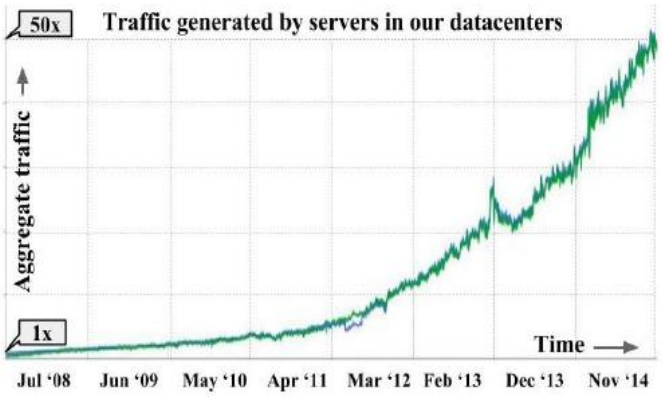
Improved Scalability of Demand-Aware Datacenter Topologies With Minimal Route Lengths and Congestion

Wenkai Dai, Maciej Pacut, Klaus-T. Foerster, Stefan Schmid (University of Vienna), Alexandre Labbe (ENSTA Paris)

Trends: Explosive Growth of Data-Centric Applications

Datacenters ("hyper-scale")

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

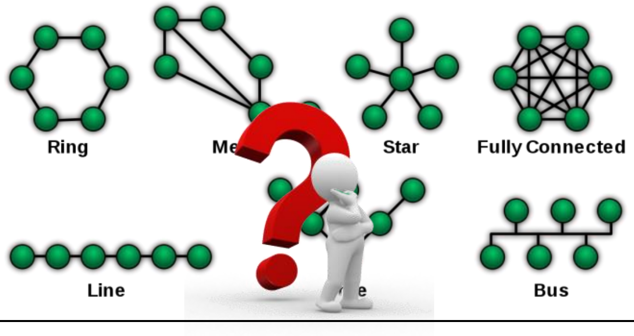


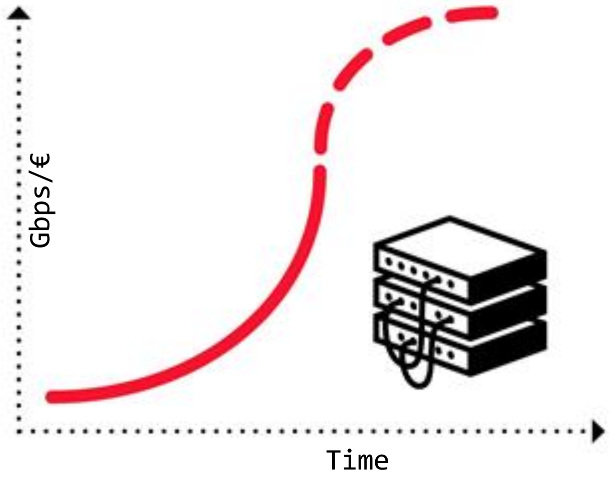
Aggregate Server Traffic in Google datacenter
Jupiter rising @ SIGCOMM 2015

Motivation: Network Infrastructure Design for Efficient Use

- “End of Moore‘s Law in networking” [1]
- Hence: more equipment, larger networks (**big investment**)
- Better infrastructure design for more efficient use

Opportunity for researchers

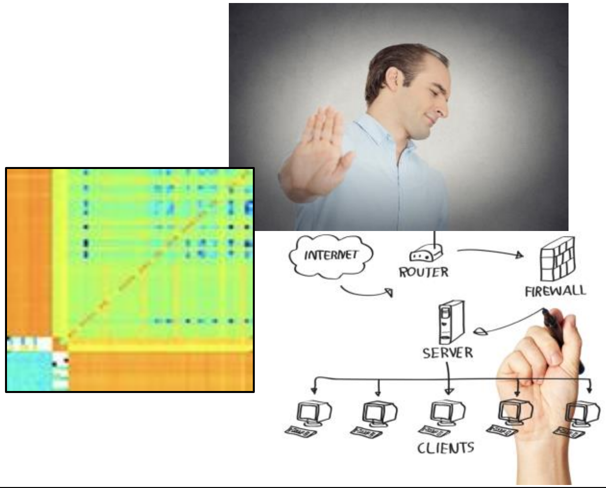


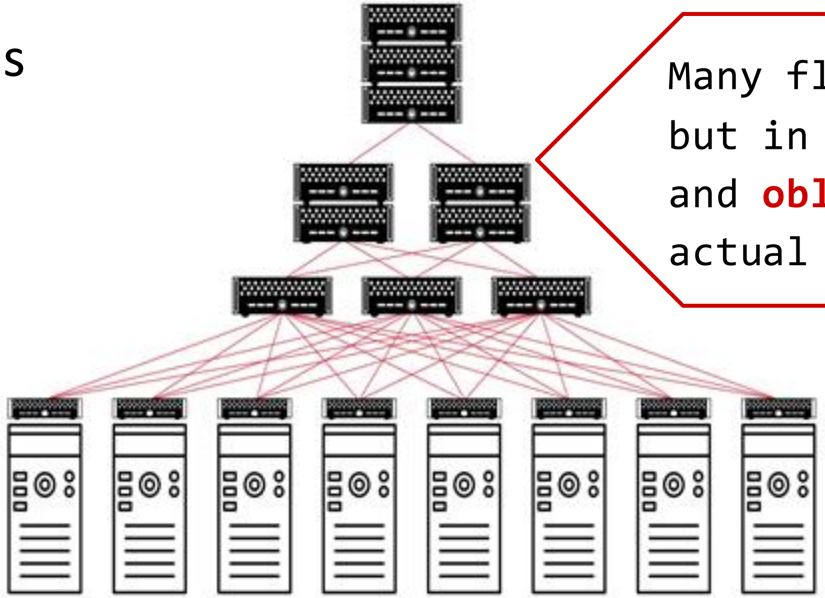


[1] Source: Microsoft, 2019

Demand Oblivious Network Design

- Traditional network topology designs are demands oblivious
  - Assuming demands are “all-to-all”
  - Optimize the worse case of assumed demands

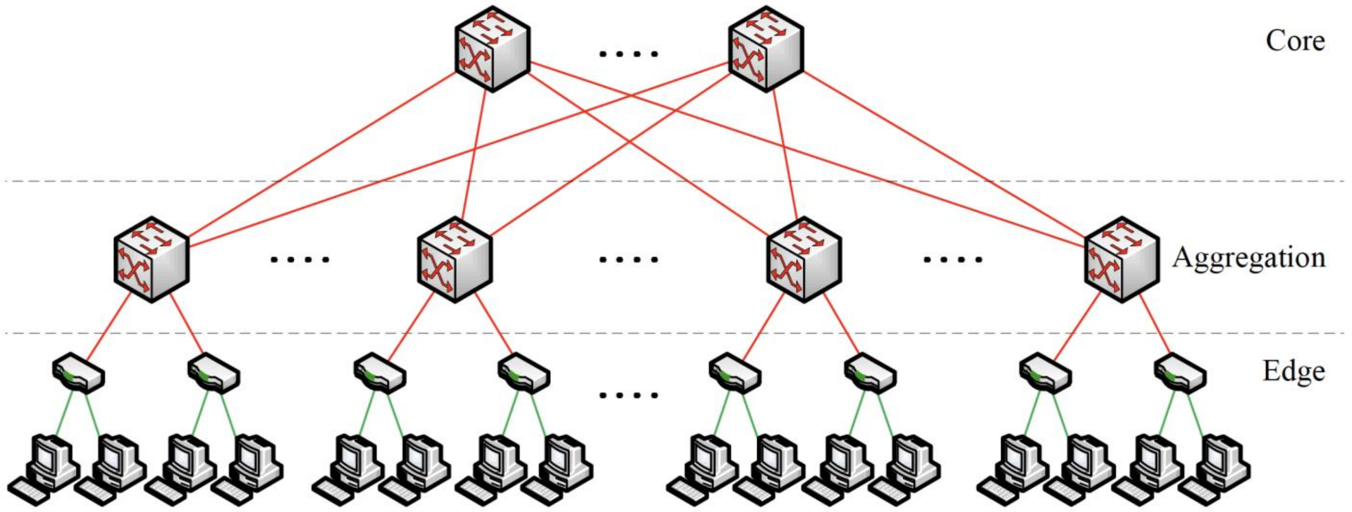




Many flavors,
but in common: fixed
and **oblivious** to
actual demand.

E.g., Fat-Tree (Clos) Topology for Data Centers

Fat-Tree is almost full bisection and good for all-to-all traffic

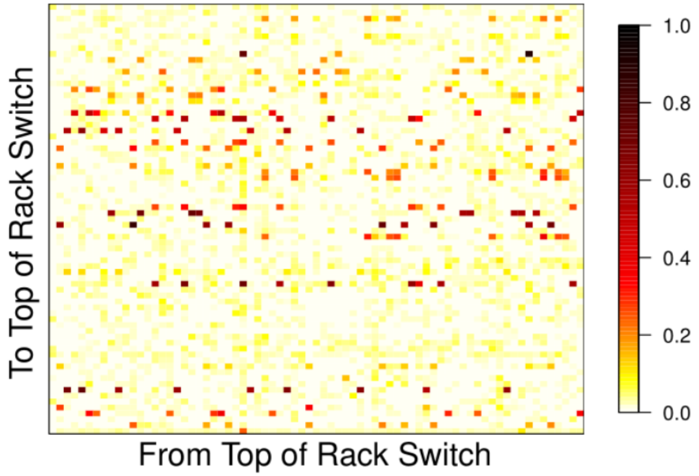


universität wien

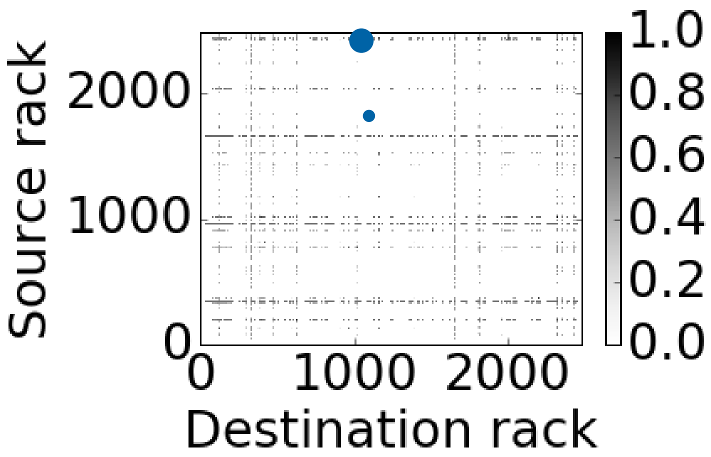
Real-World Traffic ≠ Uniform

- However, traffic, e.g., in DCN, is often ***not*** all-to-all **but sparse**

"Data reveal that 46-99% of the rack pairs exchange no traffic at all"



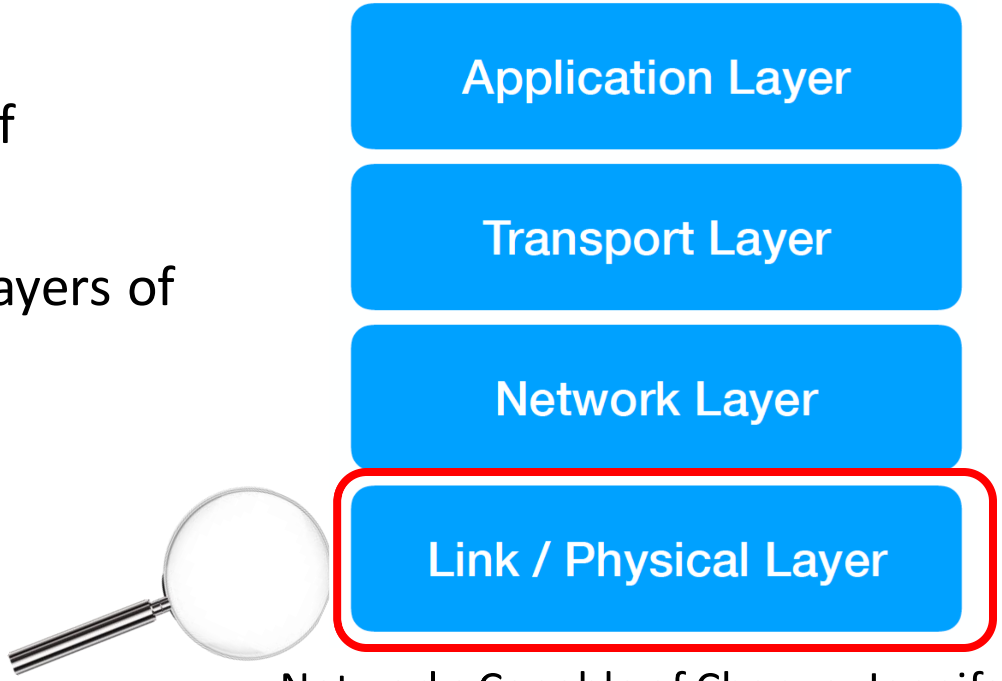
Traffic demands (normalized) between ToR switches. Halperin et al., SIGCOMM'11



Heatmap of rack to rack traffic. Color intensity is log-scale and normalized. Ghobadi et al., SIGCOMM'16

Demand-Aware Network Design

- Real communication are usually structured
- Demands-aware design assumes some features of demands are known
- Demands-aware design is broadly applied to all layers of networking
  - E.g., reconfigurable networks

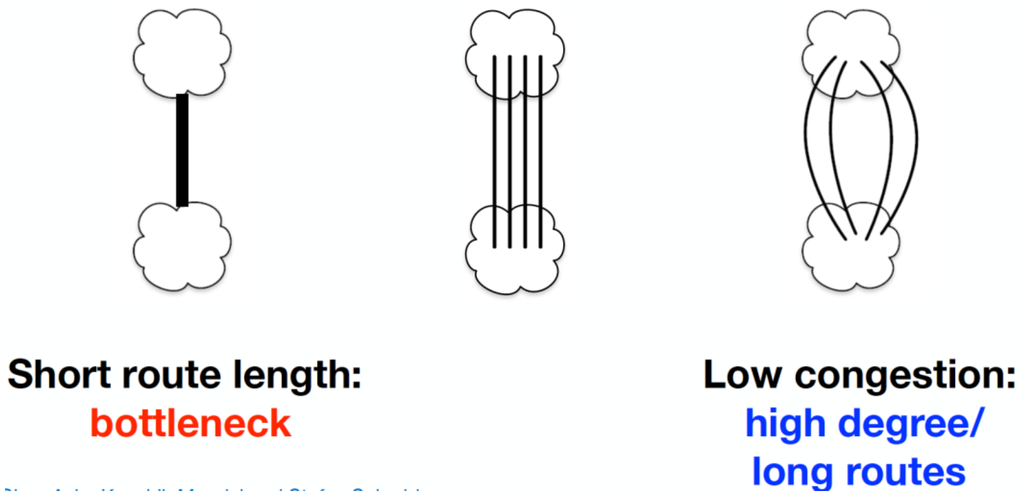


Networks Capable of Change. Jennifer Rexford. Infocom 2019 Keynote

General Goals and Challenges [Avin et al., DISC17, INFOCOM19]

- Goals of demands-aware, bounded-degree network design:
  - Satisfying a given degree bound $\Delta \geq 5$
  - Minimize both the average route length and the congestion for given demands
- Challenges:
  - A constant Δ indicates a sparse network: high diameter and low connectivity
  - Dilemma: Short route length vs Low Congestion

Short route length vs Low Congestion



Source: Avin *et al.*, *INFOCOM19*

universität wien

Input of Demands-Aware Bounded-Degree Network Design Problem

- Given:
  - A degree bound $\Delta \in \mathbb{N}_{\geq 5}$
  - Demands Matrix D:

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

Source: Avin *et al.*, *INFOCOM19*

Each entry $d_{i,j}$ means a communication intensity (frequency) from i to j

- Define demands (weighted) graph G_D: if $d_{i,j} > 0$, then an edge $\{i, j\}$ with the weight $w_{i,j} = d_{i,j} + d_{j,i}$

Definitions of Congestion and Route Length

- Given a demands matrix D

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

- A network N and a *routing scheme* $\Gamma(N)$ for serving D (unsplit flow)
- The weighted route length:

$$L(D,\Gamma(N)) = \sum_{(i,j)\in D} w_{i,j} \cdot dist_{\Gamma(N)}(i,j)$$

- The congestion:

$$C(D,\Gamma(N)) = max_{e\in\Gamma(N)} \sum_{e\in\Gamma_{i,j}} w_{i,j}$$

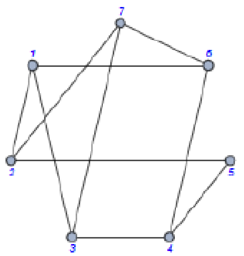
Objective of (c, d)-Bounded Network Design ((c, d)-BND Problem)

- ***Input:*** a demands D and $\Delta \in \mathbb{N}_{\geq 5}$

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

Source: Avin *et al., INFOCOM19*

- ***Output:*** a network $N \in N_\Delta$ and $\Gamma(N)$



Source: Avin *et al., INFOCOM19*

s.t.,

Optimal Route Lengths

- $L(D, \Gamma(N)) \leq c \cdot \mathrm{L}^*\big(D, \Gamma(N)\big) + c'$

- $\mathrm{C}(D, \Gamma(N)) \leq d \cdot \mathrm{C}^*\big(D, \Gamma(N)\big) + d'$

Optimal Congestion

universität wien

Previous Works

DISC 2017

Demand-Aware Network Designs of Bounded Degree*

Chen Avin[1], Kaushik Mondal[1], and Stefan Schmid[2]

1 Communication Systems Engineering Department
Ben Gurion University of the Negev, Israel
avin@cse.bgu.ac.il, mondal@post.bgu.ac.il
2 Department of Computer Science
Aalborg University, Denmark
schmiste@cs.aau.dk

Abstract

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

Motivated by these trends, this paper initiates the algorithmic study of demand-aware networks (DANs), and in particular the design of bounded-degree networks. The inputs to the network

INFOCOM 2019

Demand-Aware Network Design with Minimal Congestion and Route Lengths

Chen Avin, *Communication Systems Engineering Dept. Ben Gurion University of the Negev, Israel*
Kaushik Mondal, *Communication Systems Engineering Dept. Ben Gurion University of the Negev, Israel*
Stefan Schmid, *Faculty of Computer Science University of Vienna, Austria*

Abstract—Emerging communication technologies allow to reconfigure the physical network topology at runtime, enabling *demand-aware networks (DANs)*: networks whose topology is optimized toward the workload they serve. However, today, only little is known about the fundamental algorithmic problems underlying the design of such demand-aware networks. This paper presents the first bounded-degree, demand-aware network, *cl-DAN*, which minimizes *both* congestion and route lengths. The designed network is provably (asymptotically) optimal in each dimension individually: we show that there do not exist any bounded-degree networks providing shorter routes (independently of the load), nor do there exist networks providing lower loads (independently of the route lengths). The main building block of the designed cl-DAN networks are *ego-trees*: communication sources arrange their communication partners in an optimal tree, individually. While the union of these ego-trees forms the basic structure of cl-DANs, further techniques are presented to ensure bounded degrees (for scalability).

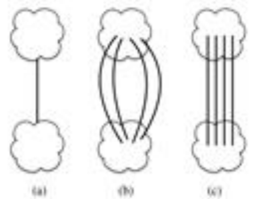


Fig. 1. Challenge of designing demand-aware networks: (a) Optimizing for route lengths only may result in bottlenecks and high loads. (b) Optimizing for congestion only, by distributing load across multiple paths, can result in long routes. (c) Ideally, we aim to design networks that minimize both congestion and route lengths, using a small number of links (constant degree).

I. Introduction

A. Motivation

Data center networks have become a critical infrastructure of our digital society. With the trend toward more data-intensive applications, data center network traffic is growing quickly [7], [31]. As much of this traffic is internal to the data center (e.g., ...

However, only little is known today about the *algorithmic* challenge of designing demand-aware networks which provide low congestion *and* short routes (in the number of hops), for ...

CCR 2019

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

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

This article is an editorial 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 the next frontier in an ongoing effort to render communication networks more flexible. While first empirical results indicate that these flexibilities can be exploited to reconfigure and optimize the network toward the workload it serves and, e.g., providing the same bandwidth at lower infrastructure cost, only little is known today about the fundamental algorithmic problems underlying the design of reconfigurable networks. This paper initiates the study of the theory of demand-aware, self-adjusting networks. Our main position is that self-adjusting networks should be seen through the lense of self-adjusting data...

Figure 1: Taxonomy of topology optimization

design of efficient datacenter networks has received much attention over the last years. The topologies underlying mod-...

TON 2016

SplayNet: Towards Locally Self-Adjusting Networks

Stefan Schmid*, Chen Avin*, Christian Scheideler, Michael Borokhovich, Bernhard Haeupler, Zvi Lotker

Abstract—This paper initiates the study of locally self-adjusting networks: networks whose topology adapts dynamically and in a decentralized manner, to the communication pattern σ. Our vision can be seen as a distributed generalization of the self-adjusting datastructures introduced by Sleator and Tarjan [22]: In contrast to their splay trees which dynamically optimize the lookup costs from a *single node* (namely the tree root), we seek to minimize the routing cost between arbitrary communication *pairs* in the network.

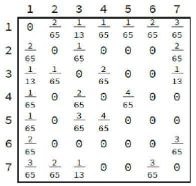
As a first step, we study distributed binary search trees (BSTs), which are attractive for their support of greedy routing. We introduce a simple model which captures the fundamental tradeoff between the benefits and costs of self-adjusting networks. We present the SplayNet algorithm and formally analyze its performance, and prove its optimality in specific case studies. We also introduce lower bound techniques based on interval cuts and

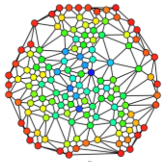
toward static metrics, such as the diameter or the length of the longest route: the self-adjusting paradigm has not spilled over to distributed networks yet.

We, in this paper, initiate the study of a distributed generalization of self-optimizing datastructures. This is a non-trivial generalization of the classic splay tree concept: While in classic BSTs, a *lookup request* always originates from the same node, the tree root, distributed datastructures and networks such as skip graphs [2], [13] have to support *routing requests* between arbitrary pairs (or *peers*) of communicating nodes; in other words, both the source as well as the destination of the requests become variable. Figure 1 illustrates the difference between classic and distributed binary search trees.

In this paper we ask: Can we reap similar benefits from self-...

Our Contributions in Complexity



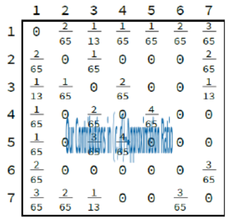


Demand Graph G_D	Optimize Route Lengths	Optimize Congestion
General Graphs	NP-hard	NP-hard
Trees	NP-hard	NP-hard

Former Results

Our Results

Our Contributions in (c, d)-Approximation Ratio

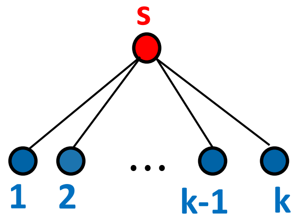


Tree Demands G_D	[Avin et al, INFOCOM19]	[This Paper]
(c,d)-Approximation c: route lengths d: congestion	$\left(log^2(\Delta_{max}+1), {}^4/_3\right)$ Δ_{max}: the max degree of N	$(2, 6)$

Star Example: Optimal Network Design N by EgoTree

- Demands D induced by a star

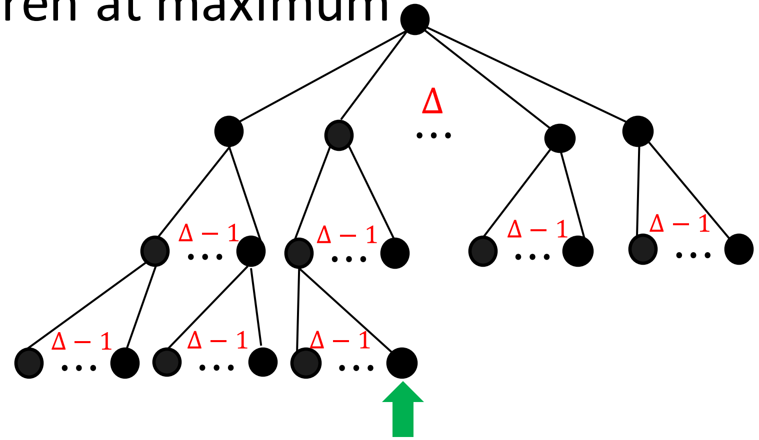
	1	2	3	4	5	6	7
1	0	$\frac{2}{65}$	$\frac{1}{13}$	$\frac{1}{65}$	$\frac{1}{65}$	$\frac{2}{65}$	$\frac{3}{65}$



Weights $W = \{w_1, w_2, \dots, w_k\}$

Construct N

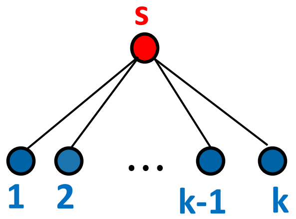
- Construct a full tree of $k+1$ nodes
- Root has Δ children, other nodes has $\Delta - 1$ children at maximum



Star Example: Optimal Network Design N by EgoTree

- Demands D induced by a star (2-level tree)

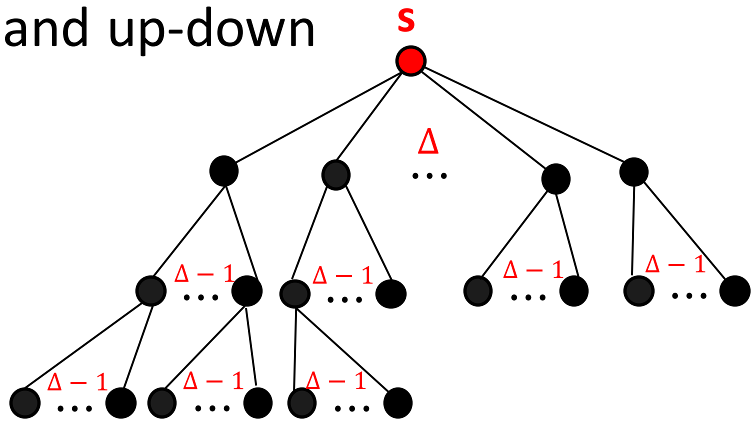
	1	2	3	4	5	6	7
1	0	$\frac{2}{65}$	$\frac{1}{13}$	$\frac{1}{65}$	$\frac{1}{65}$	$\frac{2}{65}$	$\frac{3}{65}$



Weights $W = \{w_1, w_2, \dots, w_k\}$

Construct N

- Sort W non-increasingly to $W' = \{w'_1, w'_2, \dots, w'_k\}$
- Set the root of N as the original root s
- Insert nodes according to W' sequentially by left-right and up-down

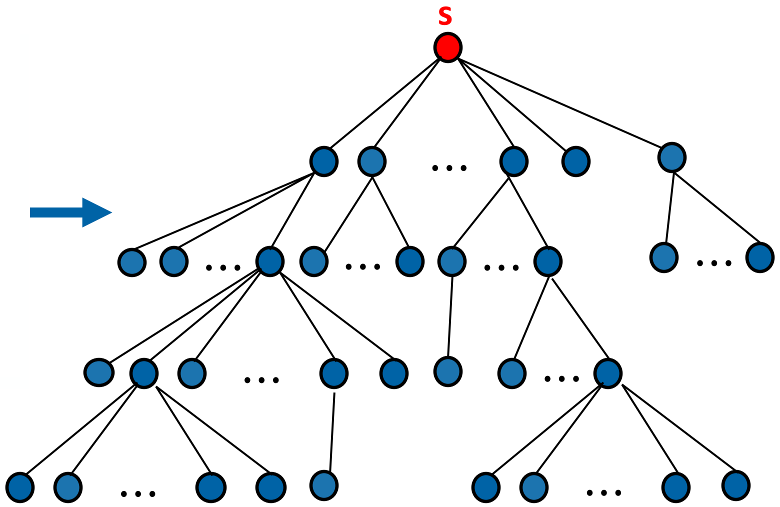


Easy to know it is optimal for route lengths

When Demands D Induced By General Trees

- Demands D induced by general trees

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



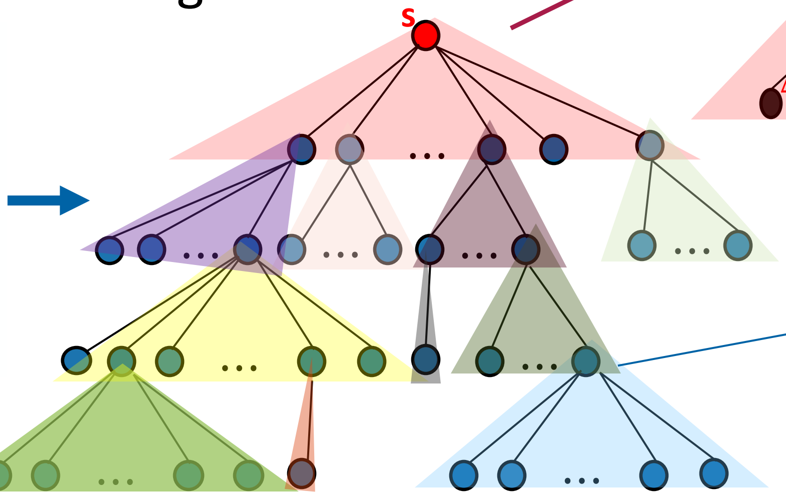
Weights $W = \{w_1, w_2, \ldots, w_k\}$

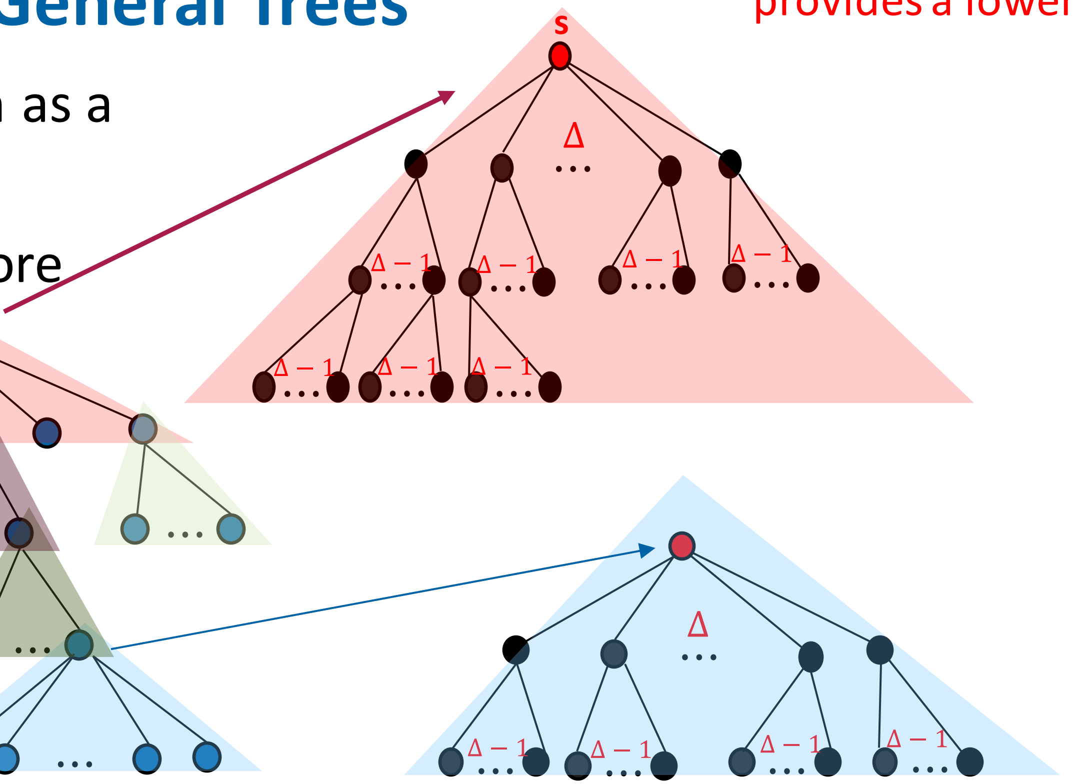
When Demands D Induced By General Trees

- Considering each node and its children as a star in demands tree G_D
- For each star to build a EgoTree as before

- Problem: these EgoTrees cannot be merged together to satisfy Δ
- But the sum of their route lengths provides a lower bound

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





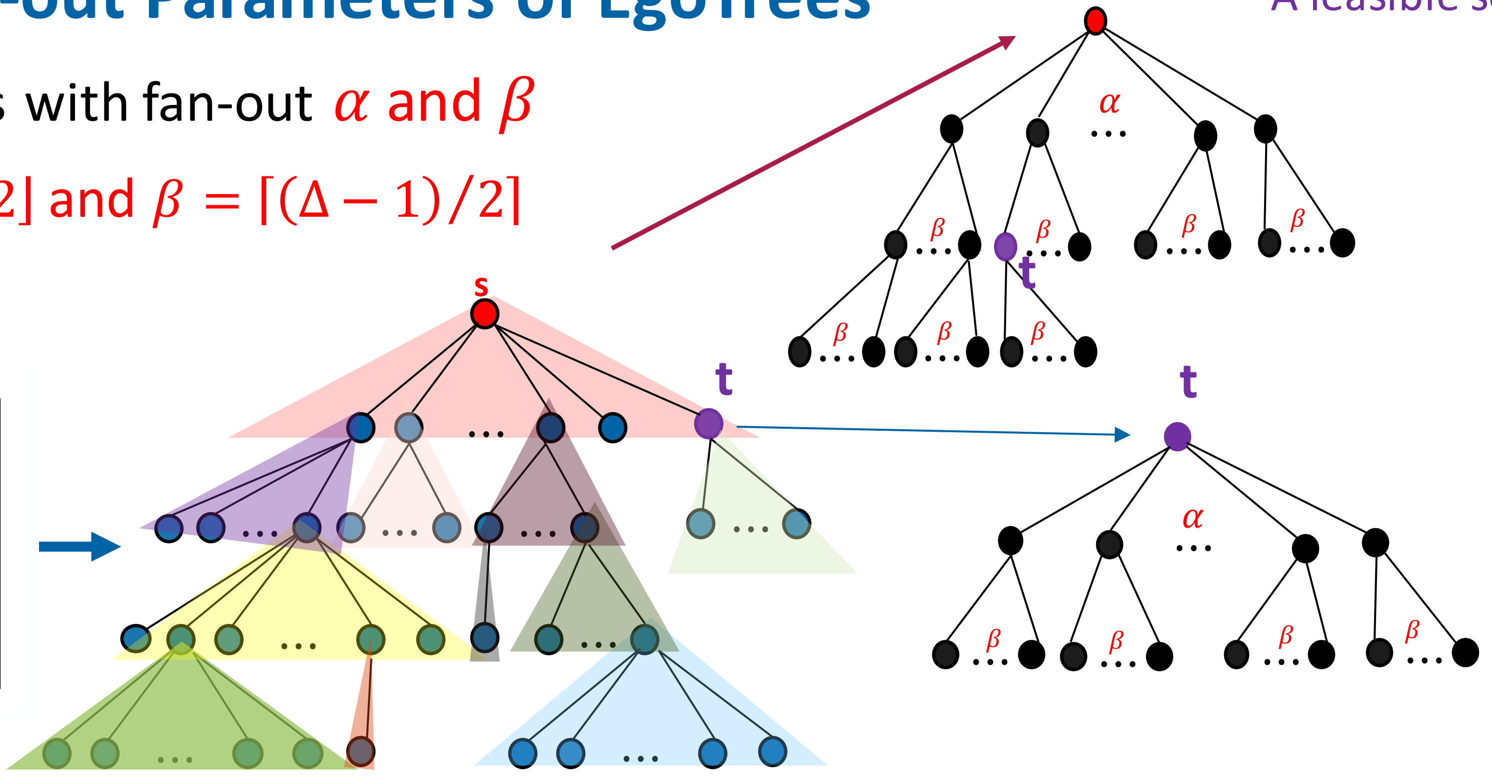
Weights $W = \{w_1, w_2, \ldots, w_k\}$

Changes Fan-out Parameters of EgoTrees

- build EgoTrees with fan-out α and β
- $\alpha = \lfloor(\Delta - 1)/2\rfloor$ and $\beta = \lceil(\Delta - 1)/2\rceil$
- $\alpha + \beta \leq \Delta$

- For each node t merging two EgoTrees at t
- Clearly, the degree of t: $\alpha + \beta \leq \Delta$
- A feasible solution is obtained

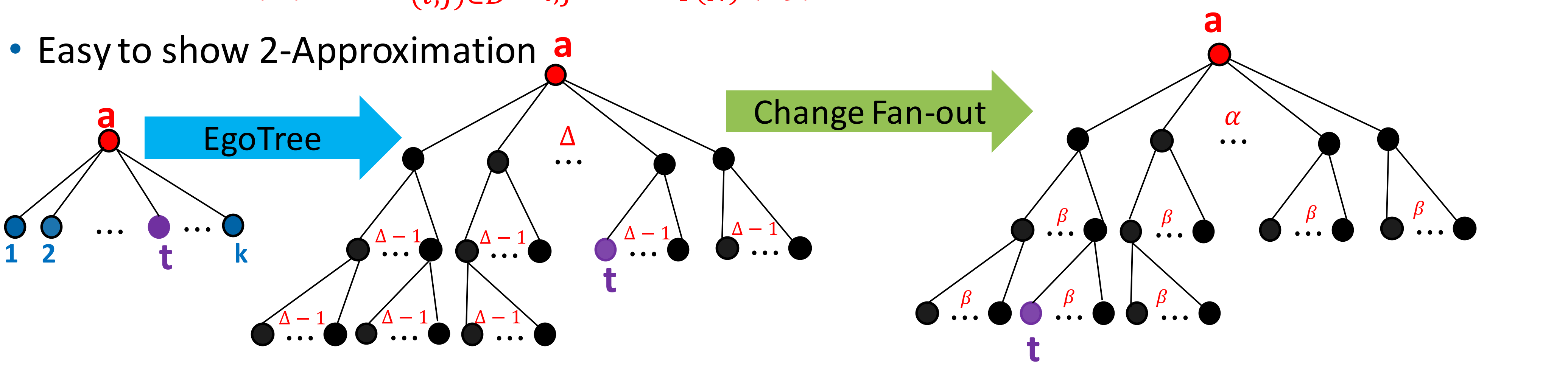
	1	2	3	4	5	6	7
1	0	$\frac{2}{65}$	$\frac{1}{13}$	$\frac{1}{65}$	$\frac{1}{65}$	$\frac{2}{65}$	$\frac{3}{65}$
2	$\frac{2}{65}$	0	$\frac{1}{65}$	0	0	0	$\frac{2}{65}$
3	$\frac{1}{13}$	$\frac{1}{65}$	0	$\frac{2}{65}$	0	0	$\frac{1}{13}$
4	$\frac{1}{65}$	0	$\frac{2}{65}$	0	$\frac{4}{65}$	0	0
5	$\frac{1}{65}$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



Weights $W = \{w_1, w_2, \dots, w_k\}$

2-Approximation of Optimal Route Lengths

- Two EgoTrees built for a star at the center a for different fan-out
- For a node **t,** its distance to a is at most doubled after decreasing fan-outs
- Recall $L(D, \Gamma(N)) = \sum_{(i,j) \in D} w_{i,j} \cdot dist_{\Gamma(N)}(i,j)$
- Easy to show 2-Approximation

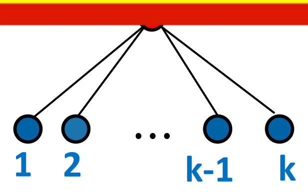


$\alpha = \lfloor(\Delta - 1)/2\rfloor$ and $\beta = \lceil(\Delta - 1)/2\rceil$

Strong Congestion Caused By EgoTrees

- Demands [...]uced by a star

	1	2
1	0	



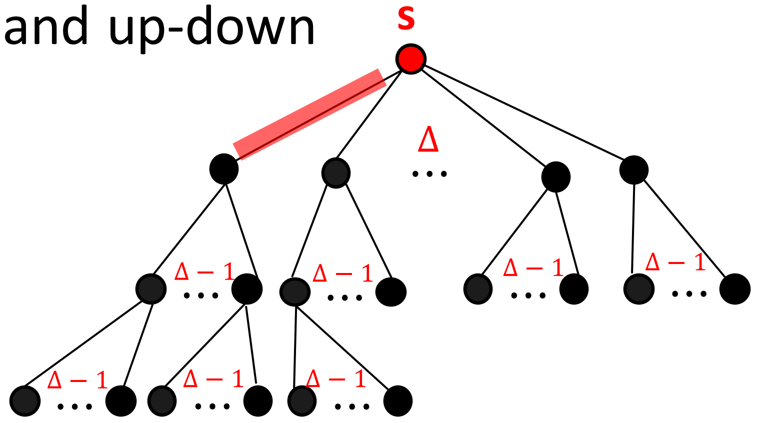
Weights $W = \{w_1, w_2, \dots, w_k\}$

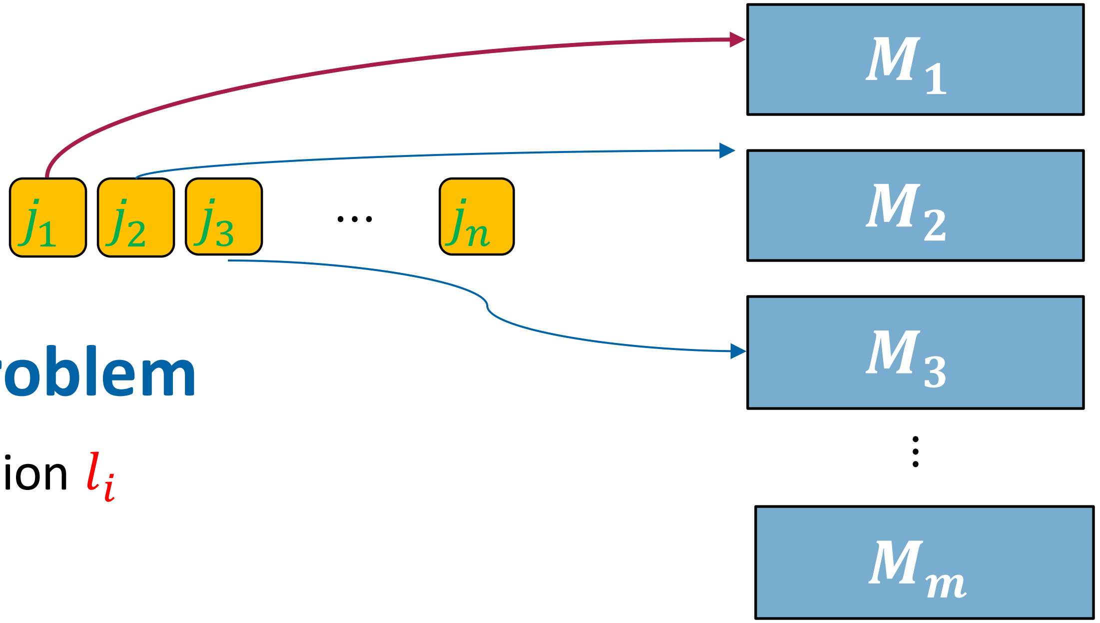
Construct N

- Sort W non-increasingly to $W' = \{w'_1, w'_2, \dots, w'_k\}$

Strong congestion on some links incident to the root

[...] by left-right and up-down





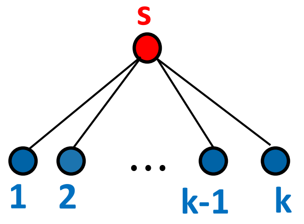
Recall Scheduling On Identical Machines Problem

- A set of jobs $J = \{j_1, j_2, \dots j_n\}$, where each $j_i \in J$ has a duration l_i
- A set of machines $M = \{M_1, M_2, \dots M_m\}$
- Assign each job to a machine to minimize the make-span
- First, sort J in the order of non-increasing duration, denoted by J'
- Idea: assign J' sequentially from M_1 to M_m by a round robin (2-approximation)

Idea of Round-Robin Tree

- Demands D induced by a star

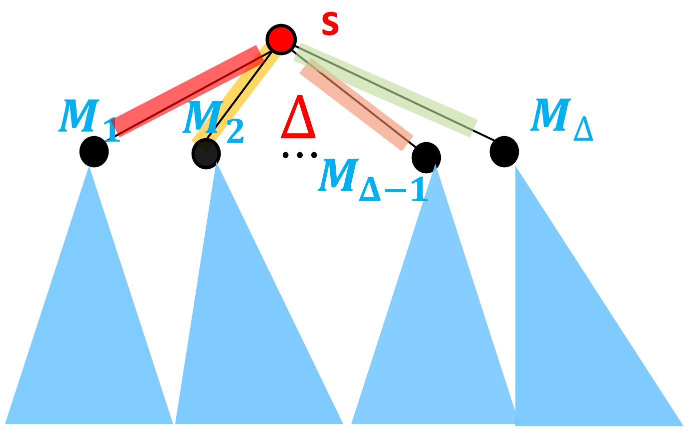
	1	2	3	4	5	6	7
1	0	$\frac{2}{65}$	$\frac{1}{13}$	$\frac{1}{65}$	$\frac{1}{65}$	$\frac{2}{65}$	$\frac{3}{65}$



Weights $W = \{w_1, w_2, \ldots, w_k\}$

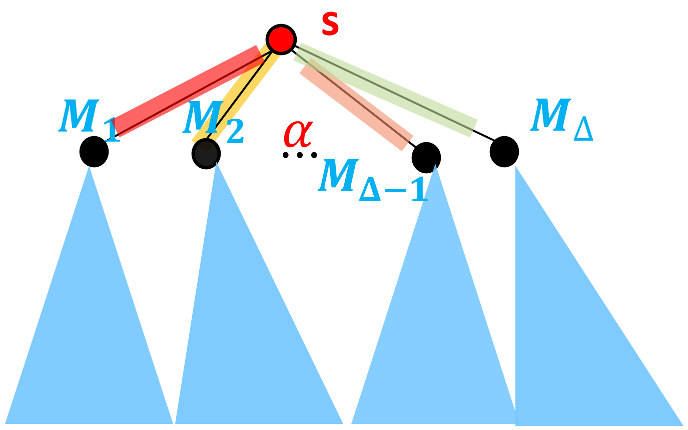
Construct N

- Sort W non-increasingly to $W' = \{w'_1, w'_2, \ldots, w'_k\}$
- Set the root of N as the original s
- Assign W' into Δ machines, s.t., nodes always stay in the subtree of its assigned machine
- The load on these highlighted links are at most double of the optimal congestion.



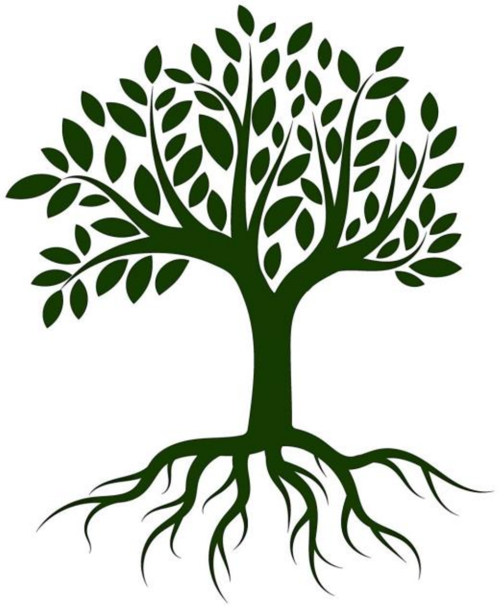
The $(2, 6)$-Approximation By Round-Robin Trees

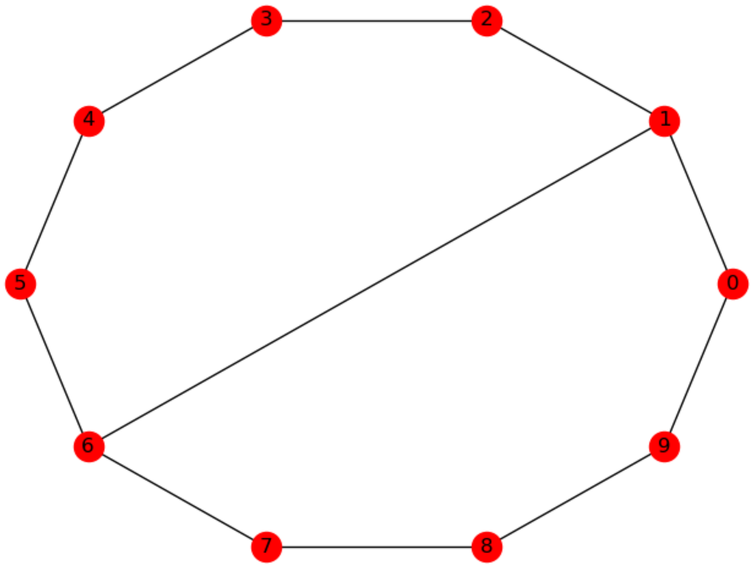
- First change Δ machines to $\alpha = \lfloor(\Delta - 1)/2\rfloor$, for each subtree, place nodes as EgoTrees
- For congestion: as $\Delta/\alpha \leq 3$, then 6-Approximation achieved by Round-Robin Trees
- As $\beta \geq \alpha$, it is easy to show 2-Approximation for route lengths



Demands Induced by Trees to Sparse Graphs

- Sparse Graphs: the average degree Δ_{avg} is a constant and some node might have non-constant degree

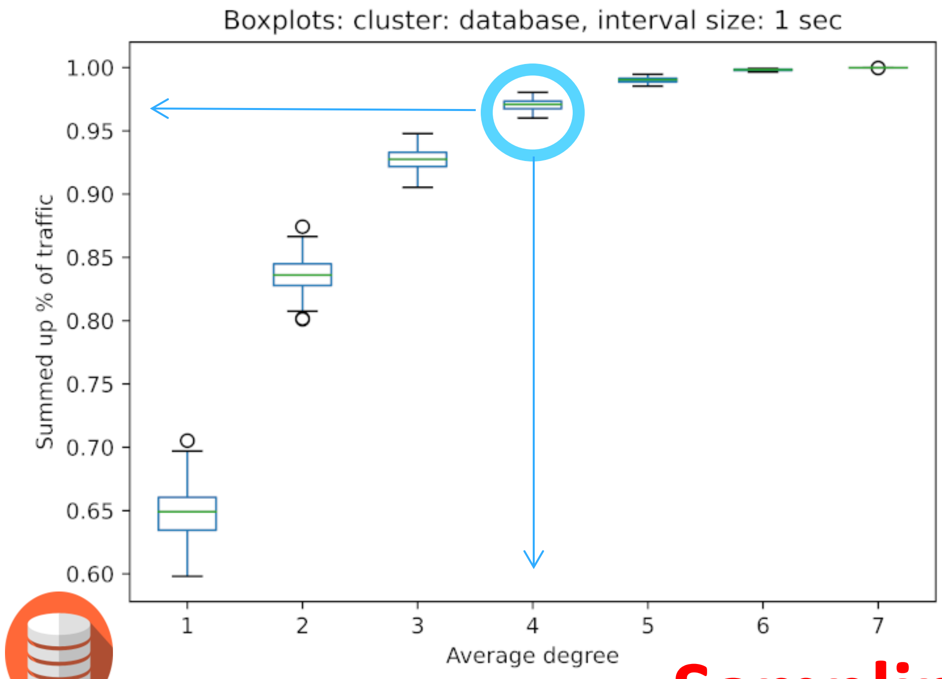


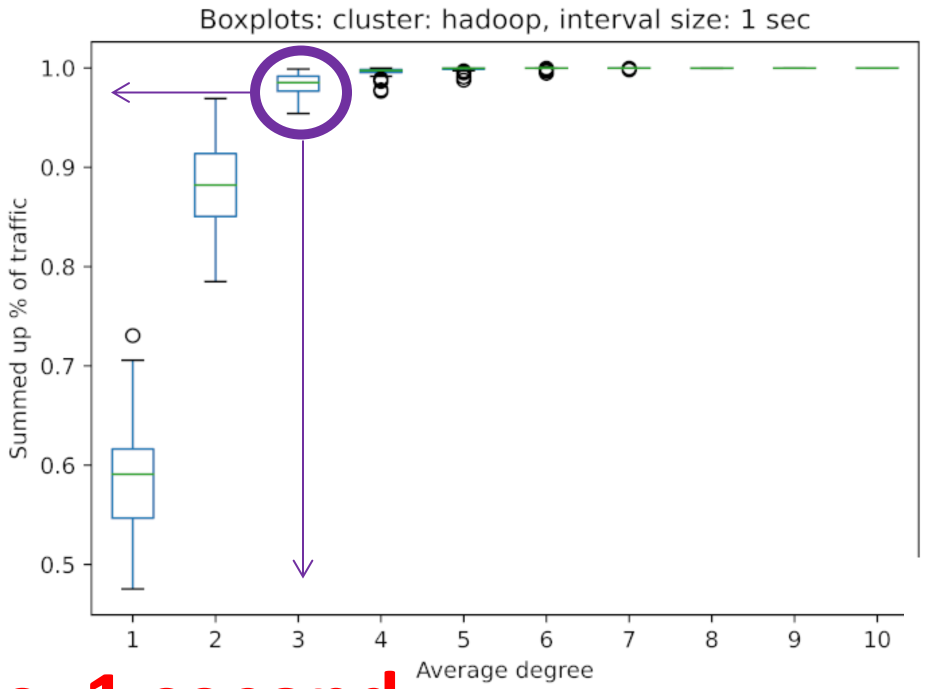


facebook

Facebook Traffic Graph Is Sparse

$\geq$ 95% Traffic Shows $\Delta_{avg} \leq 4$

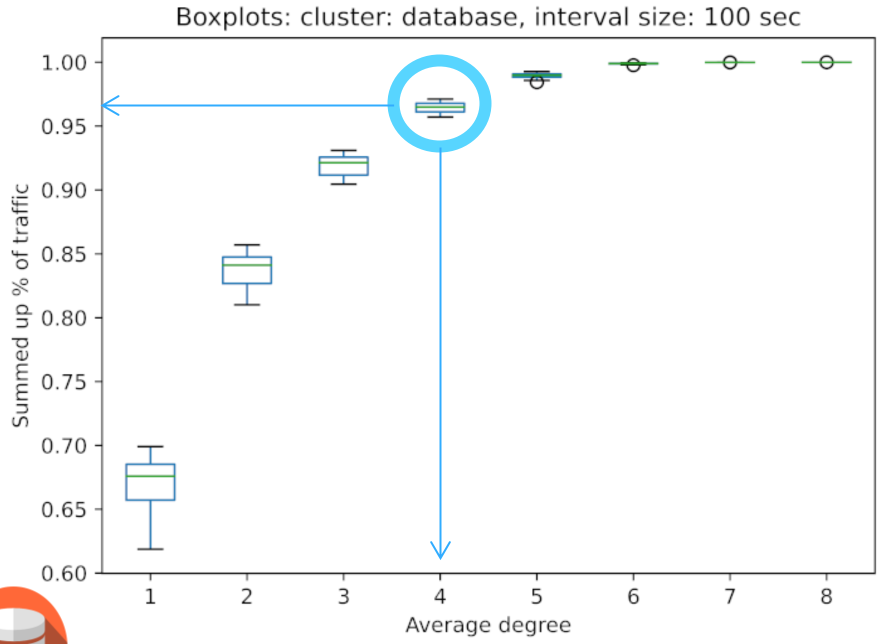


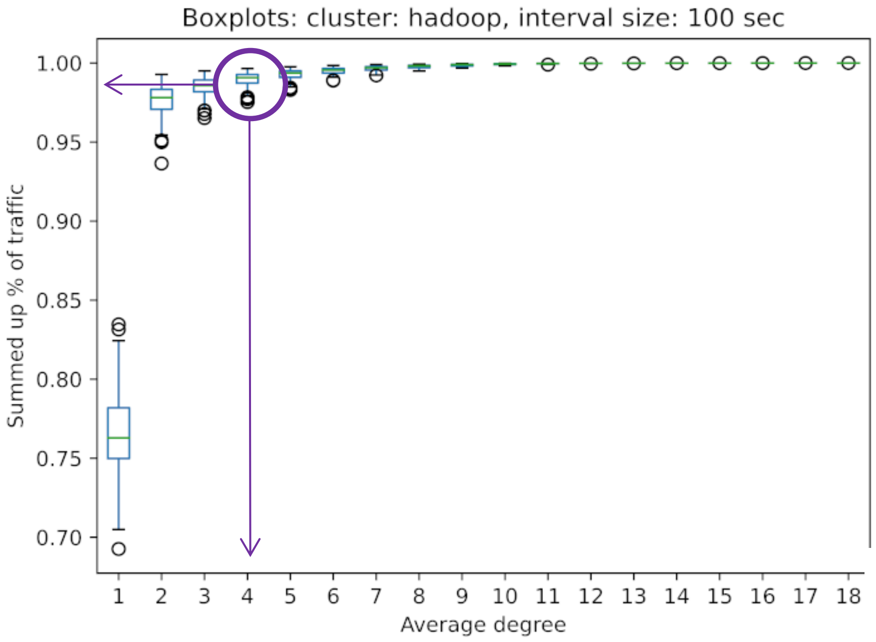


Sampling time: 1 second

Facebook Traffic Graph Is Sparse

$\geq$ 95% Traffic Shows $\Delta_{avg} \leq 4$





Sampling time: 100 second

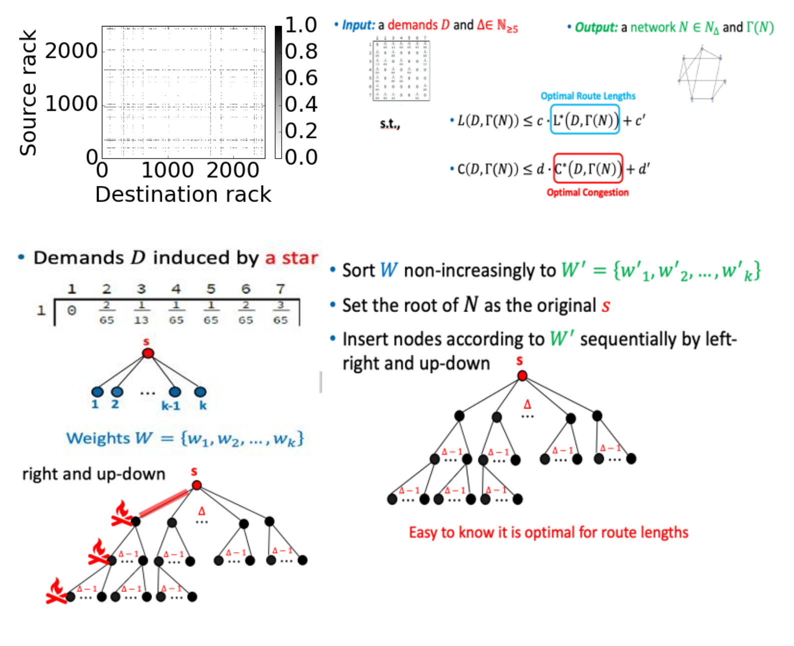
Network Design For Sparse Demands

Sparse G_D (Δ_{avg})	[Avin et al, INFOCOM19]	[This Paper]
For near-optimal Route lengths (bounds by Avin et al, INFOCOM19)	$\Delta_{max} \leq 12 \cdot \Delta_{avg}$ Δ_{max}: the max degree of N	$\Delta_{max} \leq 3 \cdot \Delta_{avg} + \mathbf{8}$ Δ_{max}: the max degree of N

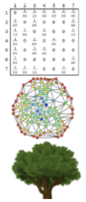
Thank you! ☺

Improved Scalability of Demand-Aware Datacenter Topologies With Minimal Route Lengths and Congestion

Wenkai Dai, Maciek Pancut, Klaus-T. Foerster, Stefan Schmid (University of Vienna), Alexandre Labbe (ENSTA Paris)



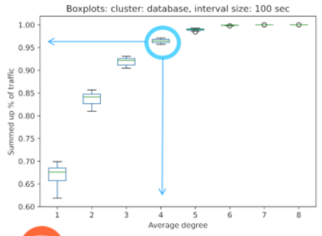
Summary of Our Contributions

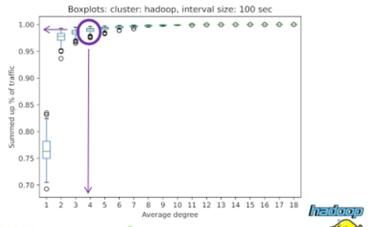


Demand Graph G_D	Optimize Route Lengths	Optimize Congestion
General Graphs	NP-hard	NP-hard
Trees	NP-hard	NP-hard

Tree Demands G_D	[Avin et al, INFOCOM19]	[This Paper]
(c,d)-Approximation c: route lengths d: congestion	$(log^2(\Delta_{max}+1), {}^4/_3)$ Δ_{max}: max degree of N	$(2,6)$

Sparse G_D (Δ_{avg})	[Avin et al, INFOCOM19]	[This Paper]
For near-optimal Route lengths (bounds by Avin et al, INFOCOM19)	$\Delta_{max} \geq 12 \cdot \Delta_{avg}$ Δ_{max}: the max degree of N	$\Delta_{max} \geq 3 \cdot \Delta_{avg} + 8$ Δ_{max}: the max degree of N

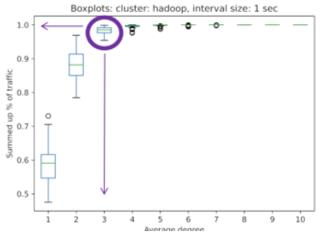


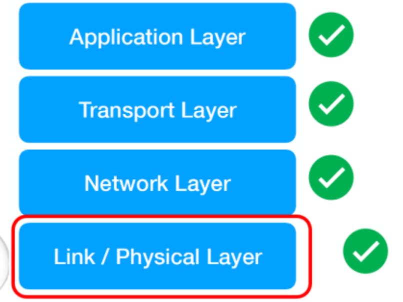


facebook

Sampling time: 100 second

$\geq 95\%$ Traffic Shows $\Delta_{avg} \leq 4$





Networks Capable of Change. Jennifer Rexford. Infocom 2019 Keynote