

# Designing Low Cost Networks with Short Routes and Low Congestion

Van Nguyen  
Dept. of Computer Science  
University of California, Davis  
California 95616  
Email: nguyenvk@cs.ucdavis.edu

Charles Martel  
Dept. of Computer Science  
University of California, Davis  
California 95616  
Email: martel@cs.ucdavis.edu

**Abstract**— We design network topologies and routing strategies which optimize several measures simultaneously: low cost, small routing diameter, bounded degree and low congestion. This set of design issues is broader than traditional network design and hence, our work is useful and relevant to a set of traditional and emerging design problems. Surprisingly, a simple idea from the research on small-world models, inspires a fruitful approach and useful techniques here.

Starting with a simple model we consider adding long links to an  $n \times n$  grid graph. Ideally, for a given budget to buy additional long links, we consider mechanisms for choosing links such that the routing diameter is small enough (poly-log of  $n$ ) while the congestion ratio (between the most used link and the average one) is minimized, assuming uniform traffic between any two of the  $n^2$  nodes. We show that by adding  $O(1)$  long links to each node we achieve an almost logarithmic routing diameter and maintain a near optimal trade-off between congestion ratio and average weight (of long links):  $Weight \times CongestionRatio = O(n)$ . Our results are comparable to the best similar network structures when the trade-off space we consider is reduced to those in the compared designs (with fewer trade-off factors). We also consider extensions of our results to more general settings.

We propose two construction schemes: 1) a static (fixed link) design and 2) a dynamic (random link) design. While the former provides our best trade-off results, the later is more scalable, better suited for dynamic and fault-tolerance issues, and can be useful for wireless ad-hoc networks.

## I. INTRODUCTION

### A. Trade-offs between weight (cost), routing diameter and congestion

We study networks (topological structures and routing strategies) which optimize several measures simultaneously: low graph weight, small routing diameter, bounded degree and low congestion<sup>1</sup>. In the context of computer networks, low weight means cheap cost for connecting cable (or bandwidth renting in building virtual private intranets); small diameter means limited hops in a path, and bounded degree means a bounded number of physical links connected to a node. Although the weight of a network link is naturally seen as (or proportional to) the Euclidean distance between nodes in our

<sup>1</sup>A real-world example (transportation networks of roads connecting a large number of locations) is suggested by Arya et al. [3]. Here, low weight means limited concrete to build roads, bounded degree means the number of roads incident to any location is bounded, and small diameter means each path can be described concisely. We add congestion to this road-network perspective.

abstract model, this can be realized by other specific measures, e.g. the transmission delay in the Internet, which also forms a metric [22].

There are complex trade-offs between these factors which makes it challenging to build a sound approach treating these issues all together. Previous work in network design usually focuses on a smaller set of aspects, which may ignore important issues. A full approach to this design problem can be useful for different classical areas of network designs, such as building a network from scratch, building a virtual private network over an existing infrastructure (say, the Internet), and is relevant to other research fields such as parallel architectures or VLSI circuit design. Specifically, we give a direct application of this work to the new paradigm of building hybrid ad-hoc networks by adding a wired infrastructure to an unstructured (ad-hoc) wireless network [11], [25]. Thus our work can be useful in both scenarios of network design: building a new network and adding links to an existing network. We discuss more about possible applications later.

Our basic model is to consider adding long links to a base graph (e.g. a simple grid) where the cost of a link is proportional to its weight (length). We consider a fundamental trade-off in this model between the total weight of the added long links, the routing diameter under a given routing algorithm (i.e. the hop-length of the longest routing path), and the congestion ratio, which indicates the ratio of traffic demand between the most used link and the average one. Ideally, for a given budget to buy additional long links, we consider mechanisms for choosing links such that the routing diameter is small enough (poly-log in the size of the node set) while the congestion ratio is as small as possible. In our basic model, where we assume uniform traffic between any two nodes of an  $n \times n$  grid<sup>2</sup>, we show that by adding  $O(1)$  long links to each node we can maintain a near optimal trade-off between congestion ratio and weight while keeping routing diameter in poly-log of  $n$ <sup>3</sup>.

Table I compares our trade-off results to existing network designs. Our comparisons assume uniform traffic and uniform

<sup>2</sup>We can think of a set of  $n^2$  sensor nodes scattered uniformly on a 2-dimensional plane such that the mean distance between any two nearby nodes is 1 (or there is 1 node per unit square on average).

<sup>3</sup>We can keep it as small as  $O(\log n)$ .

TABLE I

COMPARISON OF VARIOUS ROUTING NETWORKS IN THE BASIC SETTING

Network Designs	Degree	Routing diameter	Weight (W)	Congestion Ratio(CR)
E-Spanner	$O(1)$	$O(\log n)$	$O(\log^2 n)$	$\theta(n^2)$
Viceroy	$O(1)$	$O(\log n)$	$\Omega(n/\log n)$	$\theta(1)$
Ulysses	$O(\log n)$	$O(\frac{\log n}{\log \log n})$	$\Omega(n/\log n)$	$\theta(1)$
Ours :				
W-CR	$O(1)$	almost $O(\log n)$	$O(n^\delta)$	$O(n^{1-\delta})$
Opt-W	$O(1)$	$O(\log^{2.5} n)$	$O(1)$	$\theta(n)$

*E-Spanner*: Euclidean Spanners (Arya et al. [3]).

*Viceroy*: The Viceroy network (a randomized butterfly)[20] and similar randomized networks [1], [2].

*Ulysses*: The Ulysses network (a randomized butterfly) [29].

*W-CR*: Our scheme for  $W \times CR = O(n)$ , parameterized by  $\delta : 0 < \delta \leq 1$ .

*Opt-W*: Our scheme for optimal weight  $W$ .

node distribution <sup>4</sup>. The compared network structures are the best trade-off results in geometric spanners and peer-to-peer networks, (see more related work in [24]). These prior network designs optimize trade-offs between some of the factors, but none looks at all of them together. While Euclidean spanners [3] achieve almost (asymptotically) optimal trade-offs between degree, diameter and weight, they perform the worst for congestion <sup>5</sup>. The Viceroy network [20] and similar ones in [1], [2] achieve an optimal trade-off between degree, diameter and congestion at the expense of massive weight: it uses long links of average weight  $\Omega(n/\log n)$ , which is almost the (asymptotically) maximum  $O(n)$ , and it has no flexibility for lower weight. Ulysses network [29] has the smallest diameter and ideal congestion but also has expensive weight and does not have bounded degree. We, however, can obtain ideal degree and diameter and yet a near optimal weight-congestion trade-off:  $W \times CR = O(n)$  (<sup>6</sup>). Note that, (mostly with our W-CR scheme) we maintain a varying trade-off between weight and congestion: we can achieve any weight from  $\theta(1)$  to  $\theta(n)$  and obtain a corresponding congestion. In scheme Opt-W, we achieve optimal weight with a (small) poly-log diameter and congestion much better than in Euclidean Spanners.

The basic idea behind our mechanism of choosing links is quite simple. We construct a partitioning hierarchy, dividing the  $n \times n$  square into a multi-level system of regions, just like the popular hierarchy country-state-county-district... We then classify links into several layers: links between ‘states’, links between ‘counties’, etc. While the ‘interstate’ links can help to greatly reduce the routing diameter, they consume the biggest part of our budget and are the main sources of congestion. Nonetheless, the point is to invest aggressively in this top layer (to reduce congestion ratio) while paying just enough for the lower layer links so that local routing can

<sup>4</sup> $n^2$  nodes uniformly distributed in a  $2D$   $n \times n$  square

<sup>5</sup>In this setting, we also have  $EC = CR \times \theta(n^2)$  where edge-congestion  $EC$ , a more popular congestion measure, is defined in §II.

<sup>6</sup>In §II, we also show a lower bound indicating a small gap (a poly-log of  $n$  factor) between the two bounds

still be effective (there is a big ‘inter-state’ link with ends a few hops from our source and destination). Our construction schemes show how to do that systematically for a broad range of desired total weight (budget).

We consider two schemes of adding links: a fixed link scheme with a mechanism to specify link positions deterministically and a random link scheme, where links are generated under a special distribution. While the former can approach near optimal trade-offs as we mentioned above, the later is more scalable, better suited for dynamic issues (with events like joining/removal of nodes or a massive deletion attack). Although we base our designs on a hierarchical model, our method and results are in fact inspired from small-world models, such as by Kleinberg [15], and is different from classical hierarchical network architectures (e.g. Kleinrock and Kamoun’s [16]).

**Towards a more general model (cost-diameter-throughput).** We also consider more general models where 1) we allow non-uniform traffic and 2) nodes can be placed arbitrarily on the plane. Now some nodes are much ‘busier’ than others, so we assume that the long links can different capacities (rather than all links the same capacity as before), and thus, the cost of a link will be a product of its length and its capacity (a two-unit capacity link is equivalent to two parallel single-unit links). We have some promising ideas for network designs and routing in these more general settings (particularly if nodes are in the plane with link costs proportional to the Euclidean distances between nodes). We give more details on possible extensions in section §VI on future work.

### B. Areas of possible application.

Our results can be used in building hybrid ad-hoc networks. Helmy [11] and independently, Reznik et al. [25] propose to add a wired infrastructure to an unstructured (ad-hoc) wireless network. Helmy’s experimental approach shows that we can reduce the average path length by approximately 50% by adding random links <sup>7</sup> to  $\approx 1\%$  of the nodes. Helmy, however, does not provide any general result or systematic framework. He considers using the uniform distribution (of random links) and *only reduce the path length by a multiple constant*. Reznik et al. provide an elaborate framework with a specific parameterized distribution for choosing long links. This helps to reduce the path length to a *small power of the initial diameter*. However, they simplify their routing scheme in using at most one short-cut per route.

Our results reduce the path length (routing diameter in our model) to *only a poly-logarithmic function*. Moreover, we consider how to best deal with congestion which is potentially high in these short-cuts. Our near optimal trade-off results provide solutions to the problem of balancing between the budget (weight of long links), path length (routing diameter) and congestion. The final section gives a preliminary evaluation of our work in this area.

<sup>7</sup>Helmy achieves the maximum path length reduction while limiting the length of random links to 25 – 40% of the network diameter.

Our trade-off results for the basic model also provide a view to the capacity of the hybrid network (through our congestion and diameter measures). Gupta and Kumar’s seminal paper [10] shows that the capacity of a wireless network is bounded by a slow function of the number of nodes such that the throughput per node tends to be dismissed when the number of nodes (inside a fixed area) is increased. By adding wired long links, we expect to see better capacity. We mention more on this in section VI.

Our results also suggest new designs for some classical network design problems, such as in parallel architectures or VLSI circuits. In these classical fields, all the design issues we consider are regularly used, but usually not all in a balanced combination as we propose. Our designs here are in fact comparable to many popular parallel designs (e.g. the butterfly networks and other hypercubic networks) if projected into their scenario.

The general model provides a framework for some other complex practical problems, such as building a virtual private network over an existing infrastructure. For this particular problem, the weight of a long link of certain capacity (and maybe some other QoS requirements) can be seen as the cost to rent a certain permanent bandwidth between two nodes (from the providers of this underlying infrastructure). Although a real cost function may not be this simple (e.g. the real cost may not be linear, since the same bandwidth at a highly demanded link can be more expensive), the framework can produce some reasonable approximation methods.

### C. Related work

While a vast body of work studies the triple degree-diameter-weight trade-offs<sup>8</sup>, little attention was given to the congestion issue with respect to these other factors. *Edge-congestion* has been used as a network measure to evaluate the performance of architectures, especially parallel ones [7], [6]. The minimum edge-congestion of the network (over all routing algorithms) is also known as ‘*the edge forwarding index*’, which is introduced in [12]. Our notion of *congestion ratio* has some relation to a few recent papers. Xu et. al [29] define a network as “*c*-edge-congestion-free” if no edge handles more than *c* times the average traffic per edge (assuming a uniform all-to-all communication). Gao and Zhang [8] use a similar concept, “load-balancing ratio” to evaluate different routing algorithms for a given network graph. A survey on algorithms for Internet congestion control can be found in [28].

The recently hot area of P2P Distributed Hash Table (DHT) research has provided several nice architectures, that consider issues similar to ours (diameter, bounded-degree and even congestion). Xu et al. [29] provide a full treatment to the diameter-degree trade-off. Asymptotically optimal schemes are also provided in [20], [19]. Loguinov et al. provide a graph-theoretic framework to analyze and compare P2P networks on several properties which affect routing and resilience [19].

<sup>8</sup>Mostly for the classical Minimum Steiner Tree (extensions also include bounded diameter and/or bounded-degree).

Although the mentioned P2P networks can be optimal in degree-diameter and are sometime congestion-aware [29], they omit the physical cost (weight) issue (which is irrelevant in the P2P scenario, where links are only logical), so these designs have high weight if switched to our design context (table I).

Besides the mentioned problems in wireless ad-hoc networks (§I-B), our general model (with non-uniform demands) may also be relevant to another: given a set of nodes with end-to-end traffic demands, find a network topology that meets the QoS requirements and minimizes the maximum transmitting power of nodes [13]. The work in [4] considers a multi-path routing algorithm which minimizes congestion using a multi-commodity flow approach. We consider a similar approach in §VI.

As mentioned, this work is inspired by our study of small-world models [23], [21], which follows the seminal work by Kleinberg [15]. Kleinberg adds directed long-range random links to an undirected  $n \times n$  lattice network, where the long-range links have a non-uniform distribution which favors arcs to close nodes over more distant ones. The idea of adding long-range random links into a graph mostly based on local contacts inspires several applications such as Malkhi et al.’s Viceroy P2P network [20] and Helmy’s hybrid ad-hoc network [11].

**The structure of this paper.** In §II we present our basic model and initial facts. In §III we present our theorems on our fixed link scheme, which provide and analyze our trade-offs. In §IV we present similar results for our random link scheme, however, we focus more on routing strategy and some dynamic issues (node addition/deletion; fault-tolerance). In §VI, we discuss our future work on the above mentioned general model and ways to extend our current approach for this. A preliminary numerical evaluation is provided in §V.

## II. BASIC MODEL AND NOTION

In our basic model, we consider an  $n \times n$  grid network, where there are  $N = n^2$  nodes on the integer points of the square  $(0, 0, n - 1, n - 1)$  and where the traffic demands from any node to any other node are all equal<sup>9</sup>. Each node has (usually 4) undirected links to its neighbor nodes at distance 1. We consider more general models in §VI. We now consider adding additional (long) links to shrink the graph diameter.

*Definition 1:* In a Euclidean 2-dimensional space, consider an  $n \times n$  grid-based network where we add  $O(1)$  long links to each node. The total weight  $T$  is the total length of all these long links and the average weight  $W$  of the long links is  $T/n^2$ .

Note that the weight of the basic grid is  $\theta(n^2)$  and its average weight per node is just  $2 - o(1)$ . The average weight of the links we add will range from  $\theta(1)$  to  $\theta(n)$  as we consider strategies to minimize congestion.

<sup>9</sup>However, our results can be easily extended to a more general setting (see IV-C), where a set of  $n^2$  nodes are scattered uniformly on a 2-dimensional plane such that there is expected 1 node per unit square. Roughly, such a setting can be approximated by a grid setting where the difference between these two (in e.g. graph weight) is only a small fraction of the total.

We discuss the congestion notion as an inherent property of the network topology which is associated with a certain given routing algorithm. Under our uniform traffic model, consider a given routed network, i.e. a pair  $(G, RA)$  of a network graph  $G$  and an associated routing algorithm  $RA$ , where each pair of nodes is given a unique routing path<sup>10</sup>.

#### A. Edge-congestion and congestion ratio

*Definition 2:* The *congestion*  $EC(G, RA, e)$  of a link  $e$  is the number of routes using this link. The *edge-congestion of the network*  $EC(G, RA)$  is the maximum value of congestion over all the edges. The *congestion ratio*  $CR(G, RA)$  is the ratio of this maximum value over the average value.

*Definition 3:* For a given routed network  $(G, RA)$ , the *routing diameter*  $D(G, RA)$  is the maximum hop-count over all the routes used by  $RA$  (between any two nodes).

We will omit parameters  $G$  and  $RA$  when they are clear from the context. The congestion ratio tells how balanced the routing system can be, where a high congestion ratio means some links are much hotter than the others (so, are traffic bottlenecks). Although edge-congestion is more standard than congestion ratio, we use the later since it reflects better our objective for optimizing load-balancing (avoiding hot links) while considering the weight and diameter<sup>11</sup>. Note that our notions of congestion are defined with respect to a given routing algorithm. We do not limit ourselves to shortest path routing algorithms, which can be hard to implement (especially, for distributed scenarios). Often routes within a small factor of the shortest are good enough.

Note the difference between routing diameter and graph diameter: while the later depends only on the graph topology, the former depends on the topology and the routing algorithm associated with the network. We now bound edge-congestion  $EC$  and congestion ratio  $CR$  for a pair  $(G, RA)$  with routing diameter  $D$  and (long link) average weight  $W$ .

*Fact 1:* For any  $n \times n$  grid-based routed network with routing diameter  $D$  and (long link) average weight  $W$ , the edge-congestion  $EC = \Omega(n^3/DW)$ , the average congestion is  $O(n^2D)$ , and the congestion ratio

$$CR = \Omega\left(\frac{n}{D^2W}\right) \quad (1)$$

*Proof:* Consider the congestion of a *big link* which has length at least  $n/2D$ . The number of these big links is at most the total weight of long links ( $n^2W$ ) over  $n/2D$ , which is  $2nDW$ . Consider routing paths for pairs of nodes at distance at least  $n/2$ . Since there is at most  $D$  links in each path, we must have at least one link with length  $n/2D$  (a big link). Clearly, the number of such long paths is at least a constant fraction of the whole ( $n^2(n^2 - 1)$ ) so, it is  $\theta(n^4)$ . Therefore the average

<sup>10</sup>Equivalently, a path system can be used here instead as in many previous papers.

<sup>11</sup>Note that for our network setting, the average congestion is  $\Omega(n^2)$  and  $O(n^2D)$ , and hence  $EC$  is always bounded between  $CR \times \theta(n^2)$  and  $CR \times \theta(n^2D)$ . As we obtain small poly-log routing diameter, optimizing  $CR$  is almost equivalent to optimizing  $EC$ .

congestion over a big link is  $\Omega(n^4/2nDW) = \Omega(n^3/DW)$ ; thus, the edge-congestion  $EC(G, A) = \Omega(n^3/DW)$ .

We now consider the congestion ratio. There are a total of  $O(n^4D)$  units of load on all the edges for  $\theta(n^4)$  routes with each using at most  $D$  edges. There are  $\theta(n^2)$  edges in a bounded-degree network. Thus, the average congestion is  $O(n^4D/n^2) = O(n^2D)$ . So, the congestion ratio is  $\Omega(n^3/DW)/O(n^2D) = \Omega(n/D^2W)$ . ■

Fact 1 shows a lower bound for congestion ratio  $CR$  (and edge-congestion  $EC$ ) when routing diameter  $D$  and average weight  $W$  of additional long links are given. Equation (1) can also be rewritten as

$$CR \times W = \Omega\left(\frac{n}{D^2}\right) \quad (2)$$

This shows a trade-off between  $CR$  and average weight  $W$  when routing diameter  $D$  is given. For example, for those networks with diameter  $D = O(\log^c n)$  for some constant  $c > 0$ , we have  $CR \times W = \Omega\left(\frac{n}{\log^{2c} n}\right)$ . Later, we show that this bound is almost tight: we propose network designs with  $D = O(\log^c n)$  where  $CR \times W$  is just  $O(n)$ .

#### B. Partitioning hierarchy

The idea of using a special block decomposition, where the child block size equals a (small) power of the parent block size, is crucial in our network design and analysis. We will be using block-based measures such as block weight, block-to-block (communication) demand, block-based routing, etc.

Given a constant  $.5 \leq \mu < 1$ , we assume that  $l = m^\mu$  is an integer which divides  $m$  (the general case only adds some tedious details which can be seen in [24]). A  $\mu$ -partition of a block of size  $m$  (i.e. an  $m \times m$  square of nodes) is the set of  $q^2$  identical sub-blocks of size  $l \times l$ , where  $q = l/m$ , i.e. a  $q \times q$  grid if each  $l \times l$  block collapses to a single node. Each sub-block has its row and column index (each ranges from 1 to  $q$ ) within the parent block.

A partitioning hierarchy of a given  $n \times n$  block is a process where we start with the  $n \times n$  block, do a  $\mu$ -partition for some constant  $.5 \leq \mu < 1$ , partition the sub-blocks, and so on ... until, at a certain level we reach a node block of size 2 (or any threshold constant size). We can think of a tree of blocks with the root as the initial  $n \times n$  block and the leaves as blocks of size  $O(1)$ . At each level of this block tree, all the blocks of the level will be  $\mu$ -partitioned for the same value of  $\mu$ , which can be different for different levels. Thus, a block tree is determined given a size  $n$  and a series of values of  $\mu$ , each for the next step of partitioning.

We denote a block at level  $i$  as an  $i$ -block and a link connecting two sibling blocks at level  $i$  as an  $i$ -block link. To simplify our analysis, we assume that, within a level, all the block have the same size: equals  $n^{\mu^k}$  for level  $k > 0$ , and equals  $n^{\mu^L} = 2$  for the last level  $L$  (so,  $L = \log_{1/\mu} 2 \times \log \log n$ ). Our analysis of this simplified scenario can be extended to handle the general case.

We will mostly consider two types of partitioning, one with a fixed value of  $\mu$  for all levels, and another with two values of  $\mu$ :  $\mu_1$  for the top  $C$  levels (for some parameter  $C$ ) and

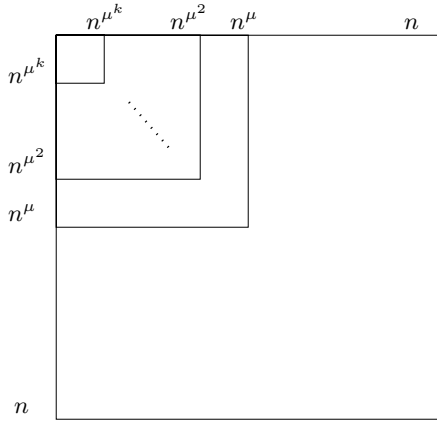


Fig. 1.  $\mu$ -hierarchy

$\mu_2$  for the remaining levels down to the bottom. We use a  $\mu$ -hierarchy to denote a partitioning hierarchy of the first type and a  $(\mu_1, \mu_2, C)$ -hierarchy to refer to one of the second type.

### III. THE FIXED-LINK SCHEME

A fixed-link scheme based on an  $n \times n$  grid is actually a partitioning hierarchy plus a (long) link mechanism: for any two ‘sibling’ blocks of the same parent block, one node is chosen from each block, and we create an undirected (long) link between these two nodes. By varying parameter  $\mu$  we can obtain different values of routing diameter  $D$  and average weight  $W$  as well as congestion ratio  $CR$ . We can change the value of  $\mu$  between different partitioning steps: by doing so selectively, we can obtain near optimal solutions.

The long link selection can be represented by a function where the inputs are the  $(row, column)$  indexes of two sibling block within the parent block, and the outputs are the relative indexes of the 2 chosen nodes within each sibling block. This assignment function can also depend on the position of the parent block within the block tree. However, we will focus on a specific scheme where the *link assignment mechanism is uniform*: the assignment function only depends on the size of the parent block (or the level) and the  $\mu$  value. That is, given fixed sizes of the parent and children blocks, the assignment function does not depend on the position of the parent block.

#### A. Our routing algorithm

Consider an  $\mu$ -hierarchy on a  $n \times n$  grid with a block tree of height  $L = \log_{1/\mu} 2 \times \log \log n$ . We suggest a natural hierarchical routing strategy on this  $\mu$ -hierarchy network. The basic idea of routing from a source node  $u$  to a destination node  $v$  is to try to get closer to  $v$  in several phases, each of which gets to a smaller, inner sub-block containing  $v$ . Suppose that  $B$  is the smallest block containing both  $u$  and  $v$ ,  $B_1$  and  $B_2$  are two separate sub-blocks in  $B$  where  $u \in B_1$  and  $v \in B_2$ . Also, suppose that  $w_1 \in B_1$  and  $w_2 \in B_2$  with link  $(w_1, w_2)$  between the two sibling blocks  $B_1$  and  $B_2$ . We now can route from  $u$  to  $w_1$ , take the link  $(w_1, w_2)$  and then route from  $w_2$  to  $v$  (see figure 2).

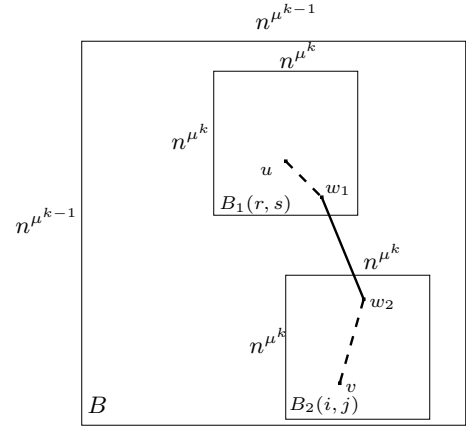


Fig. 2. Routing in  $\mu$ -hierarchy.

For our suggested assignment function in §III-B,  $w_1$  is chosen as  $(\frac{lr}{q}, \frac{ls}{q})$  within  $B_1$ , and  $w_2$  as  $(\frac{li}{q}, \frac{lj}{q})$  in  $B_2$

It is not hard to see that any routing path within a block at height  $h$  (in the block tree) has hop-length bounded by  $O(2^h)$ <sup>12</sup>. Thus, the routing diameter of this network design is  $O(2^L) = O(\log^\gamma n)$  where  $\gamma = \log_{1/\mu} 2$  (recall that  $L = \log_{1/\mu} 2 \times \log \log n$ ). The routing diameter is  $O(\log n)$  only if  $\mu = .5$

*Lemma 2:* The hop-length of the routing path between any two nodes  $u$  and  $v$  is  $O(2^{h+1})$ , where  $h$  is the height (in the block tree) of the smallest block containing both  $u$  and  $v$ . For  $\mu \geq .5$ , the routing diameter is  $O(\log^\gamma n)$  where  $\gamma = \log_{1/\mu} 2$ , and is  $O(\log n)$  if and only if  $\mu = .5$ .

The routing algorithm above works with any distribution of long links as long as there is a link to connect any pair of two sibling blocks at each level; that is the link assignment function can be arbitrary. However, to avoid hot spots it is better to have a constant density of *router* node, which are nodes incident to a long link connecting two certain sub-blocks.

#### B. Our long link assignment function

We suggest a simple assignment function which results in such a constant density of router nodes (for  $\mu \geq .5$ ). Using the notation in §II-B, consider an  $m \times m$  block which has  $q^2$  sub-blocks each of size  $l \times l$  with  $l = m^\mu$  (note that  $q \leq l$  since  $\mu \geq .5$ ). By assigning a long link to each pair of sub-blocks, each sub-block has  $q^2 - 1$  long links to other blocks. We can attain a constant density of router nodes by using a ‘perspective-reserved’<sup>13</sup> map from the virtual  $q \times q$  grid of the sub-blocks (as if each sub-block collapses to a single node) to the  $l \times l$  grid of nodes in each sub-block.

Recall that an  $(i, j)$ -block is a sub-block with row  $i$  and column  $j$  inside its parent block. A simple such function is

<sup>12</sup>See [23] for a more detailed analysis of a similar fact.

<sup>13</sup>Informally, for a link  $(w_1, w_2)$  connecting two sub-blocks  $B_1$  and  $B_2$  within parent block  $B$ , the relative position of  $w_1$  in  $B_1$  is determined by the relative position of  $B_2$  in  $B$ , and similarly,  $w_2$  in  $B_2$  is like  $B_1$  in  $B$ .

as follows: for an  $(i, j)$ -block  $A$  and an  $(r, s)$ -block  $B$ ,  $1 \leq i, j, r, s \leq q$  and  $(i, j) \neq (r, s)$ , we map  $A$  to node  $(\frac{li}{q}, \frac{lj}{q})$  in  $B$  <sup>(14)</sup>. Similarly, we also map  $B$  to node  $(\frac{lr}{q}, \frac{ls}{q})$  in  $A$ . That is, we make a long link from node  $(\frac{lr}{q}, \frac{ls}{q})$  in  $A$  to node  $(\frac{li}{q}, \frac{lj}{q})$  in  $B$  (see figure 2). These 2 nodes are called router nodes (for  $B$  in  $A$  and for  $A$  in  $B$ ). Since the size of each sub-block is at least the size of the (virtual) grid of sub-blocks (i.e.  $l \geq q$  as long as  $\mu \geq .5$ ), the map function is a ‘zoom-in’ (with the domain not bigger than the codomain). Thus, all the router nodes (of the same level) are distinct, which assures that each node has at most one long link at each level.

For  $\mu = .5$ , each node will have  $L$  long links (one per level). Although this case does not have an ideal  $O(1)$  degree (but small typically), it is useful for our later consideration. For  $\mu > .5$ , the number of router nodes inside a sub-block is strictly smaller than the number of nodes in it and hence, it is not hard to refine the assignment function so that all nodes have constant degrees <sup>(15)</sup>. For  $\mu < .5$ , the degrees are big (polynomial in  $n$ ) and hence, we only consider using  $.5 \leq \mu < 1$  in our designs.

*Fact 3:* Using our long link assignment function, the network has node degree

- a)  $\log \log n$  for  $\mu = .5$
- b)  $O(1)$  for  $\mu > .5$

### C. Our network constructions

We now consider constructing a  $\mu$ -hierarchy plus a uniform long link mechanism, using the assignment function suggested above. The network is associated with our hierarchical routing algorithm. We show that such a scheme can achieve a near optimal trade-off between average weight  $W$  and congestion ratio  $CR$ :  $W \times CR = O(n)$ , while keeping routing diameter  $D$  as a poly-log function. This upper bound on  $CR \times W$  is only a poly-log factor from the lower bound in (2), as long as  $D$  is a poly-log function. Moreover, by varying  $\mu$  between  $.5$  and  $1$ , we show that the trade-off is maintained for a broad range of  $W$ , from almost as low as  $\theta(1)$  to as high as  $\theta(n)$ .

There are two key ideas in our designs to compute  $CR$  and obtain  $W \times CR = O(n)$ . We look at the ‘big’ links, i.e. the 1-block links (connecting between the top level blocks), which handle all the traffic between each pair of 1-blocks (implied naturally from our routing algorithm). Moreover, from the uniform traffic assumption and uniform distribution of router nodes (by using our uniform assignment function) it is easy to show that these 1-block links are the ‘hottest’ and equally congested. Thus, to compute edge-congestion  $EC$  we simply compute the congestion through a single 1-block link.

<sup>14</sup>For simplicity, we often assume the fractions here are integers but rounded functions can be used in implementation.

<sup>15</sup>To assure that no node has many more long links than the others (after the assignment function is applied at all  $L$  levels of the  $\mu$ -hierarchy), for each level  $k$  large enough, we ‘round up’ each  $(\frac{li}{q}, \frac{lj}{q})$  to the closest  $(x, y)$  with  $(x + y) \bmod L = k$  only. This guarantees no two long links (of at least level  $C$  for some constant  $C$  large enough) are incident to the same node; thus the nodes have  $O(1)$  degrees.

We denote the total weight of the long links in level  $i$  by  $T_i$ , and in all levels by  $T$ . We can show that, if the 1-block links dominate the total weight such that  $T_1 = \theta(T)$ , then we get  $CR \times W = O(n)$ .

*Fact 4:* In our  $\mu$ -hierarchy networks, using our uniform long link assignment and routing algorithm, a) the 1-block links are the most congested and they are equally congested, b) if they dominate in total weight such that  $T_1 = \theta(T)$ , then if  $.5 < \mu < 1$ ,

$$CR \times W = O(n) \quad (3)$$

We will show in our following theorem that  $T_1 = \theta(T)$  when  $.5 \leq \mu < 3/4$ .

*Proof:* Part a) is clear as mentioned above. We now prove part b). Let  $n_1$  be the number of these big links, i.e. 1-block links. It is easy to see that half of these big links have (Euclidean) length almost  $n/4$ ; so,  $T_1 = n_1 \times \theta(n)$ . Since  $T_1 = \theta(T)$ , we have  $n_1 = T/\theta(n) = W \times \theta(n)$ . Hence,  $W = n_1/\theta(n)$ .

Now, each  $s - t$  routing path has at most one 1-block link, so the congestion of a 1-block link (which is just  $EC$ ), is at most  $N/n_1$  where  $N = \theta(n^4)$  is the number of all  $s - t$  routing paths. On the other hand, since each node has  $O(1)$  degree, the number of edges is  $O(n^2)$  and hence, the average congestion is at least  $N/O(n^2)$ . Thus,  $CR$  is at most

$$\frac{N}{n_1} \div \frac{N}{O(n^2)} = \frac{O(n^2)}{n_1}$$

$$\text{Therefore, } CR \times W = \frac{O(n^2)}{n_1} \times \frac{n_1}{\theta(n)} = O(n). \quad \blacksquare$$

This reflects a crucial idea in our designs: to make the 1-block links big enough to dominate in the total weight. We now show a lemma (proof in [24]), which is used to estimate the total weight of long links at each level, and then our main theorems.

*Lemma 5:* For a  $\mu$ -hierarchy with our uniform assignment function,  $.5 \leq \mu < 1$ , a block tree of height  $L$ , and for  $\delta = 3 - 4\mu$ ,

- a) the total weight of long links at the first level is  $T_1 = \theta(n^{2+\delta})$ ,
- b) the total weight of long links at level  $i \leq L$  is  $T_i = \theta(n^{2+\mu^{i-1}\delta})$ .

*Theorem 1:* Consider a  $\mu$ -hierarchy with our uniform assignment function and hierarchical routing algorithm and with a block tree of height  $L$ . Define  $\delta = 3 - 4\mu$  (i.e.  $\mu = \frac{3}{4} - \frac{\delta}{4}$ ). The  $\mu$ -hierarchy with parameter

- a)  $\frac{1}{2} \leq \mu < \frac{3}{4}$  ( $1 > \delta \geq 0$ ), achieves  $D = O(\log^{2.5} n)$ ,  $W = O(n^\delta)$ ,  $EC = O(n^{3-\delta})$  and  $CR = O(n^{1-\delta})$
- b)  $\mu = \frac{3}{4}$  ( $\delta = 0$ ), achieves  $D = O(\log^{2.5} n)$ ,  $W = O(\log \log n)$ ,  $EC = \theta(n^3)$  and  $CR = O(n)$ .
- c)  $\frac{3}{4} < \mu < 1$  ( $0 > \delta > -1$ ), achieves  $D = O(\log^\gamma n)$  where  $\gamma = \log_{1/\mu} 2$ ,  $W = O(1)$ ,  $EC = O(n^{3+|\delta|})$  and  $CR = O(n^{1+|\delta|})$ .

*Proof:* Note that  $L = \log_{1/\mu} 2 \times \log \log n$ . From lemma 2,  $D = O(2^L) = O(\log^\gamma n)$  where  $\gamma = \log_{1/\mu} 2$ . For  $\mu \leq \frac{3}{4}$ ,

$\gamma \leq \log_{\frac{4}{3}} 2 \approx 2.41$ , and hence,  $D = O(\log^{2.5} n)$ .

Now, for computing  $EC$  and  $CR$ , we need to look at the ‘big’ links, i.e. the 1-block links, which handle all the traffic between each pair of 1-blocks. From fact 4, these links dominate  $EC$ , so:

$$EC = \theta(n^{2\mu} \times n^{2\mu}) = \theta(n^{4\mu}) = \theta(n^{3-\delta})$$

The average congestion is clearly  $\Omega(n^2)$ , so  $CR = O(n^{1-\delta})$ . This implies the respective  $EC, CR$  in a), b) and c). We omit the computation of the average weight  $W$ , which can be seen in [24]. ■

Note that the constructed networks have degree  $O(1)$  except for  $\mu = .5$  where we have degree  $O(\log \log n)$  instead. Theorem 1 is only for schemes using the same  $\mu$  throughout all the partitioning levels, based on which we consider more refined schemes with varied  $\mu$  for better trade-offs.

#### D. Near-optimal $W - CR$ trade-off with almost optimal $D$

We need a more sophisticated design to achieve near-optimal  $W - CR$  trade-off for smaller routing diameter, especially for matching the diameter lower bound of  $\theta(\log n)$ . From theorem 1,  $W$  is almost minimized (asymptotically) if  $\mu = \frac{3}{4}$ . However,  $D$  is minimized (asymptotically) if  $\mu = \frac{1}{2}$ , due to lemma 2. So, the basic idea is to use a few levels with  $\mu$  around  $\frac{3}{4}$  (actually, a bit less) on top of a  $\frac{1}{2}$ -hierarchy. By choosing proper constants, these few top levels will dominate the total weight, which can be made close to the minimum, while only adding a multiplicative constant to  $D$ , which is still  $O(\log n)$  since all the remaining levels use  $\mu = \frac{1}{2}$ . To get  $D = \theta(\log n)$ , we use a  $(\mu_1, \mu_2, C)$ -hierarchy for some appropriate  $\mu \lesssim \frac{3}{4}$  for just a few,  $C$ , top levels and then  $\mu = \frac{1}{2}$  for the remaining  $L - C$  levels. In fact, the dominating contribution of the top level in total weight (i.e.  $\sum T_i = O(T_1)$ ) makes sure that we still obtain  $EC \times W = O(n^3)$  and  $CR \times W = O(n)$ . The proof of the following theorem is in [24].

*Theorem 2:* For any  $0 < \delta < 1$ , the  $(\mu_1, \mu_2, C)$ -hierarchy with  $\mu_1 = \frac{3-\delta}{4}$ ,  $\mu_2 = \frac{1}{2}$  and  $C = \lceil \log_{\frac{1}{\mu_1}} (\frac{1}{\delta} + 1) \rceil$  (with our uniform assignment and routing algorithm), achieves  $D = \theta(\log n)$ ,  $W = O(n^\delta)$ ,  $EC = O(n^{3-\delta})$ , and  $CR = O(n^{1-\delta})$ .

However, since we use  $\mu = \frac{1}{2}$  for most levels ( $L - C$  out of  $L$ ), the degree of each node becomes  $\log \log n - O(1)$  due to fact 3. We can tune the scheme to obtain degree  $O(1)$  but with a (very slightly) larger  $D = O(\log n \sqrt{\log \log n})$ . To do that, initially we group nodes into small squares of size  $\sqrt{\log \log n}$  and consider a grid of super nodes with size  $m \times m$  where  $m = n / \sqrt{\log \log n}$ . We then construct a network as in theorem 2 on this grid of super nodes and thus, each super node has degree  $L' = \log \log m = L - o(L)$ . We then can simply assign each real node (inside a super one) to handle a long link. The routing diameter of this final structure is  $D = O(\log n \sqrt{\log \log n})$  since it takes  $O(\sqrt{\log \log n})$  links to route within a super node.

#### E. Near-optimal $W - CR$ trade-off with optimal $W$

Note that in theorem 1-c (for  $\mu > 3/4$ ), although we have  $W = O(1)$  we can not maintain  $W \times CR = O(n)$ . In fact, it

TABLE II  
TRADE-OFF RESULTS IN OUR FIXED-LINK SCHEMES

$\mu$ -hierarchy	Routing diameter	Weight	Congestion Ratio
Single $\mu$ -hier.			
$\frac{1}{2} < \mu < \frac{3}{4}$	$O(\log^{2.5} n)$	$O(n^\delta)$	$O(n^{1-\delta})$
$\mu = \frac{1}{2}$	$\theta(\log n)$	$O(\frac{n}{\log \log n})$	$O(1)$
$\mu = \frac{3}{4}$	$O(\log^{2.5} n)$	$O(\log \log n)$	$O(n)$
$\frac{3}{4} < \mu < 1$	$O(\log^\gamma n)$ , $\gamma = \log_{1/\mu} 2$	$O(1)$	$O(n^{3+ \delta })$ , $\delta = 3 - 4\mu$
Complex			
$(\frac{3-\delta}{4}, \frac{1}{2}, C)$ -hier.,			
Variant 1	$O(\log n)$	$O(n^\delta)$	$O(n^{1-\delta})$
Variant 2	$O(\log n \times \sqrt{\log \log n})$	$O(n^\delta)$	$O(n^{1-\delta})$
$(\frac{3}{4}, \frac{3}{4} + o(1), 1)$ -hier.	$O(\log^{2.5} n)$	$O(1)$	$\theta(n)$

All schemes have  $O(1)$  degree except the  $\frac{1}{2}$ -hierarchy and the  $(\frac{3-\delta}{4}, \frac{1}{2}, C)$ -hierarchy, variant 1 have degree  $O(\log \log n)$ .

is easy to verify that the 1-block links do not dominate in total weight for  $\mu \geq 3/4$ . Lemma 5 shows that the series  $\{T_i\}_{i=1}^L$  increases instead, and  $T_L$  reaches the peak of  $\theta(n^2)$ . Note that  $T = \theta(n^2)$  also.

In order to obtain  $W \times CR = O(n)$ , we need to have  $T_1$  dominate the total weight again. This can be done by a simple trick, similarly as in §III-D: adding a top level with  $\mu = 3/4$  to a  $(3/4 + \delta)$ -hierarchy. This top level also has  $T_1 = \theta(n^2)$  and hence,  $T_1 = \theta(T)$ . We omit the proof of the following theorem, which uses the same ideas as in theorem 1 and 2.

*Theorem 3:* The  $(\frac{3}{4}, \mu, 1)$ -hierarchy with  $\mu > \frac{3}{4}$  (with our uniform assignment and routing algorithm), achieves  $D = O(\log^\gamma n)$  where  $\gamma = \log_{1/\mu} 2$ ,  $W = O(1)$ ,  $EC = \theta(n^3)$  and  $CR = O(n)$ .

By choosing  $\mu$  close enough to  $\frac{3}{4}$ , we have  $D = O(\log^{2.5})$ .

#### F. Concluding remarks

We summarize all the trade-off results in table II. The theorems above show how we can construct a  $\mu$ -hierarchy network with  $D$  as a slow poly-log function, with a near optimal<sup>16</sup> trade-off between  $CR$  and  $W$ :  $CR \times W = O(n)$ , and for any average weight  $W$  between  $\theta(1)$  and  $\theta(n)$ . The suggested  $\mu$ -hierarchies above are almost the best congestion for any fixed desired  $W$ . Consider a few special cases.

For  $\mu = \frac{1}{2}$  (hence,  $\delta = 1$ ), as in §III-D, we can construct a  $\mu$ -hierarchy for  $D = O(\log n \sqrt{\log \log n})$ ,  $W = O(n)$  and  $CR = O(1)$ . This is an ideal load balance on the links, however the average weight of a long link is  $O(n)$ , the most of any of our designs (asymptotically). This performance is similar to that of some P2P architectures such as in [20], [2].

<sup>16</sup>Note that the gap between the lower bound and upper bound for  $CR \times W$ , in (2) and in (3), is a multiple factor  $D^2$ , which, we conjecture, could be reduced to say,  $D$ . This is because we think the bound  $\Omega(n^2)$  on the average congestion may be too low.

The construction in §III-E is, however, the opposite scenario (low weight but high congestion), where we can obtain  $D = O(\log^{2.5} n)$ ,  $W = O(1)$  and  $CR = O(n)$ . Here we only require a cost which is within a constant multiple of the cost to connect the nodes by a minimum spanning tree or a grid (which has diameter  $\Omega(n)$ ), yet we still have a small poly-log diameter in our designs. The congestion ratio  $CR = O(n)$  is not too bad as we have  $n^2$  nodes (in practice we may not have so many distant  $s-t$  pairs to connect). Note that, among popular network architectures, only a tree structure yields a small average weight, but this creates  $CR = \theta(n^2)$  at the root.

The above  $\mu$ -hierarchy networks use our suggested assignment function, however, it is easy to see that the theorem can be extended using other assignment functions (within the uniform mechanism) as long as there is no hot area with high density of routers. Also, an interesting question is if the graph diameter of our  $\mu$ -hierarchy networks is asymptotically the same as the routing diameter or smaller. We conjecture that they are asymptotically the same, while if that is true, our designs also deal with the selfish routing issue.

There are practical issues which may complicate the implementation of our fixed link schemes as a computer network. When a fixed link is supposed to be added to a node, what if the host is not ready to serve such a long link (e.g. not enough resources)? What if a 1-block node (which acts as the only hub to the surrounding neighborhood) leaves the network? These problems suggest that back-up procedures need to be put in place, i.e. we need extra effort to deal with practical and dynamic issues. In these types of settings, we can seek a nearby node to take over the role (as an end-point of such long link). Also, this change from the initial configuration needs to be conveyed to at least some local region (by broadcast) or reported to some special server designed for this purpose of keeping current configuration. A removal of a long link (fixed), say a 1-block link, also degrades the communication flows between two corresponding end blocks, although we can repair this by routing the flows to a nearby block with a link to the target block. Details of these fault-tolerance issues are beyond the scope of this paper, but we present as follows our random link schemes which by their nature, are more resilient to dynamic changes and better suited for distributed scenarios.

#### IV. THE RANDOM LINK SCHEMES AND ROUTING

**The basic scheme.** We add to each node  $u$  one random link, which goes to another node  $v$  with probability proportional to  $d^{-\alpha}(u, v)$  for some properly selected  $\alpha > 0$ , where  $d(u, v)$  is the Euclidean distance between  $u$  and  $v$ . This is similar to Kleinberg’s small-world setting in [15]. Denote this scheme by  $R(n, \alpha)$  or  $R(\alpha)$  when the size of the grid is  $n$  by default. By varying parameter  $\alpha$  we can obtain different values of  $D, W, EC$  and  $CR$ <sup>17</sup>. This random scheme can often obtain good  $CR$  for bounded  $W$  and  $D$ , for an appropriate routing strategy. Without a central control, the scheme performs worse

than (but not much) the fixed link scheme, especially on routing.

However, there are important advantages in this scheme. The construction algorithm is simple, suitable for distributed scenarios, and can be easily extended for a more general dynamic setting. Our scheme provides natural ways to deal with dynamic changes (a node enters/leaves), and also is resilient against failures or attacks. A random link scheme can add flexibility (flexible link assignment, ‘redundancy’ of long links) to deal with those dynamic issues. We first present our basic random link scheme, then introduce refinements, and finally discuss practical and dynamic issues. Parts of this work, especially on implementation issues, are still open. Due to space limit, the proofs of our results in this section are omitted but can be seen in [24].

We observe that  $R(\alpha)$  can approximate a fixed link scheme using a  $\mu$ -hierarchy for  $\mu$  close to, but greater than,  $\alpha/4$ . The idea is that, for  $2 \leq \alpha < 4$  we use  $\alpha/4 < \mu < 1$ . Then for the  $\mu$ -hierarchy on the base grid and the corresponding block-tree, any two sibling blocks are likely to be connected by a random link. In fact, except for the lowest levels, the sub-blocks within a parent block are almost certainly connected to each other (by the random links). Hence we can apply the same idea of hierarchical routing as before: decomposing a routing task within a parent block into two routing tasks within two sub-blocks, using a ‘bridge’ - a random link connecting these two. However the problem is, for any node  $u$  in a sub-block  $S$  which wants to find a route to a node  $v$  in  $S$ ’s sibling block  $T$ , *how does  $u$  find such a link bridging  $S$  and  $T$ ?*

##### A. Routing in random link schemes

For any two sibling blocks  $S$  and  $T$ , we appoint a node  $s_T$  within  $S$  to be a router collecting information about these  $S-T$  bridges; similarly a node  $t_S \in T$  collects the same information. Such a deployment can be done by a few directed broadcast ‘waves’. For choosing the positions for  $s_T$  and  $t_S$ , we use our fixed link assignment function, i.e. select these two as the two end-points of a fixed link ( $u_S$  as  $w_1$  and  $v_T$  as  $w_2$  in figure 2). The router assignment and bridge information broadcast are done in the set-up of our network (more in §IV-C.1). Note also that each router needs to keep only  $O(1)$  entries expected<sup>18</sup>.

Now, our routing strategy is simple. To route from a node  $u$  to a node  $v$ , from node  $u$  we need to get to router  $s_T$  first, which knows a bridge link  $(x, y)$  such that  $x \in S$  and  $y \in T$ . We then need to find a route to  $x$  before taking the  $(x, y)$  link to get to  $T$ , where we also route within  $T$  to go from  $y$  to  $v$ . Thus, we have decomposed a routing task at level  $k$  into *three routing tasks at level  $k+1$*  (fig. 3). Except for the lowest levels, we can find such a bridge between two given sibling blocks with high probability. When we fail to do so ( $s_T$  or  $t_S$  reports no such link), we simply use greedy routing to get

<sup>18</sup>Since  $\mu > \frac{\alpha}{4} \geq .5$  implies the needed number of such router nodes is smaller than the size of each block  $S$  or  $T$  and hence, no node needs to act as a double-router, i.e. keeping positions of bridges from  $S$  to more than one sibling blocks

<sup>17</sup>Note these are expected values.

to  $v$ : we are at a deep level, so even a local link walk will not substantially increase the hop-length of the whole routing path.

Thus, we can show that the hop-length of a routing path is  $O(\log^\gamma n)$  with high probability (tending to 1 when  $n$  goes to infinity), for  $\gamma = \log_{1/\mu} 3 + o(1)$ <sup>19</sup>. Note that it is harder to upper bound the routing diameter, i.e. the longest hop-length of the routing paths.

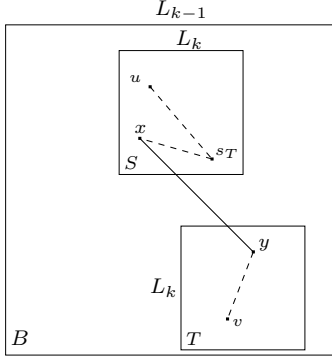


Fig. 3. Basic routing in  $R(\alpha, \mu)$ .

**Definition 4:** A routing network  $R(\alpha, \mu)$ , where  $\alpha > 0, 0 < \mu < 1$ , is a network  $R(\alpha)$  coupled with a routing algorithm which applies the above strategy to the  $\mu$ -hierarchy of  $R(\alpha)$ .

**Theorem 4:** Consider a random link scheme  $R(n, \alpha, \mu)$  for  $2 \leq \alpha < 4, \alpha/4 < \mu < 1$ . The expected performance of the network is as follows (all measures are expected values):

- The scheme achieves congestion ratio  $CR = O(n^{4\mu-2})$ , routing table size  $O(1)$ , routing path length  $O(\log^\gamma n)$  for  $\gamma = \log_{1/\mu} 3 + o(1)$ , and routing diameter  $D = O(\log^c n)$  for  $c = \log_{1/\mu} 3 + \frac{1}{4\mu-\alpha}$ .
- The scheme has average weight  $W = O(n/\log n)$  for  $\alpha = 2$ ,  $W = O(n^{3-\alpha})$  for  $2 < \alpha < 3$ ,  $W = O(\log n)$  for  $\alpha = 3$ , and  $W = O(1)$  for  $3 < \alpha < 4$ .
- By choosing  $\mu$  close enough to  $\alpha/4$  the scheme achieves routing path length  $O(\log^\gamma n)$  for  $\gamma = \log_{4/\alpha} 3 + \epsilon$  for any  $\epsilon > 0$ , and  $W \times CR = O(n^{1+\epsilon})$  for any  $\epsilon > 0$ .

The theorem (especially, part c) shows that, except for routing path length, our random link scheme can perform almost as well as the fixed link scheme<sup>20</sup> by choosing  $\mu$  close to  $\alpha/4$ . The difference between the performance of these two approaches is controlled by the difference  $\mu - \frac{\alpha}{4}$ , which also implies a ‘redundancy’ of long links (as bridges between neighborhoods) in the random link scheme. This redundancy however offers flexibility and resilience to the network as we will discuss later. Table III provides some specific random link designs with performance.

<sup>19</sup>This contrasts with  $\gamma = \log_{1/\mu} 2$  in our fixed link scheme, where we decompose a routing task at level  $k$  into only *two* routing tasks at level  $k+1$ . But we improve this in §IV-B.

<sup>20</sup>Given the same budget (weight) of added long links for each scheme.

TABLE III  
SOME RANDOM LINK SCHEMES

Schemes	Path length	Weight	Congestion Ratio
$R(2, .51)$	$O(\log^{1.7} n)$	$O(n/\log n)$	$O(n^{.04})$
$R(\alpha, .6)$	$O(\log^{2.2} n)$	$O(n^{3-\alpha})$	$O(n^{.4})$
$2 < \alpha < 2.4$			
$R(\alpha, \frac{\alpha}{4} + .05)$	$O(\log^{3.9} n)$	$O(n^{3-\alpha})$	$O(n^{\alpha-1.9})$
$2 < \alpha < 2.9$			
$R(3, .8)$	$O(\log^5 n)$	$O(\log n)$	$O(n^{1.2})$
$R(3.1, .8)$	$O(\log^5 n)$	$O(1)$	$O(n^{1.2})$

$R(\alpha, \mu)$  is as in definition 4.

### B. Shorter routing paths for higher $\mu$

Theorem 4a) states that  $R(n, \alpha, \mu)$ , where  $2 \leq \alpha < 4$  and  $\alpha/4 < \mu < 1$ , has routing path length  $O(\log^\gamma n)$  for  $\gamma = \log_{1/\mu} 3 + o(1)$ , however,  $\gamma$  can be improved (reduced) at higher values of  $\mu$ .

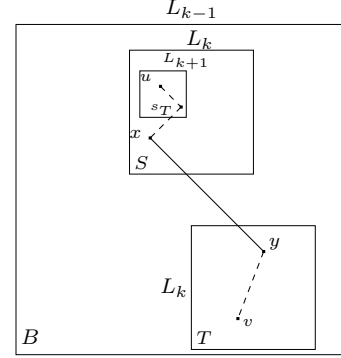


Fig. 4. Routing in  $R(\alpha, \mu)$  with improvement for  $\frac{\sqrt{5}-1}{2} < \mu < 4$ .

Consider a block  $S$  at level  $k > 0$  with its parent block  $B$  (at level  $k-1$ ). Let  $L_k$  and  $L_{k-1}$  denote the sizes of  $S$  and  $B$  respectively:  $L_k = L_{k-1}^\mu$ . For any of  $S$ 's sibling blocks  $T$  there is a router node  $s_T \in S$  keeping the positions of the  $S-T$  bridges. There are  $\frac{L_{k-1}}{L_k} - 1 \approx L_k^{\frac{1-\mu}{\mu}}$  such sibling blocks and we need that many routers (in  $S$ ) accordingly. For  $\mu > .5$  and not close to  $.5$ , observe that  $L_k \gg L_k^{\frac{1-\mu}{\mu}}$ , which means the router nodes are few and far between<sup>21</sup>. Thus, we can have a (large) number of routers which know the random links from  $S$  to  $T$ . For  $\mu$  large enough, we can even appoint one router node per each of  $S$ 's child blocks. For this particular deployment (see figure 4), if we have  $\mu^2 \geq 1 - \mu$ , i.e.  $\mu \geq \frac{\sqrt{5}-1}{2} \approx .618$ , then the size of a  $S$ 's child block  $L_{k+1} = L_k^{\mu^2}$  is greater than the number of  $S$ 's sibling blocks, and therefore, the routing table size at each router is still  $O(1)$ . It is clear that now we can break a routing task at a level  $k$  into two routing tasks at level  $k+1$  and one at level  $k+2$  (for getting to the router node at the current  $S$ 's child block).

<sup>21</sup>They are uniformly distributed by using our uniform assignment function

*Theorem 5:*  $R(n, \alpha, \mu)$  for  $2\alpha < 4$  and  $1 > \mu > \min\{\alpha/4, \frac{\sqrt{5}-1}{2}\}$  achieves routing table size  $O(1)$  and routing path length  $O(\log^\gamma n)$  for  $\gamma = \log_{1/\mu} 2.42$ .

We can refine this approach to improve on  $\gamma$  for higher  $\mu$ . In general, for any  $j \geq 2$ , for  $1 > \mu > x_0$  where  $x_0$  is the solution of  $f(x) = x^j + x - 1 = 0, 0 < x_0 < 1$  (<sup>22</sup>), we can break a routing task at level  $k$  into two routing tasks at level  $k+1$  and one at level  $k+j$ . This is a substantial improvement compared to the basic scheme, where we break that into three routing tasks all at level  $k+1$ . As  $\mu$  goes to 1 and  $j$  tends to infinity, our  $R(\alpha, \mu)$  will perform closely to the corresponding fixed-link scheme (with same  $\mu$ ), where we break a routing task at level  $k$  into just two routing tasks at level  $k+1$ .

### C. On practical and dynamic issues

Here we discuss extending our basic model towards a practical dynamic network setting. Implementation issues are left for future work.

First, we consider extensions to our grid setting. Motivated by wireless sensor networks, suppose the nodes are placed uniformly and randomly in an  $n \times n$  square area. The expected density of these (sensor) nodes is a constant (usually, 1) per unit square (with distance one as the transmission radius of a sensor node). Thus, although a node may not be placed at a  $(n \times n)$  grid node, we can use its closest grid node for the purpose of measuring distance and random link generation<sup>23</sup>. The lattice distance  $d(u, v)$  is thus, the lattice distance between the two grid nodes closest to  $u$  and  $v$ . A random link from a node  $u$  is generated by choosing a grid node  $v'$  with probability proportional to  $d^{-\alpha}(u, v')$  and then find the existing node  $v$  closest to  $v'$  <sup>24</sup>. It is not hard to see that such a network approximates our initial random link (Kleinberg's) setting and our routing schemes (data structures and algorithms) can also be adapted to this present one.

On both the events of a node's entering or leaving the network we need to inform the appropriate router node about the (possible) addition or deletion of a random link. It can be done by simply sending a message to that router node. In the case of having multiple related router nodes (as in our improved routing mechanism for higher  $\mu$ ), the event's node can trigger a local broadcast within the smallest block covering the random link. A leaving router node also has to ask a nearby node to take over its role. To detect node shut-down without warning, we can install a standard mutual signalling (periodically) between nodes of the same neighborhood. Note that the dropped nodes can create a hole around a point where a router node is supposed to be. Since seeking a router near a hole may be hard (from the other side of the hole), all the surrounding nodes (within the covering block) should store this random link information during a random link broadcast.

<sup>22</sup>This has a unique solution since  $f$  increases from under zero ( $-1$ ) at  $x = 0$  to above ( $1$ ) at  $x = 1$ .

<sup>23</sup>So, we often use the term 'sensor nodes' to distinguish from grid nodes.

<sup>24</sup>A re-generation to find different  $v'$  can be considered if no sensor node nearby or any other reason.

1) *Setup phase (Bridge-info broadcast):* The setup phase has a small communication overhead. To disseminate the bridge (random link) information, a standard broadcast/relay scheme for  $2D$  mesh structure can be used, e.g. in [26], however with the following refinement. Using our uniform assignment function, we can compute, for each pair of two sibling blocks  $S$  and  $T$ , the two positions ( $s_T \in S$  and  $t_S \in T$ ) to appoint the two routers. To complete the setup phase we need only 4 directed broadcast 'waves'<sup>25</sup>. Now, each particular  $(u, v, s_T)$  message piece is included in a proper (of the 4) directed broadcast, but is erased from that when it reaches the its target  $s_T$ . This helps to limit the scope being flooded with this piece.

2) *On a very limited long link weight.:* For wireless sensor networks, we note that we may be further limited in adding long links: we may not be capable to add one long link per unit square (or to a sensor node) but rather, one link per a block (of area) of a certain size  $d$ . However it is easy to see that we can carry over all obtained results to this new setting: we instead think of using super nodes as blocks of size  $d$  and a local link now weighs  $d$  instead of 1; thus, we add a multiple factor  $d$  to our bounds on routing path length. We mention more on this in our evaluation section.

### D. Comparison to our fixed link scheme.

As noted before, on the weight-congestion trade-off, the random link scheme performs somewhat worse than the fixed link scheme, due to the 'redundancy' of long links. However this redundancy helps with the problem of link damage. The random assignment approach leads to the need for a mechanism for link (position) look-up and update. This mechanism adds only  $O(1)$ -size routing tables to some nodes and does not cause much communication overhead (a few global broadcasts initially; see [24] for details), but causes significantly slower routing, compared to the fixed link approach where the link deployment is initially fixed and globally known. However this mechanism, as we have just seen, offers natural flexibility to deal with practical and dynamic issues.

While the fixed link approach requires a global deployment with care about partitioning and assigning fixed links between blocks, the random link approach only needs to check at the two ends of a random link (the distribution function can be computed off-line without knowledge about other nodes), and hence is suitable to distributed scenarios.

## V. NUMERICAL EVALUATION

We present a preliminary numerical evaluation for the fixed link schemes. We consider four specific schemes on three different scales from 1000 nodes (on a  $33 \times 33$  square) to 1,000,000 nodes (table IV).

Our theorems in fixed link schemes (summarized in table II) suggest that the opt-W schemes have optimal weight (asymptotically) but with a larger diameter, the opt-D-CR

<sup>25</sup>Directed from each corner of the square, each node  $u \in S$  relays the broadcast plus its possible bridge-info (link  $(u, v)$  with  $v \in T$ ) and router  $s_T$ 's position to the neighbors towards the chosen direction.

TABLE IV  
COMPARISON OF SOME FIXED LINK SCHEMES

schemes	Diameter (D)	weight (W)	Congestion Ratio ( $\approx$ )	$W \times CR$
n= 33; 1,000 nodes				
W-CR, $\delta = .1$	8	14.1	23	328
W-CR, $\delta = .4$	8	7.8	8	64
opt-W, $\delta = .1$	16	5.3	33	177
opt-D-CR	4	9.0	1	9
n= 100; 10,000 nodes				
W-CR, $\delta = .1$	16	18.5	63	1166
W-CR, $\delta = .4$	8	11.1	16	176
opt-W, $\delta = .1$	32	6.8	100	677
opt-D-CR	8	21.4	1	21
n= 1000; 1,000,000 nodes				
W-CR, $\delta = .1$	16	9.8	501	4910
W-CR, $\delta = .4$	16	20.1	63	1269
opt-W, $\delta = .1$	128	10.4	1000	10389
opt-D-CR	8	114.3	1	114.3

Scheme W-CR:  $(\frac{3-\delta}{4}, \frac{1}{2}, C)$ -hierarchy

Scheme opt-W:  $(\frac{3}{4}, \frac{3+\delta}{4}, 1)$ -hierarchy

Scheme opt-D-CR:  $\frac{1}{2}$ -hierarchy

scheme has optimal routing diameter and congestion ratio but with near maximum weight, while the W-CR schemes balance all these issues: near optimal routing diameter and a near optimal trade-off  $W \times CR = O(n)$ .

Our numerical evaluation confirms that and shows that a W-CR scheme with a medium  $\delta$  ( $= .4$ ) makes a nice balance (all small) between all these measures ( $D, W$  and  $CR$ ). Compared to Helmy's experiments [11], the performance of the considered schemes is reasonable<sup>26</sup> in a medium scale (1000 nodes) but very efficient in a large scale (10,000 and 1,000,000 nodes). It is clear that our schemes perform well for  $n \gg \log n$ .

Note also that the opt-D-CR looks attractive for up to 10,000 nodes as it minimizes diameter, congestion ratio, and  $W \times CR$ , while still being competitive in weight. Though this scheme has higher theoretical ( $O(\log \log n)$ ) node degree, even for  $n = 10^9$  (so  $10^{18}$  nodes)  $\log \log n < 5$ . Thus, opt-D-CR seems a good choice unless small weight is a critical factor.

In the case of very limited resource (§IV-C.2), e.g. we can only add one long link to each  $10 \times 10$  square block, the performance of our schemes for the  $1000 \times 1000$  grid will be reduced to that (or similarly) for the  $100 \times 100$  grid.

<sup>26</sup>Helmy only reduces the routing path length by a half, using long links of length almost half of the size of square area. See §I-B.

## VI. FUTURE WORK: MORE GENERAL MODELS

The uniform model we use so far is often suitable for adding long links to a wireless network, but for other applications, the node-to-node communication demands are typically non-uniform, and the nodes may be placed arbitrarily on the plane. In this section we consider some of these more general models and suggest possible solution approaches.

Consider a model with non-uniform node-to-node demands (so some nodes may be busy and others slow) which are fixed and known and where nodes can be placed arbitrarily on the plain (instead of a grid as before). We want to satisfy all the node-to-node demands, or if not possible, satisfy a maximum fraction (called *throughput*, defined shortly) of each of them equally<sup>27</sup>. Since the network now has a wide range of demands, we compensate by allowing some links to have higher capacity.<sup>28</sup> A natural way to model the construction cost is to consider that the weight (cost) of each link is the product of its length and capacity (there are also other possible alternatives which may be more precise for a specific setting). Our goal is now to add long links with proper capacity (for a given budget which limits the total weight) so that we can maximize the throughput.

Our new problem(s) can be formulated using multicommodity network flows. We assume a set of  $n$  nodes  $V$  on the plane and each edge  $e \in E$  connecting two vertices has Euclidean length  $d_e$ . We know these distances  $d_e$ , the flow demands  $D_{uv}$  between all pairs of nodes  $u, v \in V$ , and we assign infinite capacity to all edges. A flow  $F$  in graph  $G$  has *throughput*  $f$  if  $f$  is the maximum value such that at least a fraction  $f$  of the demands, i.e.  $f \times D_{ij}$  for all  $1 \leq i \neq j \leq n$ , are satisfied simultaneously (see, e.g., [18], [27] for more background). While maintaining short routes, we consider simultaneously optimizing the throughput  $f$  and the cost (weight) function

$$w = \sum_{e \in E} F(e) \times d_e. \quad (4)$$

For a given budget  $C$ , we want to construct a flow of short routes such that  $w \leq C$  while maximizing  $f$ .

This version of the minimum-cost multicommodity flow problem which constructs a flow  $F$  to meet the flow demands while minimizing cost  $w$  can be solved using Linear Programming. There are also faster approximation algorithms [14]. A flow for this infinite capacity graph is easily transformed into a network design by creating long links for each edge with non-zero capacity, and having its capacity be the amount used in the flow (or rounded up to the next normal capacity value for a link). Once we have a flow with (near) optimal throughput, it can be transformed into another good one (with a slightly smaller throughput) using short routes only [17]. Combining these algorithms we can develop approximation

<sup>27</sup>For example, a flow has throughput .80 if the flow satisfies at least 80% of each demand, and some demand is satisfied exactly 80%.

<sup>28</sup>We could add many more (uniform capacity) links to a high demand pair of nodes but this conflicts with our requirement of bounded degree. Two parallel unit capacity links of length  $l$  are comparable to one link of capacity two and of length  $l$ .

algorithms to create flows which maintain a near-optimum trade-off between throughput and cost while using short routes. However, these solutions may be slow to compute and a (flow) solution obtained here may suffer a number of drawbacks: it may split a stream of packets into many paths and the paths are produced with little structure (so, requiring explicit representations of all routes). Also, the solutions are likely to be fragile: a single change (e.g. in the demands) may require substantial changes to the routes.

Our design approach for the grid setting may still be useful if we assume certain structural properties of the base graphs, e.g. properties of a Euclidean metric or of a bounded-growth graph. Besides the main target of  $f-w$  trade-off, we also want the flexibility to adapt our constructions to additional issues, such as a short path requirement or dynamic fault-tolerance issues. More specifically, we hope to extend our ideas and techniques for choosing long links which lead to short routing paths. Future work here could provide a more complete picture of the maximum capacity of a general hybrid ad-hoc network (beyond Gupta and Kumar's work in wireless networks [10]): scaling laws for the network throughput as a function of the weight of the added long links and the number of the nodes.

More specifically, we suggest that our prior approach of using a partitioning hierarchy can still be used here. Links are placed into layers and as our results for the basic model suggest, most of the budget needs to go to the top layers (so the links there are the longest and highest capacity) but we need to make sure that each lower layer has enough to serve the top links. We can split the budget between layers in some specific manner and then deal with each layer as a separate flow system (once the next higher layer is determined). As before, tuning the parameters of the hierarchy can help to optimize the system.

It may also be possible to extend our approach even beyond the Euclidean setting. Recent research in bounded metrics (e.g. see [9]) suggests that several key properties in Euclidean metrics are still true in growth-restricted metrics<sup>29</sup>. For example one can still use a partitioning hierarchy in growth-restricted metrics [5].

#### ACKNOWLEDGMENT

This work was supported by NSF grant ANI-04-35525 and NSF grant 0520190. We would like to thank Xin Liu and Biswanath Mukherjee for making a number of helpful comments and suggestions.

#### REFERENCES

- [1] I. Abraham and D. Malkhi, "Compact routing on euclidean metrics," in *Proc. of ACM Symp. on Princ. of Dist. Comp. (PODC)*, 2004.
- [2] I. Abraham, D. Malkhi, and O. Dobzinski, "Land: Stretch  $(1+\epsilon)$  locality-aware networks for dhTs," in *Proc. of ACM Symp. on Discrete Algorithms (SODA)*, 2004.
- [3] S. Arya, G. Das, D. Mount, J. Salowe, and M. Smid, "Euclidean spanners: short, thin, and lanky," in *Proc. of ACM Symp. on Theory of Computing (STOC)*, 1995.

<sup>29</sup>E.g., a metrics with a dimensionality  $\beta$ , where the number of nodes in a ball of radius  $r$  is bounded by  $O(r^\beta)$ .

- [4] R. Banner and A. Orda, "Multipath routing algorithms for congestion minimization," in *NETWORKING*, Waterloo, Canada, 2005, pp. 536–548.
- [5] H. Chan, A. Gupta, B. Maggs, and S. Zhou, "On hierarchical routing in doubling metrics," in *Proc. of ACM Symp. on Discrete Algorithms (SODA)*, 2005.
- [6] C. Chang, T. Sung, and L. Hsu, "Edge congestion and topological properties of crossed cubes," *IEEE Trans. Parallel Distrib. Syst.*, vol. 11, no. 1, Jan. 2000.
- [7] C. Fiduccia and P. Hedrick, "Edge congestion of shortest path systems for all-to-all communication," *IEEE Trans. Parallel Distrib. Syst.*, vol. 8, no. 10, Oct. 1997.
- [8] J. Gao and L. Zhang, "Tradeoffs between stretch factor and load balancing ratio in routing on growth restrict graphs," in *Proc. of ACM Symp. on Princ. of Dist. Comp. (PODC)*, 2004.
- [9] A. Gupta, R. Krauthgamer, and J. R. Lee, "Bounded geometries, fractals, and low-distortion embeddings," in *IEEE Symp. on Found. of Comp. Sci. (FOCS)*, 2003, pp. 534–43.
- [10] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Trans. Inform. Theory*, vol. 46, no. 2, pp. 388–404, 2000.
- [11] A. Helmy, "Small worlds in wireless networks," *IEEE Commun. Lett.*, vol. 7, no. 10, pp. 490–492, Oct. 2003.
- [12] M. Heydemann, J. Meyer, and D. Stotterau, "On forwarding indices of networks," vol. 23, pp. 103–123, 1989.
- [13] X. Jia, D. Li, and D. Du, "Qos topology control in ad hoc wireless networks," in *IEEE INFOCOM*, 2004.
- [14] A. Kamath, O. Palmon, and S. Plotkin, "Fast approximation algorithm for minimum cost multicommodity flow," in *Proc. of ACM Symp. on Discrete Algorithms (SODA)*, 1995.
- [15] J. Kleinberg, "The small-world phenomenon: An algorithmic perspective," in *Proc. of ACM Symp. on Theory of Computing (STOC)*, 2000.
- [16] L. Kleinrock and F. Kamoun, "Hierarchical routing for large networks, performance evaluation and optimization," *Computer Networks*, vol. 1, no. 3, pp. 155–174, Jan. 1977.
- [17] P. Kolman and C. Scheidele, "Improved bounds for the unsplitable flow problem," in *Proc. of ACM Symp. on Discrete Algorithms (SODA)*, 2002, pp. 184–193.
- [18] T. Leighton and S. Rao, "An approximate maxflow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms," in *IEEE Symp. on Found. of Comp. Sci. (FOCS)*, 1988, pp. 422–431.
- [19] D. Loguinov, V. R. A. Kumar, and S. Ganesh, "Graph-theoretic analysis of structured peer-to-peer systems: Routing distances and fault resilience."
- [20] D. Malkhi, M. Naor, and D. Ratajczak, "Viceroy: A scalable and dynamic emulation of the butterfly," in *Proc. of ACM Symp. on Princ. of Dist. Comp. (PODC)*, 2002, pp. 183–192.
- [21] C. Martel and V. Nguyen, "Analyzing kleinberg's (and other) smallworld models," in *Proc. of ACM Symp. on Princ. of Dist. Comp. (PODC)*, 2004.
- [22] E. Ng and H. Zhang, "Predicting internet network distance with coordinates-based approaches," in *Proc. of ACM Symp. on Parallel Algo. and Arch. (SPAA)*, 2002.
- [23] V. Nguyen and C. Martel, "Analyzing and characterizing small-world graphs," in *Proc. of ACM Symp. on Discrete Algorithms (SODA)*, 2005.
- [24] ———. (2005) Designing networks for low weight, small routing diameter and low congestion. [Online]. Available: <http://www.sif.cs.ucdavis.edu/nguyenvk/>
- [25] A. Reznik, S. R. Kulkarni, and S. Verdu, "A small world approach to heterogeneous networks," *Communications in Information and Systems*, vol. 3, no. 4, pp. 325–348, 2004.
- [26] Z. Sen, "An optimal broadcasting schema for multidimensional mesh structures," in *Proc. of ACM Symp. on Applied Computing*, 2003.
- [27] F. Shahrokhi and D. W. Matula, "The maximum concurrent flow problem," *Journal of the ACM (JACM)*, vol. 37, no. 2, pp. 318–334, 1990.
- [28] R. Srikant, "Models and methods for analyzing internet congestion control algorithms," in *Advances in Communication Control Networks*, ser. Lecture Notes in Control and Information Sciences (LCNCIS), C. Abdallah, J. Chiasson, and S. Tarbouriech, Eds. New York: Springer-Verlag, 2004.
- [29] J. Xu, A. Kumar, and X. Yu, "On the fundamental tradeoffs between routing table size and network diameter in peer-to-peer networks," *IEEE J. Select. Areas Commun.*, vol. 22, no. 1, pp. 151–163, Jan. 2004.