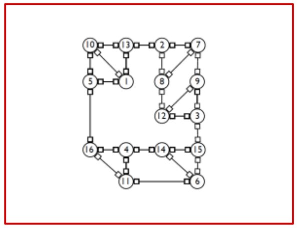


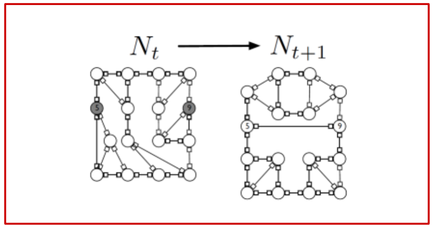
Self-adjusting BST
(Dynamic Huffman coding)



More structure: improved **access cost** / shorter **codes**



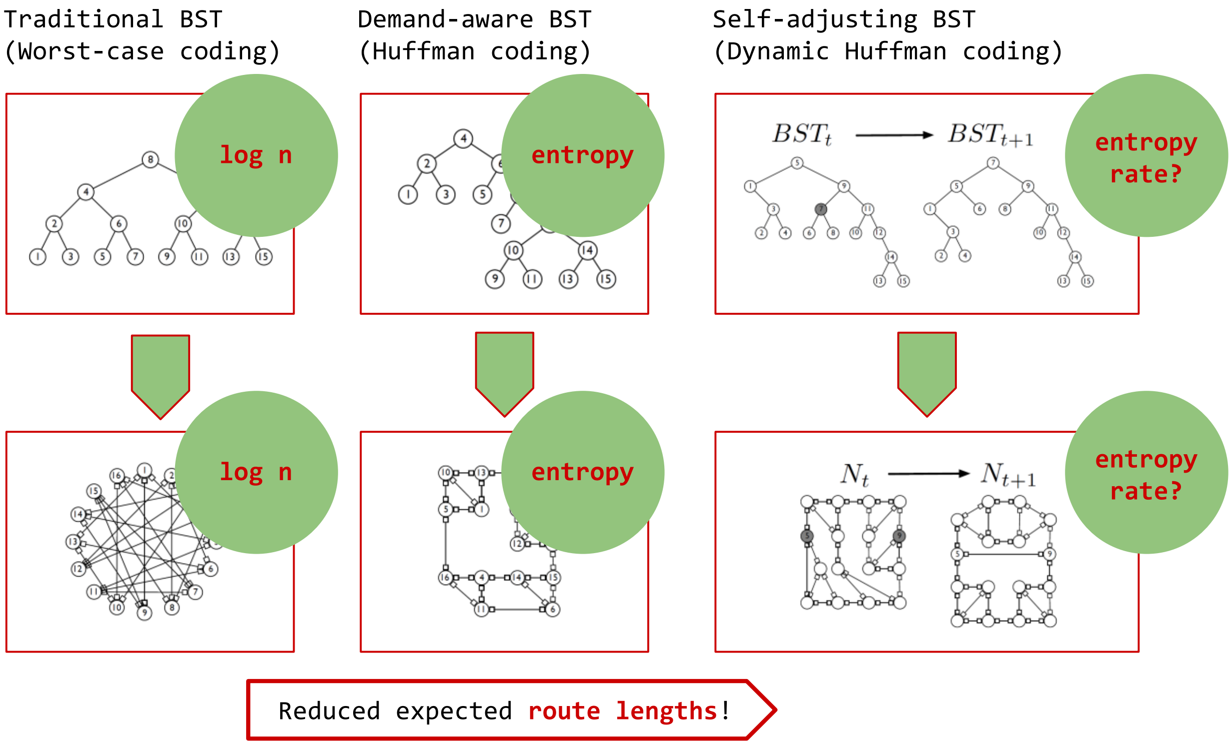




Similar **benefits**?

Insight:

Connection to Datastructures & Coding



Traditional BST
(Worst-case coding)

log n

log n

Demand-aware BST
(Huffman coding)

entropy

entropy

Self-adjusting BST
(Dynamic Huffman coding)

entropy rate?

entropy rate?

Reduced expected **route lengths**!

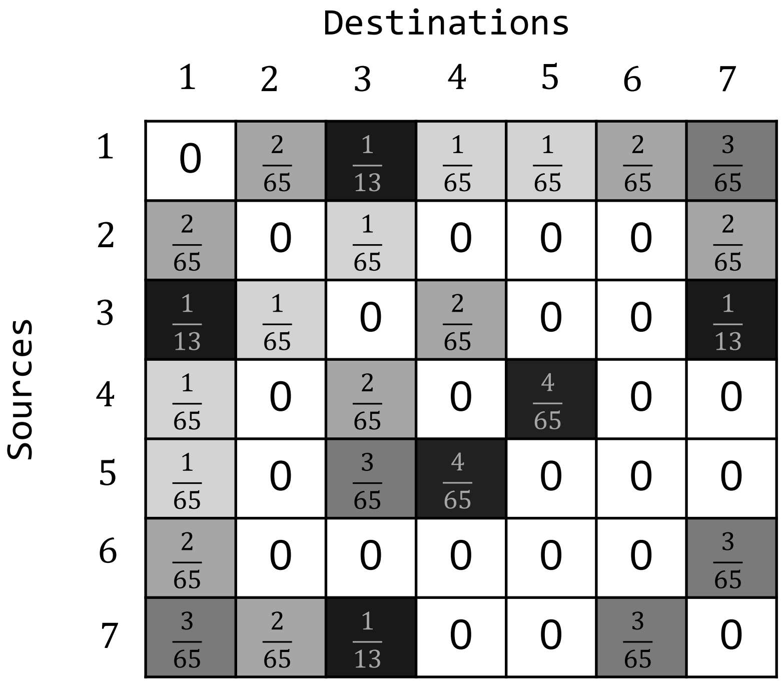
More than
an analogy!

Generalize methodology:
... and transfer entropy bounds and algorithms of data-structures to networks.

First result:
Demand-aware networks of asymptotically optimal route lengths.

Case Study “Route Lengths”

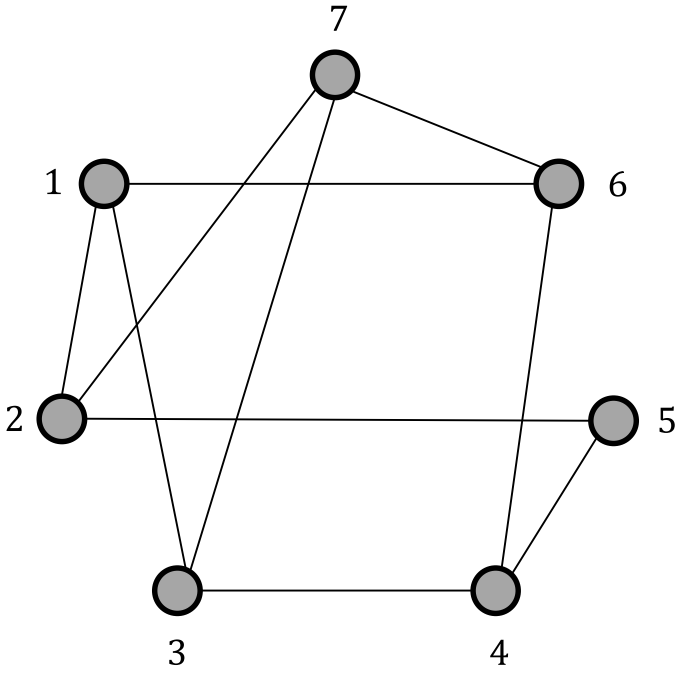
Constant-Degree Demand-Aware Network



Destinations

Sources

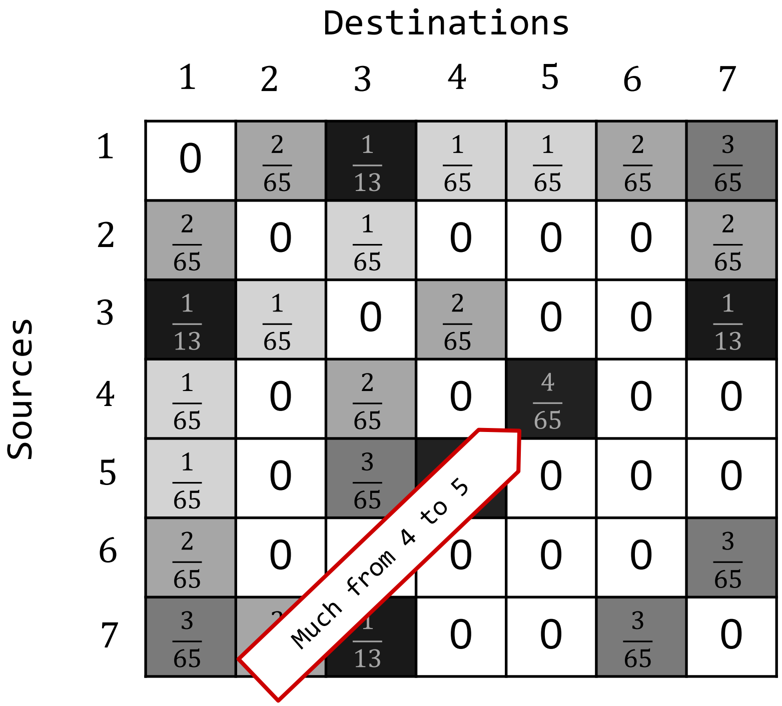
	1	2	3	4	5	6	7
1	0	$\frac{2}{65}$	$\frac{1}{13}$	$\frac{1}{65}$	$\frac{1}{65}$	$\frac{2}{65}$	$\frac{3}{65}$
2	$\frac{2}{65}$	0	$\frac{1}{65}$	0	0	0	$\frac{2}{65}$
3	$\frac{1}{13}$	$\frac{1}{65}$	0	$\frac{2}{65}$	0	0	$\frac{1}{13}$
4	$\frac{1}{65}$	0	$\frac{2}{65}$	0	$\frac{4}{65}$	0	0
5	$\frac{1}{65}$	0	$\frac{3}{65}$	$\frac{4}{65}$	0	0	0
6	$\frac{2}{65}$	0	0	0	0	0	$\frac{3}{65}$
7	$\frac{3}{65}$	$\frac{2}{65}$	$\frac{1}{13}$	0	0	$\frac{3}{65}$	0



$$\mathrm{ERL}(\mathcal{D},N) = \sum_{(u,v)\in\mathcal{D}} p(u,v) \cdot d_N(u,v)$$

Case Study “Route Lengths”

Constant-Degree Demand-Aware Network

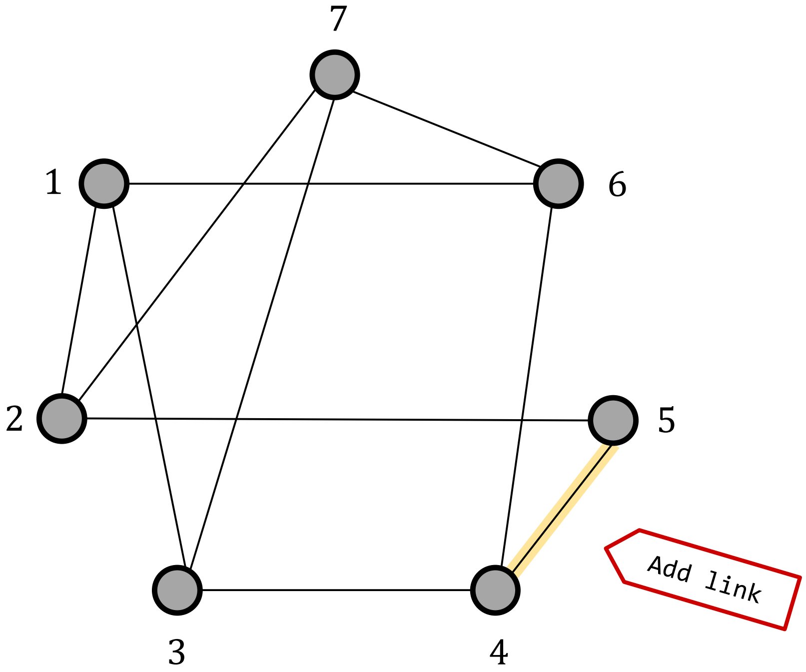


Destinations

Sources

	1	2	3	4	5	6	7
1	0	$\frac{2}{65}$	$\frac{1}{13}$	$\frac{1}{65}$	$\frac{1}{65}$	$\frac{2}{65}$	$\frac{3}{65}$
2	$\frac{2}{65}$	0	$\frac{1}{65}$	0	0	0	$\frac{2}{65}$
3	$\frac{1}{13}$	$\frac{1}{65}$	0	$\frac{2}{65}$	0	0	$\frac{1}{13}$
4	$\frac{1}{65}$	0	$\frac{2}{65}$	0	$\frac{4}{65}$	0	0
5	$\frac{1}{65}$	0	$\frac{3}{65}$	[illegible]	0	0	0
6	$\frac{2}{65}$	0	[illegible]	0	0	0	$\frac{3}{65}$
7	$\frac{3}{65}$	[illegible]	$\frac{1}{13}$	0	0	$\frac{3}{65}$	0

Much from 4 to 5



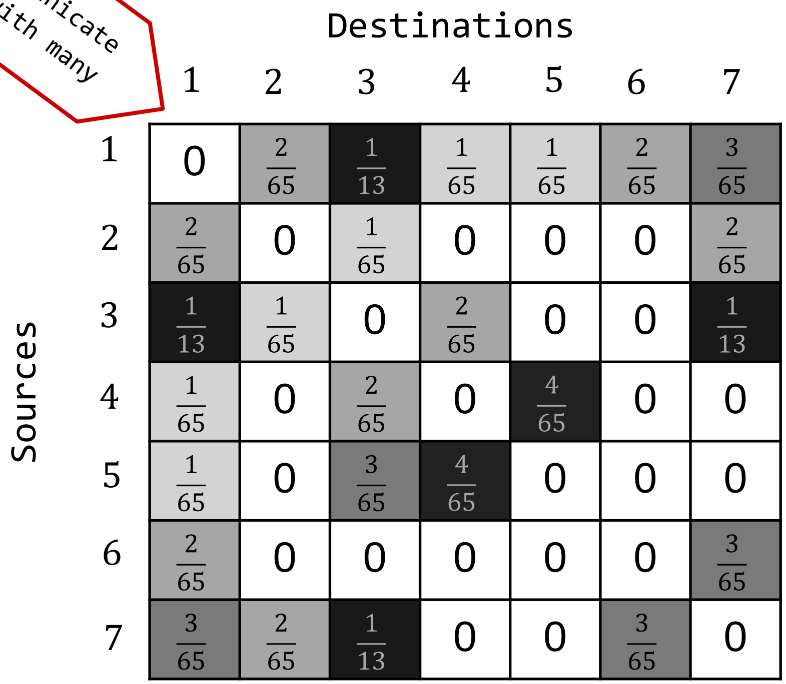
Add link

$$\mathrm{ERL}(\mathcal{D},\mathrm{N}) = \sum_{(u,v)\in\mathcal{D}} p(u,v)\cdot d_{N}(u,v)$$

Case Study “Route Lengths”

Constant-Degree Demand-Aware Network

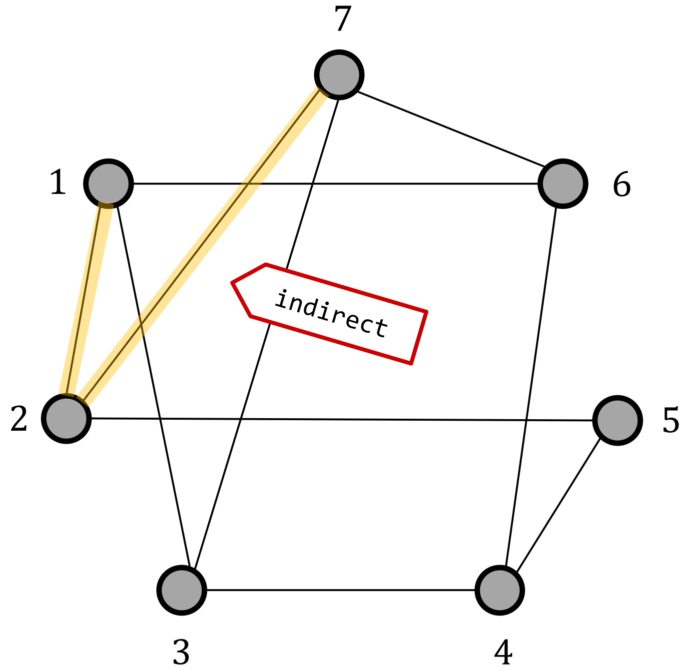
Communicated with many



Destinations

Sources

	1	2	3	4	5	6	7
1	0	$\frac{2}{65}$	$\frac{1}{13}$	$\frac{1}{65}$	$\frac{1}{65}$	$\frac{2}{65}$	$\frac{3}{65}$
2	$\frac{2}{65}$	0	$\frac{1}{65}$	0	0	0	$\frac{2}{65}$
3	$\frac{1}{13}$	$\frac{1}{65}$	0	$\frac{2}{65}$	0	0	$\frac{1}{13}$
4	$\frac{1}{65}$	0	$\frac{2}{65}$	0	$\frac{4}{65}$	0	0
5	$\frac{1}{65}$	0	$\frac{3}{65}$	$\frac{4}{65}$	0	0	0
6	$\frac{2}{65}$	0	0	0	0	0	$\frac{3}{65}$
7	$\frac{3}{65}$	$\frac{2}{65}$	$\frac{1}{13}$	0	0	$\frac{3}{65}$	0

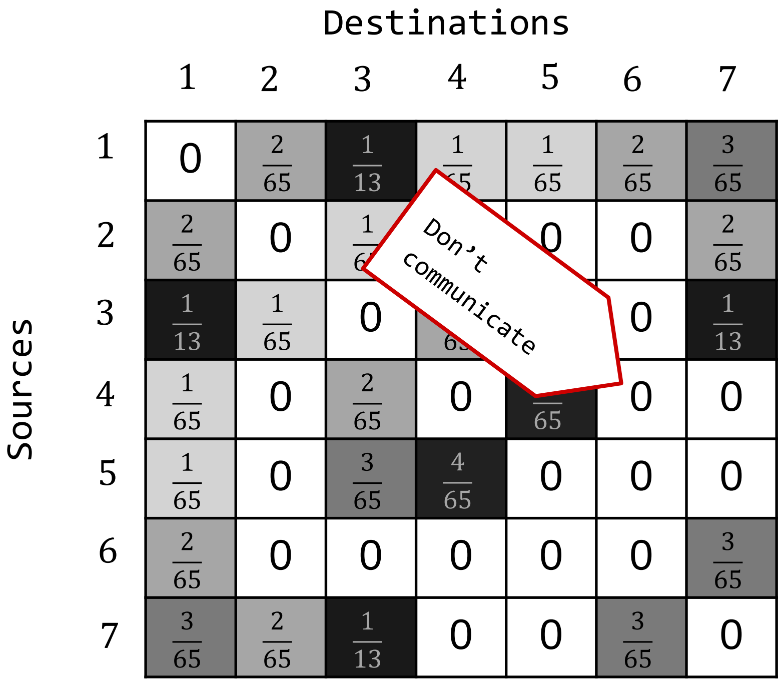


indirect

$$ERL(\mathcal{D},N) = \sum_{(u,v)\in\mathcal{D}} p(u,v) \cdot d_N(u,v)$$

Case Study “Route Lengths”

Constant-Degree Demand-Aware Network

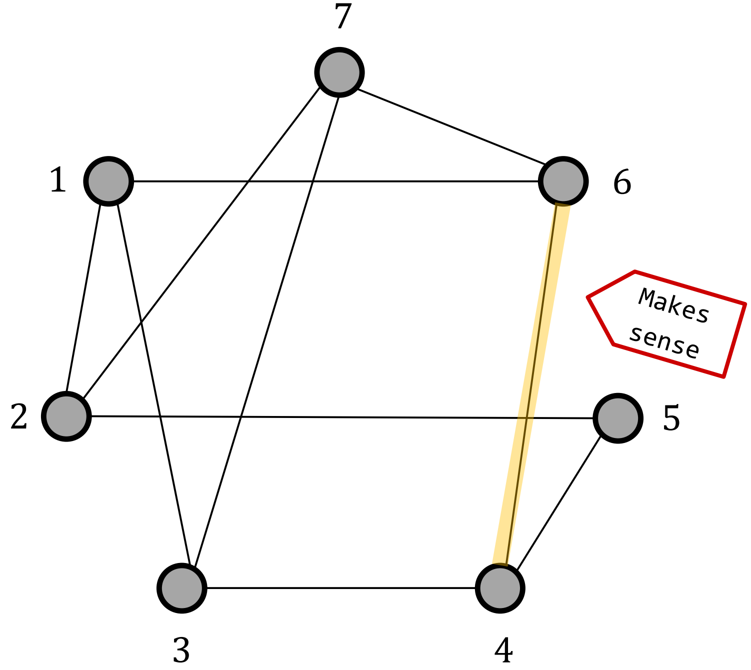


Destinations

Sources

	1	2	3	4	5	6	7
1	0	$\frac{2}{65}$	$\frac{1}{13}$	$\frac{1}{65}$	$\frac{1}{65}$	$\frac{2}{65}$	$\frac{3}{65}$
2	$\frac{2}{65}$	0	$\frac{1}{65}$	[illegible]	0	0	$\frac{2}{65}$
3	$\frac{1}{13}$	$\frac{1}{65}$	0	[illegible]	[illegible]	0	$\frac{1}{13}$
4	$\frac{1}{65}$	0	$\frac{2}{65}$	0	[illegible]	0	0
5	$\frac{1}{65}$	0	$\frac{3}{65}$	$\frac{4}{65}$	0	0	0
6	$\frac{2}{65}$	0	0	0	0	0	$\frac{3}{65}$
7	$\frac{3}{65}$	$\frac{2}{65}$	$\frac{1}{13}$	0	0	$\frac{3}{65}$	0

Don’t communicate

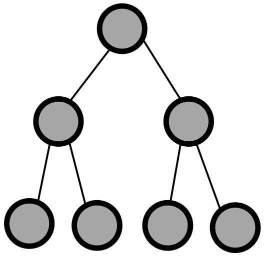


Makes sense

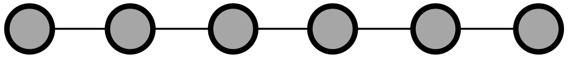
$$\mathrm{ERL}(\mathcal{D},\mathrm{N}) = \sum_{(u,v)\in\mathcal{D}} p(u,v) \cdot d_N(u,v)$$

Examples

- DAN for △=3
  - E.g., complete **binary tree** would be **log n**
  - Can we do better?

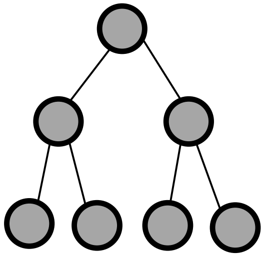


- DAN for △=2
  - Set of **lines** and **cycles**

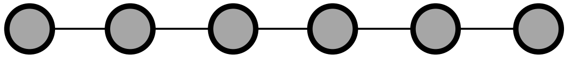


Examples

- DAN for △=3
  - E.g., complete **binary tree** would be **log n**
  - Can we do better?



- DAN for △=2
  - Set of **lines** and **cycles**



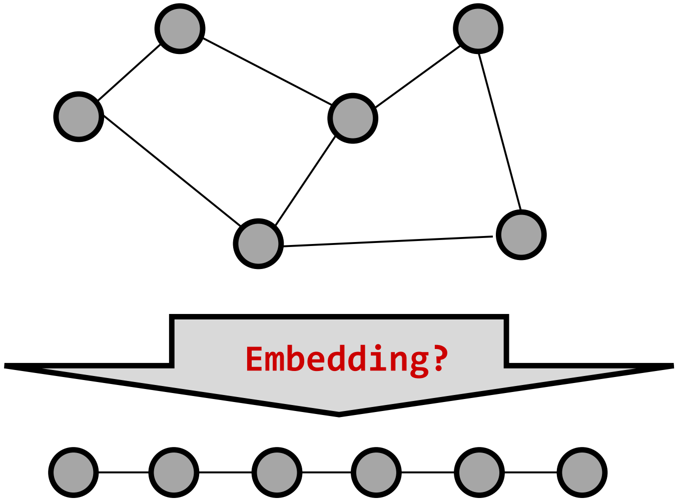
How hard?

Related Problem

Virtual Network Embedding Problem (VNEP)

Example △=2: A Minium Linear Arrangement (**MLA**) Problem

→ Minimizes sum of virtual edges

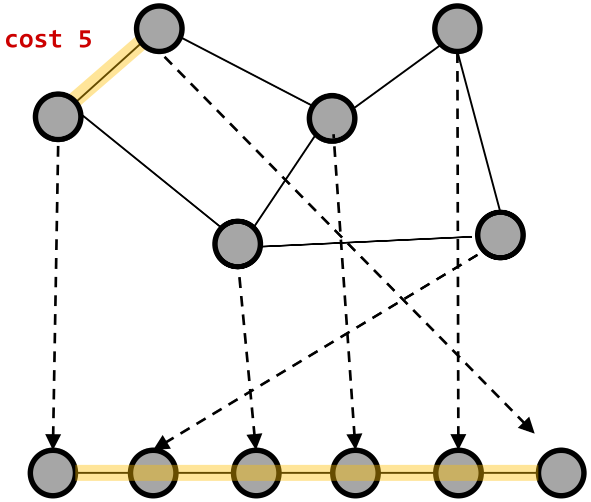


Related Problem

Virtual Network Embedding Problem (VNEP)

Example △=2: A Minium Linear Arrangement (**MLA**) Problem

→ Minimizes sum of virtual edges



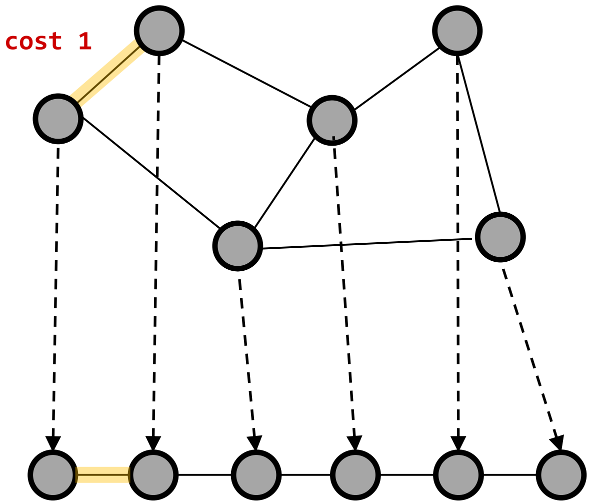
Bad!

Related Problem

Virtual Network Embedding Problem (VNEP)

Example △=2: A Minium Linear Arrangement (**MLA**) Problem

→ Minimizes sum of virtual edges



Good!

Related Problem

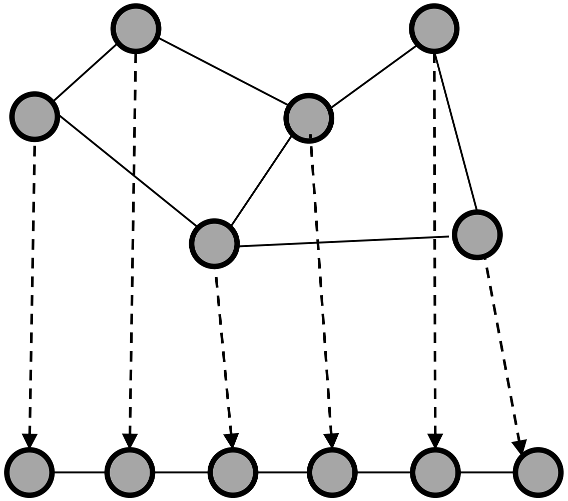
Virtual Network Embedding Problem (VNEP)

Example △=2: A Minium Linear Arrangement (**MLA**) Problem

⇀ Minimizes sum of virtual edges

MLA is **NP-hard**

⇀ … and so is our problem!



Related Problem

Virtual Network Embedding Problem (VNEP)

Example △=2: A Minium Linear Arrangement (**MLA**) Problem

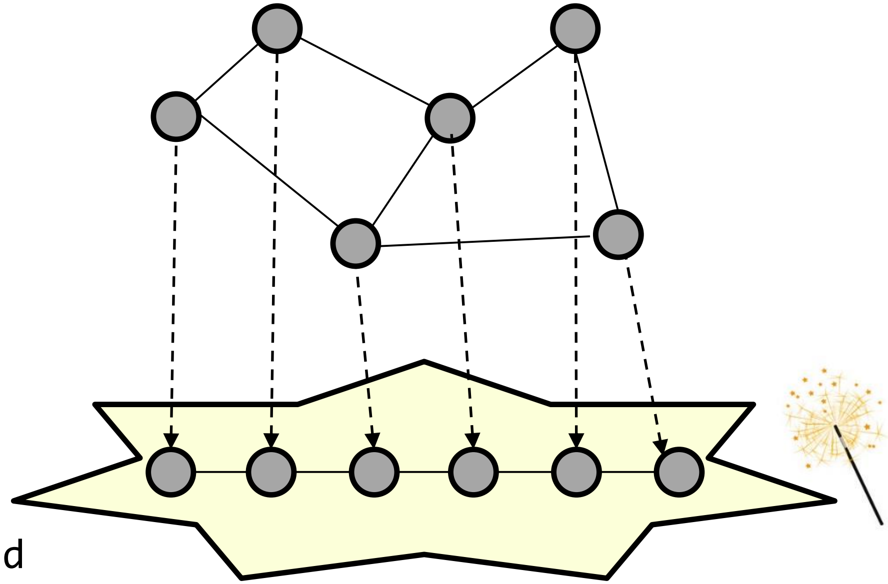
- ⇀ Minimizes sum of virtual edges

MLA is **NP-hard**

- ⇀ … and so is our problem!

But what about **△>2**?

- ⇀ Embedding problem still hard
- ⇀ But we have a new **degree of freedom**!



Related Problem

Virtual Network Embedding Problem (VNEP)

Example △=2: A Minium Linear Arrangement (**MLA**) Problem

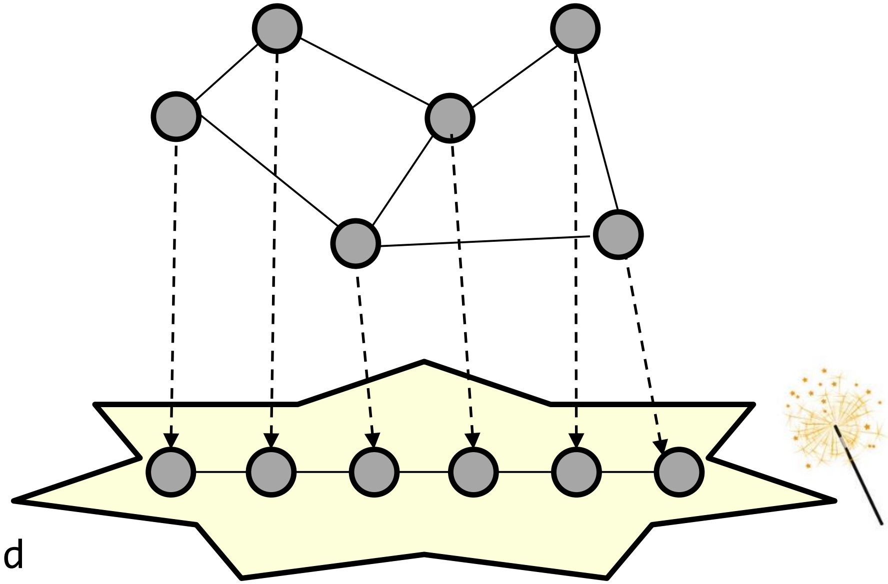
- ⇀ Minimizes sum of virtual edges

MLA is **NP-hard**

- ⇀ … and so is our problem!

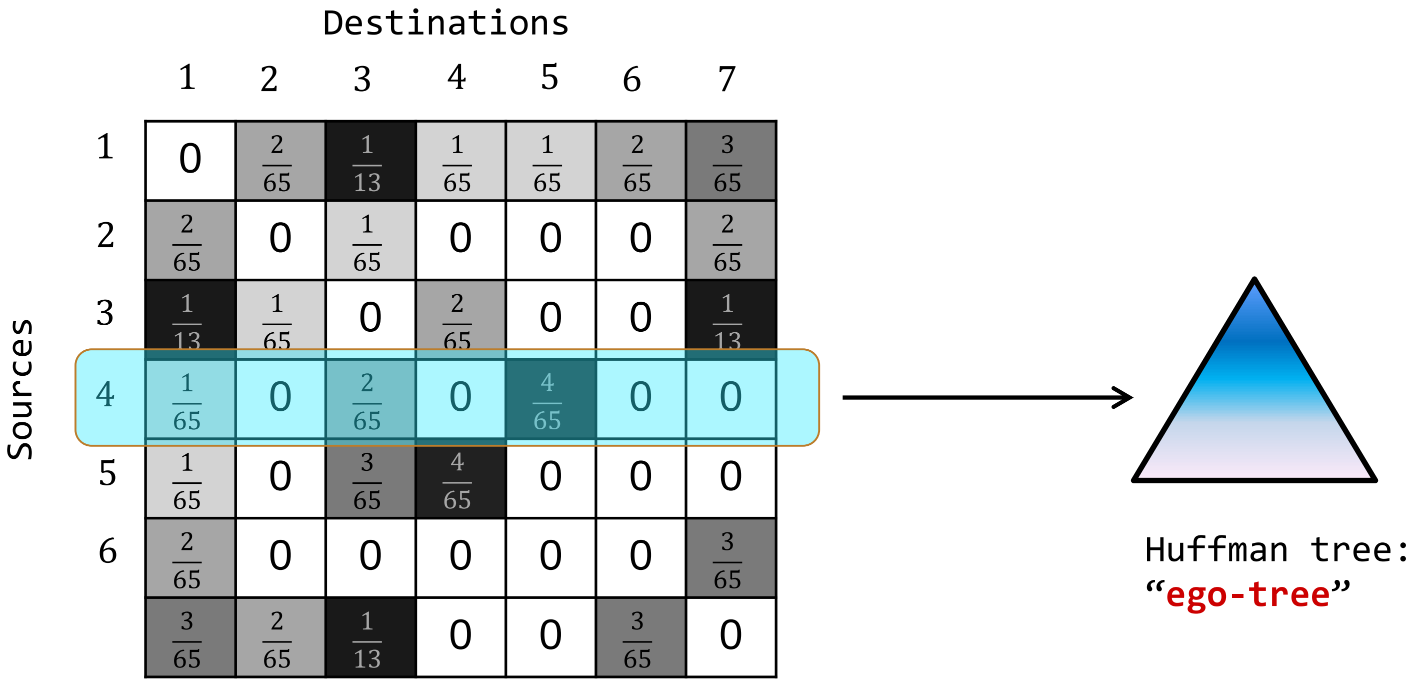
But what about **△>2**?

- ⇀ Embedding problem still hard
- ⇀ But we have a new **degree of freedom**!



Simplifies problem?!

Algorithm: Idea



Destinations

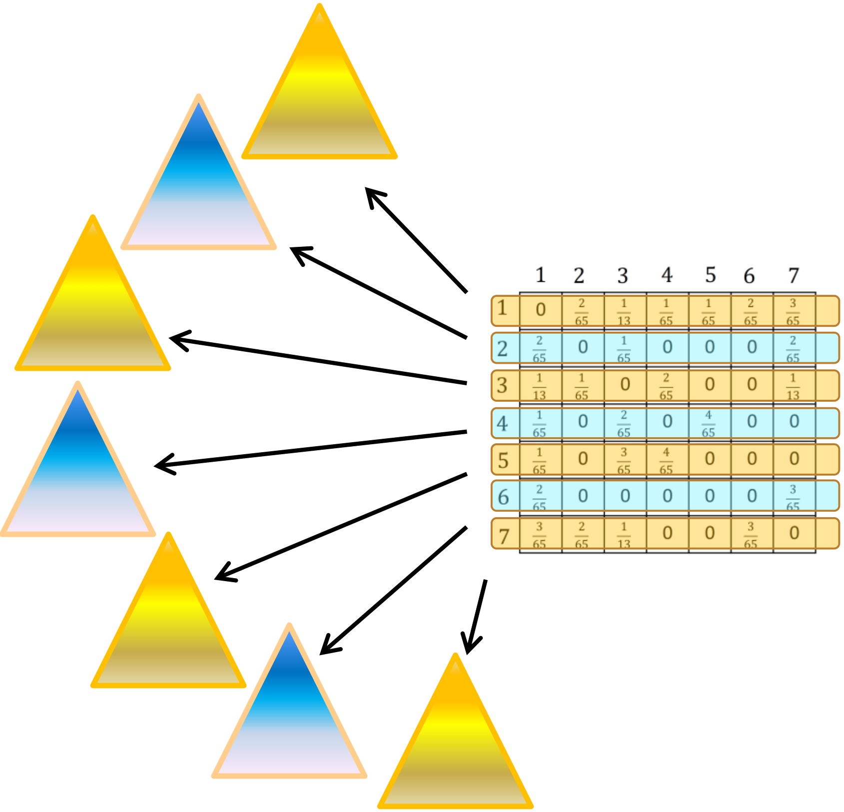
Sources

	1	2	3	4	5	6	7
1	0	$\frac{2}{65}$	$\frac{1}{13}$	$\frac{1}{65}$	$\frac{1}{65}$	$\frac{2}{65}$	$\frac{3}{65}$
2	$\frac{2}{65}$	0	$\frac{1}{65}$	0	0	0	$\frac{2}{65}$
3	$\frac{1}{13}$	$\frac{1}{65}$	0	$\frac{2}{65}$	0	0	$\frac{1}{13}$
4	$\frac{1}{65}$	0	$\frac{2}{65}$	0	$\frac{4}{65}$	0	0
5	$\frac{1}{65}$	0	$\frac{3}{65}$	$\frac{4}{65}$	0	0	0
6	$\frac{2}{65}$	0	0	0	0	0	$\frac{3}{65}$
	$\frac{3}{65}$	$\frac{2}{65}$	$\frac{1}{13}$	0	0	$\frac{3}{65}$	0

Huffman tree:
“**ego-tree**”

Entropy Upper Bound

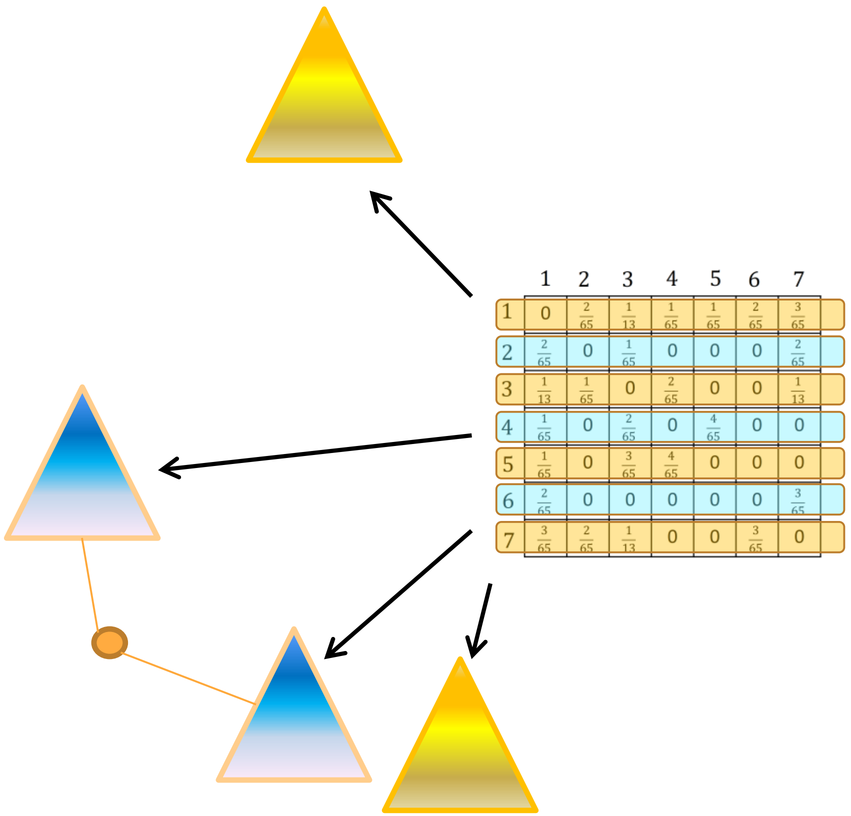
- Idea for algorithm:
  - union of trees



	1	2	3	4	5	6	7
1	0	$\frac{2}{65}$	$\frac{1}{13}$	$\frac{1}{65}$	$\frac{1}{65}$	$\frac{2}{65}$	$\frac{3}{65}$
2	$\frac{2}{65}$	0	$\frac{1}{65}$	0	0	0	$\frac{2}{65}$
3	$\frac{1}{13}$	$\frac{1}{65}$	0	$\frac{2}{65}$	0	0	$\frac{1}{13}$
4	$\frac{1}{65}$	0	$\frac{2}{65}$	0	$\frac{4}{65}$	0	0
5	$\frac{1}{65}$	0	$\frac{3}{65}$	$\frac{4}{65}$	0	0	0
6	$\frac{2}{65}$	0	0	0	0	0	$\frac{3}{65}$
7	$\frac{3}{65}$	$\frac{2}{65}$	$\frac{1}{13}$	0	0	$\frac{3}{65}$	0

Entropy Upper Bound

- Idea for algorithm:
  - union of trees
  - reduce degree
  - but keep distances

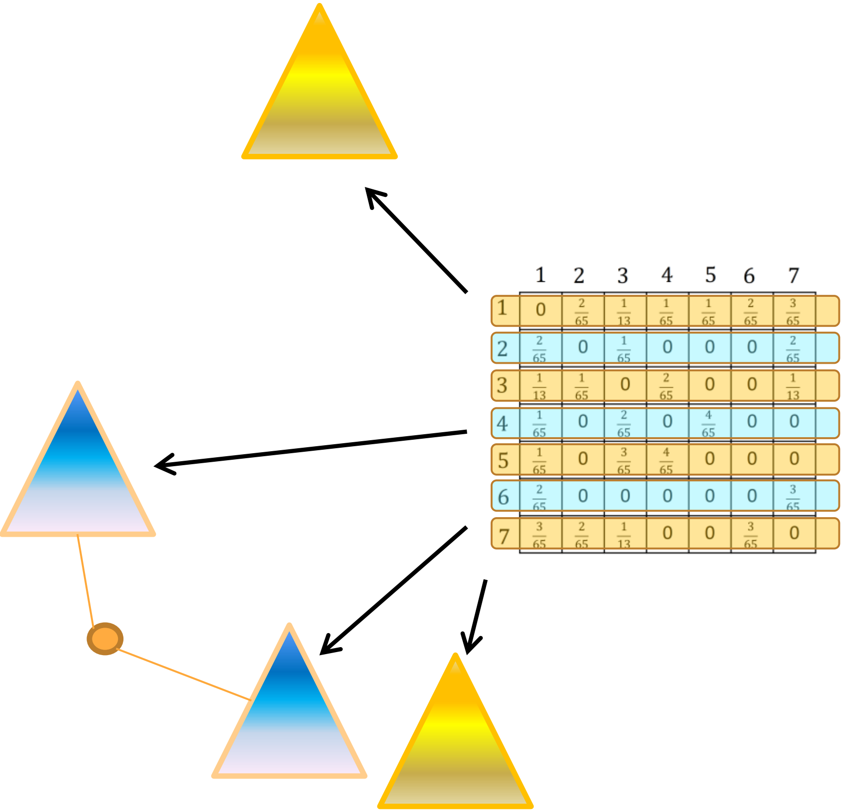


	1	2	3	4	5	6	7
1	0	$\frac{2}{65}$	$\frac{1}{13}$	$\frac{1}{65}$	$\frac{1}{65}$	$\frac{2}{65}$	$\frac{3}{65}$
2	$\frac{2}{65}$	0	$\frac{1}{65}$	0	0	0	$\frac{2}{65}$
3	$\frac{1}{13}$	$\frac{1}{65}$	0	$\frac{2}{65}$	0	0	$\frac{1}{13}$
4	$\frac{1}{65}$	0	$\frac{2}{65}$	0	$\frac{4}{65}$	0	0
5	$\frac{1}{65}$	0	$\frac{3}{65}$	$\frac{4}{65}$	0	0	0
6	$\frac{2}{65}$	0	0	0	0	0	$\frac{3}{65}$
7	$\frac{3}{65}$	$\frac{2}{65}$	$\frac{1}{13}$	0	0	$\frac{3}{65}$	0

Entropy Upper Bound

- Idea for algorithm:
  - union of trees
  - reduce degree
  - but keep distances

- Ok for sparse demands
  - not everyone gets tree
  - helper nodes

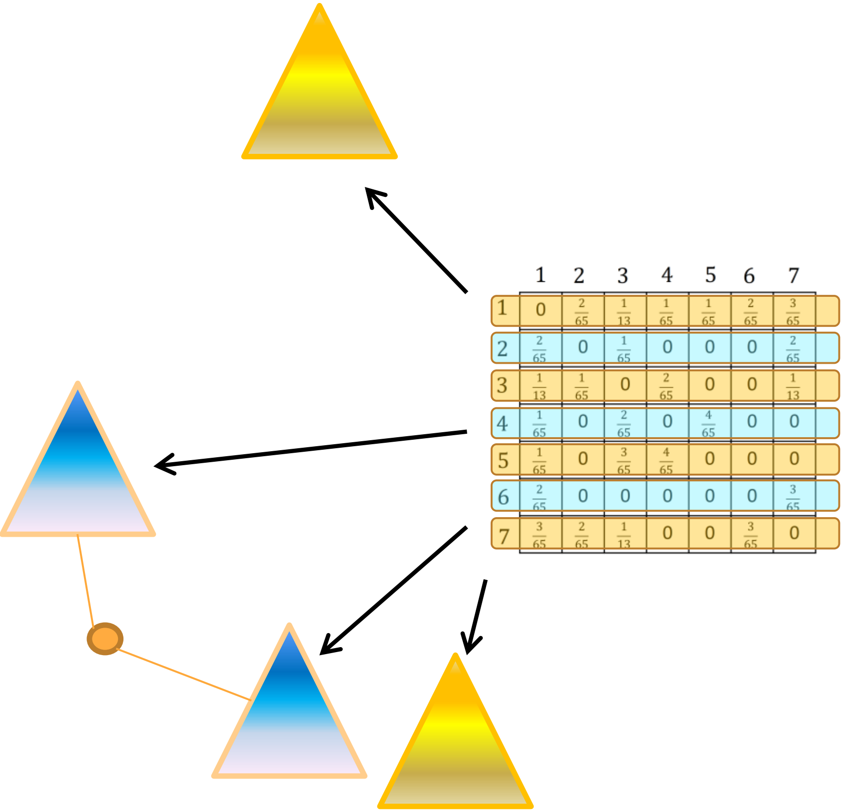


	1	2	3	4	5	6	7
1	0	$\frac{2}{65}$	$\frac{1}{13}$	$\frac{1}{65}$	$\frac{1}{65}$	$\frac{2}{65}$	$\frac{3}{65}$
2	$\frac{2}{65}$	0	$\frac{1}{65}$	0	0	0	$\frac{2}{65}$
3	$\frac{1}{13}$	$\frac{1}{65}$	0	$\frac{2}{65}$	0	0	$\frac{1}{13}$
4	$\frac{1}{65}$	0	$\frac{2}{65}$	0	$\frac{4}{65}$	0	0
5	$\frac{1}{65}$	0	$\frac{3}{65}$	$\frac{4}{65}$	0	0	0
6	$\frac{2}{65}$	0	0	0	0	0	$\frac{3}{65}$
7	$\frac{3}{65}$	$\frac{2}{65}$	$\frac{1}{13}$	0	0	$\frac{3}{65}$	0

Entropy Upper Bound

- Idea for algorithm:
  - union of trees
  - reduce degree
  - but keep distances
- Ok for sparse demands
  - not everyone gets tree
  - helper nodes

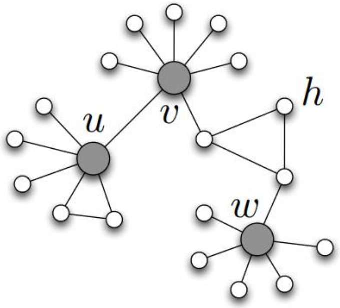
Static



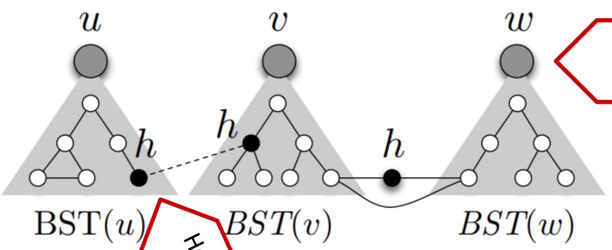
	1	2	3	4	5	6	7
1	0	$\frac{2}{65}$	$\frac{1}{13}$	$\frac{1}{65}$	$\frac{1}{65}$	$\frac{2}{65}$	$\frac{3}{65}$
2	$\frac{2}{65}$	0	$\frac{1}{65}$	0	0	0	$\frac{2}{65}$
3	$\frac{1}{13}$	$\frac{1}{65}$	0	$\frac{2}{65}$	0	0	$\frac{1}{13}$
4	$\frac{1}{65}$	0	$\frac{2}{65}$	0	$\frac{4}{65}$	0	0
5	$\frac{1}{65}$	0	$\frac{3}{65}$	$\frac{4}{65}$	0	0	0
6	$\frac{2}{65}$	0	0	0	0	0	$\frac{3}{65}$
7	$\frac{3}{65}$	$\frac{2}{65}$	$\frac{1}{13}$	0	0	$\frac{3}{65}$	0

Intuition of Algorithm

Demand graph:



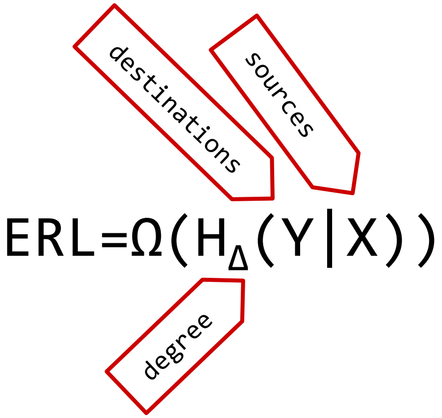
Demand-aware network:



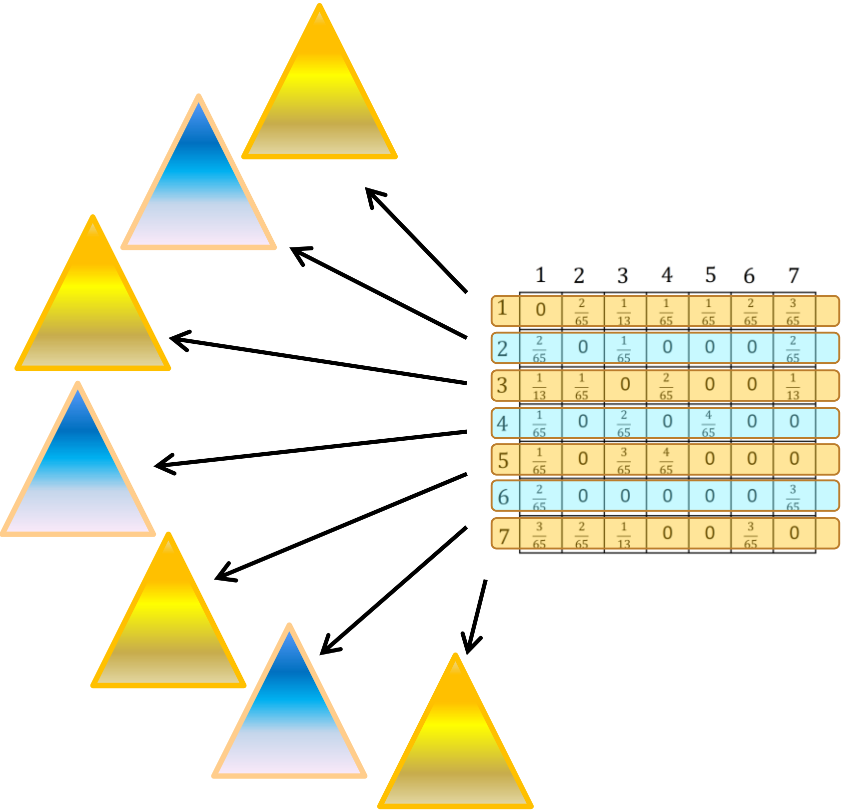
Ego-trees for large nodes

Helper node

Entropy Lower Bound



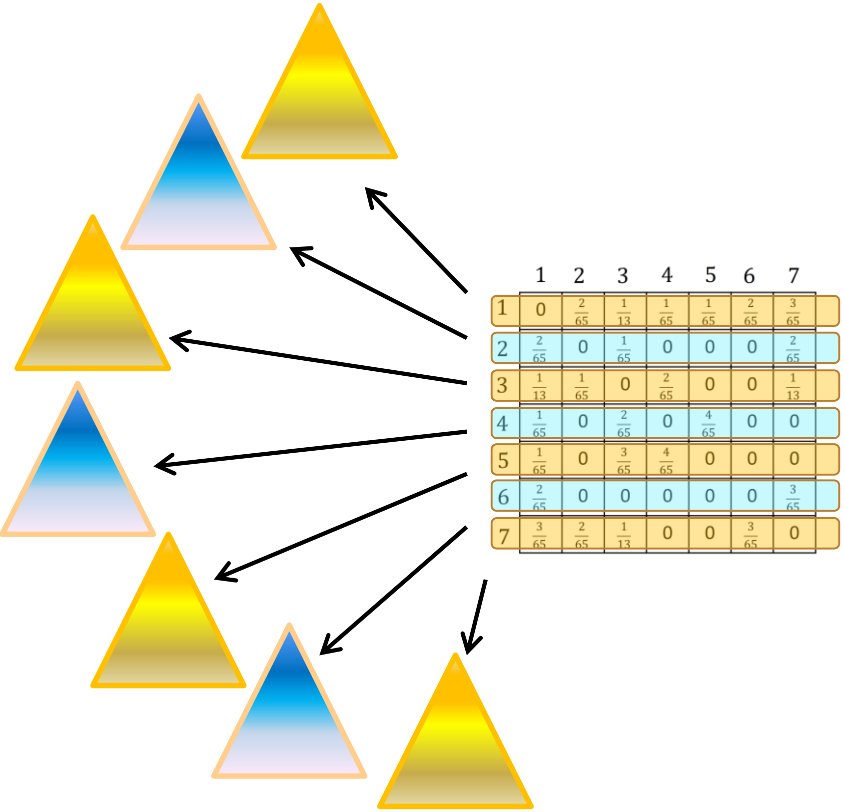
$ERL=\Omega(H_\Delta(Y|X))$



	1	2	3	4	5	6	7
1	0	$\frac{2}{65}$	$\frac{1}{13}$	$\frac{1}{65}$	$\frac{1}{65}$	$\frac{2}{65}$	$\frac{3}{65}$
2	$\frac{2}{65}$	0	$\frac{1}{65}$	0	0	0	$\frac{2}{65}$
3	$\frac{1}{13}$	$\frac{1}{65}$	0	$\frac{2}{65}$	0	0	$\frac{1}{13}$
4	$\frac{1}{65}$	0	$\frac{2}{65}$	0	$\frac{4}{65}$	0	0
5	$\frac{1}{65}$	0	$\frac{3}{65}$	$\frac{4}{65}$	0	0	0
6	$\frac{2}{65}$	0	0	0	0	0	$\frac{3}{65}$
7	$\frac{3}{65}$	$\frac{2}{65}$	$\frac{1}{13}$	0	0	$\frac{3}{65}$	0

Dynamic Algorithm

Dynamic



	1	2	3	4	5	6	7
1	0	$\frac{2}{65}$	$\frac{1}{13}$	$\frac{1}{65}$	$\frac{1}{65}$	$\frac{2}{65}$	$\frac{3}{65}$
2	$\frac{2}{65}$	0	$\frac{1}{65}$	0	0	0	$\frac{2}{65}$
3	$\frac{1}{13}$	$\frac{1}{65}$	0	$\frac{2}{65}$	0	0	$\frac{1}{13}$
4	$\frac{1}{65}$	0	$\frac{2}{65}$	0	$\frac{4}{65}$	0	0
5	$\frac{1}{65}$	0	$\frac{3}{65}$	$\frac{4}{65}$	0	0	0
6	$\frac{2}{65}$	0	0	0	0	0	$\frac{3}{65}$
7	$\frac{3}{65}$	$\frac{2}{65}$	$\frac{1}{13}$	0	0	$\frac{3}{65}$	0

Much Research Ahead

- So far: tip of the iceberg

- Many challenges
  - Demand-aware graphs for *dense matrices*?
  - *Online* competitive algorithms?
  - *Scalable control* (e.g., Google Jupiter)
  - Impact on *routing*, *congestion control*, buffer management?

Thank you! Questions?

Reality: Tech Diversity

Diverse topology components:

⇀ demand-oblivious and
demand-aware

Demand-oblivious

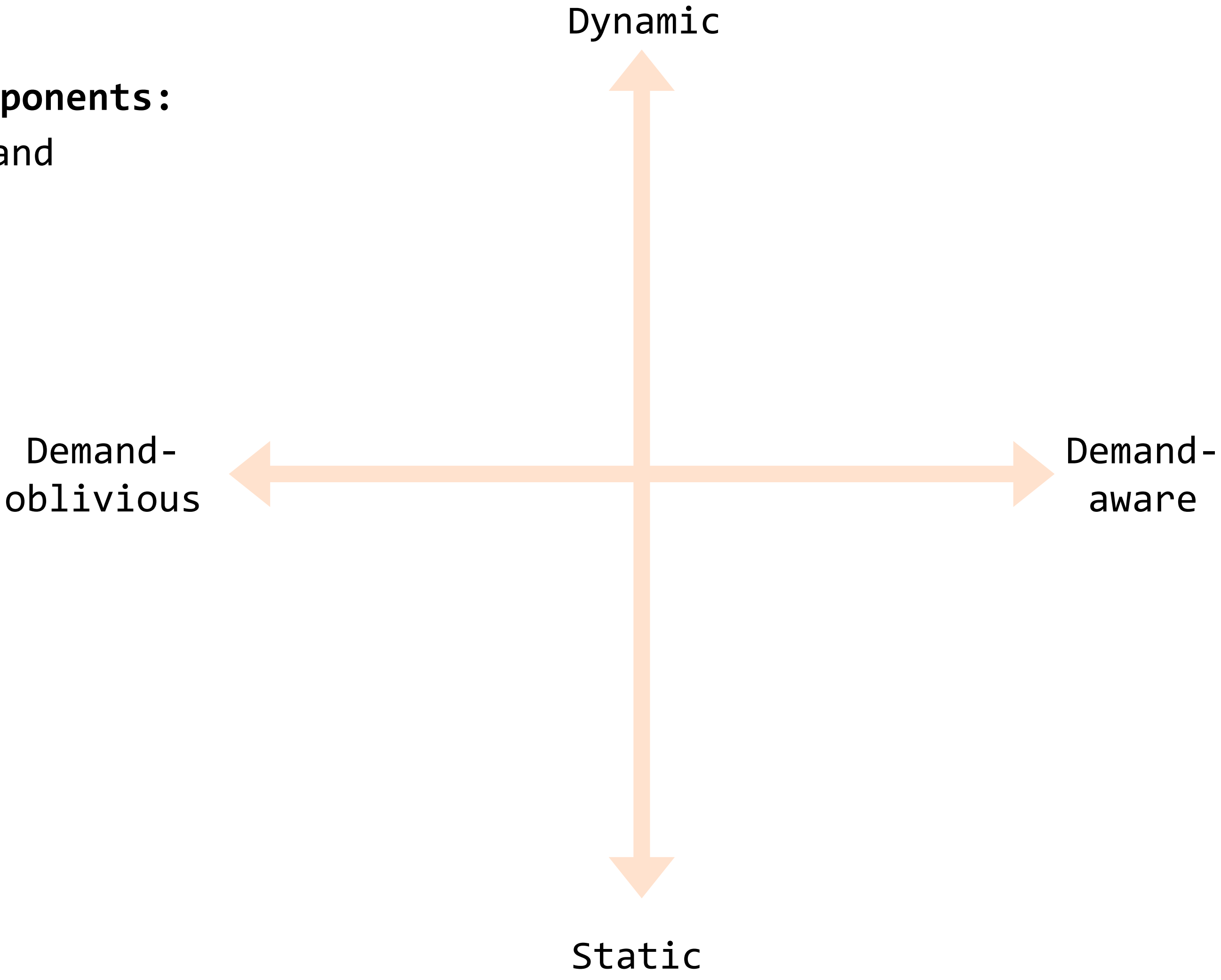
Demand-aware

Reality: Tech Diversity

Diverse topology components:

⇀ demand-oblivious and
demand-aware

⇀ static vs dynamic

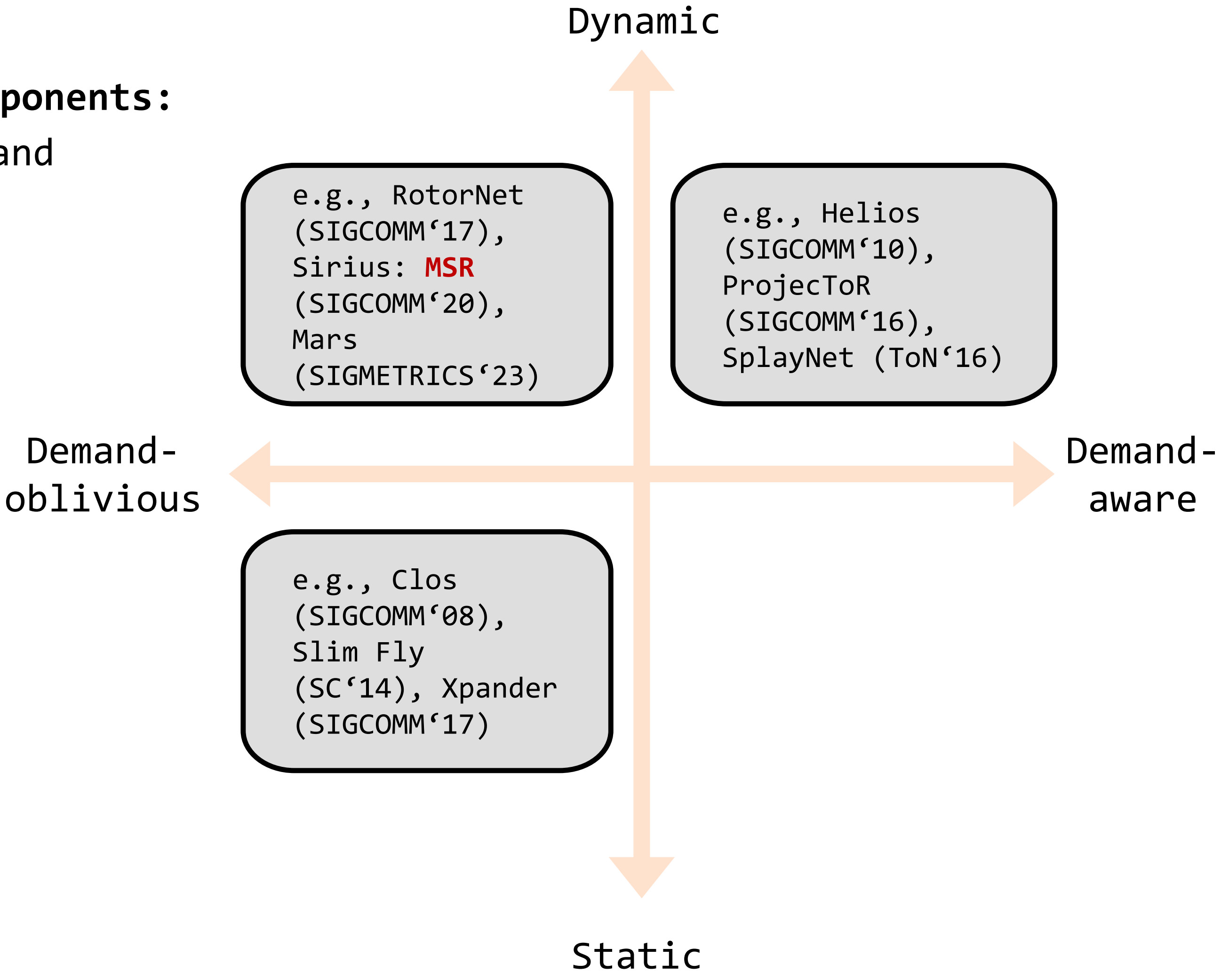


Reality: Tech Diversity

Diverse topology components:

⇀ demand-oblivious and
demand-aware

⇀ static vs dynamic



Dynamic

e.g., RotorNet (SIGCOMM‘17), Sirius: **MSR** (SIGCOMM‘20), Mars (SIGMETRICS‘23)

e.g., Helios (SIGCOMM‘10), ProjecToR (SIGCOMM‘16), SplayNet (ToN‘16)

Demand-oblivious

Demand-aware

e.g., Clos (SIGCOMM‘08), Slim Fly (SC‘14), Xpander (SIGCOMM‘17)

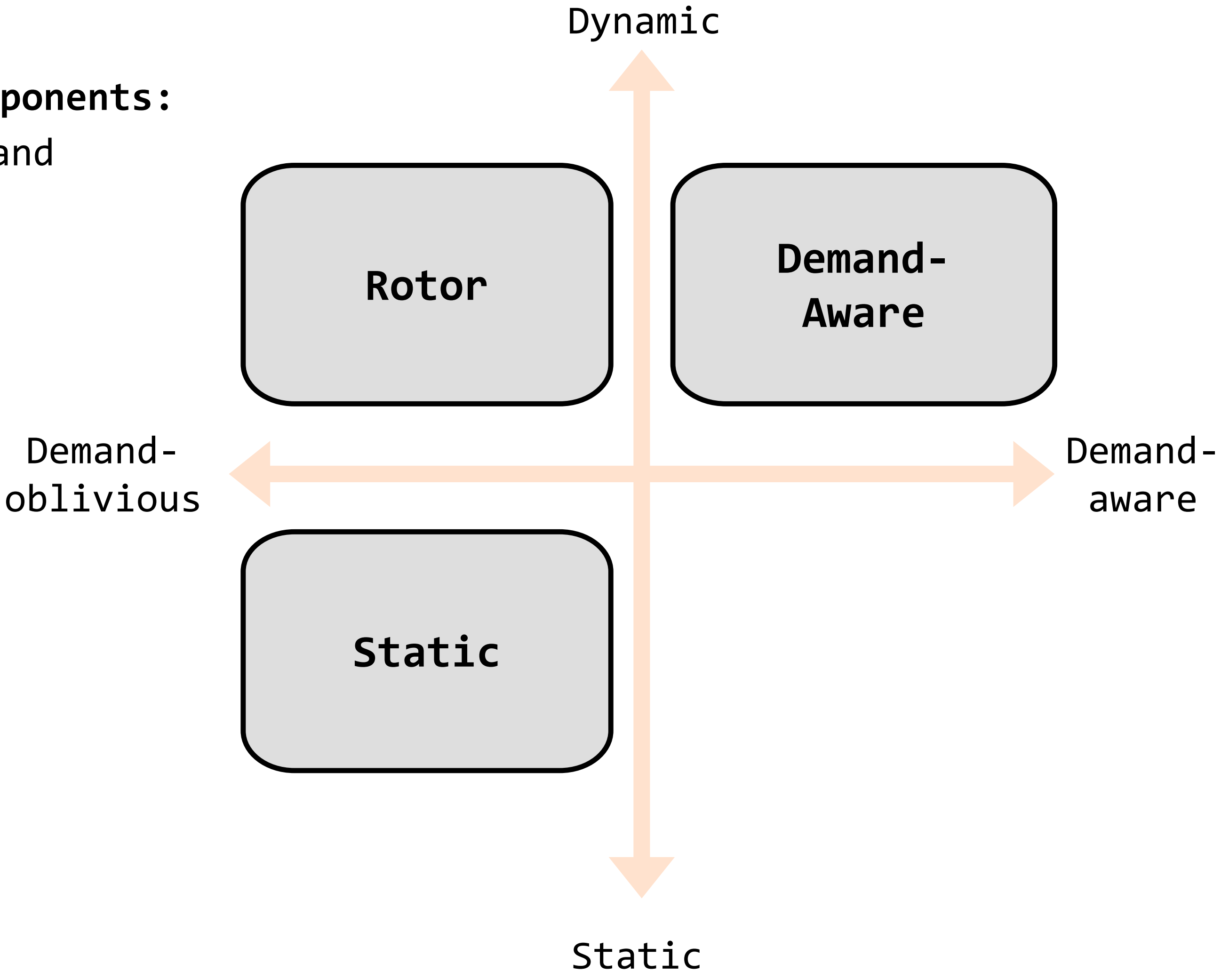
Static

Reality: Tech Diversity

Diverse topology components:

⇀ demand-oblivious and demand-aware

⇀ static vs dynamic

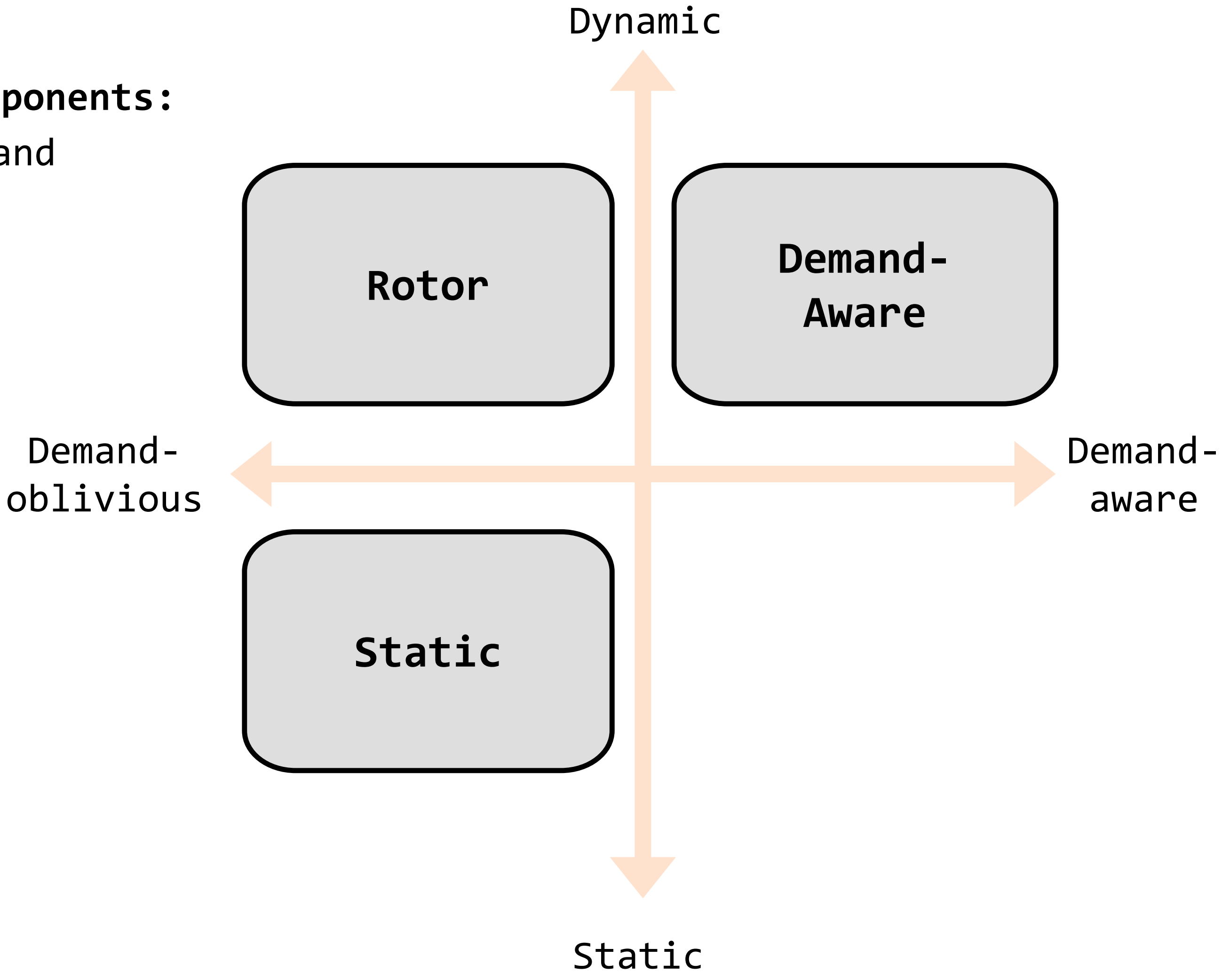


Reality: Tech Diversity

Diverse topology components:

→ demand-oblivious and
demand-aware

→ static vs dynamic



Dynamic

Rotor

Demand-Aware

Demand-oblivious

Demand-aware

Static

Static

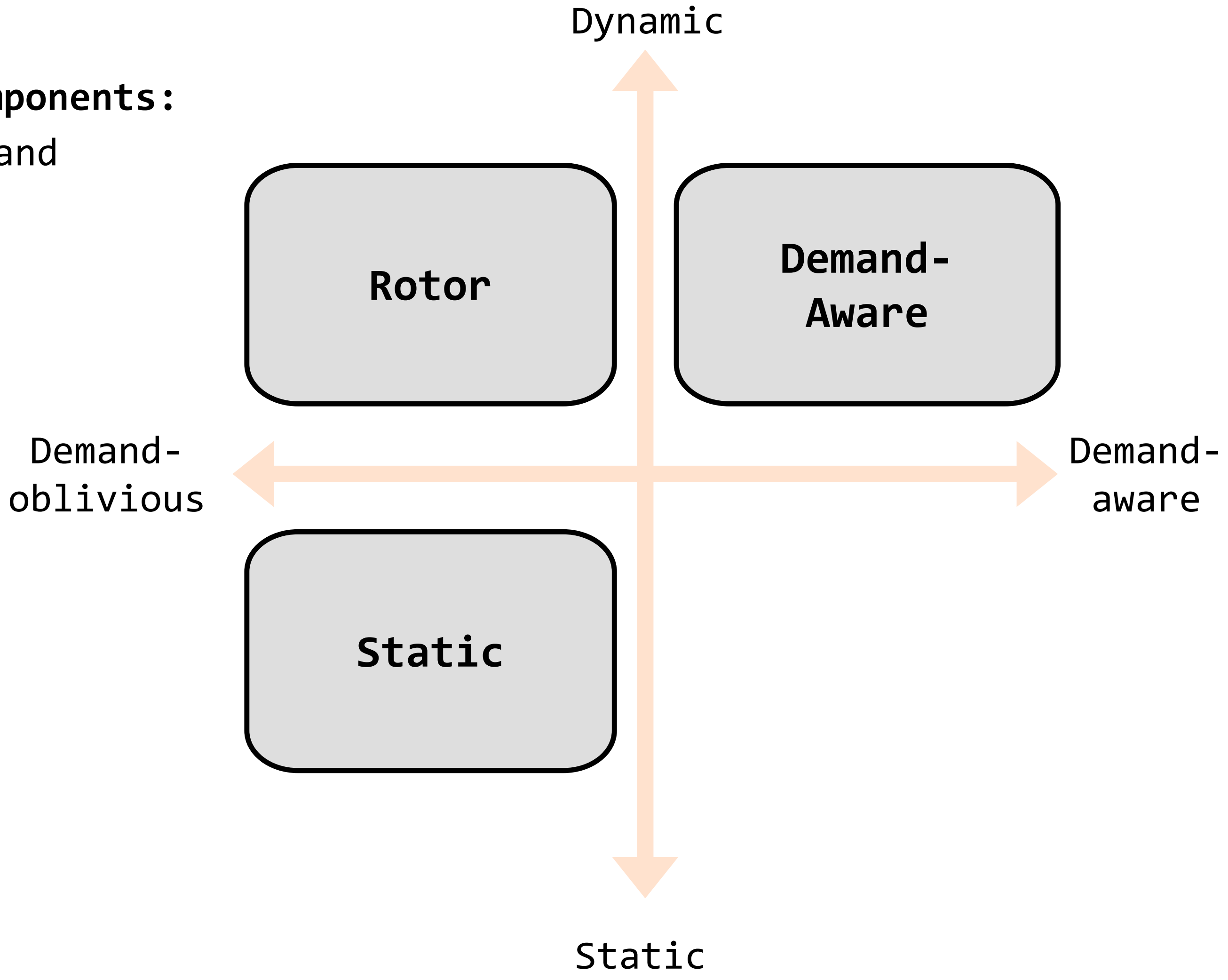
Which approach is best?

Reality: Tech Diversity

Diverse topology components:

⇀ demand-oblivious and demand-aware

⇀ static vs dynamic



Which approach is best?

As always in CS:
It depends…

… On Traffic Diversity

Diverse patterns:

⇀ Shuffling/Hadoop: all-to-all

⇀ All-reduce/ML: ring or tree traffic patterns

  ⇀ Elephant flows

⇀ Query traffic: skewed

  ⇀ Mice flows

⇀ Control traffic: does not evolve but has non-temporal structure

Diverse requirements:

⇀ ML is bandwidth hungry, small flows are latency-sensitive

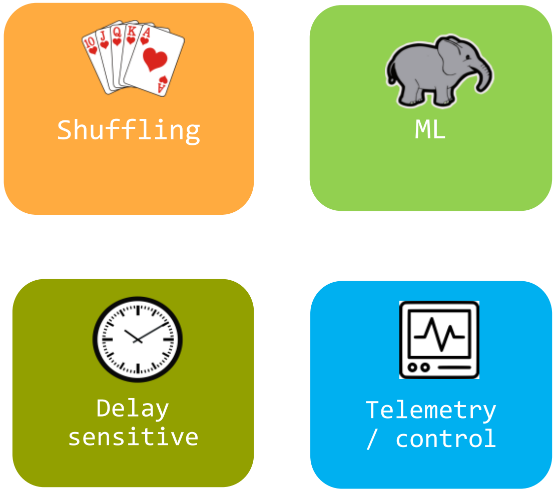
Shuffling
All-to-All

ML
Large flows

Delay sensitive

Telemetry / control

Examples:
Match or Mismatch?

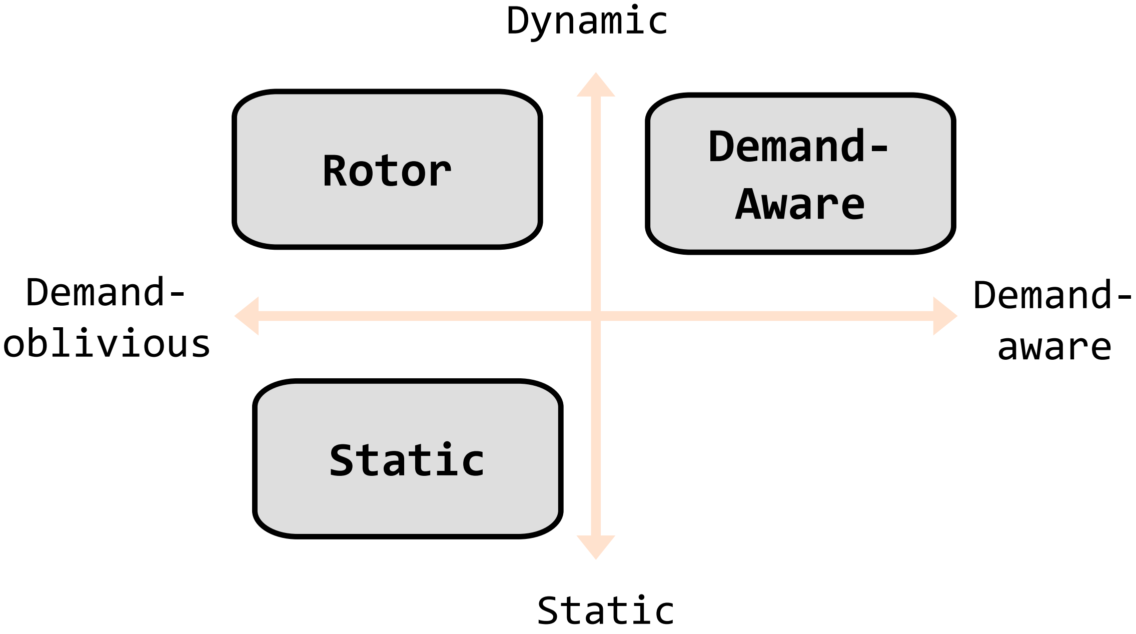

Shuffling

ML

Delay sensitive

Telemetry / control

Demand


Dynamic

Rotor

Demand-Aware

Demand-oblivious

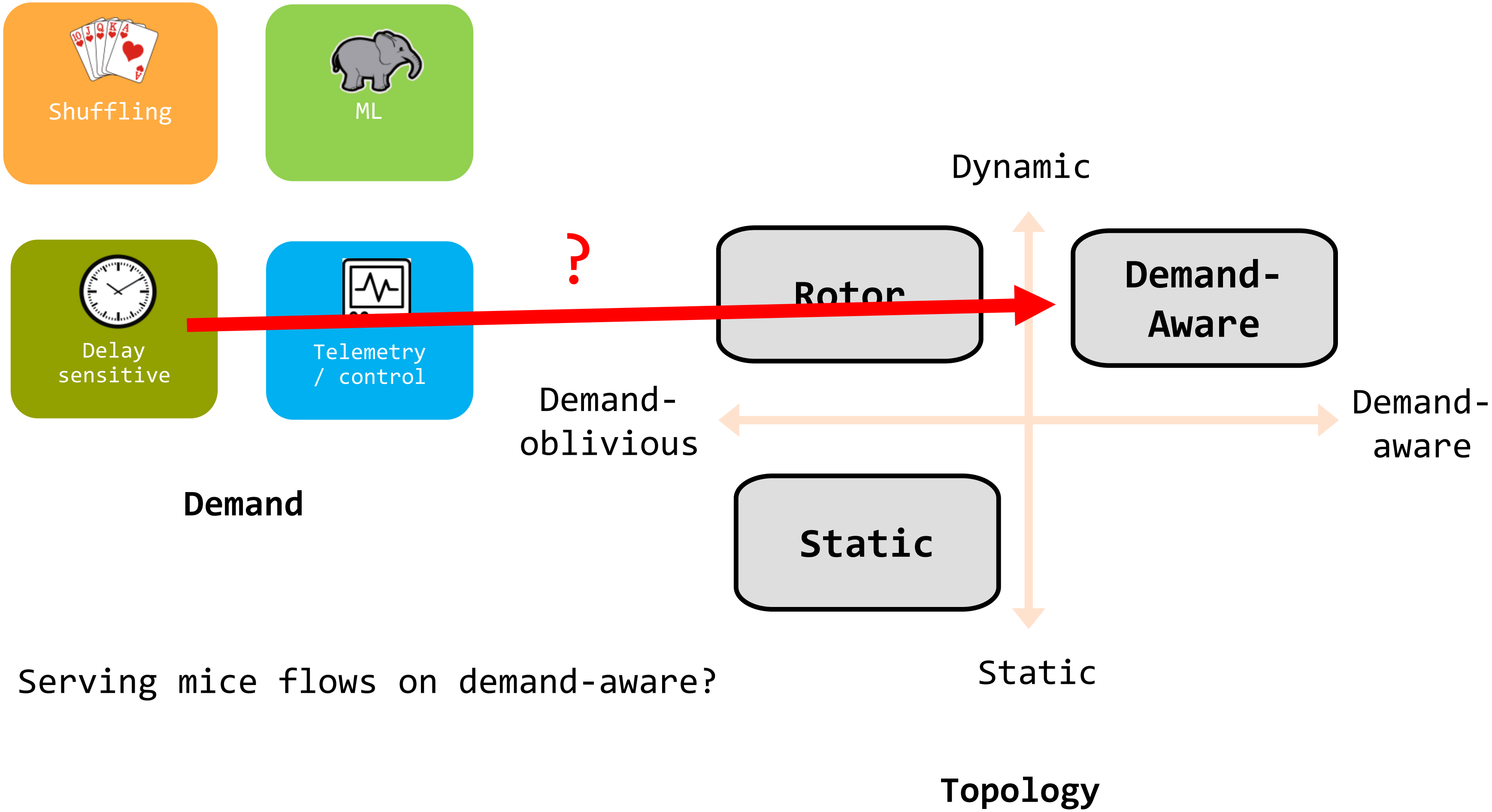
Demand-aware

Static

Static

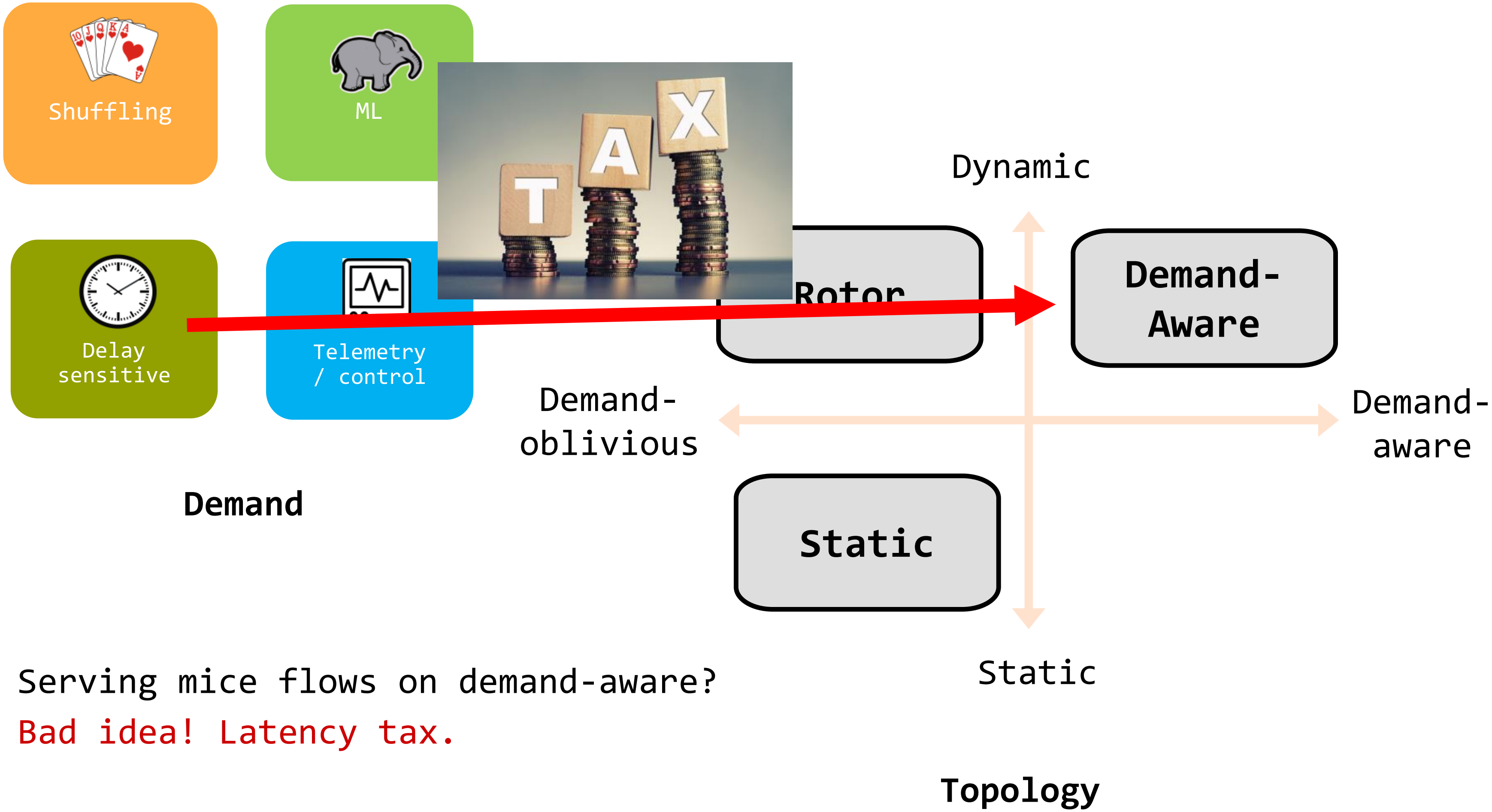
Topology

Examples:
Match or Mismatch?



Serving mice flows on demand-aware?

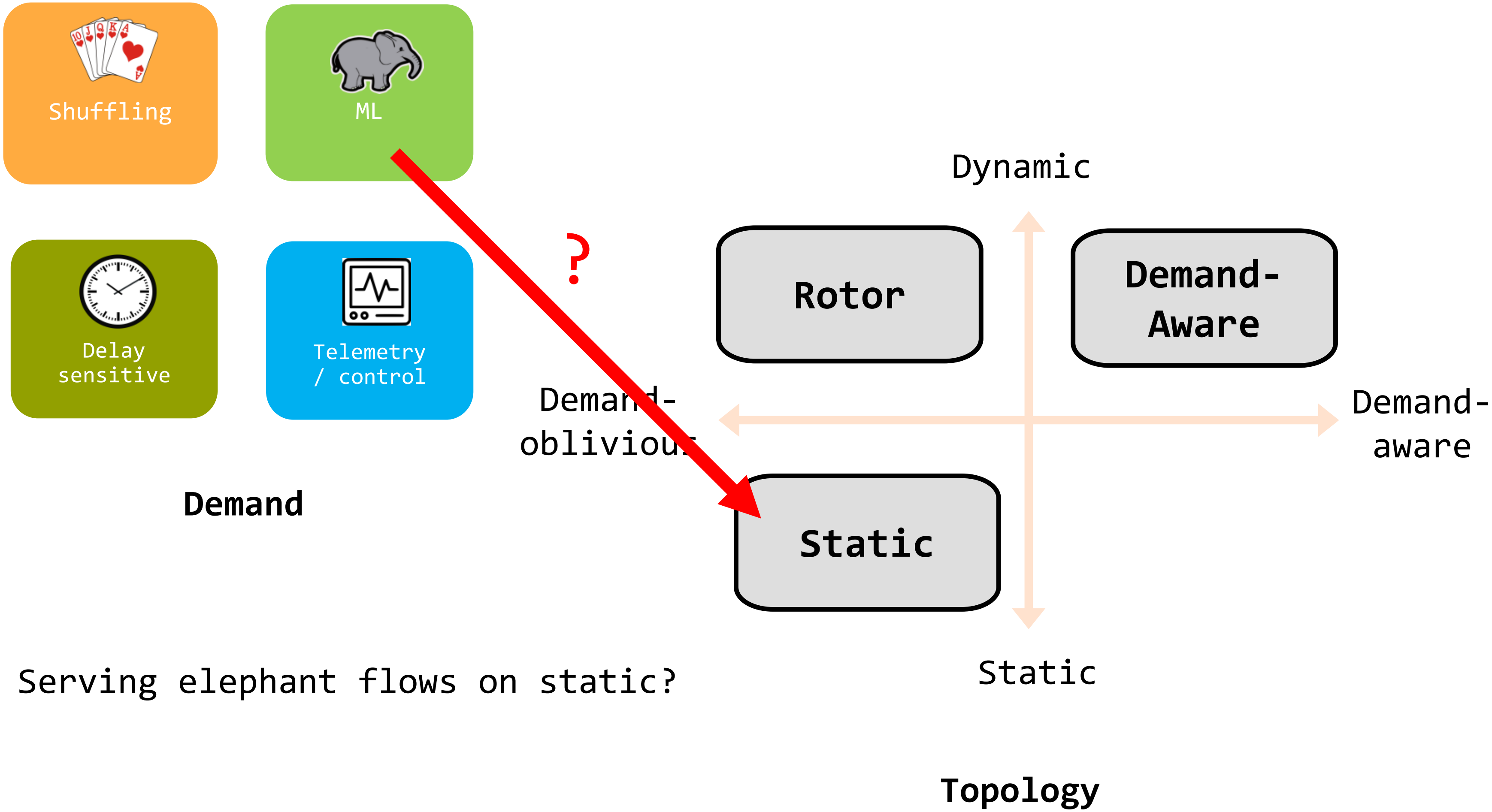
Examples:
Match or Mismatch?



Serving mice flows on demand-aware?

Bad idea! Latency tax.

Examples:
Match or Mismatch?



Shuffling

ML

Delay sensitive

Telemetry / control

Demand

?

Dynamic

Rotor

Demand-Aware

Demand-oblivious

Demand-aware

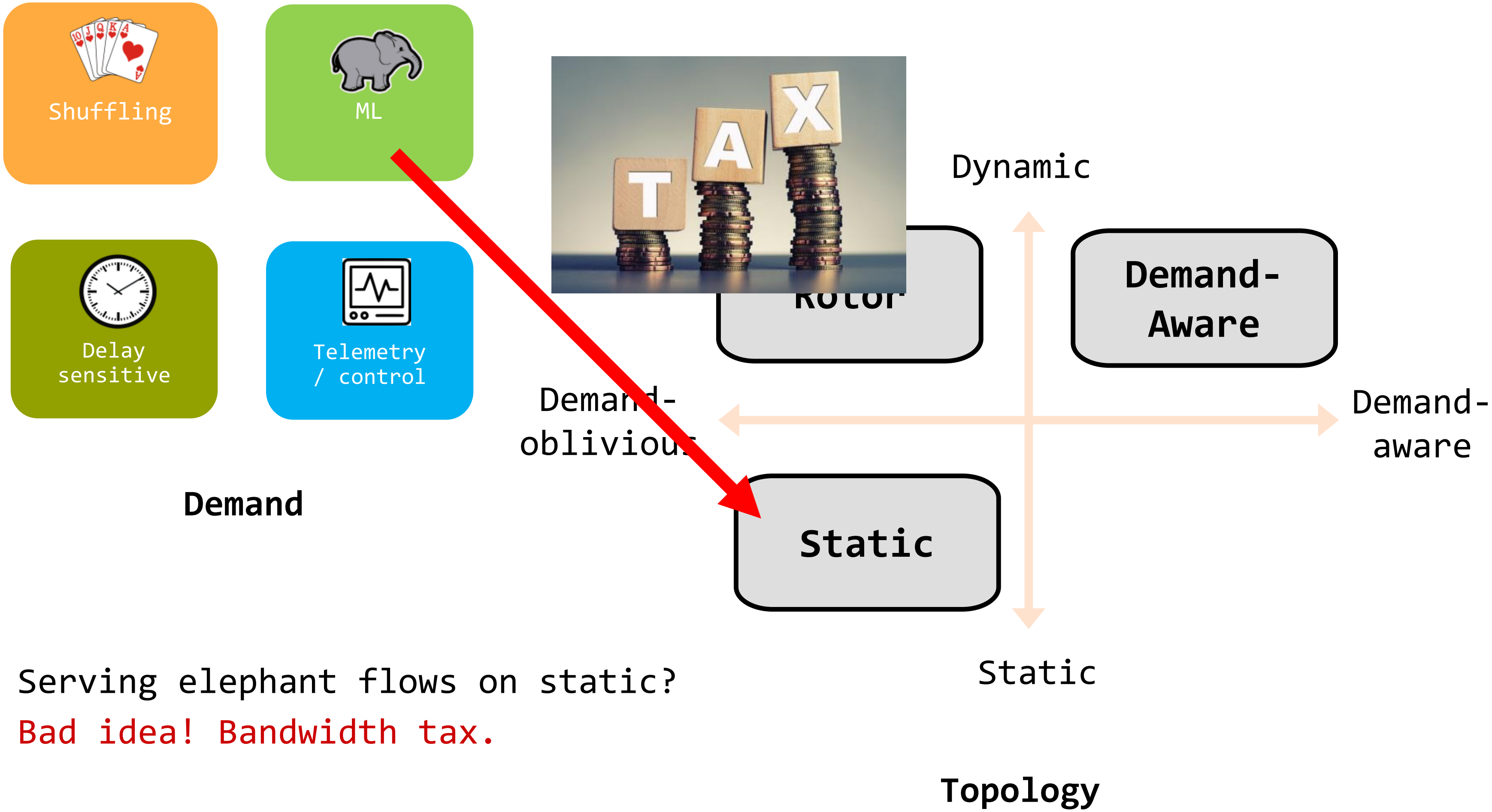
Static

Static

Topology

Serving elephant flows on static?

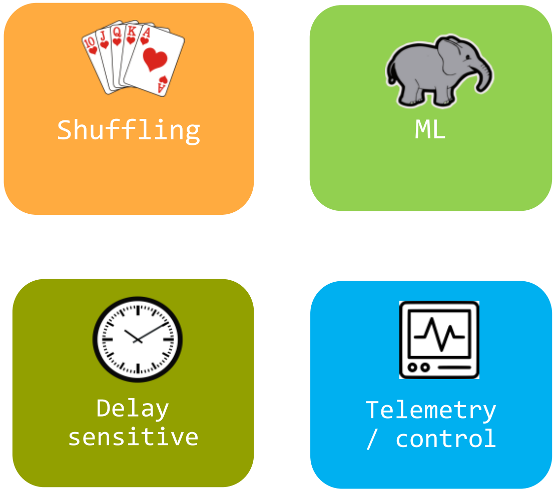
Examples:
Match or Mismatch?



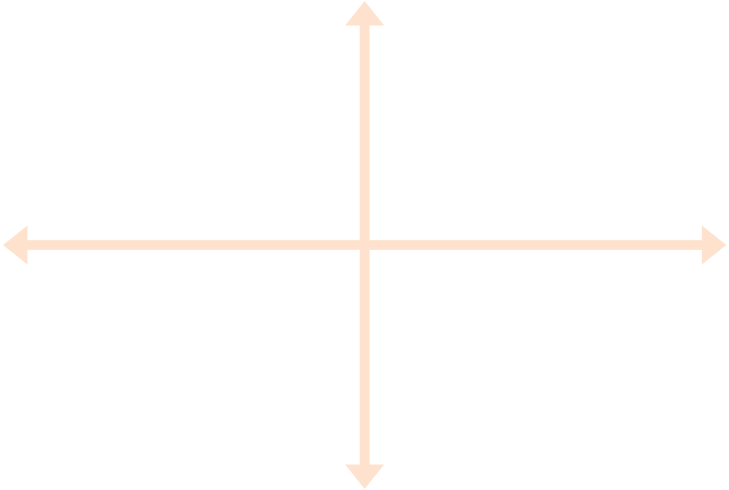
Serving elephant flows on static?

Bad idea! Bandwidth tax.

Examples:
Match or Mismatch?



Demand



Dynamic

Demand-oblivious

Demand-aware

Static

Topology

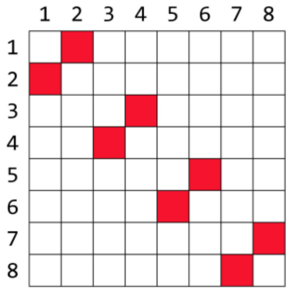
Serving elephant flows on static?

Bad idea! Bandwidth tax.

Throughput Analysis

Demand Matrix

T

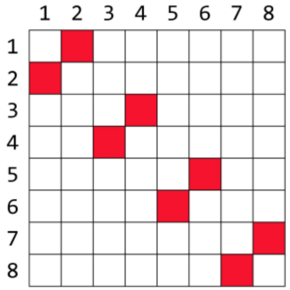


	1	2	3	4	5	6	7	8
1		🟥						
2	🟥							
3				🟥				
4			🟥					
5						🟥		
6					🟥			
7								🟥
8							🟥	

Metric: throughput
of a demand matrix…

Throughput Analysis

Demand Matrix



	1	2	3	4	5	6	7	8
1		🟥						
2	🟥							
3				🟥				
4			🟥					
5						🟥		
6					🟥			
7								🟥
8							🟥	

$T \times \theta(T)$

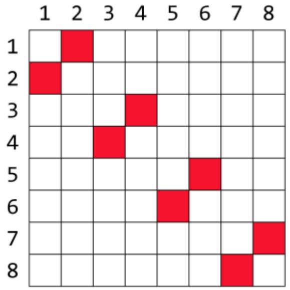
Metric: throughput of a demand matrix…

… is the maximal scale down factor by which traffic is feasible.

Throughput Analysis

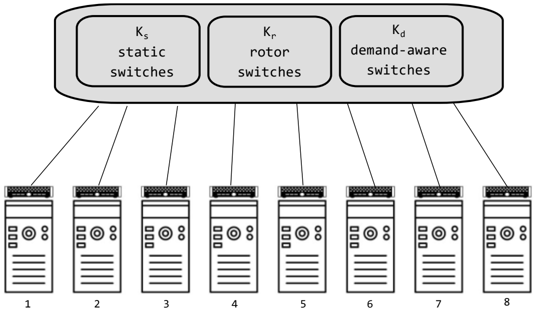
Demand Matrix

T



	1	2	3	4	5	6	7	8
1		🟥						
2	🟥							
3				🟥				
4			🟥					
5						🟥		
6					🟥			
7								🟥
8							🟥	

$\times \ \theta(T) \quad \Rightarrow$



Metric: throughput of a demand matrix…

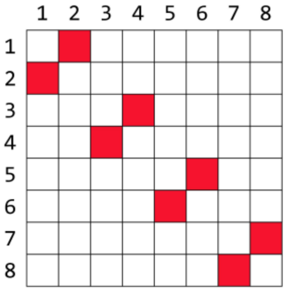
… is the maximal scale down factor by which traffic is feasible.

Throughput of network θ^*: worst case T

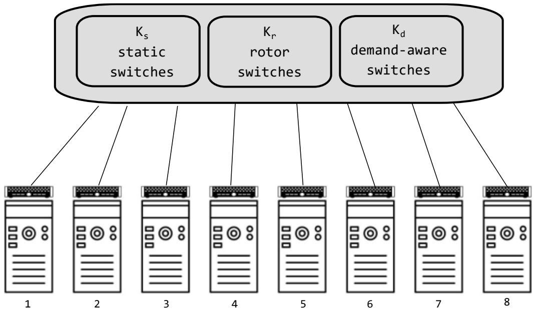
Throughput Analysis

Demand Matrix

T



$\times\ \theta(T)\ \Rightarrow$

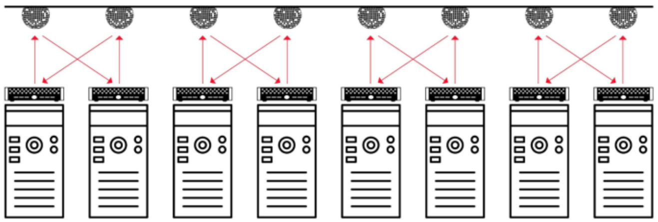


BW tax!

BW & latency tax!

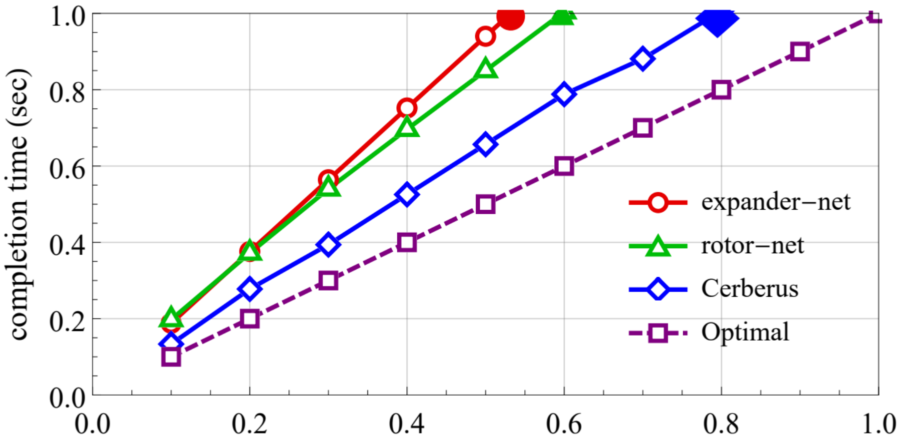
	expander-net	*rotor-net*	CERBERUS
BW-Tax	✓	✓	✗
LT-Tax	✗	✓	✓
$\theta(T)$	Thm 2	Thm 3	Thm 5
θ^*	0.53	0.45	Open
Datamining	0.53	0.6	0.8 (+33%)
Permutation	0.53	0.45	≈ 1 (+88%)
Case Study	0.53	0.66	0.9 (+36%)

Worst demand matrix for static and rotor: permutation. Best case for demand-aware!



Completion Time

- Demand completion time: How long does it take to serve a demand matrix?



Data mining workload

- Also useful in analysis: throughput can be computed more easily via demand completion time.