

Shortest Path versus Multi-Hub Routing in Networks with Uncertain Demand

Alexandre Fréchet^{*}, F. Bruce Shepherd^{*}, Marina K. Thottan[†] and Peter J. Winzer[‡]

^{*}Department of Mathematics and Statistics, McGill University, Canada

[†]Bell Labs, Emerging Technology Division, Murray Hill, NJ, USA

[‡]Bell Labs, Optical Transmission and Networks Research Department, Holmdel, NJ, USA

Abstract—We study a class of robust network design problems motivated by the need to scale core networks to meet increasingly dynamic capacity demands. Past work has focused on one of two models. First, design the network for the known point-to-point peak demands. Second, design the network to support all hose matrices (all matrices not exceeding marginal bounds at the nodes). Both models may be too conservative if additional information on traffic patterns is available. We introduce a capped hose model to explore a range of traffic scenarios, which includes the above two as special cases. It is known that optimal network designs for the hose model are always determined by single-hub routing, and for the fixed-demand model are based on shortest-path routing. We demonstrate that a wider variety of routing templates is required to address the broader spectrum of capped hose matrices. We propose the use of hierarchical multi-hub routing templates, a generalization of hub and tree routing. Our empirical analysis is based on a heuristic for the resulting robust network design problem. These lead to two important findings: (1) designs based on multi-hub routing are often preferable to both hub and shortest path, (2) it may be possible for a carrier to sample their traffic in order to determine which type of routing is most cost-effective for their network.

I. INTRODUCTION

Traditional communication networks have been designed based on knowledge of an *expected traffic demand matrix* that specifies the aggregate traffic between every pair of nodes in the network and evolves slowly, over a time scale of months or years. In an era of data-dominant communication networks with dynamic demand patterns, this traditional network design methodology loses its cost-effectiveness. For instance, in emerging data services such as YouTube, user generated video content and flash crowds are triggering an increasing amount of dynamic uncertainty in the traffic demand. One specific concern is that the real-time estimation of the dynamically changing traffic patterns in large data networks is intractable. Moreover, a pessimistic approach of designing for peak point-to-point demands (also referred to as the fixed-demand model) throws away critical information for achieving “cost-sharing” between the varying demand patterns that arise over time.

A traffic model that has gained considerable popularity in coping with dynamic demands is the *hose model* [1], [2], [3]. In this model, only bounds on the ingress/egress traffic of the nodes are known. These bounds (called *marginals*) are physically represented by the various interface speeds of networking hardware at any given node. The actually encountered point-to-point traffic distributions are subject to dynamic variations

during network operation, but the network is provisioned to be able to support any possible pattern meeting the ingress/egress bounds. Formally, in the symmetric (undirected) demand case, the space of *hose matrices* \mathcal{H}_U consists of symmetric matrices D satisfying the bound: $\sum_j D_{ij} \leq U(i)$ for each node i , with marginal $U(i)$; in addition, the main diagonal of these matrices is typically set to zero, reflecting the fact that nodes do not send traffic to themselves.

The flexibility of the hose model, however, can also be a drawback since totally random demand matrices rarely occur in practice and the operator usually has additional information a priori about their traffic patterns. For instance, it may be that there are known bounds on the maximum demand $U(i, j)$ between pairs of nodes i, j . If we add these as constraints, i.e. $D_{ij} \leq U(i, j)$, to the hose constraints, we call this a *capped hose model*. Depending on the relative sizes of the $U(i, j), U(i)$, the space of traffic patterns spanned by the capped hose model ranges from purely deterministic point-to-point demands ($D_{ij} = U(i, j)$ for each i, j if the $U(i)$'s are very large) to the space of unconstrained (uncapped) hose traffic (if the $U(i, j)$'s are large, e.g., $= \min(U(i), U(j))$). One may set these parameters accordingly, so as to avoid the massive over-provisioning that may arise for either the fixed-demand or unconstrained hose traffic models [4].

In order to solve the design problem for the capped hose model, we propose the use of a class of oblivious routing strategies, called *hierarchical hubbing routing templates* (HH) ([5]), which generalizes the popular hub and tree routing templates. We give empirical results based on a heuristic for the corresponding new class of robust network design (RND) problems.

We define indicators which quantify the extent or shape of the traffic matrices spanned by the capped hose model. Specifically, we define two “strength” vectors; one for marginals and one for peak bounds. For instance, a large marginal strength value for i , indicates the relative importance of $U(i)$, i.e., how constraining it is in this capped hose model. We show that these metrics could be used as a predictor for which traffic scenarios favour a shortest path routing (SP), and which favour HH routings.

Another significant outcome from our empirical analysis is that we often find that HH is superior in network design cost (with link and node costs) to both SP and hub routing (HUB). This lends further support to the use of HH oblivious routing

in the future.

A. Summary of Main Contributions

- 1) We systematically study the capped hose model as a generalization of the fixed-demand and the unconstrained hose models. This allows tapping into a whole new range of practically relevant traffic scenarios.
- 2) We initiate a formal study of a class of RND problems based on oblivious hierarchical hub routing templates (HH), a generalization of tree and single-hub routing used by optimal virtual private networks (VPNs). We provide theoretical examples showing that SP is sometimes cheaper than HH by a factor of $\Omega(\log n)$, and that HH may be cheaper than SP by factors as large as n , where n is the number of nodes in the network. Recent work [6] has expanded the theory further.
- 3) We perform an empirical study (based on both randomly generated and real traffic data) on the space of new capped hose traffic polytopes.
- 4) Our initial empirical findings show that for most instances, HH is better than single-hub VPN, suggesting that multi-hubbing at different layers can be important for network cost savings. Empirical results also suggest that for many instances HH is significantly better than SP.
- 5) We provide some initial understanding of which traffic scenarios favour designs based on SP-like routing, and which are more HH-like. This is achieved by introducing two metrics that quantify the strengths of the marginals and peak demands in the specification of a collection of traffic matrices. These indicators could be used as a predictor for which of the two routing templates, SP and HH, is better. This is a potentially significant tool for carriers to cope with the design problem with unknown and changing demands.

II. TRAFFIC ENGINEERING FOR CORE NETWORKS

A. Oblivious Routing and Routing Templates

Apart from the *traffic modelling* complexities mentioned above, which form the basis for network infrastructure planning and design, there is a second *operational* complication arising in any network with dynamically changing traffic patterns: how should traffic be routed on an installed infrastructure? Since centralized or distributed control planes may not have the speed and flexibility to adapt dynamically to rapidly varying traffic patterns without the risk of inducing detrimental oscillations [7], the vast majority of today's networks are designed based on the principle of *oblivious routing*, usually based on SP, both on the circuit and on the packet layer.

Informally, Valiant [8] defined oblivious routing as “the route taken by each packet be determined entirely by itself. The other packets can only influence the rate at which the route is traversed.” Formally, this amounts to specifying a *template*: for each possible communicating pair of nodes i, j ,

one specifies a path P_{ij} .¹ Traffic from node i to j is always routed on P_{ij} independent of any other demands or congestion in the network. Oblivious routing is attractive because it enables strictly local routing decisions. In contrast, stable non-oblivious routing would require a global network control plane performing path optimization on a time scale much faster than the congestion and traffic variation dynamics, which is impractical in most scenarios.

For the *fixed-demand traffic model*, a minimum cost design is achieved using an oblivious SP routing template. Using SP for hose traffic, however, may result in significant over-provisioning of network resources (if blocking is to be avoided)[4]. On the other hand, it was proved in [9] that the optimal design (in the undirected setting) for the *hose model* is always induced by a tree routing template (i.e. there exists a network-specific tree, and the template is to use the path P_{ij} in that tree between node i, j). In [2], [3] a simple polynomial-time algorithm was given to produce the optimal tree template. Their analysis shows that the resulting network (called a virtual private network - VPN) actually has enough capacity to support *hub routing* (HUB) within the tree. That is, there is a hub node h and enough capacity in the tree, for each node to reserve a private circuit of size $U(i)$ to h . We use HUB to denote such routing templates.

B. The Capped Hose Model

Suppose we are given a capped hose model with nonnegative data $U(i), i \in [n]$ and $U(i, j) : i, j \in [n]$. It is reasonable to assume that $U(i, j) \leq \min(U(i), U(j))$ since otherwise we could simply lower its value to this upper bound with no effect on the space of matrices. Hence we define the *relevance interval* for $U(i, j)$ as

$$[0, \min(U(i), U(j))]$$

and assume all bounds lie in their relevance interval. Note that this implies that for each i , $U(i) \geq \max_j(U(i, j))$.

Note next that if $U(i) > \sum_j U(i, j)$ then the hose constraint for i has no impact on the space of capped hose matrices. We thus define the *relevance interval* for $U(i)$ to be:

$$[\max_j U(i, j), \sum_j U(i, j)].$$

We now see that for all data involved, the closer they are to the upper limit of their relevance interval, the less constraining they are on the resulting class of capped hose demand matrices. We use this intuition to now define our strength metrics.

C. Marginal and Peak Demand Strength Metrics

For any capped hose model, we propose two vector measures π (for peak) and μ (for marginal) to indicate the relevance of the different traffic measurements in defining a given collection of traffic patterns. For example, if $\mu(i)$ is large, it means that the marginal $U(i)$ has a significant role. On the other hand if $\pi(i, j)$ is large, it means that the peak

¹Valiant actually considered a randomized version where the flows could be fractional.

demand $U(i, j)$ has a more significant role in determining a cost efficient routing template. One aspect of this study is to examine whether these π and μ measures can be used to classify traffic instances as being more favourable for a SP or hub-like routing template.

Given marginals and peak demands (in their relevance intervals), we let the *strength* of node i 's marginal be:

$$\mu(i) = \frac{\sum_j U(i, j) - U(i)}{\sum_j U(i, j) - \max_j U(i, j)}.$$

Similarly, the *strength* of the peak demand between two nodes i, j is:

$$\pi(i, j) = \frac{\min(U(i), U(j)) - U(i, j)}{\min(U(i), U(j))}.$$

These metrics range from 0 (weak) to 1 (strong), and we claim that they are related to the cost optimality of hub routing and shortest paths templates, respectively.

We thus define our strength vectors $\mu \in \mathbb{R}^n$ for marginals and $\pi \in \mathbb{R}^{n \times n}$ for peak demands, with $\mu_i = \mu(i)$ and $\pi_{i,j} = \pi(i, j)$ respectively. A capped hose instance that is hose-like “should” have relatively high marginal strengths, and if it is peak-demand-like it should have high peak demand strengths. Based on this intuition, we attempt to classify a capped hose instance by the Euclidean norm of these two vectors (see Figure 5 for a preview).

D. Hierarchical Hub Routing and Circuits

It is well-known that designing a network for hose matrices is quantitatively similar (e.g., up to a factor 2 in capacity) to designing for a single “uniform multiflow” instance. Since uniform multiflow instances can be viewed as a convex combination of hub routings ([8], [4]), it seems natural that one of these hubs would deliver a good overall network design in terms of link capacity cost. However, in the capped hose model, we may “punch holes” into the space of demand matrices by letting some $U(i, j)$ be much smaller than others. This allows for the existence of certain regions within which there is much denser traffic than between such regions. Since these regions have their own capacity sharing benefits, one should no longer expect a single hub, but multiple hubs, each serving its own region of dense traffic.

This forms one motivation for our choice of *hierarchical hub routing templates* (HH). Each HH template is induced by an associated *hub tree*. If the hub tree is a star, it corresponds to standard hub routing. If the hub tree has more layers, its internal nodes represent possible hub nodes for different subnetworks (these subnets are necessarily nested due to the tree structure). A detailed description is given in Section IV.

Our work was partly motivated by work in [4], where Selective Randomized Load Balancing (SRLB) is used to design minimum-*cost* networks. They showed that networks whose design is based on oblivious routing techniques (and specifically SRLB) can be ideal to capture cost savings in IP networks. This is because there exist optimal designs reserving capacity on long paths, where one may employ high-capacity optical circuits. These “long circuits” partially avoid expensive

IP processing and equipment at every internal node. (Their empirical study incorporated IP router, optical switching, and fiber costs.) In this paper, instead of a design based on hub routing to a small number of hubs, we consider more general HH routing based on hub trees (as deployed in [5]). Once again, however, if $e = uv$ is an edge of a hub tree we reserve a dedicated circuit between $\eta(u), \eta(v)$ (where η identifies where u, v are mapped to in the network).

E. The Robustness Paradigm

We are given an undirected network topology with per-unit costs of reserving bandwidth. In addition we are given a space of demand matrices (in our case, the capped hose model) which the network must support via oblivious routing. This space, or “traffic model”, is meant to capture the time series of all possible demand matrices which the network may encounter. The goal is to find a routing template which minimizes the overall cost of reserved capacity needed to support all demands in the traffic space. The formal definition of robust network design (RND) appears in Section III.

Due to the optimality arguments of various routing templates for the traffic models under consideration, we focus on RND restricted to the routing templates SP, HUB, TR (tree routing) and HH. For a class of templates X , we denote by RND_X , the RND problem where one must restrict to the corresponding class of templates (RND is the unconstrained problem). While RND_{SP} , RND_{TR} and RND_{HUB} have been well-studied, we initiate a formal analysis of RND_{HH} in Section IV. We also propose a heuristic for this version in Section V.

F. Related Work

Oblivious routing approaches to network optimization have been used in many different contexts from switching (e.g., [10], [11], [12]) to overlay networks ([13], [14]), or fundamental tradeoffs in distributed computing [8], [15] to name a few. In each case mentioned above, the primary performance measure is network congestion (or its dual problem throughput). In fact, one could summarize the early work by Valiant, Borodin and Hopcroft as saying that randomization is necessary and sufficient for oblivious routing to give $\mathcal{O}(\log n)$ congestion in many packet network topologies.

The present work's focus is not on congestion, but on total link capacity cost. This falls in the general space of RND problems, which includes the Steiner tree and VPN problems as special cases. Exact or constant-approximation polytime algorithms for several other important traffic models have also been designed, e.g., the so-called symmetric and asymmetric hose models [16], [5]. It has been shown however, that it is NP-hard to approximate RND in undirected graphs within polylogarithmic factors for general traffic polytopes [5].

Our work also extends the long stream of work on designing VPNs [1], [3], [9], [17], [18]. Previous work has focused on provisioning for a VPN based on either the hose or the fixed-demand models (c.f. [19]). Designing for more general traffic models in this context has not received much attention. We

take an intermediate step by examining the capped hose model which to the best of our knowledge has not been studied.

With some thought, the new class of RND_{HH} problems is almost equivalent to solving RND but only using tree routing in the metric completion of the graph. The latter approach is exactly the idea used by Anupam Gupta (c.f. a long version of [20] contains full details) to show an $\mathcal{O}(\log n)$ -approximation for general RND via metric tree embeddings.

III. THE ROBUST NETWORK DESIGN (RND) MODEL

To understand the hierarchical hub routing problem we first introduce the robust network design paradigm in full generality.

A. Formal Definition

An instance of RND consists of a *network topology* (undirected graph) $G = (V, E)$, as well as per-unit costs $c(e)$ of bandwidth on each link $e \in E$ (the edge costs may be implicit if G is weighted). In addition, we are given a convex region $\mathcal{D} \subseteq \mathbb{R}^{V \times V}$. The region² \mathcal{D} represents the *traffic model* or *demand universe*, that is, the set of demands which have to be supported. In other words, any matrix $D \in \mathcal{D}$ must be routable in the capacity we buy on G . The cost of the network increases as this space grows.

One may define several versions of RND depending on assumptions on the routing model. Most common in practice (and the focus of this paper) is the *oblivious routing* model described above.

Our main objective in the context of RND is then to compute an oblivious routing template in a specific class of templates which leads to a link capacity vector cap_e whose overall cost is minimized. We use $\text{RND}_X(G, c, \mathcal{D})$ to denote the cost of an optimal solution

Given a fixed routing template $\mathcal{P} = (P_{ij} : \forall i, j)$, the cost of supporting all demands in the demand polytope \mathcal{P} can be computed in a straightforward fashion. For each edge $e \in E(G)$, we compute a “worst demand” matrix in \mathcal{P} in terms of routing on e (using the routing \mathcal{P}). This gives rise to the following sub-optimization:

$$\text{cap}_e(\mathcal{P}, \mathcal{D}) := \max_{D \in \mathcal{D}} \sum_{\substack{i, j \in V: \\ e \in P_{ij}}} D_{ij}. \quad (1)$$

RND then asks for a template \mathcal{P} that minimizes the cost of the overall required capacity:

$$\text{RND}_X(G, c, \mathcal{D}) = \min_{\mathcal{P} \in \mathcal{X}} \sum_{e \in E} c(e) \text{cap}_e(\mathcal{P}, \mathcal{D}).$$

B. Hierarchical Hub Routing Templates (HH)

We now define hierarchical hub routing templates HH [5], a generalization of (single-hub) routing. Such oblivious routing templates are derived from a tree T whose leaves consist of the terminal nodes, that is, $V(G)$ without loss of generality. We call this the *hub tree*. The internal nodes are “abstract”

²We always assume that \mathcal{D} is itself well-described in the sense that we can solve its separation problem efficiently, e.g. in polytime.

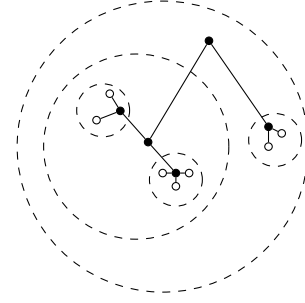


Fig. 1. A hub tree. A “region” of terminals is represented by a dashed circle. Internal black nodes of the tree are the ones that will be mapped to hubs in the network topology. White leaf nodes are terminals.

in the sense that they represent nodes (yet to be identified in G) which will act as hubs for certain subsets of terminal nodes. For instance, for any non-leaf node $v \in T$, let T_v be the subtree rooted at v dangling below it. The interpretation is that the leaves L_v in T_v represent a region of terminal nodes which should have a dedicated hub (this hub will correspond to v). The family of regions arising from a tree obviously forms a nested family (Figure 1).

Obtaining an oblivious routing template from the HH class is a two step process. First, we need to pick a hub tree, and then map its internal node to physical nodes. In this second stage, we find a map from each internal node v to a real node in G to act as a hub. Such a *hub map* $\eta : V(T) \rightarrow V(G)$ must satisfy $\eta(v) = v$ for each leaf of T .

Given such a hub mapping, there is an *induced* oblivious routing template defined as follows: for each pair of leaf nodes i, j in T , let $i = v_0, v_1, v_2, \dots, v_l = j$ be the unique ij path in T . For each $q = 0, 1, \dots, l-1$, let P_q denote a shortest $\eta(v_q)\eta(v_{q+1})$ path in G . Then we define P_{ij} to be the concatenation of paths P_0, P_1, \dots, P_{l-1} (resulting in a possibly non-simple path). Naturally, the special case of HUB routings corresponds to taking T equal to a star.

Note that each hub tree thus gives rise to many possible routing templates, one for each mapping η (analogous to the specific choice of hub node in hub routing). Call this the class of hierarchical hub templates (HH). Note that, in some scenarios, there are single-path oblivious routing templates that are unachievable by hierarchical hub templates. For instance in a complete graph on four nodes, with unit weight edges, there is no hierarchical hub template which corresponds to the shortest path template (each P_{ij} consists of the edge ij).

HH routing also responds to one practical impediment of hub routing architectures, pointed out in [4]. Namely, network providers are generally opposed to having all traffic routed via a single hub (or a small cluster of hubs) in the center of a large network. In particular, traffic within some local region should be handled by hubs within their own “jurisdiction”. Hub trees potentially give a mechanism for specifying these local jurisdictions.

IV. THE OPTIMIZATION PROBLEM RND_{HH}

There are several layers to the optimization problem RND_{HH} . First, we must find a hub tree T - this is the core decision for which we provide a heuristic algorithm in Section V. Second, given a hub tree, we must map it to the network graph to obtain a valid routing template. An efficient solution to this mapping problem is known for T -topes³ [5]; we show that this can be invoked to solve the problem for general polytopes in Section IV-A. Finally, given a template, we must ultimately solve for capacities it induces, as we discuss in Equation (2) below.

In Section IV-B we discuss several classical problems which arise as special cases of RND_{HH} . Finally, we provide some theoretical bounds between RND_{HH} and RND_{SP} in Section IV-C.

A. Turning a Hub Tree into a Solution

Consider the case where a hub tree T (with leaf set $V(G)$) is prescribed for the network G . This case is of independent practical interest when a network provider wishes to self-identify a regional clustering within their network. In other words, they provide the hub tree, and it only remains to identify which nodes to act as the hubs.

From the definition of HH templates, once we know the mapping $\eta : V(T) \rightarrow V(G)$ of hub nodes to network nodes, we have determined our HH routing template. In general RND we would then calculate the capacities required via (1). In RND_{HH} we proceed slightly differently.

As discussed in Section II-D, our strategy is to have a dedicated circuit between $\eta(w), \eta(v)$ for each edge $e = vw \in E(T)$. Hence for each edge of T , we must determine the largest demand using this circuit (under the HH template). This is analyzed as follows. We let $\text{fund}(e)$ denote the ‘‘fundamental cut’’ for $e \in T$; that is, it consists of all i, j pairs whose path in T uses the edge e . These are precisely the demands which ultimately route on the circuit between $\eta(w), \eta(v)$ in G . Hence if there are demand matrices that route d units of flow on e in T , then we need to reserve d units of capacity on the path between $\eta(w), \eta(v)$ in G . We now compute the size of the circuit as follows, where \mathcal{D} is the demand polytope.

$$u_{\mathcal{D}, T}(e) := \max_{D \in \mathcal{D}} \sum_{i, j \in \text{fund}(e)} D_{ij}. \quad (2)$$

We hence see that a tree T plus the polytope \mathcal{D} induce some maximum circuit capacities for the edges in T ⁴. Given a hub mapping η , our total network cost is then obtained as follows: for each $e = vw \in T$, reserve $u_{\mathcal{D}, T}(e)$ on a shortest path between $\eta(u), \eta(v)$ in G . This is because we want these circuits to be operational, independently of one another and of current traffic conditions. The problem of mapping hubs from

³A T -tope is a polytope of demands, denoted by $\mathcal{H}_{T, b}$, which are routable on some tree T with edge capacities b . This class generalizes hose polytopes, which can be viewed as demands routable on a star.

⁴It is worth noting that $u_{\mathcal{D}, T}$ is defined to be the smallest capacity vector u such that $\mathcal{D} \subseteq \mathcal{H}_{T, u}$. In other words, designing a network to support the T -tope $\mathcal{H}_{T, u_{\mathcal{D}, T}}$ also supports \mathcal{D} .

T to G becomes: given a tree T with edge capacities u , find a valid mapping $\eta : V(T) \rightarrow V(G)$ which minimizes

$$c_{\text{HH}}(T) = \sum_{e=uv \in T} u_{\mathcal{D}, T}(e) \text{dist}_G(\eta(v)\eta(w)) \quad (3)$$

where $\text{dist}_G(s, t)$ is the length of a shortest st -path in G with edge costs c . The problem of determining an optimal η for capacitated trees is precisely the *hub placement problem* considered in [5]. They give a (polytime) dynamic programming algorithm to find a valid mapping $\eta : V(T) \rightarrow V(G)$ which minimizes

As pointed out in [6], minimizing (3) is slightly different from computing η so as to minimize capacity cost based on (1). Indeed we could define two versions of our problem depending on which formula we use for computing link capacities: $\text{RND}_{\text{HH}}^{(1)} \leq \text{RND}_{\text{HH}}^{(3)}$. For the hose polytope the two quantities are the same; this follows from the fact that the optimal tree template is induced by a hub routing [2], [3].

B. RND_{HH} and Other Algorithmic Problems

RND_{HH} contains well-studied algorithmic problems as special cases. First, if $\mathcal{D} = \mathcal{H}_U$ is the class of hose matrices, then, as mentioned earlier, an optimal solution to RND is induced by a hub routing template, which is in turn induced by a star hub tree. Hence in this case $\text{RND}_{\text{HH}} = \text{RND}_{\text{HUB}} = \text{RND}$. In [5] it is asked whether this also holds for any T -tope (Generalized VPN Conjecture).

Second, consider the case where \mathcal{D} consists of a single demand matrix D . Obviously, the optimal solution is induced by a SP template. RND_{HH} however, asks for an optimal HH template. This is a classical problem in combinatorial optimization called the *minimum communication cost tree* (MCT). Find some tree (not necessarily in G), containing $V(G)$, such that if we route the demands D on T , the overall routing cost is minimized.⁵ Being allowed to use a hub tree for the routing, as opposed to a subtree of G , essentially means that we are restricting to MCT with metric costs⁶. For the special case where G is itself a complete graph with unit-cost edges, Hu [22] showed that an optimal solution is induced by a so-called Gomory-Hu tree on the complete graph with edge capacities D_{ij} .⁷

As we now see, RND_{HH} may incur a logarithmic increase over RND , and in fact RND_{SP} . On the positive side, it is known that in general there is a tree in the metric completion of G whose ‘‘distortion’’ is at most $\mathcal{O}(\log n)$ [23]. Gupta has shown (see [24] for full details of his argument) how to obtain an $\mathcal{O}(\log n)$ approximation for RND_{TR} via metric embeddings. This implies $\text{RND}_{\text{HH}} = \mathcal{O}(\log n)\text{RND}$ as can be seen by noting that RND_{HH} costs no more than a best tree routing on T in the metric completion. To argue this, define a hub tree T' by hanging a leaf edge vv' from each node of T .

⁵This is directly related in turn to the average stretch tree problem [21].

⁶Costs are *metric* if they are induced by a cost function that is a metric.

⁷In general, the Gomory-Hu tree for a capacitated graph need not be subtree of the graph; hence the requirement that G is complete.

Then consider the mapping where each v, v' maps to $v \in G$. This establishes:

Fact 1. For any instance of RND,

$$\text{RND}_{\text{HH}} \leq \text{RND}_{\text{TR}}.$$

We mention, however, that there may yet be polytime $\mathcal{O}(1)$ -approximations for MCT and indeed RND_{HH} . For the purposes of our study, we use a heuristic algorithm instead of metric embeddings to find our best hub trees. This is outlined in Section V.

C. Basic Results for RND_{HH}

In this section we show that RND_{SP} and RND_{HH} are incomparable; depending on the instance, one may be significantly cheaper than the other. This motivates our main goal of classifying the demand spaces according to which template is better.

We now establish a family of RND instances where restricting to HH templates (i.e. RND_{HH}) incurs a penalty, and in fact a worse penalty than restricting to SP templates (i.e. RND_{SP}).

Theorem 1. *There is a sequence of (unit-cost) graphs $\{G_n\}_{n \in \mathbb{N}}$ on n nodes and demand polytopes $\{\mathcal{D}_n\}_{n \in \mathbb{N}}$ such that for these instances*

$$\text{RND}_{\text{HH}}(G_n) = \Omega(\log n \text{RND}_{\text{SP}}(G_n)).$$

Proof. We describe $G = G_n$ and $\mathcal{D} = \mathcal{D}_n$ for a fixed n . The network graph G consists of a d -regular, α -expander graph on n vertices, where $d = \mathcal{O}(1)$ and $\alpha = \Omega(1)$ (see [25] for existence proof). Directly we have that $|E(G)| = \mathcal{O}(n)$. Moreover, all edges have unit cost.

The demand polytope \mathcal{D} consists of a single matrix: one unit of demand between the endpoints of any edge in G . The RND_{SP} solution then consists of the network itself (route each demand on its own edge). Its cost is $\mathcal{O}(n)$, the number of edges in the graph. Let T be any hub tree. We show that the cost of any HH solution induced by T is $\Omega(n \log n)$. Hence, RND_{HH} is $\Omega(n \log n)$.

Let $c \in V(T)$ be a *center* of the tree, that is each component of $T \setminus \{c\}$ has at most $\frac{n}{2}$ leaves (it is well-known, and easy to see that every tree has a center). We call an edge $e \in E(G)$ *separated by c in T* if its endpoints lie in different components of $T \setminus \{c\}$. We let $E_c \subseteq E(G)$ be the set of all edges separated by c in T . Let C_i be one of the components in $T \setminus \{c\}$ and let $\ell(C_i)$ be the leaves of C_i . We know that $|\ell(C_i)| \leq \frac{n}{2}$. Hence, since G is an α -expander, there is at least $\alpha|\ell(C_i)|$ edges going out of C_i . So we have that $|E_c|$ (or the total number of edges between the C_i 's) is at least

$$|E_c| \geq \frac{1}{2} \sum_i \alpha |\ell(C_i)| = \frac{\alpha}{2} \sum_i |\ell(C_i)| = \frac{\alpha n}{2}$$

as $\sum_i |\ell(C_i)| = n$ (the leaves of T consist of the nodes of G).

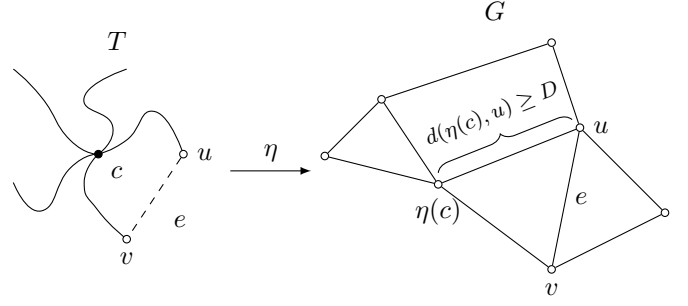


Fig. 2. A bad edge $e \in E(G)$. The edge e is separated by c in T (as per the left portion of the figure), and one of its endpoint u is not close to c (as per the right portion of the figure).

Let $\eta : V(T) \rightarrow V(G)$ be the optimal hub map for T . Call a node $v \in V(G)$ *close to c* if

$$\text{dist}_G(\eta(c), \eta(v) = v) < D = \log_d \left(\frac{d-1}{d} \min\left\{\frac{1}{2}, \frac{\alpha}{2d}\right\} n \right).$$

Let V_c be the set of close vertices in G . Since G is a d -regular graph, its size is

$$|V_c| = \sum_{i=0}^{\lfloor D \rfloor} d^i = \frac{d^{\lfloor D \rfloor + 1} - 1}{d - 1} \leq \frac{d}{d - 1} d^D = \min\left\{\frac{1}{2}, \frac{\alpha}{2d}\right\} n$$

We now merge the two concepts and say that an edge $e = xy \in E_c$ that is separated by c in T is *good* if both $x, y \in V_c$, i.e. x, y are close to c - see Figure 2. Conversely, an edge $e = xy \in E_c$ is *bad* if one of its endpoints is not close to c in T . Let B be the set of bad edges.

By the d -regularity of G , a bound on the number of bad edges is given by

$$|B| \geq |E_c| - \frac{d|V_c|}{2} \geq \frac{\alpha n}{2} - \frac{d\alpha n}{4d} \geq \frac{\alpha}{4} n.$$

Since the bad edges are unit demands that must be simultaneously routable in T , the “non-close” endpoint of these demand edges must route through the image $\eta(c)$ of c , imposing a cost of at least $\log_d(n)$. We have $\Omega(n)$ bad demands, each incurring a cost of $\Omega(\log n)$, and thus the cost of the optimal HIERARCHICAL HUBBING induced by T routing is $\Omega(n \log n)$. \square

The next result shows that RND_{TR} may be much cheaper than RND_{SP} . Hence, we can get an unbounded gap between RND_{SP} and RND_{TR} . By Fact 1, this implies that RND_{HH} may also be much cheaper than RND_{SP} .

Theorem 2. *There is a sequence of weighted graphs $\{G_n\}_{n \in \mathbb{N}}$ and demand polytopes $\{\mathcal{D}_n\}_{n \in \mathbb{N}}$ such that*

$$\text{RND}_{\text{SP}}(G_n) = \Omega(n^2)$$

and

$$\text{RND}_{\text{TR}}(G_n) = \mathcal{O}(1).$$

Proof. The network $G = (V, E)$ (actually G_{2n+2}) consists of two stars centered at nodes a, b connected by an edge. The leaves are v_i, v'_i with the following edges:

$$E = \{av_i : 1 \leq i \leq n\} \cup \\ \{bv'_i : 1 \leq i \leq n\} \cup \\ \{v_i v'_j : 1 \leq i, j \leq n\} \cup \\ \{ab\}.$$

The edges are weighted as follows: the bridge edge ab has weight one, the star edges av_i or bv'_i for $1 \leq i \leq n$ have weight $\frac{1}{2n}$ and the edges $v_i v'_j$ for $1 \leq i, j \leq n$ have weight $1 - \frac{1}{n}$.

The (symmetric) demand polytope \mathcal{D} consists of convex combinations of unit demands between v_i and v'_j . That is, for each i, j let $D^{i,j}$ be the matrix with $D_{v_i v'_j} = D_{v'_j v_i} = 1$ and all other entries 0. Then $\mathcal{D} = \text{conv}(D^{i,j})$.

The optimal shortest path template in the network consists of routing the demand between v_i and v'_j through the edge of cost $1 - \frac{1}{n}$ connecting v_i and v'_j . This template induces a cost of $\text{RND}_{\text{SP}} = n^2(1 - \frac{1}{n}) = n^2 - n = \Omega(n^2)$.

Now consider the tree template induces by the union of the two stars. In order to support \mathcal{D} , the sub-optimization problem only reserves one unit of capacity on the bridge edge ab , as well as one unit on all star edges. Hence, this capacity cost is $\mathcal{O}(1)$, and is an upper bound for the best tree solution. \square

V. A HEURISTIC FOR CHOOSING A HUB TREE

We now develop a heuristic for RND_{HH} . It is tailored for the capped hose model, but could be extended to handle any traffic polytope. The high-level intuition is to group nodes so that traffic within a group is large compared to traffic leaving the group. This suggests the need for a hub to handle the group's local traffic. It is straightforward to show that there is always an optimal hub tree for RND_{HH} where the tree is binary (we defer the details to the end of this section). Hence our algorithm's focus is to find a "good" binary hub tree.

To construct a binary tree from the terminal nodes V , one could either proceed by a top-down approach (recursively splitting groups of nodes in two) or a bottom-up approach (recursively merging nodes together). The former has the flavour of repeatedly solving **SPARSEST CUT** problems. This problem is **APX**-hard and the current best approximation factors are polylogarithmic [26], [27]. Instead, we follow the bottom-up approach. We are thus seeking to repeatedly merge pairs of node sets which share a lot of traffic with each other. The key to this calculation is a sparsest cut-like measure which we introduce next. This is then used in a *Binary Tree Sparsest Merging (BTSM)* algorithm.

A SPARSITY MEASURE

We devise a sparsity measure that seeks to maximize the traffic between two groups, compared to the traffic sent outside the merged groups. This measure is very similar to the capacity on a (fundamental) cut explained in Section III-A - see Equation (1).

In the setting of the capped hose model, the maximum possible demand between a pair of disjoint sets of nodes $A, B \subseteq V$ is given by

$$u^*(A, B) = \text{maximize } \sum_{i \in A, j \in B} D_{ij} \\ \text{subject to } D_{ij} \leq U_{ij} \quad \text{for all } i, j \\ \sum_j D_{ij} \leq U(i) \quad \text{for all } i \\ D_{ii} = 0 \quad \text{for all } i \\ D \geq 0$$

This problem can be solved efficiently, since it can be cast as a b -matching or max-flow problem. For instance, consider a bipartite graph H with bipartition A, B . Each edge ij has a capacity of U_{ij} and is oriented from A to B . We also add a source s with edges (s, i) for $i \in A$ with capacity $U(i)$. Similarly we add sink t with edges from B . A max st -flow in this graph gives precisely the value $u^*(A, B)$.

The *sparsity* of a pair of disjoint sets of nodes $A, B \subseteq V$ is then:

$$sc(A, B) = \frac{u^*(A \cup B, V \setminus (A \cup B))}{u^*(A, B)}.$$

One notes that it trades off the flow out of $A \cup B$ versus the flow between A, B .

THE SPARSEST MERGING ALGORITHM

Our proposed heuristic algorithm (see Algorithm 1) builds up a binary hub tree T , starting from a forest consisting of a set of singletons. At each step it has a forest and looks for a pair of rooted trees that has the minimum sparsity value. Specifically, it computes the sparsity measure for the leaf sets for such a pair of trees. The two subtrees are then merged, that is, a new root node is added to T and becomes the parent of the two previous rooted subtrees. When the forest becomes a single tree, we stop. Clearly this can be done in polytime.

Algorithm 1 Binary Tree with Sparsest Merging algorithm.

Require: A set of nodes nodes V with peak demands $U(i, j)$ for each pair of nodes $i, j \in V$ and marginals $U(i)$ for each node $i \in V$.

Ensure: A binary demand tree T .

```

 $T = (V, \emptyset)$ 
 $S \leftarrow V(T)$ 
while  $|S| > 1$  do
    Find  $i, j \in S$  that has minimum sparse cut value.
    Add a new node  $u$  to  $T$ .
    Connect  $u$  to  $i$  and  $j$ .
    Add  $u$  to  $S$ .
    Remove  $i$  and  $j$  from  $S$ 
end while
return  $T$ 

```

RESTRICTING TO BINARY TREES.

In this section, we show that there is always a binary hub tree that is optimal for RND_{HH} .

Let T be any hub tree that has at least one node u of degree at least four. Then it is possible to form a new hub tree T' with $c_{\text{HH}}(T') \leq c_{\text{HH}}(T)$, and such that the sum of the degrees of

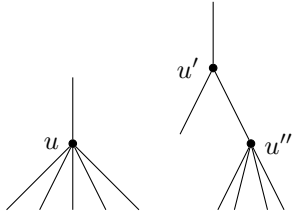


Fig. 3. Dividing internal nodes to get a binary tree.

nodes with degree greater than four is smaller in T' . Hence, repeating the process eventually reduces the number of degree at least four nodes to zero, and we end up with a binary tree. Here is the process in details.

Consider a node u of T with degree at least four. Replace u by two nodes u' and u'' such that u 's old parent is now connected to u' , and u' is connected to u 's first child as well as to u'' . Then u'' is connected to the remaining children of u (see Figure 3). Note that the sum of degrees of nodes with degree at least four is decreased by one, since u' has degree three and u'' has degree $\deg(u) - 1$.

The capacity on the edge between u 's parent and u' is equal to the capacity between u and its parent; the capacity between u' and u'' can be inferred using equation (2). All other capacities remain the same - the capacity between one of u 's children and its new parent u' or u'' is the same as the capacity between u and that child in T . Note that we only split u into two nodes and added enough capacity between those two nodes to route any feasible traffic matrix. We have the new capacitated T' can support exactly the same demand matrices as T . (The T' -tope and T -topes are the same.)

Finally, an optimal hub map of T can be implemented for T' as follows. One maps u' and u'' to the node that u is mapped to. Hence, T' can produce the same oblivious routing as T , so its optimal cost $c_{\text{HH}}(T')$ is at most $c_{\text{HH}}(T)$.

VI. EVALUATION STUDIES

We compare the network design costs for capped hose models using the routing strategies SP and HH. We solve the associated optimization problems (RND_{SP} , RND_{HH}) across many instances of the capped hose traffic model. For the purposes of this study, we implemented an exact method for SP, and employed our BTSM heuristic in the case of HH routing.

For the evaluation set-up, we use two carrier network topologies: the American backbone network Abilene (11 nodes, 14 edges), and the Australian telecom network Telstra [28] (104 nodes, 151 edges). We assume that per-unit link capacity costs are proportional to physical distances so we use this as our cost vector c for determining the best RND solution. Our traffic data is based both on randomly generated traffic instances within the capped hose model constraints, as well as on a traffic scenario based on real data.

For the simulated traffic scenarios, for each network we randomly generate many traffic instances (each corresponding to a capped hose polytope) on which to solve RND. That is, each instance arises from some collection of peak demands and marginals. Our random instances are generated starting from actual population statistics (see Section VI-A). In addition on Abilene, we use real data based on previous work from [29] which gives a time series of point-to-point demand measurements that can be used to create capped hose models (described later).

Our results show that HH templates are often the most cost-effective. That is, RND_{HH} is very often less than both RND_{SP} , and RND_{HUB} . In particular, this means that by adding peak capacities $U(i, j)$ into the hose model, the single-hub tree-routing template is no longer cost-effective. We need additional hubs for a cost-effective network design. This is strong evidence for the use of HH routing.

A. Generating Traffic

To generate our traffic demand matrices, namely marginals $U(i)$ and peak demands $U(i, j)$, we try to mimic how traffic is naturally generated in today's core networks. It has been widely accepted that core traffic is stochastic in nature ("bursty") and that peak demand is much larger than average demand. This change in traffic patterns presents a moving target for service providers. The uncertainty is due to novel content-based network applications combined with factors such as data center consolidations and content mobility (content migrating from location to location based on where they are consumed). Therefore in our evaluation scenario, we first assume peak demands, and then impose marginals based on the equipment choice at each node (capacity and number of ports at the ingress and egress nodes). Our process is anchored by first fixing the peak demand between every pair of node locations based on their population [30]. This gravity model takes into account the population size of the two sites as well as their distance. Since larger sites attract more traffic and closer proximity can lead to greater attraction, the gravity model incorporates these two features. The relative traffic between two places is determined by multiplying the population of one city by the population of the other city, and then dividing the product by the distance between the two cities. In our case, the peak demand between two cities i, j is

$$U(i, j) = \alpha(i, j) \text{Population}(i) \text{Population}(j) \quad (4)$$

where $\alpha(i, j) = 1/\text{distance}(i, j)^2$ and $\text{distance}(i, j)$ is the geographical distance between node i and node j .

As datacenter based traffic dominates over the Internet, significant research activity is underway to learn the characteristics of this traffic. The routing of data-center bound traffic within the ISP network is typically controlled by BGP policies. However, as noted in [31], we see that the end-to-end traffic matrix showing how the total traffic to YouTube video-servers is divided among different data centers follows the gravity model. Thus, the proposed HH routing template could also

be extended for routing within data center network with the appropriate for hub placement.

GENERATING MARGINALS

To generate marginals, we propose sampling within the individual relevance intervals, as discussed in Section II-B. We start by discretizing $U(i)$'s relevance interval into $s > 0$ steps, for each of the n nodes i . Then, for each value $\sigma_i \in \{0, 1, 2, \dots, s\}$ we get a $U(i)$ within its relevance intervals. Formally:

$$U(i) = \max_j U(i, j) + \frac{\sigma_i}{s} \left(\sum_j U(i, j) - \max_j U(i, j) \right).$$

Thus each vector $\sigma \in \{0, 1, \dots, s\}^n$ yields an instantiation of the marginals. For fixed s and n , this gives $(s+1)^n$ possible data points, yielding all possible marginals within their relevance interval up to a precision based on the coarseness of the discretization (size of s).

It is impractical to generate all $(s+1)^n$ points for large n and s . To bypass this, we use a smaller value k in place of n to generate $(s+1)^k$ points. We then sample uniformly from these σ to assign the values for each node, i.e. we mimic the distribution of the short vector σ of length k to create our longer marginal vector of length n . So we get

$$U(i) = \max_j U(i, j) + \frac{\sigma_{U[k]}}{s} \left(\sum_j U(i, j) - \max_j U(i, j) \right). \quad (5)$$

where $\sigma \in \{0, 1, \dots, s\}^k$ and $U[k]$ is chosen uniformly at random from $\{1, 2, \dots, k\}$.

B. Time-Series Traffic Matrix Data

For the Abilene network, we use the estimation of point-to-point demands from the work of [29] to create a realistic data point. The data consists of 24 weeks worth of traffic matrices sampled each five minutes. In other words, we get a sequence $\{D_{i,j}^t\}_{t=1}^T$ ($T = 48384$) of demands from i to j for all nodes i, j in the network. We transform this data into marginals and peak demands for a capped hose model as follows. The peak demand between two nodes i, j is by definition $U(i, j) = \max_t D_{i,j}^t$. Similarly, node i 's marginal is the (minimum) capacity required to route the maximum traffic out of that node at a given time step: $U(i) = \max_t \sum_j D_{i,j}^t$. We consider different sampling intervals, i.e. various T 's.

C. Computation of Link Costs

It is simple to compute the capacity $u(e)$ needed on each edge e for HH - we just add up the capacities from every hub tree edge $f = ij$ for which the $\eta(i)\eta(j)$ -path passes through e . For SP, we follow a similar sub-optimization as used for HH templates. Namely, we find the maximum value of $u(e) = \sum_{ij} w_{ij}(e) D_{ij}$ over all demand matrices D in our capped hose universe \mathcal{D} where $w_{ij}(e)$ is 1 if P_{ij} uses edge e , and is 0 otherwise. For any edge $e = st$, it is straightforward to see that the set of ij 's with $w_{ij}(e) = 1$ induces a bipartite graph $H = (V(G), \{ij : w_{ij}(e) = 1\})$ when the P_{ij} 's are shortest paths. Suppose H is not bipartite, then there is an odd cycle $C = i_1 i_2 \dots i_{2k+1}$. Thus, $w_{i_2 k+1 i_1} = 1$ and $w_{i_j i_{j+1}} = 1$ for

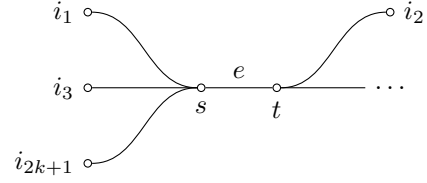


Fig. 4. Going from i_j to i_{j+1} , always through $e = st$ and along shortest paths.

all $1 \leq j \leq 2k+1$. Without loss of generality, the fact that P_{i_j} 's are shortest paths implies that i_j is closer to s for odd j 's and closer t for even j 's (see Figure 4). But then this means the path $P'_{i_{2k+1} i_1} = \text{SP}_G(i_{2k+1}, s) \cup \text{SP}_G(s, i_1)$ is strictly shorter than P_{i_{2k+1}, i_1} as it doesn't use e ; contradiction. Hence we may use a max flow routing algorithm to compute $u(e)$. The computation of link capacities can thus be done with a collection of all shortest paths $\{P_{i_j}\}$ generated using standard algorithms.

Algorithm 2 illustrates our experimental procedure.

Algorithm 2 Experimental procedure.

Require: Network graph $G = (V, E)$ (where edge costs correspond to the geographical distance between edge's endpoints) and parameters $s, k \in \mathbb{N}$.

for each pair of node cities $i, j \in V$ **do**

Fix point-to-point peak demand between i and j as per Equation (4).

end for

for each $\sigma \in (s+1)^k$ **do**

for each $i \in V$ **do**

Fix node i 's marginal as per Equation (5).

end for

This gives a capped hose model $c\mathcal{H}$ arising from marginals $U(i)$'s and peak demands $U(i, j)$'s.

Find marginal strength vector μ of $c\mathcal{H}$.

Find peak demand strength vector π of $c\mathcal{H}$.

Solve $\text{RND}_{\text{SP}}(G, c\mathcal{H})$ optimally.

Use our BTSM heuristic and Olver et al's algorithm [5] to approximately solve $\text{RND}_{\text{HH}}(G, c\mathcal{H})$ and get a solution of cost $\text{RND}'_{\text{HH}}(G, c\mathcal{H})$.

Report $\|\mu\|_2$, $\|\pi\|_2$ and $\frac{\text{RND}_{\text{SP}}(G, c\mathcal{H})}{\text{RND}'_{\text{HH}}(G, c\mathcal{H})}$.

end for

D. Results

In Figures 5 and 6, we plot the results of the low-cost routing templates for two different network topologies Abilene and Telstra⁸. Traffic matrix instances are plotted according to their marginal and peak demand strengths and coloured based on the ratio between the edge cost of the (optimal) SP routing

⁸A much smaller number of points were sampled to cover most of the spectrum due to the computational constraints caused by the size of the Telstra network.

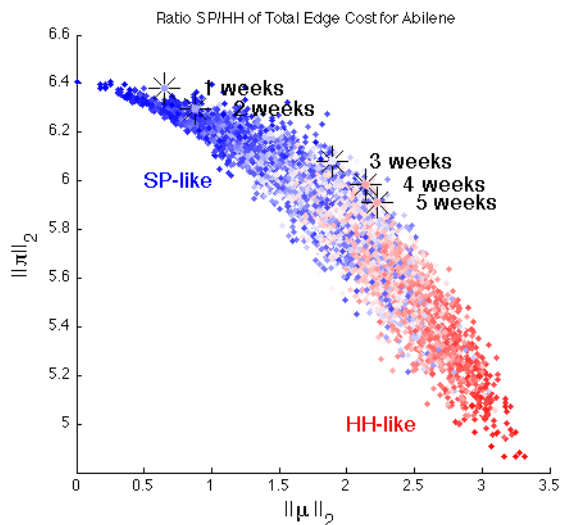


Fig. 5. The ratio between the edge cost of the (optimal) SP routing and the HH routing found with our heuristic plotted against the norms of the peak demand and marginal strength vectors for varying traffic, on the Abilene network. Data points from the time-series of traffic matrices added, with duration considered as label. *Better viewed in color.*

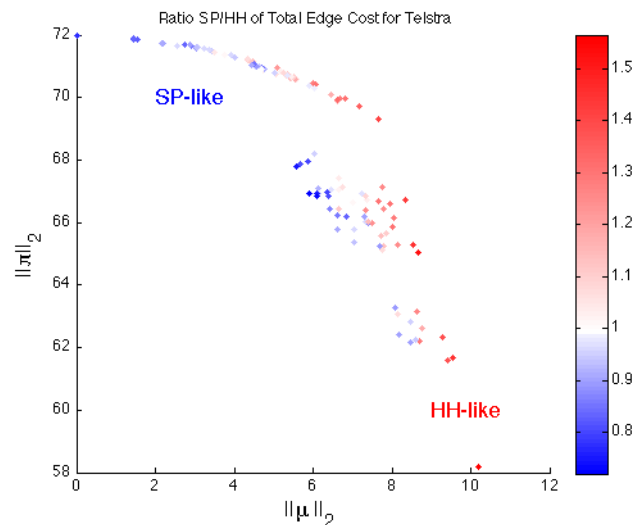


Fig. 6. The ratio between the edge cost of the (optimal) SP routing and the HH routing found with our heuristic plotted against the norms of the peak demand and marginal strength vectors for varying traffic, on the Telstra network. *Better viewed in color.*

template and the HH routing template. A blue point indicates that SP is better and a red point indicates that HH is better. Moreover, the intensity of the colouring is proportional to the proximity of the ratio to one. If the costs of SP and HH are very similar, the color is almost white, otherwise it is highly saturated

Intuitively, and as discussed in Section II-C, the x -axis corresponds to the affinity of the traffic to the hose model. A large μ value means more constraining marginals. Thus the optimal routing should primarily consider the marginals only. We expect hub routing to do much better. Similarly, the y -axis corresponds to the affinity of the traffic to the peak-demand model. Again, a large π value means more constraining peak demands, and thus we expect SP to do much better. Furthermore, there is a duality between the μ and π values: a demand universe can only have strong marginals or strong peak demands, not both. This explains the quarter circle “swoosh” shape of the plot.

VII. DISCUSSION

In [4], hub routing was used to compare the cost of two architectures: one based on IP routing and one employing circuits. Our work focuses on how the link and/or node costs vary with the choice of routing template. We now address some of the issues and our findings.

A. $\mu - \pi$ as indicators for the choice of a routing template

We see that there is a continuum of traffic scenarios as a function of the newly introduced μ and π measures. As such we propose that it is possible that the value of (μ, π) could serve as an important routing indicator since it accounts for the strengths of the marginals and peak demands. As such, knowing these values is useful information for determining the

appropriate routing template. This is an important observation because it also suggests a new sampling strategy for service providers faced with designing for changing demands. In addition to just using marginal demands, they can sample the demand space to determine the relative strengths of the peak and marginal demands. This may be used to choose an initial routing template and, if necessary, evolve the template to obtain/maintain cost-efficient routing. This further justifies the need to measure link level traffic, both peak and average demand, since it can feedback into the selection of an optimal routing template for a given network topology.

An important consideration is the sampling interval. The impact of the sampling interval can be clearly seen by the evolution of the real data points in Figures 5 and 7. Each such data point comes from a compilation of the real traffic matrices over a certain period of time, as labelled next to the points. As the interval increases from one week to four, we see the capped hose model shift toward a more hose-like universe, with stronger marginal (and consequently weaker peak demands). If samples are over a short time horizon, then the demand matrix does not change significantly and as such we are designing for a single demand. In this scenario, SP yields the cheapest routing template. However as the time horizon gets longer, the different demand matrices could be significantly different and in such scenarios the more flexible design of HH/VPN brings in the added cost benefits.

B. Single versus Multi Hub Routing Templates

An interesting observation of this study is that there are significant cost benefits for using a multi-hub routing template versus the single-hub templates which yield optimal VPNs. Table I shows the number of traffic instances where HH is more cost effective than a single hub VPN solution. It is

TABLE I
NUMBER OF INSTANCES WHERE HH OUTPERFORMS SP

Network	HH Optimal Multiple Hubs	HH Optimal Single Hub
Abilene (total number of instances: 6561)	3045	8
Telstra (total number of instances: 353)	246	0

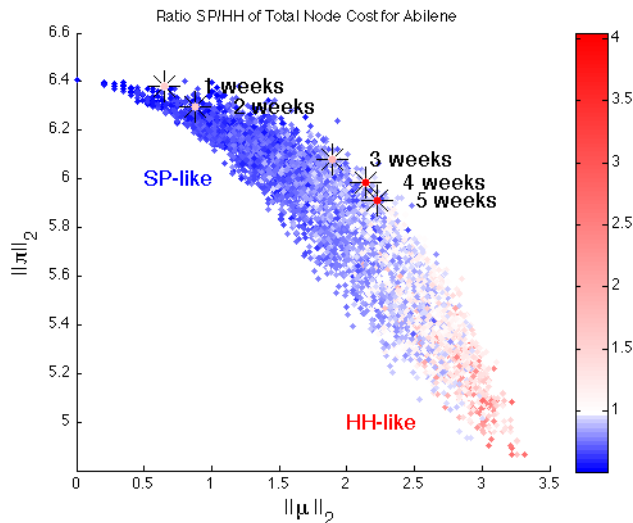


Fig. 7. The ratio between the node cost of the (optimal) SP routing and the HH routing found with our heuristic plotted against the norms of the peak demand and marginal strength vectors for varying traffic, and the Abilene network. Data points from the time-series of traffic matrices added, with duration considered as label. *Better viewed in color.*

straightforward to see that a best solution induced by any hub tree is always as good (in terms of total cost) as a single-hub routing. Indeed, mapping all internal nodes of a hub tree to a single network node is always a valid hub placement that yields the single-hub VPN-like routing. Since the hub placement algorithm [5] is optimal, it is guaranteed to find a hub placement that is as good as the best single-hub routing.

C. Routing Costs and Impact on Network Design

We evaluate the cost-effectiveness of the HH template against SP using both link and node costs. Link or edge capacity costs for HH are computed by solving a maximization problem over the universe of traffic matrices. For node costs we use the maximum possible link capacities that are incident on a particular node. The intuition behind using these maximum incident capacities is based on the hardware port requirements for routing/switching the total traffic through that node. However it is very rare for all incident links at the node to simultaneously carry their maximum allowable traffic rates.

From Figures 5 and 7 for link and node costs respectively, we see that HH shows significant cost savings for link costs since we take advantage of dense traffic sharing points through appropriate location of hub nodes. However, since the node costs are being evaluated using physical link capacities the gains are not as significant.

VIII. CONCLUSION

The capped hose model introduced in this paper shines light on the applicability of current routing template archetypes (HUB and SP) parametrized by the marginal and peak demands of the given traffic matrices. We have shown that using the capped hose model provides additional flexibility to network engineers for cost-optimal network design and shows the relevance of the newly introduced HH templates. In particular, this means that by adding peak capacities $U(i, j)$ into the hose model, the single-hub tree routing template may no longer be cost-effective. Moreover, the peak demand and marginal strengths presented here are a first attempt at choosing routing templates based solely on traffic information. Finally, the RND_{HH} problem is of theoretical interest by itself, encompassing many other well-known algorithmic problems, and is open to further investigation.

REFERENCES

- [1] N. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. Ramakrishnan, and J. Van der Merwe, "Resource management with hoses: point-to-cloud services for virtual private networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 10, no. 5, p. 692, 2002.
- [2] J. Fingerhut, S. Suri, and J. Turner, "Designing least-cost nonblocking broadband networks," *Journal of Algorithms*, vol. 24, no. 2, pp. 287–309, 1997.
- [3] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener, "Provisioning a virtual private network: a network design problem for multicommodity flow," in *Proceedings of the thirty-third annual ACM symposium on Theory of computing*. ACM, 2001, pp. 389–398.
- [4] F. Shepherd and P. Winzer, "Selective randomized load balancing and mesh networks with changing demands," *Journal of Optical Networking*, vol. 5, no. 5, pp. 320–339, 2006.
- [5] N. Olver and F. B. Shepherd, "Approximability of robust network design," in *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2010.
- [6] N. Olver, "A note on hierarchical hubbing for a generalization of the vpn problem," Tech. Rep., 2013.
- [7] R. Gao, C. Dovrolis, and E. Zegura, "Avoiding oscillations due to intelligent route control systems," in *Proc. of IEEE Infocom*, 2006.
- [8] L. Valiant, "A scheme for fast parallel communication," *SIAM journal on computing*, vol. 11, p. 350, 1982.
- [9] N. Goyal, N. Olver, and F. Shepherd, "The VPN conjecture is true," in *Proceedings of the 40th annual ACM symposium on Theory of computing*. ACM, 2008, pp. 443–450.
- [10] D. Mitra and R. Cieslak, "Randomized parallel communications on an extension of the omega network," *Journal of the ACM (JACM)*, vol. 34, no. 4, pp. 802–824, 1987.
- [11] I. Widjaja and A. Elwalid, "Exploiting parallelism to boost data-path rate in high-speed IP/MPLS networking," in *IEEE INFOCOM*, vol. 1. Citeseer, 2003, pp. 566–575.
- [12] I. Keslassy, S. Chuang, K. Yu, D. Miller, M. Horowitz, O. Solgaard, and N. McKeown, "Scaling Internet routers using optics," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2003, p. 200.
- [13] M. Kodialam, T. Lakshman, and S. Sengupta, "Efficient and robust routing of highly variable traffic," in *HotNets III*, 2004, p. 15.
- [14] R. Zhang-Shen and N. McKeown, "Designing a predictable Internet backbone network," in *HotNets III*, 2004.
- [15] H. Räcke, "Minimizing congestion in general networks," in *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, 2002.
- [16] A. Gupta, A. Kumar, M. Pal, and T. Roughgarden, "Approximation via cost sharing: Simpler and better approximation algorithms for network design," *Journal of the ACM (JACM)*, vol. 54, no. 3, p. 11, 2007.
- [17] F. Grandoni, V. Kaibel, G. Oriolo, and M. Skutella, "A short proof of the vpn tree routing conjecture on ring networks," *Operations Research Letters*, 2007.

- [18] M. Ghobadi, S. Ganti, and G. Shoja, "Resource optimization algorithms for virtual private networks using the hose model," *Computer Networks*, vol. 52, no. 16, pp. 3130–3147, 2008.
- [19] A. Kumar, R. Rastogi, A. Silberschatz, and B. Yener, "Algorithms for provisioning virtual private networks in the hose model," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 4, pp. 135–146, 2001.
- [20] N. Goyal, N. Olver, and F. B. Shepherd, "Dynamic vs. Oblivious Routing in Network Design," *Algorithms-ESA 2009*, pp. 277–288, 2009.
- [21] D. Peleg and E. Reshef, "Deterministic polylog approximation for minimum communication spanning trees (extended abstract)," *Automata, Languages and Programming*, p. 670, 1998.
- [22] T. Hu, "Optimum communication spanning trees," *SIAM Journal on Computing*, vol. 3, no. 3, pp. 188–195, 1974. [Online]. Available: <http://link.aip.org/link/?SMJ/3/188/1>
- [23] J. Fakcharoenphol, S. Rao, and K. Talwar, "A tight bound on approximating arbitrary metrics by tree metrics," in *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*. ACM, 2003, pp. 448–455.
- [24] N. Goyal, N. Olver, and F. B. Shepherd, "Dynamic vs. Oblivious Routing in Network Design," 2013.
- [25] S. Hoory, N. Linial, and A. Wigderson, "Expander graphs and their applications," *Bull. Amer. Math. Soc. (N.S.)*, vol. 43, pp. 439–561, 2006.
- [26] S. Chawla, R. Krauthgamer, R. Kumar, R. Yuval, and D. Sivakumar, "On the hardness of approximating multicut and sparsest-cut," *Computational Complexity*, vol. 15, pp. 94–114, 2006.
- [27] S. Arora, S. Rao, and U. Vazirani, "Expander flows, geometric embeddings and graph partitioning," *Journal of the ACM (JACM)*, vol. 56, no. 2, p. 5, 2009.
- [28] N. Spring, R. Mahajan, and D. Wetherall, "Measuring isp topologies with rocketfuel," *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 4, pp. 133–145, Aug. 2002. [Online]. Available: <http://doi.acm.org/10.1145/964725.633039>
- [29] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale ip traffic matrices from link loads," in *Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, ser. SIGMETRICS '03. New York, NY, USA: ACM, 2003, pp. 206–217. [Online]. Available: <http://doi.acm.org/10.1145/781027.781053>
- [30] H. Chang, M. Roughan, S. Uhlig, D. Alderson, and W. Willinger, "The many facets of internet topology and traffic," *Networks and Heterogeneous Media*, 2006.
- [31] V. Adhikari, S. Jain, and Z. Zhang, "Youtube traffic dynamics and its interplay with a tier-1 isp: an isp perspective," in *Proceedings of the 10th annual conference on Internet measurement*, 2010.