Demand-Aware Small-World Networks on Clustered Demands

Chen Avin
Ben-Gurion University of the Negev, Israel

Robert Elsässer
University of Salzburg, Austria

Aleksander Figiel
TU Berlin, Germany

Darya Melnyk
TU Berlin, Germany

Stefan Schmid
TU Berlin, Germany

Abstract

Small-world networks are attractive for the efficient routing they provide, requiring only a low link density. They have hence also been considered for the design of distributed systems, such as peer-to-peer networks. However, existing small-world network designs are oblivious to the actual traffic they serve. In this paper, we initiate the study of demand-aware small-world networks. In particular, we extend the Kleinberg graph model, by allowing the nodes to choose the distribution of long-range links according to the traffic demand. We present a formal analysis of the weighted route lengths for the important case of clustered demands. We show both in theory and in simulations, using real-world traffic workloads, that demand-aware small-world graphs can significantly outperform their demand-oblivious counterparts.

2012 ACM Subject Classification Networks → Peer-to-peer networks; Networks → Data center networks; Theory of computation → Social networks

Keywords and phrases Small-world networks, demand-aware network designs, algorithms and analysis, clustering

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2025.28

Funding Supported by the European Research Council (ERC), grant agreement No. 864228 (AdjustNet), Horizon 2020, 2020-2025

1 Introduction

Small-world networks have fascinated researchers for many decades. They are not only used to describe natural and social networks, but also to build distributed systems such as peer-to-peer systems [20, 26]. In his influential experiment, Milgram [30] uncovered that distances among two people are often surprisingly short, also known as the six-degrees-of-separation phenomenon. The latest since Kleinberg's algorithmic explanation [24], showing that simple greedy strategies can lead to (poly-)logarithmic routes in augmented grids, such navigable networks have also inspired computer scientists to develop communication networks aiming to imitate the desirable properties of small-world networks, e.g., [10, 28].

A popular approach to model and design small-world networks, first introduced by Watts and Strogatz [36], is to combine two networks: as the basis, we take a network that has a large cluster coefficient or a d-dimensional grid, and then we augment such a graph with random links, according to a certain distribution, typically a power law distribution. This

augmentation represents acquaintances that connect nodes to parts of the network that would otherwise be far away.

In this paper, we revisit such small-world networks from a novel perspective and initiate the study of *demand-aware* small-world network designs. Our perspective is motivated by the observation that existing small-world network designs are optimized in a demand-oblivious manner, for the worst-case diameter or route lengths, in case of uniform all-to-all communication demands. However, in practice, communication demands typically come with much structure and are highly skewed [5], which may be exploited to provide even shorter routes. Demand-aware networks have recently also received much attention in the context of datacenter network designs, enabled by novel reconfigurable optical switches [22].

We first consider a one-dimensional grid and assume that the demands are clustered and sparse, where a majority of the nodes does not communicate at all. Such demand graphs can be found in real-world datasets such as Facebook datacenters [15] and in high performance computing (HPC) cluster traffic traces [5] that we analyze in this work. We show a strategy to augment the one-dimensional grid network such that the expected distances between any pair of nodes in the demand-aware setting are improved over the demand-oblivious strategies. We also extend our theoretical analysis to the standard case of two-dimensional grids. In our experiments on synthetic and real-world data, we show that the demand-aware strategy considerably outperforms the existing demand-oblivious strategies.

1.1 Contribution

In this work, we initiate the study of demand-aware small-world network designs. We present a theoretical analysis of two main settings where the nodes are arranged along a cycle (one-dimensional grid) and along a two-dimensional grid. The demands are assumed to be sparse and clustered. The cluster sizes are analyzed for the case of Uniform, Poisson, and power law distributions. We show that adding edges to a network locally in a randomized but demand-aware manner outperforms the original demand-oblivious augmentation methods based on Watts and Strogatz [36] and Kleinberg [24].

Our theoretical analysis is motivated by the empirical fact that the demands are clustered in datacenter networks and that many nodes do not participate in communication. Indeed, we perform an empirical analysis of the HPC cluster traffic traces [5] and show that the actual traces are well represented by the theoretical demand-aware small-world model in the one-dimensional grid case. We then run simulations of our algorithm on synthetically generated data as well as on the HPC cluster traffic traces. Also our empirical evaluations show that demand-aware strategies to augment the network outperform their demand-oblivious counterparts.

1.2 Further Related Work

Our paper combines two active areas of research: small-world networks and demand-aware network designs.

The first algorithmic approach to capture the small-world phenomenon was introduced by Watts and Strogatz [36], who presented networks suited for decentralized search. This approach was later generalized by Kleinberg [24] who initiated the study on decentralized search algorithms of networks augmented with random edges. Navigability in networks has since then received a lot of attention in the literature [3, 7, 17, 18, 27]. Besides the original model, also other small-world models [12, 29] have been proposed, hyperbolic metric

spaces have been studied to model complex networks [32, 25, 8], and different routing methods [16, 34] have been investigated in the past.

Demand-aware networks are motivated by their applications in datacenters, where reconfigurable optical communication technologies have recently introduced unprecedented flexibility in network design. An optical circuit switch controls which edges in the network are active, as defined by a schedule. Commonly, at any given time the active edges form a matching and connectivity in the network is established over time. Prior work focuses on designing schedules for the optical circuit switch in a demand-aware manner to ensure efficient transmission of data [2, 4, 6, 11, 19, 21, 23, 35, 37, 38, 39]. Contrary to these works, where edges are added by some deterministic process, our work assumes that augmented edges are added only with some probability. For a review of the enabling technologies of dynamic datacenter networks, we refer to the recent survey by Hall et al. [22].

1.3 Overview

We start by presenting the formal model of demand-aware small-world networks in Section 2. We thereby differentiate between the case of directed cycles and grids. In Section 3 we present an algorithm that augments the original communicating network in a demand-aware manner and show that this strategy outperforms the demand-oblivious strategy presented by Kleinberg [24]. In Section 4, we extend this result to the case of grids. In Section 5, we present a practical evaluation of our demand-aware algorithms for synthetically generated and real-world demand matrices. We show that the analyzed sparse communication structures indeed appear in real-world datacenter networks. Finally, we conclude in Section 6.

2 Model

In this section, we present the demand-aware variant of the small-world network model originally introduced by Kleinberg [24]. Given a set of n nodes V, we assume that the nodes are communicating with respect to a predefined $n \times n$-*demand matrix* D. This demand matrix represents the communication in a datacenter network and is therefore assumed to be sparse. We differentiate between two main communication patterns - communication on a cycle and communication in a grid.

Communication on a cycle

The demand graph for communication on a cycle is defined as follows: Communicating nodes are placed at the nodes of a large cycle in a one-to-one fashion. Subsets of up to x communicating nodes are placed at neighboring nodes along a cycle each forming a connected component, called *cluster*. We assume that there are y clusters that are disjoint from each other, that is, any pair of clusters along the cycle has at least one (possibly many) non-communicating node between them. We further assume that for any pair of neighboring clusters (i.e., clusters between which only non-cluster nodes lie) a direct link connects the two closest nodes. This ensures connectivity in the demand graph. Observe that $x \cdot y \ll n$ as the demand graph is sparse.

We now define the weights in the demand matrix D. In this matrix, any node u located inside a cluster communicates with probability p to nodes within its cluster. With probability q, u communicates to the nodes in other clusters. Here we have $q = 1 - p$. Observe that the demands to all other nodes outside of clusters as well as the demands between the nodes outside of the clusters are set to 0.

Communication in a grid

In addition to the one-dimensional case mentioned above, we consider the demand graph for communication in a grid. We view each node of the grid as a supernode that contains a cluster of communicating nodes. The y clusters are arranged in a $\sqrt{y} \times \sqrt{y}$-grid-like structure. Each cluster contains up to x nodes. The communicating nodes inside a cluster are arranged along a cycle, i.e., in a worst-case manner. To make sure that the clusters are connected as in a grid, we connect the designated "first" node in a cluster to the designated "last" node in the cluster on the top and to its left, while the last node in a cluster is connected to the first node of the cluster below and to its right.

The demand matrix D for the communications between and inside the clusters is defined equivalently to the case of cycles.

Decentralized algorithms

The idea of small-world networks is to augment the original graph with edges such that the expected distance between any two nodes is minimized. Each communicating node in the network is allowed to add one directed (long-range) edge to another node in the network. This augmentation is performed by each node locally in a randomized fashion.

The communication between the nodes works according to a decentralized search algorithm. In particular, we assume that the nodes use *greedy routing* to forward the messages: Assume that a node u wants to send a message m to node v. We assume that node u knows the location of the destination v, the location of its adjacent nodes, as well as the distances from its adjacent nodes to v. When using greedy routing, u will send its message to the adjacent node closest to the destination v. This process is repeated by every node that receives m until m reaches its destination.

In this paper, the goal is to find a random distribution according to which the nodes add a directed edge such that the weighted expected distance between all communicating nodes is minimized. For simplicity, we assume that the entries of D are normalized in the analysis. Let $\tilde{D}$ denote the demand graph augmented with random edges and let $G_{\tilde{D}}(u, v)$ denote the greedy routing distance between the nodes u and v on the augmented graph $\tilde{D}$. We are interested in minimizing the expected routing distance (ERD) between any two nodes in the augmented network:

$$E[G(u, v)|\tilde{D}] = \sum_{u,v \in V} G_{\tilde{D}}(u, v) \cdot D(u, v).$$

3 Demand-aware small-world phenomenon on a cycle

In this section, we present how the demand matrix can be used in the randomized process of selecting augmenting edges. Observe that in the demand-oblivious version of small-world networks, the edges are added proportional to the inverse of the distance between two nodes. In a sparse demand matrix, however, many nodes may not be communicating at all and thus adding an edge in a demand-oblivious manner may only reduce the communication distance between non-communicating nodes and not the ERD, meaning that we "waste" augmenting edges.

We start this section by presenting the distribution according to which augmenting edges are chosen locally and demand-aware. We then analyze different sparse demand matrices for communication in the cycle (see Section 2) and show that the presented demand-aware process of finding augmenting edges outperforms the demand-oblivious strategy.

Distribution of augmented edges

Let D be the demand matrix as presented in Section 2. Here, we discuss how $\tilde{D}$ is constructed from D by adding long-range edges to the demand matrix. Let $d_x(\cdot,\cdot)$ denote the distance along the line (the number of hops) between any two vertices that belong to the same cluster. To simplify the analysis, we assume that the nodes inside a cluster are also connected along a cycle. Let $d_I(I_u, I_v)$ denote the "cluster hop" distance between two clusters I_u and I_v computed by assuming that each cluster can be represented as a supernode, and that there is an edge between the two closest nodes of any two neighboring clusters. The "cluster hop" distance between node $u \in I_u$ and $v \in I_v$, is defined as the cluster hop distance between the corresponding clusters.

Then, each node adds an extra edge with the following probabilities: A node u in cluster I_u adds exactly one long-range edge. It adds this edge to a distinguished node identified as the first node in cluster I_v with probability proportional to $q \cdot d_I^{-1}(I_u, I_v)$. If two nodes u and v are in the same cluster, an edge is added with probability proportional to $p \cdot d_x^{-1}(u, v)$.

To determine the actual probabilities used to add edges, we calculate the normalization factor:

$$\sum_{\substack{v,w \in I \\ v \neq w}} 2p \cdot d_x^{-1}(v,w) + \sum_{\substack{I_i, I_j \\ I_i \neq I_j}} 2q \cdot d_I^{-1}(I_i, I_j) = 2p \cdot \sum_{i=1}^{x/2} \left(\frac{1}{i}\right) + 2q \cdot \sum_{i=1}^{y/2} \left(\frac{1}{i}\right)$$

$$\leq 2 + 2p + 2q + 2p \cdot \log(x/2) + 2q \cdot \log(y/2).$$

Observe that u can only connect to one node, i.e., the normalization factor only considers nodes within the same clusters, while every other cluster is viewed as a supernode. Therefore, for two nodes $u \in I_u$ and $v \in I_v$, the probability that u connects to v (that is, to cluster I_v) is

$$\frac{1}{2p \cdot \log(x) + 2q \cdot \log(y) + c} \cdot d^{-1}(u,v).$$

Here $d(\cdot,\cdot)$ represents the distance inside the same cluster or the cluster hop distance, depending on where the nodes u and v are located.

3.1 Analysis on clusters of the same size

We start by analyzing the demand-aware small-world network on a restricted case, where each cluster consists of exactly x nodes. This case is then generalized to cluster sizes drawn from the Poisson and power law distributions in Sections 3.2 and 3.3. This analysis considers the worst-case demand matrix D, where the two furthest nodes inside a cluster (two furthest clusters respectively) communicate with probability p (probability q respectively). The derived expected greedy routing distance in this section is thus an upper bound on $E[G(u,v)|\tilde{D}]$.

In the next steps, we will show that the expected number of steps to reach the destination is $O\big(\log(xy)(p\log(x) + q\log(y))\big)$. We will perform the analysis in two steps: in the first part, we show that the expected cluster distance decreases exponentially until the destination cluster is reached. In the second part, we analyze the expected number of hops needed inside the destination cluster to reach the destination node. Note that the long-range edges are always added to the first node of a cluster thus making it possible to split the analysis. This analysis follows the analysis outline in [13, Chapter 20] for one-dimensional grids.

▶ **Lemma 1** (Routing between the clusters). *Routing between the clusters takes* $O(\log(y) \cdot (p \cdot \log(x) + q \cdot \log(y)))$ *steps in expectation.*

Proof. We start by considering clusters as supernodes and say that the routing starts in cluster I_u and terminates in cluster I_v. Let J denote the set of clusters in the $|J|$-neighborhood of I_v. Assume further that source cluster I_u is exactly $|J|$ cluster hops away from cluster I_v. We first show that it takes $O(p \cdot \log(x) + q \cdot \log(y))$ rounds in expectation for the greedy routing from I_u to I_v to end up in $J/2$, i.e. inside the $|J|/2$-neighborhood of I_v.

There are at least $|J|$ clusters that any node in I_u can connect to in $J/2$. The probability for a node u to have a direct link to a cluster in $J/2$ is at least

$$|J| \cdot \frac{1}{2p \cdot \log(x) + 2q \cdot \log(y) + c} \cdot d_I^{-1}(I_u, I_w)$$

where I_w is the farthest cluster at distance $3|J|/2$. That is,

$$\frac{|J|}{2p \cdot \log(x) + 2q \cdot \log(y) + c} \cdot d_I^{-1}(I_u, I_w) > \frac{|J|}{(p \cdot \log(x) + q \cdot \log(y) + c') \cdot 3|J|} = \frac{1}{3(p \cdot \log(x) + q \cdot \log(y) + c')}.$$

Here, we lower bounded the probability assuming that the largest cluster hop distance is $3|J|$. Recall that this is possible since any node that draws its long-range edge to another cluster considers the cluster as a supernode. This view helps us to deal with the fact that we may iterate over nodes from the same cluster for many steps.

Let X_i be a random variable denoting the number of rounds for the greedy routing to reach a cluster in $J/2$. The probability that a node u does not reach a node in $J/2$ within r rounds is

$$\Pr[X_i > r] \leq \left(1 - \frac{1}{3(p \cdot \log(x) + q \cdot \log(y) + c')}\right)^{r-1}.$$

Here we used the fact that all clusters have the same size and thus every node uses the same probability distribution for its long-range edges. Next, we bound the expected value of X_i, i.e. the expected time (number of steps) to half the distance to the destination cluster:

$$\mathrm{E}[X_i] = \sum_{j=1}^{\infty} Pr[X_i > j].$$

This results in $\mathrm{E}[X_i] = 3(p \cdot \log(x) + q \cdot \log(y) + c')$.

Let X denote the number of rounds to reach the destination cluster. Since we half the cluster distance until we end up in the destination cluster, we have $X = X_1 + X_2 + \ldots + X_{\log y}$. Then,

$$\mathrm{E}[X] \leq \log(y) \cdot 3\big(p \cdot \log(x) + q \cdot \log(y) + c'\big).$$ ◀

Similarly, we can derive the number of rounds that a node needs to route inside the cluster.

▶ **Lemma 2** (Routing within a cluster). *Routing within a cluster takes* $O\big(\log(x) \cdot (p \cdot \log(x) + q \cdot \log(y))\big)$ *steps in expectation.*

We omit the proof of this lemma as it is analogous to the analysis in [13], with the exception that the normalization factor from Lemma 1 is applied. By summing up the expected number of steps from Lemma 1 and 2.

▶ **Theorem 3** (Routing with equal cluster sizes). *Greedy routing on a demand-aware cycle containing y clusters of size x each together with augmented edges takes* $O\big(\log(xy)(p\log(x) + q\log(y))\big)$ *steps in expectation.*

Observe that this bound corresponds to the upper bound of [13] when the matrix is dense, like for example in the stochastic block model [1] where all nodes belong to some cluster. In that case $\log(x) + \log(y) = \log(xy) = \log(n)$. When the demand matrix is sparse, i.e., $xy \ll n$, the average distances become much smaller than in the demand-oblivious model.

3.2 Cluster sizes following the Poisson distribution

So far, we assumed that all clusters have the same size x. In this section, we relax this condition and assume that each cluster $I_k, k \in [y]$, has a different size $|I_k| > 1$, and that the cluster sizes are distributed according to the Poisson distribution. Under this assumption, each node has to compute its own normalization factor, as the normalization factor depends on the size of the cluster to which the node belongs:

$$\sum_{\substack{v,w \in I_k \\ v \neq w}} 2p \cdot d_x^{-1}(v,w) + \sum_{\substack{I_i, I_j \\ I_i \neq I_j}} 2q \cdot d_I^{-1}(I_i, I_j) = 2p + 2q + 2p \cdot \log(|I_k|) + 2q \cdot \log(y).$$

In the following, we restrict the distribution according to which the cluster sizes of y clusters are chosen and compute the number of cluster hops that are needed to reach the destination cluster.

▶ **Theorem 4** (Routing between clusters under Poisson distribution)**.** *Assume that the cluster sizes X follow the Poisson distribution $Pois(k, \lambda)$, i.e., $\Pr(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$. Routing between the clusters takes*

- $O(\log(y) \cdot (p \cdot \log(\lambda + C_1\sqrt{\lambda \log n}) + q \cdot \log(y)))$ *steps in expectation if $\lambda > c \log n$,*
- $O(\log(y) \cdot (p \cdot \log(C_2 \log n) + q \cdot \log(y)))$ *steps in expectation if $\lambda < c \cdot \log n$,*
- $O(\log(y) \cdot (p \cdot \log(C_3 \frac{\log n}{\log\log n}) + q \cdot \log(y)))$ *steps in expectation if $\lambda = const$,*

where C_1, C_2, C_3 and c are large constants.

Before proving the theorem, we first show a concentration of the cluster sizes for each choice of λ and then proceed with the analysis as in Lemma 1. Similar tail bounds for Poisson distribution have been analyzed in the literature. In the following, we adapt the tail bounds from [33] to our approach.

▶ **Lemma 5.** *Let $\lambda > c \log n$. Then, the cluster sizes are concentrated in the interval $\left[\lambda - C\sqrt{\lambda \log n}, \lambda + C\sqrt{\lambda \log n}\right]$, where $C \geq 2$ is a constant, with probability at least $1 - 1/n^3$.*

Proof. In the following, we use Stirling's approximation $k! \approx \sqrt{2\pi k}\left(\frac{k}{e}\right)^k$ to approximate the factorial. We set $\lambda = c' \log n$, where $c' > c$.

$$\begin{aligned}
\Pr[X = \lambda + C\sqrt{\lambda \log n}] &\approx \frac{\lambda^{\lambda + C\sqrt{\lambda \log n}}}{\sqrt{2\pi(\lambda + C\sqrt{\lambda \log n})}\left(\frac{\lambda + C\sqrt{\lambda \log n}}{e}\right)^{\lambda + C\sqrt{\lambda \log n}}} e^{-\lambda} \\
&< e^{\lambda + C\sqrt{\lambda \log n}} \frac{\lambda^{\lambda + C\sqrt{\lambda \log n}}}{\left(\lambda + C\sqrt{\lambda \log n}\right)^{\lambda + C\sqrt{\lambda \log n}}} e^{-\lambda} \\
&= e^{C\sqrt{\lambda \log n}} \frac{1}{\left(1 + \frac{C}{\sqrt{c'}}\right)^{\frac{\sqrt{c'}}{C} \cdot C\sqrt{\lambda \log n}} \left(1 + \frac{C\sqrt{\log n}}{\sqrt{\lambda}}\right)^{C\sqrt{\lambda \log n}}} \\
&\overset{(a)}{=} e^{C\sqrt{\lambda \log n}} \frac{1}{e^{C\sqrt{\lambda \log n}} \left(1 + \frac{C\sqrt{\log n}}{\sqrt{\lambda}}\right)^{C\sqrt{\lambda \log n}}}
\end{aligned}$$

$$= \frac{1}{\left(1+\frac{C\sqrt{\log n}}{\sqrt{\lambda}}\right)^{\frac{\sqrt{\lambda}}{C\sqrt{\log n}}C^2\log n}} \overset{(b)}{\approx} \left(\frac{1}{e}\right)^{C^2\log n} = \frac{1}{n^{C^2}} < \frac{1}{n^4}$$

For Equations (a) and (b), we assume that $c' \gg C^2$.

Note that the number of clusters is upper bounded by n by definition. Using union bound, we can show:

$$\Pr[X > \lambda + C\sqrt{\lambda \log n}] < n \cdot \Pr[X = \lambda + C\sqrt{\lambda \log n}] < \frac{1}{n^3}.$$

Analogously, we can prove the other side:

$$\Pr[X < \lambda - C\sqrt{\lambda \log n}] < \frac{1}{n^3}.$$

This concludes the proof of the lemma. ◀

▶ **Lemma 6.** *Let $\lambda < c \cdot \log n$. Then, the cluster sizes can be upper bounded by $C \cdot \log n$, where $C > 2c \geq 2$, with probability at least $1 - 1/n^3$.*

Proof.

$$\Pr[X = C\log n] = \frac{\lambda^{C\log n}}{(C\log n)^{C\log n}} e^{-\lambda} \leq \left(\frac{c\log n}{C\log n}\right)^{C\log n} = \left(\frac{c}{C}\right)^{C\log n} < \left(\frac{1}{2}\right)^{C\log n} < \frac{1}{n^4}$$

Note that for the last inequality, we assume that $C > 4$. As in the proof of Lemma 5, we can use the union bound together with the fact that there can be at most n clusters to prove the Lemma statement:

$$\Pr[X \geq C\log n] < n \cdot \Pr[X = C\log n] < \frac{1}{n^3}.$$ ◀

▶ **Lemma 7.** *Let λ be a constant. Then, the cluster sizes can be upper bounded by $\frac{\log n}{\log\log n}$ with probability at least $1 - 1/n^3$.*

Proof.

$$\Pr\left[X = C\frac{\log n}{\log\log n}\right] = \frac{\lambda^{C\frac{\log n}{\log\log n}}}{\left(C\frac{\log n}{\log\log n}\right)^{C\frac{\log n}{\log\log n}}} \cdot e^{-\lambda}$$

Next, we reformulate the numerator $\lambda^{C\log n} = 2^{c'\log n}$ for some constant $c' = C\log\lambda$ and receive

$$\lambda^{C\frac{\log n}{\log\log n}} = n^{\frac{c'}{\log\log n}} = n^{o(1)}.$$

For the denominator, we can write

$$\left(C\frac{\log n}{\log\log n}\right)^{C\frac{\log n}{\log\log n}} > n^{c''}$$

for $c'' = C^{\frac{\log n}{\log\log n}\cdot\log_n C^{\frac{\log n}{\log\log n}}}$. Thus,

$$\Pr\left[X = C\frac{\log n}{\log\log n}\right] < \frac{n^{o(1)}}{n^{c''}} < \frac{1}{n^4}$$

for a sufficiently large value of n.

As before, we apply the union bound together with the fact that there can be at most n clusters to show the theorem statement:

$$\Pr\left[X \geq C\frac{\log n}{\log\log n}\right] < n \cdot \Pr\left[X = C\frac{\log n}{\log\log n}\right] < \frac{1}{n^3}.$$ ◀

Using the above bounds, we can now prove Theorem 4:

Proof of Theorem 4. As in the proof of Lemma 1, we consider the clusters as supernodes and let J denote the $|J|$-neighborhood of the destination cluster I_v. We first show that it takes $O(p \cdot \log(\lambda) + q \cdot \log(y))$ steps in expectation for the greedy routing starting in u to reach a node in $J/2$, i.e. inside the $|J|/2$-neighborhood of I_v.

Node u can connect to at least $|J|$ clusters in $J/2$. The probability for u from cluster I_u to have a direct link to a cluster in $J/2$ is at least

$$|J| \cdot \frac{1}{2p \cdot \log(|I_k|) + 2q \cdot \log(y) + c} \cdot d_I^{-1}(I_u, I_w)$$

where I_w is the farthest cluster at distance $3|J|/2$. That is,

$$\frac{|J|}{2p \cdot \log(|I_u|) + 2q \cdot \log(y) + c} \cdot d_I^{-1}(I_u, I_w) > \frac{|J|}{(p \cdot \log(|I_u|) + q \cdot \log(y) + c') \cdot 3|J|} \tag{1}$$
$$= \frac{1}{3(p \cdot \log(|I_u|) + q \cdot \log(y) + c')}.$$

Let Y_i be a random variable denoting the number of steps for a node u to reach a cluster in $J/2$. Note that Y_i depends on the cluster size of the nodes that are visited on the path from u to v. To upper bound the expected number of steps, we do a case distinction depending on the size of λ:

Case $\lambda > c \log n$. We can bound the term in Equation (1) w.r.t. the average cluster size using Jensen's inequality for concave functions:

$$\Pr[Y_i > r] \leq \prod_{k=1}^{r} \left(1 - \frac{1}{3(p \cdot \log(|I_k|) + q \cdot \log(y) + c')}\right)$$
$$\leq \left(1 - \frac{1}{3(p \cdot \log\left(\frac{1}{r}\sum_{k\in[r]} |I_k|\right) + q \cdot \log(y) + c')}\right)^r. \tag{2}$$

Here, the clusters I_k represent the clusters of the nodes visited within r steps. In the following, we will focus on bounding the average $\frac{1}{r}\sum_{k\in[r]} |I_k|$. In the case $\lambda > c \log n$, the cluster sizes are concentrated in $\left[\lambda - C_1\sqrt{\lambda \log n}, \lambda + C_1\sqrt{\lambda \log n}\right]$ (see Lemma 5). Due to $r \leq n$ and the union bound, the probability that this sum contains a cluster outside of the interval is less than $1/n^2$. Thus, with probability st least $1 - 1/n^2$ we have

$$\frac{1}{r}\sum_{k\in[r]} |I_k| < \lambda + C_1\sqrt{\lambda \log n}$$

and also $\Pr[Y_i > r] \leq \left(1 - \frac{1}{3(p \cdot \log(\lambda + C_1\sqrt{\lambda \log n}) + q \cdot \log(y) + c')}\right)^r$.

From here on, we can now apply

$$\mathrm{E}[Y_i] = \sum_{j=1}^{n} \Pr[Y_i > j].$$

This is because, among all n clusters, w.h.p., there will be no cluster outside of the concentration bounds.

Let the random variable $Y = Y_1 + Y_2 + \ldots + Y_{\log y}$ denote the number of rounds to reach the destination cluster. Then,

$$\mathrm{E}[Y] \leq \log(y) \cdot 3\big(p \cdot \log(\lambda + C_1\sqrt{\lambda \log n}) + q \cdot \log(y) + c'\big).$$

Case $\lambda < c \cdot \log n$. We will derive the upper bound on the expected number of rounds analogously to the previous case. The average cluster size of the visited clusters is upper bounded by

$$\frac{1}{r}\sum_{k\in[r]} |I_k| < C_2 \log n$$

with probability $1/n^2$. By plugging in this value into Equation (2), we get

$$\Pr[Y_i > r] \leq \left(1 - \frac{1}{3(p \cdot \log(C_2 \log n) + q \cdot \log(y) + c')}\right)^r.$$

For the expected number of rounds holds

$$\mathrm{E}[Y] \leq \log(y) \cdot 3\big(p \cdot \log(C_2 \log n) + q \cdot \log(y) + c'\big).$$

Case $\lambda = const$. In this case, the average cluster size of the visited clusters is upper bounded by

$$\frac{1}{r}\sum_{k\in[r]} |I_k| < C_3 \frac{\log n}{\log\log n}$$

with probability $1/n^2$. Plugging in this value into Equation (2) results in

$$\Pr[Y_i > r] \leq \left(1 - \frac{1}{3\left(p \log\left(C_3 \frac{\log n}{\log\log n}\right) + q \log(y) + c'\right)}\right)^r.$$

And finally, the expected value is

$$\mathrm{E}[Y] \leq \log(y) \cdot 3\left(p \cdot \log\left(C_3 \frac{\log n}{\log\log n}\right) + q \cdot \log(y) + c'\right)$$

with high probability. ◀

The expected number of steps when routing within a cluster can be computed for each cluster separately, i.e., the expected number of steps within a cluster of size x is $O\big(\log(x) \cdot (p \cdot \log(x) + q \cdot \log(y))\big)$. Since the expected cluster size under the Poisson distribution is λ, we have

▶ **Lemma 8** (Routing within a cluster under Poisson distribution). *Routing within a cluster takes $O\big(\log(\lambda) \cdot (p \cdot \log(\lambda) + q \cdot \log(y))\big)$ steps in expectation.*

In total, the expected number of steps for greedy routing is

▶ **Theorem 9** (Routing with Poisson-distributed cluster sizes). *Assume that the cluster sizes x follow the Poisson distribution $Pois(k, \lambda)$. Then, greedy routing from a source to a destination takes*

- $O(\log(\lambda y) \cdot (p \cdot \log(\lambda + C_1\sqrt{\lambda \log n}) + q \cdot \log(y)))$ *steps in expectation if* $\lambda > c \log n$,
- $O(\log(c \log n \cdot y) \cdot (p \cdot \log(C_2 \log n) + q \cdot \log(y)))$ *steps in expectation if* $\lambda < c \cdot \log n$,
- $O(\log(y) \cdot (p \cdot \log(C_3 \frac{\log n}{\log\log n}) + q \cdot \log(y)))$ *steps in expectation if* $\lambda = const$,

where C_1, C_2, C_3 and c are large constants.

3.3 Cluster sizes following the power law distribution

In this section, we consider the case where clusters are distributed according to the power law distribution. Other than in the previous section, this distribution allows one to have few large clusters, while most of the clusters have a constant size. We will therefore divide the clusters into intervals containing similar cluster sizes and analyze the intervals separately. A similar approach of analyzing connected components by grouping their sizes has been used in [9] on random graphs. The analysis in this section therefore differs from the previous two sections.

▶ **Lemma 10** (Routing between clusters under power law distribution). *Assume that the cluster sizes follow the Power Law distribution PowerLaw(α), i.e., $f(x;\alpha) = (\alpha-1)x^{-\alpha}$ for $\alpha \in (2,4]$ and a discrete random variable x. Then, routing between the clusters takes $O(\log(y) \cdot (4p \cdot \log\log(n) + q \cdot \log(y)))$ steps in expectation.*

Proof. Let X denote the size of a cluster. We can calculate the probability that this cluster exceeds a certain size s as

$$\Pr[X > s] = \sum_{k=s+1}^{\infty} \frac{1}{k^\alpha} \leq \int_{s+1}^{\infty} \frac{1}{x^\alpha}\, dx = \frac{1}{\alpha-1}\frac{1}{(s+1)^{\alpha-1}} < \frac{1}{s^{\alpha-1}}.$$

Note that this bound is not sufficient to upper bound the number of large clusters.

We therefore divide the n possible cluster sizes into intervals $[2^i C, 2^{i+1}C]$, where C is a large constant. The probability that the size of a cluster I lies in the interval $[2^i C, 2^{i+1}C]$ can be upper bounded by

$$\Pr\left[|I| \in [2^i C, 2^{i+1}C]\right] < \frac{1}{(2^i C)^{\alpha-1}}. \tag{3}$$

Let r denote the number of clusters traversed by the greedy routing. We can assume that these r clusters are chosen uniformly at random with the probabilities chosen as in Equation (3). The average number of clusters in an interval $[2^i C, 2^{i+1}C]$ is upper bounded by $\frac{r}{(2^i C)^{\alpha-1}}$.

Assume first that $r > c' \log^2 n$. In this case, we can apply the Chernoff bound to show that the cluster sizes of the intervals containing small clusters, where $c^i \leq \log n$, are concentrated around $n/(c^{i+1})$. Let Z be the random variable denoting the number of clusters of size $[2^i C, 2^{i+1}C]$:

$$\Pr\left[Z > 5\frac{r}{2^i C} \mid c^i \leq c'\right] < \Pr\left[Z > 5\frac{c' \log^2 n}{c^i} \mid c^i \leq c'\right] < e^{-4\log n} \leq \frac{1}{n^4}.$$

For larger i, we can upper bound the number of clusters in an interval by

$$\Pr\left[Z > c''\left(\log n + \frac{r}{2^i C}\right)\right] < e^{-c' \log n} \leq \frac{1}{n^4}.$$

Finally, we upper bound the number of clusters for any interval in the case where $r \leq c' \log^2 n$:

$$\Pr\left[Z > c''\left(\log n + \frac{r}{2^i C}\right)\right] \leq \frac{1}{n^4}.$$

In order to calculate the expected number of hops from a start to the destination, we will consider the above cases separately. First observe that in the case $r > c' \log^2 n$, where i is large, there are at most $\log n$ intervals with cluster sizes from size $\log n$ up to size n, each of which has at most $\log n$ clusters with high probability (probability larger than $(1 - \frac{1}{n^4})$).

Recall that the probability that the greedy routing needs more than r hops can be upper bounded as

$$\Pr[Y_i > r] \leq \prod_{k=1}^{r} \left(1 - \frac{1}{3(p \cdot \log(|I_k|) + q \cdot \log(y) + c')}\right)$$
$$\leq \left(1 - \frac{1}{3(p \cdot \log\left(\frac{1}{r}\sum_{k\in[r]} |I_k|\right) + q \cdot \log(y) + c')}\right)^r. \tag{4}$$

We will now upper bound $\frac{1}{r}\sum_{k\in[r]} |I_k|$ for $r > c' \log^2 n$. We therefore split the sum into intervals of size up to $\log n$ and into all larger intervals. Observe that there are at most $\log n - \log\log n$ intervals that each contain less than $\log n$ clusters of size $\log n$ to n. It would take less than $\log^3 n$ steps to traverse them in the worst case.

On the other hand, the number of clusters in each small interval is concentrated around $\frac{r}{c^i}$. This can be used to calculate the actual average cluster size.

The average cluster size of the traversed clusters is upper bounded by

$$\frac{1}{r}\sum_{k\in[r]} |I_k| < \frac{1}{r}\sum_{i=1}^{\log\log n} 2^i C \cdot \frac{r}{c^i} + \log^3 n \leq C \cdot \log\log n + \log^3 n < 2\log^3 n$$

with probability $1 - \frac{1}{n^3}$, as we can sample up to $r \leq n$ clusters in total.

For $r < c\log^2 n$, we assume that each cluster has a size in the order of n, and use the same upper bound as used for clusters of sizes $\log\log n$ to n. Note that there are up to $r < \log^2 n$ such clusters, and therefore the expected number of hops to traverse these clusters is upper bounded by $\log^2 n \cdot \log n \log n < \log^3 n$. Then we can calculate the expectation as follows:

$$\mathrm{E}[Y_i] = \sum_{j=1}^{c\log^2 n} \Pr[Y_i > j] + \sum_{j=c\log^2 n}^{n} \Pr[Y_i > j] < \; 3\big(p \cdot \log(\log^4 n) + q \cdot \log(y) + c'\big)$$
$$+ 3\big(p \cdot \log(2\log^3 n) + q \cdot \log(y) + c'\big) < \; 6\big(4p \cdot \log\log(n) + 2q \cdot \log(y) + c'\big).$$

In the final step, we again use Y to denote the number of rounds to reach the destination cluster. Since we halve the cluster distance until we end up in the destination cluster, we have $Y = Y_1 + Y_2 + \ldots + Y_{\log y}$. Then,

$$\mathrm{E}[Y] \leq \log(y) \cdot 6\big(4p \cdot \log\log(n) + 2q \cdot \log(y) + c'\big). \quad \blacktriangleleft$$

Note that in the proof of Lemma 10, we accounted for the number of steps needed to traverse large clusters in the analysis. To calculate the expected number of steps for greedy routing, we only need to add the number of steps needed to traverse clusters of size up to $c\log^2 n$. Thus, we have

▶ **Theorem 11** (Routing with power law-distributed cluster sizes). *Assume that the cluster sizes follow the Power Law distribution $PowerLaw(\alpha)$ for $\alpha \in (2, 4]$. Then, greedy routing takes $O(\log(2y\log n) \cdot (4p \cdot \log\log(n) + q \cdot \log(y)))$ steps in expectation.*

4 Extension to grid structures with equal cluster sizes

In this section, we assume that the clusters are connected in a grid as described in Section 2. The probability of communicating within a cluster or between clusters remains as in Section 2. As in Section 3, we start by presenting the demand-aware probability distribution according to which the augmenting edges are chosen.

Distribution of augmented edges

We assume that the clusters are placed at nodes of a $\sqrt{y} \times \sqrt{y}$-dimensional square grid. The distance $d_I(\cdot,\cdot)$ between any two clusters on the grid is defined in terms of their lattice distance, i.e., the sum of the horizontal and the vertical grid distances between the two clusters.

The long-range edges inside the clusters are added as described in the case of cycles in Section 3. The long-range edges to different clusters are added inversely proportional to the square of the cluster distance, i.e., proportional to $1/d_I^2(\cdot,\cdot)$.

As in the case of cycles, we start by calculating the normalization factor for adding a link to another node or cluster. Note that the sum of probabilities within a cluster remains the same. Since the clusters are arranged in a grid, each cluster has four neighboring clusters at a cluster distance of 1, eight neighboring clusters at a cluster distance of 2, etc. We can upper bound the sum of probabilities by

$$\sum_{I_i,I_j;I_i\neq I_j} 2q \cdot d_I^{-2}(I_i,I_j) \leq \sum_{\ell=1}^{2\sqrt{y}-2} \frac{4\ell}{\ell^2} < 4\log(6y^{1/2}).$$

When considering links added inside and between the clusters, we get

$$\sum_{v,w\in I_k;v\neq w} 2p \cdot d_x^{-1}(v,w) + \sum_{I_i,I_j;I_i\neq I_j} 2q \cdot d_I^{-2}(I_i,I_j) \leq 2p + 2p\log(x) + 8q\log(6) + 4q\log(y).$$

Thus, the normalization factor is lower bounded by $1/(2p + 24q + 2p\log(x) + 4q\log(y))$.

4.1 Analysis on clusters of the same size

In the following, we only analyze the case of routing between the clusters of the same size x. The analysis of routing inside a cluster remains as in the previous section.

▶ **Lemma 12** (Routing between the clusters on grids). *Routing between clusters of the same size that are arranged in a grid structure takes $O(\log(y) \cdot 4(2p\log(x) + 4q\log(y) + c))$ steps in expectation for a large constant c.*

Proof. To prove this statement, we will follow the demand-oblivious analysis on grids by Kleinberg [24]. Let I_v be the destination. We divide the analysis into phases j, where a phase is defined as the expected number of steps to half the cluster hop distance to I_v. As earlier, we are going to consider clusters as supernodes. We define a ball $B_j(I_v)$ containing all clusters at a cluster distance of at most 2^j from I_v in the grid. Note that there are at least $\sum_{i=1}^{2^j} i = \frac{(2^j+1)2^j}{2} > 2^{2j-1}$ such clusters in $B_j(I_v)$.

The probability that any node in a cluster outside $B_j(I_v)$ has a long-range edge to a cluster in $B_j(I_v)$ is lower bounded by

$$\frac{2^{2j-1}}{(2p + 24q + 2p\log(x) + 4q\log(y)) \cdot d^2(I_i,I_j)} > \frac{1}{8(2p\log(x) + 4q\log(y) + c)}.$$

From here on, the analysis continues as in the case of cycles. Let X_j denote the number of steps to reach $B_j(I_v)$. The expected value of X_j is

$$8(2p\log(x) + 4q\log(y) + c).$$

Let X denote the number of rounds to reach I_v. In every phase, the distance to the destination cluster is halved and thus $X = X_{log(\sqrt{y})} + \ldots + X_2 + X_1$. Then

$$E[X] < \log(y) \cdot 4(2p\log(x) + 4q\log(y) + c).$$ ◀

The routing within clusters can be computed similarly to Section 3.1, with an adapted normalization factor. By summing up the expected number of steps between and within the clusters, we receive the following result:

▶ **Theorem 13** (Demand-aware routing on grids). *Greedy routing on a grid consisting of y clusters, each containing x nodes, together with demand-aware augmented edges takes $O(\log(xy) \cdot 4(2p\log(x) + 4q\log(y) + c))$ steps in expectation.*

Also in this case, clusters of different size following the Poisson or the power law distribution can be considered. The main difference in the analysis between the case of cycles and grids lies in the first step where the long-range edges are added to the graph. Observe that the rest of the analysis as well as the normalization factor chosen by the nodes almost does not change. The same analysis can therefore be performed with Poisson and power law distributed cluster sized also on grids, resulting in similar bounds on the expected number of steps.

5 Empirical evaluation

We simulate our model on a cycle as described in Section 2. Our goal is to evaluate the expected routing distance (ERD) $E[G(u,v)|\tilde{D}] = \sum_{u,v \in V} G_{\tilde{D}}(u,v) \cdot D(u,v)$, where $G(\cdot,\cdot)$ denotes the greedy distance and $\tilde{D}$ is the augmented demand matrix, on instances derived from real-world datacenter traffic traces. Additionally we provide further evaluation on simulated artificial instances in Appendix A. Since our model is randomized, in our simulations we perform 100 runs and report on the averages.

We additionally compare to a "demand oblivious" model similar to that of Kleinberg [24] which we will simply refer to as the "oblivious" model. For each vertex belonging to a cluster, we add exactly one edge to one of the other vertices belonging to any cluster with probability proportional to the inverse of the shortest path distance on the demand graph D. In contrast to this, we will refer to the model in our paper as the "demand-aware" model.

5.1 Real-world instances

We base our analysis on real-world high performance computing (HPC) cluster traffic traces from Avin et al. [5]. The data comprises communication requests between pairs of nodes including timestamps. We count for each pair of nodes the number of communication requests in the trace and store the resulting counts in a matrix, disregarding the directionality of the request. Finally, the resulting matrix is normalized so that its entries sum to one. We use this as the demand matrix.

These traffic traces only contain information for nodes that participate in the communication. The original data was recorded on the HPC Hopper, which is NERC's Cray XE6 system, comprising a total of 153,216 nodes [31] (here CPU cores). Our traces contain 1024 nodes. Note that the traffic was measured on a message passing interface (MPI) not on physical NICs. The nodes in the trace are therefore CPU cores, which could be (partially) collocated on the same physical servers. Nonetheless, most nodes of the HPC did not participate in the communication in our traces.

Since we do not know the physical mapping of these nodes onto servers, to obtain clusters of nodes from these instances we try to identify groups of frequently communicating nodes. We model this as a Correlation Clustering problem, wherein for each pair of nodes a weight between 0 (the nodes are very dissimilar) and 1 (the nodes are very similar) is assigned. The goal of Correlation Clustering is to cluster similar nodes, and place dissimilar

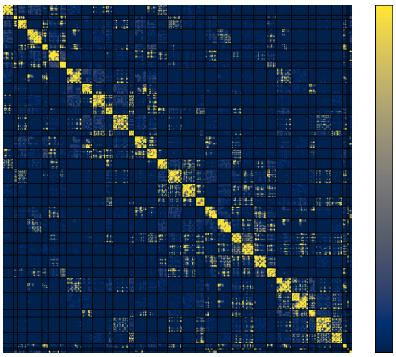


Figure 1 Visualization of the demand matrix for the "hpc nekbone" trace after clustering. Bright colors indicate frequent communication and dark indicate less communication. Black lines are drawn to separate clusters.

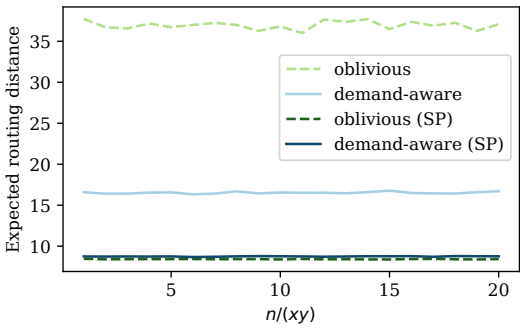


Figure 2 Impact of the fraction of non-communication nodes on the expected routing distance for the real-world traffic traces. SP denotes a variant that uses shortest path routing instead of greedy routing.

nodes in different clusters[1]. To solve these instances we use the VOTE/BOEM heuristic proposed by Elsner and Schudy [14] due to its ease of use and good quality of solutions. An example visualization of the clustered demand matrices can be seen in Figure 1. In the following we present the evaluation of the expected routing distance in the HPC clusters. An extended analysis of the inter- and intracluster communication is presented in Appendix A.2.

Expected Routing Distance

Since only a fraction of the nodes of the HPC took part in the communication, we analyze how the routing distance changes with the total number of nodes. This is summarized in Figure 2. The results are quite similar to those of the artificial instances, and the demand-aware model achieves better results than the oblivious one when using greedy routing. Note that the within cluster communication probability p varies for each node, and the cluster sizes x are not fixed, unlike in the simulations on artificial data. The theoretical bound from our analysis would be a straight line that is factor 3 larger than the other results, because it assumes the very worst case where we communicate between the two furthest nodes.

6 Conclusion

This paper presented a demand-aware analysis of small-world networks. Motivated by the structure of real-world high performance computing cluster traffic traces, we analyzed sparse demand matrices with closely communicating clusters. We proposed a demand-aware randomized edge augmentation technique based on [24] and showed that a demand-aware edge augmentation outperforms the demand-oblivious strategy. Our empirical evaluations support using demand-aware edge augmentation and show that the local greedy routing technique proposed in [24] is a good alternative to the global shortest paths routing technique on real-world datasets.

[1] Note that some preprocessing was necessary to obtain CORRELATION CLUSTERING instances from our demand matrices. Looking at the data we saw many clusters, with many in-cluster edges missing. To increase the clustering coefficient we applied the following preprocessing: for each vertex v and neighbour u: add the weight of the edge u, v divided by degree of u to v, w where w is a neighbour of u but not of v.

References

1. Emmanuel Abbe. Community detection and stochastic block models: Recent developments. *Journal of Machine Learning Research*, 18(177):1–86, 2018.
2. Vamsi Addanki, Chen Avin, and Stefan Schmid. Mars: Near-optimal throughput with shallow buffers in reconfigurable datacenter networks. *Proc. ACM Meas. Anal. Comput. Syst.*, 7(1), mar 2023.
3. Farhan Amin, Rashid Abbasi, Abdul Rehman, and Gyu Sang Choi. An advanced algorithm for higher network navigation in social internet of things using small-world networks. *Sensors*, 19(9), 2019. URL: `https://www.mdpi.com/1424-8220/19/9/2007`, `doi:10.3390/s19092007`.
4. Daniel Amir, Tegan Wilson, Vishal Shrivastav, Hakim Weatherspoon, Robert Kleinberg, and Rachit Agarwal. Optimal oblivious reconfigurable networks. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, 2022. `doi:10.1145/3519935.3520020`.
5. Chen Avin, Manya Ghobadi, Chen Griner, and Stefan Schmid. On the complexity of traffic traces and implications. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(1):1–29, 2020.
6. Chen Avin and Stefan Schmid. Renets: Statically-optimal demand-aware networks. In *Proc. SIAM Symposium on Algorithmic Principles of Computer Systems (APOCS)*, 2021.
7. Marian Boguna and Dmitri Krioukov. Navigability of complex networks. *Nature Physics*, 5(1):74–80, 2009.
8. Marián Boguná, Fragkiskos Papadopoulos, and Dmitri Krioukov. Sustaining the internet with hyperbolic mapping. *Nature communications*, 1(1):62, 2010.
9. Fan Chung and Linyuan Lu. Connected components in random graphs with given expected degree sequences. *Annals of combinatorics*, 6(2):125–145, 2002.
10. Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Designing privacy enhancing technologies: international workshop on design issues in anonymity and unobservability Berkeley, CA, USA, July 25–26, 2000 Proceedings*, pages 46–66. Springer, 2001.
11. Fred Douglis, Seth Robertson, Eric Van den Berg, Josephine Micallef, Marc Pucci, Alex Aiken, Maarten Hattink, Mingoo Seok, and Keren Bergman. Fleet—fast lanes for expedited execution at 10 terabits: Program overview. *IEEE Internet Computing*, 25(3):79–87, 2021.
12. M. Draief and A. Ganesh. Efficient routing in poisson small-world networks. *Journal of Applied Probability*, 43(3):678–686, 2006.
13. David Easley and Jon Kleinberg. *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press, 2010.
14. Micha Elsner and Warren Schudy. Bounding and comparing methods for correlation clustering beyond ilp. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, ILP '09, page 19–27, USA, 2009. Association for Computational Linguistics.
15. Klaus-Tycho Foerster, Thibault Marette, Stefan Neumann, Claudia Plant, Ylli Sadikaj, Stefan Schmid, and Yllka Velaj. Analyzing the communication clusters in datacenters. In *Proceedings of the ACM Web Conference (WWW)*, pages 3022–3032, 2023.
16. Pierre Fraigniaud, Cyril Gavoille, and Christophe Paul. Eclecticism shrinks even small worlds. In *Proceedings of the Twenty-Third Annual ACM Symposium on Principles of Distributed Computing*, PODC '04, 2004. `doi:10.1145/1011767.1011793`.
17. Pierre Fraigniaud and George Giakkoupis. On the searchability of small-world networks with arbitrary underlying structure. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, STOC '10, page 389–398, New York, NY, USA, 2010. Association for Computing Machinery. `doi:10.1145/1806689.1806744`.
18. Massimo Franceschetti and Ronald Meester. Navigation in small-world networks: a scale-free continuum model. *Journal of Applied Probability*, 43(4):1173–1180, 2006. `doi:10.1239/jap/1165505216`.

19 Monia Ghobadi, Ratul Mahajan, Amar Phanishayee, Nikhil Devanur, Janardhan Kulkarni, Gireeja Ranade, Pierre-Alexandre Blanche, Houman Rastegarfar, Madeleine Glick, and Daniel Kilper. Projector: Agile reconfigurable data center interconnect. In *Proceedings of the 2016 ACM SIGCOMM Conference*, pages 216–229. ACM, 2016.

20 Šarunas Girdzijauskas. *Designing peer-to-peer overlays: a small-world perspective*. PhD thesis, EPFL, Lausanne, 2009. URL: `https://infoscience.epfl.ch/handle/20.500.14299/33278`, `doi:10.5075/epfl-thesis-4327`.

21 Chen Griner, Johannes Zerwas, Andreas Blenk, Manya Ghobadi, Stefan Schmid, and Chen Avin. Cerberus: The power of choices in datacenter topology design (a throughput perspective). *Proc. ACM Meas. Anal. Comput. Syst.*, 5(3), dec 2021.

22 Matthew Nance Hall, Klaus-Tycho Foerster, Stefan Schmid, and Ramakrishnan Durairajan. A survey of reconfigurable optical networks. *Optical Switching and Networking*, 41:100621, 2021.

23 Navid Hamedazimi, Zafar Qazi, Himanshu Gupta, Vyas Sekar, Samir R Das, Jon P Longtin, Himanshu Shah, and Ashish Tanwer. Firefly: A reconfigurable wireless data center fabric using free-space optics. In *ACM SIGCOMM Comput. Commun. Rev. (CCR)*, volume 44, pages 319–330. ACM, 2014.

24 Jon Kleinberg. The small-world phenomenon: an algorithmic perspective. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, STOC '00, 2000. `doi:10.1145/335305.335325`.

25 Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguñá. Hyperbolic geometry of complex networks. *Phys. Rev. E*, 82:036106, Sep 2010. `doi:10.1103/PhysRevE.82.036106`.

26 Mei Li, Wang-Chien Lee, and A. Sivasubramaniam. Semantic small world: an overlay network for peer-to-peer search. In *Proceedings of the 12th IEEE International Conference on Network Protocols, 2004. ICNP 2004.*, pages 228–238, 2004. `doi:10.1109/ICNP.2004.1348113`.

27 Yu A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(4):824–836, apr 2020. `doi:10.1109/TPAMI.2018.2889473`.

28 Yu A Malkov and Dmitry A Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836, 2018.

29 Chip Martel and Van Nguyen. Analyzing kleinberg's (and other) small-world models. In *Proceedings of the Twenty-Third Annual ACM Symposium on Principles of Distributed Computing*, PODC '04, 2004. `doi:10.1145/1011767.1011794`.

30 Stanley Milgram. The Small World Problem. *Psychology Today*, 2:60–67, 1967.

31 NERC. Hopper, nercs's cray xe6 system. `https://web.archive.org/web/20120717045546/https://www.nersc.gov/use\protect\penalty\z@rs/computational-systems/hopper/`.

32 Fragkiskos Papadopoulos, Dmitri Krioukov, Marian Boguna, and Amin Vahdat. Greedy forwarding in dynamic scale-free networks embedded in hyperbolic metric spaces. In *2010 Proceedings IEEE INFOCOM*, 2010. `doi:10.1109/INFCOM.2010.5462131`.

33 David Pollard. The moment generating function method. *Yale Univ., Mini-Empirical Rep*, 2021.

34 Fatemeh Shirazi, Milivoj Simeonovski, Muhammad Rizwan Asghar, Michael Backes, and Claudia Diaz. A survey on routing in anonymous communication protocols. 51(3), 2018. `doi:10.1145/3182658`.

35 Min Yee Teh, Zhenguo Wu, and Keren Bergman. Flexspander: augmenting expander networks in high-performance systems with optical bandwidth steering. *IEEE/OSA Journal of Optical Communications and Networking*, 12(4):B44–B54, 2020.

36 D.J. Watts and S.H. Strogatz. Collective dynamics of 'small-world'networks. *Nature*, 393(6684):440–442, 1998.

37 Tegan Wilson, Daniel Amir, Nitika Saran, Robert Kleinberg, Vishal Shrivastav, and Hakim Weatherspoon. Breaking the vlb barrier for oblivious reconfigurable networks. In *Proceedings*

of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, 2024. doi:10.1145/3618260.3649608.

38 Johannes Zerwas, C Gyorgyi, A Blenk, Stefan Schmid, and Chen Avin. Duo: A high-throughput reconfigurable datacenter network using local routing and control. In *Proc. ACM SIGMETRICS*, 2023.

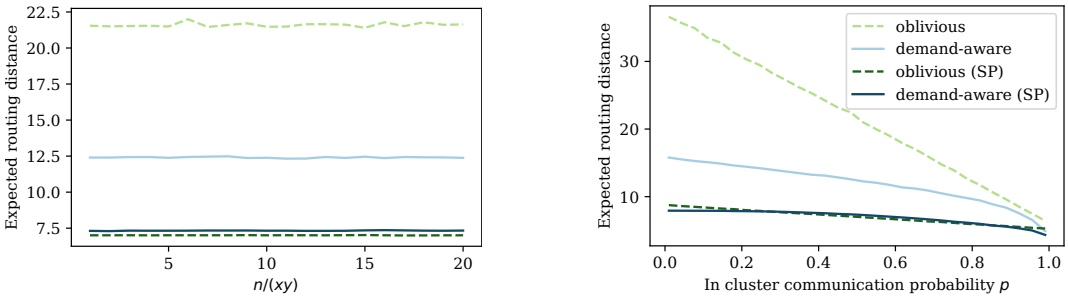
39 Mingyang Zhang, Jianan Zhang, Rui Wang, Ramesh Govindan, Jeffrey C Mogul, and Amin Vahdat. Gemini: Practical reconfigurable datacenter networks with topology and traffic engineering. 2021. arXiv:2110.08374.

A Further empirical evaluation

A.1 Artificial instances

We construct artificial instances parameterized by the values p, q, x, y, n where $q = 1 - p$. More specifically we create a demand matrix D for the n nodes, in which only $x \cdot y$ nodes have some demand to each other. The remaining nodes do not participate in the communication, and have a demand of 0 to all other nodes. The demand matrix is constructed in such a way that a node communicates with its own cluster with probability p, and among the $x - 1$ other nodes in the cluster the probability is divided uniformly. With probability q, a node communicates outside its own cluster, and the probability is divided uniformly among the $(y - 1) \cdot x$ possible nodes. Finally, we embed y clusters of x nodes each onto a cycle with n nodes, such that the spacing between the clusters differs by at most 1.

We perform experiments, to analyze the impact of the various instance parameters on the expected routing distance. Since the number of parameters is quite high, we fix all but one parameter and vary the remaining parameter. See Figure 3 for the results.



(a) Fixed parameters $p = q = 0.5, x = 5, y = 20$

(b) Fixed parameters $x = 16, y = 32, n = 3xy$

Figure 3 Simulation results on artificial data. Different instance parameters are varied and their impact on the routing cost is depicted. The two routing strategies: greedy and shortest path (SP) are compared.

From Figure 3a it is apparent, that the routing distance does not change at all. As desired and expected (by Theorem 3) our model has no dependence on the number of nodes that do not participate in the communication. The greedy and shortest path routing strategies appear quite close for the demand-aware regime, with the shortest path routing resulting in roughly 70% smaller distances.

In Figure 3b we analyze the dependence of the expected routing distance on the probability p that the communication of a node falls within its own cluster. For the oblivious setting, this appears to be a linear dependence. Instances with small values of p, indicating that the nodes in the clusters mostly communicate outside their own clusters, exhibit smaller routing distances in the demand-aware model than in the oblivious one. This is due to the fact that

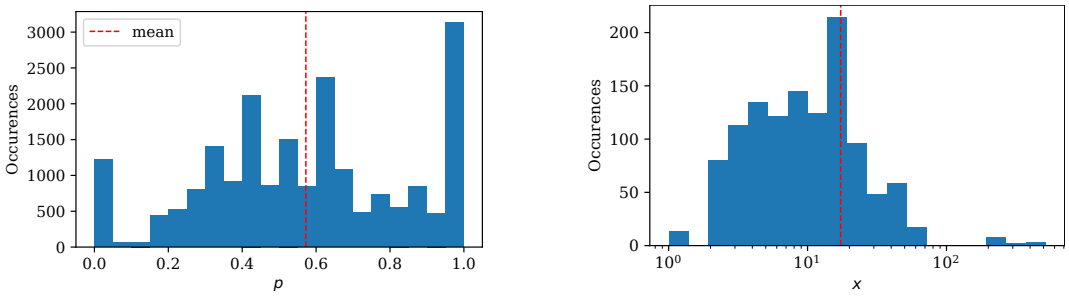


Figure 4 Histogram of the cluster sizes x and within cluster communication probabilities p on the traffic traces.

the demand-aware model is more likely to augment the graph with shortcut edges to more distant clusters than the oblivious model. For large p both models perform similarly, as most augmenting edges will be added within a cluster or to nodes close to the current cluster.

A.2 Parameter analysis of real-world instances

Based on the computed clustering we compute the probability p that a node communicates within its own cluster, the cluster sizes x and the number of clusters y. This is summarized in Figure 4. The within cluster communication probability p appears somewhat uniformly distributed, and the cluster sizes x are usually at most 20 with few larger ones. Some traces contain up to 150 clusters, but the median is around 30 clusters per trace.