Self-Adjusting Networks
The Power of Choices in Datacenter Network Design

Stefan Schmid

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

(Folklore)
Trend

Data-Centric Applications

Datacenters ("hyper-scale")

Interconnecting networks: a critical infrastructure of our digital society.

Source: Facebook
Traffic does not only grow but also has much structure:

traffic matrices sparse and skewed  
traffic bursty over time
Recent Representation of Trace Structure:

Complexity Map

Griner et al., SIGMETRICS 2020
Recent Representation of Trace Structure: Complexity Map

Different structures!
Many flavors, but in common: fixed and oblivious to actual demand.
Highway which ignores actual traffic: frustrating!

One Solution?

Today: Demand-Oblivious Topology

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

Today: Demand-Oblivious Topology

Bandwidth tax!
Emerging Alternatives

E.g., Demand-Aware Reconfigurable Datacenter

e.g., mirrors

new flexible interconnect
Emerging Alternatives

E.g., Demand-Aware Reconfigurable Datacenter

e.g., mirrors

new flexible interconnect
Emerging Alternatives
E.g., Demand-Aware Reconfigurable Datacenter

Matches demand

demand matrix:

1 2 3 4 5 6 7 8
1 0 0 0 0 0 0 0 0
2 1 0 0 0 0 0 0 0
3 1 1 0 0 0 0 0 0
4 1 1 1 0 0 0 0 0
5 1 1 1 1 0 0 0 0
6 1 1 1 1 1 0 0 0
7 1 1 1 1 1 1 0 0
8 1 1 1 1 1 1 1 0

E.g., mirrors
new flexible interconnect
Emerging Alternatives
E.g., Demand-Aware Reconfigurable Datacenter
e.g., mirrors
new flexible interconnect
new demand:
Emerging Alternatives

E.g., Demand-Aware Reconfigurable Datacenter

Matches demand:

new demand:

e.g., mirrors

new flexible interconnect
Crazy? No!

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

Optical Circuit Switch

→ Optical Circuit Switch rapid adaption of physical layer
  → Based on rotating mirrors
The Big Picture

- Flexibility
- Structure
- Self-Adjusting Networks
- Efficiency

Now is the time!
But: Introduces Tradeoff

ProjecToR is demand-aware through reconfigurations

However, reconfigurations take time
Spectrum of Topologies

Diverse topology components:
→ demand-obliviou and
  demand-aware
Diverse topology components:

→ demand-oblivious and demand-aware
→ static vs dynamic

Spectrum of Topologies

- e.g., Clos (SIGCOMM‘08), BCube (SIGCOMM‘09), Xpander (SIGCOMM‘17)
Diverse topology components:
→ demand-oblivious and demand-aware
→ static vs dynamic

Dynamic

Demand-oblivious

Static

e.g., ProjecToR (SIGCOMM'16), FireFly (SIGCOMM'14), SplayNet (ToN'16)

e.g., Clos (SIGCOMM'08), BCube (SIGCOMM'09), Xpander (SIGCOMM'17)
Spectrum of Topologies

Diverse topology components:
→ demand-oblivious and demand-aware
→ static vs dynamic

Static

Demand-oblivious

→ e.g., Clos (SIGCOMM'08), BCube (SIGCOMM'09), Xpander (SIGCOMM'17)

Demand-aware

→ e.g., RotorNet (SIGCOMM'17), Opera (NSDI'20), Sirius (SIGCOMM'20)

→ e.g., ProjecToR (SIGCOMM'16), FireFly (SIGCOMM'14), SplayNet (ToN'16)

Dynamic
Diverse topology components:
→ demand-oblivious and demand-aware
→ static vs dynamic
Spectrum of Topologies

Diverse topology components:
→ demand-oblivious and demand-aware
→ static vs dynamic

Which approach is best?
Diverse topology components:
→ demand-oblivious and demand-aware
→ static vs dynamic

Which approach is best?

As always in CS: It depends…
Agenda
Exploit Trends for Throughput

More!
Structure

Cerberus

Technology

Throughput
Unified Network Model

Two-Layers ToR Interconnect

Typical rack internconnect: ToR-Matching-ToR (TMT) model
Typical rack internconnect: **ToR-Matching-ToR (TMT) model**
Rotor switch: periodic matchings (demand-oblivious)
Demand-Aware Switch

Demand-aware switch: optimized matchings

Si:
Static switches: **combine** for optimized static topology

- **S1:**
- **S2:**
- **S3:**

  e.g., tree, Xpander, Clos
Unified Model: From Switches to Topologies

Typical rack internconnect: ToR-Matching-ToR (TMT) model
→ All spine switches are rotor switches
→ Can use 1 or 2 hop routings (VLB)
→ Emulating a complete graph using (TDMA)
Unified Model

Demand-Aware Net

→ All spine switches are demand-aware switches
→ Can use only 1 hop routings (multi-hop, in on-going work)
→ Temporal / dynamic network

e.g., Ghobadi et al., SIGCOMM 2016
Unified Model

**Expander-Net**

- All spine switches are static switches
- Uses multi-hop routing
- Use known static topologies: e.g., Xpander*, Clos, electrical

*Valadarsky et al., CoNEXT 2016*
Design Tradeoffs (1)

The “Awareness-Dimension”

Rotor

Demand-Aware

Good for all-to-all traffic!
- Oblivious: very fast
  periodic direct connectivity
- Simpler control plane?

Good for elephant flows!
- Optimizable toward traffic
- But slower
Design Tradeoffs (1)

The “Awareness-Dimension”

Good for all-to-all traffic!
- Oblivious: very fast
  periodic direct connectivity
  Simpler control plane?

Good for elephant flows!
- Optimizable toward traffic
  But slower

Compared to static networks: latency tax!
Design Tradeoffs (2)

The “Flexibility-Dimension”

Good for high throughput!
→ direct connectivity saves bandwidth along links

Good for low latency!
→ no need to wait for reconfigurable links
→ compared to dynamic: bandwidth tax (multi-hop)
Design Tradeoffs (2)
The “Flexibility-Dimension”

Good for high throughput!
→ Direct connectivity saves bandwidth along links

Good for low latency!
→ No need to wait for reconfigurable links
→ Compared to dynamic: bandwidth tax (multi-hop)
Summary: Tax Map

Latency Tax

Bandwidth Tax

- Static
- Rotor
- Demand-Aware
The Spectrum of Traffic

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
Main Observations

→ **Observation 1**: Different topologies provide different tradeoffs.

→ **Observation 2**: Different traffic requires different topology types.

→ **Observation 3**: A *mismatch of demand* and topology can decrease *throughput* and increase *flow completion times*. 
Main Observations

- **Observation 1:** Different topologies provide different tradeoffs.

- **Observation 2:** Different traffic requires different topology types.

- **Observation 3:** A *mismatch of demand* and topology can decrease *throughput* and increase *flow completion times*.

So: Can we match traffic to topology?
Examples: Match or Mismatch?

Static - oblivious

Dynamic

Rotor

Demand-Aware

Demand-aware

Static

Shuffling
All-to-All

ML
Large flows

Delay
sensitive

Telemetry
/ control
Examples: Match or Mismatch?

- Rotor
- Static
- Demand-Aware
- Demand-oblivious

- Shuffling
  - All-to-All

- ML
  - Large flows

- Delay sensitive

- Telemetry / control
Examples: Match or Mismatch?

- Static
- Demand-aware
- Demand-oblivious
- Dynamic

```
Shuffling
All-to-All
```
```
ML
Large flows
```
```
Delay
sensitive
```
```
Telemetry
/ control
```

Examples:
- Match or Mismatch?
- Rotor
- Demand-Aware
- Static
- Demand-oblivious
- Demand-aware
- Static
- ?
- Telemetry / control
Cerberus: It's a Match!

- **Rotor**
- **Demand-Aware**
- **Static**
- **Demand-oblivious**

**Dynamic**

**Static**

- **Shuffling All-to-All**
- **ML Large flows**
- **Delay sensitive**
- **Telemetry / control**
Our system Cerberus* serves traffic on the “best topology”!

* Griner et al., ACM SIGMETRICS 2022
Flow Size Matters

Observation 1: Most flows are small, most bytes in big flows.

---

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.
Flow Size Matters

Observation 1: Most flows are small, most bytes in big flows.

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.

---

Flow transmission time (40Gbps)

CDF of bytes

Flow size (bytes)
Flow Size Matters

--- Observation 1: Most flows are small, most bytes in big flows.
--- 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 1: Most flows are small, most bytes in big flows.

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.
Observation 1: Most flows are small, most bytes in big flows.
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

Optical Switches

1  2  3  4  5  6  7  8
Cerberus*  

* 3-headed dog from Greek mythology
Scheduling: Small flows go via static switches...
Scheduling: ... medium flows via rotor switches...
Scheduling: ... and large flows via demand-aware switches (if one available, otherwise via rotor).
Cerberus Framework

Flow sizes

Optimal Partition (static, rotor, demand-aware)

Flow size thresholds (small, medium, large)

n ToRs
k spine switches
reconfig times
$R_r, R_d, \delta$

Throughput analysis

vs Rotor-Net and Expander-Net
Throughput Analysis

Demand Matrix

\[ T \]

**Metric:** throughput of a demand matrix...
Throughput Analysis

\[ T \times \theta(T) \]

**Metric:** throughput of a demand matrix...

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

\[ 0 \leq \theta(T) \leq 1. \]
Throughput Analysis

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 \(\theta^*\): worst case \(T\)

Abdu et al., SC 2016
Namyar et al., SIGCOMM 2021
Throughput: Rotor-Net

Demand Matrix

\[ T \]

Permutation matrix

\[ \theta(T) \leq \frac{1}{2 - \phi(T)} \cdot \frac{\delta}{R_r + \delta} \]

Skew parameter

Bandwidth tax

Latency tax

\[ \theta^* \leq \frac{n}{2n - 1} \cdot \frac{\delta}{R_r + \delta} \]
Throughput: Expander-Net

Demand Matrix

\( T \)

Permutation matrix

\[ \theta^* \leq \frac{1}{\text{epl}(G(k))} \]

Expected path length

Bandwidth tax

Namyar et al., SIGCOMM 2021
Throughput: Demand-Aware

Permutation matrix is the best demand matrix for demand-aware net!
Throughput: Cerberus

Demand Matrix

\[ T \]

\[ K_s \]
static switches

\[ K_r \]
rotor switches

\[ K_d \]
demand-aware switches

\[ \theta(T) = \frac{\hat{T}(1, \ell)}{nk_d^*} \left( R_d \right) \left[ \frac{1}{|f|} \right] + \frac{1}{r} \]

Bandwidth tax  Latency tax
Throughput: Summary

Demand Matrix

For the given input parameters: \( n, k, R_d, R_r \)

<table>
<thead>
<tr>
<th></th>
<th>expander-net</th>
<th>rotor-net</th>
<th>Cerberus</th>
</tr>
</thead>
<tbody>
<tr>
<td>BW-Tax</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>LT-Tax</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( \theta(T) )</th>
<th>Thm 2</th>
<th>Thm 3</th>
<th>Thm 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \theta^* )</td>
<td>0.53</td>
<td>0.45</td>
<td>Open</td>
</tr>
<tr>
<td>Datamining</td>
<td>0.53</td>
<td>0.6</td>
<td>0.8 (+33%)</td>
</tr>
<tr>
<td>Permutation</td>
<td>0.53</td>
<td>0.45</td>
<td>( \approx 1 (+88%) )</td>
</tr>
<tr>
<td>Case Study</td>
<td>0.53</td>
<td>0.66</td>
<td>0.9 (+36%)</td>
</tr>
</tbody>
</table>
Diverse traffic requires diverse technologies/topologies

Cerberus aims to assign traffic to its best topology
Depending on flow size

Skipped: simulations and prototype

Many challenges
Impact on routing and congestion control
Sensitivity analysis
Simulation & prototyping
Websites

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

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

Thank you!
Further Reading

Cerberus: The Power of Choices in Datacenter Topology Design
A Throughput Perspective

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

The bandwidth and latency requirements of modern datacenter applications have led researchers to propose various topology designs using static, dynamic demand-oblivious (rotor), and/or dynamic demand-aware switches. However, given the diverse nature of datacenter traffic, there is little consensus about how these designs would fare against each other. In this work, we analyze the throughput of existing topology designs under different traffic patterns and study their unique advantages and potential costs in terms of bandwidth and latency "tax". To overcome the identified inefficiencies, we propose Cerberus, a unified, two-layer leaf-spine optical datacenter design with three topology types. Cerberus systematically matches different traffic patterns with their most suitable topology type. It leverages passive flow capture, transmitting view static lookup.

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 present a novel efficiency and complexity characterization of real-world packet traces.
**Static DAN**

**Overview: Models**

Toward Demand-Aware Networking: A Theory for Self-Adjusting Networks

Chen Avin
Ben Gurion University, Israel
avin@bgu.ac.il
Stefan Schmid
University of Vienna, Austria
stefan.schmid@univie.ac.at

This article is an extended note submitted to CCR. It has not been peer reviewed. The authors take full responsibility for this article’s technical content. Comments can be posted through CCR Online.

**ABSTRACT**

The physical topology is emerging as an asset to linearize in the ongoing effort to make communication networks more flexible. While first experimental results indicate that these flexible topologies can be exploited to regulate and optimise the networks toward the workload it services, e.g., providing the same bandwidth at lower administration cost, little is known today about the fundamental algorithmic principles underlying the design of reconnectable networks. This paper initiates the study of the theory of demand-aware, self-adjusting networks. Our main promise is that self-adjusting networks--networks that dynamically restructure themselves in response to dynamic network requirements--allow one to develop scalable algorithms that adapt to the workload. To this end, we present a taxonomy of topology optimization design of efficient demand networks has received much attention over the last years. The topologies underlying modern demand networks range from trees [7], [11] over hypercubes [5], [10] to expander networks [14] and provide high connectivity at low cost [5].

Until now, these networks also have in common that their topology is fixed and oblivious to the actual demand. Consequently, they are static. An overview of models is presented in Section 1. The static model is based on the observation that the bandwidth in a network can be allocated in a way that is independent of the actual demand, i.e., the demand is not considered at all.

Automatically generated by the task of the paper is the initialization of models of demand-aware networks (MDNs), and in particular the design of bounded degree networks, which are networks where the number of connections to any node is bounded. It is known that bounded degree networks can be rearranged to meet the demand while maintaining the bandwidth. However, such rearrangements are often complex and delicate. The key insight of the paper is that the bandwidth in a network can be allocated in a way that is independent of the actual demand, i.e., the demand is not considered at all.

The main insight of this paper is that the bandwidth in a network can be allocated in a way that is independent of the actual demand, i.e., the demand is not considered at all. The key insight is that the bandwidth in a network can be allocated in a way that is independent of the actual demand, i.e., the demand is not considered at all.

1. Introduction

The problems of the following sections are motivated by the observation that the bandwidth in a network can be allocated in a way that is independent of the actual demand, i.e., the demand is not considered at all. The key insight is that the bandwidth in a network can be allocated in a way that is independent of the actual demand, i.e., the demand is not considered at all.

Robust DAN

**Overview: Models**

1. Introduction

The problems of the following sections are motivated by the observation that the bandwidth in a network can be allocated in a way that is independent of the actual demand, i.e., the demand is not considered at all. The key insight is that the bandwidth in a network can be allocated in a way that is independent of the actual demand, i.e., the demand is not considered at all.

Dynamic DAN

**Overview: Models**

1. Introduction

The problems of the following sections are motivated by the observation that the bandwidth in a network can be allocated in a way that is independent of the actual demand, i.e., the demand is not considered at all. The key insight is that the bandwidth in a network can be allocated in a way that is independent of the actual demand, i.e., the demand is not considered at all.

Concurrent DANs

**Overview: Models**

1. Introduction

The problems of the following sections are motivated by the observation that the bandwidth in a network can be allocated in a way that is independent of the actual demand, i.e., the demand is not considered at all. The key insight is that the bandwidth in a network can be allocated in a way that is independent of the actual demand, i.e., the demand is not considered at all.

Further Reading

**Overview: Models**

1. Introduction

The problems of the following sections are motivated by the observation that the bandwidth in a network can be allocated in a way that is independent of the actual demand, i.e., the demand is not considered at all. The key insight is that the bandwidth in a network can be allocated in a way that is independent of the actual demand, i.e., the demand is not considered at all.

**Static Optimality**

ReNets: Toward Statically Optimal Self-Adjusting Networks

Chen Avin1
Ben Gurion University, Israel
avin@bgu.ac.il
Stefan Schmid2
University of Vienna, Austria

**Abstract**

This paper studies the design of self-adjusting networks whose topology dynamically adapts to the workload, in an online and demand-aware manner. This problem is motivated by emerging optical technologies which allow to reconfigure the network topology at runtime. Our main contribution is ReNets, a self-adjusting network that maintains a balance between the benefits and costs of reconfigurations. In particular, we show that ReNets are statistically optimal for arbitrary sparse communication demands, i.e., perform at least as good as any fixed communication network designed with a perfect knowledge of the future demand. Furthermore, ReNets provide compact and fast routing, by leveraging known self-adjusting factorizations.

1. Introduction

Modern demand-aware networks rely on efficient network topologies (based on regular graphs [2], hypercubes [3], or expanders [4]) to provide a high connectivity at low cost [3]. These demand-aware networks have in common that their topology is fixed and oblivious to the actual demand (i.e., workload or communication pattern) they currently serve. Rather, they are designed for all-to-all communication patterns, by ensuring properties such as full benevolent branchability or (O(log n) edge-lengths between any node pair in a constant-degree n-node network. However, demand-oblivious networks can be inefficient for more specific demand patterns, as they usually use in
Selected References

On the Complexity of Traffic Traces and Implications
Chen Avin, Manya Ghobadi, Chen Griner, and Stefan Schmid.
ACM SIGMETRICS, Boston, Massachusetts, USA, June 2020.

Survey of Reconfigurable Data Center Networks: Enablers, Algorithms, Complexity
Klaus-Tycho Foerster and Stefan Schmid.

Toward Demand-Aware Networking: A Theory for Self-Adjusting Networks (Editorial)
Chen Avin and Stefan Schmid.

Dynamically Optimal Self-Adjusting Single-Source Tree Networks
Chen Avin, Kaushik Mondal, and Stefan Schmid.
14th Latin American Theoretical Informatics Symposium (LATIN), University of Sao Paulo, Sao Paulo, Brazil, May 2020.

Demand-Aware Network Design with Minimal Congestion and Route Lengths
Chen Avin, Kaushik Mondal, and Stefan Schmid.

Distributed Self-Adjusting Tree Networks
Bruna Peres, Otavio Augusto de Oliveira Souza, Olga Goussevskaia, Chen Avin, and Stefan Schmid.

Efficient Non-Segregated Routing for Reconfigurable Demand-Aware Networks
Thomas Fenz, Klaus-Tycho Foerster, Stefan Schmid, and Anaïs Villedieu.
IFIP Networking, Warsaw, Poland, May 2019.

DaRTree: Deadline-Aware Multicast Transfers in Reconfigurable Wide-Area Networks
Long Luo, Klaus-Tycho Foerster, Stefan Schmid, and Hongfang Yu.

Demand-Aware Network Designs of Bounded Degree
Chen Avin, Kaushik Mondal, and Stefan Schmid.
31st International Symposium on Distributed Computing (DISC), Vienna, Austria, October 2017.

SplayNet: Towards Locally Self-Adjusting Networks
Stefan Schmid, Chen Avin, Christian Scheideler, Michael Borokhovich, Bernhard Haeupler, and Zvi Lotker.

Characterizing the Algorithmic Complexity of Reconfigurable Data Center Architectures
Klaus-Tycho Foerster, Monia Ghobadi, and Stefan Schmid.