Self-Adjusting Trees Using Rotor Walks

Chen Avin, Marcin Bienkowski, Iosif Salem, Robert Sama, Stefan Schmid, Paweł Schmidt

“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
The Problem

Huge Infrastructure, Inefficient Use

---

Network equipment reaching capacity limits
→ Transistor density rates stalling
→ “End of Moore’s Law in networking” [1]

---

Hence: more equipment, larger networks

---

Resource intensive and: inefficient

Annoying for companies, opportunity for researchers

[1] Source: Microsoft, 2019
Root Cause

Fixed and Demand-Oblivious Topology

How to interconnect?
Root Cause

Fixed and Demand-Oblivious Topology

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

Fixed and Demand-Oblivious Topology

Highway which ignores actual traffic: frustrating!

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

e.g., mirrors

new flexible interconnect
Our Vision
Flexible and Demand-Aware Topologies

demand matrix:

1 2 3 4 5 6 7 8
1 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

new demand:
Our Vision
Flexible and Demand-Aware Topologies

Matches demand:

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

Self-Adjusting Networks

new demand:

1 2 3 4 5 6 7 8

1 2 3 4 5 6 7 8

e.g., mirrors

new flexible interconnect
Our Motivation
Much Structure in the Demand

Empirical studies:

traffic matrices sparse and skewed

traffic bursty over time

Our hypothesis: can be exploited.
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.”
The Big Picture

Flexibility

New!

Structure

Self-Adjusting Networks

Efficiency

Now is the time!
**Static Problem**

Demand-Aware Network of Bounded Degree

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

\[
WPL(\mathcal{D},N) = \sum_{(u,v) \in \mathcal{D}} p(u,v) \cdot d_N(u,v)
\]
### Static Algorithm

**Reduction to Ego-Trees**

<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>
</tbody>
</table>

Huffman tree: “ego-tree”
Static Algorithm

Reduction to Ego-Trees

→ Idea for algorithm:
  → union of trees
  → reduce degree
  → but keep distances
This Paper: Dynamic Maintaining Ego-Trees Dynamically

Input: sequence of nodes \( \sigma = (v_1, v_2, \ldots) \)
Cost: access cost + number of swaps
Like splay trees, but unordered trees
Goal: online algorithm which is competitive to offline
Useful property: most recently used (MRU)
This Paper: Dynamic
Maintaining Ego-Trees Dynamically

Input: sequence of nodes $\sigma = (v_1, v_2, \ldots)$
Cost: access cost + number of swaps
Like splay trees, but unordered trees
Goal: online algorithm which is competitive to offline
Useful property: most recently used (MRU)
Input: sequence of nodes $\sigma = (v_1, v_2, \ldots)$
Cost: access cost + number of swaps
Like splay trees, but unordered trees
Goal: online algorithm which is competitive to offline
Useful property: most recently used (MRU)

How to maintain MRU?
Swap $u, v$: breaks MRU!
Input: sequence of nodes \( \sigma = (v_1, v_2, \ldots) \)
Cost: access cost + number of swaps
Like splay trees, but unordered trees
Goal: online algorithm which is competitive to offline
Useful property: most recently used (MRU)

How to maintain MRU?
Idea: pushdown along path? Not competitive!
Input: sequence of nodes $\sigma = (v_1, v_2, ...)$
Cost: access cost + number of swaps
Like splay trees, but unordered trees
Goal: online algorithm which is competitive to offline
Useful property: most recently used (MRU)
This Paper: Dynamic
Maintaining Ego-Trees Dynamically

Input: sequence of nodes $\sigma = (v_1, v_2, \ldots)$
Cost: access cost + number of swaps
Like splay trees, but unordered trees
Goal: online algorithm which is competitive to offline
Useful property: most recently used (MRU)

Random walk preservers MRU!
Constant competitive.
Competitive deterministic?
Our Contributions

→ Rotor-push: select push-down path by rotor walk
→ Each node has a toggle switch, left or right
→ Upon traversal, flip the switch
→ “Deterministic random walk”
Rotor-push: select push-down path by \textit{rotor walk}

- Each node has a toggle switch, left or right
- Upon traversal, flip the switch
- “Deterministic random walk”

Theorem: gives \textbf{12-competitive} tree

- We also improved \textit{random push} bound from 60 to 16
Empirical Results

**Takeaway 1:** The larger the network, the more beneficial self-adjustments compared to static.

**Takeaway 2:** The more locality in the demand, the more beneficial as well.

**Takeaway 3:** In practice, Rotor Push and Random Push have almost same cost.
Conclusion

→ Self-adjusting tree: building block for self-adjusting general graphs ("datacenters")

→ Rotor walk: a constant-competitive online algorithm, finds optimal tradeoff between routing and adjustment costs

→ Future work
  → Non-asymptotic analysis
  → Accounting also for load?

Thank you!
Websites

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

https://trace-collection.net/
Trace collection website
**Static DAN**

**Overview: Models**

**Static Optimality**

**Dynamic DAN**

**Robust DAN**

**Concurrent DANs**

---

**Further Reading**
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 Oliveire 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 Anais 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.
Golden Gate Zipper
Reconfigurable Optical Networks Will Move Supercomputer Data 100X Faster

Newly designed HPC network cards and software that reshapes topologies on-the-fly will be key to success

By Michelle Hampson