

# Designing Networks for Low Weight, Small Routing Diameter and Low Congestion \*

Van Nguyen and Chip Martel  
{nguyenvk,martel}@cs.ucdavis.edu  
Computer Science, UC Davis, CA 95616

August 17, 2005

## 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.

## 1 Introduction

### 1.1 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 abstract model, this can be realized by other specific measures, e.g. the transmission delay in the Internet, which also forms a metric [34].

---

\*This work was supported by NSF grant CCR-85961

<sup>1</sup>A real-world example (transportation networks of roads connecting a large number of locations) is suggested by Arya et al. [6]. 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.

Table 1: 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. [6]).

*Viceroy*: The Viceroy network (a randomized butterfly)[30] and similar randomized networks [1, 2].

*Ulysses*: The Ulysses network (a randomized butterfly) [42].

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

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

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 [20, 37]. 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 1 compares our trade-off results to existing network designs. Our comparisons assume uniform traffic and uniform 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 [36]). These prior network designs optimize trade-offs between some of the factors, but none looks at all of them

<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  $O(\log n)$ .

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

together. While Euclidean spanners [6] achieve almost (asymptotically) optimal trade-offs between degree, diameter and weight, they perform the worst for congestion <sup>5</sup>. The Viceroy network [30] 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 [42] 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 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 [24], and is different from classical hierarchical network architectures (e.g. Kleinrock and Kamoun’s [25]).

**Towards a more general model (cost-diameter-throughput).** We also consider more general models, where we no longer assume uniform traffic and uniform location distribution. The nodes can be placed arbitrarily on the plane and the communication demands between them can be non-uniform, so some nodes are much ‘busier’ than others. Further, we assume that the long links have varying capacity (rather than a single size thickness as before), and thus, the weight 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). The design issues here, however, change a bit. By assuming that the node-to-node communication demands are fixed and known initially, we can consider our issues in a multicommodity network flow context. For example we can consider a minimum-cost multicommodity flow problem, where we want to find a minimum budget to build connecting links which satisfy all the demands yet the routing diameter is also small as needed.

However, if the demands can not be met by any realistic budget, we need to think of some negotiating issues, where we target possible trade-offs between the total link weight and the network *throughput* (definition in §6). Thus, in this more general setting, we consider trade-offs between weight (cost), routing diameter and throughput. The trade-off between weight-routing diameter -congestion in the basic model can be seen as a special case here.

---

<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 §2.

<sup>6</sup>In §2, we also show a lower bound indicating a small gap (a poly-log of  $n$  factor) between the two bounds

## 1.2 Areas of possible application.

Our results can be used in building hybrid ad-hoc networks. Helmy [20] and independently, Reznik et al. [37] 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. §5 gives a preliminary evaluation of our work in this area.

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 [19] 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 §6.

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.

## 1.3 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 [12, 9]. The minimum edge-congestion of the network (over all routing algorithms) is also known as ‘*the edge forwarding index*’, which is introduced in [21]. Our notion of *congestion ratio* has some relation to a few recent papers. Xu et. al [42] 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 [13] 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 [40].

---

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

<sup>8</sup>For the classical Minimum Steiner Tree with extensions include bounded diameter and/or bounded-degree

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. [42] provide a full treatment to the diameter-degree trade-off. Asymptotically optimal schemes are also provided in [30, 29]. Loguinov et al. provide a graph-theoretic framework to analyze and compare P2P networks on several properties which affect routing and resilience [29]. Although the mentioned P2P networks can be optimal in degree-diameter and are sometime congestion-aware [42], 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 1).

Besides the mentioned problems in wireless ad-hoc networks (§1.2), 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 [22]. The work in [7] considers a multi-path routing algorithm which minimizes congestion using a multi-commodity flow approach. We consider a similar approach in §6.

As mentioned, this work is inspired by our study of small-world models [35, 33], which follows the seminal work by Kleinberg [24]. 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 [30] and Helmy’s hybrid ad-hoc network [20]. We elaborate more on the related work in §7.

**The structure of this paper.** In §2 we present our basic model and initial facts. In §3 we present our theorems on our fixed link scheme, which provide and analyze our trade-offs. In §4 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 §6, 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 §5. Finally, in §7 we add an overview of research fields with results related to our work.

## 2 Basic notions and model

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. Each node has (usually 4) undirected links to its neighbor nodes at distance 1. We consider more general models in §6. 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.

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>9</sup>.

---

<sup>9</sup>Equivalently, a path system can be used here instead as in many previous papers.

## 2.1 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>10</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 congestion over a big link is  $\Omega(n^4/2nDW) = \Omega(n^3/DW)$ ; thus, the edge-congestion  $EC(G, RA) = \Omega(n^3/DW)$ .

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  $\Omega(n^3/DW)/O(n^2D) = \Omega(n/D^2W)$ .  $\square$

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  $O(n)$ .

<sup>10</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  $EC$ .

Our objective is to design network architectures, as a local-contact graph plus a set of additional long links (in a metric space), coupled with a good routing algorithm, such that we can obtain close-to-optimal trade-offs between  $D$ ,  $W$  and  $CR$ . Typically, we require  $D$  to be poly-log (i.e.  $D = O(\log^c n)$  for some constant  $c > 0$ ) and vary  $W$  to see what is the best  $CR$  we can achieve. For example, for a limited ‘budget’ and relatively low traffic, we typically desire  $D = O(\log n)$ ,  $W = O(1)$  and  $CR$  lowest possible. We have found two promising construction schemes for this purpose. We give a basic description of these schemes and show more details of how we can use them to construct satisfied designs in next sections.

**Fixed-link construction.** We use some ideas from our analysis of Kleinberg’s setting. We divide the square  $(0, 0, n - 1, n - 1)$  into blocks: squares of size  $n^\mu$  for some  $\mu \in [.5, 1)$ . For any pair of blocks, we select a node from each (selection rules are considered later, but for now can be seen as a uniformly random selection), and add a link between these two nodes. We further divide the blocks into sub-blocks and add links between any two sub-blocks as before and so on. By varying parameter  $\mu$  we can obtain different values of routing diameter  $D$  and average weight  $W$  as well as the edge-congestion  $EC$ . In fact, we can even change the value of  $\mu$  between different steps of this block-decomposition process. By selectively choosing  $\mu$  for different steps, we can obtain almost optimal solutions with this determined scheme.

**Random-link construction.** 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  $\alpha > 0$ . This is basically Kleinberg’s setting in [24]. By varying parameter  $\alpha$  we can obtain different values of  $D, W, EC$  and  $CR$ . This random scheme can often obtain good  $CR$  for bounded  $W$  and  $D$ , for an appropriate routing strategy. With fewer tuning parameters, the scheme performs worse than (but not much) the fixed link scheme, especially on routing diameter. 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. Especially, with the nature of random networks, the scheme can provide natural ways to deal with dynamic events (such as when a node enters or leaves the network), and also is resilient against failures or attacks.

## 2.2 A network design theme

In fact, this paper and some others mentioned in the related work can be seen as a design theme which has a flavor of building an economical transportation system. Here we try to picture these connections together. We provide a list of optimization problems in an order of increased complexity, and show which model or work can be used as a solution to each of these problems.

**Minimum Diameter.** Given integers  $n, m > 0$ . Construct a network of  $n$  nodes and  $m$  edges to minimize the graph diameter.

To make it look more like a ‘transportation system’ problem, we can think of the following scenario. Given a fixed size area (say a square), consider constructing a system of  $m$  metro routes (like worm holes), where traveling on a route is free (no matter the distance) but a small constant fare  $c$  to get in. We need to minimize the maximum cost to travel between any two distinct points, assuming some constant cost  $C$  for traveling a unit distance on the ground to reach a metro route. It is not hard to see that this problem can be reduced to minimizing diameter. Choosing an  $n < m$ , we just need to put  $n$  nodes evenly-spaced within the area (as a grid’s nodes) and then find the way to draw  $m$  edges between them to obtain a graph of smallest diameter. Then, vary  $m$  and find the minimum diameter between those smallest diameter values. For  $c \ll C$ , it is easy to see we need  $n = m - 1$ .

This ‘worm holes’ context can be used for a wire-wireless scenario where we consider dropping  $m$  wired lines into a field of wireless sensor nodes (assuming a node per unit square density) so that the maximum communication delay between any two nodes can be minimized.

Clearly, the simple star topology (with diameter 2) is a right candidate. However, it suffers substantial congestion at the center node. P2P architectures like Viceroy [30], LAND[1] or Symphony

[31] (using Kleinberg’s idea of additional long links) have slightly larger diameter ( $\log n$ ) but only have edge-congestion  $EC = \theta(\log n)$ .

**Diameter-Cost.** Apart from small diameter, we also need to optimize the construction cost of the network which is the total edge weight (length in a Euclidean setting).

The networks from Bounded-diameter Minimum Sterner Trees (see §7) can be used here, however the obtained tree structures again suffer congestion at the root. P2P architectures like Viceroy, LAND or Symphony, however, cost too much with the unlimited long links.

Now we consider adding the congestion issue.

**Diameter-Cost-Congestion.** Given a fixed budget, say, enough to build  $L$  units of metro length (of an unique capacity), we need to consider small-diameter networks with optimized congestion ratio.

**Diameter-Cost-Throughput.** Given a fixed budget, say, enough to build  $L$  capacity-length units (so, cost is the product of capacity/bandwidth and length), we need to construct small-diameter networks with edges of varying capacity so the network throughput is maximized.

Thus, this paper deals with Diameter-Cost-Congestion and partially address Diameter-Cost-Throughput in §6.

### 2.3 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. It suggest a divide-and-conquer approach, a scalable manner of decomposing a given problem into many similar ones of a smaller size, recursively. 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 the appendix). 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 two (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.

For a given partitioning hierarchy, such a block- tree is determined; any intermediate block can be identified by its level and by the path from the root to the block’s corresponding node, i.e. by the chain of the  $(row, column)$  index of the nodes on the path down from the root. 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. For a  $\mu$ -hierarchy, we assume the block 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  $\mu_2$  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.

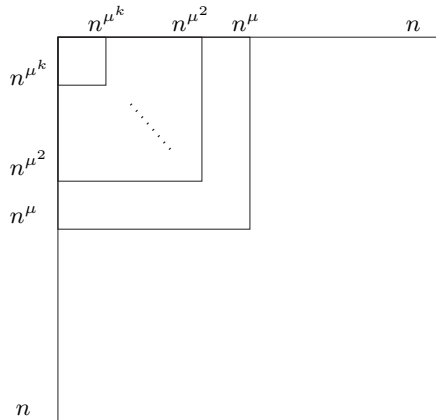


Figure 1:  $\mu$ -hierarchy

### 3 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. This 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, however, has hidden parameters which should reflect 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 and the  $\mu$  value. That is, given fixed sizes of the parent and children blocks, the assignment function is uniform everywhere, not depending on the position of the parent block in the block tree.

#### 3.1 Our routing algorithm.

Let consider an  $\mu$ -hierarchy on a  $n \times n$  grid with block tree of height  $L = \log_{1/\mu} 2 \times \log \log n$ . We suggests 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  $v \in B_2$ ,  $B$  has height  $k$ , i.e.  $B$  is at level  $L - k$  in the block tree. 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).

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>(11)</sup>. Thus, the routing diameter of this network design is  $O(2^L) = O(\log^\gamma n)$  where  $\gamma = \log_{1/\mu} 2$  (noting that  $L = \log_{1/\mu} 2 \times \log \log n$ ). The routing diameter achieves  $O(\log n)$  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 any level; that is the link assignment function can be

<sup>11</sup>A more detailed analysis can be seen in [35].

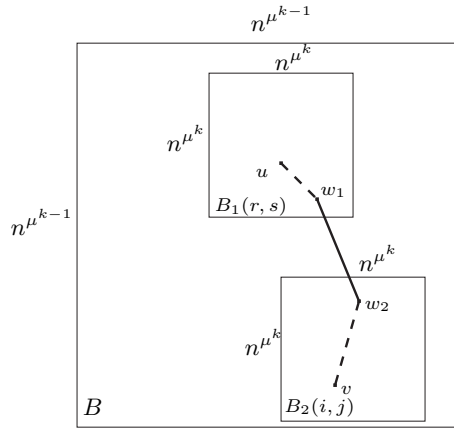


Figure 2: Routing in  $\mu$ -hierarchy.

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

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.

### 3.2 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 §2.3, 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>12</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 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>13</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>14</sup>. For  $\mu < .5$ , the degrees are big (polynomial in  $n$ ) and hence, we only consider using  $.5 \leq \mu < 1$  in our designs.

<sup>12</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  $B$ .

<sup>13</sup>For simplicity, we often assume the fractions here are integers but rounded functions can be used in implementation.

<sup>14</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.

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

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

### 3.3 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. This is 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.

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 and*
- b) *if they dominate in total weight such that  $T_1 = \theta(T)$ , then if  $.5 < \mu < 1$ ,*

$$CR \times W = O(n) \tag{3}$$

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 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 paths 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

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

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

This fact reflects a crucial idea in our designs: to make the 1-block links big enough to dominate in the total weight. We now just need another lemma, which is used to estimate the total weight of long links at each level, before we show 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  $\delta = 4(1 - \mu) - 1$*

- 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  $T_i = \theta(n^{2+\mu^{i-1}\delta})$ .*

*Proof.* For the first level, the number of blocks is  $(\frac{n}{n^\mu})^2 = n^{2(1-\mu)}$ . For each block, there are at least half of all the other blocks which are at least at distance  $n/4$  away, hence the length of each long link connecting such pair is  $\theta(n)$ . Therefore, the total weight of long links at this first level is

$$T_1 = \theta((n^{2(1-\mu)})^2 \times n) = \theta(n^{4(1-\mu)+1}) = \theta(n^{2+\delta}) \quad (4)$$

For the  $i^{\text{th}}$  level, the number of  $i$ -blocks within an  $(i-1)$ -block is  $(\frac{n^{\mu^{i-1}}}{n^{\mu^i}})^2 = n^{2(1-\mu)\mu^{i-1}}$ . Thus, the number of long links connecting these  $i$ -blocks inside a  $(i-1)$ -block (approx. the square of the above number) is  $\theta(n^{4(1-\mu)\mu^{i-1}})$ , while the links has average length  $\theta(n^{\mu^{i-1}})$ . The total weight of long link at level  $i$  is the sum of ( $i^{\text{th}}$ -level) long link weight over the  $(i-1)$ -blocks, which are as many as  $(\frac{n}{n^{\mu^{i-1}}})^2$ , and hence, equals

$$T_i = \theta(n^{4(1-\mu)\mu^{i-1}}) \times \theta(n^{\mu^{i-1}}) \times (\frac{n}{n^{\mu^{i-1}}})^2 = \theta(n^{2+\mu^{i-1}\{4(1-\mu)-1\}}) = \theta(n^{2+\mu^{i-1}\delta}) \quad (5)$$

□

**Theorem 1.** Consider a  $\mu$ -hierarchy with our uniform assignment function and hierarchical routing algorithm and with a block tree of height  $L$ . Define  $\delta = 4(1-\mu) - 1$  (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 roughly all the traffic between each pair of 1-blocks. From lemma 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 now consider the average weight  $W$ .

a) From lemma 5, the total weight of long link in the first level is  $T_1 = \theta(n^{2+\delta})$ , and the total weight of long link at level  $i \leq L$  is  $T_i = \theta(n^{2+\mu^{i-1}\delta})$ . Now for  $i \geq 2$ ,  $\frac{T_i}{T_1} = O(n^{(\mu^{i-1}-1)\delta}) = O(n^{(\mu-1)\delta}) = O(\frac{1}{L})$ . So, the total weight of all long links is  $\theta(T_1) = \theta(n^{2+\delta})$  and hence, the average weight  $W = O(T/n^2) = \theta(n^\delta)$ .

b) Similarly as above, we find  $T_i = \theta(n^2)$  for all  $i = 1..L$ . Therefore, the total weight  $T = \theta(L \times n^2) = \theta(\log \log n \times n^2)$  and hence,  $W = O(T/n^2) = O(\log \log n)$ .

c) Using lemma 5,  $T_i = \theta(n^{2+\mu^{i-1}\delta}) = n^2 \times \theta(n^{\mu^{i-1}\delta})$ . On the other hand, since  $\delta < 0$ ,  $\sum_{i=1}^L n^{\mu^{i-1}\delta} < \sum_{i=1}^L \mu^{i-1}\delta = \frac{(1-\mu)\delta}{1-\mu^{L+1}} = O(1)$ . Thus,  $T = O(n^2)$  and  $W = O(T/n^2) = O(1)$  <sup>(15)</sup>.

□

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 hierarchy levels. Based on this theorem, we consider more refined constructions using varying  $\mu$ , which can achieve better trade-offs.

<sup>15</sup>Note that while part a verifies  $CR \times W = O(n)$  with  $T = O(T_1)$ , part b and c show the other side:  $CR \times W > O(n)$  with  $T_1 = n^{2-|\delta|} = o(T)$ .

### 3.4 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 than that) 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  $CR \times W = O(n)$ .

**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  $CR = O(n^{1-\delta})$ .*

*Proof.* By lemma 5, we have  $T_1 = \theta(n^{2+\delta})$  and  $T_i = o(n^{2+\delta})$  for  $i = 2..C$  ( $\mu = \mu_1$ ).  $T_i$  will be different for  $i \geq C + 1$  ( $\mu = \mu_2$ ). Now, let  $m = n^{\mu_1^C}$ , the size of the  $C$ -blocks. Similarly as with (5), we have

$$T_i = \theta(m^{4(1-\mu)\mu^{i-C-1}}) \times \theta(m^{\mu^{i-C-1}}) \times \left(\frac{n}{m^{\mu^{i-C-1}}}\right)^2$$

where  $i \geq C + 1$  and  $\mu = \frac{1}{2}$ . Thus, for  $i \geq C + 1$  we have  $T_i = \theta(n^2 m^{\mu^{i-C-1}})$ , which is  $\theta(n^2 m) = \theta(n^{2+\mu_1^C})$  for  $i = C + 1$  but  $o(n^{2+\mu_1^C})$  for  $i > C + 1$ . However, note that  $\mu_1^C = \frac{1}{(\frac{1}{\mu_1})^C} < \frac{1}{\frac{1}{\delta} + 1} = \frac{\delta}{1+\delta} < \delta$ . Clearly  $T_i = o(T_1 \times L)$  for all  $i = (C + 1)..L$ , where  $L = O(\log \log n)$ . Also  $T_i = O(T_1)$  for all  $i = 2..C$ . Thus, the total weight of all the long links is  $O(T_1) = O(n^{2+\delta})$  and hence, the average weight of long links  $W = O(n^\delta)$ .

On the other hand, since we only have a constant number ( $C$ ) of top levels with  $\mu = \mu_1$ , the routing paths of this  $(\mu_1, \mu_2, C)$ -hierarchy are mostly within the remaining  $L - C$  levels (a  $\mu_2$ -hierarchy). That is the routing diameter is within a constant multiple of that of the corresponding  $\mu_2$ -hierarchy, which is  $\Omega(\log^\gamma n)$  where  $\gamma = \log_{1/\mu_2} 2 = 1$ . So,  $D = \theta(\log n)$ .

Computing  $EC, CR$  can be done similarly as before (theorem 1). □

However, since we use  $\mu = \frac{1}{2}$  for most levels ( $L - C$  out of  $L$ ), the degree of each node becomes  $\log \log n$  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.

### 3.5 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 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.

$\mu$ -hierarchy	Degree	Routing diameter ( $D$ )	Weight ( $W$ )	Congestion Ratio(CR)
Single $\mu$ -hierarchy				
$\frac{1}{2} < \mu < \frac{3}{4}$	$O(1)$	$O(\log^{2.5} n)$	$O(n^\delta)$	$O(n^{1-\delta})$
$\mu = \frac{1}{2}$	$O(\log \log n)$	$\theta(\log n)$	$O(\frac{n}{\log \log n})$	$O(1)$
$\mu = \frac{3}{4}$	$O(1)$	$O(\log^{2.5} n)$	$O(\log \log n)$	$O(n)$
$\frac{3}{4} < \mu < 1$	$O(1)$	$O(\log^\gamma n),$ $\gamma = \log_{1/\mu} 2$	$O(1)$	$O(n^{3+ \delta }),$ $\delta = 4(1 - \mu) - 1$
Complex hierarchy				
$(\frac{3-\delta}{4}, \frac{1}{2}, \lceil \log_{\frac{1}{\delta}} (\frac{1}{\delta} + 1) \rceil)$ -hierarchy,				
Variant 1	$O(\log \log n)$	$O(\log n)$	$O(n^\delta)$	$O(n^{1-\delta})$
Variant 2	$O(1)$	$O(\log n \sqrt{\log \log n})$	$O(n^\delta)$	$O(n^{1-\delta})$
$(\frac{3}{4}, \mu, 1)$ -hierarchy	$O(1)$	$O(\log^{2.5} n)$	$O(1)$	$\theta(n)$

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 theorem 2: 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  $D = O(\log^{2.5})$ .

### 3.6 Concluding remarks

We summarize all the trade-off results in table 3.6. 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 at the extreme.

For  $\mu = \frac{1}{2}$  (hence,  $\delta = 1$ ), as in §3.4, 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 [31, 2].

The construction in §3.5 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 only 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.

<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 generous.

We conjecture that they are asymptotically the same, while if that is true, our designs also deal with the selfish routing issue.

## 4 The random schemes and routing

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 certain node, what if the host is not ready to serve such a long link (e.g. not enough resource)? 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 near-by 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 an 1-block link, also degrades the communication flows between two corresponding end blocks, although we can repair by routing the flows to a nearby block with the hope it has a bridge to the target block.

A random link scheme can add flexibility (flexible link assignment, ‘redundancy’ of long links) to deal with those dynamic issues. Random link schemes can be much more resilient to dynamic changes in a network. Our random schemes are also easier to implement in a distributed scenario, although the performance of routing is lower than for fixed link schemes.

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.

**The basic scheme.** Recall that the basic scheme is to add to each node  $u$  (of the  $n \times n$  grid) one random link, which goes to another node  $v$  with probability proportional to  $d^{-\alpha}(u, v)$  for some properly selected  $\alpha > 0$ . Denote this scheme by  $R(n, \alpha)$  or  $R(\alpha)$  when the size of the grid is  $n$  by default. We now show 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  $2 < \alpha < 4$  and  $\alpha/4 < \mu < 1$ , 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. The proof of the next lemma can be seen in the appendix.

**Lemma 6.** *Consider  $R(\alpha)$ ,  $2 < \alpha < 4$  and a  $\mu$ -hierarchy for  $\alpha/4 < \mu < 1$ . For any two sibling blocks  $S$  and  $T$  (chosen without knowledge about the random links) at level  $k$ , the probability of having at least one random link  $(s, t)$  with  $s \in S, t \in T$  is*

$$Pr[S \rightarrow T] \geq 1 - e^{-cL^{4\mu-\alpha}}$$

for some constant  $c > 0$ , where  $L = n^{\mu^{k-1}}$  is the size of  $S$  and  $T$ 's parent block (note that  $4\mu - \alpha > 0$ ).

Given  $2 < \alpha < 4$ , lemma 6 suggests that for a properly chosen  $\mu$  and for a level  $k$  not too deep, the blocks at this level are almost 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 an  $S$ 's sibling blocks  $T$ , how does  $u$  find such a link bridging  $S$  and  $T$ ?

### 4.1 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  $t_S \in T$  to collect the same information. Such a deployment

can be done by a few directed broadcast ‘waves’: starting from one corner of the grid, upon receiving broadcast from a neighbor each node relays the broadcast plus its random link target to the next nodes in a chosen direction (say, from left bottom corner to right top one). During each wave, each appointed node can collect the necessary random links information. For choosing (the positions for)  $s_T$  and  $t_S$ , it is natural to use our (fixed link) assignment function, i.e. to 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 (random link) information broadcast are done in the set-up of our network (more in). Note also each router need to keep only  $O(1)$  entries expected (routing table size)<sup>17</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  $u_T$  first, which is to provide 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<sup>18</sup> routing tasks at level  $k+1$  (see figure 3). At most levels but the ones very near the bottom, 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 to  $v$ : we are at a deep level, so even a local link walk will not substantially increase the length of the whole routing path.

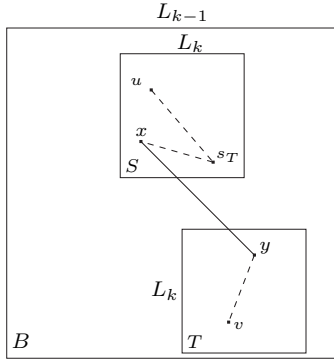


Figure 3: Basic routing in  $R(\alpha, \mu)$ .

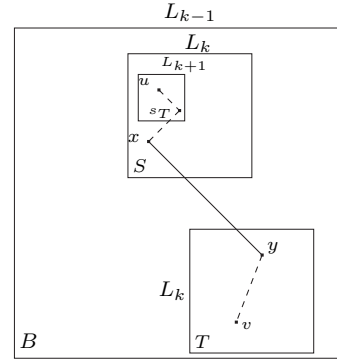


Figure 4: Routing in  $R(\alpha, \mu)$  with improvement for  $\frac{\sqrt{5}-1}{2} < \mu < 4$ .

For  $2 < \alpha < 4$  and  $\alpha/4 < \mu < 1$  we use  $R(\alpha, \mu)$  to denote a network  $R(\alpha)$  coupled with a routing algorithm which applies to the above strategy for this  $\mu$ -hierarchy.

**Lemma 7.** *Consider a network  $R(\alpha, \mu)$  where  $2 < \alpha < 4$  and  $\alpha/4 < \mu < 1$ . The length of a routing path is  $O(\log^d n)$  with high probability (tending to 1 when  $n$  goes to infinity), for  $d = \log_{1/\mu} 3 + \epsilon$ , for any  $\epsilon > 0$ . This is still true for  $d = \log_{4/\alpha} 3 + \epsilon$  for any  $\epsilon > 0$ , by choosing a proper  $\mu$ .*

*Proof(sketch).* A routing path by our routing algorithm consists of some random links (bridges between sibling blocks) and local link paths between them. For fixed  $\beta > 0$ , we consider all levels  $k \geq k_0$  where  $k_0$  is the greatest integer such that  $L_{k_0} = n^{\mu^{k_0-1}} > (\beta^2 \log \log n)^{\frac{1}{4\mu-\alpha}}$ . As can be seen later, we can choose  $\beta$  large enough so that at any such level, we are (almost surely) guaranteed to have the needed bridges to decompose a routing task into 3 pieces of the next level. This decomposition can be made recursively until we get to level  $k_0$ .

The length of a routing path between two nodes can be simply upper bounded (probabilistically) by the length of this path: using random links (as bridges) when we are at the levels  $\geq k_0$  and using local

<sup>17</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 need to act as a double-router, i.e. keeping positions of bridges from  $S$  to more than one sibling blocks

<sup>18</sup>Compared to only two in our fixed link scheme. The routing diameter, therefore, is significantly longer. However we also show some improvement later.

links only for the levels under  $k_0$ . Without loss of generality, we assume that such our decomposition tree of routing tasks (of height  $k_0$ ) is full. It is easy to see that the number of all long links in such a path  $M = O(3^{k_0})$ . On the other hand,  $n^{\mu^{k_0}} \geq 2$ , so  $k_0 \leq \log_{\frac{1}{\mu}} \log n$  and hence  $M = O(\log^{\log_{1/\mu} 3} n)$ . Also  $M = O(\log^{\log_{4/\alpha} 3 + \epsilon/2} n)$  for any  $\epsilon > 0$ , by choosing  $\mu$  close enough to  $\alpha/4$ .

Since any local link walk between two bridges has length bounded by  $L_{k_0} = O((\beta^2 \log \log n)^{\frac{1}{4\mu - \alpha}})$ , which is  $O(\log^{\epsilon/2} n)$  for any  $\epsilon > 0$ , for  $n$  large enough. Thus the length of a routing path (with high probability) is upper bounded by  $O(M \times L_{k_0}) = O(\log^{\log_{1/\mu} 3 + \epsilon} n)$  for any  $\epsilon > 0$ . Also it is bounded by  $O(\log^{\log_{4/\alpha} 3 + \epsilon} n)$  for any  $\epsilon > 0$ , by choosing  $\mu$  close enough to  $\alpha/4$ .

Now, for any two sibling blocks  $S$  and  $T$  at level  $k \geq k_0$ , from lemma 6, for  $n$  large enough, we have  $Pr[S \rightarrow T] \geq 1 - \log^{-\beta} n$ . We can choose  $\beta$  large enough such that  $\beta \gg \log_{4/\alpha} 3 > \log_{1/\mu} 3$ . With probability  $(1 - \log^{-\beta} n)^M$ , which is almost 1, we are guaranteed that we always find needed bridges when we are not at a level under  $k_0$ . □

However, it is harder to upper bound the routing diameter, i.e. the longest length of the routing paths. We need to keep  $Pr[S \rightarrow T]$  high not only for a given two sibling block  $S$  and  $T$  at a level  $k$ , but also high for every such pair of sibling blocks at level  $k$ . As a result, we can not maintain such a property down to a level as deep as  $k_0$  in lemma 7. We have to stop our routing decomposition at some level  $k_1 < k_0$  with  $L_{k_1}$  at least  $\Omega(\log^{\frac{1}{4\mu - \alpha}} n)$ . This adds to the increase of the upper bound of routing diameter:  $D = O(\log^d n)$  where  $d = \log_{\frac{1}{\mu}} 3 + \frac{1}{4\mu - \alpha}$ . See the appendix for more detail.

We now consider the average weight (expected length) of the random links. Let  $C_u$  be the normalization coefficient at a node  $u$ . Consider  $2 < \alpha < 4$ . Since there are  $\theta(k)$  nodes at distance  $k$  from a given node  $u$

$$\frac{1}{C_u} = \theta(\sum_{k=1}^n k \times k^{-\alpha}) = \theta(1)$$

So, the expected length of a random link is,

$$\theta(\sum_{k=1}^n k \times k \times C_u k^{-\alpha}) = \theta(\sum_{k=1}^n k^{2-\alpha}),$$

which is  $\theta(n^{3-\alpha})$  for  $2 < \alpha < 3$ ,  $\theta(\log n)$  for  $\alpha = 3$ , and  $\theta(1)$  for  $3 < \alpha < 4$ .

For  $\alpha = 2$ ,  $\frac{1}{C_u} = \theta(\sum_{k=1}^n k^{-1}) = \theta(\log n)$ . The expected length of a random link is  $\theta(\sum_{k=1}^n C_u \times k^{2-2}) = \theta(n/\log n)$ .

Similarly as with our fixed link scheme, the 1-block links are the most congestive and virtually equally between themselves. Thus, it is easy to see that the scheme  $R(\alpha, \mu)$  has congestion ratio  $CR = O(n^{4\mu-1})$ . We now combine all these facts into the following theorems.

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

- a) *The scheme achieves congestion ratio  $CR = O(n^{4\mu-2})$ , routing table size  $O(1)$ , routing path length  $O(\log^d n)$  for  $d = \log_{1/\mu} 3 + o(1)$ , and routing diameter  $D = O(\log^c n)$  for  $c = \log_{1/\mu} 3 + \frac{1}{4\mu - \alpha}$ .*
- b) *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  $3 < \alpha < 4$ .*
- c) *By choosing  $\mu$  close enough to  $\alpha/4$  the scheme achieves routing path length  $O(\log^d n)$  for  $d = \log_{4/\alpha} 3 + \epsilon$  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 only little worse than the fixed link scheme<sup>19</sup> by choosing  $\mu$  close to  $\alpha/4$ . The difference between these two performance is marginally controlled by the difference  $\mu - \alpha/4$ , 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.

<sup>19</sup>Given an (asymptotically) equal weight for each scheme.

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

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

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>20</sup>. Thus, we suggest that, instead of one, we can have a (large) number of routers in charge of each sibling block  $T$  (i.e. to know the random links from  $S$  to  $T$ ); 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  $L_{k+1} = L_{k-1}^{\mu^2}$  is greater than the number of  $S$ 's sibling blocks, and therefore, the routing table size at each router is still  $O(1)$ .

**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  $O(\log^d n)$  for  $d = \log_{1/\mu} 2.42$ .

*Proof (Sketch).* 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). Therefore we can relate to the series  $\{u_k | u_{i+1} = 2u_i + u_{i-1} + 2\}$  to upper-bound the number of bridges used (at each sub-tree of height  $k$  in the routing decomposition tree). On the other hand this series has solution  $u_k = \theta((\sqrt{2} + 1)^k)$  and hence similarly as in lemma 7, the number of bridges (long links) used in a routing path is  $O(c^{k_0})$  where  $c = \sqrt{2} + 1 < 2.42$  and  $k_0 = \log_{1/\mu} \log n$ . We then continue as in lemma 7.  $\square$

We can refine this approach to improve on  $d$  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>21</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 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  $k + 1$ .

## 4.3 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>22</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>23</sup>. It is not hard to see that such a network approximates

<sup>20</sup>They are uniformly distributed by using our uniform assignment function

<sup>21</sup>This has a unique solution since  $f$  increases from under zero ( $-1$ ) at  $x = 0$  to above ( $1$ ) at  $x = 1$ .

<sup>22</sup>So, we often use the term 'sensor nodes' to distinguish from grid nodes.

<sup>23</sup>A re-generation to find different  $v'$  can be considered if no sensor node nearby or any other reason.

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 (figure 5).

### 4.3.1 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 [38], 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’: 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 (figure 6). 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: the  $(u, s_T)$  piece is relayed by any node on the rectangle with one diagonal  $u - s_T$  (during the proper directed broadcast) except  $s_T$ , but is dismissed beyond this area.

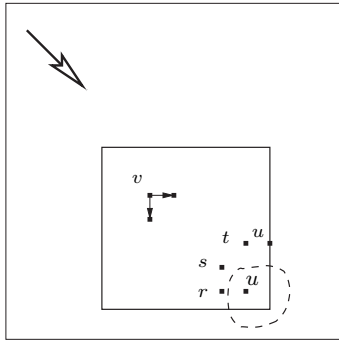


Figure 5: Broadcasting to a hole. The broadcast targets at grid node  $u$  inside a ‘hole’. All the nodes ( $r, s, t$  and  $u$ ) surrounding the hole but inside the block store the random link info from  $v$ .

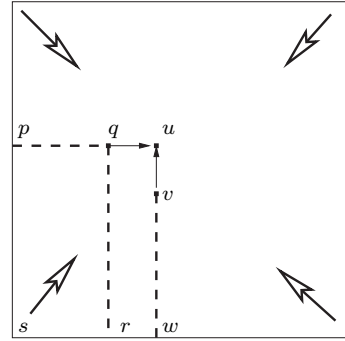


Figure 6: Initial broadcast to all router nodes. Possible overlapping avoidance:  $q$  sends  $u$  only the info from rectangle  $pqrw$  while  $v$  only segment  $vw$ .

### 4.3.2 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.

#### 4.4 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 appendix 4.3.1), 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.

### 5 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 2).

Our theorems in fixed link schemes (summarized in table 3.6) suggest that the opt- $W$  schemes have optimal weight (asymptotically) but with a larger diameter, the opt-D-CR 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 [20], the performance of the considered schemes is reasonable<sup>24</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 (§4.3.2), 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.

### 6 Future Work: towards a more general model

The uniform model we use so far is suitable to the problem of adding long links to a wireless network but we also consider other possible applications, where the node-to-node communication demands can be non-uniform and the set of nodes can be placed arbitrarily on the plane.

Consider the model where the node-to-node demands are non-uniform (which may make some nodes very busy and some others slow) but fixed and known initially. We want to satisfy all these demands, or if not possible, satisfy a maximum fraction (called *throughput*, defined later) of all of them equally<sup>25</sup>. Since using a unique capacity for all links is not reasonable for such an un-balanced

---

<sup>24</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 §1.2.

<sup>25</sup>A throughput 80% means 80% of each of the demands are satisfied, but not more than 80% for all

Table 2: 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

network <sup>26</sup>, we consider using additional long links with varying ‘thickness’ (i.e. capacity): the links which supply high demand nodes should be thicker. Thus, the weight of each link is seen as the product of its length and thickness. We want to assign long links with proper capacity (for a given budget which limits the total weight) so that we can maximize the throughput <sup>27</sup>.

We can approach this general model using multicommodity network flows. We consider 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 possible capacity constraints on the edges. A flow system over graph  $G$  has a 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., [28, 39] for more background).

There are two related problems (in classical network flows research) in a more general setting where each edge has an arbitrary length. The first is the maximum concurrent flow problem [39], where the throughput is maximized, subject to the edge capacity constraints. The second, the minimum-cost multicommodity flow problem (see [23] for more background), is to construct a flow  $F$  to meet all the flow demands (and possible edge capacity constraints) while minimizing the cost (weight) function

$$w = \sum_{e \in E} F(e) \times d_e. \quad (6)$$

We consider the general trade-off between throughput  $f$  and cost  $w$ , where given a budget  $C$ , we want to construct a flow system such that  $w \leq C$  while maximizing throughput  $f$ . We propose a design approach where we construct networks in a (Euclidean) metric setting. We also consider the flexibility to adapt our constructions to additional issues, such as a short path requirement or dynamic fault-tolerance issues. More specifically, we continue using our ideas and techniques for choosing additional long links which lead to short routing paths. Thus, we aim at a balance between cost (weight), routing diameter, and throughput (in place of congestion)<sup>28</sup>. Future work here would provide a more complete picture on the capacity of a general hybrid ad-hoc network (beyond Gupta and Kumar’s work in wireless networks [19]): scaling laws for the network throughput as a function of the weight of the added long links and the number of the nodes.

We suggest that our approach of using a partitioning hierarchy can still be used here, where links are placed into layers (levels of the partitioning hierarchy). As our early study on the basic model suggests, an essential part of the budget needs to go to the top layers (so the links there are the longest and thickest) but we need to make sure that each lower layer has enough to serve the top links (i.e. to sublease the bandwidth of top links). We can split the budget layer-wise in some specific manner and then deal with each layer as a separate flow system (once the next upper layer is determined). As before, we can tune the parameters of the hierarchy to optimize the system.

It is also possible to extend our approach beyond the Euclidean setting with uniform node distribution. Recent research in bounded metrics (e.g. see [18]) 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 [8].

<sup>26</sup>We can add many more (unique capacity) links to a demanding pair of nodes but such a policy conflicts our requirement of bounded degree. Also, two parallel unit capacity links of length  $l$  can be seen equivalent to a two-unit capacity link of length  $l$ .

<sup>27</sup>Using congestion as the number of routes through a link (and our early approach minimizing the maximum congestion) is not suited to this scenario. Instead, congestion should be defined as the rate of traffic per unit capacity, which is the same (1.0) for all links upon optimal throughput (but  $< 100\%$ ).

<sup>28</sup>Note that indeed our approach will result in usually unsplitable flows (each  $s - t$  pair will be given a unique routing path to flow through), which is desired in applications for high-speed services, such as video streaming.

<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  $O(r^\beta)$ .

## 7 Related work

We aim at designing bounded-degree networks for a good combination of diameter, weight and congestion. In its general form, the problem is complicated and previous research seems only to approach it partially. While a vast body of work studies the triple degree-diameter-weight trade-offs, for the classical Minimum Steiner Tree<sup>30</sup>, little attention was given to the congestion issue with respect to these other factors. In fact, using tree structures naturally leads to some highly congestive links which connect two subtrees of roughly the same size. However, we show in this paper that near-optimal trade-offs (between weight, diameter and congestion) can be obtained for the Euclidean scenario, where the weight of a link is the Euclidean distances between the two incident nodes.

We now review related areas and classical problems related to our problem. We start with work which focuses on routing while minimizing congestion. We then consider other areas of network architectures and graphs, ranging from traditional topics like Steiner trees, spanners or hypercubes and other parallel architectures to more recent ones as peer-to-peer networks or hierarchical routing in doubling metrics.

**Edge-congestion and load-balancing.** *Edge-congestion* has been used as a network measure to evaluate the performance of architectures, especially parallel ones. For a network and a given routing algorithm, the edge-congestion is the maximum (over all edges) number of paths going through an edge. Fiduccia and Hedrick [12] study the problem of choosing a static shortest-path system that minimizes the edge-congestion in a network, assuming a uniform all-to-all communication. The minimum edge-congestion of the network (over all routing algorithms) is also known as ‘*the edge forwarding index*’, which is introduced in [21] and is studied extensively in a lot of formal work<sup>31</sup>. Chang et al. [9] study the topological properties of crossed cubes (some variants of the hypercubes) and show that the edge-congestion of crossed cubes is the same for the hypercubes (and hence their bisection width is comparable). Low-congested interval routing schemes for many popular interconnection networks such as trees, rings, grids and the butterflies are considered in [11]. Recently, Gkantsidis et al. [15] use congestion (and conductance) to argue that the Internet-like topologies, which grow in a dynamic, decentralized fashion, can support routing with performance characteristics comparable to those of their regular counterparts. This may explain why despite the Internet’s explosive growth, throughput is still good.

Gao and Zhang [13] study tradeoffs between stretch factor and load balancing in routing on growth restricted graphs. Their paper mentions a natural routing conflict: if you do shortest (or near shortest) paths you may overload certain central nodes (nodes with many shortest paths crossing them), or, to evenly distribute load between nodes, you may have to use significantly longer paths. Their paper shows upper bounds for the load balancing ratio when stretch factor is a constant. Typically, they look at the disk graph model where the nodes are a set of wireless sensor nodes, any two of which are linked if they are within a disk of a unit radius. There is some common flavor between this work and ours, but they do not consider using additional long links and do not consider graph weight.

Our notion of *congestion ratio* has some relation to a few recent papers. Xu et. al [42] 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 [13] use a similar concept, “load-balancing ratio” to evaluate different routing algorithms for a given network graph. The load-balancing ratio of a routing algorithm is defined as the ratio between the maximum load over a node by this routing algorithm (for a given set of routing requests) and the optimal load in general, which is the minimum value of all maximum loads produced by any routing algorithm (in their settings, usually, shortest path routing will not produce low load-balancing ratio). Our notion of *congestion ratio* is on edges, as in [42], but actually from a different context as in [13], where the measure is both (network)

---

<sup>30</sup>Extensions include bounded diameter and/or bounded-degree

<sup>31</sup>Before that ‘*vertex forwarding index*’ was introduced in [10].

topology and (routing) algorithm specific.

Some papers in statistical physics address topics in network navigating/routing with a focus on congestion awareness ([5, 16]). Guimera et al. [16] present the following question, given a search algorithm that uses local information only and two fixed size sets of nodes and links, which topology that optimizes the search process? For a large number of parallel searches, the optimal topology they suggested is similar to our results.

**Spanners.** Arya et al. [6] investigated the problem of constructing spanners optimizing various measures simultaneously. As a result, they show how to obtain bounded-degree Euclidean spanners with  $O(n)$  edges,  $O(\log n)$  diameter and  $O(w(T) \log n)$  weight, where  $w(T)$  denotes the weight of the minimum spanning tree (on the complete graph over the given  $n$  nodes). Argawal et al. [3] show that this trade-off is almost tight by proving an  $\Omega(n \log n / \log \log n)$  upper bound on the weight of any  $O(\log n)$  diameter graph over the 1-D array of integer nodes. Note however, that Arya et al.'s spanner is a constant number of trees, thus creating congestion problems at the roots of those trees.

**Bounded diameter minimum Steiner tree.** The problem ( $d$ -MST) is, given a weighted graph  $G = (V, E)$  and a parameter  $d > 0$ , we want to find a minimum Steiner tree spanning  $V$  such that the diameter is bounded by  $d$ . The problem is clearly NP-hard (problem [ND4] in [14]), and not even approximable within  $(1 - \epsilon) \ln |V|$  for any  $\epsilon > 0$ , unless  $NP \subset DTIME(n^{\log \log n})$ , but is  $d \ln |V|$ -approximated for any constant  $d \geq 6$  by Kortsarz and Peleg [26]. For general  $d$ , Kortsarz and Peleg also show how to obtain an  $O(|V|^\epsilon)$ -approximate solution for any  $\epsilon > 0$ . Althaus et al. [4] show  $O(\log n)$ -approximate solution for the related problem, so called  $k$ -hop minimum spanning tree, where the height of the solution spanning tree is bounded by a given parameter  $k > 0$ . Marathe et al. [32] study a general bicriteria design problem: pick any two of the three (degree, diameter, weight) and consider minimizing one of these two while conditioning the other.

**The hypercubes and related parallel architectures.** Leighton's classic text [27] provides a good literature survey. The hypercubes are shown to be a key architecture with a lot of important properties. The popular hypercubic networks (the butterfly, Benes network, Shuffle-Exchange graph, de Bruijn Graphs and the cube-connected-cycle) are considered computationally equivalent. Although having  $\log n$  degrees, hypercubes are shown not much better, computationally, than the hypercubic networks, which only have  $O(1)$  degrees.

Parallel architectures typically ignore the total edge weight or consider all the links to have equal cost. In fact, a family of our network constructions, when considered with uniform weight links, is comparable to the butterfly networks.

**P2P architectures.** The recently hot area of peer-to-peer (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. [42] study a diameter-degree trade-off, which also considers congestion. Asymptotically optimal schemes with bounded degree and  $O(\log n)$  diameter are provided, for example, by Malkhi et al. [30] and Loguinov et al. [29]. Loguinov et al. also provide a graph-theoretic framework to analyze and compare P2P networks on several properties, such as bisection width, average routing path, path overlap, etc. which affect routing and resilience of those networks; as a result, de Bruijn networks have been shown as an ideal architecture (already optimal for diameter-degree). Gummadi et al. [17] instead study the basic geometries underlying the P2P schemes, including the hypercubes, rings, tree-like structures, and the butterfly networks. They focus on the flexibility these geometries provide in the selection of neighbors and routes and show that the ring geometry allows the greatest flexibility, and hence achieves the best resilience and proximity performance.

Although the mentioned P2P networks can be optimal in degree-diameter and are sometime congestion-aware, 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. This can be easily seen in, say, Abraham et al. [1, 2] and Malkhi et al. [30], where they consider using

selective random (long-range) links to achieve optimal routing. Abraham and Malkhi [1] propose a network design coupled with a compact routing scheme on Euclidean metrics. Informally, given  $n$  nodes in a 2-dimensional area of diameter  $D$ , they show how to link them in a way such that the out degree is  $O(1)$ , the expected graph diameter is  $O(\log D)$  and there is a routing scheme using only  $O(\log D)$  memory bits per node to achieve  $1 + \epsilon$  stretch ratio. Abraham et al. [2] also use a similar approach to construct a stretch  $(1 + \epsilon)$  locality-aware P2P architecture. The main idea in choosing long links is actually similar to the Viceroy network (Malkhi et. al [30]), which is based on Kleinberg’s distribution of random link [24]. It is easy to see that, the total length (weight) of these graphs is  $O(nD)$ , or the average weight per node is  $D$  (compared to only  $O(D/\sqrt{n})$  in our designs, which are not stretch-optimized).

**Hierarchical routing.** We propose to use a hierarchical routing mechanism as a part of our network constructions. Most work in this area, such as Kleinrock and Kamoun’s seminal paper “Hierarchical routing for large networks” [25], focuses on different issues, such as how to optimize a hierarchical clustering structure of network nodes to minimize the routing table length (in a packet switching scenario). Recently, Chan et al. [8] study the problem of routing in doubling metrics and show how to perform hierarchical routing with small stretch and compact routing tables. The partitioning hierarchy they use is similar to ours, however, they do not consider congestion.

**Small-world Models.** As mentioned, this work is inspired by our study of small-world models [35, 33], which follows the seminal work by Kleinberg [24]. Small-world networks are being used and studied in many disciplines, including the social and natural sciences. These networks possess a striking property, the so called small-world phenomenon, also often spoken of as “six degrees of separation” (between any two people in the United States). As a de facto standard, small-world networks are recently seen as the networks which feature two properties, small diameter and significant clustering. The influential model by Watts and S. Strogatz [41] and follow-up work first considered a small-world model where long random links are added into a local-contact graph (a  $k$ -degree ring lattice). Recently, Kleinberg [24], building on the work of Watts and Strogatz [41], proposed a family of small-world networks to study another compelling aspect of Milgram’s original findings: a greedy algorithm using only local information can construct short paths.

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. These graph models have generated considerable interest and follow-up work. Kleinberg leaves important issues open in the analysis of routing in his model and we complete the analysis and then extend our techniques to a much broader range of settings in [33, 35]. 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 network, an optimal degree-diameter P2P architecture [30] and Helmy’ small-worlds in wireless sensor networks. Given that Kleinberg’s setting is a very specific one, in [35] we characterize important features underlying the distribution of random links and the grid structure, which produce those nice small-world properties.

## Acknowledgment

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 dhsts,” in *Proc. of ACM Symp. on Discrete Algorithms (SODA)*, 2004.
- [3] P. Agarwal, Y. Wang, and P. Yin, “Lower bound for sparse euclidean spanners,” in *Proc. of ACM Symp. on Discrete Algorithms (SODA)*, 2005.
- [4] E. Althaus, S. Funke, S. Har-Peled, J. Konemann, E. Ramos, and M. Skutella. (2004) Approximating k-hop minimum-spanning trees. [Online]. Available: <http://www.mathematik.uni-dortmund.de/~skutella/dmst.pdf>
- [5] A. Arenas, A. Cabrales, A. Diaz-Guilera, R. Guimera, and F. Vega-Redondo. (2004) Search and congestion in complex networks. [Online]. Available: <http://xxx.lanl.gov/abs/cond-mat/0301124>
- [6] 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.
- [7] R. Banner and A. Orda, “Multipath routing algorithms for congestion minimization,” in *NETWORKING*, Waterloo, Canada, 2005, pp. 536–548.
- [8] 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.
- [9] 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.
- [10] F. Chung, E. Coffman, M. Reiman, and B. Simon, “The forwarding index of communication networks,” *IEEE Trans. Inform. Theory*, vol. 33, no. 2, pp. 224–232, Mar. 1987.
- [11] S. Cicerone, G. D. Stefano, and M. Flammini, “Static and dynamic low-congested interval routing schemes,” *Theor. Comput. Sci.*, vol. 276, no. 1-2, pp. 315–354, 2002.
- [12] 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.
- [13] 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.
- [14] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY: W. H. Freeman & Co., 1979.
- [15] C. Gkantsidis, M. Mihail, and A. Saberi, “Conductance and congestion in power law graphs,” in *Proc. of ACM SIGMETRICS*, 2003.
- [16] R. Guimera, A. Diaz-Guilera, F. Vega-Redondo, A. Cabrales, and A. Arenas, “Optimal network topologies for local search with congestion,” *Phys. Rev. Lett.*
- [17] K. P. Gummadi, R. Gummadi, S. D. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica, “The impact of dht routing geometry on resilience and proximity,” in *ACM SIGCOMM*, 2003.
- [18] 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.
- [19] P. Gupta and P. Kumar, “The capacity of wireless networks,” *IEEE Trans. Inform. Theory*, vol. 46, no. 2, pp. 388–404, 2000.

- [20] A. Helmy, “Small worlds in wireless networks,” *IEEE Commun. Lett.*, vol. 7, no. 10, pp. 490–492, Oct. 2003.
- [21] M. Heydemann, J. Meyer, and D. Stotteau, “On forwarding indices of networks,” vol. 23, pp. 103–123, 1989.
- [22] X. Jia, D. Li, and D. Du, “Qos topology control in ad hoc wireless networks,” in *IEEE INFOCOM*, 2004.
- [23] A. Kamath, O. Palmon, and S. Plotkin, “Fast approximation algorithm for minimum cost multi-commodity flow,” in *Proc. of ACM Symp. on Discrete Algorithms (SODA)*, 1995.
- [24] J. Kleinberg, “The small-world phenomenon: An algorithmic perspective,” in *Proc. of ACM Symp. on Theory of Computing (STOC)*, 2000.
- [25] 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.
- [26] G. Kortsarz and D. Peleg, “Approximating shallow-light trees,” in *Proc. of ACM Symp. on Discrete Algorithms (SODA)*, 1997, pp. 103–110.
- [27] F. Leighton, *Introduction to parallel algorithms and architectures: arrays - trees - hypercubes*. Morgan Kaufmann Pub., 1992.
- [28] 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.
- [29] D. Loguinov, V. R. A. Kumar, and S. Ganesh, “Graph-theoretic analysis of structured peer-to-peer systems: Routing distances and fault resilience.”
- [30] 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.
- [31] G. Manku, M. Bawa, and P. Raghavan, “Symphony: Distributed hashing in a small world,” in *USENIX Symp. on Internet Tech. and Sys.*, 2003.
- [32] M. Marathe, R. Ravi, R. Sundaram, S. Ravi, D. Rosenkrantz, and H. H. III, “Bicriteria network design problems,” in *ICALP’95*.
- [33] C. Martel and V. Nguyen, “Analyzing kleinberg’s (and other) smallworld models,” in *Proc. of ACM Symp. on Princ. of Dist. Comp. (PODC)*, 2004.
- [34] 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.
- [35] V. Nguyen and C. Martel, “Analyzing and characterizing small-world graphs,” in *Proc. of ACM Symp. on Discrete Algorithms (SODA)*, 2005.
- [36] ——. (2005) Designing networks for low weight, small routing diameter and low congestion. [Online]. Available: <http://www.wcsif.cs.ucdavis.edu/nguyenvk/>
- [37] 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.

- [38] Z. Sen, “An optimal broadcasting schema for multidimensional mesh structures,” in *Proc. of ACM Symp. on Applied Computing*, 2003.
- [39] F. Shahrokhi and D. W. Matula, “The maximum concurrent flow problem,” *Journal of the ACM (JACM)*, vol. 37, no. 2, pp. 318–334, 1990.
- [40] 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.
- [41] D. Watts and S. Strogatz, “Collective dynamics of small-world networks.”
- [42] 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.

## Appendix

*More details on  $\mu$ -partition (§2.3).* Given a constant  $.5 \leq \mu < 1$ , a  $\mu$ -partition of a block of size  $m$  (i.e. a  $m \times m$  square of nodes) is the following set of sub-blocks. Let  $l = \lfloor m^\mu \rfloor$ ; let  $m = ql + r$  where  $q, r$  are integers but  $0 \leq r < l$ . If  $r = 0$  then we have exactly  $q^2$  identical sub-blocks of size  $l \times l$ , i.e. a  $q \times q$  grid if each  $l \times l$  block collapses to a single node. The block of indexes  $i$  and  $j$  is  $[(i-1)l .. il-1, (j-1)l .. jl-1]$ , for any  $1 \leq i, j \leq q$ . For  $r > 0$  we can still keep this  $q \times q$  virtual grid by extending some of the  $q^2$  sub-blocks with one more row, column or both. This can be done by the following simple procedure. Define the sequence  $\{a_i\}_{i=1}^q$  as  $a_i = (i-1)(l+1)$  for  $i \leq r$  and  $a_i = (i-1)l + r$  for  $r \leq i \leq q$ . For any  $1 \leq i, j \leq q$ , the  $(i, j)$ -block is  $(a_i .. a_{i+1} - 1, a_j .. a_{j+1} - 1)$ .  $\square$

*Proof of lemma 6.* Given an arbitrary node  $u \in S$ , let  $p$  denote  $Pr[u \text{ ‘misses’ } T]$ , i.e. the random link from  $u$  does not go to any node in  $T$ , and similarly, let  $P = Pr[S \text{ ‘misses’ } T]$ . The random link from  $u$  goes to a fixed node  $v \in T$  with probability at least  $\Omega(L^{-\alpha})$ , so, to  $T$  with probability  $\Omega(|T| \times L^{-\alpha}) = \Omega(L^{2\mu} \times L^{-\alpha}) = \Omega(L^{(2\mu-\alpha)})$ . So  $p \leq 1 - \Omega(L^{(2\mu-\alpha)}) \leq e^{cL^{(2\mu-\alpha)}}$  <sup>(32)</sup>, for some constant  $c > 0$ . Now combining all the events  $u \text{ ‘misses’ } T$  for each  $u \in S$ , we have  $P = p^{|S|} \leq e^{cL^{(4\mu-\alpha)}}$ .  $\square$

---

<sup>32</sup>Using the basic calculus fact  $1 + x \leq e^x$