Revolutionizing Datacenter Networks via Reconfigurable Topologies

Stefan Schmid (TU Berlin)

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

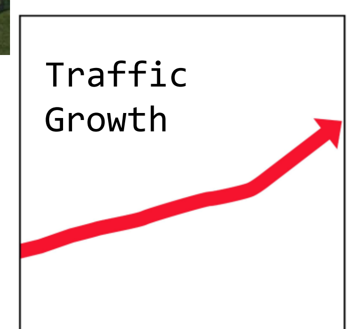
Acknowledgements:

We live in the age of

Distributed Computation

Datacenters (“hyper-scale”)

+network

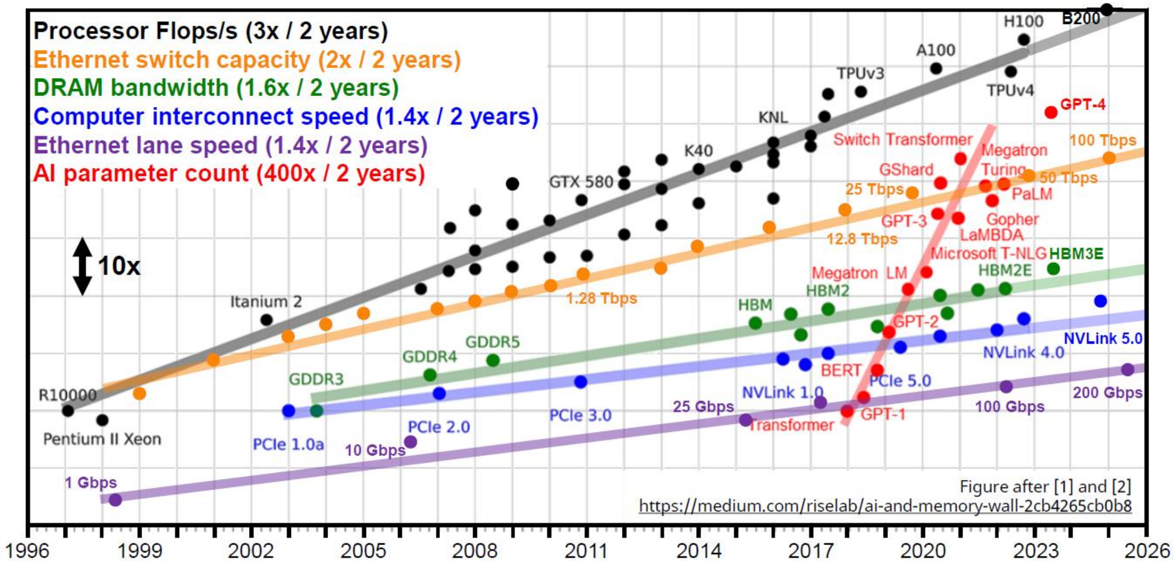


Source: Facebook

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

Technological Trends

Increasing Gap Between Compute and Network



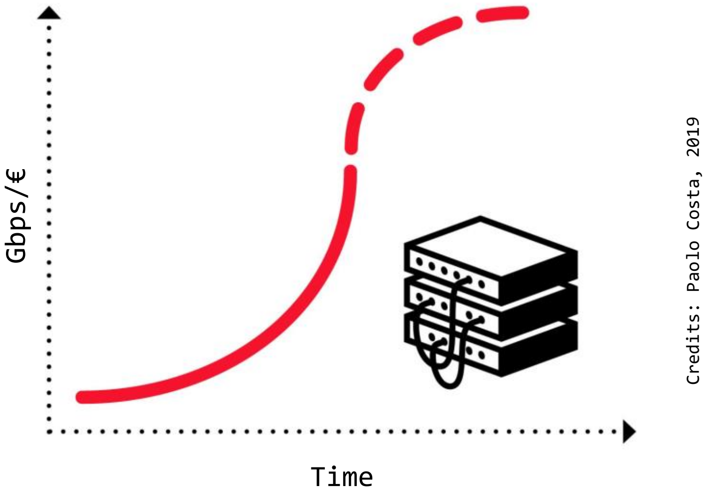
Credits: Nicola Calabretta

The Problem

Huge Infrastructure, Inefficient Use

⇢ Hence: more equipment, larger networks

⇢ Resource intensive and: **inefficient**

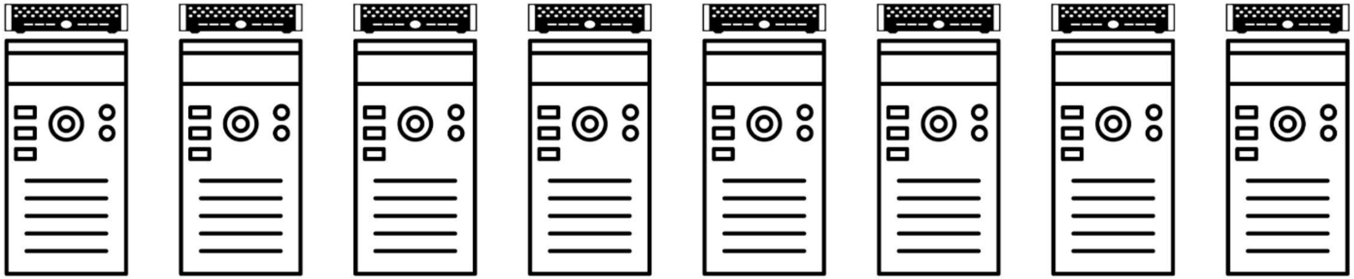


Annoying for companies, **opportunity** for researchers!

An Inefficiency

Fixed and Demand-Oblivious Topology

How to interconnect? Focus on this talk: scale-out network.

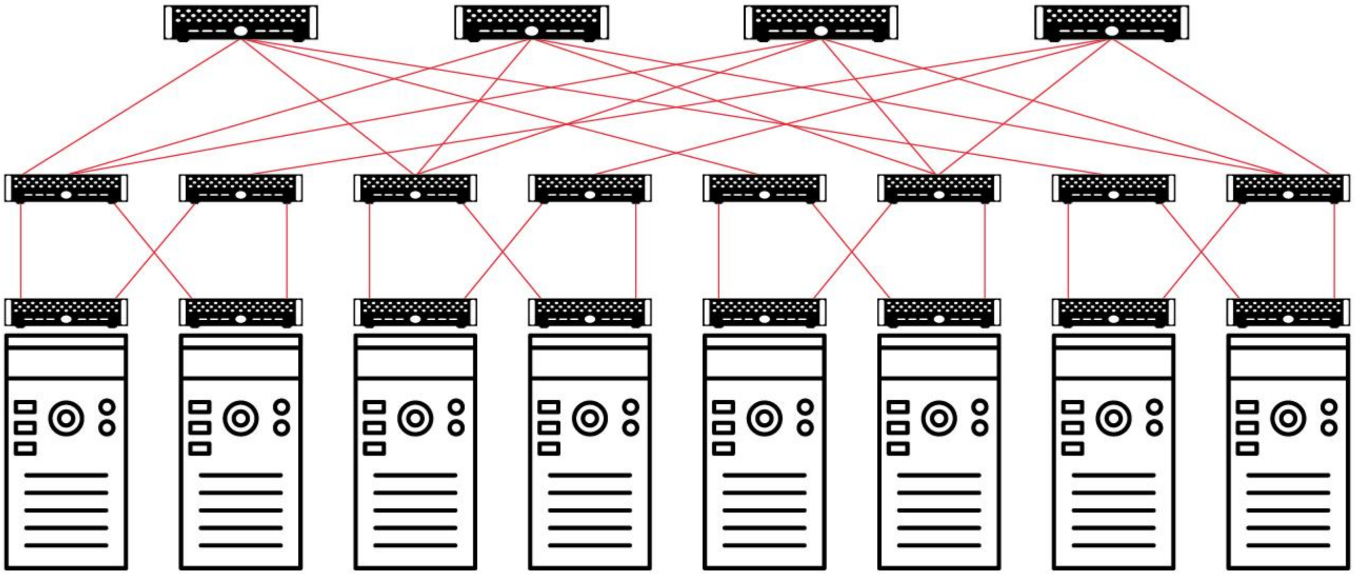


An Inefficiency

Fixed and Demand-Oblivious Topology

⇢ Example: fat-tree topology (bi-regular)

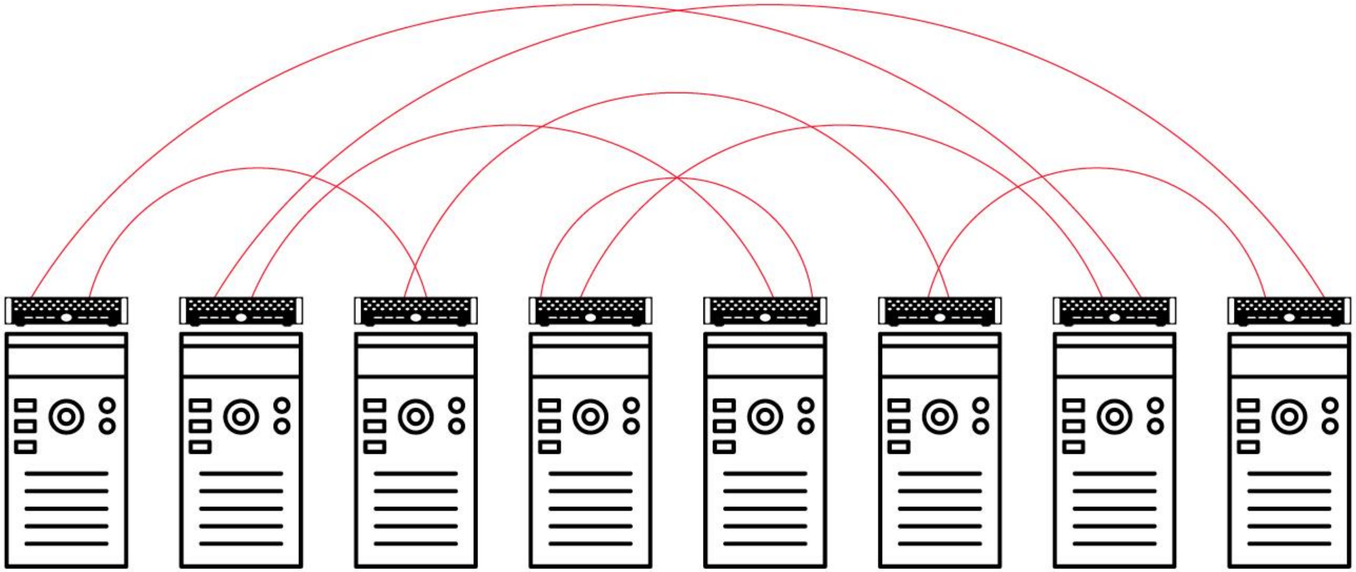
→ 2 types of switches: top-of-rack (ToR) connect to hosts, additional switches connecting switches to increase throughput



An Inefficiency

Fixed and Demand-Oblivious Topology

- Example: expander topology (uni-regular)
  - Only 1 type of switches:
    lower installation and management overheads

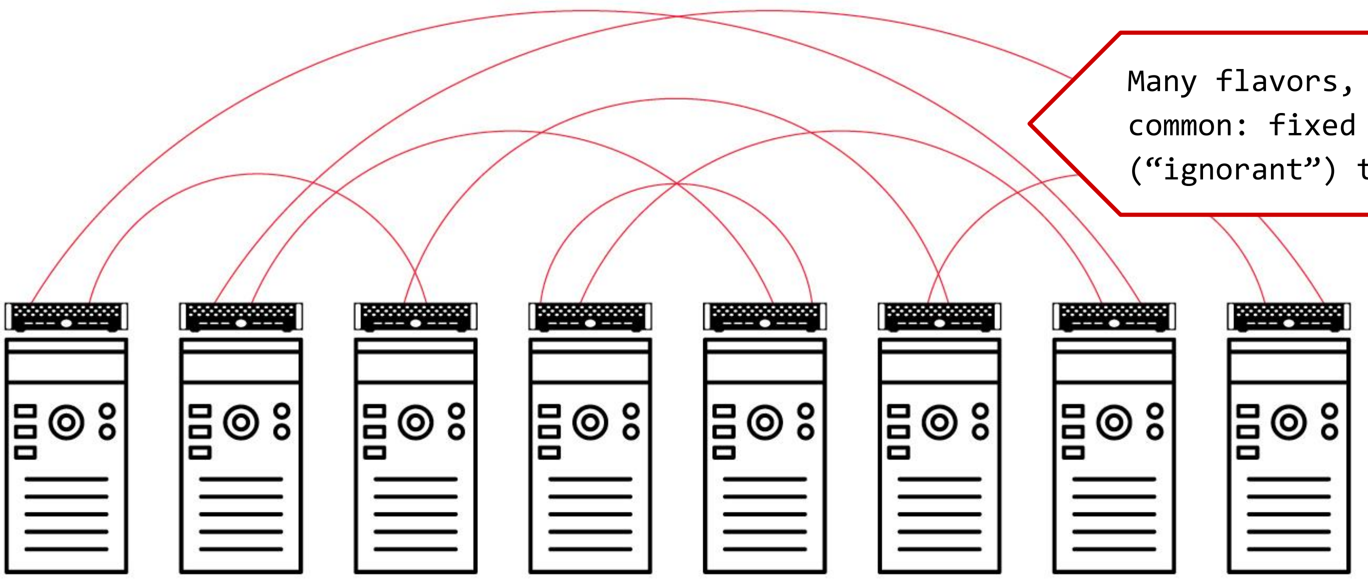


An Inefficiency

Fixed and Demand-Oblivious Topology

⇢ Example: expander topology (uni-regular)

→ Only 1 type of switches:
lower installation and management overheads



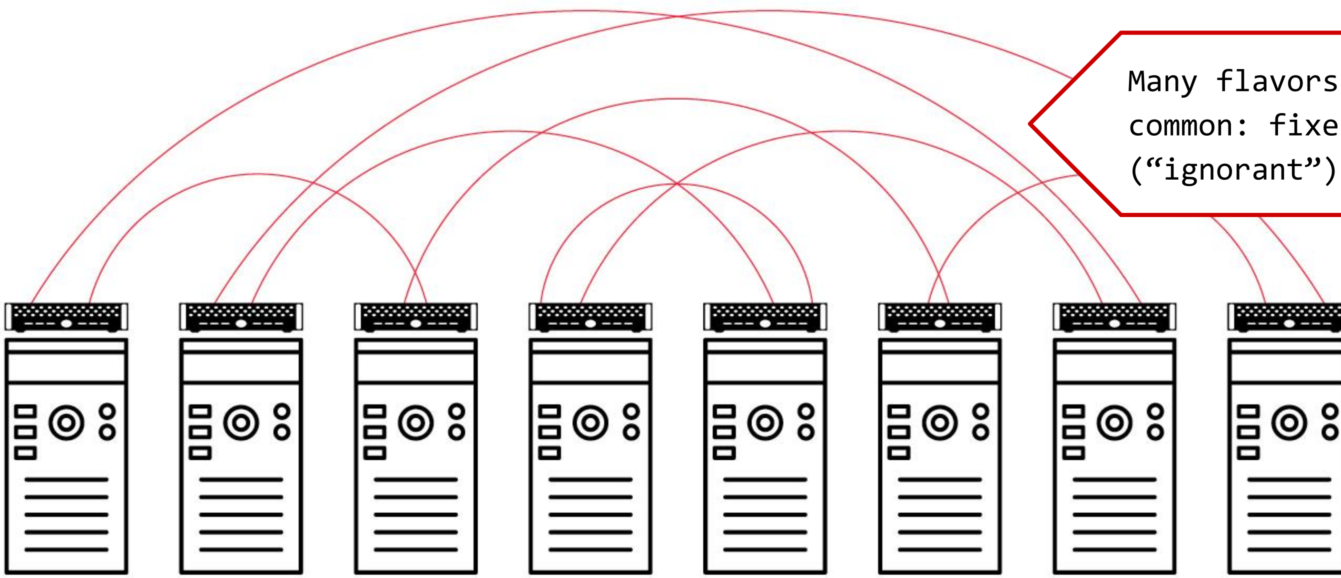
Many flavors, but in common: fixed and **oblivious** (“ignorant”) to actual demand.

An Inefficiency

Fixed and Demand-Oblivious Topology

Highway which ignores actual traffic: frustrating!

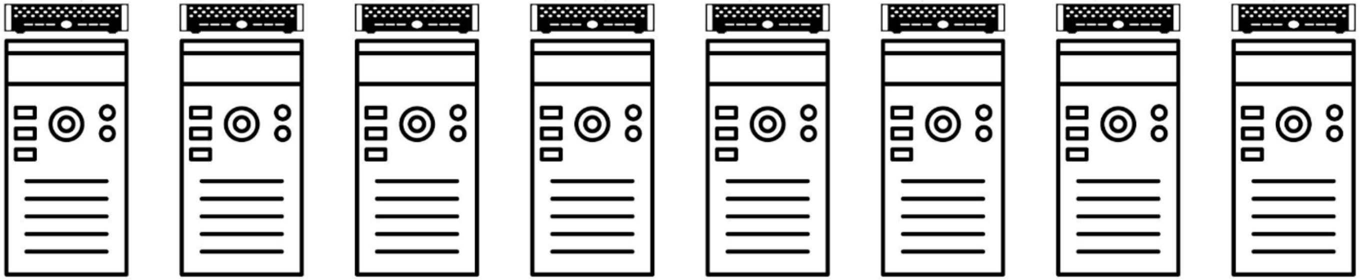
- Example: expander topology (uni-regular)
  - Only 1 type of switches: lower installation and management overheads



Many flavors, but in common: fixed and **oblivious** ("ignorant") to actual demand.

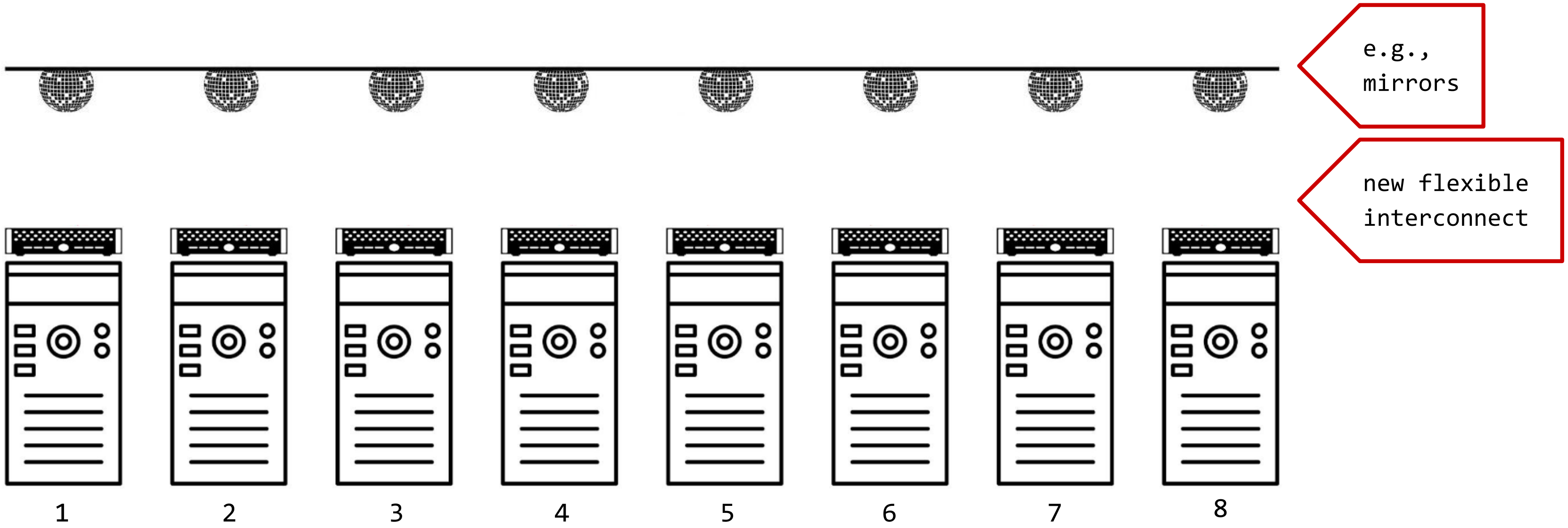
A Vision

Flexible and Demand-Aware Topologies



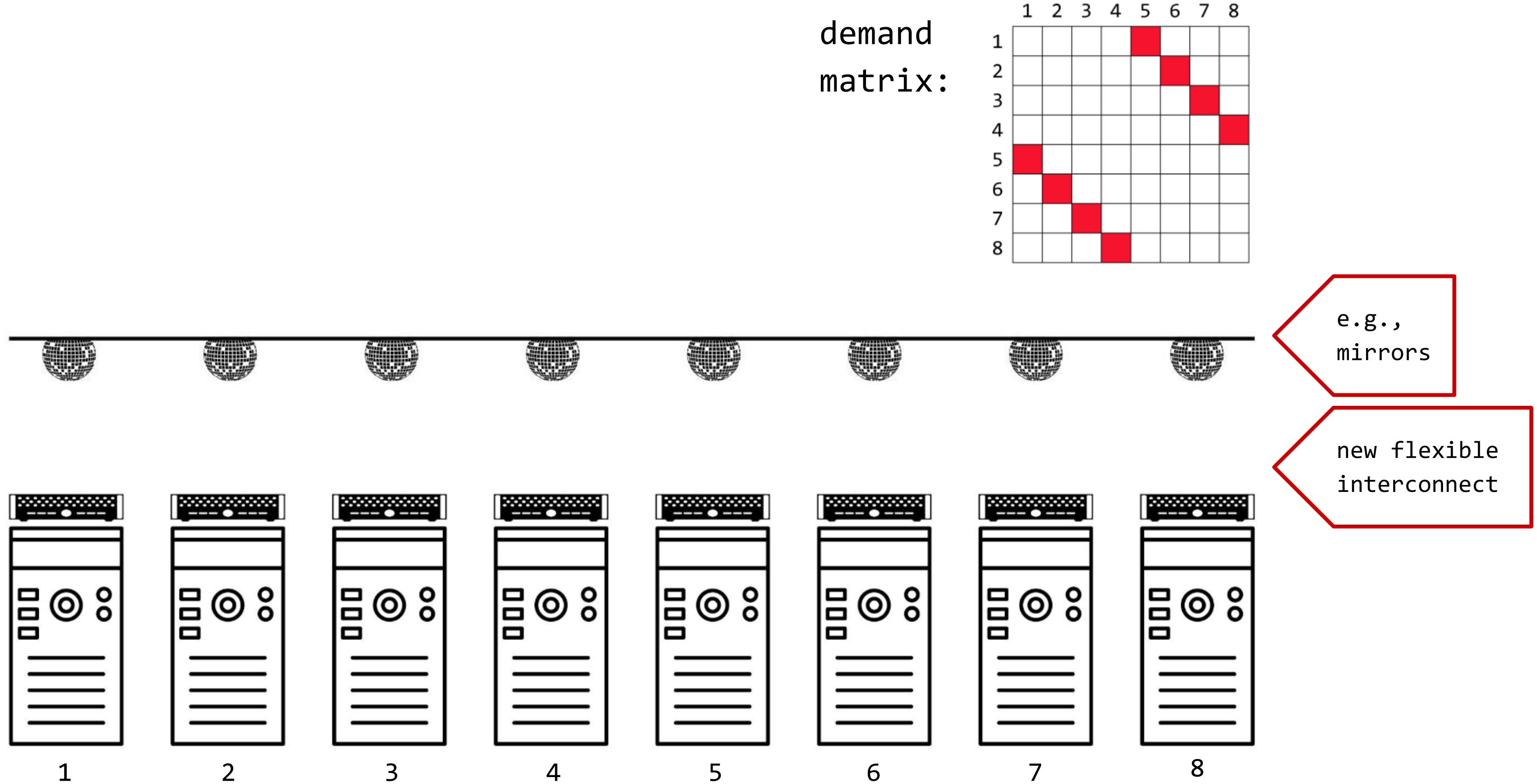
A Vision

Flexible and Demand-Aware Topologies



A Vision

Flexible and Demand-Aware Topologies

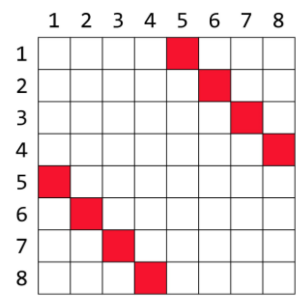


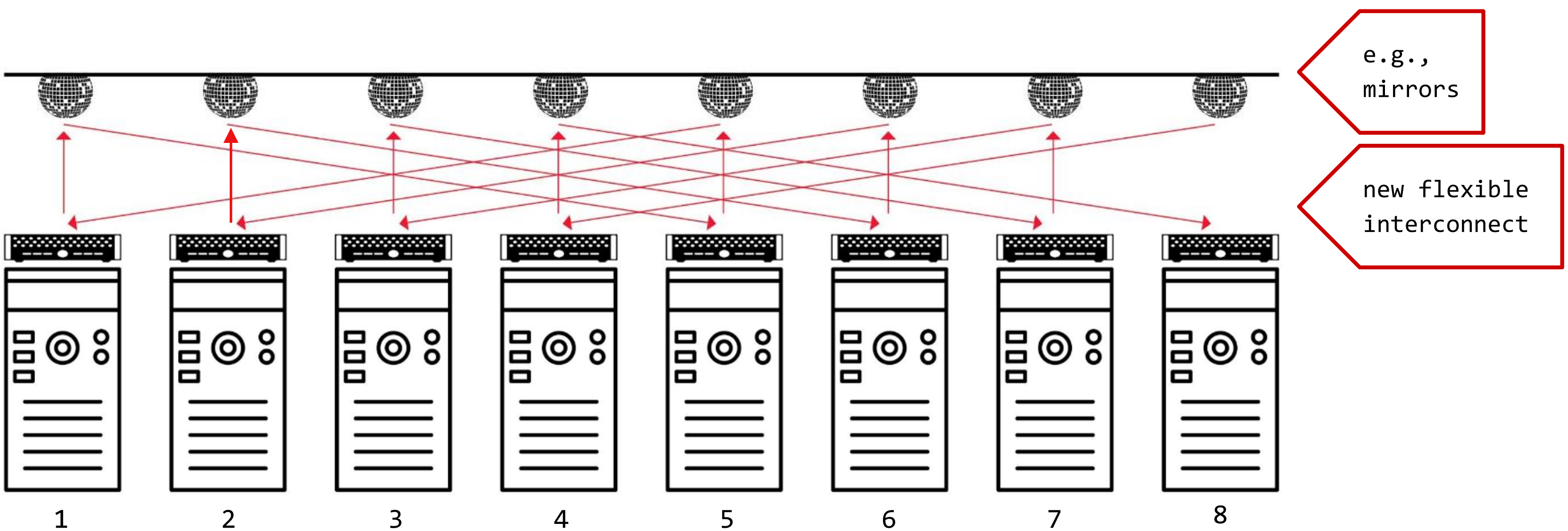
A Vision

Flexible and Demand-Aware Topologies

Matches demand

demand
matrix:

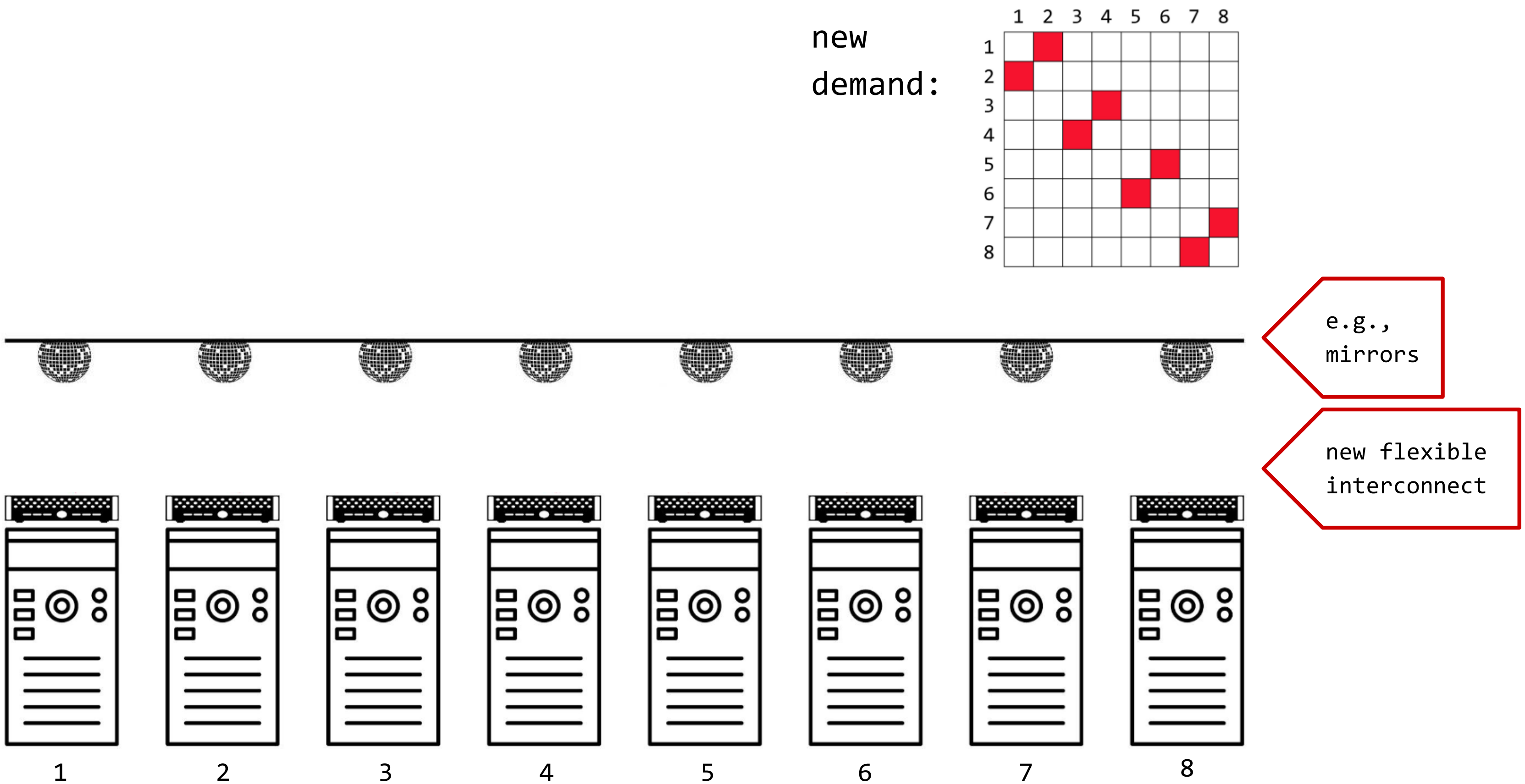




A Vision

Flexible and Demand-Aware Topologies

new demand:

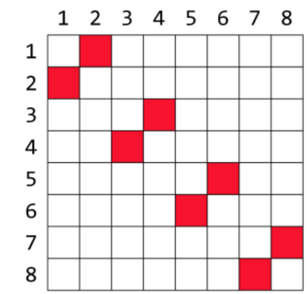


A Vision

Flexible and Demand-Aware Topologies

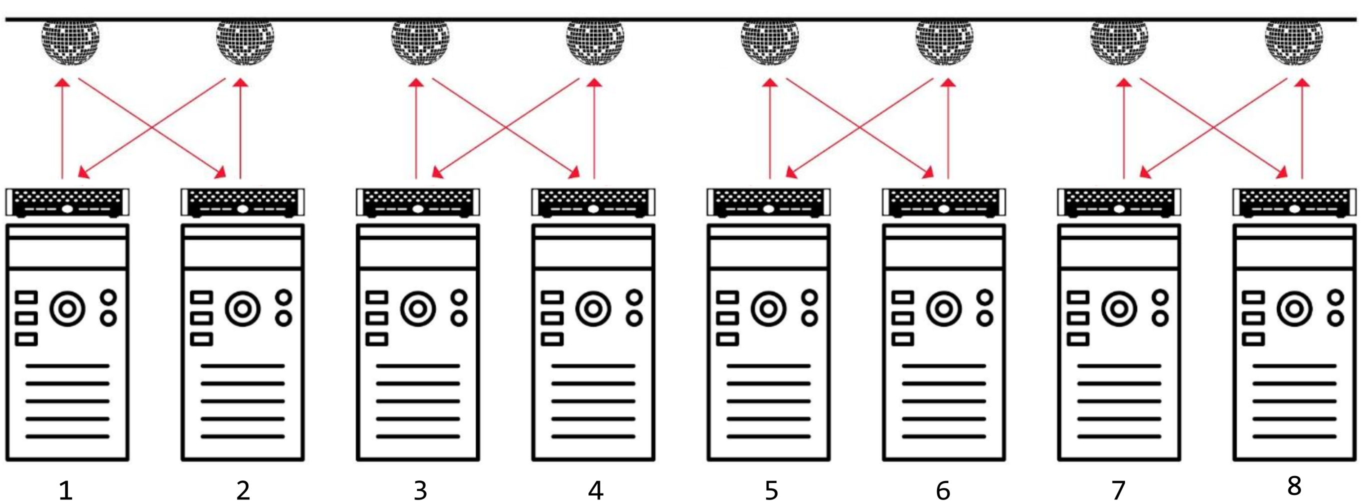
Matches demand

new demand:



e.g., mirrors

new flexible interconnect

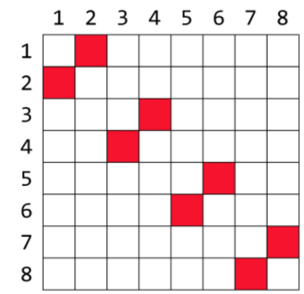


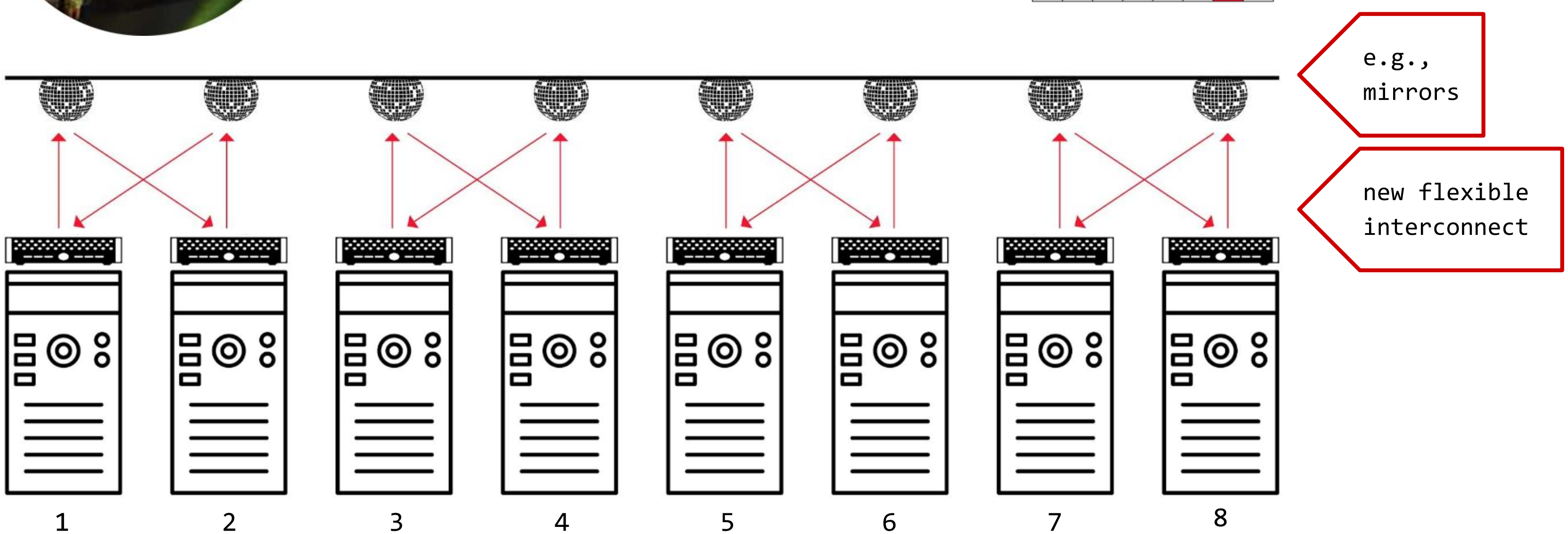
A Vision

Flexible and Demand-Aware Topologies

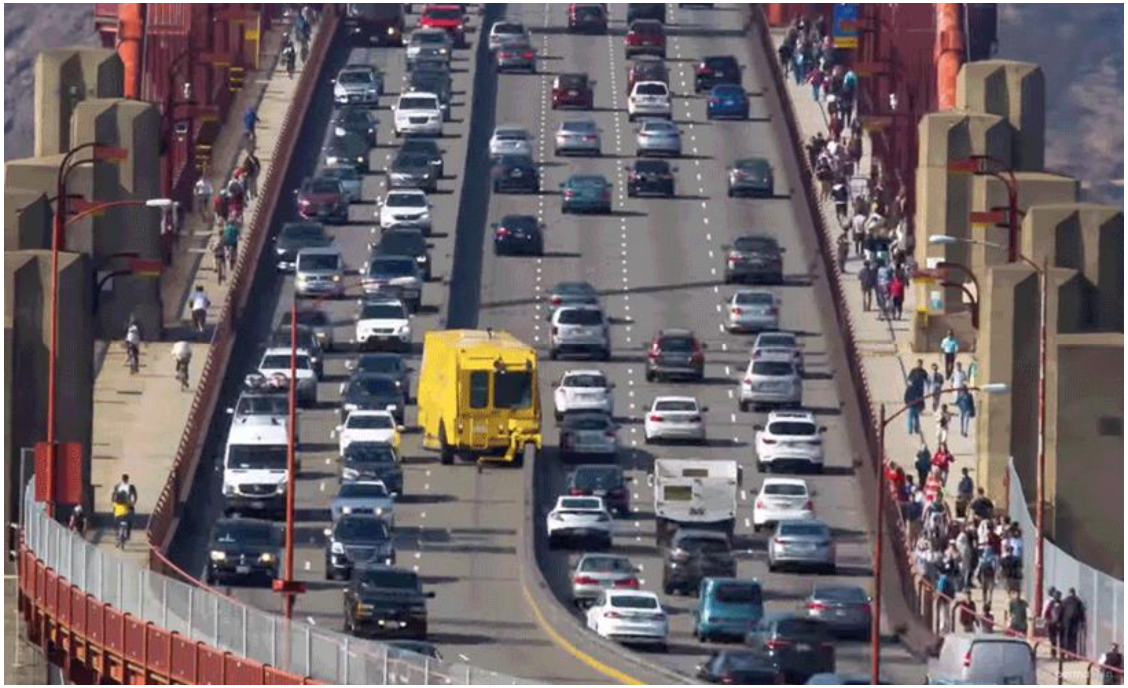
Self-Adjusting Networks

new demand:





Analogy



Golden Gate Zipper

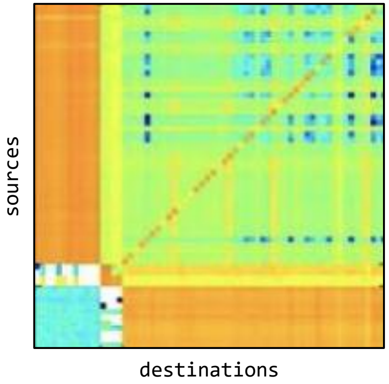
The Motivation

Much Structure in the Demand: Complexity Map

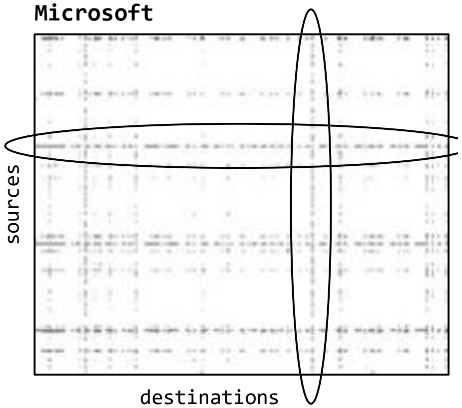
Empirical studies:

traffic matrices sparse and skewed

Facebook

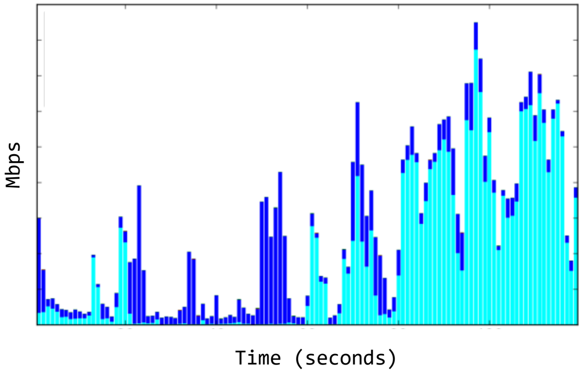


Microsoft



traffic bursty over time

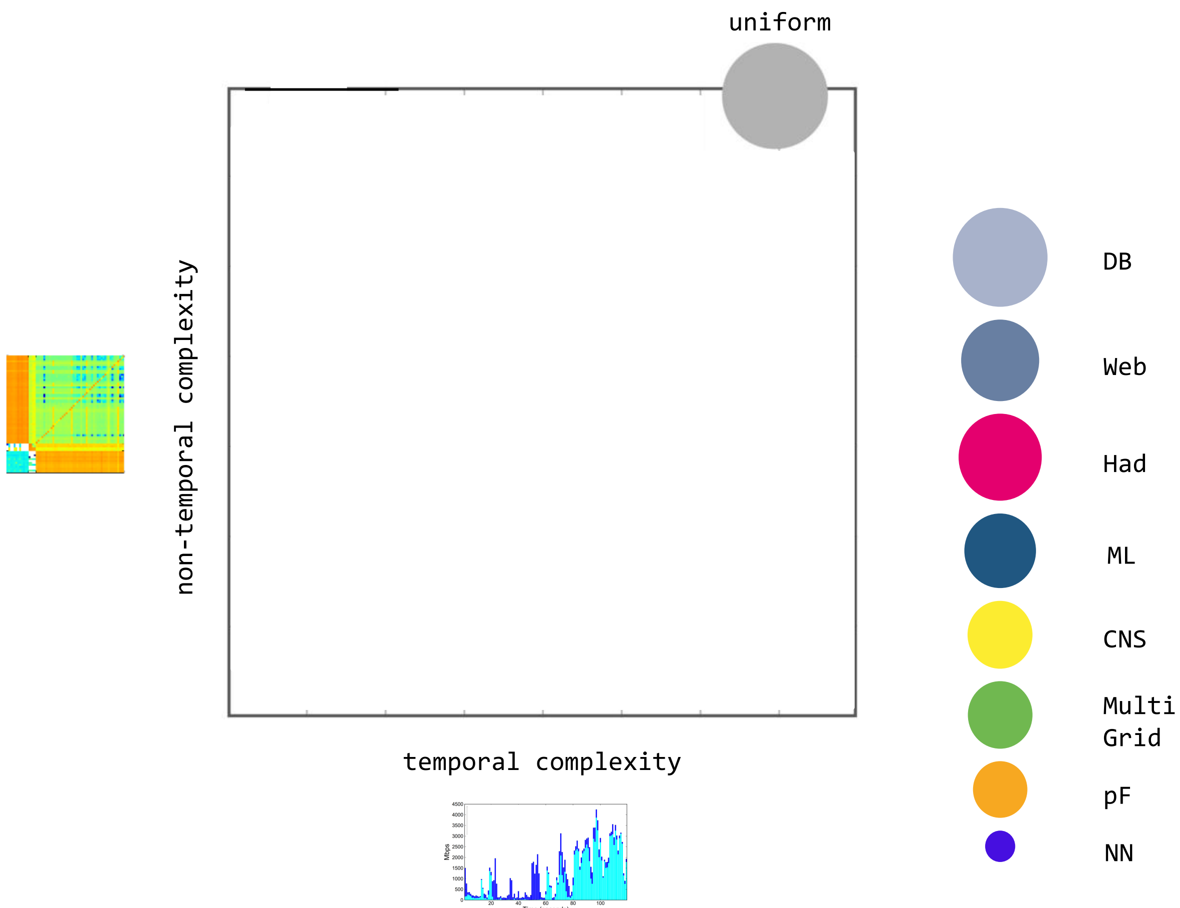
Facebook



The hypothesis: can be exploited.

The Motivation

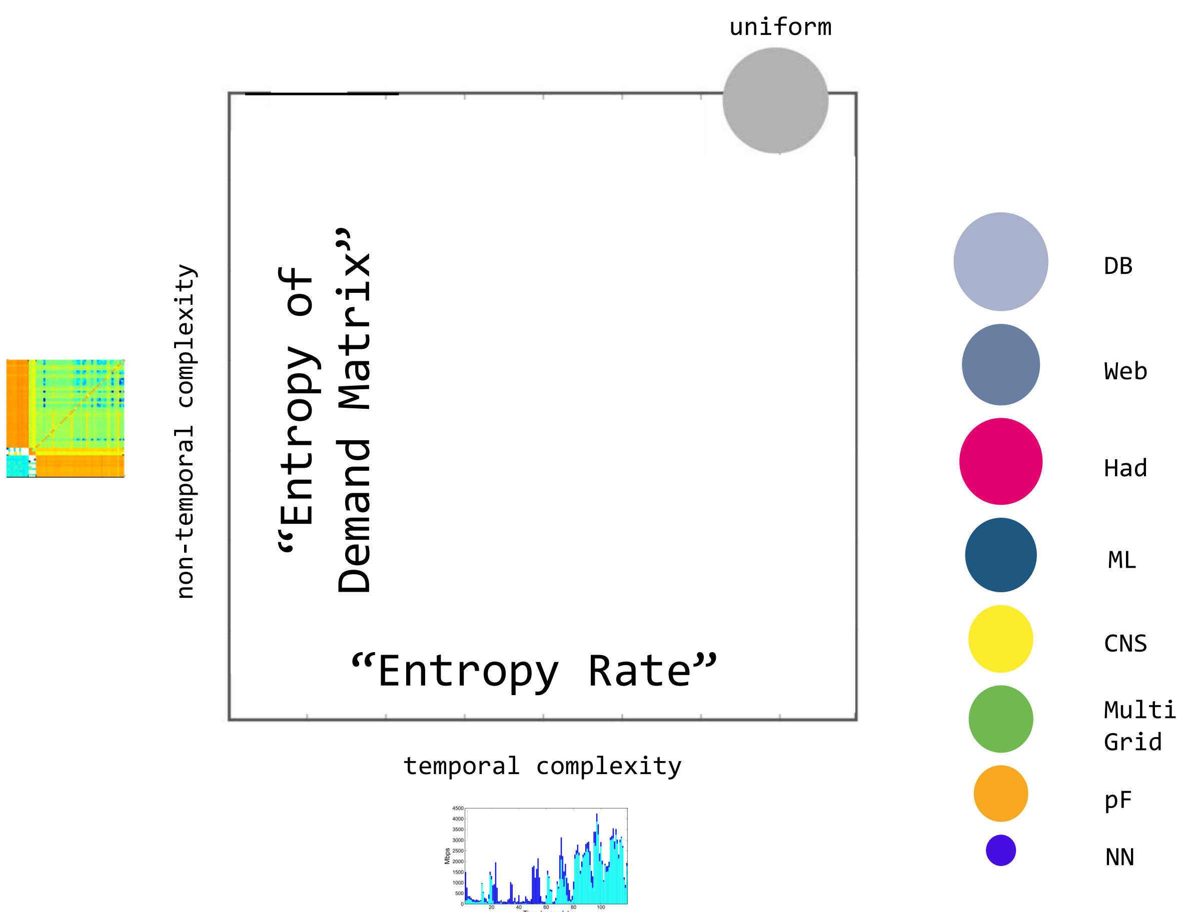
Much Structure in the Demand: Complexity Map



Griner et al., SIGMETRICS 2020

The Motivation

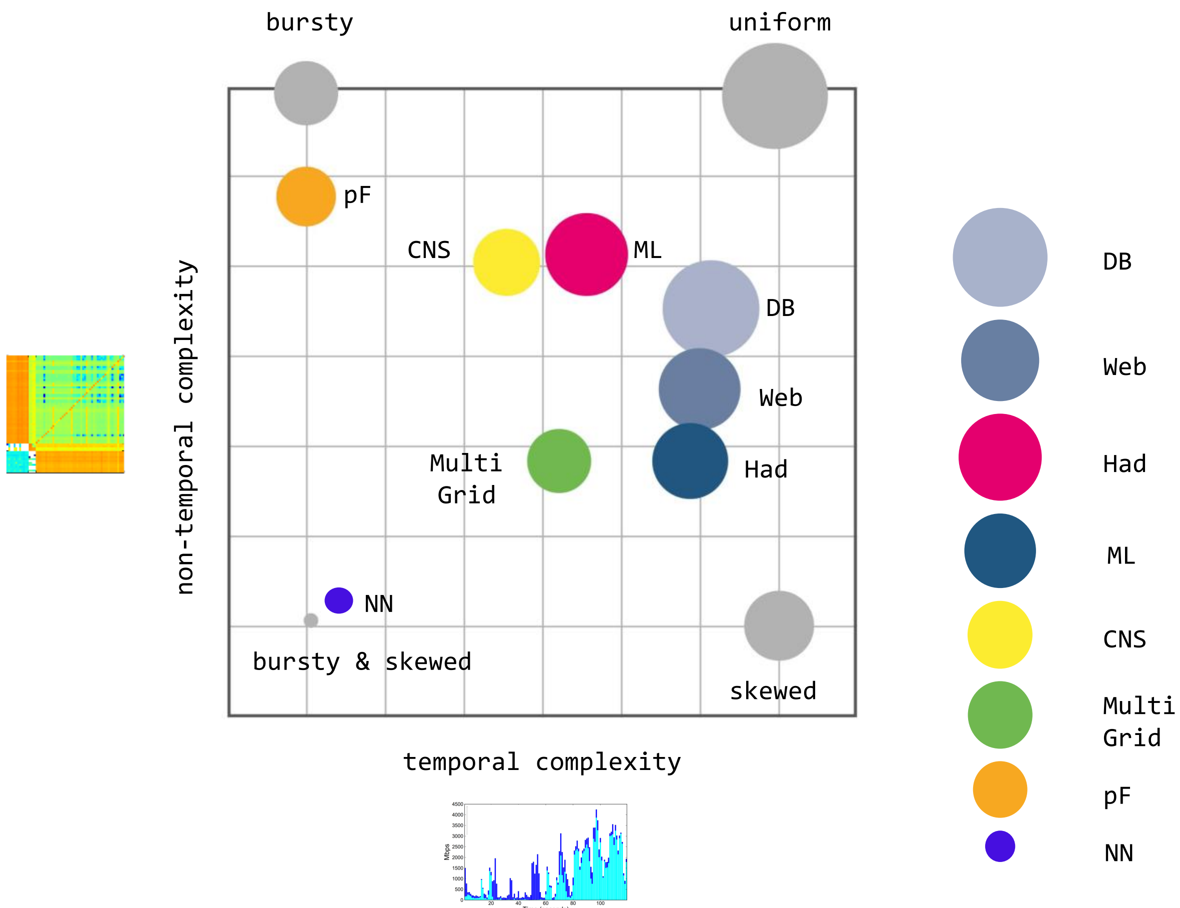
Much Structure in the Demand: Complexity Map



Griner et al., SIGMETRICS 2020

The Motivation

Much Structure in the Demand: Complexity Map

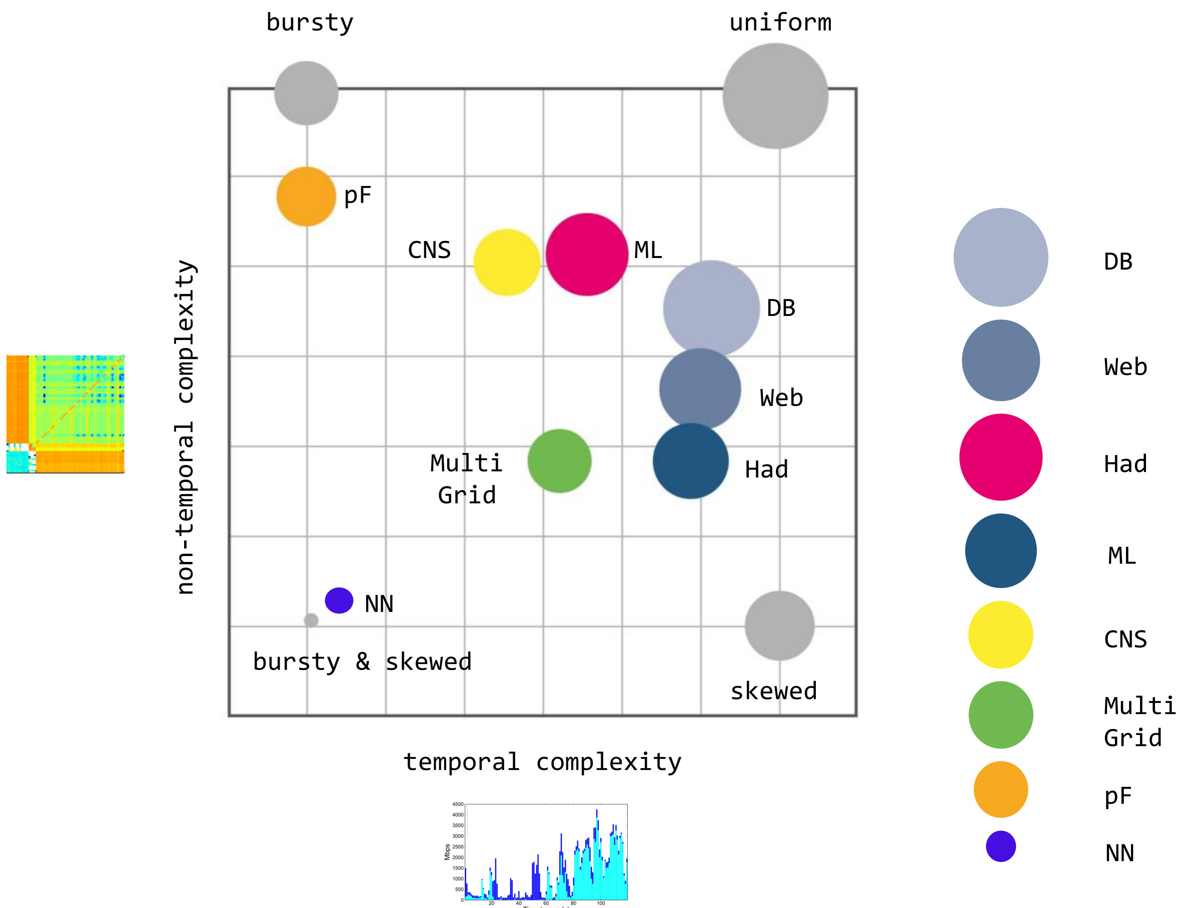


Different structures!

Griner et al., SIGMETRICS 2020

The Motivation

Much Structure in the Demand: Complexity Map



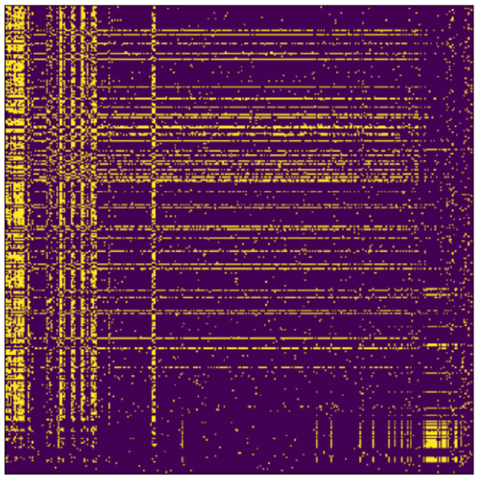
Different structures!

Hypothesis: can be exploited.

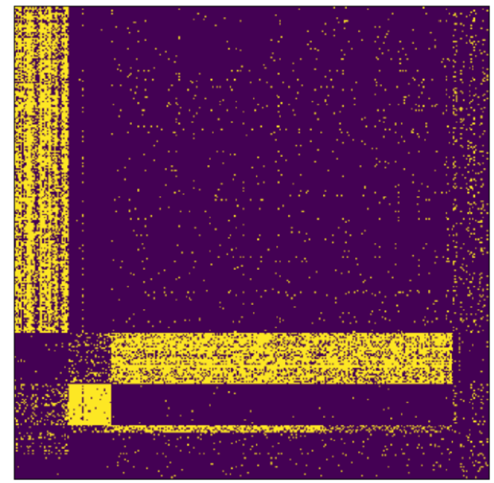
Griner et al., SIGMETRICS 2020

Traffic is also clustered: bi-clustering results

Small Stable Clusters



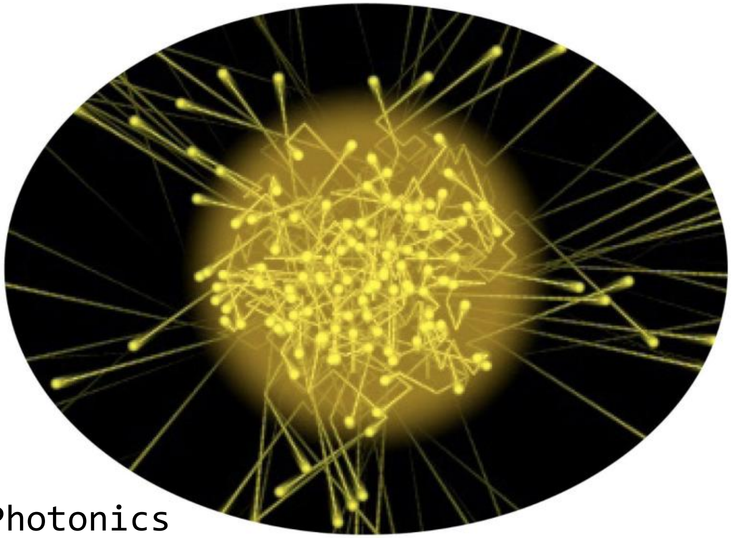
reordering based on bicluster structure



Opportunity: *exploit* with little reconfigurations!

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

Sounds Crazy? Emerging Enabling Technology.



Photonics

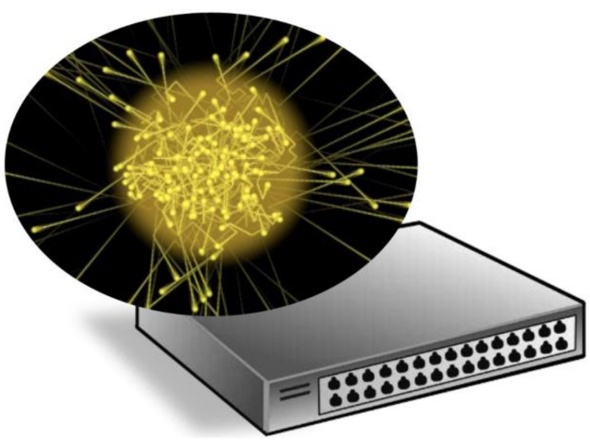
H2020:
“Photonics one of only five key enabling technologies for future prosperity.”

US National Research Council:
“Photons are the new Electrons.”

Enabler

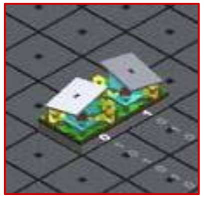
Novel Reconfigurable Optical Switches

⇢ **Spectrum** of prototypes
- ⇀ Different sizes, different reconfiguration times
- ⇀ From our ACM **SIGCOMM** workshop OptSys



Prototype 1

Moving antenna (ms)



Prototype 2

Moving mirrors (mus)

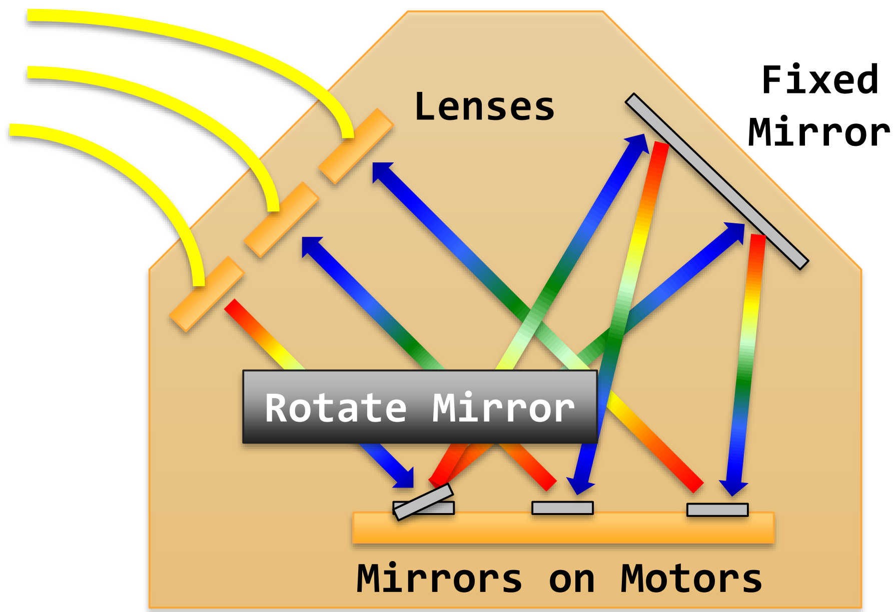
Prototype 3

Changing lambdas (ns)

Example

Optical Circuit Switch

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



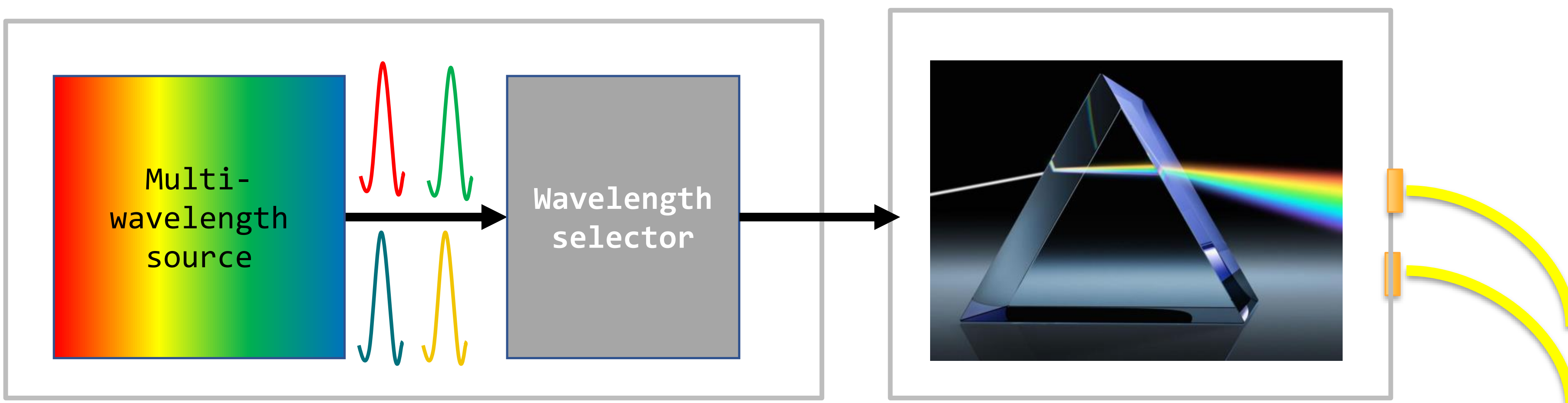
Optical Circuit Switch

By Nathan Farrington, SIGCOMM 2010

Another Example

Tunable Lasers

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



Electrical switch
with tunable laser

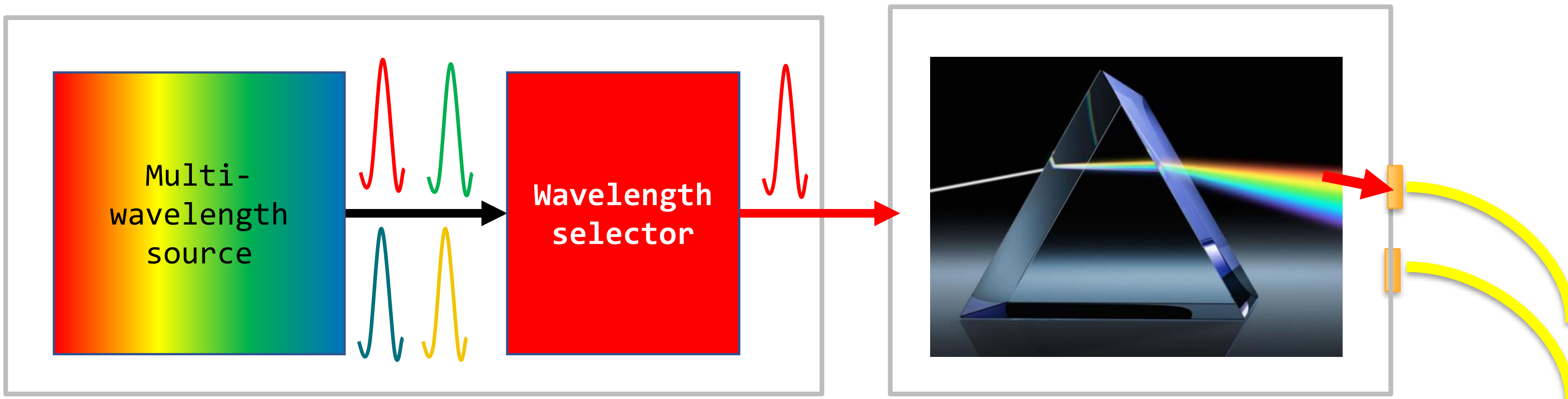
Optical switch
Passive

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

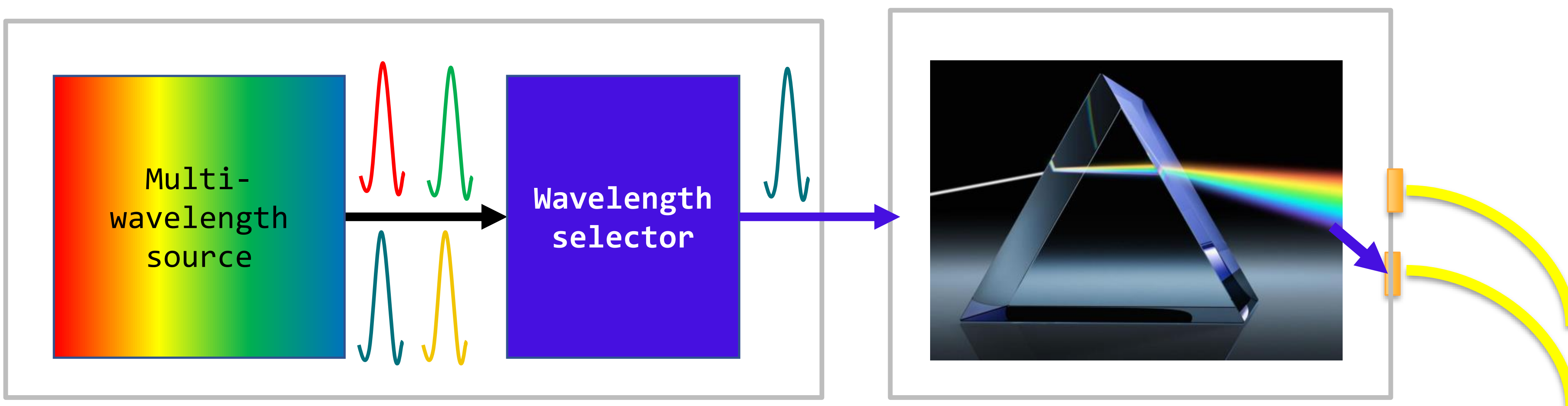
Optical switch ***Passive***

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

First Deployments

E.g., Google’s Datacenter Jupiter


Systems

Jupiter evolving: Reflecting on Google’s data center network transformation

August 24, 2022

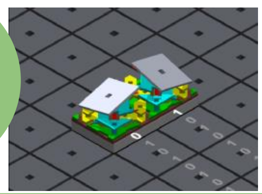
Amin Vahdat

VP & GM, Systems and Services Infrastructure

The Big Picture

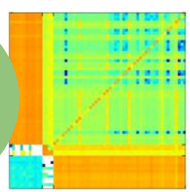
Flexibility

New!



Structure

More!



Self-Adjusting Networks

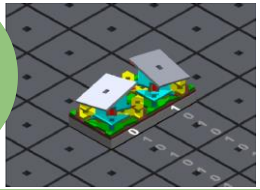
Efficiency

Now is the time!

The Big Picture

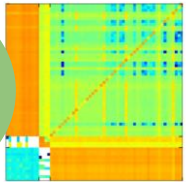
New!

Flexibility



More!

Structure



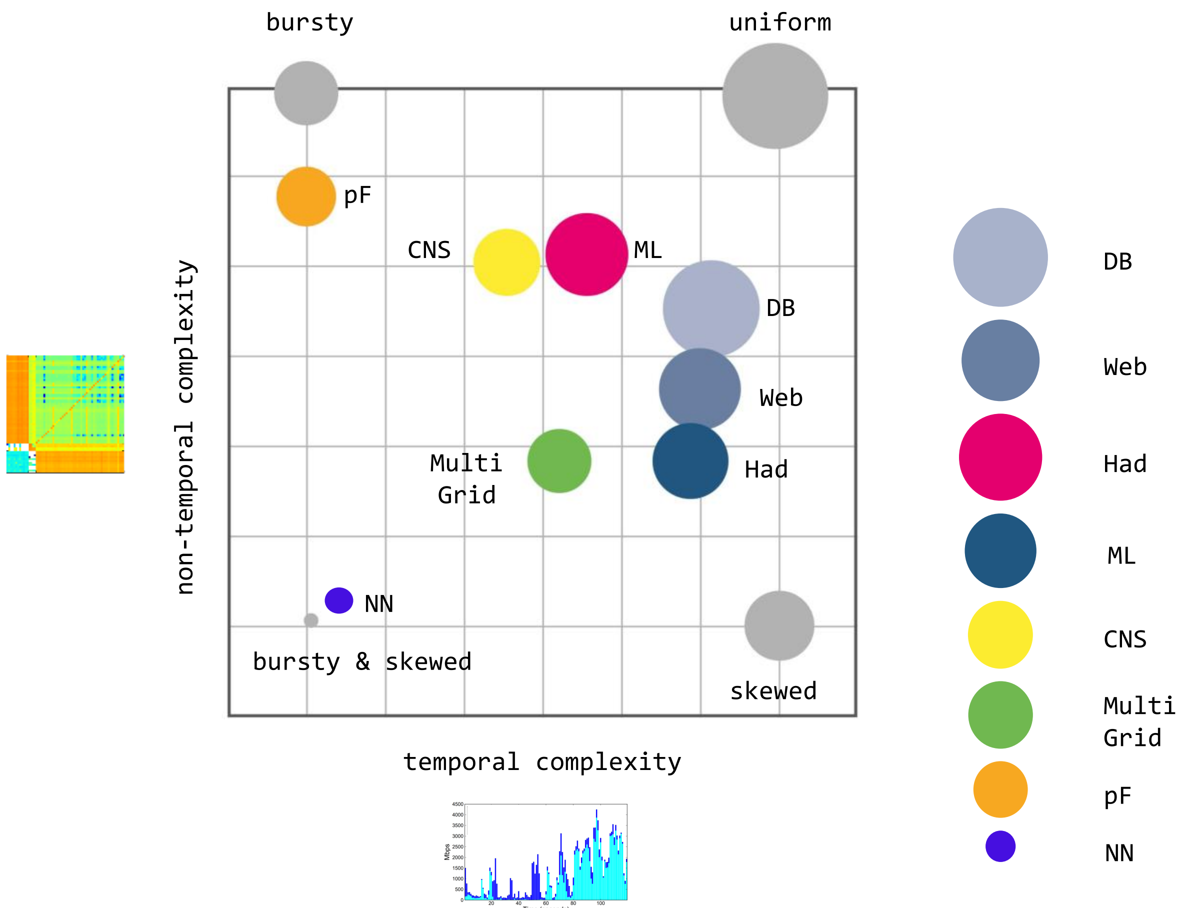
Self-Adjusting Networks

Now is the time!

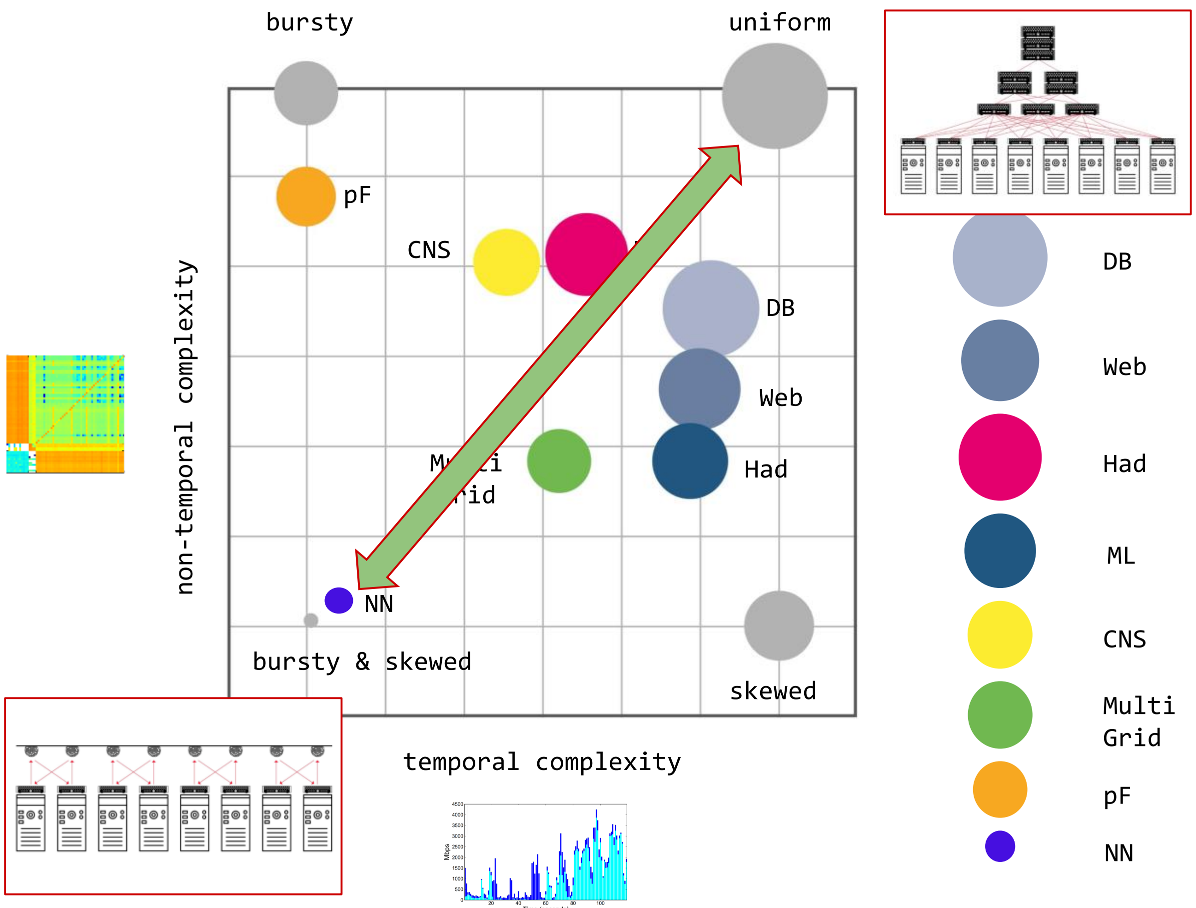
Efficiency

Missing: Foundations of demand-aware, self-adjusting networks.

Potential Gain



Potential Gain



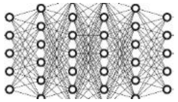
Unique Position

Demand-Aware, Self-Adjusting Systems

Everywhere, but mainly in software

Algorithmic trading

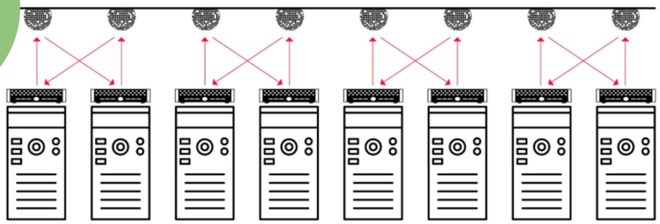
Recommender systems



Neural networks

VS

Our focus in this talk: in hardware



The Natural Question:

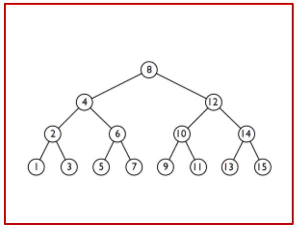
Given This Structure, What Can Be Achieved? Metrics and Algorithms?

A first insight: entropy of the demand.

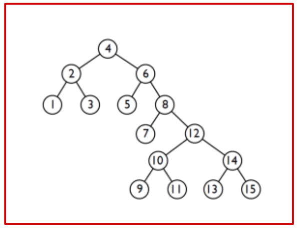
Insight:

Connection to Datastructures

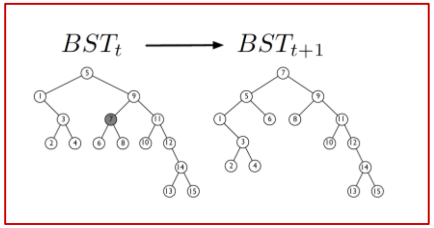
Traditional BST



Demand-aware BST



Self-adjusting BST

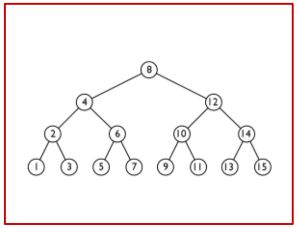


More structure: improved **access cost**

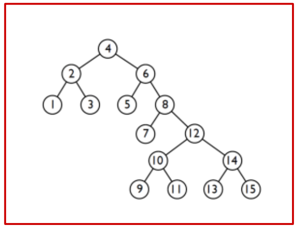
Insight:

Connection to Datastructures & Coding

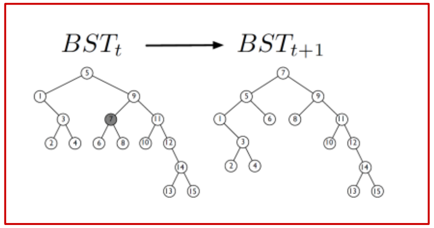
Traditional BST
(Worst-case coding)



Demand-aware BST
(Huffman coding)



Self-adjusting BST
(Dynamic Huffman coding)

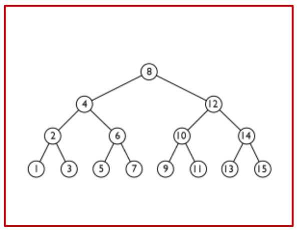


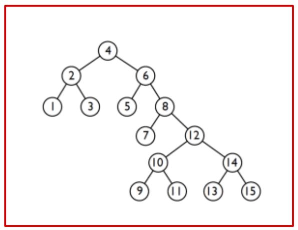
More structure: improved **access cost** / shorter **codes**

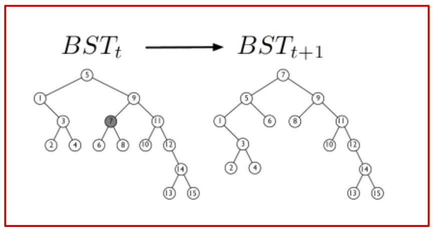
Insight:

Connection to Datastructures & Coding

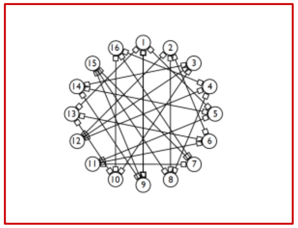
Traditional BST (Worst-case coding)	Demand-aware BST (Huffman coding)	Self-adjusting BST (Dynamic Huffman coding)

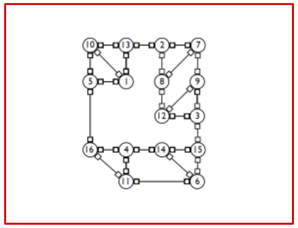


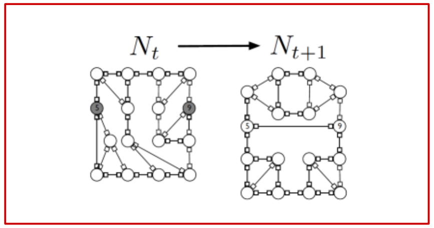




More structure: improved **access cost** / shorter **codes**





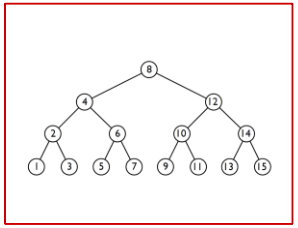


Similar **benefits**?

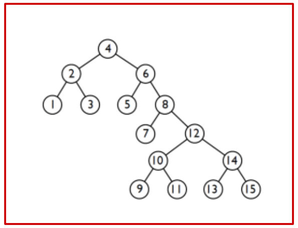
Insight:

Connection to Datastructures & Coding

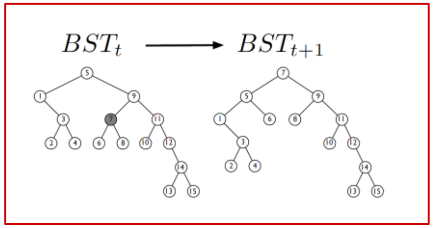
Traditional BST
(Worst-case coding)



Demand-aware BST
(Huffman coding)

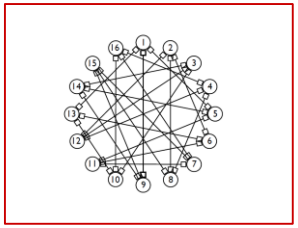


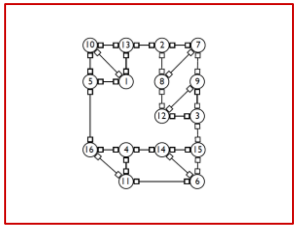
Self-adjusting BST
(Dynamic Huffman coding)

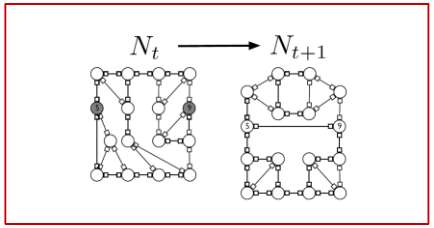


More than an analogy!

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





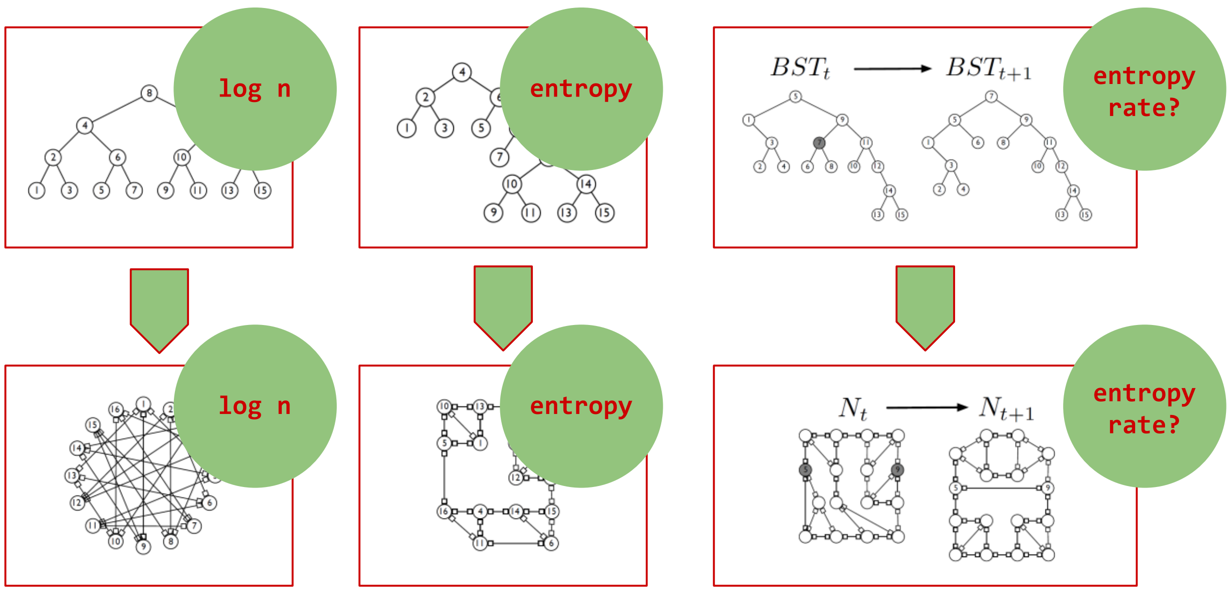


Similar **benefits**?

Insight:

Connection to Datastructures & Coding

Traditional BST (Worst-case coding) | Demand-aware BST (Huffman coding) | Self-adjusting BST (Dynamic Huffman coding)



Reduced expected **route lengths**!

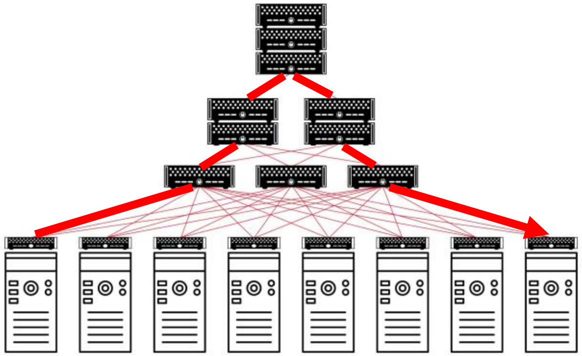
More than an analogy!

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

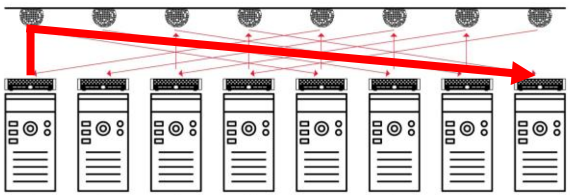
First results:
Demand-aware networks of asymptotically optimal route lengths.

Reality: A Tradeoff

⇀ Self-adjusting networks may be really useful to serve large flows (elephant flows): avoiding multi-hop routing



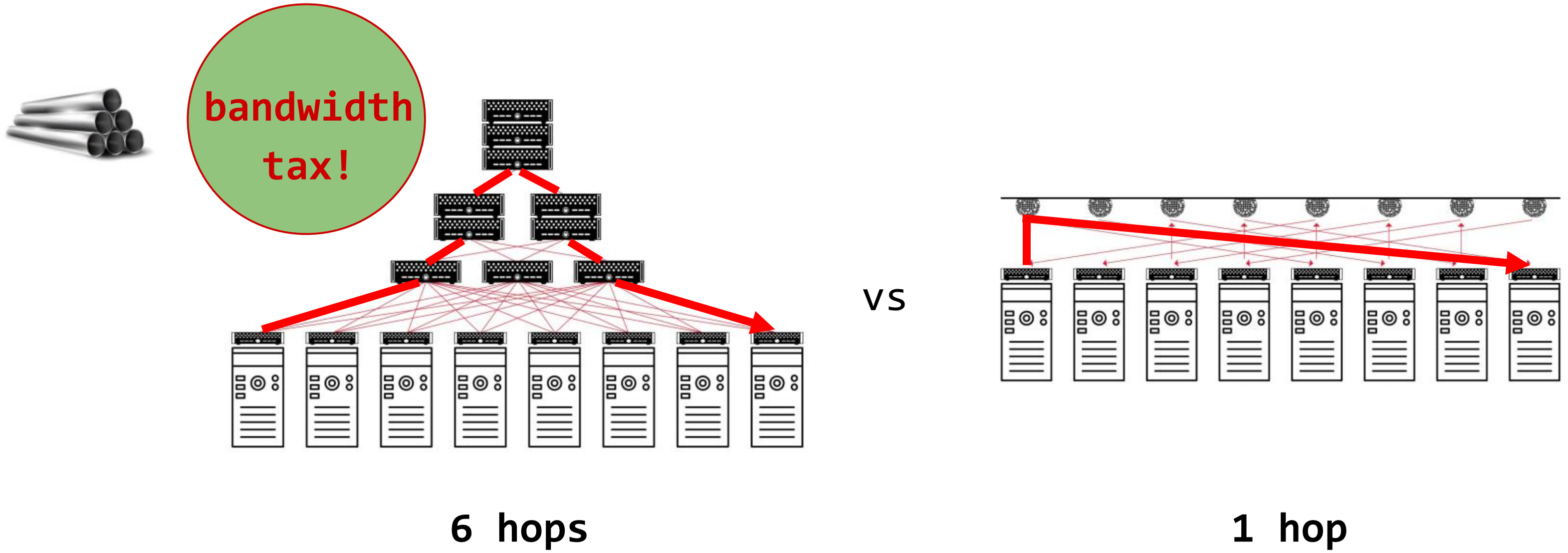
vs



6 hops **1 hop**

Reality: A Tradeoff

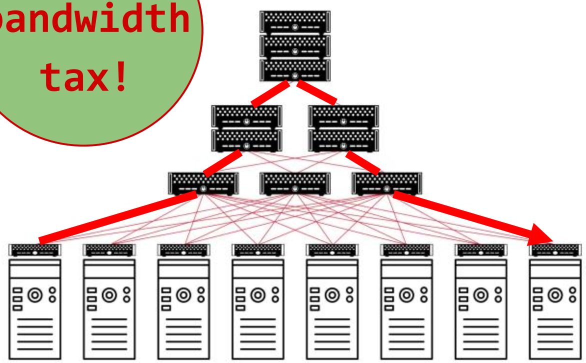
→ Self-adjusting networks may be really useful to serve large flows (elephant flows): avoiding multi-hop routing



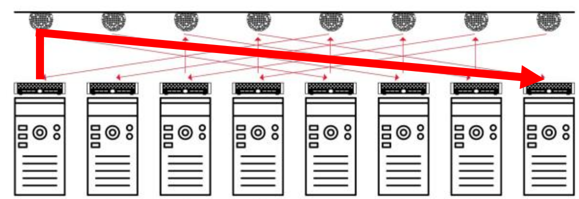
Reality: A Tradeoff

→ Self-adjusting networks may be really useful to serve large flows (elephant flows): avoiding multi-hop routing

bandwidth tax!



vs



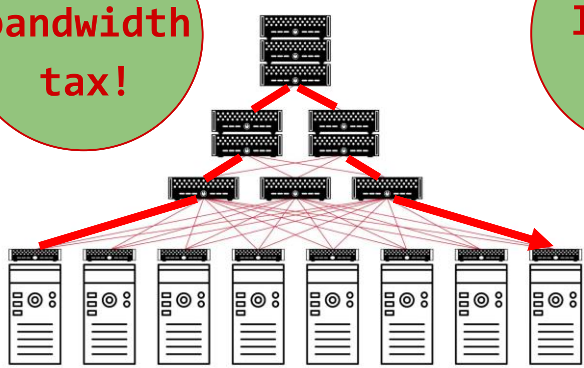
6 hops **1 hop**

→ However, requires optimization and adaption, which takes time

Reality: A Tradeoff

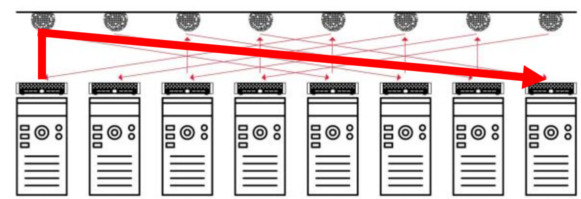
⇀ Self-adjusting networks may be really useful to serve large flows (elephant flows): avoiding multi-hop routing

bandwidth tax!



latency tax!

vs



6 hops **1 hop**

⇀ However, requires optimization and adaption, which takes time

Challenge: 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

Opportunity: Tech Diversity

Diverse topology components:

→ demand-oblivious and
demand-aware

Demand-oblivious

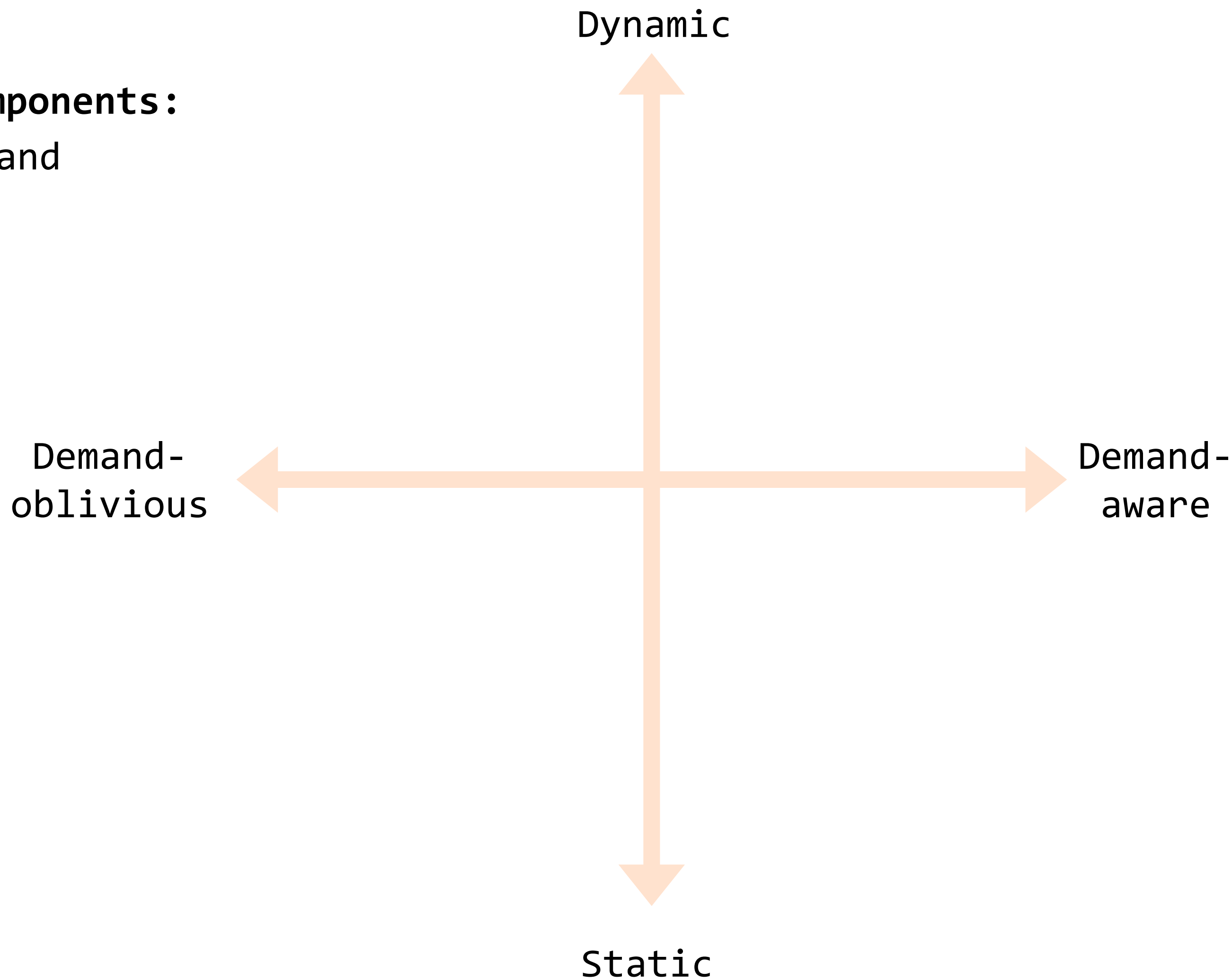
Demand-aware

Opportunity: Tech Diversity

Diverse topology components:

⇀ demand-oblivious and
  demand-aware

⇀ static vs dynamic

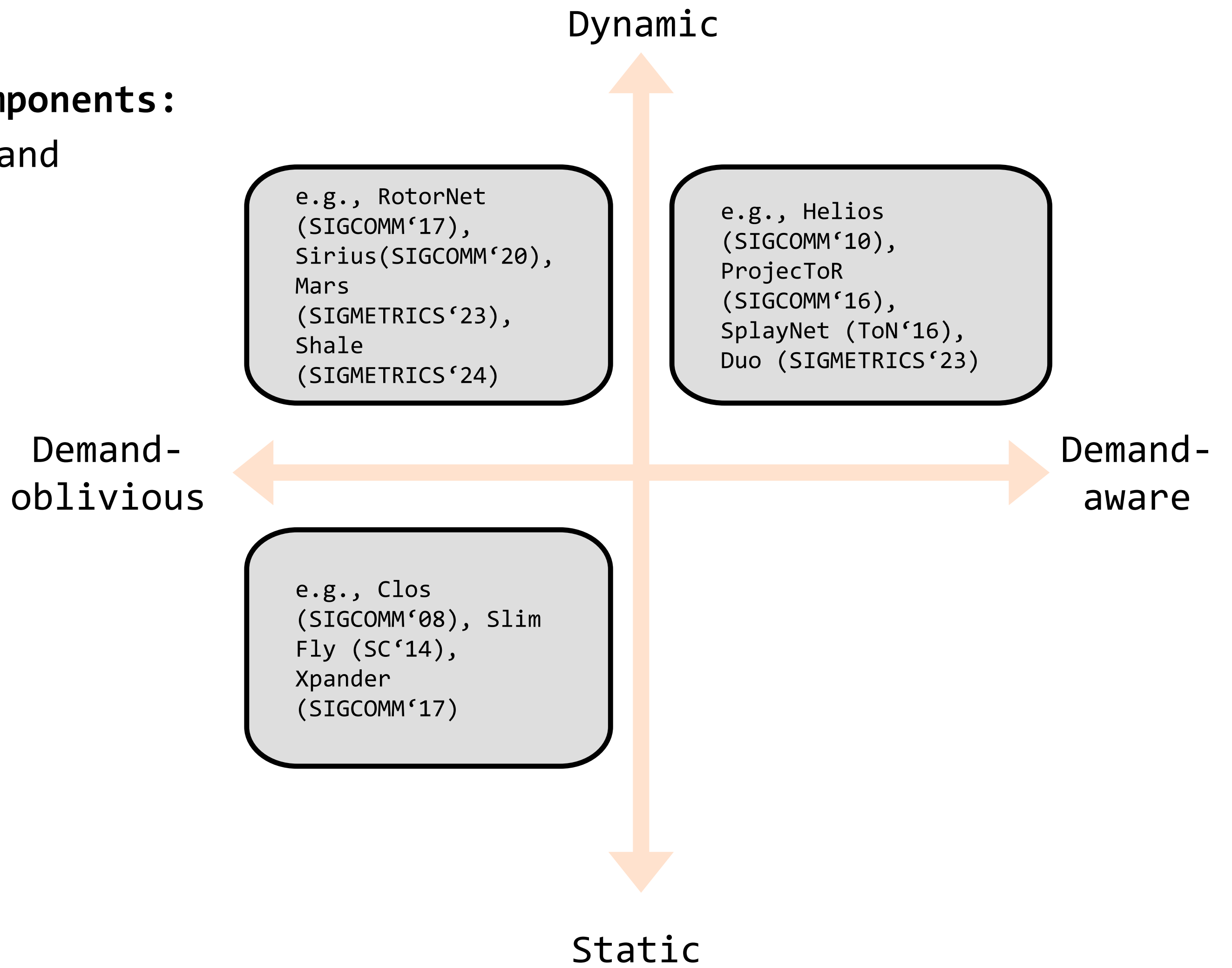


Opportunity: Tech Diversity

Diverse topology components:

⇀ demand-oblivious and demand-aware

⇀ static vs dynamic

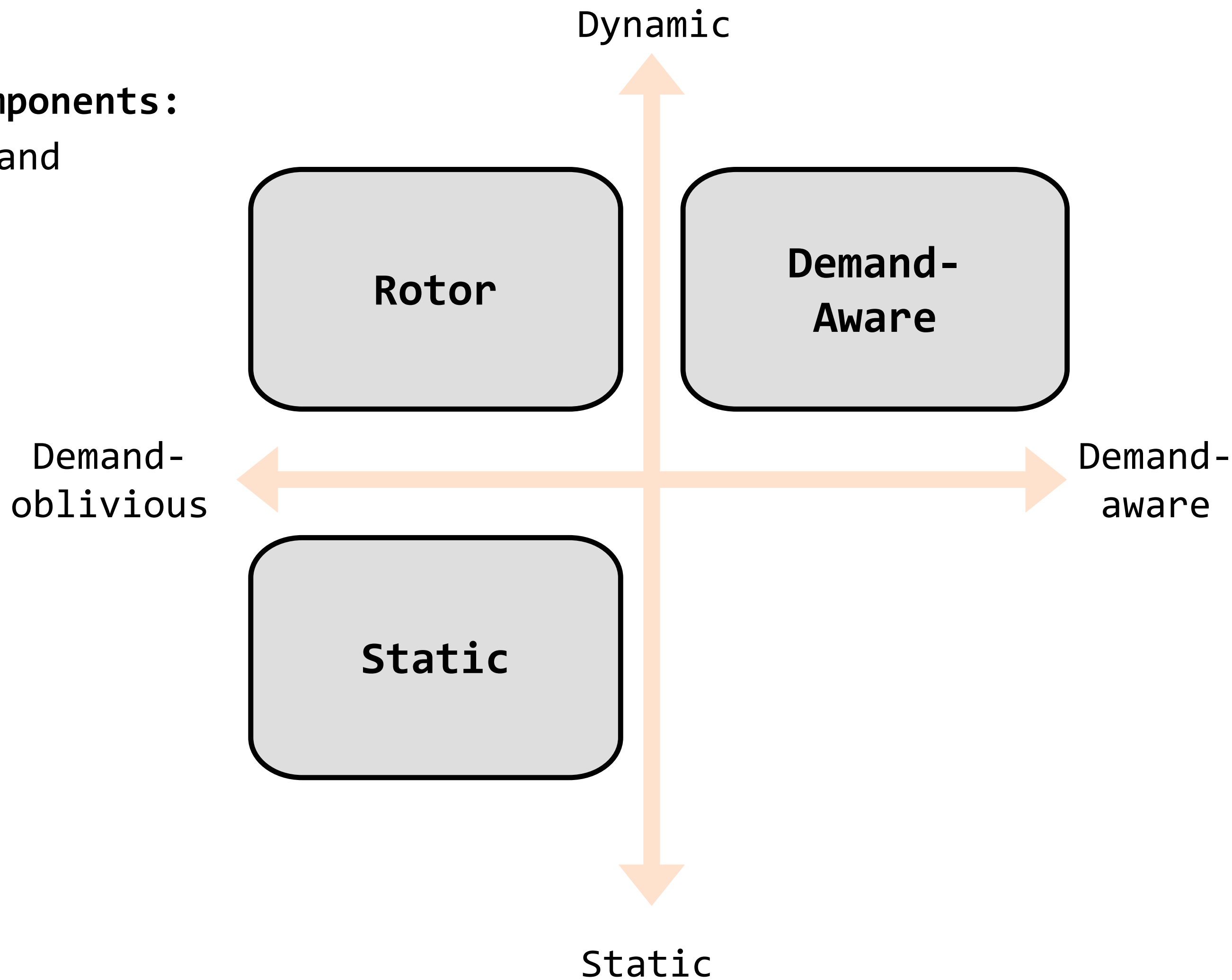


Opportunity: Tech Diversity

Diverse topology components:

⇀ demand-oblivious and
demand-aware

⇀ static vs dynamic

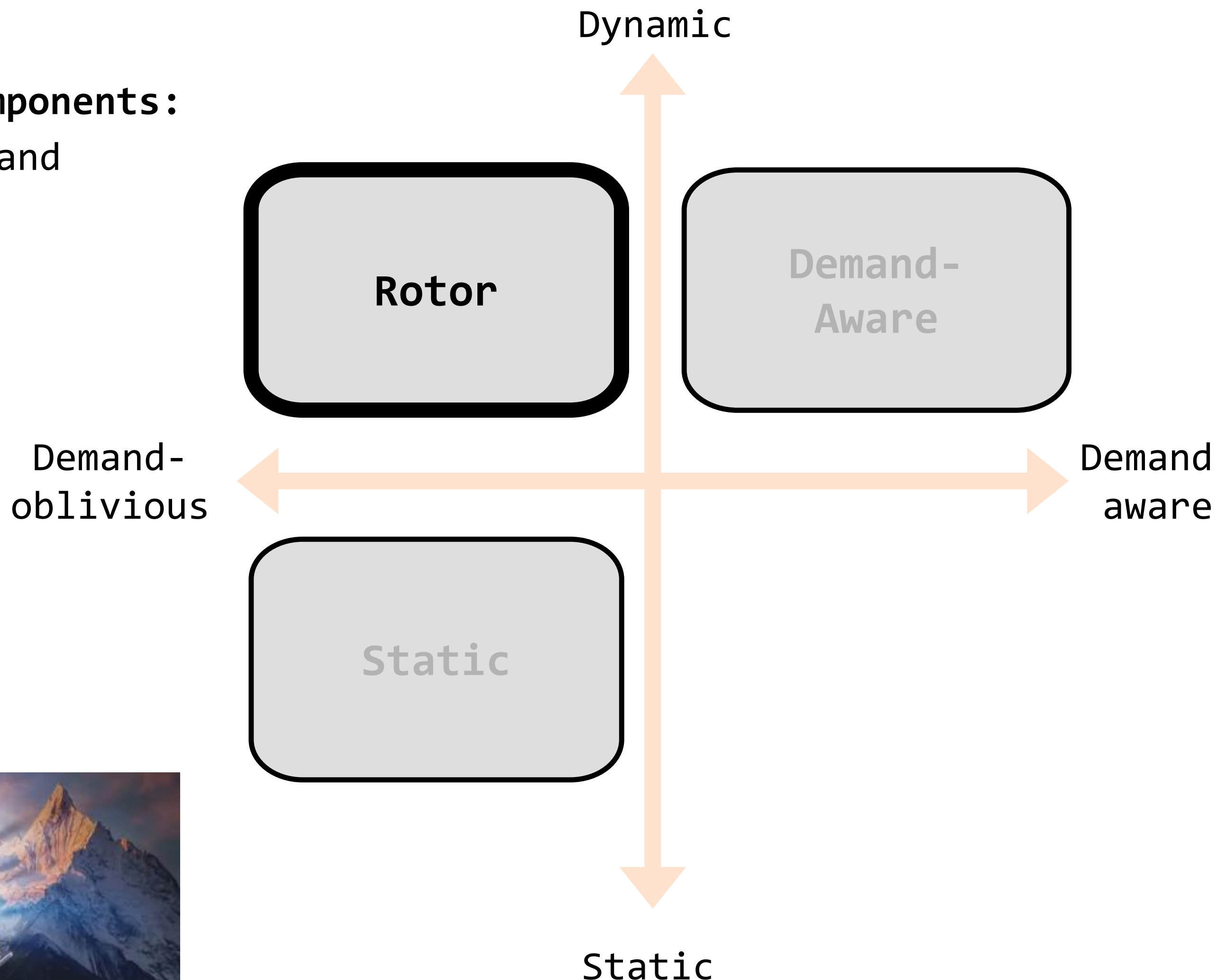


Opportunity: Tech Diversity

Diverse topology components:

→ demand-oblivious and demand-aware

→ static vs dynamic

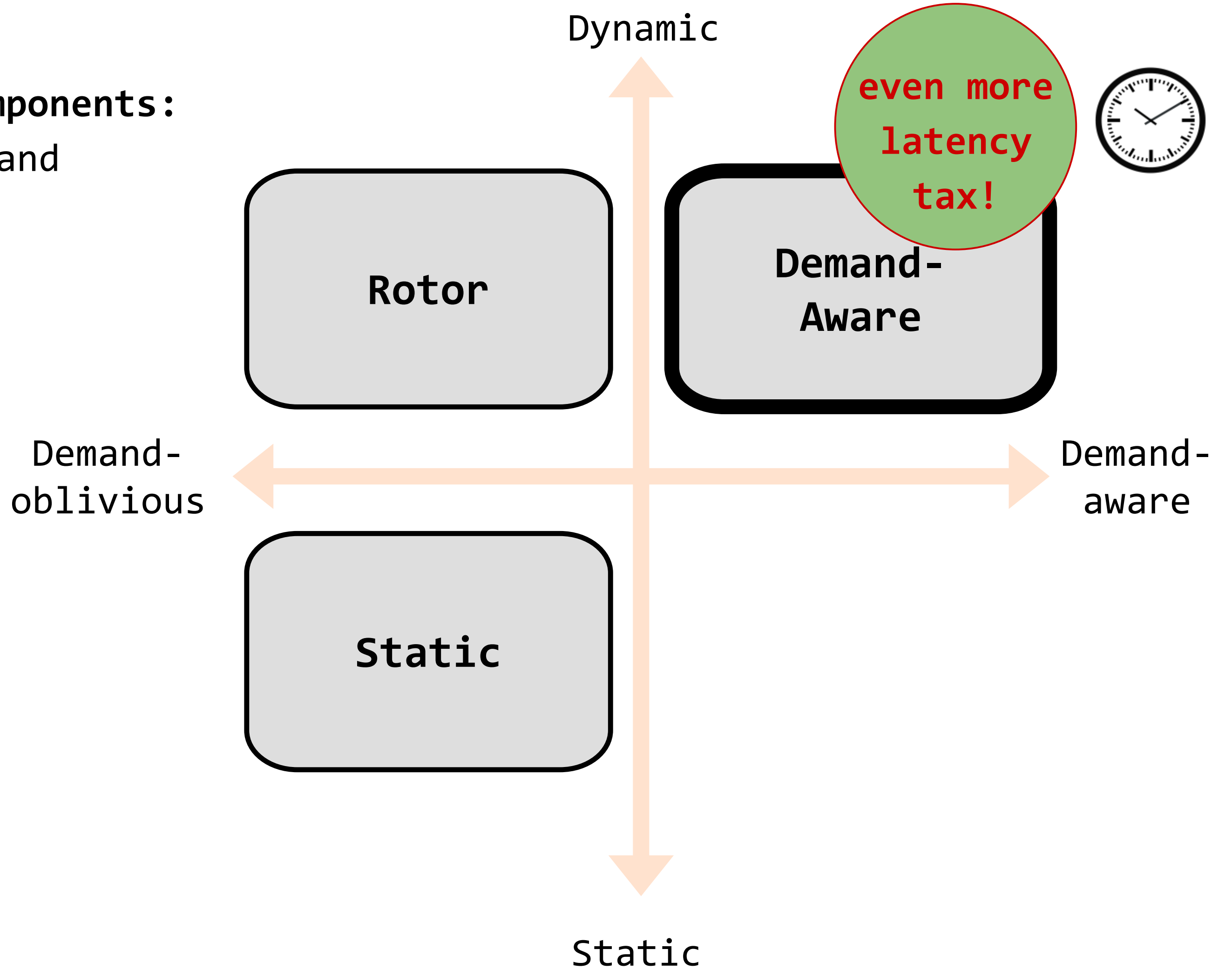


Opportunity: Tech Diversity

Diverse topology components:

⇀ demand-oblivious and demand-aware

⇀ static vs dynamic



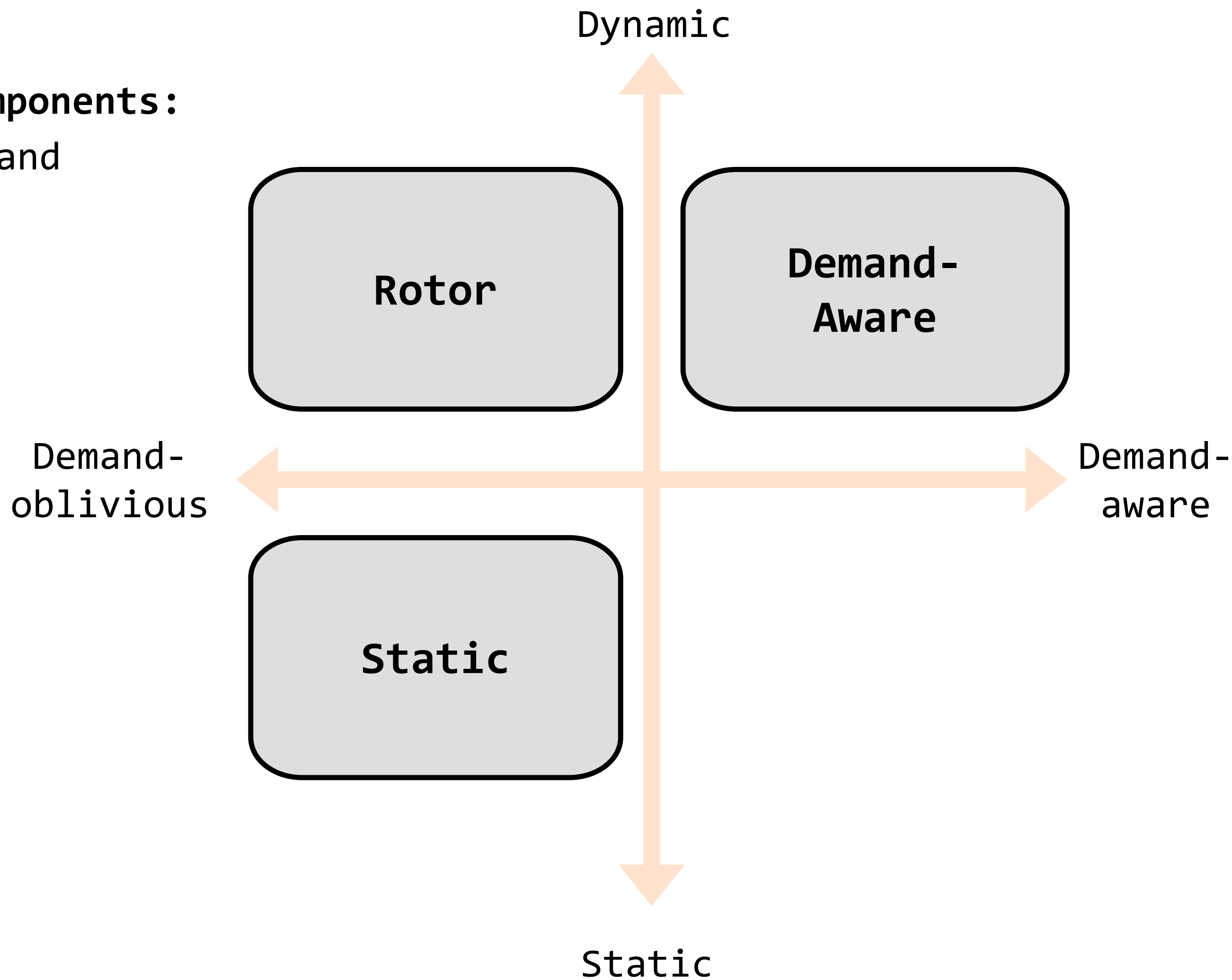
Opportunity: Tech Diversity

Diverse topology components:

⇀ demand-oblivious and demand-aware

⇀ static vs dynamic

Which approach is best?

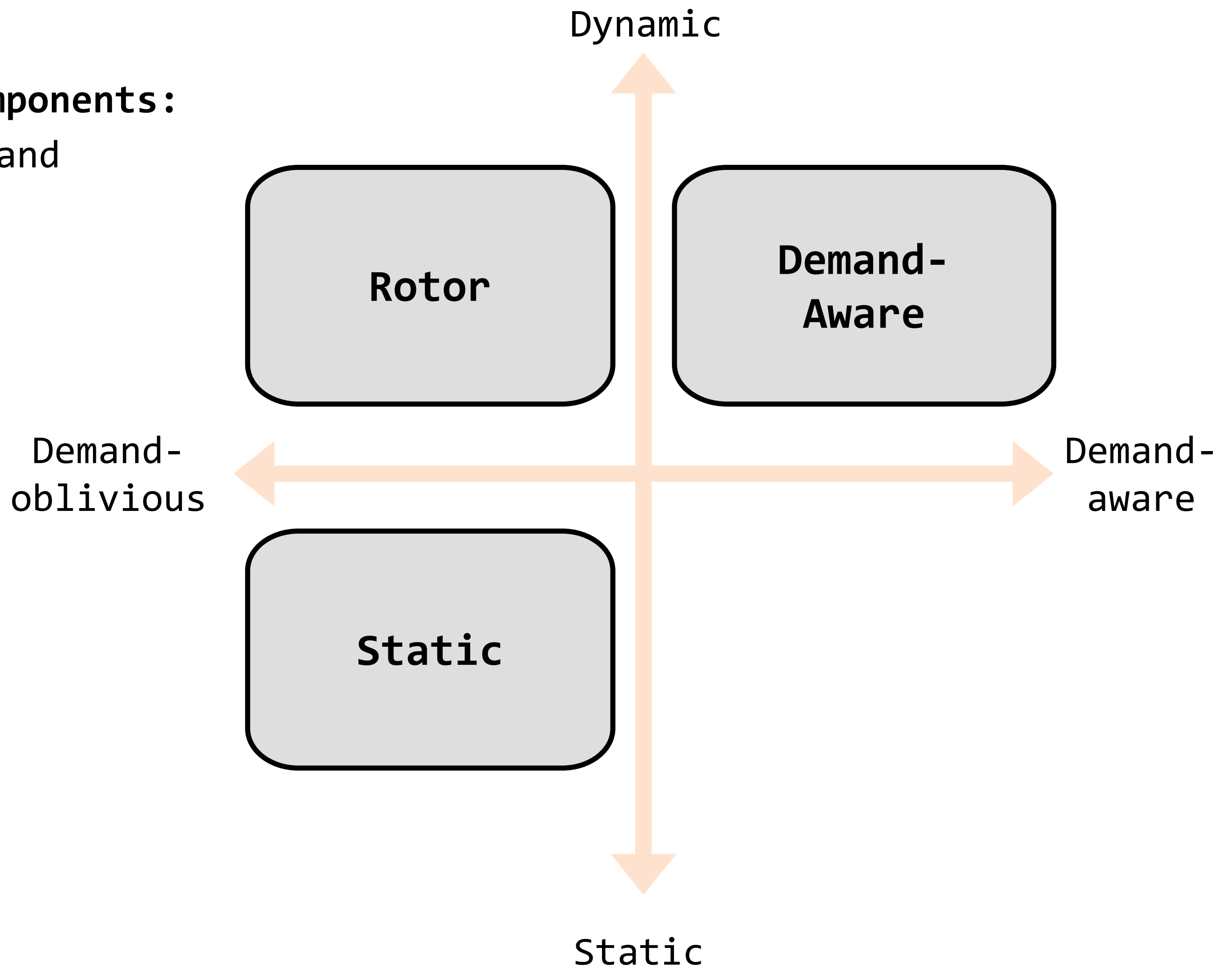


Opportunity: Tech Diversity

Diverse topology components:

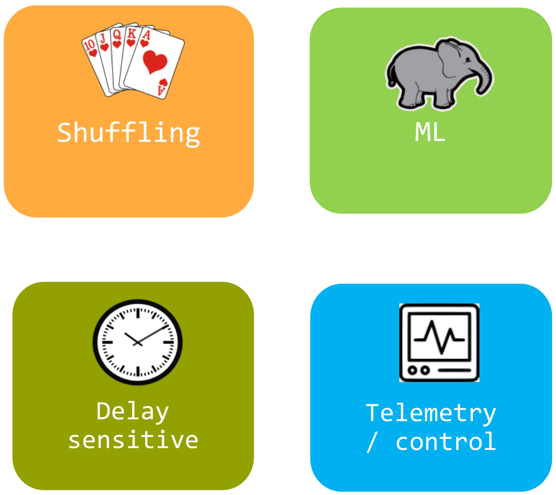
⇀ demand-oblivious and
  demand-aware

⇀ static vs dynamic

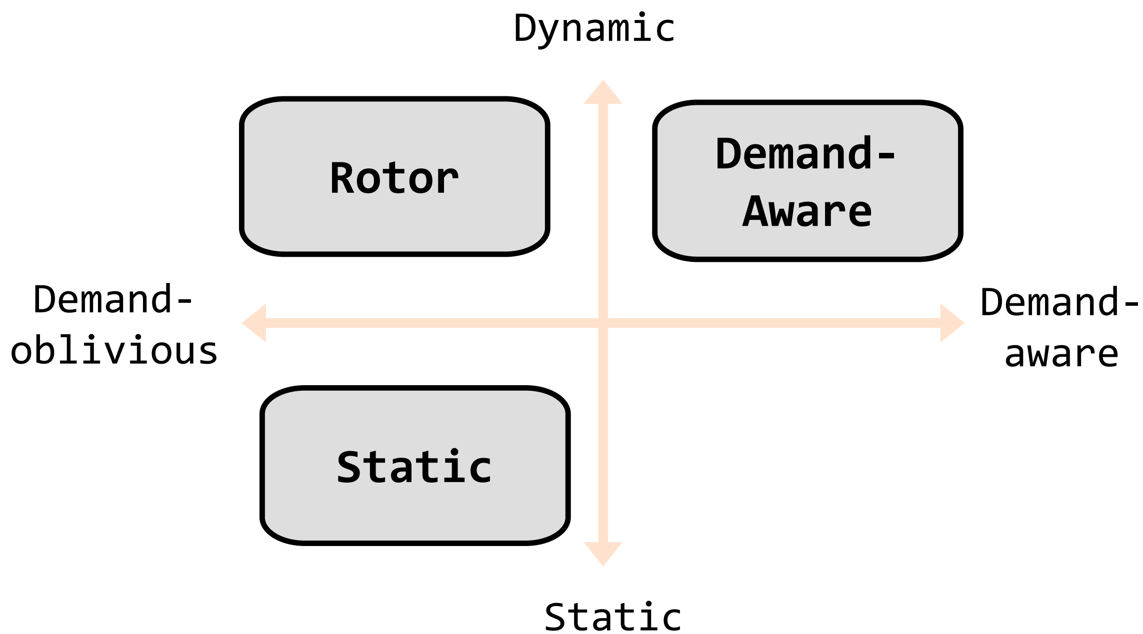


As always in CS:
It depends…

Examples:
Match or Mismatch?

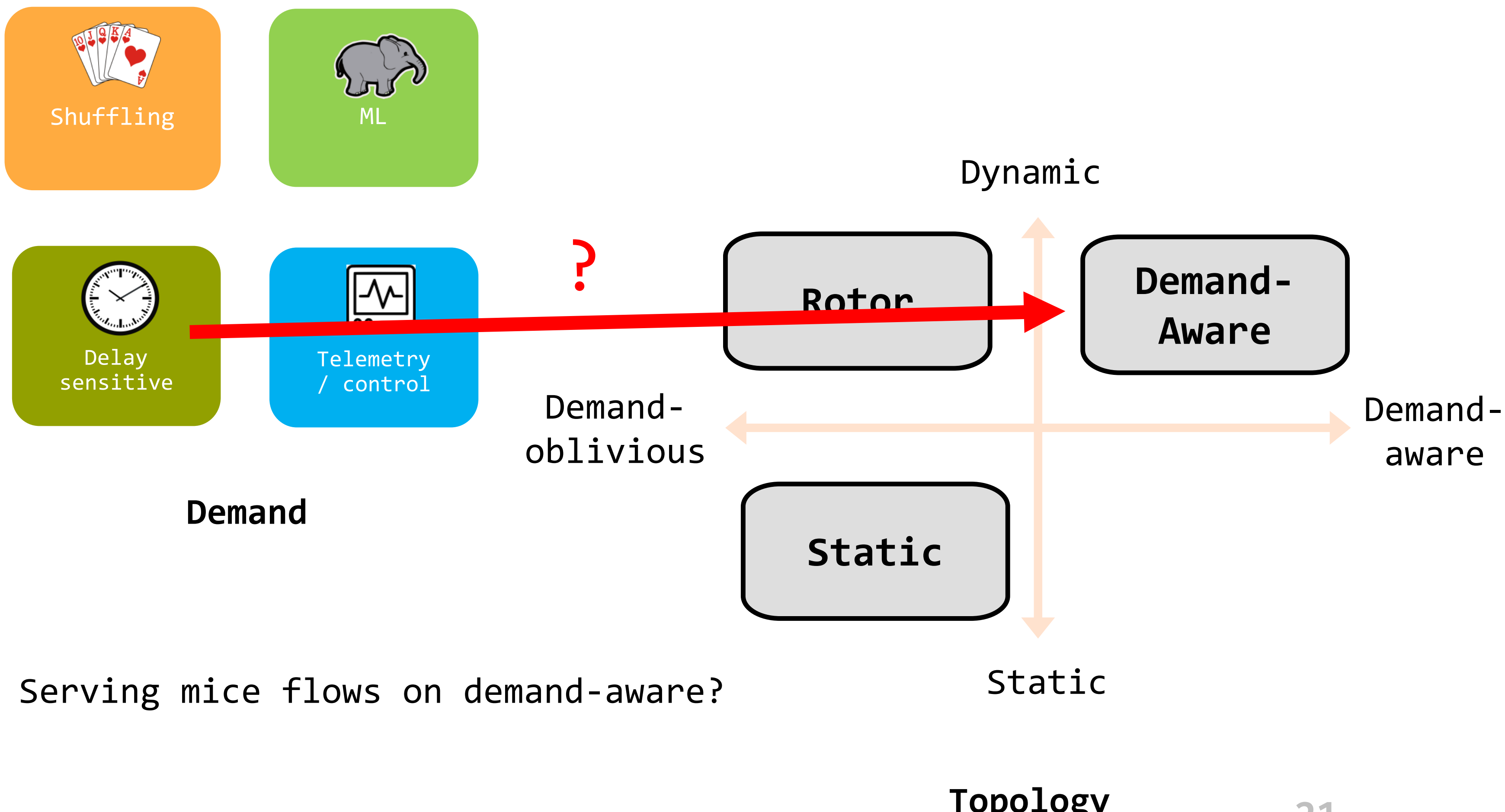


Demand



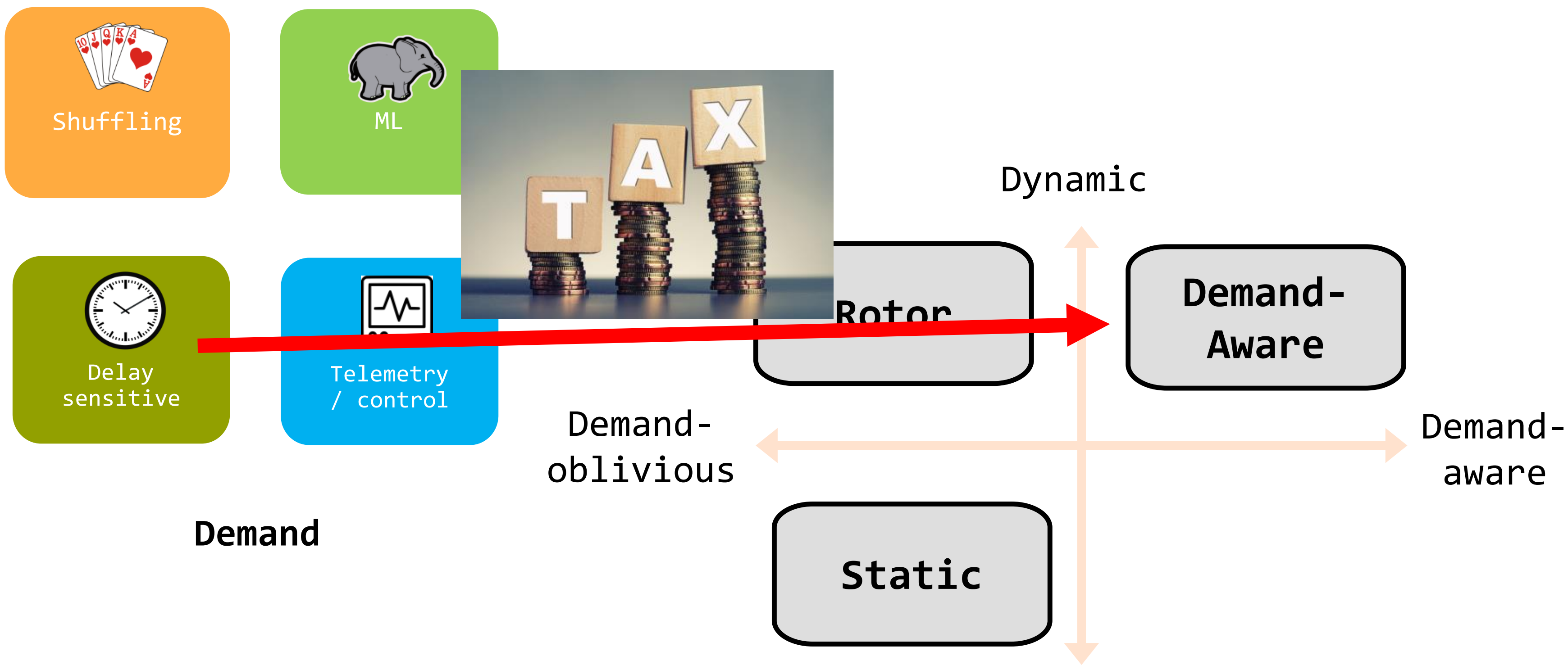
Topology

Examples:
Match or Mismatch?



Serving mice flows on demand-aware?

Examples:
Match or Mismatch?

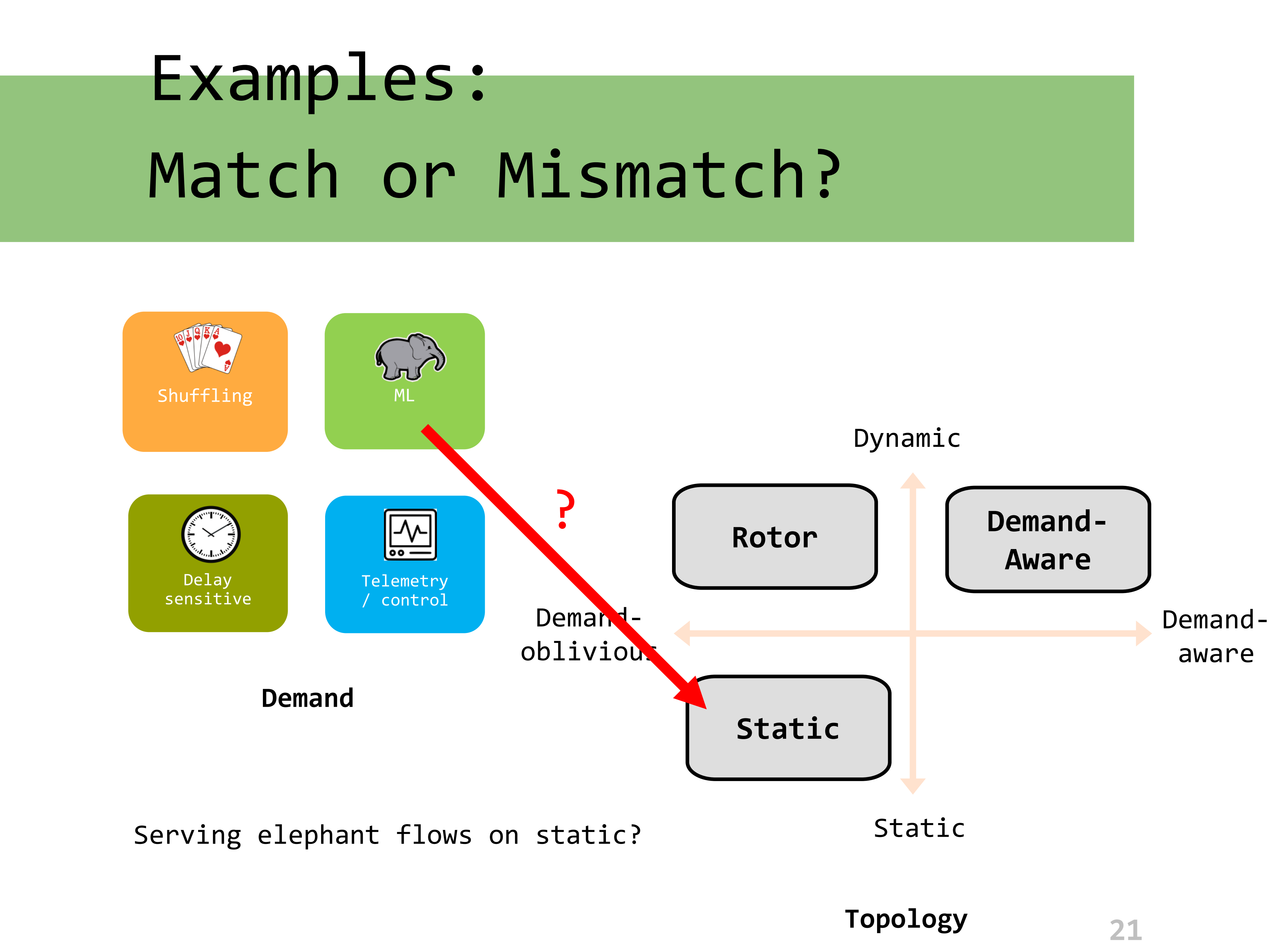


Static

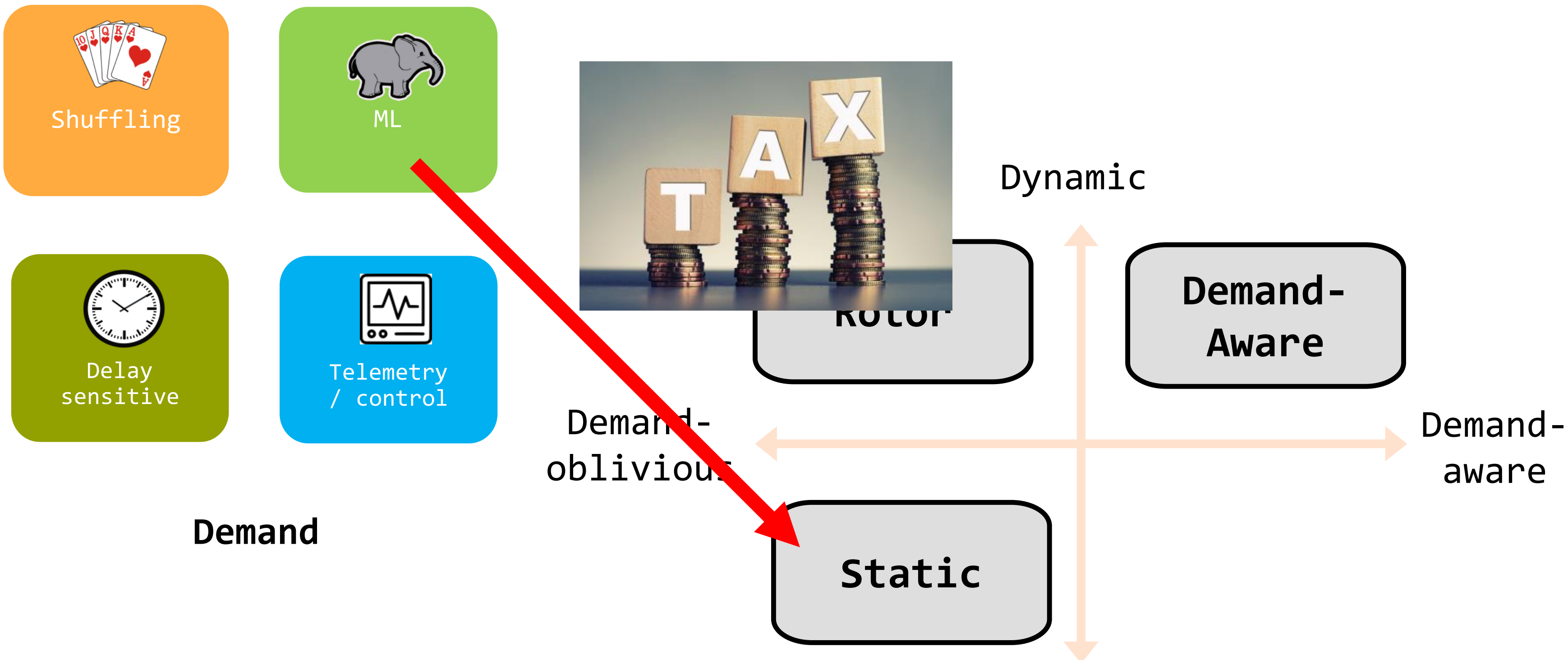
Serving mice flows on demand-aware?

Bad idea! Latency tax.

Topology


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?
21

Examples:
Match or Mismatch?



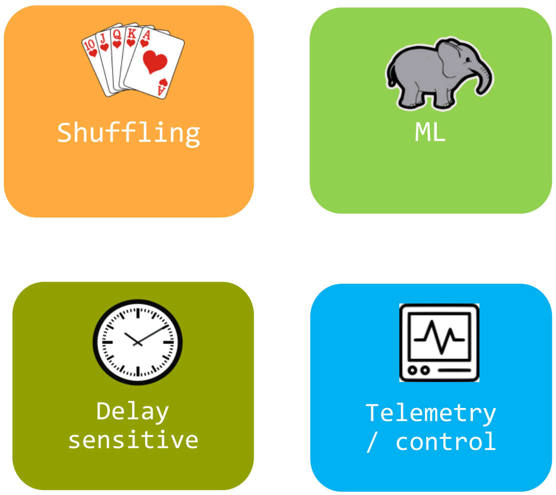
Static

Serving elephant flows on static?

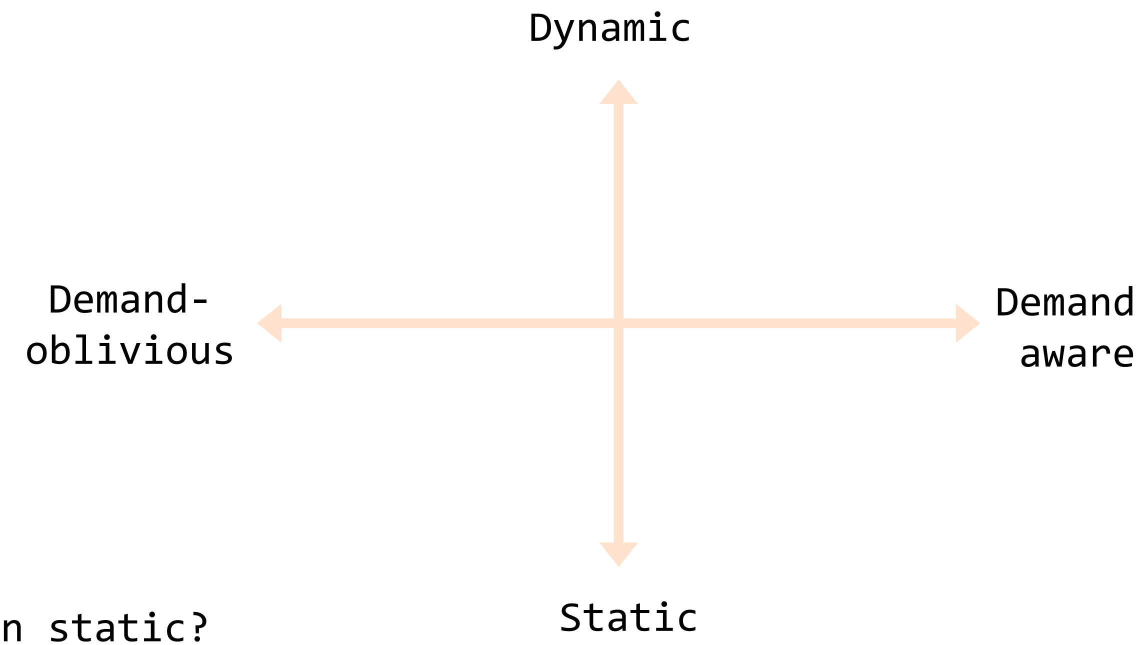
Bad idea! Bandwidth tax.

Topology

Examples:
Match or Mismatch?



Demand

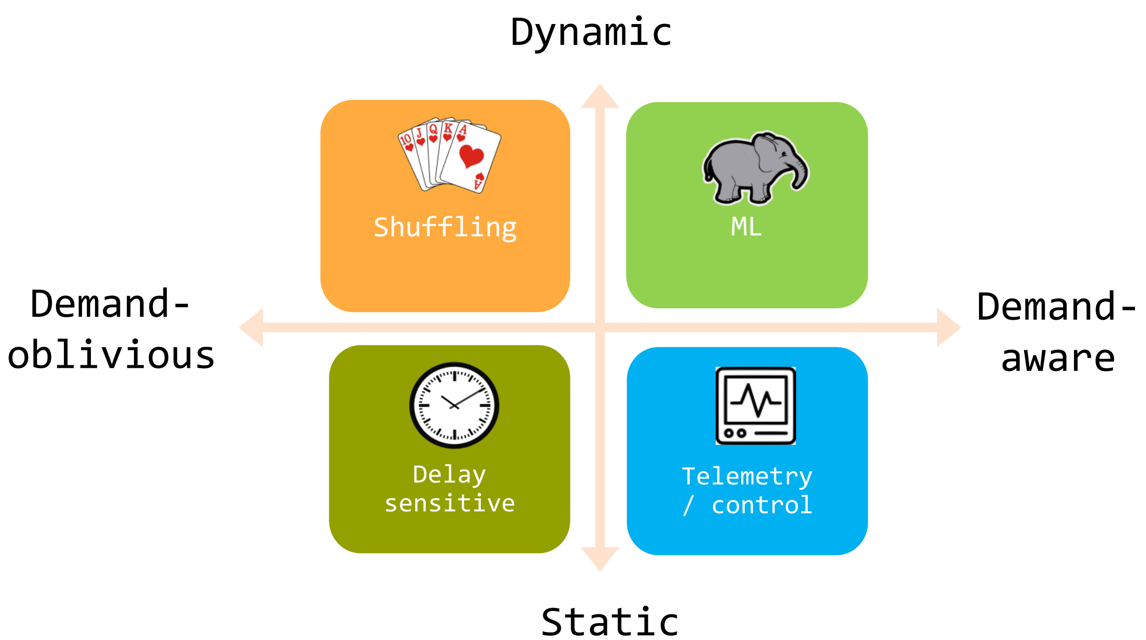


Serving elephant flows on static?

Bad idea! Bandwidth tax.

Topology

Conceptual Solution



Conceptually ideal solution:

Cerberus*** serves traffic on the “best topology”!

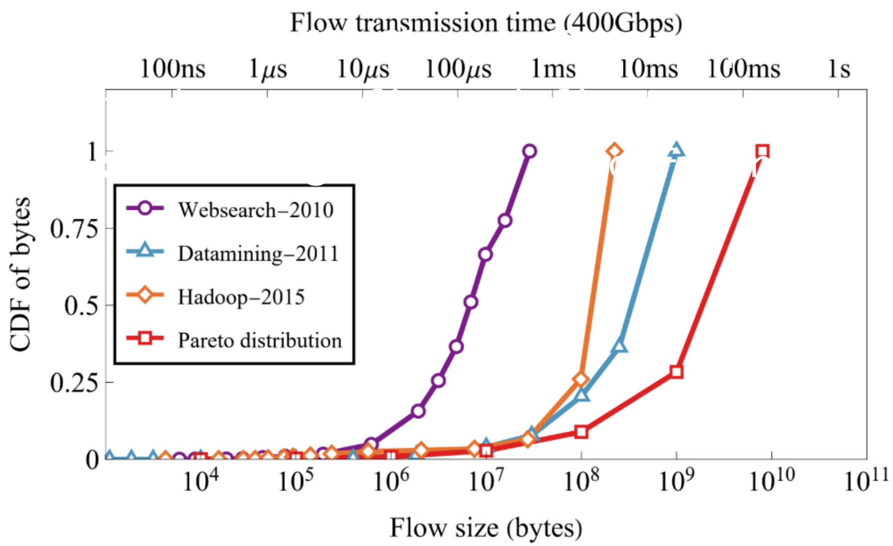
* Griner et al., ACM SIGMETRICS 2022

Flow Size Matters

On what should topology type depend? We argue: flow size.

Flow Size Matters

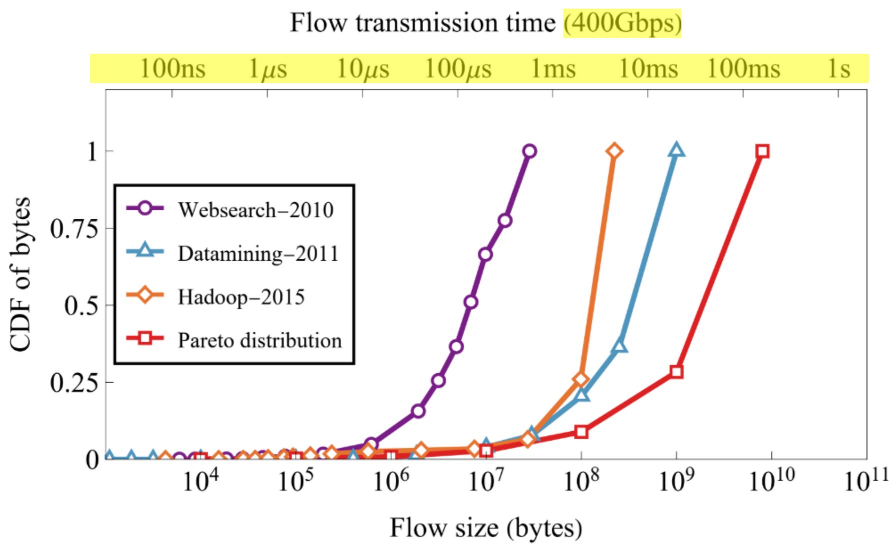
On what should topology type depend? We argue: flow size.



⇢ **Observation 1:** Different apps have different flow size distributions.

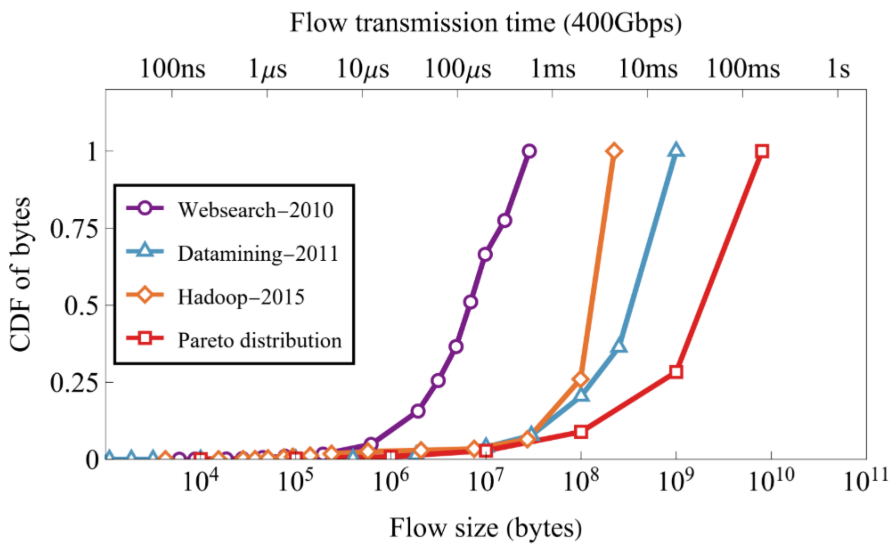
Flow Size Matters

Similar tradeoff for
800Gbps or 1.6 Tpbs



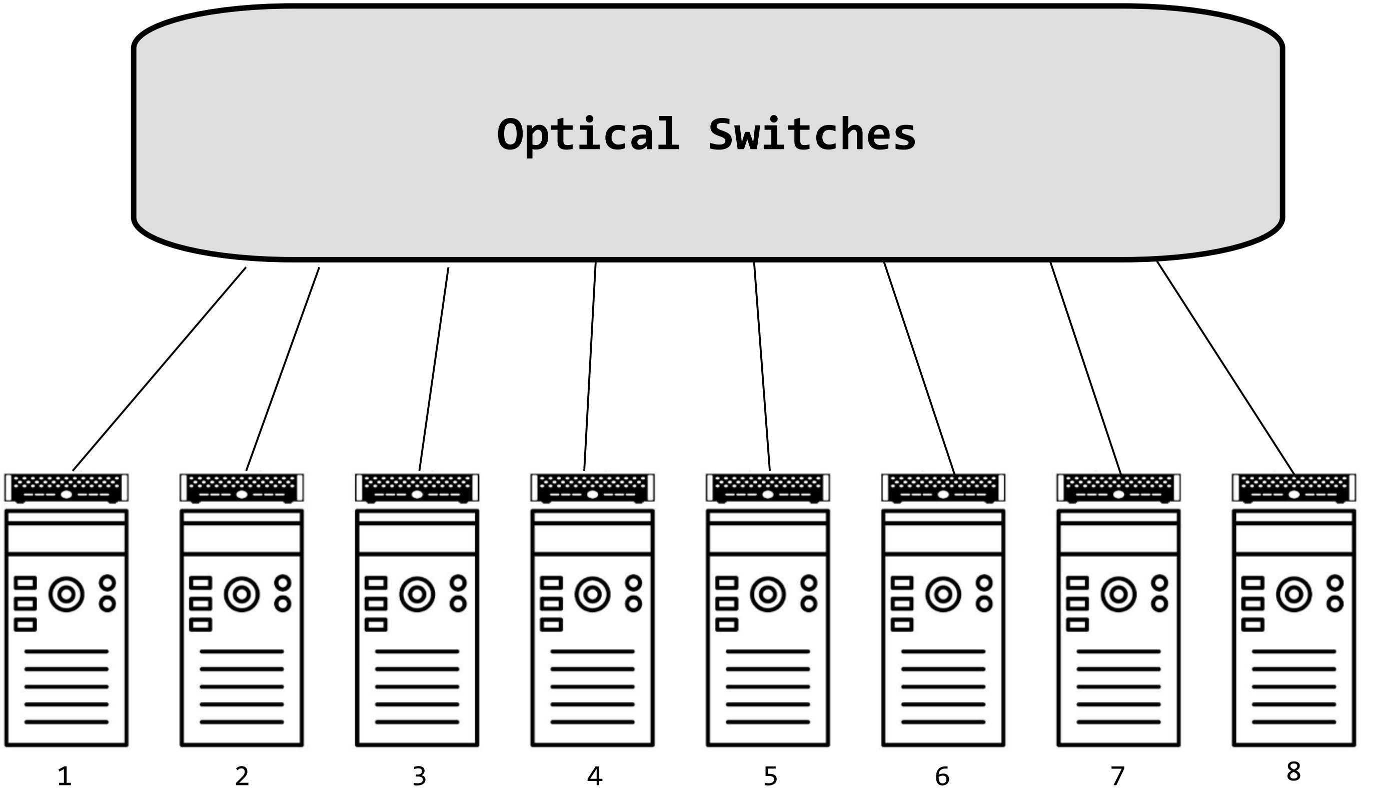
⇢ **Observation 1:** Different apps have different flow size distributions.
⇢ **Observation 2:** The transmission time of a flow depends on its size.

Flow Size Matters

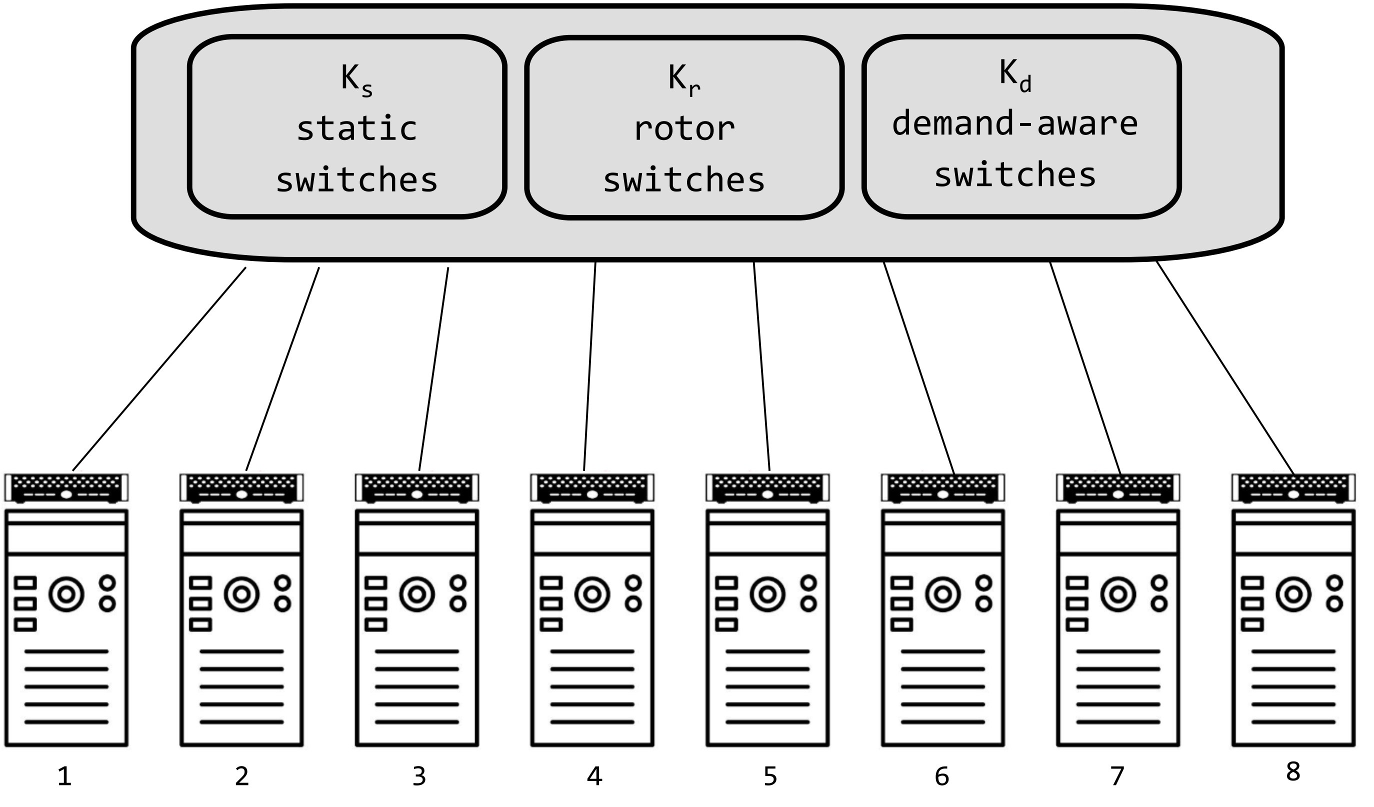


- **Observation 1:** Different apps have different flow size distributions.
- **Observation 2:** The transmission time of a flow depends on its size.
- **Observation 3:** For small flows, flow completion time suffers if network needs to be reconfigured first.
- **Observation 4:** For large flows, reconfiguration time may amortize.

Cerberus

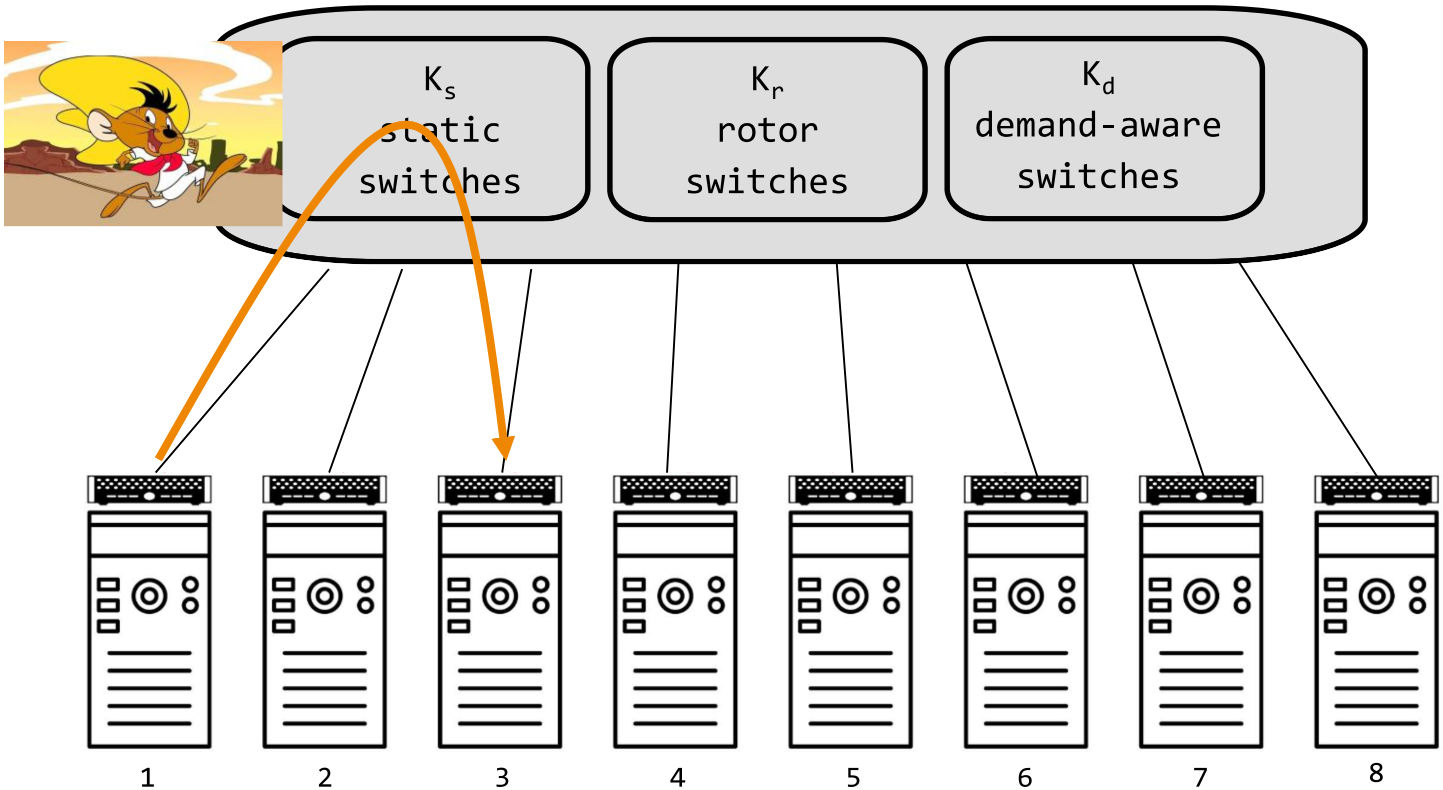


Cerberus



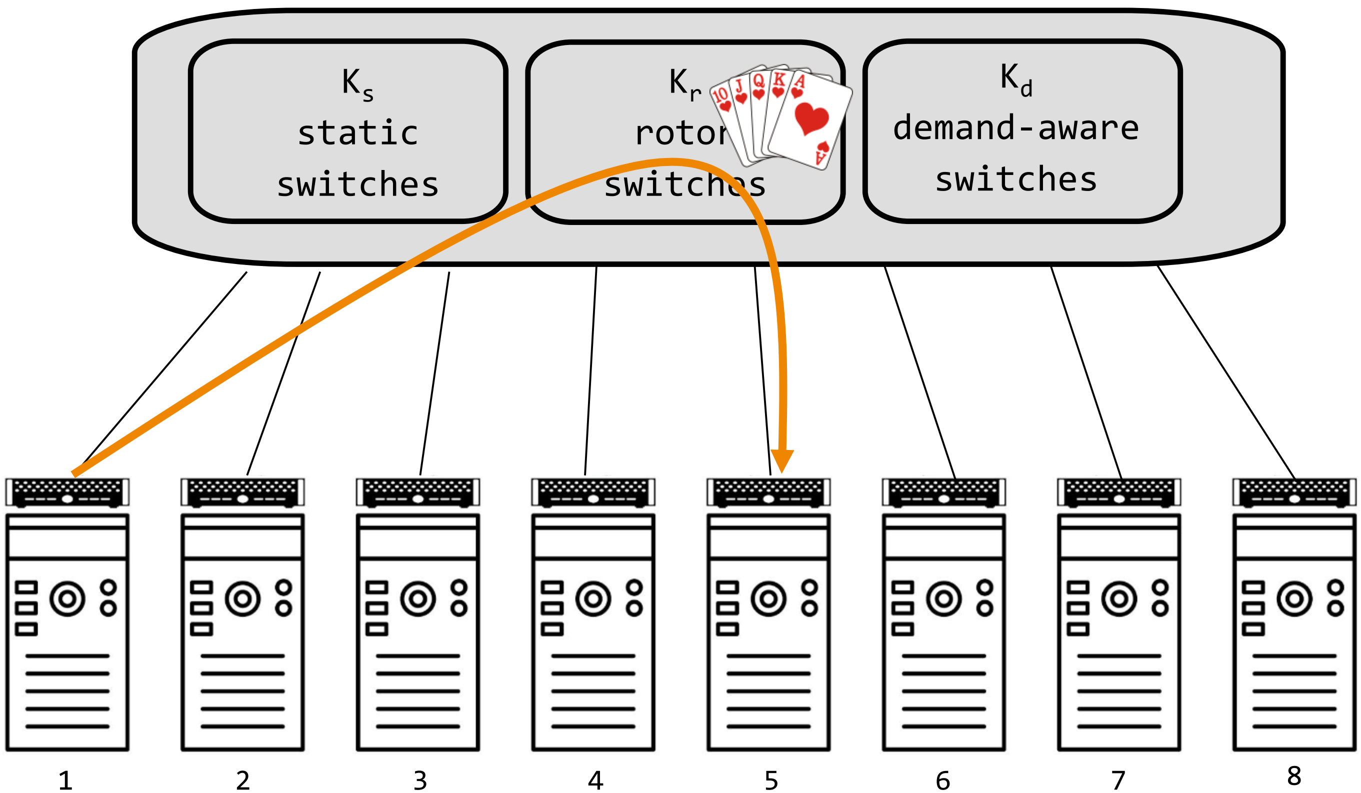
K_s
static
switches
K_r
rotor
switches
K_d
demand-aware
switches
1
2
3
4
5
6
7
8

Cerberus



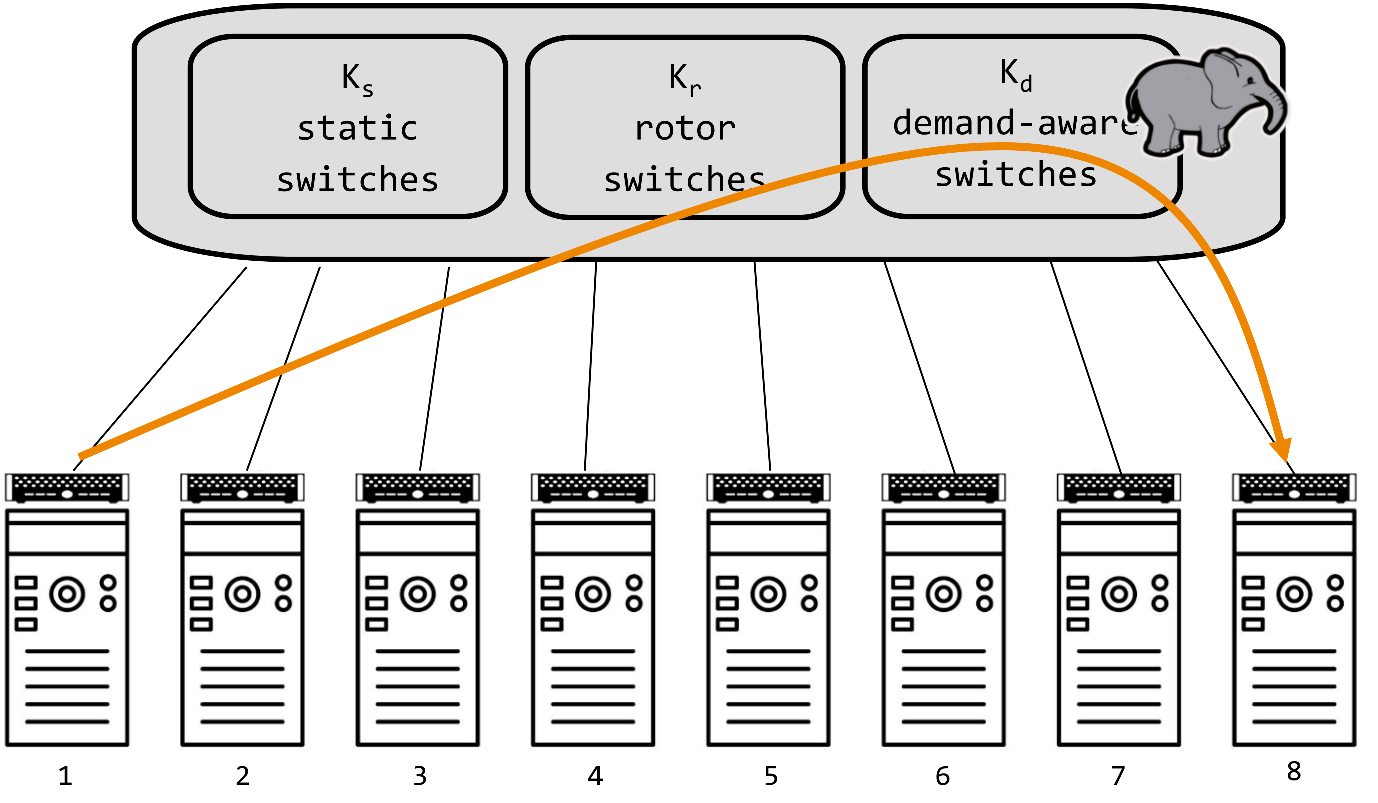
Scheduling: Small flows go via static switches…

Cerberus



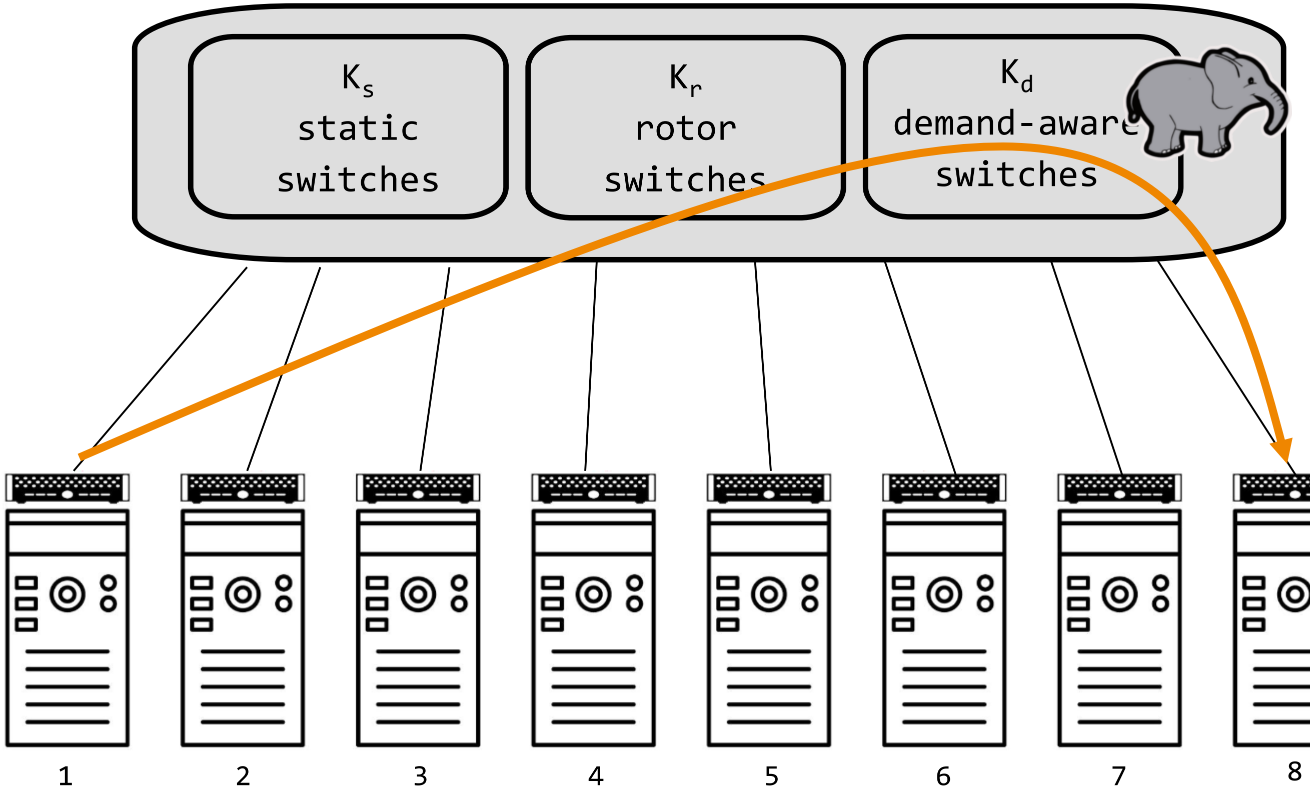
Scheduling: … medium flows via rotor switches…

Cerberus



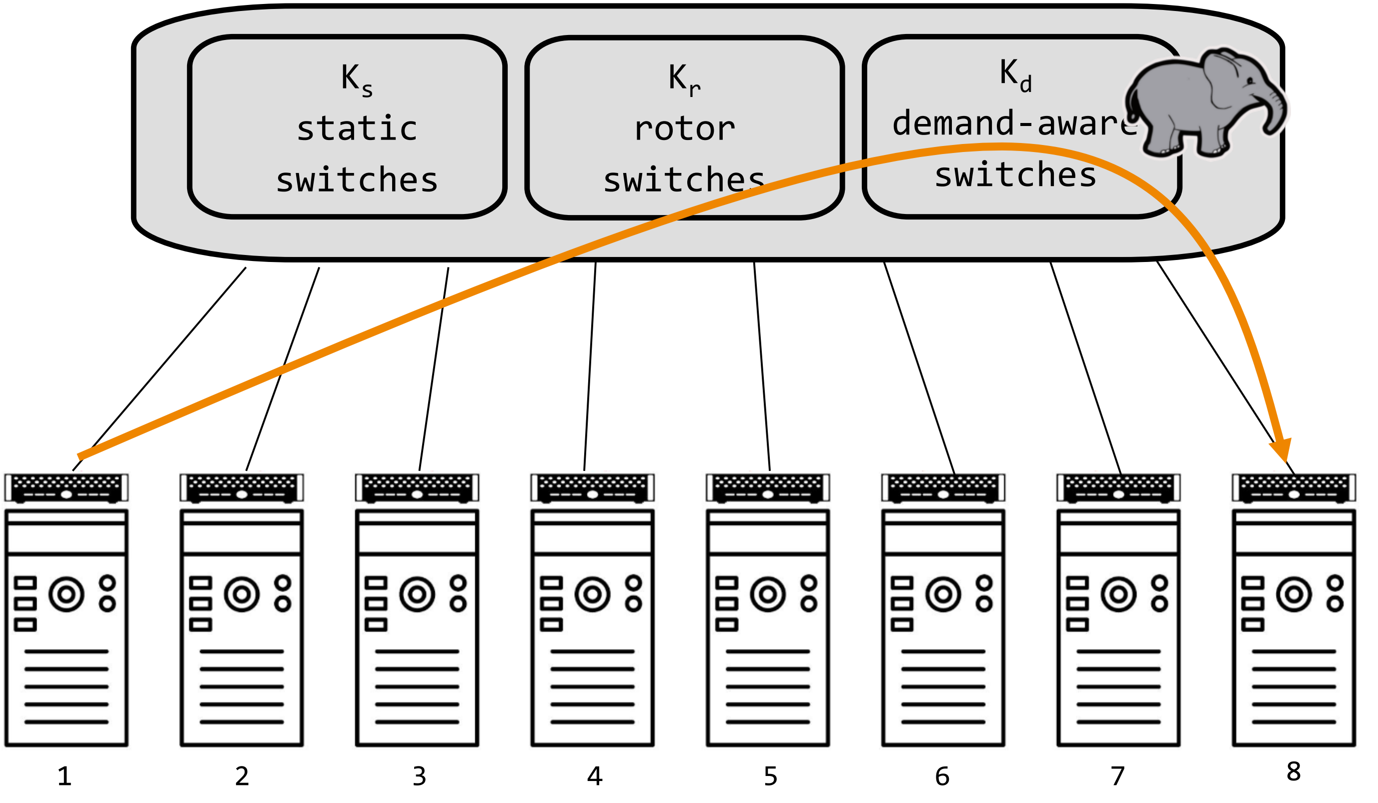
Scheduling: … and large flows via demand-aware switches (if one available, otherwise via rotor).

Cerberus



How good is it? Open problem. But there are bounds.

Cerberus

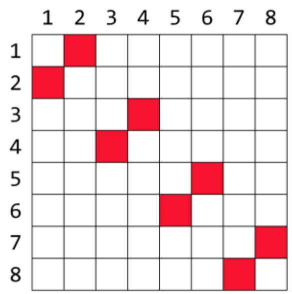


How good is it? Open problem. But there are bounds.
How to realize such an architecture?! Scalable control plane?

Throughput of RDCNs?

Demand Matrix

T

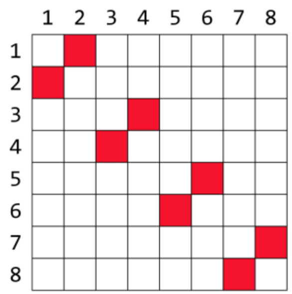


Metric: throughput
of a demand matrix…

Abdu et al., SC 2016
Namyar et al., SIGCOMM 2021

Throughput of RDCNs?

Demand Matrix

T 

$\times\ \theta(T)$

Metric: throughput of a demand matrix…

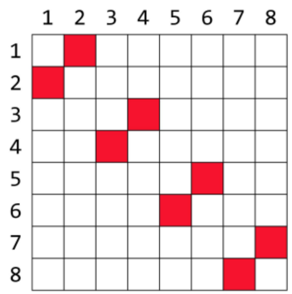
… is the maximal scale down factor by which traffic is feasible $0 \leq \theta(T) \leq 1$.

Abdu et al., SC 2016
Namyar et al., SIGCOMM 2021

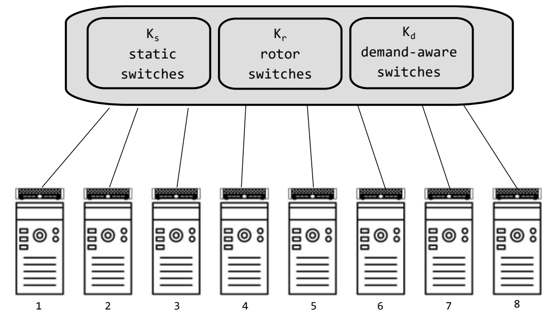
Throughput of RDCNs?

Demand Matrix

T



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



Metric: throughput of a demand matrix…

… is the maximal scale down factor by which traffic is feasible $0 \leq \theta(T) \leq 1$.

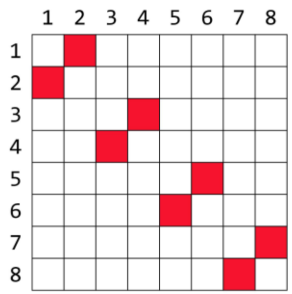
Throughput of network $\boldsymbol{\theta}^*$: worst case T

Abdu et al., SC 2016
Namyar et al., SIGCOMM 2021

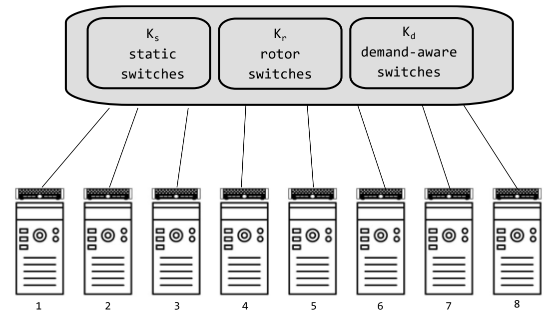
Throughput of RDCNs?

Demand Matrix

T



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



Metric: throughput of a demand matrix…

… is the maximal scale down factor by which traffic is feasible $0 \le \theta(T) \le 1$.

Throughput of network $\boldsymbol{\theta}^*$: worst case T

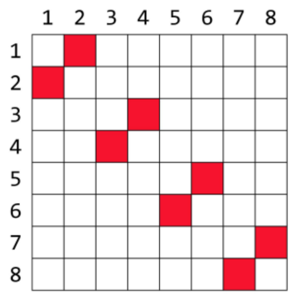
Worst T for different networks?

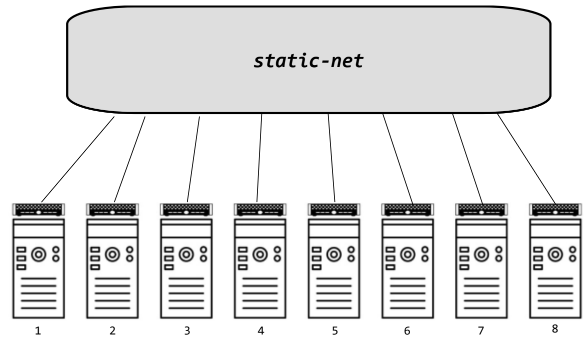
Abdu et al., SC 2016
Namyar et al., SIGCOMM 2021

Throughput: Expander

Demand Matrix

T



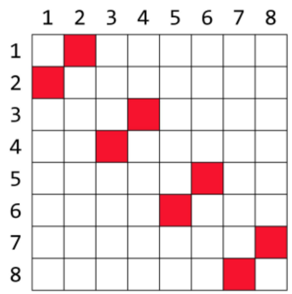


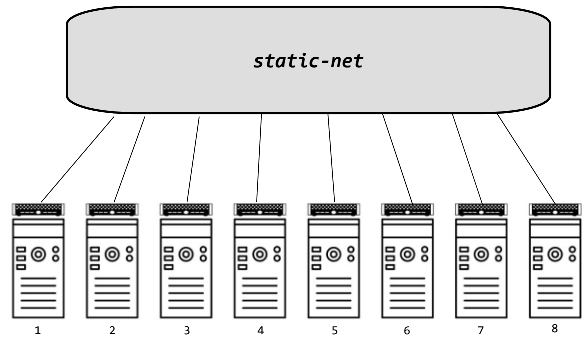
Permutation matrix
is the worst demand

Throughput: Expander

Demand Matrix

T





Permutation matrix
is the worst demand

$$\theta^* \leq \frac{1}{\mathrm{epl}(G(k))}$$

Bandwidth tax

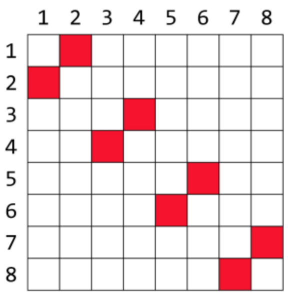
Expected path length

Namyar et al., SIGCOMM 2021

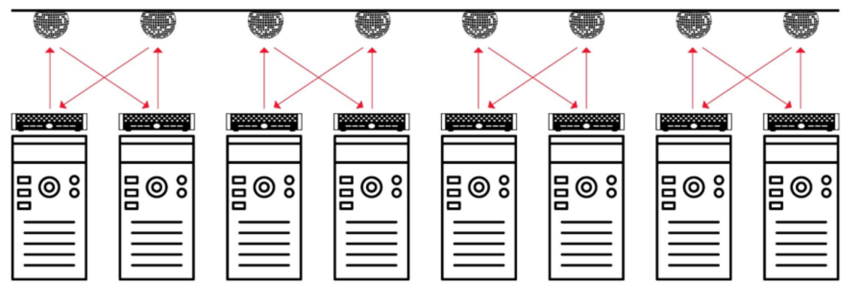
Throughput: Demand-Aware

Demand Matrix

T



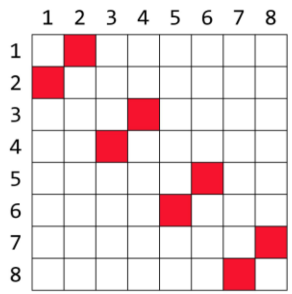
Permutation matrix
is the best demand ☺

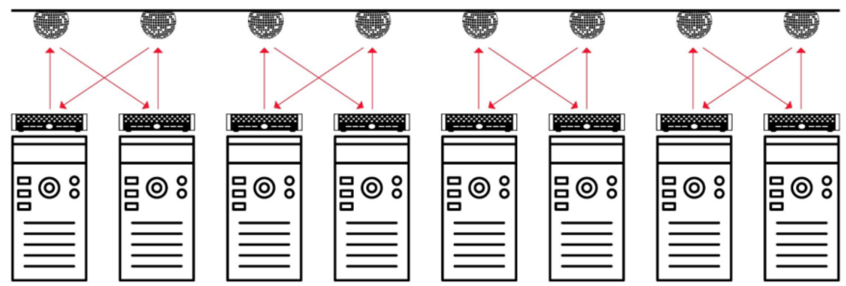


Throughput: Demand-Aware

Demand Matrix

T





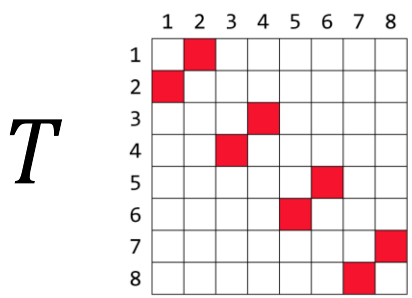
Permutation matrix
is the best demand ☺

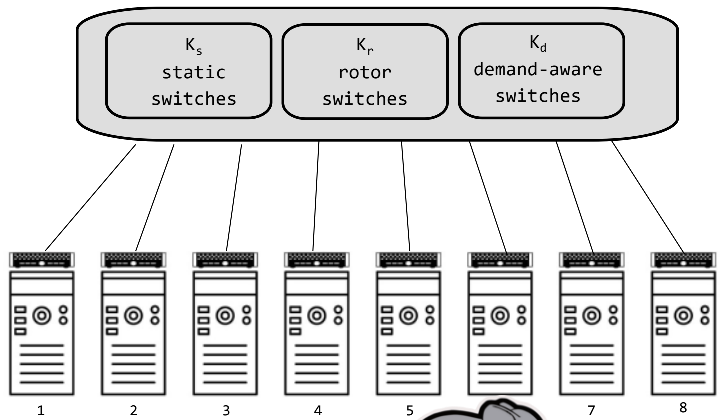
Demand-aware performs poorly for unstructured demand.

Throughput formula is a function of Latency tax. 

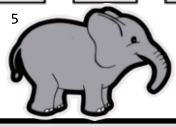
Throughput: Cerberus

Demand Matrix









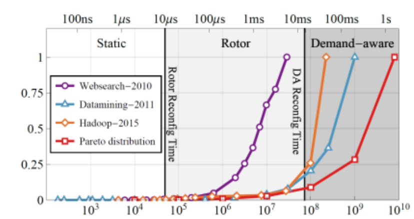
K_s
static
switches

$\overset{\theta}{=}$

K_r
rotor
switches

$\overset{\theta}{=}$

K_d
demand-aware
switches

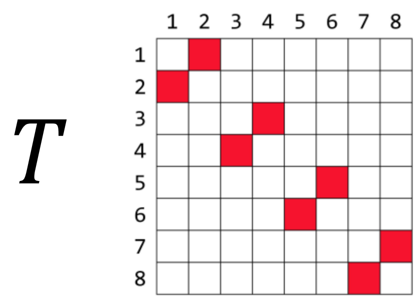


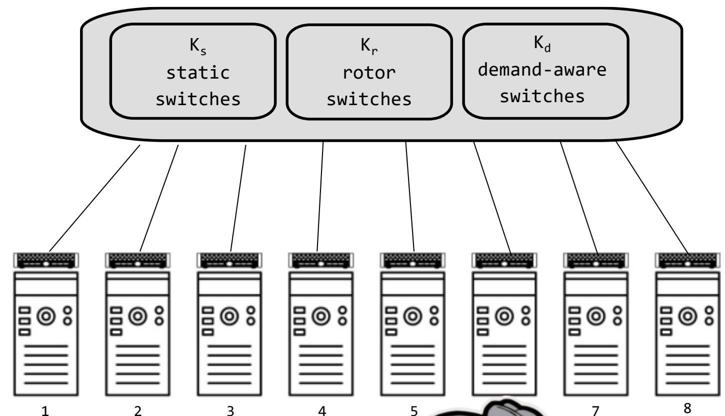
Cerberus balances optimally across switch types.

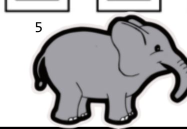
Throughput depends on both:  Bandwidth tax + Latency tax 

Throughput: Cerberus

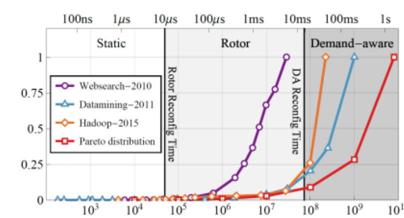
Demand Matrix







K_s static switches $\overset{\theta}{=}$ K_r rotor switches $\overset{\theta}{=}$ K_d demand-aware switches



Cerberus balances optimally across switch types.

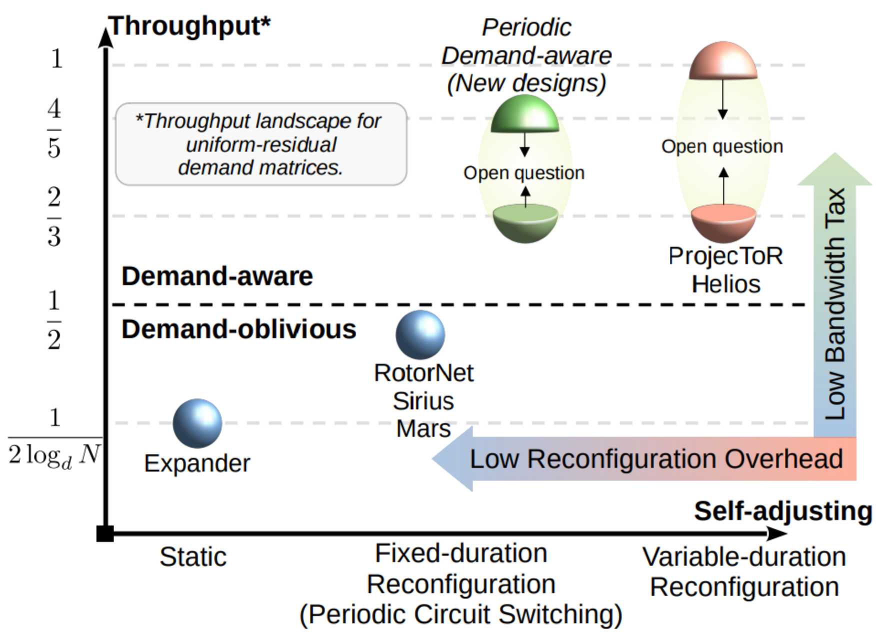
Throughput depends on both:  Bandwidth tax + Latency tax 

But theoretical problem open: no exact formula known yet.

Throughput:

Many More Open Questions

→ Throughput bounds for many designs not fully understood yet



Addanki et al., arXiv 2025: https://arxiv.org/pdf/2405.20869

Addanki et al., Vermillion: https://arxiv.org/pdf/2504.09892

How to support such dynamic networks on other layers?

More Challenges:

Network Layer?

- *ECMP* reconvergence?! Benefits of *Valiant* routing?
- How to avoid packet *reorderings*? *RDMA* network cards don’t like them!
- Routing in hybrid networks: segregated vs *non-segregated*?
- First ideas: *local* routing! Techniques from dynamic P2P systems?

Duo: A High-Throughput Reconfigurable Datacenter Network Using Local Routing and Control

JOHANNES ZERWAS, TUM School of Computation, Information and Technology, Technical University of Munich, Germany
CSABA GYÖRGYI, University of Vienna and ELTE Eötvös Loránd University, Austria and Hungary
ANDREAS BLENK, Siemens AG, Germany
STEFAN SCHMID, TU Berlin & Fraunhofer SIT, Germany
CHEN AVIN, Ben-Gurion University, Israel

The performance of many cloud-based applications critically depends on the capacity of the underlying datacenter network. A particularly innovative approach to improve the throughput in datacenters is enabled by emerging optical technologies, which allow to dynamically adjust the physical network topology, both in an oblivious or demand-aware manner. However, such topology engineering, i.e., the operation and control of dynamic datacenter networks, is considered complex and currently comes with restrictions and overheads.

We present Duo, a novel demand-aware reconfigurable rack-to-rack datacenter network design realized with a simple and efficient control plane. Duo is based on the well-known de Bruijn topology (implemented using a small number of optical circuit switches) and the key observation that this topology can be enhanced using dynamic (“opportunistic”) links between its nodes.

In contrast to previous systems, Duo has several desired features: i) It makes effective use of the network

More Challenges:

Congestion Control?

⇢ First ideas for quickly reacting TCP: ***ReTCP***, ***PowerTCP***, …

⇢ Or better completely different approach? Even centralized?!

Adapting TCP for Reconfigurable Datacenter Networks

Matthew K. Mukerjee*†, Christopher Canel*, Weiyang Wang°, Daehyeok Kim*‡, Srinivasan Seshan*, Alex C. Snoeren°

Carnegie Mellon University, °UC San Diego, †Nefeli Networks, ‡Microsoft Research

Abstract

Reconfigurable datacenter networks (RDCNs) augment traditional packet switches with high-bandwidth reconfigurable circuits. In these networks, high-bandwidth circuits are assigned to particular source-destination rack pairs based on a schedule. To make efficient use of RDCNs, active TCP flows between such pairs must quickly ramp up their sending rates when high-bandwidth circuits are made available. Past studies have shown that TCP performs well on RDCNs with millisecond-scale reconfiguration delays, during which time the circuit network is offline. However, modern RDCNs can reconfigure in as little as 20 μs, and maintain a particular con-…

switches incur non-trivial reconfiguration delays while they adjust the high-bandwidth topology, and portions of the circuit network may be unavailable during these periods. Hence, such hybrid designs often result in fluctuations between periods of high bandwidth—when a circuit is provisioned—and low bandwidth—when the packet network is in use. While periods of higher bandwidth are attractive in principle, recent proposals suggest adjusting the topology frequently. The resulting bandwidth fluctuations pose a problem for end-host applications: their active TCP connections must rapidly increase transmission rates to use the available bandwidth and then slow down again to avoid massive queuing. In this paper, we …

POWERTCP: Pushing the Performance Limits of Datacenter Networks

Vamsi Addanki, *University of Vienna, TU Berlin*

Oliver Michel, *University of Vienna, Princeton University*

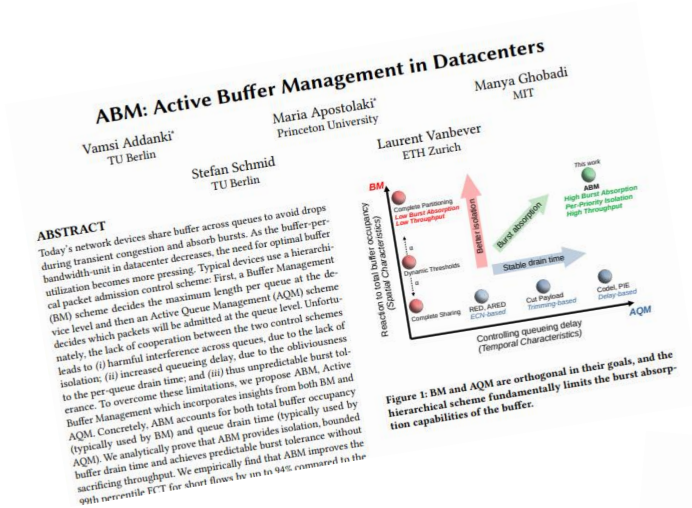
Stefan Schmid, *University of Vienna, TU Berlin*

Abstract

Increasingly stringent throughput and latency requirements in datacenter networks demand fast and accurate congestion control. We observe that the reaction time and accuracy of existing datacenter congestion control schemes are inherently …

stringent performance requirements are introduced by today's trend of resource disaggregation in datacenters where fast access to remote resources (e.g., GPUs or memory) is pivotal for the overall system performance [36]. Building systems with strict performance requirements is especially challenging under bursty traffic patterns as they are commonly observed …

More Challenges:
Buffering?



ABM: Active Buffer Management in Datacenters

Vamsi Addanki*
TU Berlin

Maria Apostolaki*
Princeton University

Manya Ghobadi
MIT

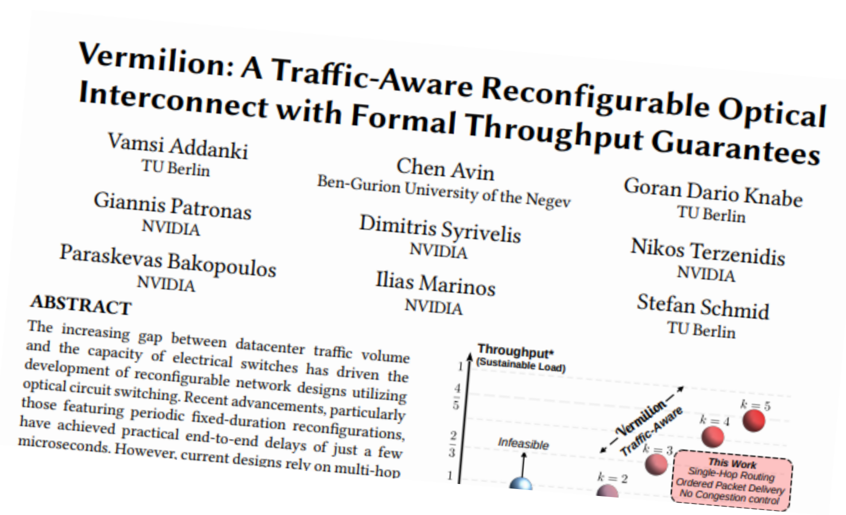
Stefan Schmid
TU Berlin

Laurent Vanbever
ETH Zurich

ABSTRACT

Today's network devices share buffer across queues to avoid drops during transient congestion and absorb bursts. As the buffer-per-bandwidth-unit in datacenter decreases, the need for optimal buffer utilization becomes more pressing. Typical devices use a hierarchical packet admission control scheme: First, a Buffer Management (BM) scheme decides the maximum length per queue at the device level and then an Active Queue Management (AQM) scheme decides which packets will be admitted at the queue level. Unfortunately, the lack of cooperation between the two control schemes leads to (i) harmful interference across queues, due to the lack of isolation; (ii) increased queueing delay, due to the obliviousness to the per-queue drain time; and (iii) thus unpredictable burst tolerance. To overcome these limitations, we propose ABM, Active Buffer Management which incorporates insights from both BM and AQM. Concretely, ABM accounts for both total buffer occupancy (typically used by BM) and queue drain time (typically used by AQM). We analytically prove that ABM provides isolation, bounded buffer drain time and achieves predictable burst tolerance without sacrificing throughput. We empirically find that ABM improves the 99th percentile FCT for short flows by up to 94% compared to the

Figure 1: BM and AQM are orthogonal in their goals, and the hierarchical scheme fundamentally limits the burst absorption capabilities of the buffer.



Vermilion: A Traffic-Aware Reconfigurable Optical Interconnect with Formal Throughput Guarantees

Vamsi Addanki
TU Berlin

Chen Avin
Ben-Gurion University of the Negev

Goran Dario Knabe
TU Berlin

Giannis Patronas
NVIDIA

Dimitris Syrivelis
NVIDIA

Nikos Terzenidis
NVIDIA

Paraskevas Bakopoulos
NVIDIA

Ilias Marinos
NVIDIA

Stefan Schmid
TU Berlin

ABSTRACT

The increasing gap between datacenter traffic volume and the capacity of electrical switches has driven the development of reconfigurable network designs utilizing optical circuit switching. Recent advancements, particularly those featuring periodic fixed-duration reconfigurations, have achieved practical end-to-end delays of just a few microseconds. However, current designs rely on multi-hop

Case Study:

PowerTCP

Existing congestion control algorithms based on either

- State ("voltage") like BDP, queue length, loss, e.g.:
  - DCTCP: uses ECN/loss
  - Swift: RTT
  - HPCC: inflight packets
- Gradient ("current") like reaction to queue length change
  - Timely: RTT-gradient based

Case Study:

PowerTCP

Existing congestion control algorithms based on either

- State (“voltage”) like BDP, queue length, loss, e.g.:
  - DCTCP: uses ECN/loss
  - Swift: RTT
  - HPCC: inflight packets

☺ Can achieve near-zero queue equilibrium

☹ Slow reaction

- Gradient (“current”) like reaction to queue length change
  - Timely: RTT-gradient based

Case Study:

PowerTCP

Existing congestion control algorithms based on either

- State ("voltage") like BDP, queue length, loss, e.g.:
  - DCTCP: uses ECN/loss
  - Swift: RTT
  - HPCC: inflight packets
- Gradient ("current") like reaction to queue length change
  - Timely: RTT-gradient based

☺ Fast reaction

☹ No equilibrium

Case Study:

PowerTCP

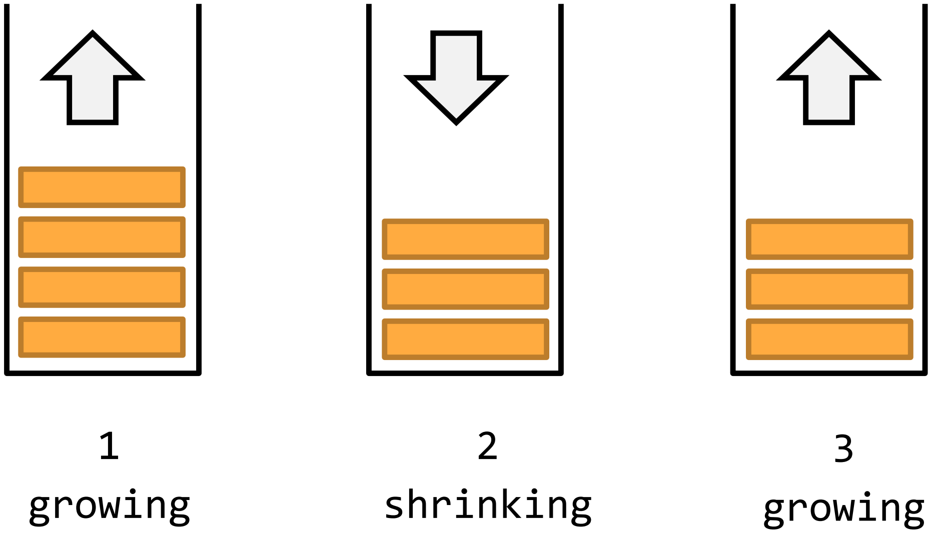
Existing congestion control algorithms based on either

- State ("voltage") like BDP, queue length, loss, e.g.:
  - DCTCP: uses ECN/loss
  - Swift: RTT
  - HPCC: inflight packets
- Gradient ("current") like reaction to queue length change
  - Timely: RTT-gradient based

Limitation: using only one of the two may miss useful information for fine-grained adaptions!

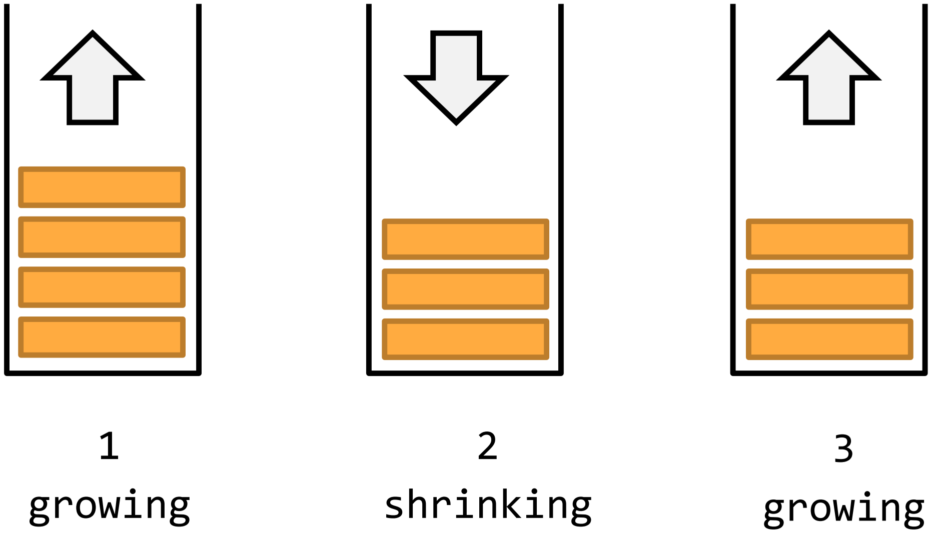
Limitation of SOTA

⇢ Consider a queue which may be in three different states:



Limitation of SOTA

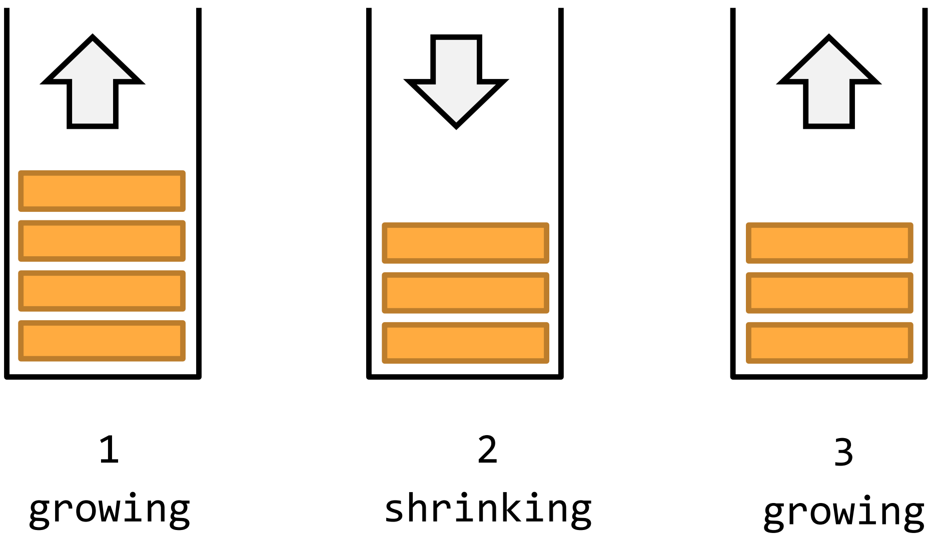
⇢ Consider a queue which may be in three different states:



2 and 3: impossible to
distinguish for voltage-based CCA

Limitation of SOTA

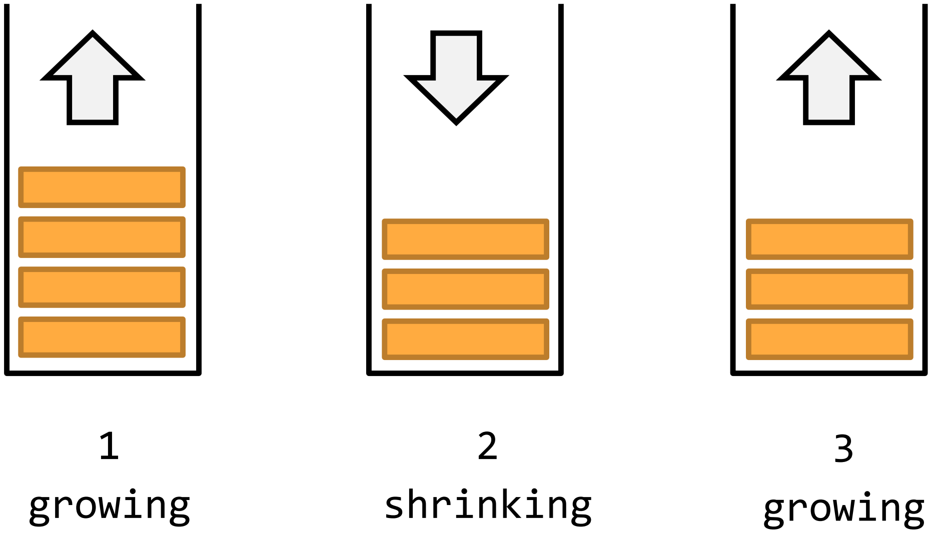
⇢ Consider a queue which may be in three different states:



1 and 3: impossible to
distinguish for current-based CC

Limitation of SOTA

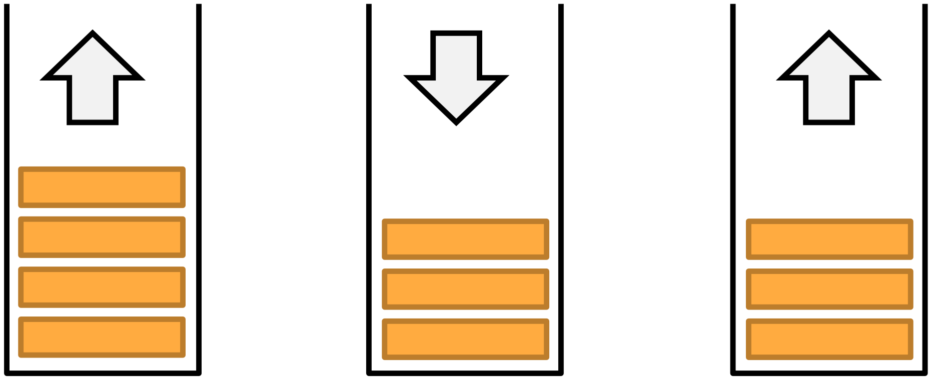
⇢ Consider a queue which may be in three different states:



We need both: Power (Voltage x Current)

Limitation of SOTA

⇢ Consider a queue which may be in three different states:



1	2	3
growing	shrinking	growing

We need both: Power (Voltage x Current)

Inspired: POWERTCP

More benefits of optical & reconfigurable switching

So far: focus on throughput performance.

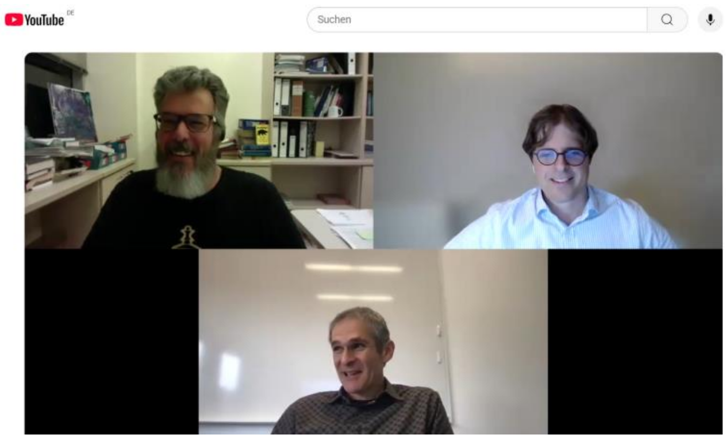
Benefit 1:

Evolving Datacenters

⇢ Reconfigurable datacenter networks naturally support *heterogeneous* network elements

⇢ And therefore also *incremental* hardware upgrades

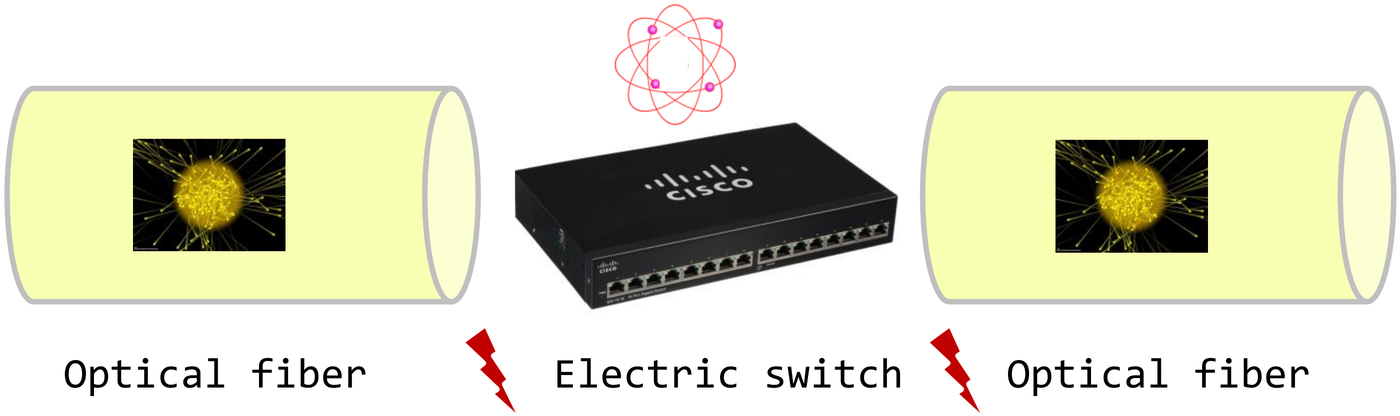
See interview with Amin Vahdat, Google in CACM: https://www.youtube.com/watch?v=IxcV1gu8ETA



Benefit 2:

Energy and Latency

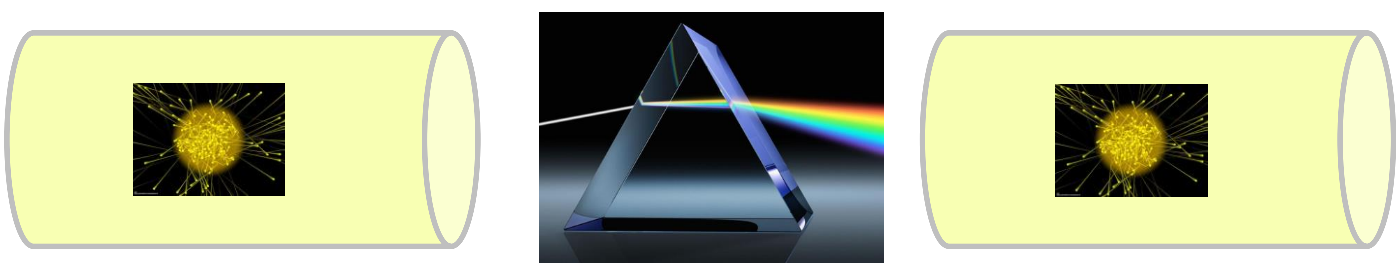
- No need to *convert* photons in fiber to electrons in switch (and back)
- Can safe *energy* and reduce *latency* (in addition to enabling almost unlimited throughput)



Benefit 2:

Energy and Latency

- No need to *convert* photons in fiber to electrons in switch (and back)
- Can safe *energy* and reduce *latency* (in addition to enabling almost unlimited throughput)

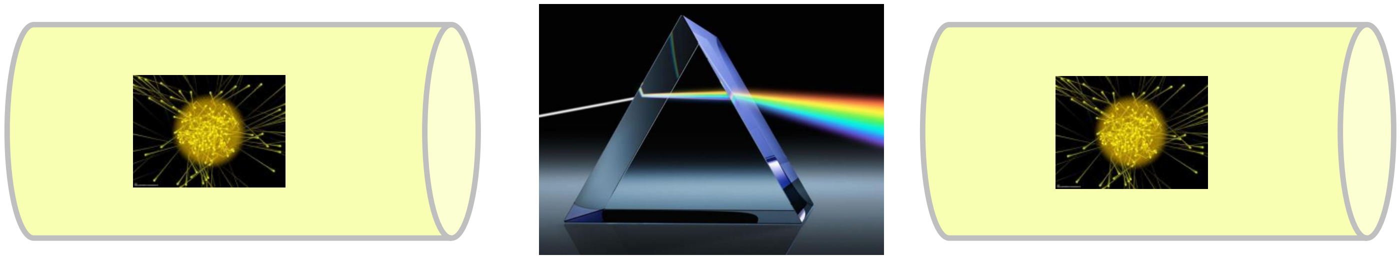


Optical fiber —— Optical switch —— Optical fiber

Benefit 2:

Energy and Latency

- No need to *convert* photons in fiber to electrons in switch (and back)
- Can safe *energy* and reduce *latency* (in addition to enabling almost unlimited throughput)



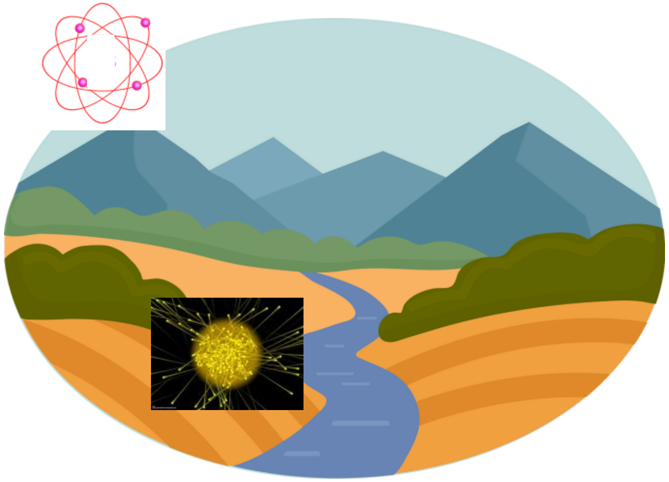
Optical fiber — Optical switch — Optical fiber

- Interesting for emerging *distributed datacenters*!

Benefit 3:

Resilience

Floodings in South Germany destroyed much electrical network infrastructure

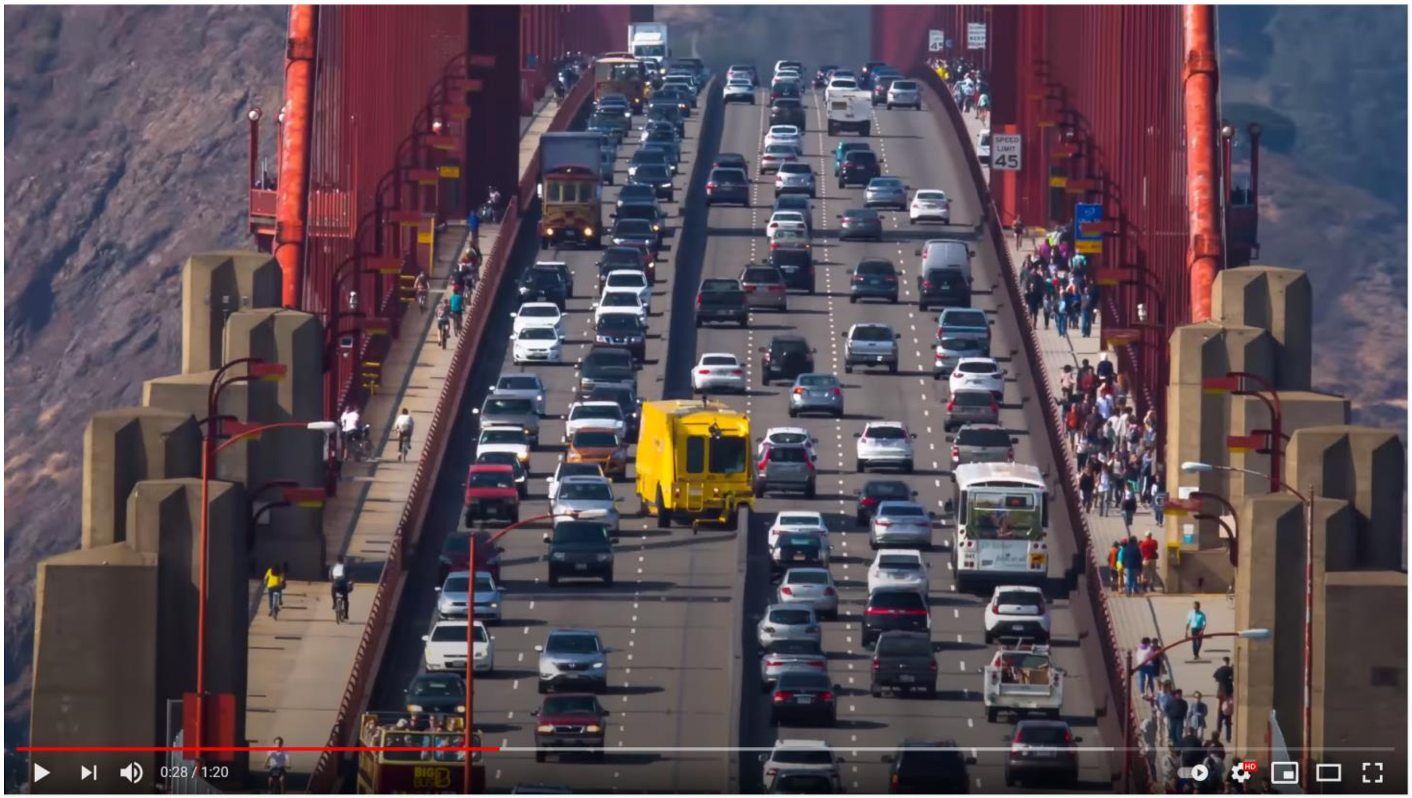


Solution: deploy optical infrastructure (in valleys) and electrical *on hills* where safe?

Conclusion

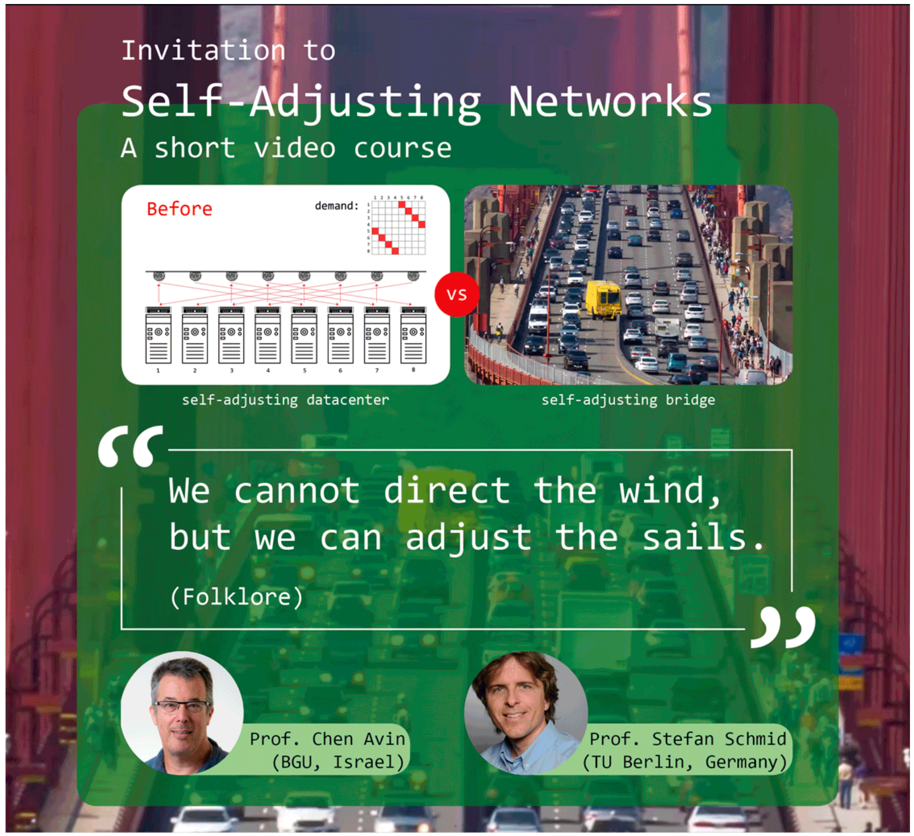
- Opportunity: *structure* in demand and *reconfigurable* networks
- So far: tip of the iceberg
- Many challenges
  - Optimal design depends on traffic pattern
  - How to *measure/predict* traffic?
  - Impact on other *layers*?
  - *Scalable control* plane
  - *Application-specific* self-adjusting networks?
- Many more *opportunities* for optical networks

Thank you! Questions?



Slides available here:

Online Video Course



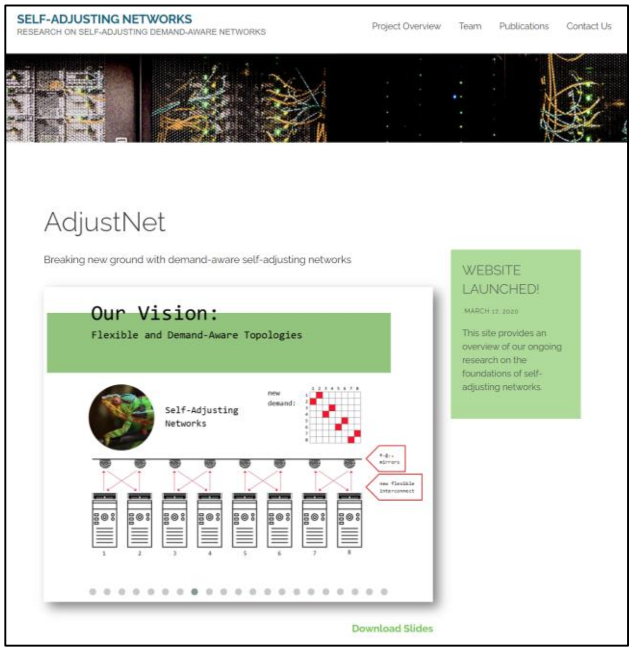
https://self-adjusting.net/course

YouTube Interview & CACM

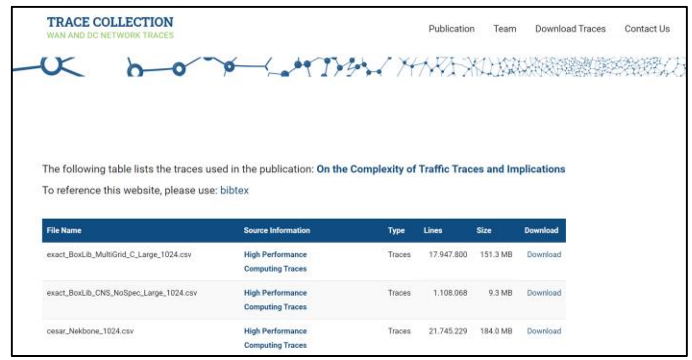
Check out our **YouTube interviews**
on Reconfigurable Datacenter Networks:

[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

Websites



http://self-adjusting.net/
Project website



https://trace-collection.net/
Trace collection website

June’25 CACM Article

Revolutionizing Datacenter Networks via Reconfigurable Topologies

CHEN AVIN, is a Professor at Ben-Gurion University of the Negev, Beersheva, Israel
STEFAN SCHMID, is a Professor at TU Berlin, Berlin, Germany

With the popularity of cloud computing and data-intensive applications such as machine learning, datacenter networks have become a critical infrastructure for our digital society. Given the explosive growth of datacenter traffic and the slowdown of Moore’s law, significant efforts have been made to improve datacenter network performance over the last decade. A particularly innovative solution is reconfigurable datacenter networks (RDCNs): datacenter networks whose topologies dynamically change over time, in either a demand-oblivious or a demand-aware manner. Such dynamic topologies are enabled by recent optical switching technologies and stand in stark contrast to state-of-the-art datacenter network topologies, which are fixed and oblivious to the actual traffic demand. In particular, reconfigurable demand-aware and “self-adjusting” datacenter networks are motivated empirically by the significant spatial and temporal structures observed in datacenter communication traffic. This paper presents an overview of reconfigurable datacenter networks. In particular, we discuss the motivation for such reconfigurable architectures, review the technological enablers, and present a taxonomy that classifies the design space into two dimensions: static vs. dynamic and demand-oblivious vs. demand-aware. We further present a formal model and discuss related research challenges. Our article comes with complementary video interviews in which three leading experts, Manya Ghobadi, Amin Vahdat, and George Papen, share with us their perspectives on reconfigurable datacenter networks.

KEY INSIGHTS

- **Datacenter networks have become a critical infrastructure** for our digital society, serving explosively growing communication traffic.
- **Reconfigurable datacenter networks (RDCNs)** which can adapt their topology dynamically, based on innovative **optical switching technologies**, bear the potential to improve datacenter network performance, and to simplify datacenter planning and operations.
- Demand-aware dynamic topologies are particularly interesting because of the **significant spatial and temporal structures** observed in real-world traffic, e.g., related to distributed machine learning.
- The study of RDCNs and self-adjusting networks raises many **novel technological and research challenges** related to their design, control, and performance.

References (1)

Revolutionizing Datacenter Networks via Reconfigurable Topologies
Chen Avin and Stefan Schmid.
Communications of the ACM (**CACM**), 2025.

Cerberus: The Power of Choices in Datacenter Topology Design (A Throughput Perspective)
Chen Griner, Johannes Zerwas, Andreas Blenk, Manya Ghobadi, Stefan Schmid, and Chen Avin.
ACM **SIGMETRICS** and ACM Performance Evaluation Review (**PER**), Mumbai, India, June 2022.

Mars: Near-Optimal Throughput with Shallow Buffers in Reconfigurable Datacenter Networks
Vamsi Addanki, Chen Avin, and Stefan Schmid.
ACM **SIGMETRICS** and ACM Performance Evaluation Review (**PER**), Orlando, Florida, USA, June 2023.

Duo: A High-Throughput Reconfigurable Datacenter Network Using Local Routing and Control
Johannes Zerwas, Csaba Györgyi, Andreas Blenk, Stefan Schmid, and Chen Avin.
ACM **SIGMETRICS** and ACM Performance Evaluation Review (**PER**), Orlando, Florida, USA, June 2023.

On the Complexity of Traffic Traces and Implications
Chen Avin, Manya Ghobadi, Chen Griner, and Stefan Schmid.
ACM **SIGMETRICS** and ACM Performance Evaluation Review (**PER**), Boston, Massachusetts, USA, June 2020.

Toward Demand-Aware Networking: A Theory for Self-Adjusting Networks (Editorial)
Chen Avin and Stefan Schmid.
ACM SIGCOMM Computer Communication Review (**CCR**), October 2018.

Credence: Augmenting Datacenter Switch Buffer Sharing with ML Predictions
Vamsi Addanki, Maciej Pacut, and Stefan Schmid.
21st USENIX Symposium on Networked Systems Design and Implementation (**NSDI**), Santa Clara, California, USA, April 2024.

PowerTCP: Pushing the Performance Limits of Datacenter Networks
Vamsi Addanki, Oliver Michel, and Stefan Schmid.
19th USENIX Symposium on Networked Systems Design and Implementation (**NSDI**), Renton, Washington, USA, April 2022.

TCP's Third Eye: Leveraging eBPF for Telemetry-Powered Congestion Control
Jörn-Thorben Hinz, Vamsi Addanki, Csaba Györgyi, Theo Jepsen, and Stefan Schmid.
SIGCOMM Workshop on eBPF and Kernel Extensions (**eBPF**), Columbia University, New York City, New York, USA, September 2023.

References (2)

ABM: Active Buffer Management in Datacenters
Vamsi Addanki, Maria Apostolaki, Manya Ghobadi, Stefan Schmid, and Laurent Vanbever.
ACM **SIGCOMM**, Amsterdam, Netherlands, August 2022.

ExRec: Experimental Framework for Reconfigurable Networks Based on Off-the-Shelf Hardware
Johannes Zerwas, Chen Avin, Stefan Schmid, and Andreas Blenk.
16th ACM/IEEE Symposium on Architectures for Networking and Communications Systems (**ANCS**), Virtual Conference, December 2021.

Demand-Aware Network Design with Minimal Congestion and Route Lengths
Chen Avin, Kaushik Mondal, and Stefan Schmid.
IEEE/ACM Transactions on Networking (**TON**), 2022.

A Survey of Reconfigurable Optical Networks
Matthew Nance Hall, Klaus-Tycho Foerster, Stefan Schmid, and Ramakrishnan Durairajan.
Optical Switching and Networking (**OSN**), Elsevier, 2021.

SplayNet: Towards Locally Self-Adjusting Networks
Stefan Schmid, Chen Avin, Christian Scheideler, Michael Borokhovich, Bernhard Haeupler, and Zvi Lotker.
IEEE/ACM Transactions on Networking (**TON**), Volume 24, Issue 3, 2016.

.

Bonus Material

Hogwarts Stair

Question:

How to Quantify such “Structure” in the Demand?

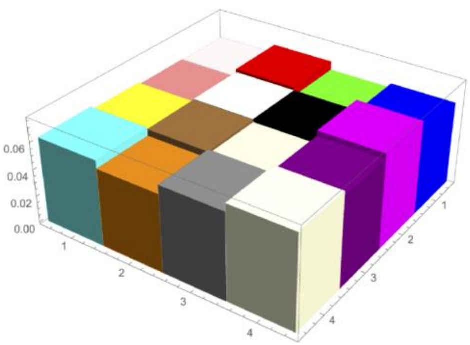
Intuition

Which demand has more structure?

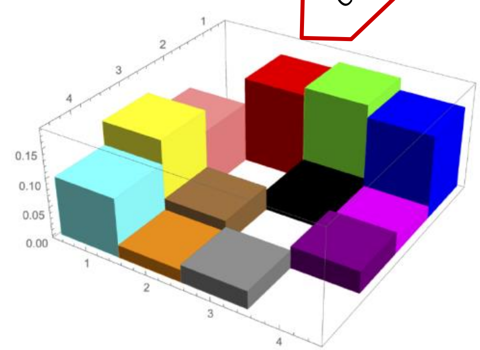
⇢ Traffic matrices of two different distributed ML applications

→ GPU-to-GPU

Color = communication pair



VS



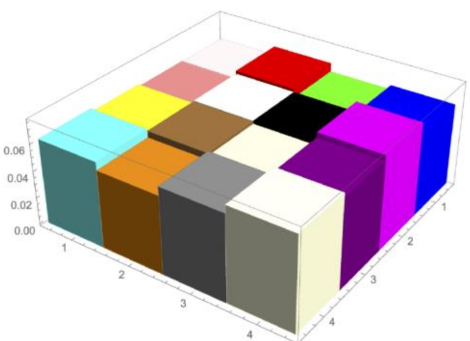
Intuition

Which demand has more structure?

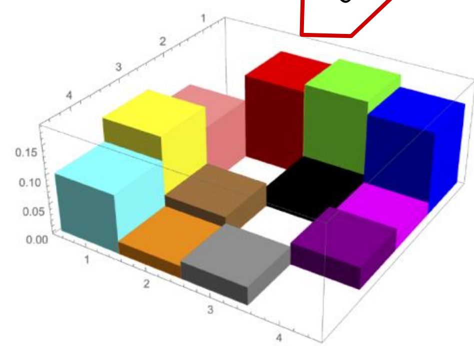
⇢ Traffic matrices of two different distributed ML applications

⇀ GPU-to-GPU

Color = communication pair



VS



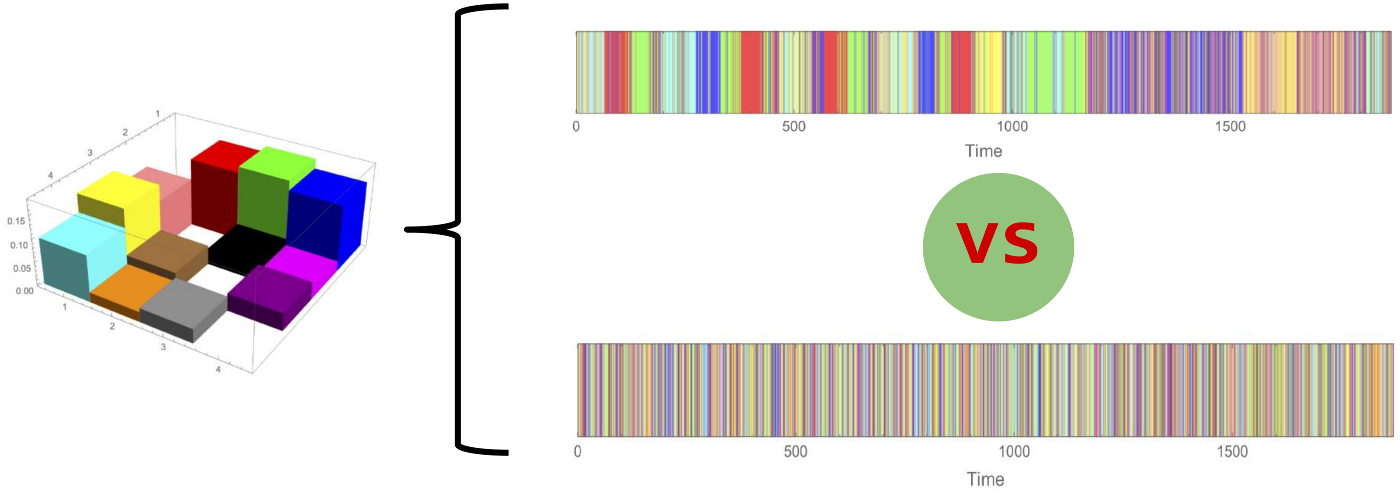
More uniform

More structure

Intuition

Spatial vs temporal structure

- Two different ways to generate same traffic matrix:
  - Same non-temporal structure
- Which one has more structure?

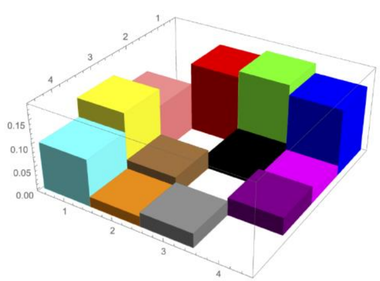


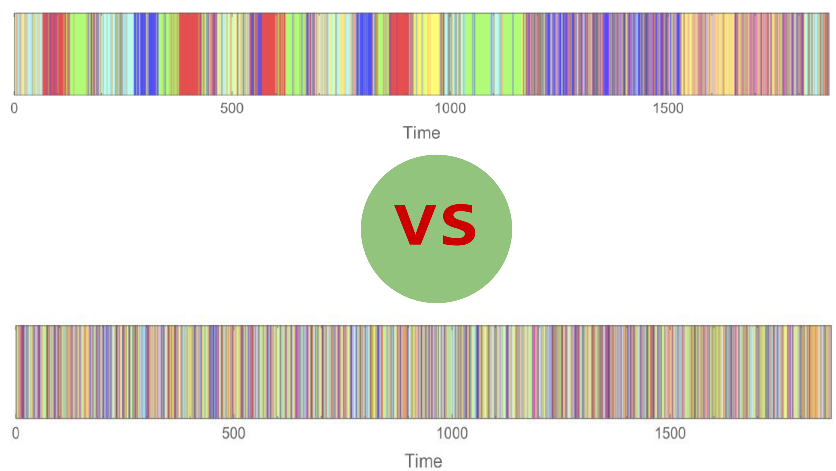
Intuition

Spatial vs temporal structure

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

⇢ Which one has more structure?



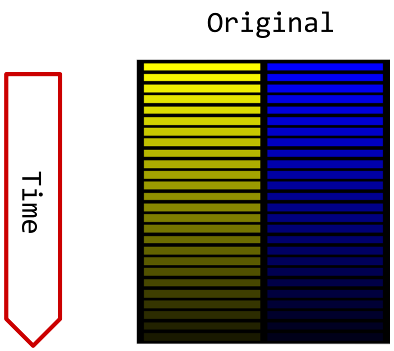


Systematically?

Trace Complexity

Information-Theoretic Approach

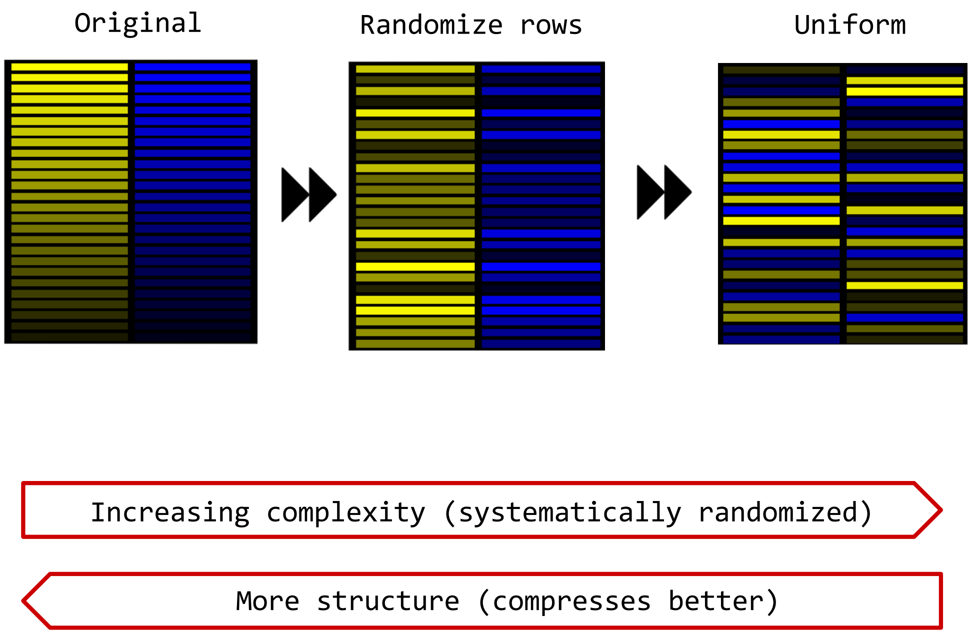
“Shuffle&Compress”



Trace Complexity

Information-Theoretic Approach

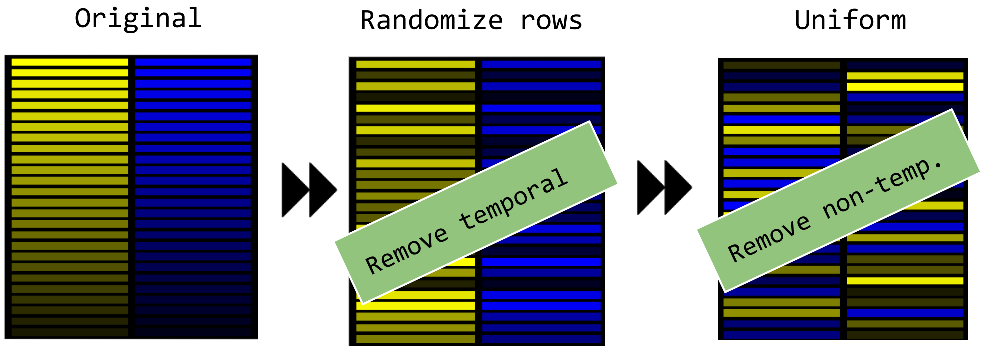
“Shuffle&Compress”



Trace Complexity

Information-Theoretic Approach

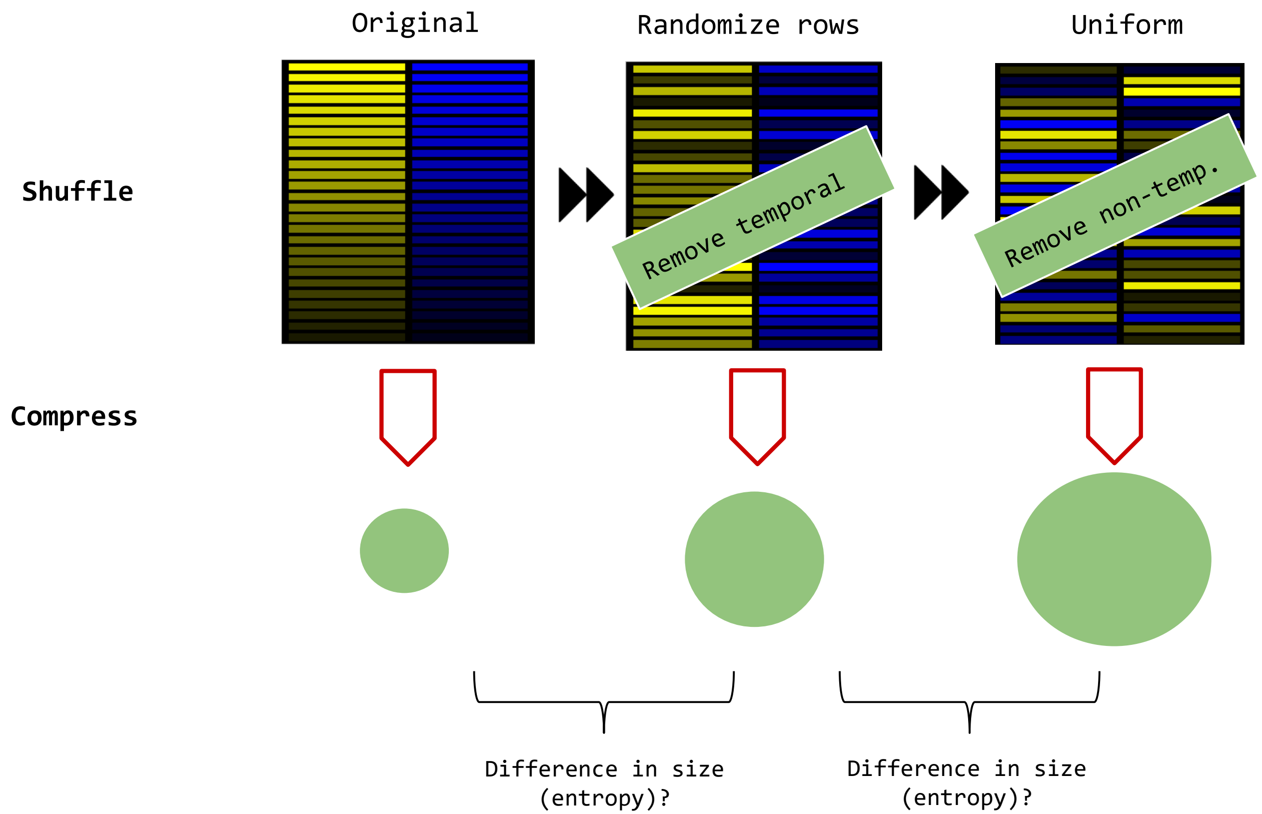
“Shuffle&Compress”



Trace Complexity

Information-Theoretic Approach

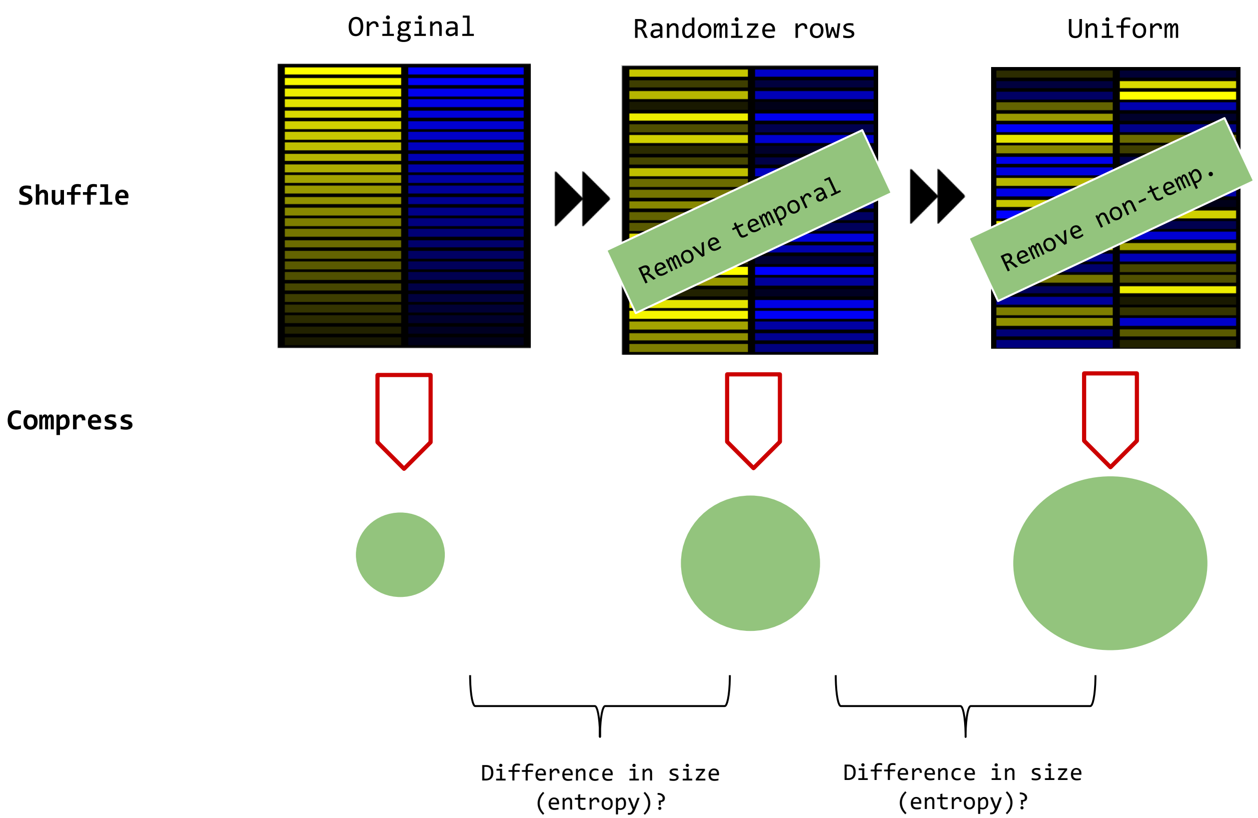
“Shuffle&Compress”



Trace Complexity

Information-Theoretic Approach

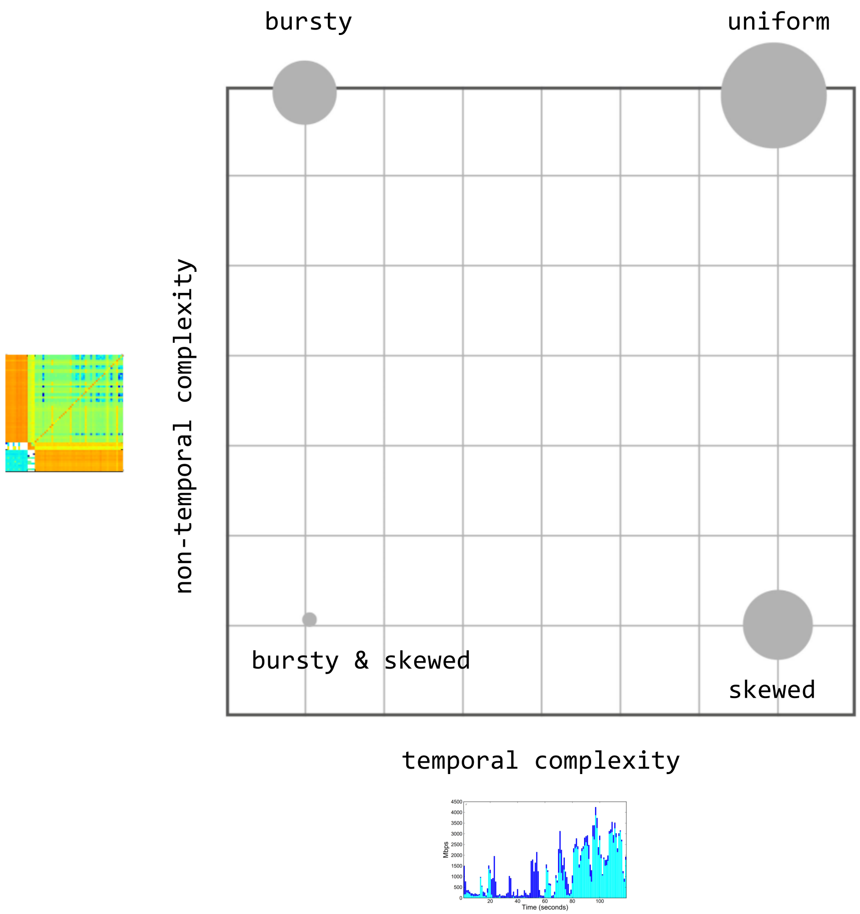
“Shuffle&Compress”



Can be used to define 2-dimensional complexity map!

Our Methodology

Complexity Map

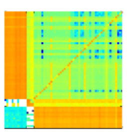


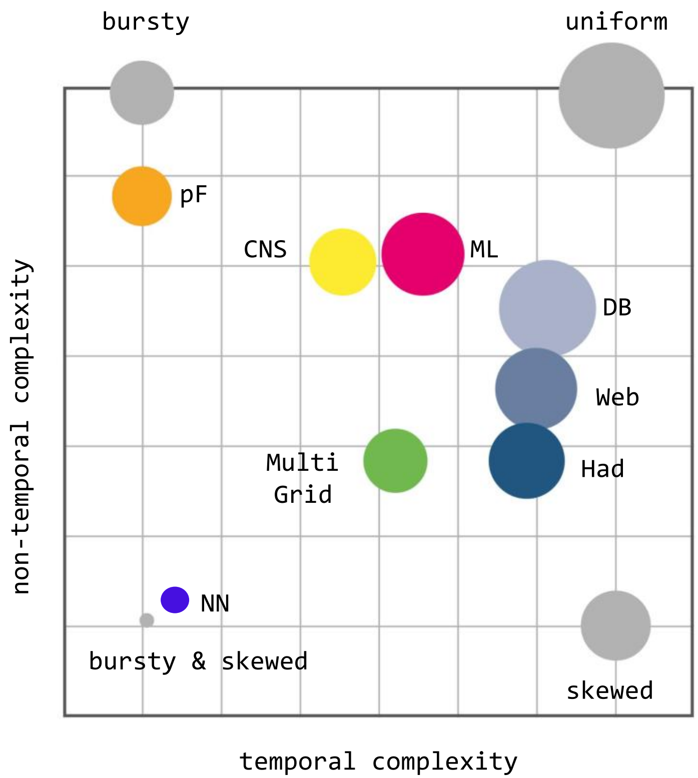
No structure

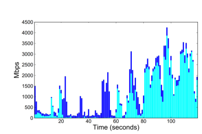
Our **approach:** iterative **randomization and compression** of trace to identify dimensions of structure.

Our Methodology

Complexity Map







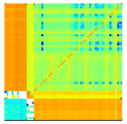
No structure

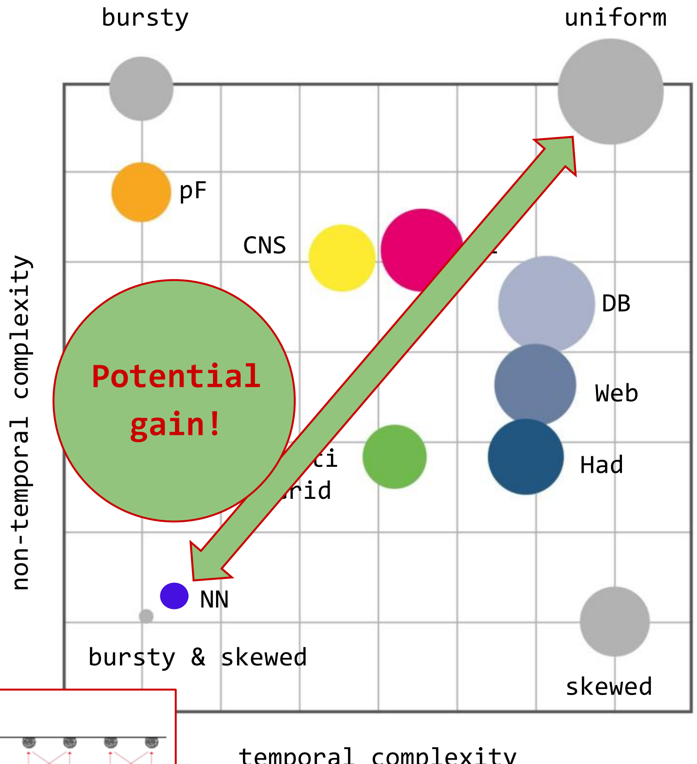
Our **approach:** iterative **randomization and compression** of trace to identify dimensions of structure.

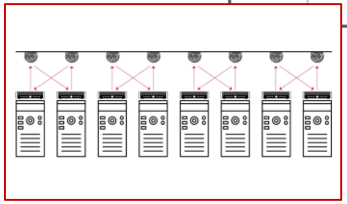
Different structures!

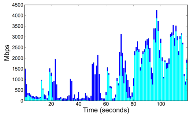
Our Methodology

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

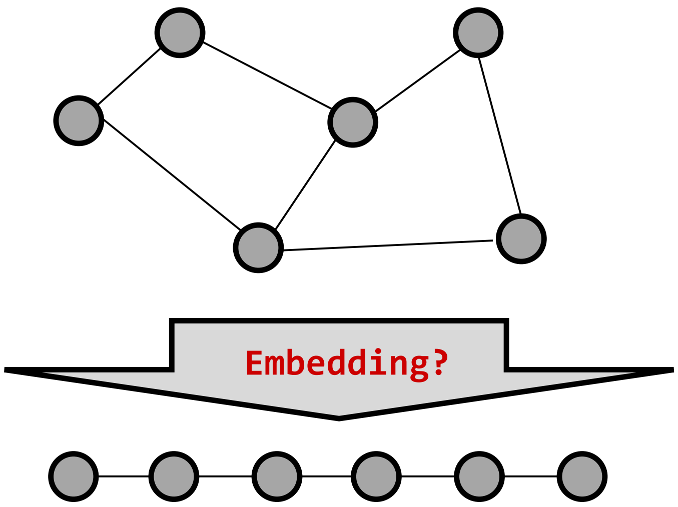
Packet traces collected from networking applications, such as datacenter traffic, have been shown to feature much *structure*: datacenter traffic matrices are sparse and skewed [16, 39], exhibit

Related Problem: Remember Bernardetta's Talk

Virtual Network Embedding Problem (VNEP)

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

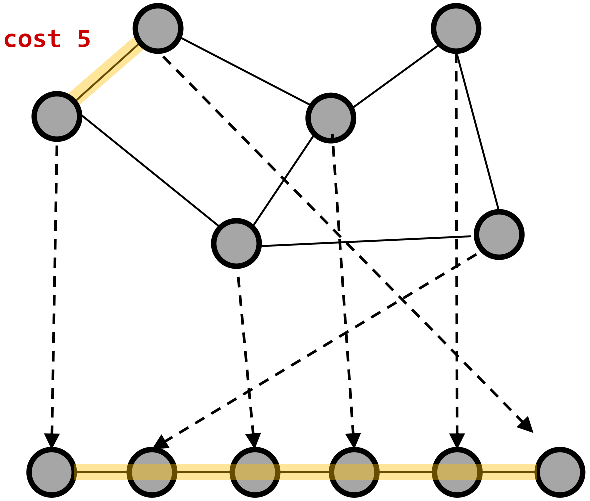
→ Minimizes sum of virtual edges



Virtual Network Embedding Problem (VNEP)

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

→ Minimizes sum of virtual edges



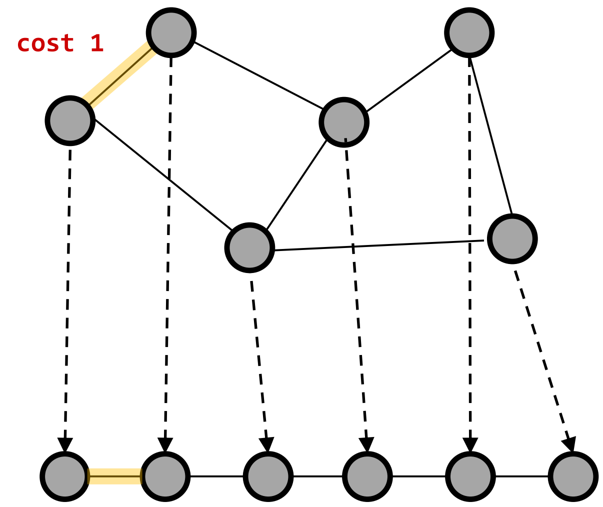
Bad!

Related Problem: Remember Bernardetta’s Talk

Virtual Network Embedding Problem (VNEP)

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

→ Minimizes sum of virtual edges



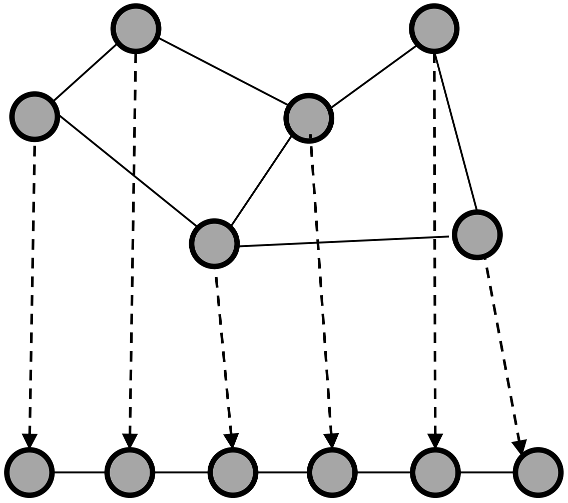
Good!

Related Problem: Remember Bernardetta's Talk

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: Remember Bernardetta's Talk

Virtual Network Embedding Problem (VNEP)

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

⇀ Minimizes sum of virtual edges

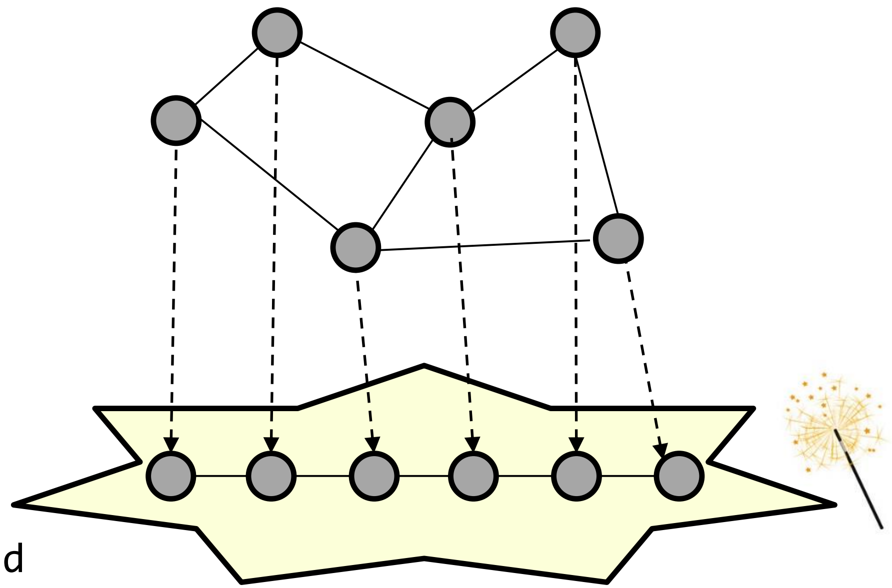
MLA is **NP-hard**

⇀ … and so is our problem!

But what about **△>2**?

⇀ Embedding problem still hard

⇀ But we have a new **degree of freedom**!



Related Problem: Remember Bernardetta's Talk

Virtual Network Embedding Problem (VNEP)

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

→ Minimizes sum of virtual edges

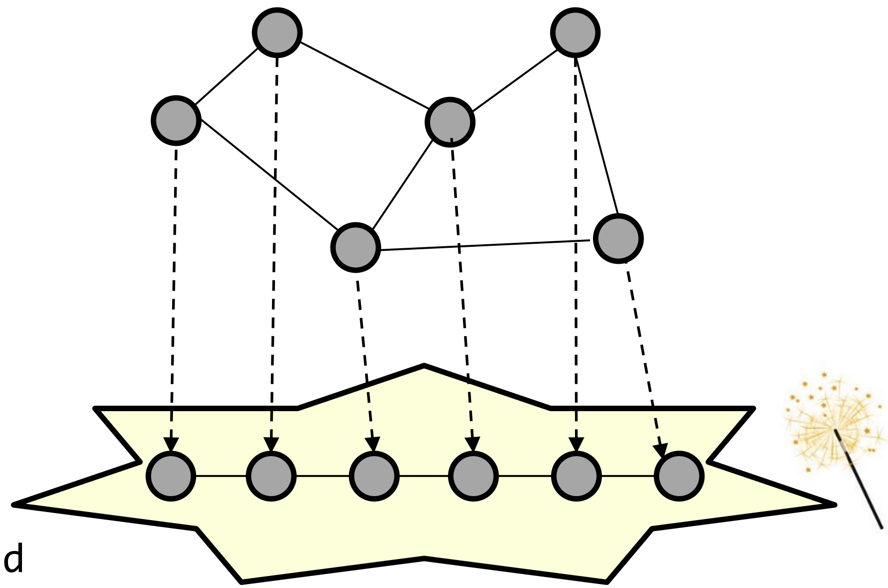
MLA is **NP-hard**

→ … and so is our problem!

But what about **△>2**?

→ Embedding problem still hard

→ But we have a new **degree of freedom**!



Simplifies problem?!

Another Related Problem

Low Distortion Spanners

- Classic problem: find *sparse*, *distance-preserving* (low-distortion) spanner of a graph
- But:
  - Spanners aim at low distortion *among all pairs*; in our case, we are only interested in the local distortion, 1-hop communication neighbors
  - We allow *auxiliary edges* (not a subgraph): similar to geometric spanners
  - We require *constant degree*

An Algorithm

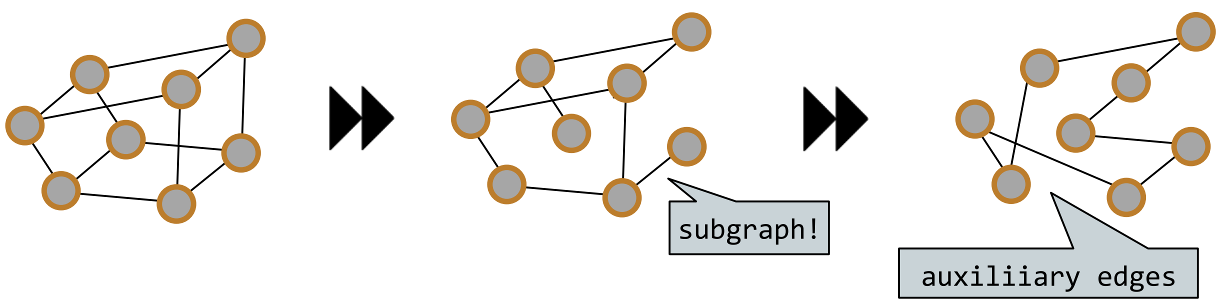
⇢ Yet, can leverage the connection to spanners sometimes!

Theorem: If demand matrix is regular and uniform, and if we can find a constant distortion, linear sized (i.e., constant, sparse) spanner for this request graph: then we can design a constant degree DAN providing an optimal expected route length *(i.e., $O(H(X|Y)+H(Y|X))$.*

r-regular *and* ***uniform*** **demand***:*

Sparse, ***irregular (constant)*** **spanner***:*

Constant *degree* ***optimal*** **DAN** (ERL at most ***log r***):



An Algorithm

⇢ Yet, can leverage the connection to spanners sometimes!

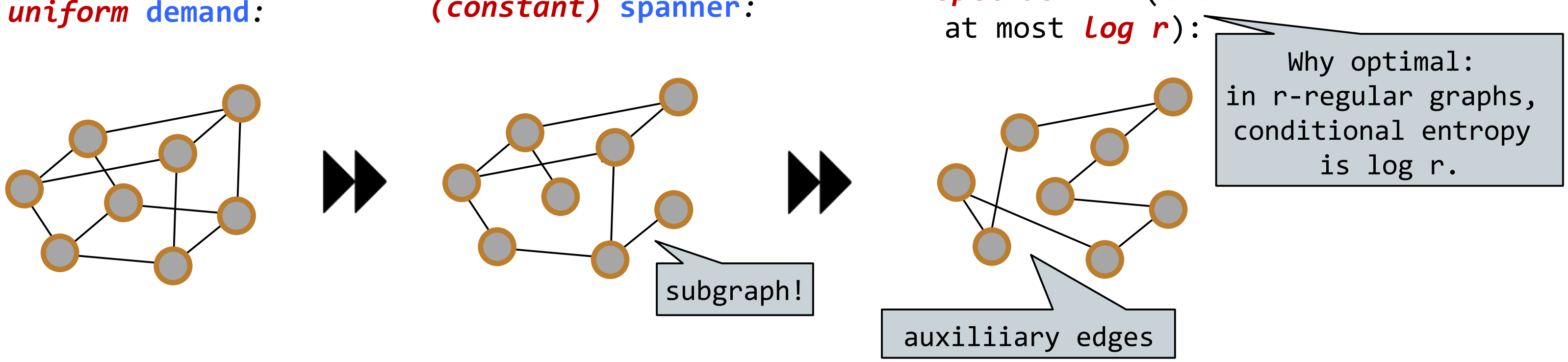
Theorem: If demand matrix is regular and uniform, and if we can find a constant distortion, linear sized (i.e., constant, sparse) spanner for this request graph: then we can design a constant degree DAN providing an optimal expected route length *(i.e., $O(H(X|Y)+H(Y|X))$.*

Our degree reduction trick again!

r-regular *and* ***uniform*** ***demand****:*

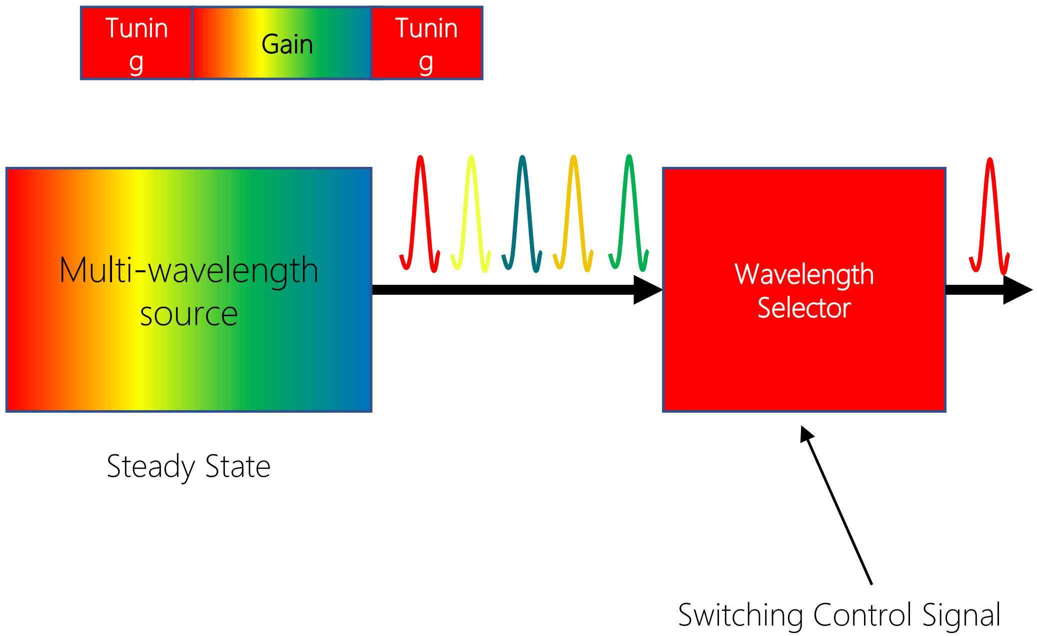
Sparse, ***irregular*** ***(constant)*** ***spanner****:*

Constant *degree* ***optimal*** **DAN** (ERL at most ***log r***):



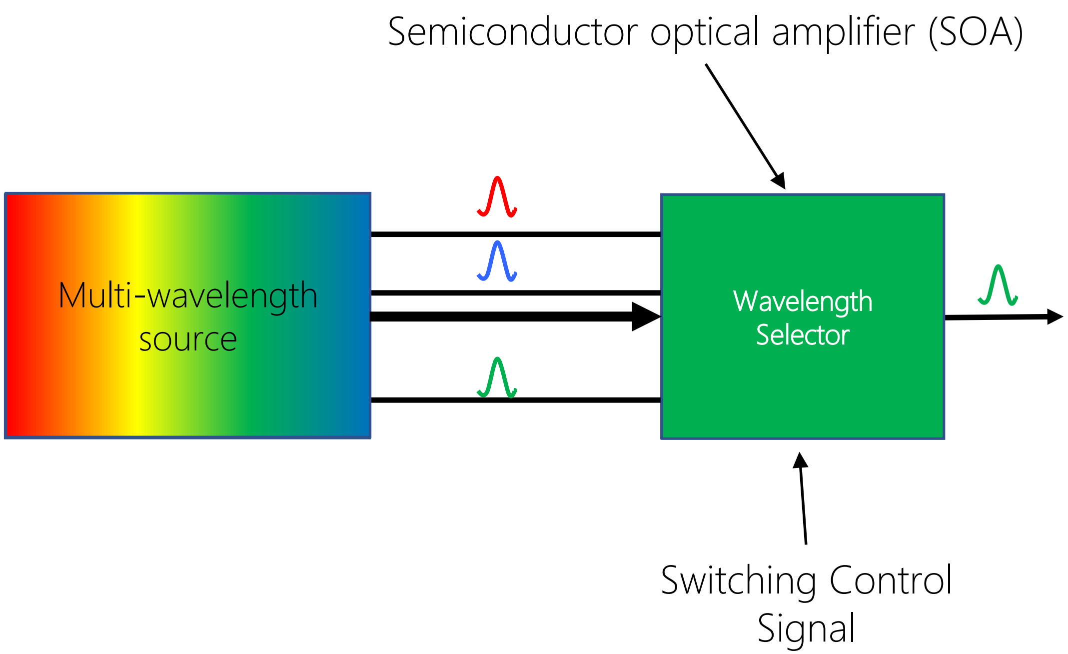
Idea

Disaggregated Laser



Ballani et al., Sirius, ACM SIGCOMM 2020.

Example Design



Sirius also implemented other designs
(details in the paper)