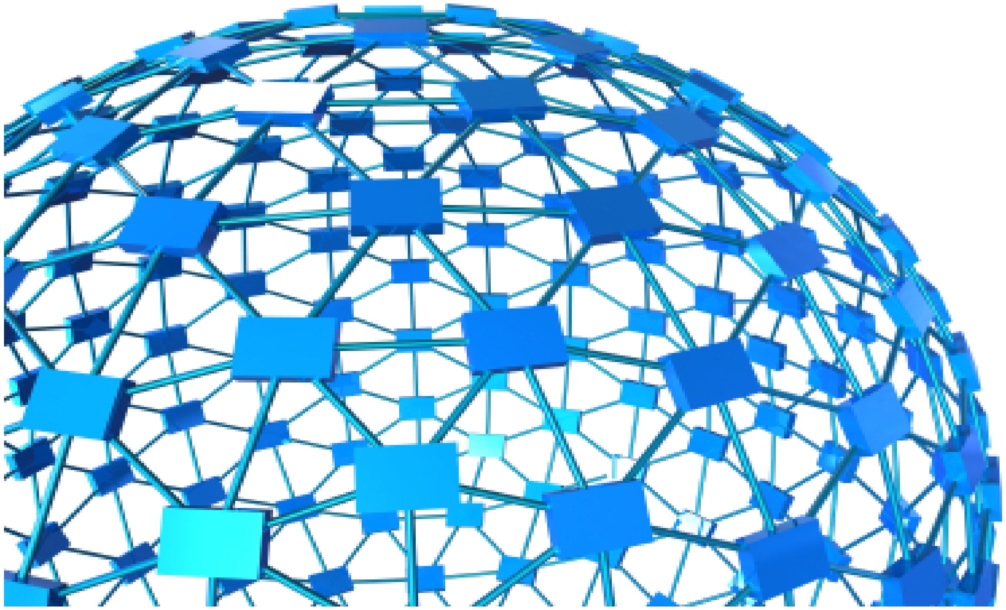
Robust Optimization of Networks



Joint works with:
N. Olver, N. Goyal
and
C. Chekuri, G. Oriolo, M.G. Scutella
Peter Winzer

Overview

I What is Robust Optimization?

II What is Network Design?

III Uncertainty and Network Design

IV Routing Models with Uncertain or Changing Traffic

V What we know about Robust Network Design

VI The VPN Conjecture

VII IP over Optical Routing

I. The Robustness Paradigm

Consider a MINIMUM COST "SYSTEM DESIGN" PROBLEM where for a given instance, not all of the parameters are known ahead of time.

Assumption: This uncertainty is known and bounded. Each instance comes with a universe of possible inputs $\mathcal{U}$.

I. The Robustness Paradigm

Consider a *minimum cost system design problem* where for a given instance, not all of the parameters are known ahead of time.

Assumption: This uncertainty is known and bounded. Each instance comes with a universe of possible realizations $\mathcal{U}$.

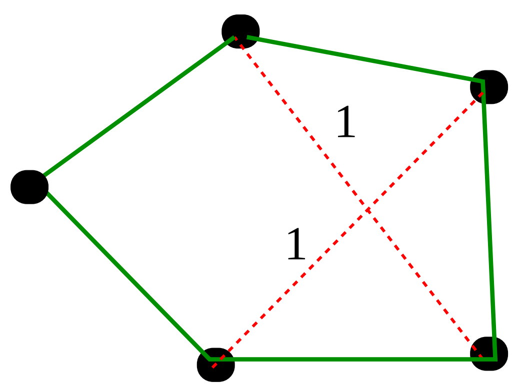
A Robust Solution is one that supports the feasible implementation of the system for any possible realization from $\mathcal{U}$.

Goal: Find a cheapest robust solution.

II. Network Design

A standard minimum cost network design problem consists of:

- a graph G with a per-unit cost $c(e)$ for each link $e \in E(G)$
- a demand matrix D_{ij} representing the *demand* between nodes i, j



Network Design

A standard minimum cost network design problem consists of:

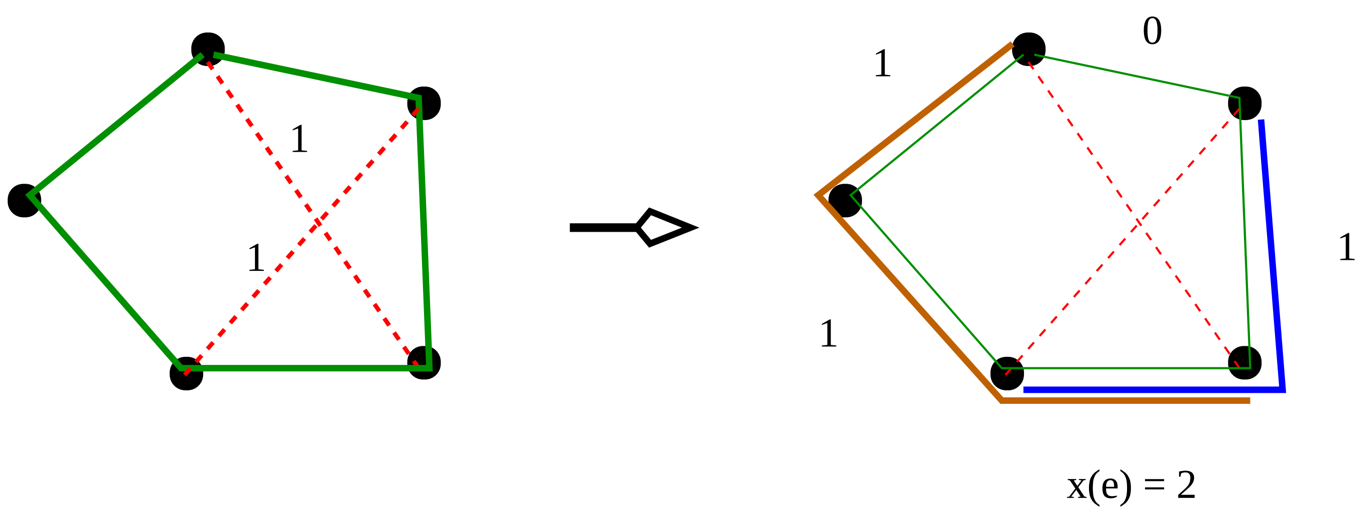
- a graph G with a per-unit cost $c(e)$ for each link $e \in E(G)$
- a demand matrix D_{ij} representing the *demand* between nodes i, j

Output: Minimum cost *edge capacity reservation* $x(e)$ *that supports the simultaenous routing of all demands* D_{ij}.

assumption: *our networks and flows are generally undirected.*

Network Design

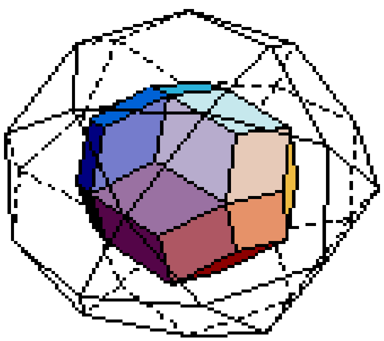
Typically there are many side constraints: routing constraints, buy-at-bulk capacity constraints, node costs, resilience requirements etc.



III. Uncertain Demand in Network Design

Dont know a single demand matrix - we have a collection of possible "valid" demand matrices.

Generally take the universe $\mathcal{U}$ to be a convex region of demands (D_{ij}).



(Vanilla) Robust Network Design

UNIVERSE: *The collection $\mathcal{U}$ of "valid" demand matrices.*

ROBUSTNESS: *build enough network capacity to "fulfil" each demand matrix*

$$(D_{ij}) \in \mathcal{U}.$$

Goal: Find a robust edge capacity vector $x(e)$ that minimizes $\sum_e c(e)x(e)$.

Some "Known" Universes

<u>*Forecasts on Future Traffic*</u>

1. *Given traffic "forecasts"* D^i *for* $i = 1, 2, \ldots, s$*:*

$$\mathcal{U} = conv\{D^1, D^2, \ldots, D^s\}.$$

Some "Known" Universes

Forecasts on Future Traffic

1. *Given target matrix D and other "forecasts" D^i for $i = 1, 2, \ldots, s$:*

$$\mathcal{U} = conv\{D, D^1, D^2, \ldots, D^s\}.$$

Traffic is Known Approximately

2. *(Applegate-Cohen 2006) Given a target matrix $\tilde{D}$ and threshold parameter $\alpha \geq 1$,*

$$\mathcal{U} = \{(D_{ij}) : s.t.\ D_{ij} \in [\,\frac{\tilde{D}_{ij}}{\alpha}\,,\ \alpha \cdot \tilde{D}_{ij}] \quad \forall i, j\}.$$

Some “Known” Universes

“Unit Ball” Demand Polyhedra

3. *Consider a unit ball around $\tilde{D}_{ij}$ defined by some norm L.*

$$P(\tilde{D}, L) = \{D : ||D - \tilde{D}||_L \leq 1\}.$$

- *l_p norms give rise to classes of concave cost flow problems.*
- *The $L2$-norm and ellipsoidal balls were considered by Ben Tal and Nemirovski (also Belotti and Pinar) to study associated stochastic optimization problems.*

Design cheapest network such that the probability a link's capacity is exceeded is at most $p(e)$.

Some Known Universes

Bounds on Injected Traffic

4. *The Hose Model (Fingerhut et al. 1997, Duffield et al. 1999)*

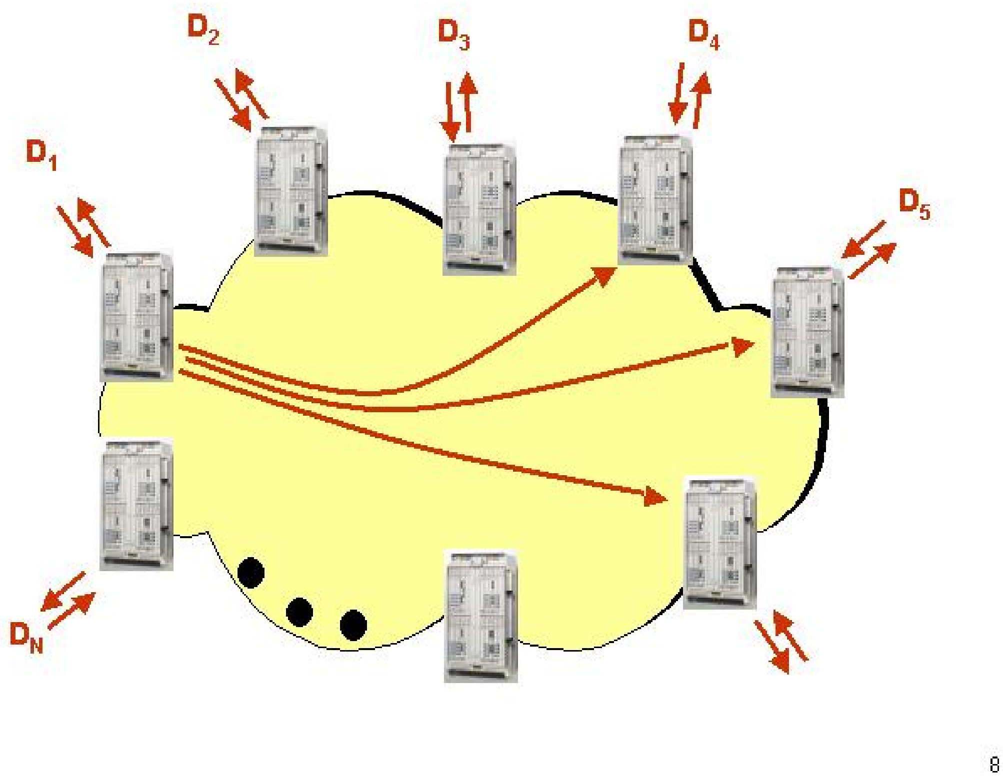
Given marginal demands D_i *for each node* i

$$\mathcal{U} = \{(D_{ij}) : \sum_j D_{ij} \leq D_i, \quad \sum_i D_{ij} \leq D_j \quad \forall i, j\}$$

.

Taking D_i*'s as* 1 *captures the basic idea.*

Virtual Private Network (VPN) Design



Given marginal demands D_i *for each node* i

$\mathcal{U} = \{(D_{ij}) : \sum_j D_{ij} \leq D_i$ *and* $\sum_j D_{ji} \leq D_i \quad \forall i\}.$

Have you seen the Hose Model before?

- **Permutation Routing Model:** *Each terminal has one packet to send, and one packet to receive.*
  Valiant *(Randomized Load Balancing method for routing with* $O(\log n)$ *congestion in hypercube and other networks)*, Borodin-Hopcroft *showed that randomized methods are needed to obtain such bounds.*

Have you seen the Hose Model before?

- **Permutation Routing Model:** *Each terminal has one packet to send, and one packet to receive.*

  Valiant *(Randomized Load Balancing method for routing with* $O(\log n)$ *congestion in hypercube and other networks)*, Borodin-Hopcroft *showed that randomized methods are needed to obtain such bounds.*

- **Uniform Multiflows (UMCF):** $D_{ij} = \frac{1}{k}$ *for all* $i, j \in X$ *(set of "terminals").*

  UMCF lies in the Hose polytope (marginals are 1 *for terminals)*

  Leighton-Rao *(1988) showed that UMCF can be routed in any "*X*-expander" with congestion* $O(\log n)$.

  Observation*: If you can route UMCF, then you can route any hose matrix with congestion at most 2.*

Have you seen the Hose Model before?

- **Permutation Routing Model:** *Each terminal has one packet to send, and one packet to receive.*

  Valiant *(Randomized Load Balancing method for routing with $O(\log n)$ congestion in hypercube and other networks),* Borodin-Hopcroft *showed that randomized methods are needed to obtain such bounds.*

- **Uniform Multiflows (UMCF):** $D_{ij} = \frac{1}{k}$ *for all* $i, j \in X$ *(set of "terminals").*

  UMCF lies in the Hose polytope (marginals are 1 *for terminals)*

  Leighton-Rao *(1988) showed that UMCF can be routed in any "X-expander" with congestion* $O(\log n)$.

  Observation*: If you can route UMCF, then you can route any hose matrix with congestion at most 2.*

- **Traffic Estimation in Emerging Markets** *(*Kruithof *1937,* Krupp *1979)*

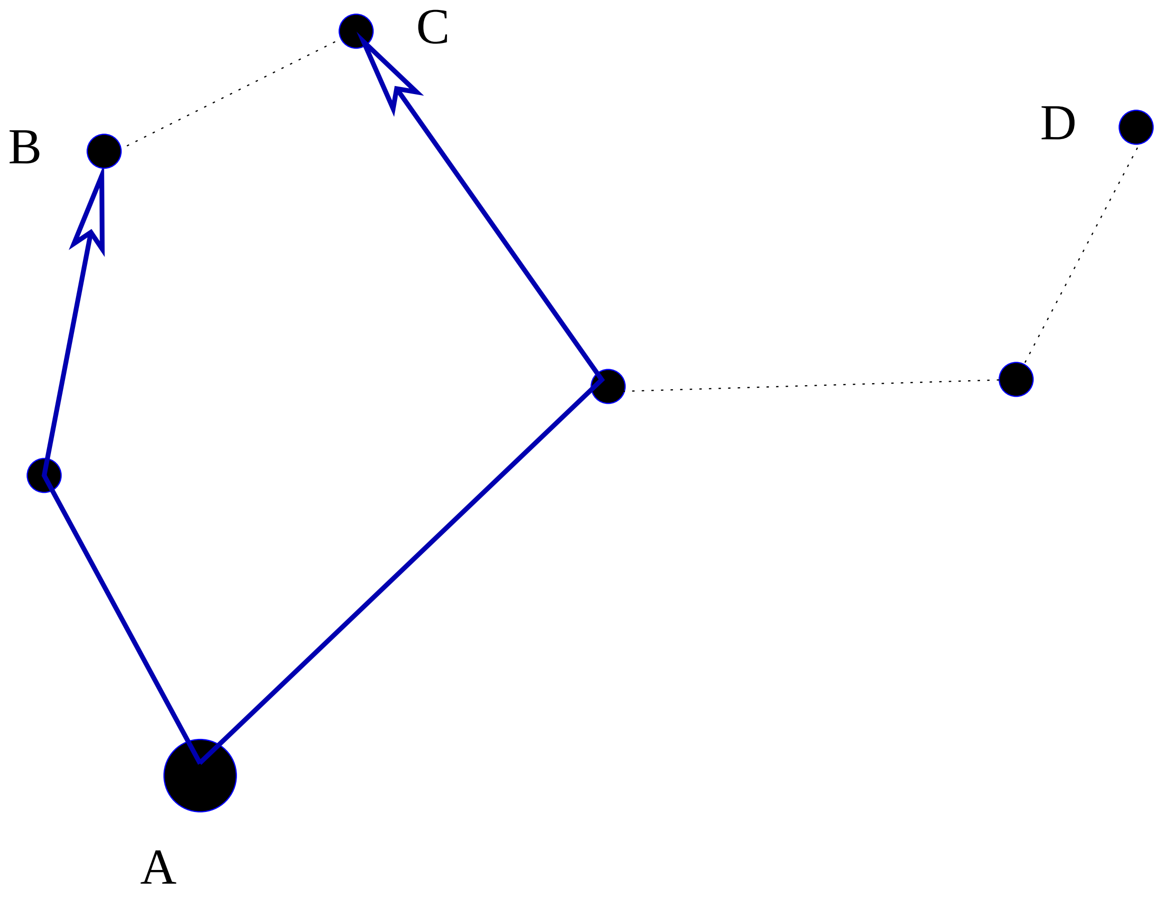
IV. Routing Models for Uncertain Demands

ROBUST: enough network capacity to "fulfil" each demand matrix in the polytope $\mathcal{U}$.

Need to say more about how we "fulfil" demand.

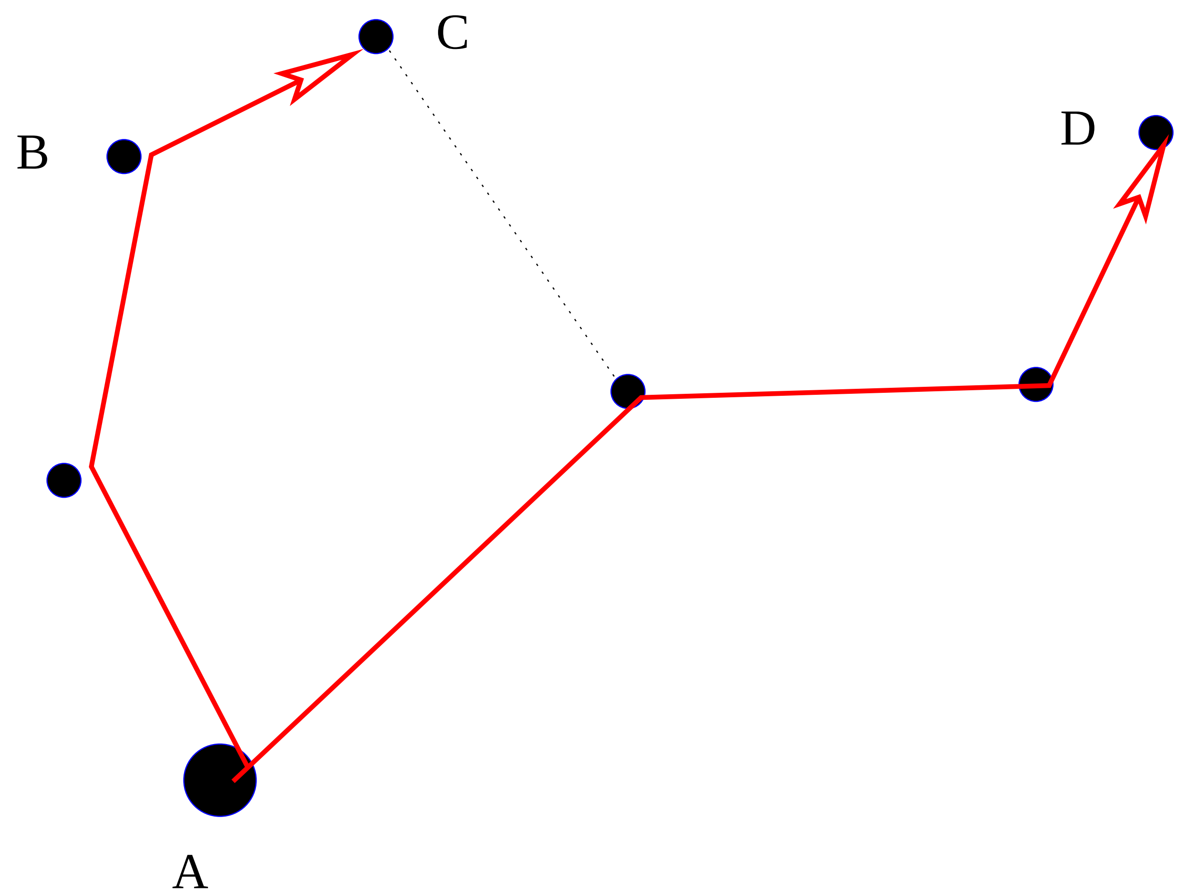
Most basic question: What are we allowed to do when demand patterns change?

Dynamic Routing: Demands=AB,AC



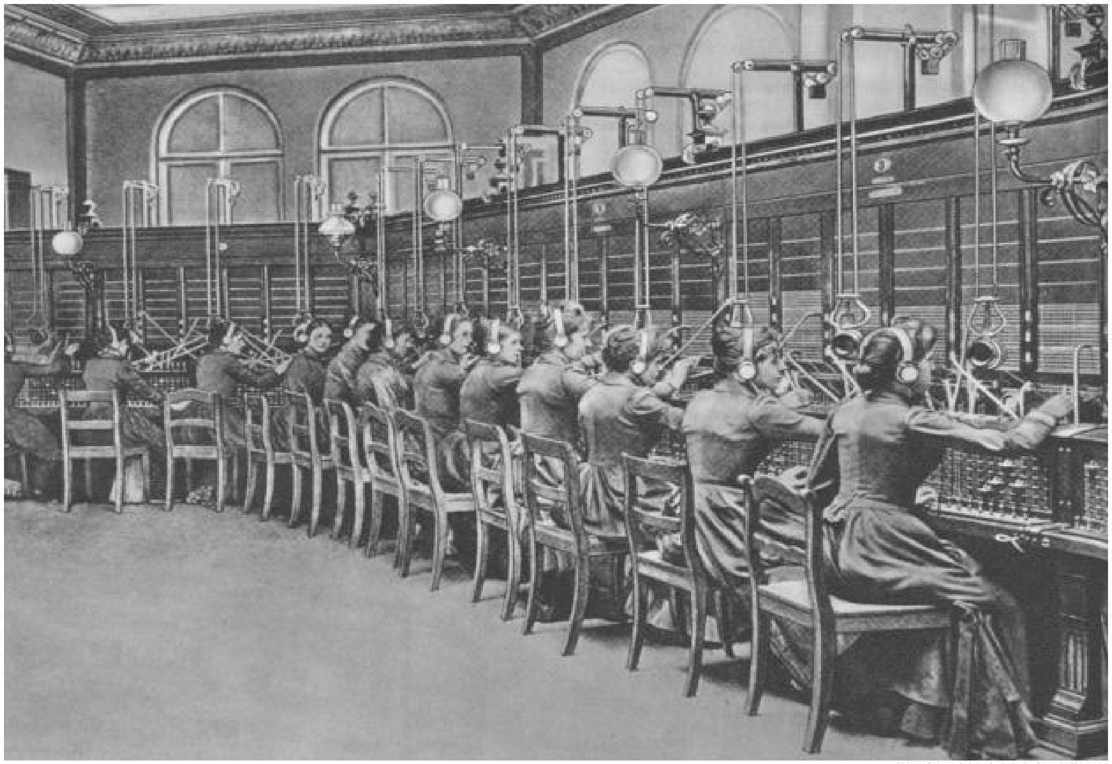
Dynamic Routing: Demands AC, AD

Reroute the demand AC



Partly Dynamic Control Plane

One must leave existing traffic alone.



http://www.s-storbeck.de

Neither model is realistic in modern data networks. We cant change routings on the fly. We also do not have the global picture of what traffic patterns D_{ij} *are (even after analyzing data logs).*

A Third Way: Oblivious Packet Routing

"...the route taken by each packet is determined entirely by itself. The other packets can only influence the rate at which the route is traversed"

Valiant '81

Oblivious Routing

Specify a Routing Template ahead of time, so routing is independent of current conditions in the network.

For each pair of nodes i, j we have designated flow values $f(P)$ to paths P joining i and j such that:

$$\sum_{P \text{ joins } i \text{ and } j} f(P) = 1$$

.

Oblivious Routing

Specify a Routing Template ahead of time, so routing is independent of current conditions in the network.

For each pair of nodes i, j we have designated flow values $f(P)$ to paths P joining i and j such that:

$$\sum_{P \text{ joins } i \text{ and } j} f(P) = 1$$

.

Interpretation: *If in the future we handle some traffic matrix D, then we should send $f(P)D_{ij}$ flow down path P.*

Oblivious Routing

Specify a Routing Template ahead of time, so routing is independent of current conditions in the network.

For each pair of nodes i, j we have designated flow values $f(P)$ to paths P joining i and j such that:

$$\sum_P f(P) = 1$$

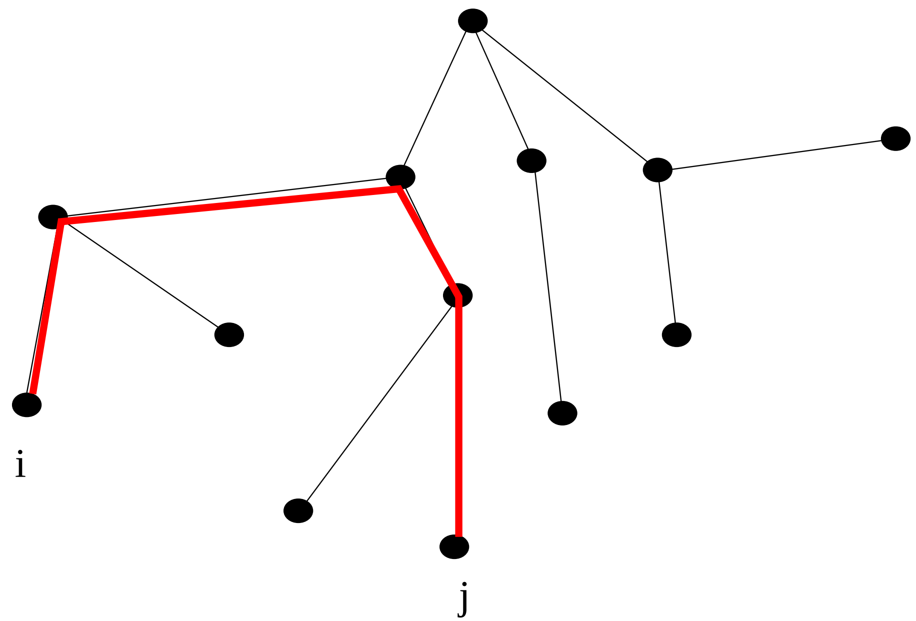
.

Single Path Routing (SPR): $f(P) = 0$ *or* 1. *(Template* $\mathcal{T} = (P_{ij})$*)*

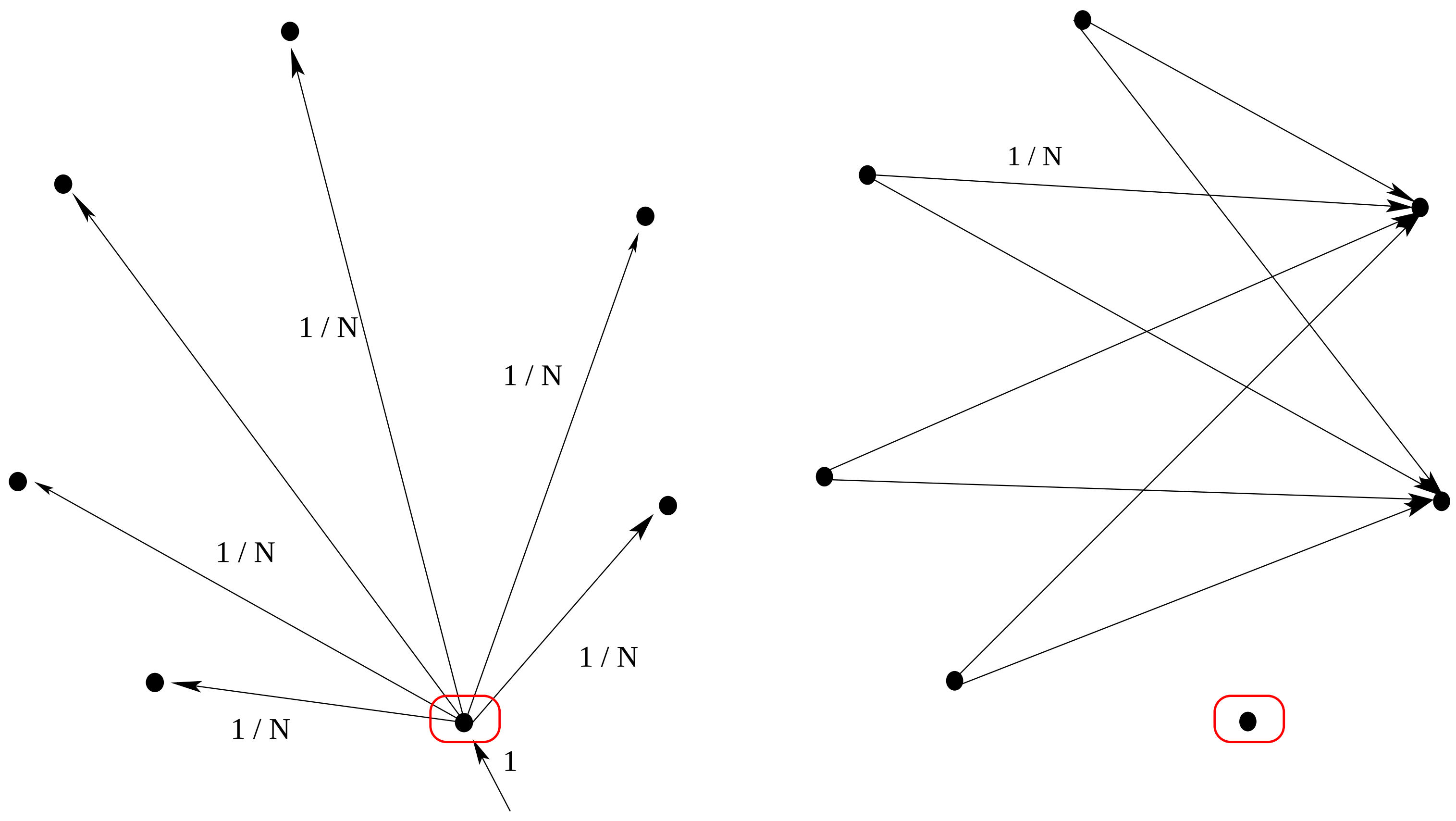
Multi-Path Routing (MPR): $f(P)$ *allowed to be fractions.*

Examples:

Tree Routing



Randomized Load Balancing (Valiant)



Also, Räcke's Oblivious Routing Template

The Robust Network Design Problem Defined (SPR Model)

Given an undirected network $G = (V, E)$ *with edge costs* $c(e)$*, and a universe (convex body)* $\mathcal{U}$ *of demand matrices assumed to be "nice" (e.g., polytime separable).*

Consider a single-path routing template $\mathcal{T} = (P_{ij})$ *for each pair of nodes* i, j*. The template induces the following capacity an edge* e

$$u_{\mathcal{T}}(e) = \max_{D \in \mathcal{U}} \sum_{P_{ij} : e \in P_{ij}} D_{ij}$$

The Robust Network Design Problem Defined (SPR Model)

Given an undirected network $G = (V, E)$ *with edge costs* $c(e)$*, and a universe (convex body)* $\mathcal{U}$ *of demand matrices assumed to be "nice" (e.g., polytime separable).*

Consider a single-path routing template $\mathcal{T} = (P_{ij})$ *for each pair of nodes* i, j*. The template induces the following capacity on an edge* e

$$u_{\mathcal{T}}(e) = \max_{D \in \mathcal{U}} \sum_{P_{ij} : e \in P_{ij}} D_{ij}$$

Find $\mathcal{T}$ *that minimizes* $\sum_e c(e) u_{\mathcal{T}}(e)$*.*

V. Summary of what has been figured out

Given a network topology G, link costs c_e and polytope $\mathcal{U}$ of demands, we define 3 parameters:

$$\text{DYN}, \text{MPR}, \text{SPR}$$

as the cheapest robust networks for $\mathcal{U}$ where we allow dynamic, multi-path oblivious, and single-path oblivious routing respectively.

Two types of questions:

- *how easy is it to compute these parameters?*
- *how do networks costs compare between different models?*

Trivially:

$$\text{DYN} \leq \text{MPR} \leq \text{SPR}$$

What's Known about Complexity

All the parameters are approximable to within a logarithmic factor by metric embeddings (Gupta et al. 2001)

Computing MPR *can be solved in polytime (Ben Ameur, Kerivin 2003). Compact LP formulations given by several groups. These results actually follow from Ben-Tal, Nemirovski 1999.*

There is a polytime $O(1)$ *approximation when* $\mathcal{U}$ *is defined by an "asymmetric" hose model (Gupta, Kumar, Pal, Roughgarden 2003)*

What's Known about Complexity

Computing DYN *is also a convex minimization problem. However the separation problem is coNP-hard via a reduction from* SPARSEST CUT *(Chekuri, Scutella, Oriolo, S. 2005).*

Theorem. If for any $\epsilon > 0$, there is a polytime $(2 - \epsilon)$-approximate algorithm for the dynamic robust separation problem,
THEN there is a polytime constant approximation for SPARSEST CUT.

SPR *is Max-SNP hard (Fingerhut et al. 1997, Gupta et al. 2001) by a simple reduction from* STEINER TREE.

SPR *is hard to approximate within polylog (Olver, S. 2010). Reduction from uniform Buy-At-Bulk (cf. hardness results of M. Andrews).*

Approximability Summary for general polytopes $\mathcal{U}$

MPR	*exact*
SPR	$\Theta(polylog)$
DYN	**?**

What's known about Gaps

$O(logn)$ Gupta 2001

OPT(Dynamic) −−−−−− OPT (MPR) −−−−−− OPT(SPR) −−−−− OPT(Tree)

What's known about Gaps

$O(logn)$ Gupta 2001

OPT(Dynamic) ------ OPT (MPR) ------ OPT(SPR) ----- OPT(Tree)

$\Omega(\log n)$ Goyal, Olver, S. 2009

What's known about Gaps

$O(logn)$

$\Omega(polylog)$ Olver, S. 2010

OPT(Dynamic) ------ OPT (MPR) ------ OPT(SPR) ----- OPT(Tree)

$\Omega(\log n)$ Goyal, Olver, S. 2009

VI. The VPN Problem

Input: *an undirected network* $G = (V, E)$ *with edge costs* $c(e)$*, and a hose polytope* $\mathcal{U}$ *defined by marginal values* D_i*.*
(Can think of these as 1 *if you like)*

Recall Hose Demand Matrices:
$\mathcal{U} = \{(D_{ij}) : \sum_j D_{ij} \leq D_i, \quad \sum_j D_{ji} \leq D_i \quad \forall i\}$

VI. The VPN Problem

Input: *an undirected network* $G = (V, E)$ *with edge costs* $c(e)$*, and a hose polytope* $\mathcal{U}$ *defined by marginal values* D_i

Consider a single-path routing template $\mathcal{T} = (P_{ij})$ *for each pair of terminals* i, j*. How much capacity is needed on edge* e*:*

$$u_{\mathcal{T}}(e) = \max_{D \in \mathcal{U}} \sum_{P_{ij} : e \in P_{ij}} D_{ij}$$

VI. The VPN Problem

Input: *an undirected network* $G = (V, E)$ *with edge costs* $c(e)$*, and a hose polytope* $\mathcal{U}$ *defined by marginal values* D_i*.*

Consider a single-path routing template $\mathcal{T} = (P_{ij})$ *for each pair of terminals* i, j*. How much capacity is needed on edge* e*:*

$$u_{\mathcal{T}}(e) = \max_{D \in \mathcal{U}} \sum_{P_{ij} : e \in P_{ij}} D_{ij}$$

Find: *a template* $\mathcal{T}$ *that minimizes* $\sum_e c(e) u_{\mathcal{T}}(e)$*.*

The resulting network with capacities $u(e)$ *called an optimal VPN.*
Note: The capacities in a VPN may be fractional even though template is integral.

The VPN Conjecture

VPN Conjecture. *There is an optimal VPN induced by a template P_{ij} only using edges in some fixed tree T.*

I.e., there is an optimal VPN induced by a Tree-Routing Template.

Not immediately clear this is true even if G is a ring. (Hurkens, Keijsper, Stougie 2007)

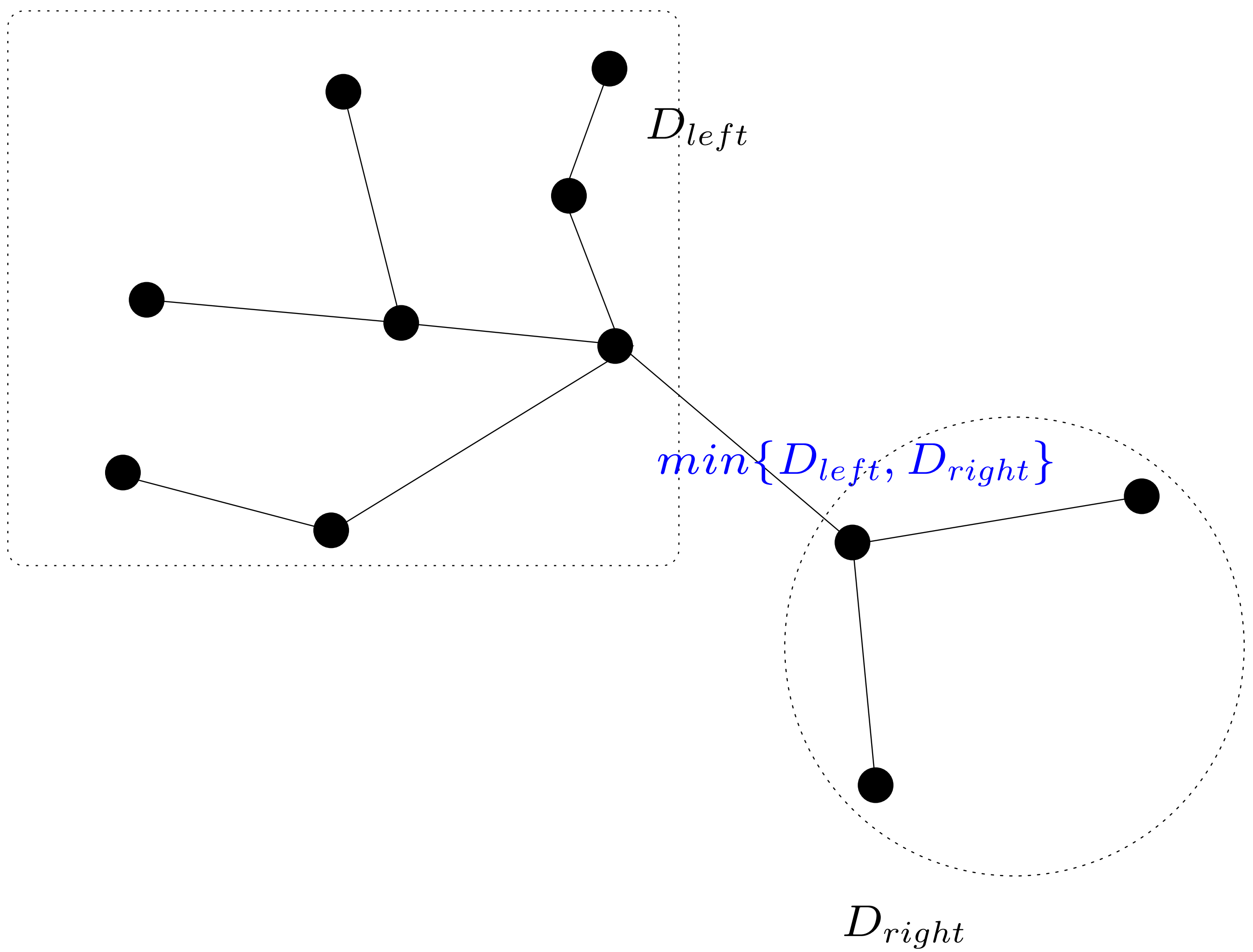
Step 1. Can we compute the Best Tree-Template?

Let OPT *be the optimal cost of a VPN.*
Let VPN-TREE *denote the optimal capacity cost if we are only allowed tree-templates.*

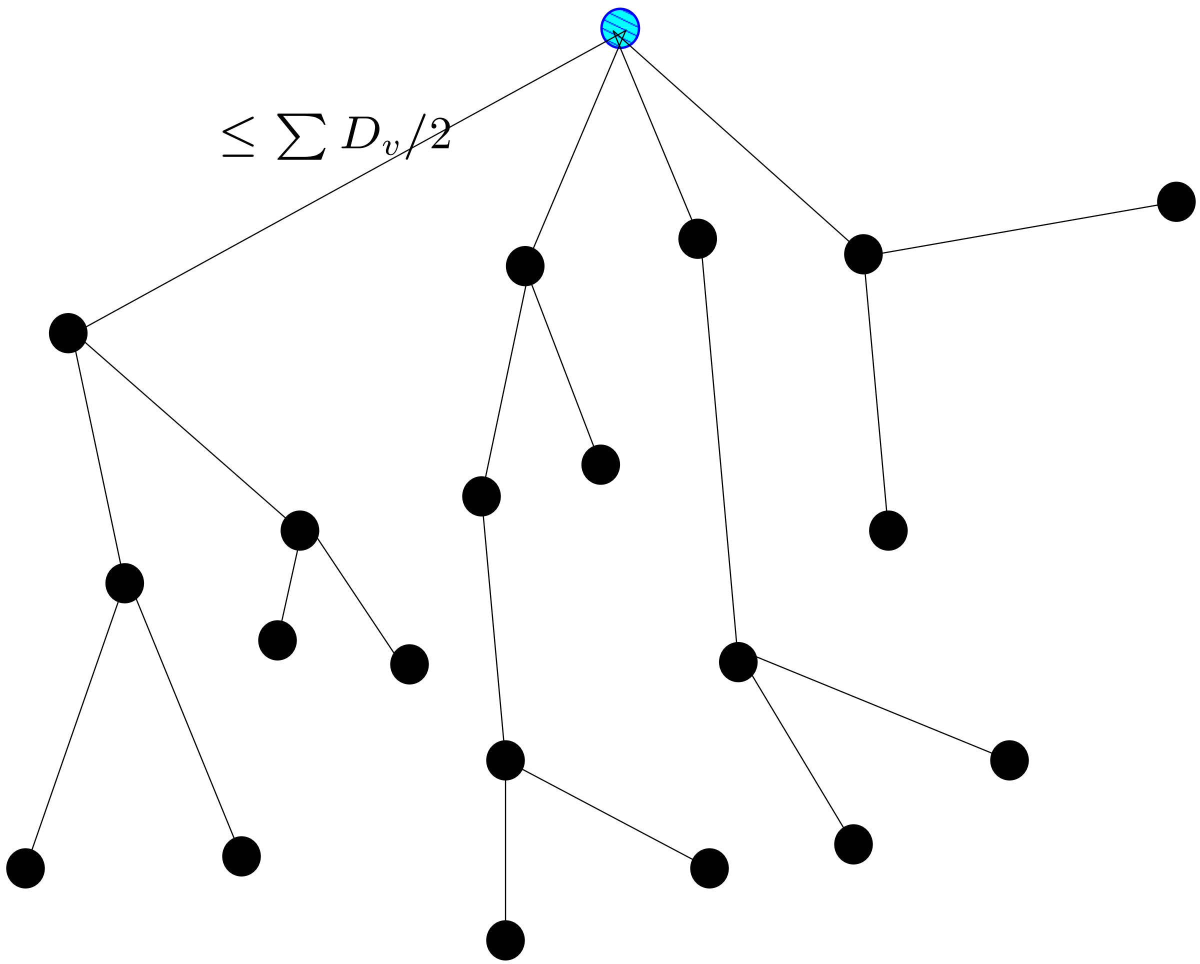
Theorem: (Fingerhut, Suri, Turner 1997, Gupta, Kleinberg, Kumar, Rastogi, Yener 2001)

$$\text{VPN-TREE} \;\leq\; 2\,\text{OPT}$$

Step 1.1. Given a fixed tree, how much capacity is needed?

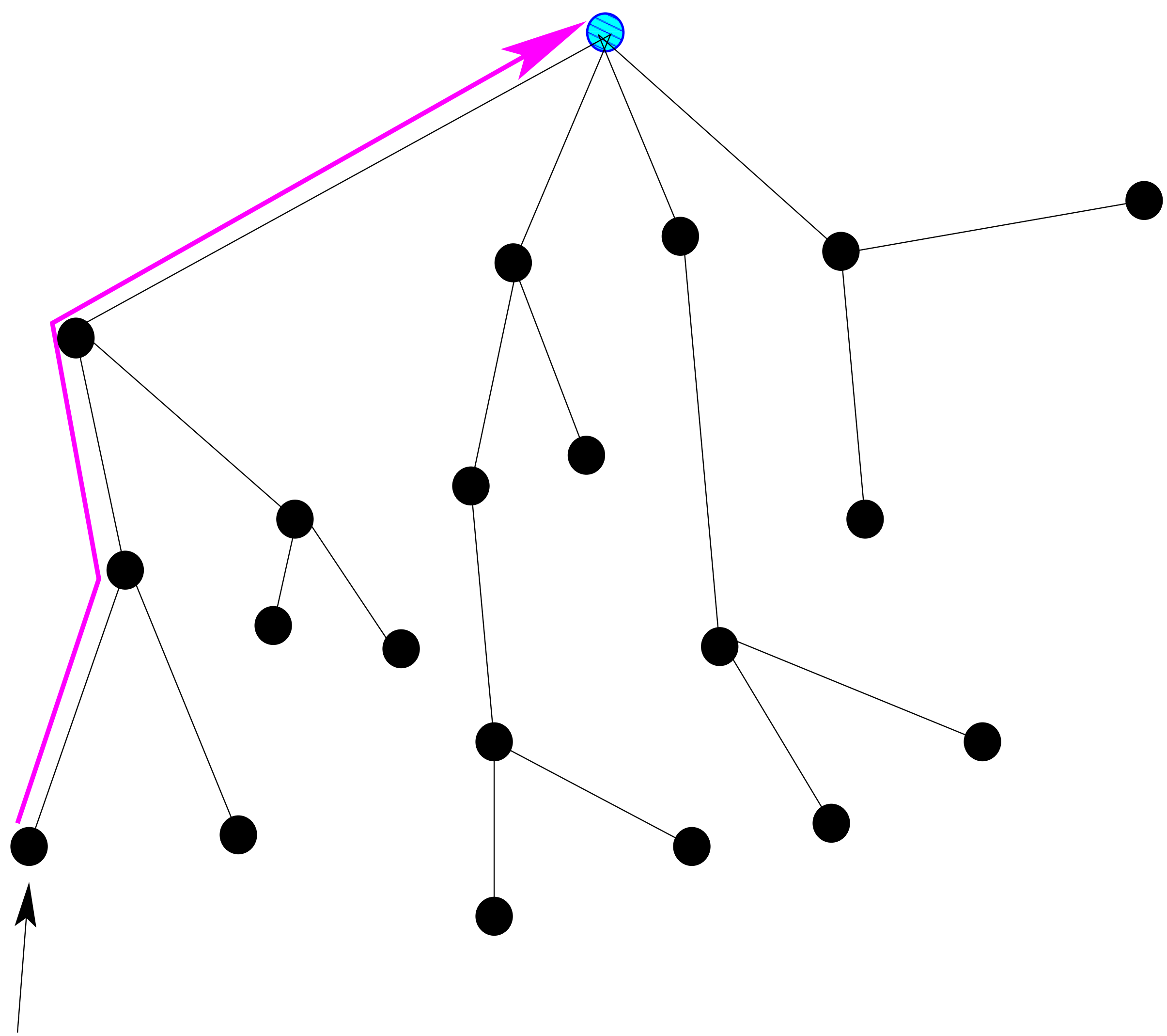


How much capacity needed for an optimal tree?



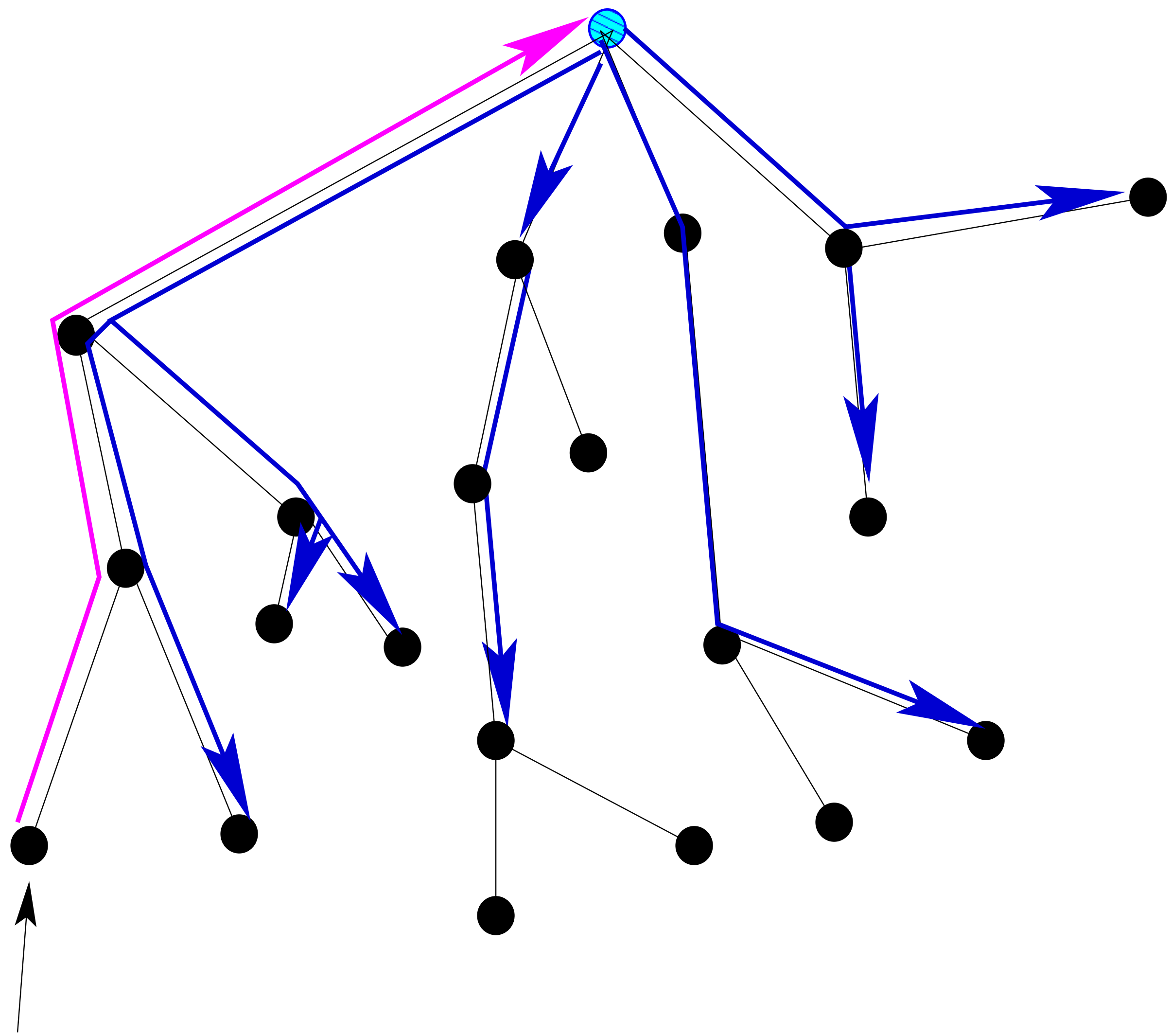
Find the "center" of an optimal tree.

There is enough capacity for terminals to route to the Center



Corollary. *Terminals should route on shortest paths*
$\Rightarrow$ *find best VPN-tree by computing n shortest path trees.*

Alternative Oblivious Template: HUB Routing



Natural Approach to Prove VPN Conjecture

Look at the fractional relaxation and prove that it always has an integral optimal solution.

In other words, show that the multi-path routing LP satisfies

$$\mathrm{MPR} = \mathrm{SPR}$$

and then pray that a tree appears in the solution somehow.

Natural (but Wrong) Approach to Prove VPN Conjecture

Look at the fractional relaxation and prove that it always has an integral optimal solution.

In other words, show that the multi-path routing LP satisfies

$$\mathrm{MPR} = \mathrm{SPR}$$

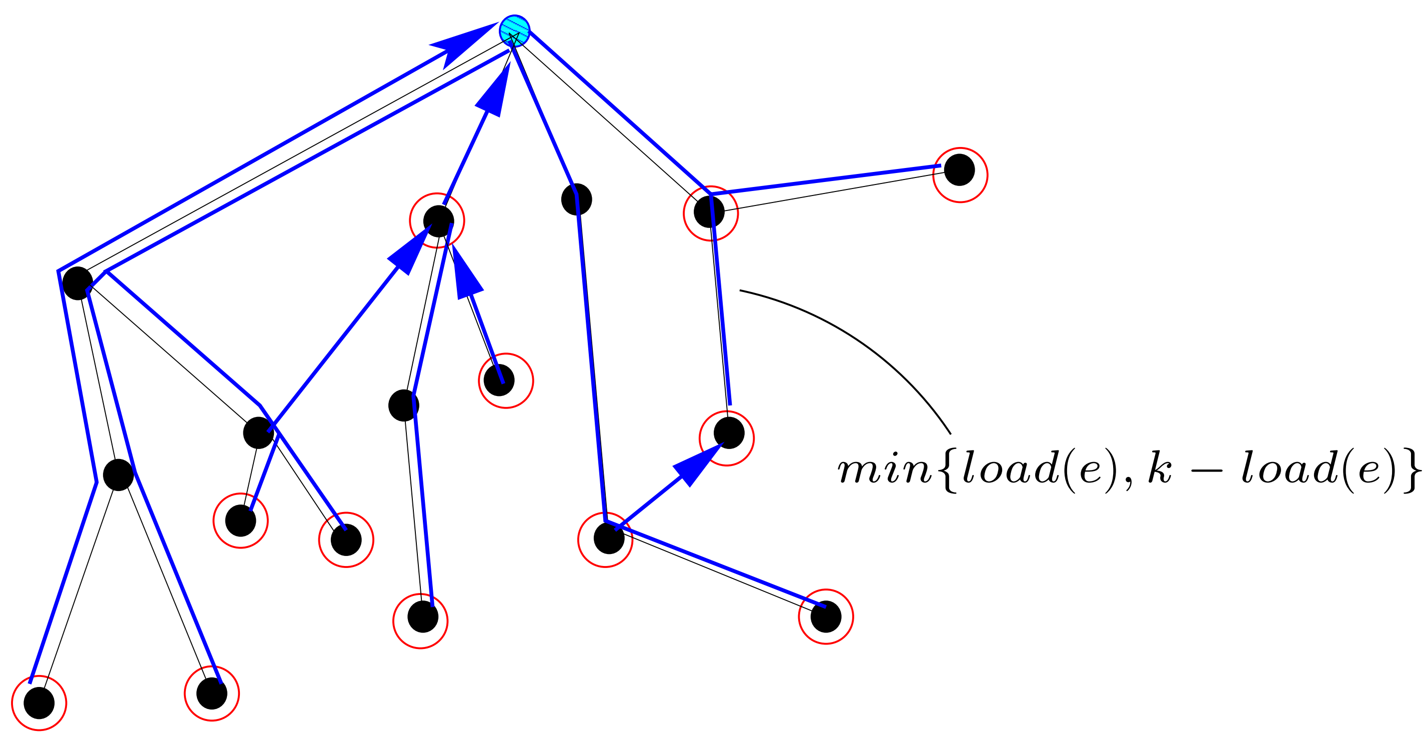
and then pray that a tree appears in the solution somehow.

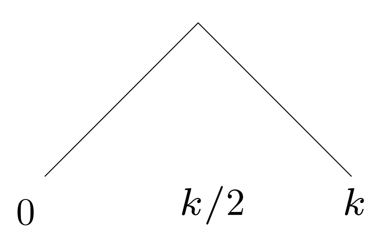
The Strong VPN Conjecture turns out to be false. We can have

$$\mathrm{MPR} < \mathrm{SPR}$$

Step 2. Pyramidal Hub Routing (Grandoni, Kaibel, Oriolo, Skutella 2007)

Given: A set of terminals W each sending one unit flow to a root r.





Pyramidal Routing Conjecture (PRC)

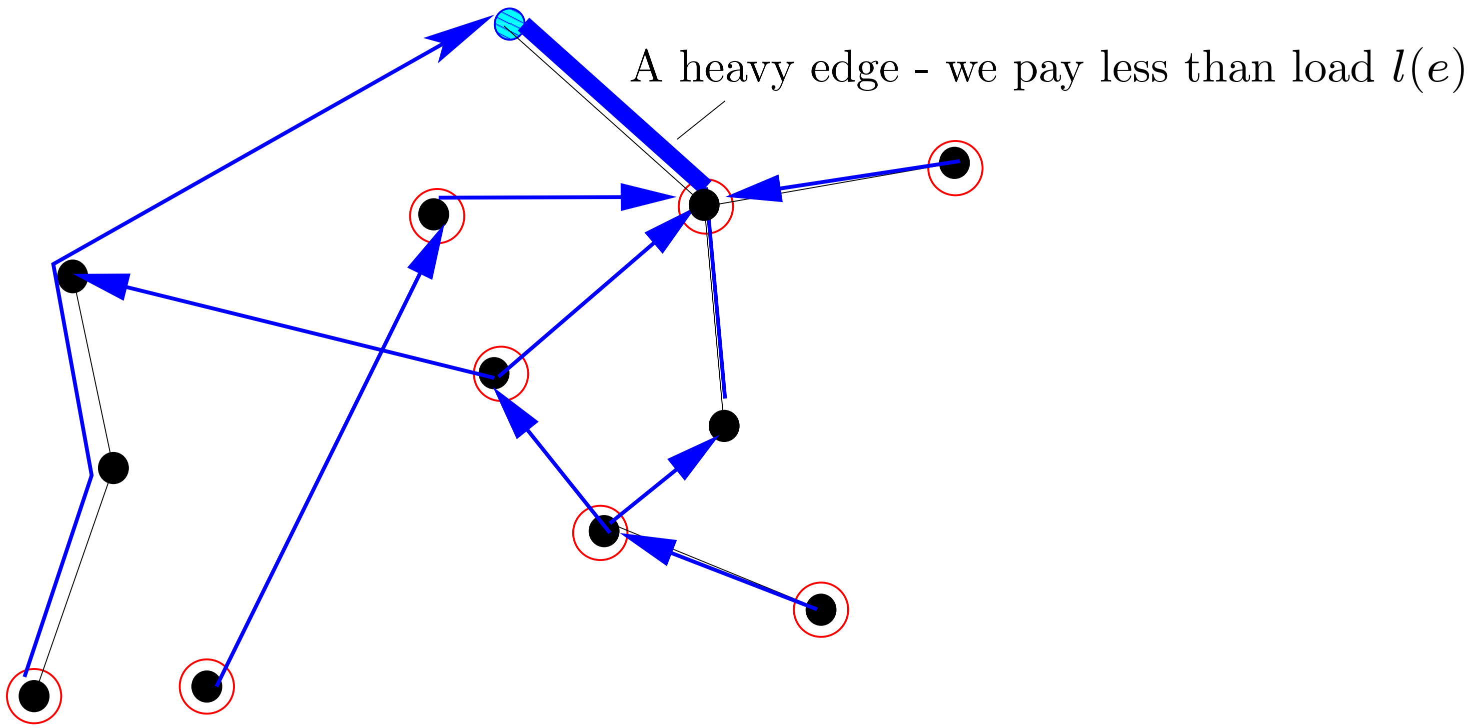
PRC: There is an optimal pyramidal routing where the terminals route to the hub on a tree.

Grandoni et al. show:

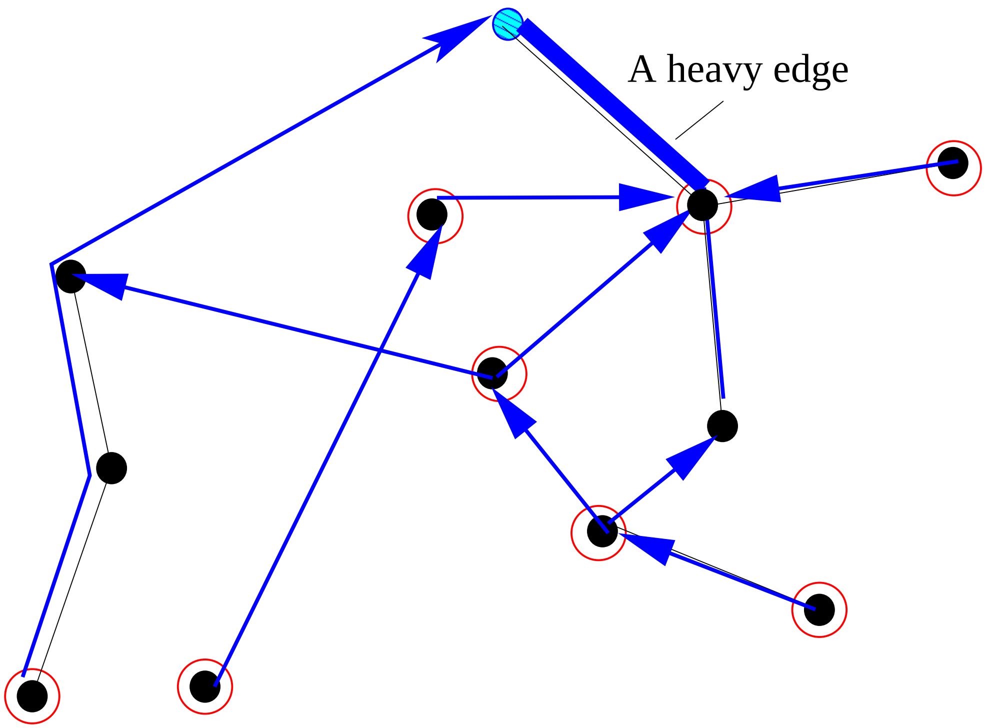
$$PRC \Rightarrow \quad VPN \quad Conjecture$$

Pyramidal is actually equivalent to VPN design.
From now on we always consider $D_i = 1$.

Heavy Edges are the problem



Step 3. Cost Sharing



Pay for your own light edges, but other peoples' heavy edges

Cost Sharing

Let H be the subgraph of heavy edges.

Let P_{ur} be the path chosen by u in a pyramidal routing.

Define

$$C'(u) = \sum_{e \in P_{ur} \Delta H} c(e).$$

$C'(u)$ *will be* u*'s share of the cost of the pyramidal routing.*

Cost Sharing

Let P_{ur} *be the path chosen by* u *in a pyramidal routing.*

Define $C'(u) = \sum_{e \in P_{ur} \Delta H} c(e)$.

$$\begin{aligned} \sum_u C'(u) = & \quad \sum_u \sum_{e \in P_{ur} \Delta H} c(e) \\ = & \quad \sum_{e \in E \setminus H} l(e)c(e) + \sum_{e \in H} (k - l(e))c(e) \end{aligned}$$

which is just the cost of the pyramidal routing.

Step 4. T-joins

- *Recall: For any even set of nodes T. A T-join is a set of edges J such that the odd-degree nodes in the graph (V, J) are precisely T.*
- *Let T' be those nodes with odd degree in the heavy subgraph H.*
  N.B. $|T'|$ even and H is a T'-join.

T-joins

- *Recall: For any even set of nodes T. A T-join is a set of edges J such that the odd-degree nodes in the graph (V, J) are precisely T.*
- *Let T' be those nodes with odd degree in the heavy subgraph H.*
  N.B. $|T'|$ even.
- *For each terminal u, define $T_u = T' \Delta \{u, r\}$.*
  N.B. $|T_u|$ even.
- *Let M_u be a minimum cost T_u-join.*

T-joins

- *Let T' be those nodes with odd degree in the heavy subgraph H. N.B. $|T'|$ even.*
- *For each terminal u, define $T_u = T'\Delta\{u, r\}$. N.B. $|T_u|$ even.*
- *Let M_u be a minimum cost T_u-join.*

Claim: $C'(u) \geq C(M_u)$

Proof:

1. $C'(u) = C(P_{ur}\Delta H)$.
2. P_{ur} *is a* $\{u, r\}$*-join and* H *is a* T'*-join* $\Rightarrow$ $P_{ur}\Delta H$ *is a* T_u*-join.*

T-joins

- *Let T' be those nodes with odd degree in the heavy subgraph H. N.B. $|T'|$ even.*
- *For each terminal u, define $T_u = T'\Delta\{u, r\}$. N.B. $|T_u|$ even.*
- *Let M_u be a minimum cost T_u-join.*

Claim: $C'(u) \geq C(M_u)$

Proof:

1. $C'(u) = C(P_{ur}\Delta H)$.
2. *P_{ur} is a $\{u, r\}$-join and H is a T'-join $\Rightarrow$ $P_{ur}\Delta H$ is a T_u-join.*

Corollary: *Pyramidal Routing costs at least $\sum_{u\in W} C(M_u)$.*

N.B. Lower bound depends on H only via its odd-degree nodes T'.

A T-join Inequality

Let $C_{SP}(v)$ denote the cost of routing one unit from each terminal in W to v on a shortest path. Ie $\sum_{u \in W} SP(u, v)$.

To prove the PR Conjecture, it is sufficient to find a node v such that $C_{SP}(v)$ is at most the pyramidal routing cost.

A T-join Inequality

Let $C_{SP}(v)$ denote the cost of routing one unit from each terminal in W to v on a shortest path. Ie $\sum_{u \in W} SP(u, v)$.

To prove the PR Conjecture, it is sufficient to find a node v such that $C_{SP}(v)$ is at most the pyramidal routing cost.

Theorem 1. *If F is a a multigraph obtained as the union of the "joins" $(M_u : u \in W)$. Then there is some node v such that F has enough capacity for each terminal to route to v in F.*

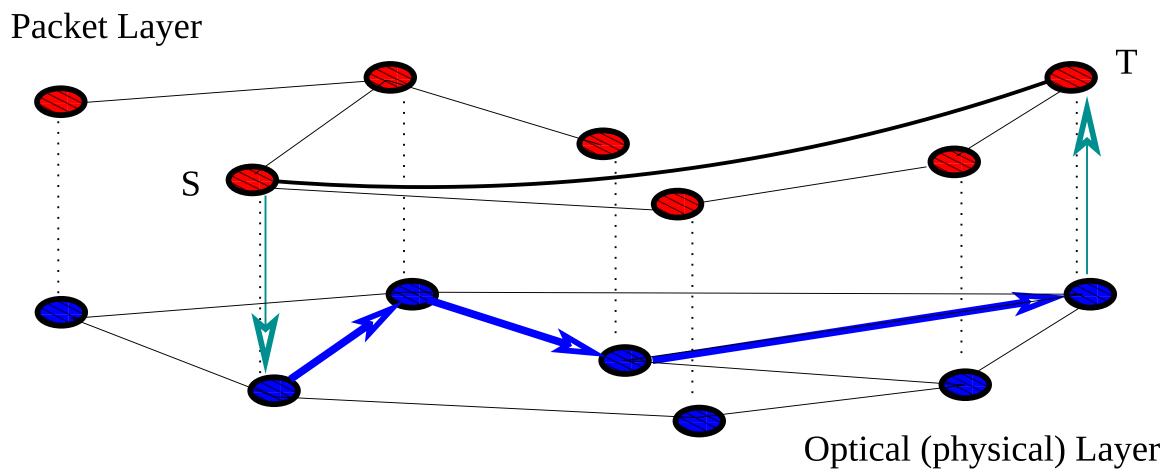
Proof uses uncrossing on the following inequalities

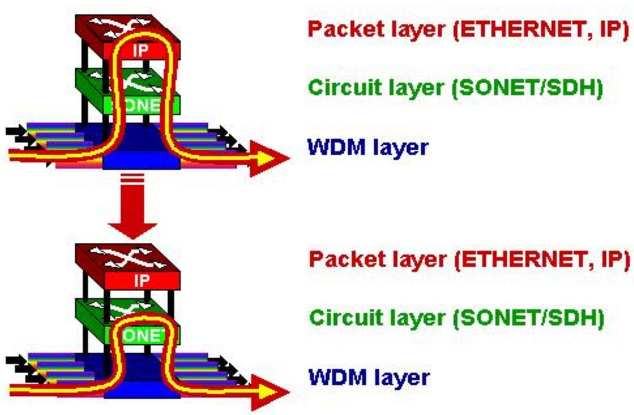
$$|\delta_F(S)| \geq |S \cap W| \quad \text{for every } T'\text{-odd set } S$$

VI. IP over Optical

Switching Costs

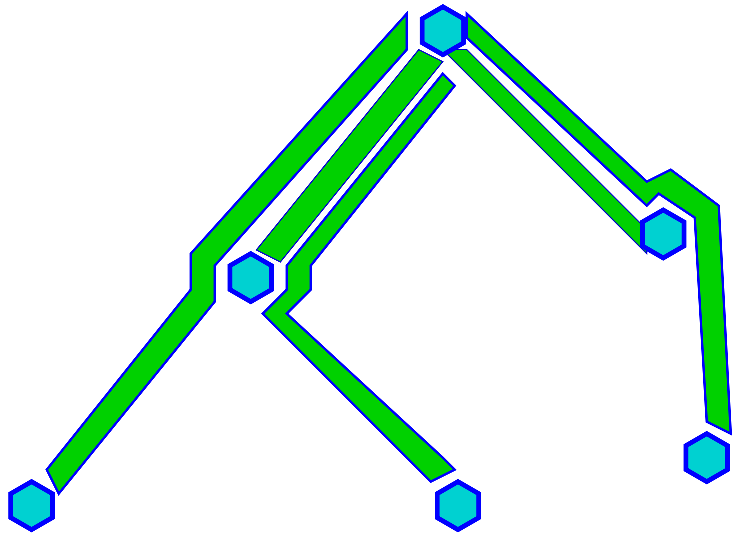
Groom traffic onto cost-efficient pipes (circuits) but avoid managing traffic at every node?





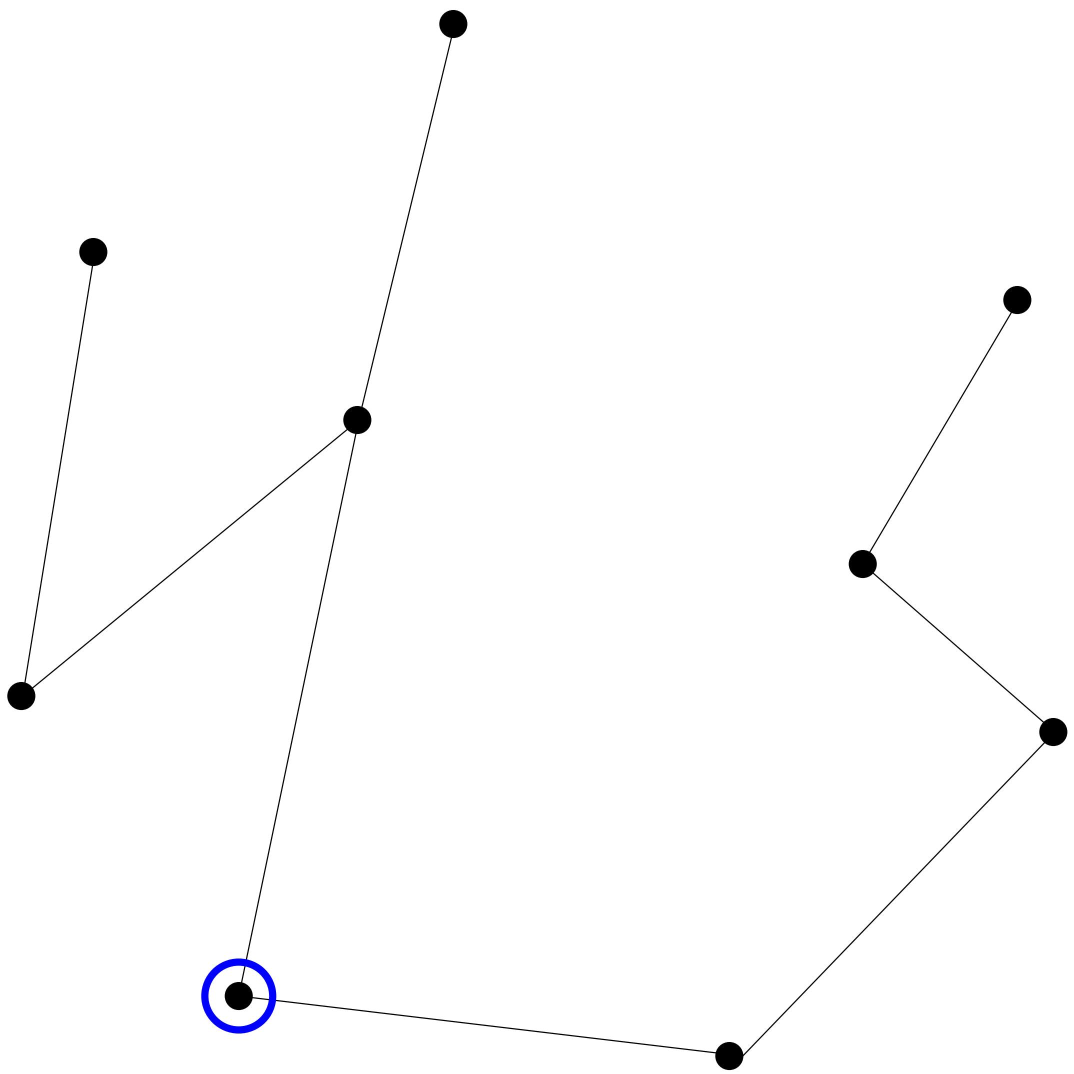
Hub Routing Using Circuits

In Optimal VPN-tree, a node i provisions a STATIC circuit to the root of size $2D_i$.

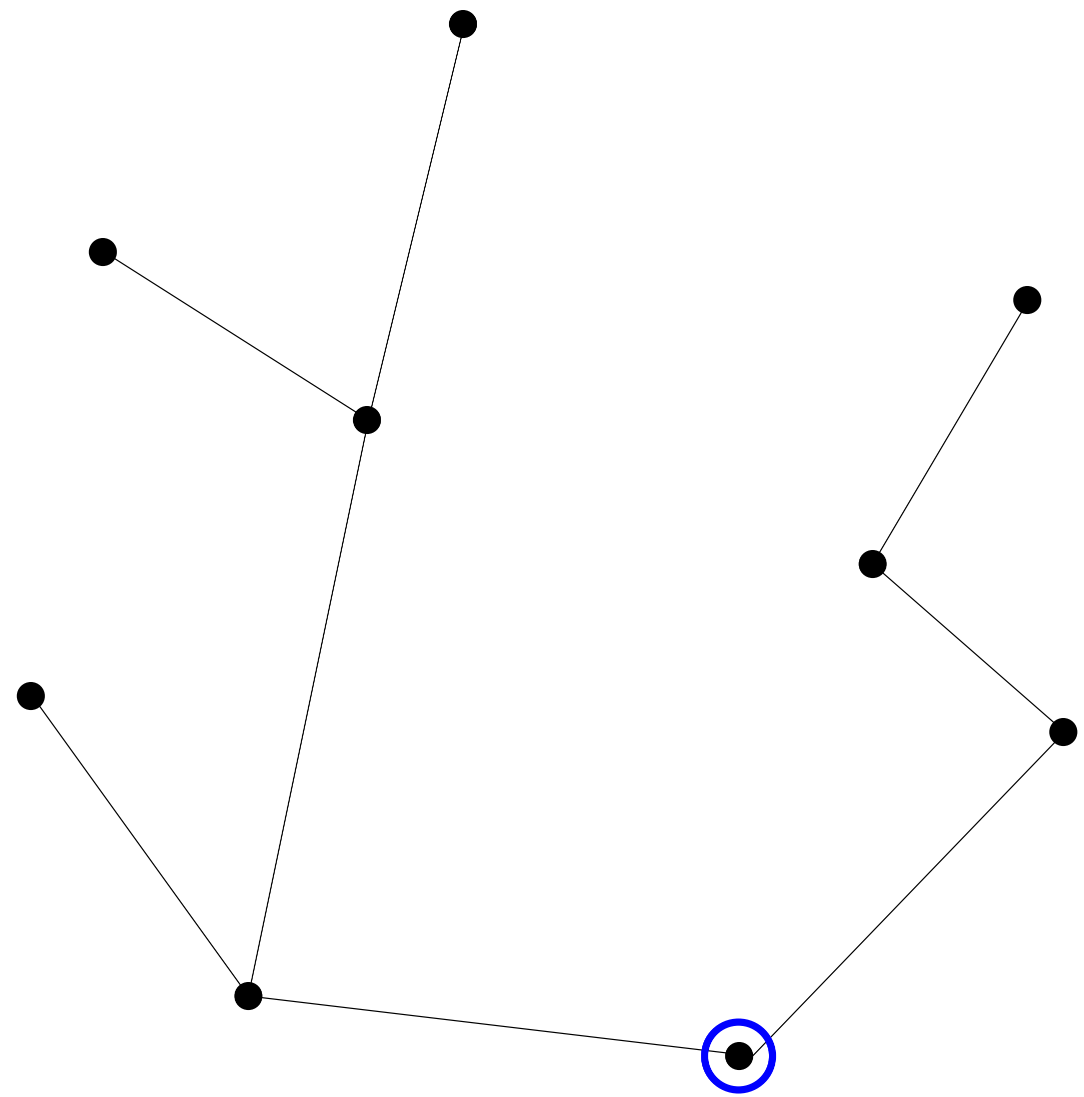


Downside: **single point of failure**.

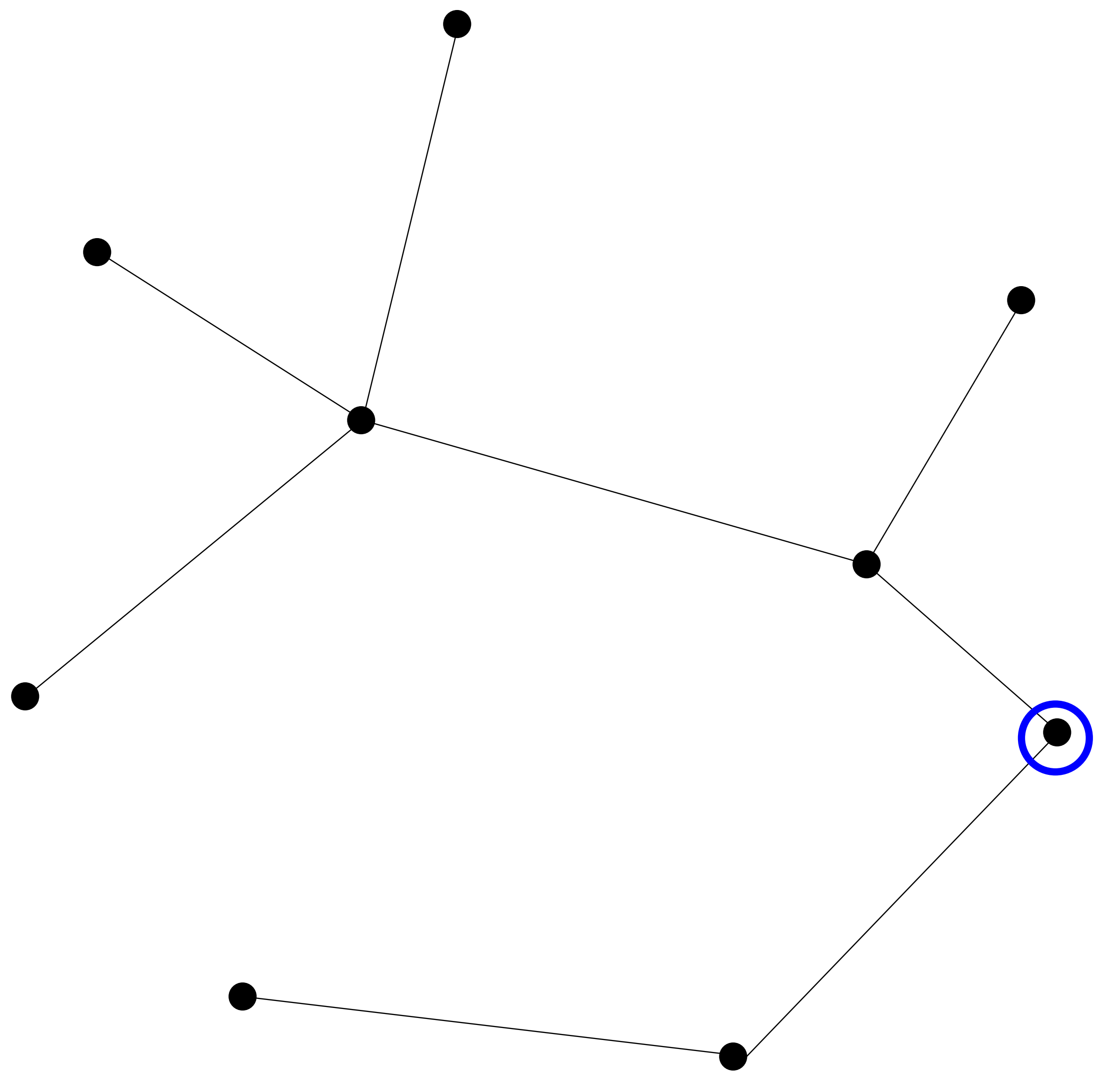
Load Balancing = Convex Combination of Hub Routing



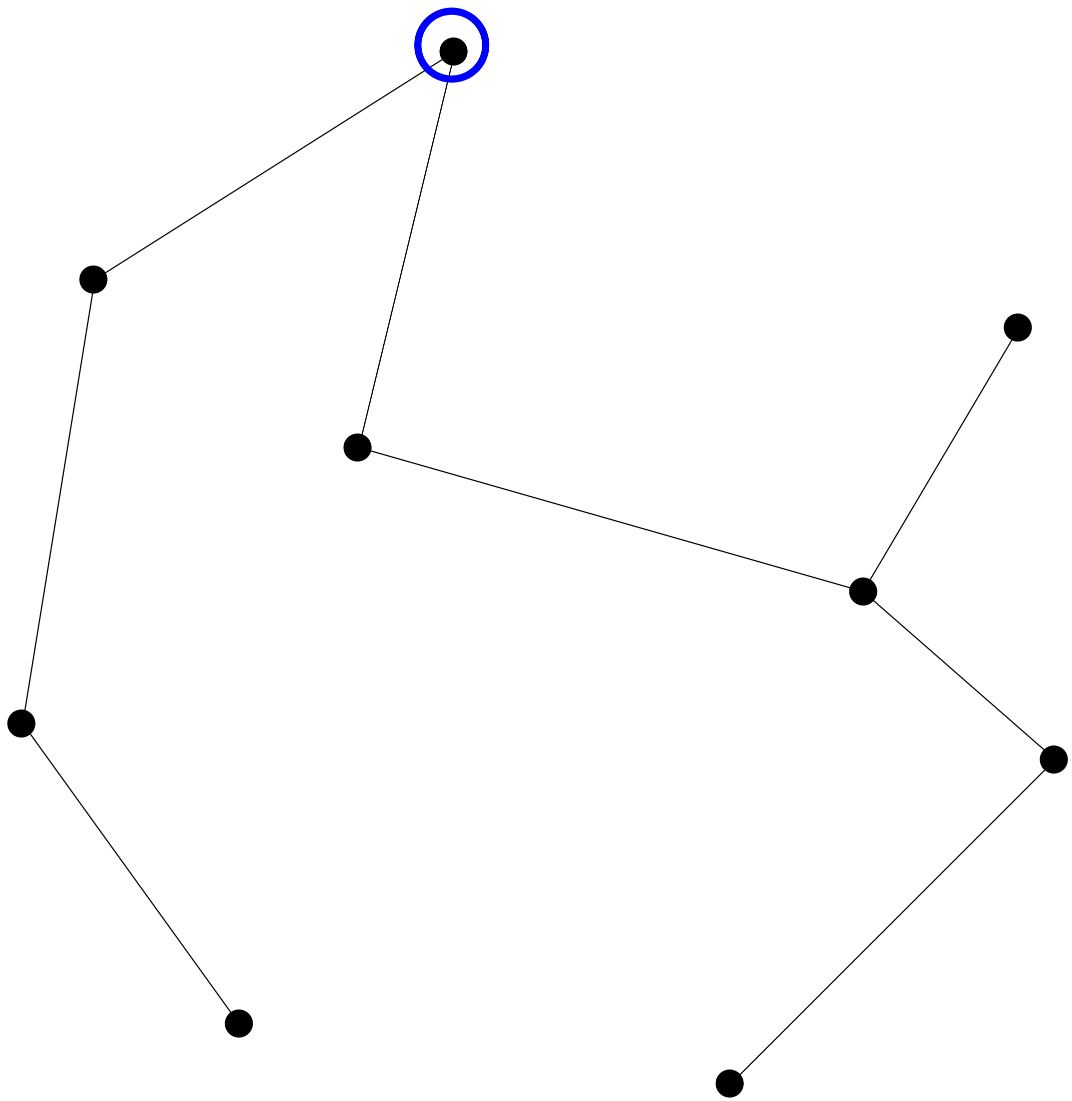
Load Balancing = Convex Combination of Hub Routing



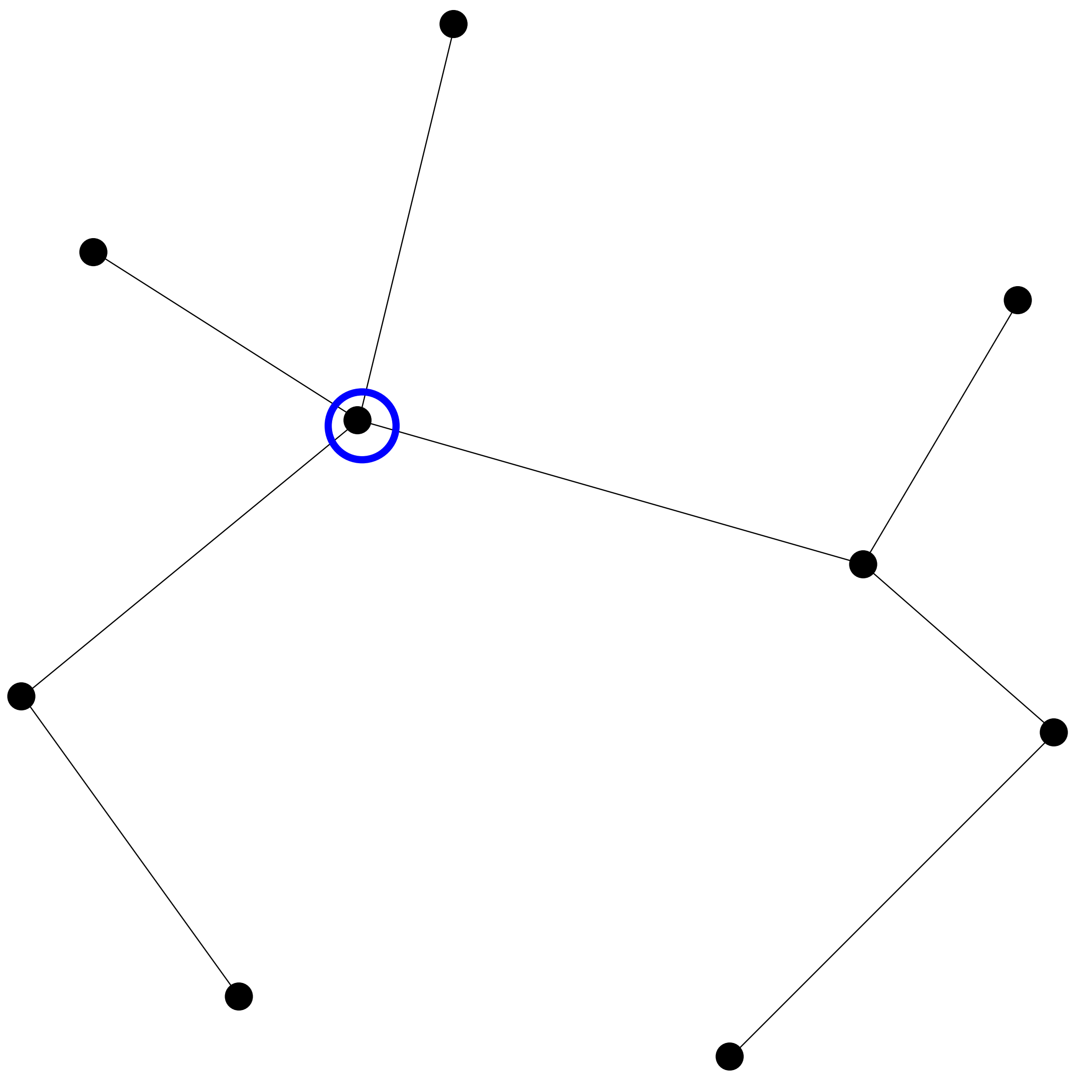
Load Balancing = Convex Combination of Hub Routing



Load Balancing = Convex Combination of Hub Routing



Load Balancing = Convex Combination of Hub Routing



$\Rightarrow$ *Load Balancing is MORE expensive than hub routing.*

Selective Load Balancing (Multi-Hub Routing)

The costs of the different shortest path trees can vary widely.

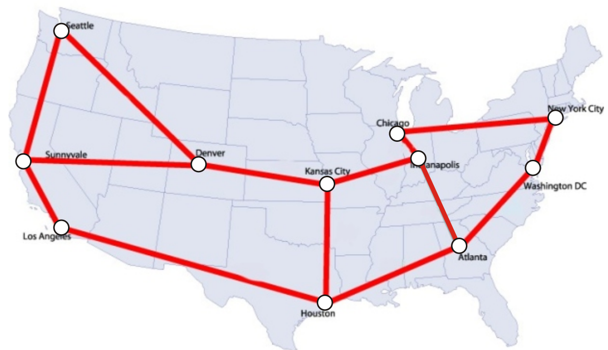
There tends to be a "core" of network nodes which have similiar costs to optimal hub

$\Rightarrow$ *Load Balance across the M best nodes*

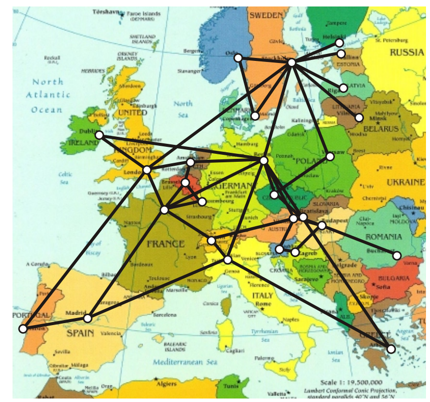
Eliminates single point of failure, but cost is comparable to VPN-tree solution.

JANET, Abilene, Geant

(a)JANET topology

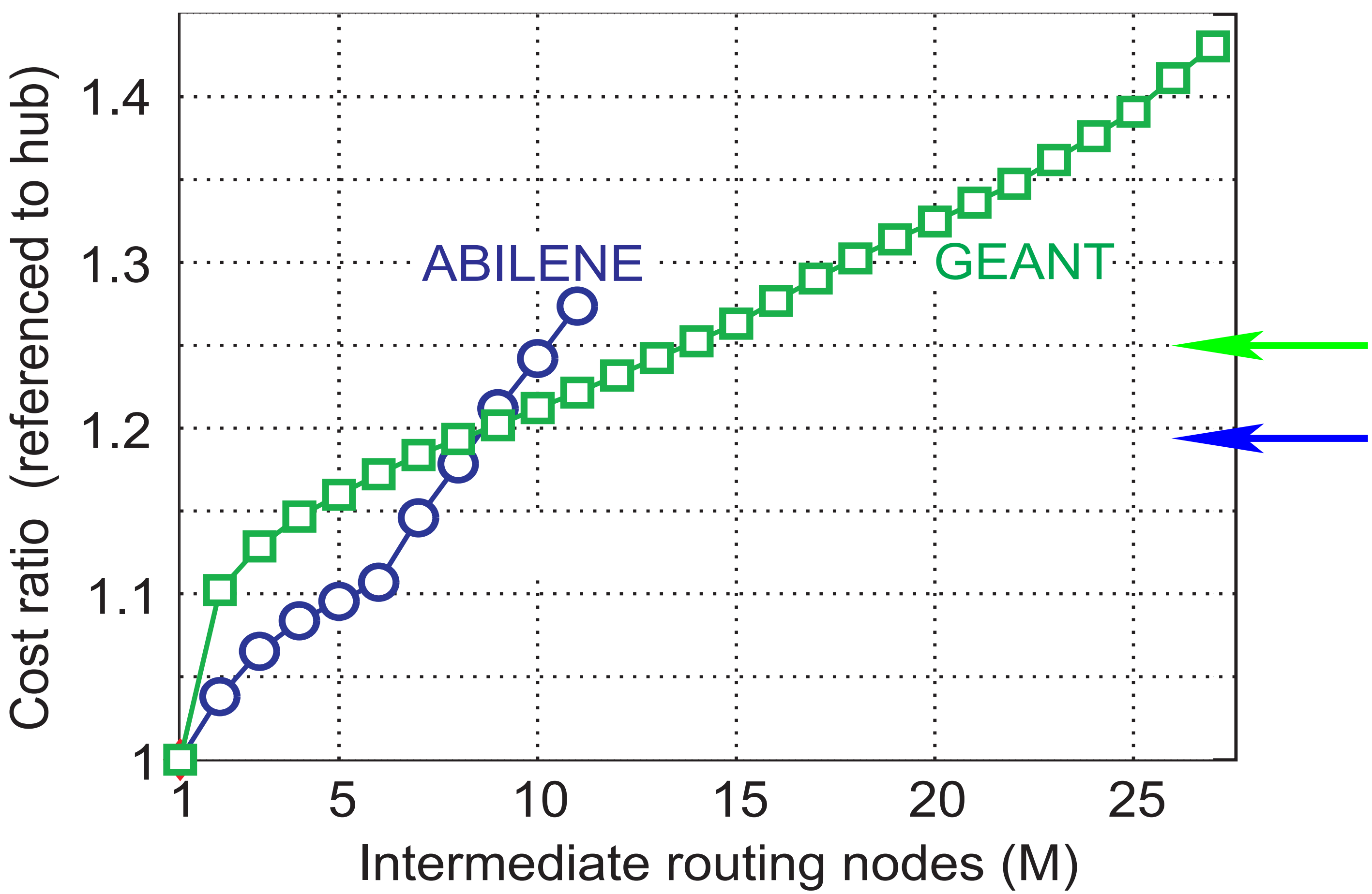


(b) ABILENEtopology



(c)GEANT topology

Empirical Results: Selective Load Balancing versus Multi-Hop Routing



Empirical Results: Robustness Premium ρ

Cost of supporting all hose matrices divided by cost of routing a single benchmark demand matrix.

	JANET	*ABILENE*	*GEANT*
(single-hop static) ρ_{static}	8	11	27
ρ_{SP}	2.43	2.39	2.48
ρ_{VPN}	1.41	1.36	1.13
VPN/SP	1.63	1.46	1.31

More expensive than premiums witnessed by Applegate-Cohen, but then we are optimizing for a larger universe of matrices.

Possible Directions

1. *Generalized VPN Conjecture*
2. *several interesting universes of demand matrices where we do not know if constant factor algs exist.*
3. *What happens with capacities?*
4. *more (empirical or theoretical) work on multihub routing in capacitated networks; resilience.*
5. *hierarchical hubbing*
6. *settle the approximation factor for dynamic routing*