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

(Folklore)
Today’s Datacenters

Fixed and Demand-Oblivious Topology

Many flavors, but in common: fixed and oblivious to actual demand.
Today’s Datacenters
Fixed and Demand-Oblivious Topology

Highway which ignores actual traffic: frustrating!

Many flavors, but in common: fixed and oblivious to actual demand.
Our Vision:
Flexible and Demand-Aware Topologies
Our Vision:
Flexible and Demand-Aware Topologies

e.g., mirrors
new flexible interconnect
Our Vision:
Flexible and Demand-Aware Topologies

demand matrix:

e.g., mirrors
new flexible interconnect
Our Vision:
Flexible and Demand-Aware Topologies

Matches demand

demand matrix:
e.g., mirrors
new flexible interconnect
Our Vision:
Flexible and Demand-Aware Topologies

e.g., mirrors
new flexible interconnect
Our Vision:
Flexible and Demand-Aware Topologies

Matches demand

e.g., mirrors
new flexible interconnect

new demand:
Our Vision:
Flexible and Demand-Aware Topologies

Self-Adjusting Networks

new demand:
e.g., mirrors

new flexible interconnect
Sounds Crazy?
Emerging Enabling Technology.

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

\[ \text{Spectrum of prototypes} \]

- Different sizes, different reconfiguration times
- From our last ACM SIGCOMM OptSys’19 workshop

Prototype 1
Prototype 2
Prototype 3
Putting Things Together

Demand-Aware Networks

Flexibility

Demand Structure

Demand-Aware Networks

Efficiency

Now is the time!
Now is the time!

But how much does it help? As usual in computer science: it depends! We need metrics for demand structure and for possible efficiency.
Our Perspective
Information Theory and Entropy

Demand entropy: Spatial and temporal structure of traffic

Entropy: A tight metric for the achievable route lengths in demand-aware networks
Question 1:

How to Quantify such “Structure” in the Demand?
Intuition

Which demand has more structure?

→ Traffic matrices of two different distributed ML applications
  → GPU-to-GPU
Intuition
Which demand has more structure?

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

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”

Original → Randomize rows → Uniform

Increasing complexity (systematically randomized)
More structure (compresses better)
Trace Complexity
Information-Theoretic Approach
“Shuffle&Compress”
Trace Complexity
Information-Theoretic Approach
“Shuffle&Compress”

Difference in size (entropy)?

Difference in size (entropy)?
Trace Complexity
Information-Theoretic Approach
“Shuffle&Compress”

Can be used to define 2-dimensional complexity map!
Trace Complexity

Complexity Map

bursty uniform

No structure

bursty & skewed skewed

temporal complexity

non-temporal complexity
Trace Complexity

Complexity Map

- bursty
- uniform
- non-temporal complexity
- bursty & skewed
- temporal complexity

Different structures!

No structure
Trace Complexity

Complexity Map

non-temporal complexity

Potential gain!

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:

1 INTRODUCTION
Packet traces collected from networking applications, such as datacenter traffic, have been shown to feature much structure: datacenter traffic matrices are sparse and skewed [16, 39], exhibit
Question 2:
How to Exploit Structure Algorithmically? Metrics for Achievable Efficiency?

Insight: Information-theoretic perspective useful here as well!

Case Study “Route Lengths”
Models and Connection to Datastructures & Coding

Traditional networks (worst-case traffic)

More structure: lower routing cost
Models and Connection to Datastructures & Coding

Traditional networks
(worst-case traffic)

Demand-aware networks
(spatial structure)

More structure: lower routing cost
Models and Connection to Datastructures & Coding

Traditional networks (worst-case traffic)
Demand-aware networks (spatial structure)
Self-adjusting networks (temporal structure)

More structure: lower routing cost
Models and Connection to Datastructures & Coding

- Traditional networks (worst-case traffic)
- Demand-aware networks (spatial structure)
- Self-adjusting networks (temporal structure)

More structure: lower routing cost
Models and Connection to Datastructures & Coding

- Traditional networks (worst-case traffic)
- Demand-aware networks (spatial structure)
- Self-adjusting networks (temporal structure)

- More structure: lower routing cost

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

- More structure: improved access cost / shorter codes
Models and Connection to Datastructures & Coding

Traditional networks (worst-case traffic)
Demand-aware networks (spatial structure)
Self-adjusting networks (temporal structure)

More structure: **lower routing cost**

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

More structure: improved **access cost / shorter codes**
Models and Connection to Datastructures & Coding

More than an analogy!

- **Traditional networks** (worst-case traffic)
- **Demand-aware networks** (spatial structure)
- **Self-adjusting networks** (temporal structure)

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

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

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

More structure: improved **access cost** / shorter **codes**
Case Study “Route Lengths”

Constant-Degree Demand-Aware Network

<table>
<thead>
<tr>
<th>Sources</th>
<th>Destinations</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>0</td>
<td>2/65</td>
<td>1/13</td>
<td>1/65</td>
<td>1/65</td>
<td>2/65</td>
<td>3/65</td>
</tr>
<tr>
<td>2</td>
<td>2/65</td>
<td>0</td>
<td>1/13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2/65</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1/13</td>
<td>1/65</td>
<td>0</td>
<td>2/65</td>
<td>0</td>
<td>0</td>
<td>1/13</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1/65</td>
<td>0</td>
<td>2/65</td>
<td>0</td>
<td>4/65</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1/65</td>
<td>0</td>
<td>3/65</td>
<td>4/65</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2/65</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3/65</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3/65</td>
<td>2/65</td>
<td>1/13</td>
<td>0</td>
<td>0</td>
<td>3/65</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

ERL(\mathcal{D}, N) = \sum_{(u, v) \in \mathcal{D}} p(u, v) \cdot d_N(u, v)
Case Study “Route Lengths”

Constant-Degree Demand-Aware Network

<table>
<thead>
<tr>
<th>Sources</th>
<th>Destinations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2 0 3 4 5 6 7</td>
</tr>
<tr>
<td>2</td>
<td>2/65 0 1/65 0 0 2/65</td>
</tr>
<tr>
<td>3</td>
<td>1/13 1/65 0 2/65 0 0 1/13</td>
</tr>
<tr>
<td>4</td>
<td>1/65 0 2/65 0 4/65 0 0</td>
</tr>
<tr>
<td>5</td>
<td>1/65 0 3/65 0 0 0 0</td>
</tr>
<tr>
<td>6</td>
<td>2/65 0 0 0 0 0 3/65</td>
</tr>
<tr>
<td>7</td>
<td>3/65 0 0 0 3/65 0 0</td>
</tr>
</tbody>
</table>

ERL(\mathcal{D}, N) = \sum_{(u,v)\in\mathcal{D}} p(u, v) \cdot d_N(u, v)
Case Study “Route Lengths”

Constant-Degree Demand-Aware Network

$$\text{ERL}(\mathcal{D}, N) = \sum_{(u, v) \in \mathcal{D}} p(u, v) \cdot d_N(u, v)$$
Case Study “Route Lengths”

Constant-Degree Demand-Aware Network

\[
\text{ERL}(\mathcal{D},N) = \sum_{(u,v) \in \mathcal{D}} p(u, v) \cdot d_N(u, v)
\]
## Entropy Lower Bound

### Huffman tree: “ego-tree”

<table>
<thead>
<tr>
<th>Sources</th>
<th>Destinations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 (\frac{2}{65}) (\frac{1}{13}) (\frac{1}{65}) (\frac{1}{65}) (\frac{2}{65}) (\frac{3}{65})</td>
</tr>
<tr>
<td>2</td>
<td>(\frac{2}{65}) 0 (\frac{1}{65}) 0 0 0 (\frac{2}{65})</td>
</tr>
<tr>
<td>3</td>
<td>(\frac{1}{13}) (\frac{1}{65}) 0 (\frac{2}{65}) 0 0 (\frac{1}{13})</td>
</tr>
<tr>
<td>4</td>
<td>(\frac{1}{65}) 0 (\frac{2}{65}) 0 (\frac{4}{65}) 0 0</td>
</tr>
<tr>
<td>5</td>
<td>(\frac{1}{65}) 0 (\frac{3}{65}) (\frac{4}{65}) 0 0 0</td>
</tr>
<tr>
<td>6</td>
<td>(\frac{2}{65}) 0 0 0 0 0 (\frac{3}{65})</td>
</tr>
<tr>
<td></td>
<td>0 (\frac{2}{65}) (\frac{1}{13}) 0 0 (\frac{3}{65}) 0</td>
</tr>
</tbody>
</table>
Entropy Lower Bound

\[ \text{ERL} = \Omega(H_\Delta(Y|X)) \]
Idea for algorithm:
- union of trees
- reduce degree
- but keep distances

What about dynamic case?
Idea for algorithm:

- union of trees
- reduce degree
- but keep distances

Ok for sparse demands

- not everyone gets tree
- helper nodes

What about dynamic case?
Dynamic Setting

→ Dynamic the same:
  → union of dynamic ego-trees

→ E.g., SplayNets

→ Online algorithms
Dynamic Objectives

- Awareness: Demand-Aware
- Topology: Reconfigurable
- Input: Offline, Online
- Algorithm: OFF, ON
- Property: Static Optimality, Dynamic Optimality, Working Set
Future Work: Models, Metrics, Alg.os

Notion of self-adjusting networks opens a large uncharted field with many questions:

→ Metrics and algorithms: by how much can load be lowered, energy reduced, quality-of-service improved, etc. in demand-aware networks? Even for route length not clear!
→ How to model reconfiguration costs?
→ Impact on other layers?

Requires knowledge in networking, distributed systems, algorithms, performance evaluation.
**Websites**

http://self-adjusting.net/

Project website

https://trace-collection.net/

Trace collection website
Further Reading

Static DAN

Demand-Aware Network Designs of Bounded Degree
Chen Avin, Konrad Slind, and Stefan Schmid

Abstract: Traditionally, networks such as datacenter networks are designed to optimize worst-case performance under adversarial traffic patterns. Such network designs can however be far from optimal when considering realistic workload and traffic patterns which they serve. This insight led to the development of demand-aware datacenter networks which can be reconfigured dynamically depending on the workload.

Maintained for this reason, this paper initiates the algorithmic study of demand-aware networks (DANs), and in particular the design of bounded-degree static DANs. In a static DAN, the network topology is fixed, and all traffic is assumed to follow some communication request distribution. A bounded-degree static DAN consists of a bounded maximum degree and a bounded minimum degree. The core of the paper is the design of efficient algorithms for dynamic topology control and reconfiguration in static DANs.

Overview: Models

Toward Demand-Aware Networking: A Theory for Self-Adjusting Networks
Chen Avin, Stefano Schmid

Abstract: The physical topology is emerging as the next frontier in ongoing efforts to optimize communication networks. While these efforts are expected to magnify the performance gap between communication networks and other domains and the expected benefits are likely to be realized first in communication networks. As a result, this paper focuses on the design of efficient algorithms for dynamic topology control and reconfiguration in static DANs.

Static Optimality

ReNets: Toward Statically Optimal Self-Adjusting Networks
Chen Avin, Stefano Schmid

Abstract: This paper studies the design of statically optimal 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 datacenter topology at runtime. Our main contribution is ReNets, a self-adjusting network which maintains a balance between the benefits and costs of reconfigurations. In particular, we show that ReNets are statically optimal for arbitrary symmetric communication demand, i.e., perform as in 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 three key self-adjusting mechanisms.

Robust DAN

Dynamic DAN

SplayNet: Towards Locally Self-Adjusting Networks
Stefan Schmid, Chen Avin, Christian Schindelhauer, Michael Berekovich, Benjamin Hansper, Zili Loker

Abstract: This paper initiates the study of highly self-adjusting networks whose topology adapts in online time. The main reason is that such networks are expected to be the key building blocks in future datauser networks. In contrast to prior works which dynamically adjust the placement of the nodes and the links in the network, the main contribution of this paper is to introduce the concept of a splaynet, a network whose topology is dynamically adjusted based on the topology of a static backbone network. In particular, we show how to design a splaynet that optimizes a given objective function, such as minimizing the number of hops, while balancing the costs of reconfigurations. We also show how to design a splaynet that adapts to the topology of a static backbone network.

Concurrent DANs

CBNet: Minimizing Adjustments in Concurrent Demand-Aware Tree Networks
Shivam Aggarwal and Oliver Spatscheck

Abstract: Concurrent demand-aware tree networks are networks in which the nodes are arranged in a binary tree which requests real-time identifications. A BST topology is attractive as it supports greedy routing, a node can directly locate its parent in worst-case Θ(1) iterations. However, in a concurrent setting, such a topology incurs Ω(n^2) adjustments. We prove that an optimal sequence of adjustments exists in which any node first finds a direct parent in the binary tree of nodes, and then requests a real-time identification. We also show that the algorithm for minimizing adjustments is optimal in the worst case.
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.

Measuring the Complexity of Network Traffic Traces
Chen Griner, Chen Avin, Manya Ghobadi, and Stefan Schmid.
arXiv, 2019.

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.