Cutting the Cord: a Robust Wireless Facilities Network for Data Centers

Yibo Zhu, Xia Zhou[§], Zengbin Zhang, Lin Zhou, Amin Vahdat[†], Ben Y. Zhao and Haitao Zheng
University of California, Santa Barbara [§]Dartmouth College [†]University of California, San Diego & Google
{yibo,zengbin,linzhou,ravenben,htzheng}@cs.ucsb.edu, xia@cs.dartmouth.edu, vahdat@cs.ucsd.edu

ABSTRACT

Today's network control and management traffic are limited by their reliance on existing data networks. Fate sharing in this context is highly undesirable, since control traffic has very different availability and traffic delivery requirements. In this paper, we explore the feasibility of building a dedicated wireless *facilities network* for data centers. We propose *Angora*, a low-latency facilities network using low-cost, 60GHz beamforming radios that provides robust paths decoupled from the wired network, and flexibility to adapt to workloads and network dynamics. We describe our solutions to address challenges in link coordination, link interference and network failures. Our testbed measurements and simulation results show that Angora enables large number of low-latency control paths to run concurrently, while providing low latency end-to-end message delivery with high tolerance for radio and rack failures.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

Keywords

Data centers, 60 GHz wireless, wireless beamforming

1. INTRODUCTION

With the recent rapid growth in data center networks, come dramatic increases in management complexity. Yet despite advances in Software Defined Networks (SDNs) [11, 15] and network/traffic engineering design [7, 10, 12], little has changed in how data centers deliver control traffic.

We believe the time is right to introduce the *facilities network* as a core tool for managing data center networks. The facilities network is orthogonal to the data plane, and is responsible for multiple critical jobs. For example,

- *Control Plane* – Data center networks require an orthogonal network to support network control protocols. With the arrival of SDNs, a variety of protocols will traverse the control plane between network control servers [28] and hardware switches. For example, the control plane can install forwarding table updates or reset hardware in response to switch failures in the data plane [40].
- *Facility Bring up and Installation* – Hardware devices do not arrive with appropriate operating system images, device configurations, or security keys. These images must be installed on the data center floor. While the process is automated, it cannot rely on a data plane network to communicate with switches/servers, because much of the data plane could be inoperative.

MobiCom'14, September 7-11, 2014, Maui, Hawaii, USA.

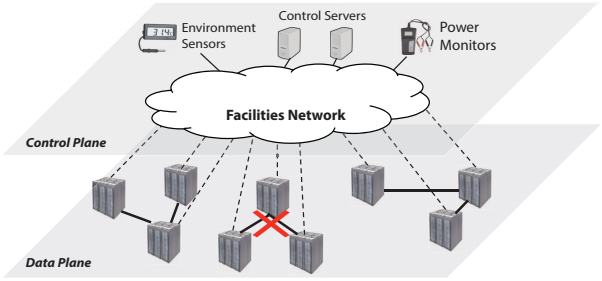


Figure 1: A facilities network providing robust delivery of control traffic in a data center.

Compared to traditional data plane networks, the facilities network has very different requirements. First, it requires substantially lower bandwidth, *e.g.* hundreds of Mbps rather than 40Gbps [7, 10, 12], and its bandwidth demands grow at much slower rates than those of the data plane. Second, control traffic has much tighter constraints on packet delivery latency. Even "moderate" delivery delays to route updates or policy changes can have a dramatic impact on overall network performance [12, 32, 33].

Third and most importantly, the facilities network requires significantly higher availability and long-term survivability than the data plane. Hence, it cannot share fate with the data plane network, and must be isolated from faults and outages in the data plane. For example, it must remain available during constant physical upgrades to the data center. In a facility with 10's−100's of MW of infrastructure, substantial portions of the building are always undergoing upgrades that may tear out entire rows of cable trays. The facilities network should remain available for the lifetime of the building, *e.g.*, 10−20 years, rather than the lifetime of a cluster, *e.g.*, 3−5 years. While brief outages in data plane networks are acceptable, the underlying facilities network must fail last and fail least, because it is the basis for recovering from data plane failures and monitoring critical hardware. This rules out in-band solutions such as VLANs [5].

We show an example of a facilities network in Figure 1, where an orthogonal facilities network provides control servers with access to sensors at data center racks, environmental sensors, and power

junctions. Even as the data network experiences failures from hardware faults or planned maintenance, the facilities network remains available.

Design Space. Given its substantially lower bandwidth and higher availability requirements, one approach to building a facilities network is simply to use a second, smaller instance of the data plane network. However, this faces a number of practical challenges. First, a data plane network is built for speed. 40Gb/s ports are likely overkill for a facilities network. The size and reach of the network would also dramatically increase hardware costs. Second, data plane networks are wired networks. Whenever a new switch, power unit, A/C unit or battery is added, it must be connected via cables. Planning the necessary cable tray infrastructure to connect arbitrary points in the network is a costly logistical challenge [2, 38].

Third, cables in the facilities network would likely be copper (for cost, reach, and compatibility with older devices), and typically cannot coexist with optics in the same cable tray because of interference and weight issues [31]. Further, copper cables are 10x the bulk of optical fiber [23]. So while the facilities network has fewer ports, the bulk of the control cable infrastructure may actually exceed the data plane.

Finally, fault isolation is a challenge. It is difficult to build a wired control infrastructure that is orthogonal to and independent of a data plane, because the data plane undergoes physical upgrades on a regular basis. Upgrading power and cooling units is a common operation in the maintenance of a data center building. The simplest and fastest way[1] is to cordon off an area and "bulldoze" equipment. But if a wired control infrastructure shares physical facilities, *e.g.* cable trays, with the data plane, they will suffer the same outage patterns. Trying to isolate and preserve a "bisection" of copper cables for the wired facilities network in practice is difficult or intractable.

A Wireless Facilities Network. These issues motivate us to explore the design and architecture of a wireless facilities network *physically decoupled from the data plane*. We consider both wireless networks on commodity WiFi, and networks built using recent advances on 60GHz 3D beamforming links [44]. Our experiments confirm that contention-based access produces large, unpredictable packet delivery delays in WiFi networks, far outside the acceptable range for a facilities network. Instead, we consider using directional wireless links in the unlicensed 60GHz band. These links are highly directional, provide high data rates, and attenuate quickly with distance [20, 44]. This limits interference footprint and allows multiple wireless links to transmit simultaneously.

Two challenges follow from the choice of directional wireless links. First, based on significantly better throughput and interference characteristics, directional links would be implemented using horn antennas over phased antenna arrays. However, the benefits come at the cost of slow tuning delays from the physical tuning mechanism. This significantly limits the ability of the facilities network to support low-delay communication between racks and control servers. Second, even with much smaller interference footprints, directional links can still experience interference and associated delays in a high density setting.

In this paper, we describe the design of *Angora*, a facilities network that employs a structured 60GHz wireless overlay to support low delay, robust, any-to-any communication. We summarize the key contributions of our work below.

[1]Upgrade speed translates directly to dollars; consider the depreciation cost of leaving 10MW of servers idle for four weeks.

- First, we propose a fixed structured network design for Angora based on Kautz graphs, which addresses the issue of directional link coordination and antenna tuning delays. This provides any-to-any communication with bounded delays, and eliminates link coordination except when significant numbers of racks fail or move their positions.
- Second, we use location-aware ID assignment to manage physical positioning of wireless links and reduce interference between nearby directional flows. This improves Angora's ability to support parallel wireless links.
- Third, we modify the Kautz graph to support arbitrary network sizes, and design routing algorithms that leverage redundant paths for fault recovery.
- Finally, we evaluate Angora using both simulations and experimental measurements on a 60GHz testbed. We find that Angora paths work well in practice: the 3D beamforming links are very stable over time, and offer throughput and latency comparable to wired networks. Angora's structured network topology creates large angular separation between nearby links, effectively minimizing interference and producing low, predictable end-to-end message latency. Simulations show that Angora can provide high levels of flow concurrency and low-latency packet delivery, and is highly robust against node and link failures.

Network management is an important challenge for data centers growing in both size and complexity. Our work provides a first step towards a robust facilities network capable of delivering control traffic with low latency, even during periods of disruption that could hinder or disrupt portions of the data network. As network managers gain experience deploying and operating facilities networks, we expect their requirements to continue to adapt to an evolving set of monitoring and management applications.

2. REQUIREMENTS AND DESIGN SPACE

We now describe the requirements for facilities networks, and explore the design space and basic challenges. We assume standard wired connectivity from servers to Top of Rack (ToR) switches, and focus on building a facilities network at the rack level.

2.1 Requirements

A viable design for a data center facilities network must satisfy three key requirements.

Reliability. The facilities network must be reliable under equipment failures and upgrades, and support a range of management applications [7, 28, 37]. Upon single component (*e.g.* radio, rack) failures or the removal of an entire row of racks for upgrades, it must continue to route traffic around failures with minimal delays.

It must also support management applications running on either a single controller [7] or a cluster of controllers [28, 37]. Because controllers can run on any racks, the facilities network must adapt to different controller configuration and placement. This argues for support of *any-to-any* rack communication.

Bounded Delay. An effective facilities network must deliver control traffic within bounded time [28]. For example, flow control mechanisms such as Hedera [7] require the controller to pull a large volume ($\approx$ 1.3MByte) of flow statistics from each switch within hundreds of ms [12]. Similarly, a common SDN paradigm [11] involves punting the first packet of flows to a central controller to install specific per-flow hardware rules.

Scalability. The implementation of the facilities network must scale to large data centers. In particular, protocols for the facilities network must incur minimal overhead and scale gracefully to a large number of flows.

2.2 Candidate Wireless Solutions

With these requirements in mind, we consider potential wireless technologies for the facilities network.

WiFi. WiFi is the obvious choice given its cost and availability. The problem, however, is its large interference footprint. In densely packed data centers, WiFi MIMO or channel allocation techniques can only mitigate interference to a limited degree. Thus flows will contend for medium access, resulting in large, unpredictable contention latencies. For example, with only 10 competing flows, it takes up to $2s$ to download a 2MB message using 802.11n [1]. We further confirm this by performing latency experiments where racks upload messages to a central controller using off-the-shelf 802.11n and 802.11ac equipment[2]. As expected, upload latency per message grows with the number of competing flows, and varies significantly across flows. 20 1.3MB 802.11n flows can take anywhere from $539ms$ to $3.6s$ to complete, with an average of $2.5s$.

While recent advances in multi-user MIMO/interference alignment allow multiple flows to run simultaneously, they require significant coordination among transmitters, which can translate into unpredictable MAC latencies. Clearly, it would be difficult for a WiFi-based network to meet the bounded latency requirements in dense data center environments.

Another concern regarding WiFi is the danger of information leakage and electromagnetic attacks. In particular, attackers can also use high-power radiation devices to produce heavy interference and disrupt an entire facilities network.

60GHz. Recent studies have proposed the use of 60GHz links in data centers [20, 35, 44]. 60GHz technology operates on a band of 7GHz unlicensed spectrum, with multi-Gbps data rates at a range of 100+m [44]. 60GHz has several features that enable it to provide wired-like connectivity.

- *Stability*: Placed on the top of server racks of 7-9 feet in height, 60GHz static links are *highly stable* in the data center scenario. Their transmissions follow the free-space propagation model without multipath fading [20, 44]. We also confirmed this by testing two different off-the-shelf 60GHz radios over 3 weeks (see §5.1). The standard deviation of link throughput is less than 0.5% of the average throughput.
- *Small interference footprint*: 60GHz links are directional and experience fast link attenuation. In particular, we leverage recent work on 3D beamforming [44], which reflects 60GHz beams off the ceiling[3], bypassing obstacles in the 2D plane and further reducing the interference footprint.
- *Security*: 60GHz signals are directional and cannot penetrate walls or large obstacles, thus are generally immune to eavesdropping and electromagnetic attacks in data centers.
- *Availability*: Recent low-cost silicon implementations make 60GHz radios affordable. While high-end 60GHz radios offer 1Gbps at 100+m [44], the WiloCity chipset costs only $37.5, and already offers 1Gbps at 20m using a basic 2x8 antenna array. Its range can be extended using narrow-beam antennas [20]. Since typical med-sized data centers (*e.g.* 320 racks) require a 40-50m range, 60GHz and WiFi hardware costs and energy consumptions are comparable.

Given these considerations, we rule out WiFi as the core transmission technology, and instead consider potential facilities network designs using 60GHz beamforming links.

[2]802.11ac: Netgear 6300 AP with 6200 adapters; 802.11n: D-Link DIR-825 AP with laptops using BCM4322/4331 chipsets.

[3]3D beamforming requires clearance above racks, which is already in place based on images of some of Google's data centers [3].

One may consider a hybrid wired-wireless solution that places 60GHz radios on the ceilings (as access points) and 60GHz radios on top of the racks (as stations), forming LoS links between the APs and stations. We can connect these 60GHz APs via a wired backbone (on the ceiling). While feasible, this solution faces the same fault tolerance problem, *i.e.* the "bulldoze" problem stated in the introduction. It also requires additional cabling and switches to connect the 60GHz APs on the ceiling. Therefore, in this paper we focus on designing a completely wireless solution using 60GHz 3D beamforming, which does not require any wired backbone.

2.3 Key Design Challenges

While 60GHz links seem promising as a link layer primitive for a facilities network, there are a number of challenges to address in any practical system.

Challenge 1: Coordination of Directional Links. The first problem is link coordination. 60GHz radios are highly directional, and thus must align their antennas before communication. Current 60GHz data center designs set up links dynamically [20, 44], requiring an extra "coordination" signaling path and a central scheduler to coordinate end-points. This brings considerable latency and complexity when deploying and managing the facilities network.

Challenge 2: Limited Number of Radios. Fixed surface area on server racks limits the number of radios placed atop each rack. For example, today's standard rack is 4ftx2ft and a 60GHz radio is 1ftx1ft [44], so at most 8 radios can sit atop each rack. Because 60GHz links are highly directional, each rack can only communicate with a small, constant number of peers in parallel. This limited "node degree" makes it hard for a rack to reach any other rack with bounded delay.

Challenge 3: Link Interference. 60GHz links produce small interference footprints, which are further reduced using 3D reflection. However, two links still interfere if their destinations are close and their arrival angles are similar. Such interference makes it hard to guarantee bounded delay.

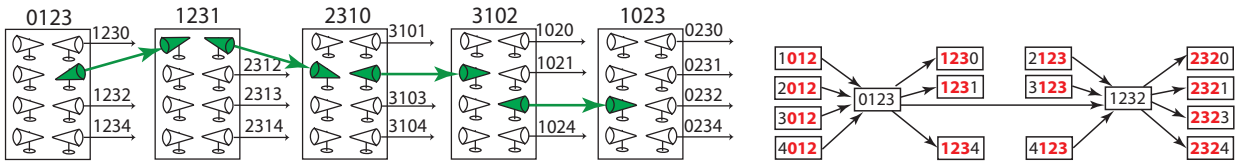
3. ANGORA 60GHZ OVERLAY

To address the above challenges, we introduce *Angora*, a wireless facilities network formed by connecting server racks with static 60GHz 3D beamforming links. Angora is a *schedule-free wireless overlay*, built using 60GHz radios with fixed antenna directions, inter-connecting all racks using a small *constant* number of hops (see Figure 2(a)).

This facilities network design provides three key benefits.

- Antenna directions are pre-tuned and fixed for a given topology[4], and no antenna alignment/rotation is necessary regardless of changing traffic patterns. This eliminates the need for link coordination and associated delays, while simplifying bootstrapping and fault detection/recovery.
- A well-designed overlay guarantees short paths between any two racks, with a maximum diameter that scales logarithmically with the number of racks. Thus, latency between two racks is both small and predictable.
- The overlay supports any model of control traffic. A single network can support arbitrary control traffic patterns, including one-to-one, one-to-many and many-to-many.

[4]Physical rack placements in data centers rarely change, except during upgrades or repairs. Thus it is feasible to fix antenna directions for a given overlay topology while applying fault recovery (see §4.3) to handle unexpected interruptions.



(a) The Angora overlay built from 3D beamforming links: an example path

(b) Kautz Overlay

Figure 2: High-level view of the Angora wireless facilities network. (a) An example path from one ToR switch (rack ID 0123) to a controller hosted in the rack of ID 1023. Each rack has 8 radios (connected by a small switch), 4 for incoming and 4 for outgoing links. Radios used by this Angora path are in green, and each green arrow represents a 3D beamforming link. (b) Two nodes in a Kautz graph (d=4,k=4). Each node points to 4 nodes whose $(k-1)$-digit prefix matches its own $(k-1)$-digit suffix.

The key to attaining these properties is the structure of the overlay network. We considered two approaches, an unstructured approach, *i.e.* a random topology, and a structured approach, *i.e.* using a compact routing graph.

A Randomized (Unstructured) Approach. Wireless radios give us the flexibility to set up or reconfigure wireless links between any two racks as necessary. In a simple, *randomized* approach to network construction, links are created by connecting pairs of randomly chosen nodes (racks) with available ports. Prior work in wired data center networks shows that a randomized graph topology can provide flexibility and short paths [36].

In our context, a randomized graph topology offers several potential benefits. First is simplicity. Implementing randomized link creation is straightforward and potentially more robust against implementation errors. Second, randomized topologies should offer short paths in practice [36]. Finally, they should provide built-in path redundancy, potentially offering high availability and robustness to random failures.

The tradeoff, is unpredictability. While random graphs provide good properties for *most* paths, there will always be a non-insignificant tail in any probabilistic distribution, *i.e.* some portion of all paths will experience undesirable properties such as longer paths or high link interference. These performance artifacts will have an outsized impact on overall performance, as end-to-end performance is usually determined by the weakest link [14].

A Structured Approach. The alternative is to impose structure on the overlay in return for stronger performance guarantees and more predictability. A naive approach is to organize all links into a single tree rooted at some network controller. This, however, makes a strong assumption that traffic patterns (and controller locations) are static, and more importantly, known to the system *a priori*. It also limits bisection bandwidth and utility of parallel flows.

Instead, we turn our attention to Kautz graphs [26], a structured graph topology that guarantees paths between all node pairs are bounded. We chose it over other well-known structures such as de Bruijn graphs [13] because of four reasons. *First*, for a network of size N, Kautz graph guarantees the path between any two nodes has at most $\log_d(\frac{d}{d+1} \cdot N)$ hops, thus strictly bounding latency between any two racks. Such guarantee on all node pairs also eliminates the need for knowing traffic patterns and controller locations a priori. *Second*, Kautz graphs require only constant degree per node, which address our constraint of limited radios per rack. Specifically, each node/rack has d incoming and d outgoing edges, which translates into $2d$ radios placed atop each rack. *Third*, relative to de Bruijn (and other graphs), Kautz graphs distribute flows more evenly through the network, and guarantee a smaller network diameter[5]. *Finally*, routing on a Kautz topology is simple and uses digit-shifting, which is easily implemented in today's switches using prefix-matching.

We now briefly describe the Kautz topology to provide context for Angora. In a Kautz (d, k) graph with N nodes (racks), each node's out-degree d and the graph diameter k satisfy the equation $N = d^k + d^{k-1}$. Each node's ID has k digits with base $d+1$. We represent a nodeID by $x_0x_1..x_{k-1}$, where $x_i \neq x_{i+1}, x_i \in [0, d]$, and $0 \leq i < k$. A node n_i keeps pointers to d nodes whose first $k-1$ digits match n_i's last $k-1$ digits (shown in Figure 2(b)). Routing proceeds by choosing the successor who will match the destination ID with one more digit, left-shifting the node ID by one digit at each hop. For example, the route from 0123 to 4321 would proceed as $0123 \rightarrow 1234 \rightarrow 2343 \rightarrow 3432 \rightarrow 4321$. Thus at most k hops connect any two nodes. For a network of 320 racks, each with 8 radios, any rack can reach another within 4 hops.

Summary. Given the above discussion, we choose the Kautz topology over a probabilistic, randomized network topology to satisfy the deterministic performance properties desired in a data center context. Our choice echoes designs made years ago in the peer-to-peer networking space, where structured overlays with bounded performance guarantees were chosen over random topologies [16]. A structured network is also easier to bootstrap. Each node's nodeID can be statically mapped to an unique IP address using a carefully chosen hash function. We will also show in §4 that the organized structure of Kautz graphs enables further reduction of wireless interference in data centers by optimizing node ID assignment.

4. ANGORA DESIGN

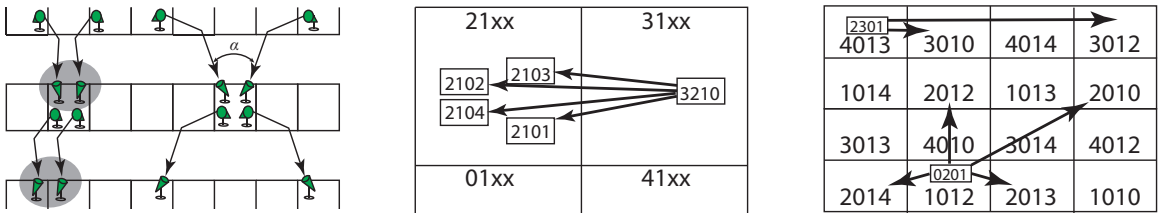
Angora's basic design addresses two fundamental challenges facing the facilities network: pre-tuned antenna directions remove the need of link coordination, while constant-degree overlay supports any traffic pattern using a fixed set of radios. When implementing Kautz graphs using wireless links, however, we identified a new set of practical challenges: 1) handling the correlation between link interference and ID assignment, 2) developing Kautz graph algorithms to support incomplete graphs, and 3) providing robustness against node and link failures. We now describe our solutions to these three key challenges.

4.1 Interference-Aware ID Assignment

Our first challenge is to assign logical node IDs in a Kautz graph to physical racks in the data center[6]. This assignment determines link placement and thus network interference conditions. Even with 3D beamforming 60GHz links, interference is a serious problem in dense data centers. A suboptimal link placement can de-

[5]For a fixed degree d and node count $N = d^k + d^{k-1}$, the Kautz graph has the smallest diameter of any possible directed graph.

[6]Given the small surface space atop each rack, we found the mapping of radios to links at each node has minimum impact on overlay performance. We thus applied a random mapping uniformly to all the nodes. This differs from [44] which requires antenna rotation.



(a) Link interference (b) Hierarchical ID Assignment (c) Hybrid ID Assignment

Figure 3: (a) Using 3D beamforming, nearly parallel links targeting the same region can interfere with each other, but increasing the 3D angular separation α between links reduces interference. (b) Hierarchical ID assignment increases angular separation from many nodes to a single target, but links from a single source to nearby destinations can still interfere. (c) The proposed Hybrid ID assignment interleaves nodes by prefix and suffix, increasing angular separation and minimizing interference for links. For both (b) and (c), each directed link represents a ceiling-reflected beamforming link toward the destination, as illustrated in (a).

grade link concurrency, reducing the number of simultaneous transmissions possible.

Intuition says that 60GHz links targeting nearby destinations will experience stronger interference, but interference can be effectively reduced by increasing 3D angular separation between two links, *i.e.* α in Figure 3(a). We test this property using two types of off-the-shelf 60GHz radios. Results confirm the correlation with 3D angular separation, and found that in general, 10-12° of angular separation is sufficient to nullify link interference (see §5.1). Motivated by this intuition, we seek to identify node ID assignments that can maximize 3D angular separations between links.

Naive Solutions. Our search started from a few intuitive solutions, *i.e. sequential*, where we assign IDs to racks in row or column order, *Hilbert*, where we assign IDs by ordering the racks using a space-filling Hilbert Curve [22], and *random* ID assignment. We run simulations to test these schemes in terms of path concurrency under various traffic patterns. Results show that both sequential and Hilbert produced significant interference and reduced concurrency, allowing at most 15-20% of links to be simultaneously active. This is because in a Kautz link, the source and destination IDs match in all but 1 digit, *e.g.* 0123 points to 1230, 1231, 1232 and 1234 (Figure 2(b)). Thus, for both sequential and Hilbert assignments a sizable number of link pairs will experience significant interference. Random assignment performs better, but is unpredictable, and a small portion of assignments always experienced poor concurrency.

Hierarchical Scheme. Instead, we propose a "hierarchical" scheme where we divide the data center into d regions. For every group of d nodes whose IDs differ only on the first digit, *e.g.* $0xyz$, $2xyz$, $3xyz$, and $4xyz$, we assign them into each of the d different regions (Figure 3(b)). This maximizes angular separation for *incoming links* to a single rack/node, since the previous hop for each node comes from a different region. Experiments confirmed that this gets near-perfect concurrency for all-to-one traffic workloads but does not address interference between links coming from the same rack, *i.e.* one-to-many workloads. As shown in Figure 3(b), the 4 outgoing links from 3210 arrive at closely located racks and interfere with each other.

Hybrid Scheme. A better "hybrid" assignment can maximize angular separation between pairs of incoming links and also between pairs of outgoing links of each rack. For clarity, we use *middle-digits* to refer to all digits of a nodeID except the first and last digits. To achieve maximum angular separation, we partition the network layout into d^2 regions. d^2 nodes share each unique string of middle-digits. For example, in a Kautz (4,4) graph, there are 16 nodes with ID that matches $x01y$, where $x \in \{1, 2, 3, 4\}$, and $y \in \{0, 2, 3, 4\}$. We distribute them into the d^2 different regions such that IDs that differ only in the first or the last digit will be maximally separated in physical space.

The layout shown in Figure 3(c) is for node degree $d = 4$. Consider rack 0201; its outgoing links point to 2010, 2012, 2013 and 2014, which are well-separated from each other. Also consider rack 2301's two outgoing links pointing to 3010 and 3012. The two links might appear to be in parallel, yet they are well-separated in the 3D beamforming context.

LEMMA 1. *The hybrid ID assignment scheme achieves the optimal angular separation between pairs of incoming (and outgoing) links on each rack.*

The proof is in the Appendix. We also show that in a typical rectangular data center layout, the above angular separation is at least 14°. This means that with today's off-the-shelf radios, the hybrid ID assignment eliminates interference between incoming (and outgoing) links on each rack.

Optimal ID Assignment & Channel Allocation. Ultimately, we seek to minimize interference among all links in the network. However, doing so is challenging - finding the optimal ID assignment is NP-hard (proof omitted due to space limits). Fortunately, Angora can leverage channel allocation to reduce interference across potentially interfering links. In this paper, we apply a simple greedy channel assignment because it already leads to reasonable performance under heavy traffic (see §6). We leave optimization of ID assignment and channel allocation to future work.

4.2 Handling Arbitrarily Sized Networks

Our next challenge comes from the fact that Kautz graph algorithms do not currently address *incomplete* graphs, *i.e.* networks whose size does not match a complete Kautz graph where $N = d^k + d^{k-1}$. Prior algorithms for routing in incomplete Kautz graphs [18, 19] are unusable because they require node in-degrees (thus the number of radios per rack) to grow arbitrarily. De Bruijn graphs face the same problem. Thus we need a solution that makes it easy to incrementally grow the Kautz graph, *i.e.* add racks to grow from a Kautz (d, k) network to a Kautz $(d, k + 1)$ network, all while maintaining the Kautz properties (bounded degree per node and bounded hops between any pair of nodes).

Growing a Kautz Graph. Our solution works by allowing nodes with different length nodeIDs to coexist in the same network. We start with a complete Kautz (d, k) network of $d^k + d^{k-1}$ nodes, where each nodeID has k digits. We add new nodes to the network by assigning nodeIDs of length $k + 1$, and inserting them into the middle of existing links. We show an example in Figure 4, where we add new nodes to a Kautz (4,4) graph to become an incomplete

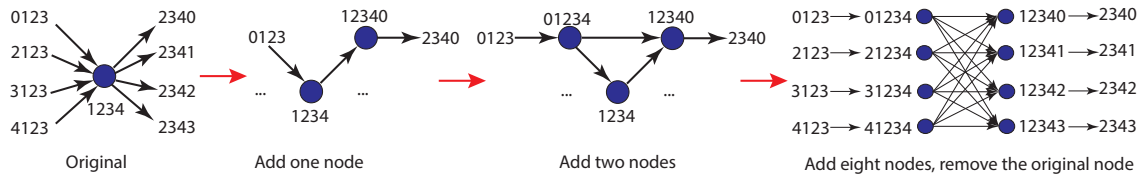


Figure 4: Add seven nodes to existing graph by replacing Node 1234 with eight new nodes 01234, 21234, 31234, 41234, 12340, 12341, 12342, 12343. The eight new nodes together perform the same functionality as Node 1234 from outside's view.

Kautz (4,5). To add a new node n_i, we first randomly choose a 4-digits node n_0, *e.g. 1234*, from Kautz (4,4). Then we assign a 5-digit nodeID to new node n_i, and insert it into one of links adjacent to n_0. If the link routes to n_0, we set n_i's ID by adding 1 digit prefix to n_0; otherwise we add 1 digit suffix.

When adding another node n_j, we repeat the process on another link adjacent to n_0. After inserting n_j, we also check if there are other existing 5 digit nodes it should link to based on the Kautz (4,5) protocol. In Figure 4, adding new node 01234 requires creating an additional link to node 12340. New nodes are added sequentially until all incoming and outgoing links to n_0 have new nodes attached, except 1. When this happens, the original n_0 node modifies its nodeID to 5 digits, and attaches itself to the last link. In our example, 1234 becomes 12343 after 7 new nodes have been inserted into all but 1 of its adjacent links.

This process is repeated at every node in the original Kautz (d, k) network, gradually replacing each original node with $2d$ new nodes with $k+1$ digit nodeIDs. At this point, the entire network will be a complete Kautz $(d, k+1)$ graph. During this process, the network is an *incomplete* Kautz $(d, k+1)$ graph and all properties of the Kautz graph hold: a) each node has at most d incoming and d outgoing links; b) maximum hop count of an incomplete Kautz $(d, k+1)$ is $3k/2$.

Serialization. Our algorithm requires new node additions to be serialized across the network. In other words, nodes are added sequentially, not in parallel. This is to avoid corner cases in the network topology, where an older node n_0 with k digits might be disconnected from one of its newly neighbors with $k+1$ digits, yet remain the destination of another node. By serializing node additions, we guarantee a consistent view of whether n_0 exists or has been renamed. This constraint is reasonable in data center, since it involves the manual addition of a rack by an administrator/operator.

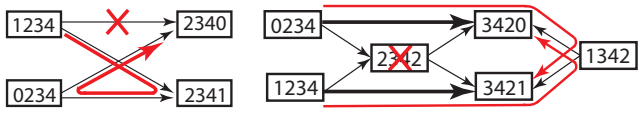
Finally, the ID assignment for incomplete Kautz graphs is very similar to that of complete graphs. We leave the details for brevity.

4.3 Fault Detection and Recovery

Availability and robustness to faults are key properties of a facilities network. The network should be available despite individual component failures (*i.e.* radios, rack switches), external interruptions (*i.e.* link disruption or blockage), and even when significant portions of the data plane are down for maintenance or upgrades. Here we describe fault-detection and recovery algorithms to deal with link, node and correlated failures.

We define three types of faults as follows:

- *Link Failures*: a 60GHz 3D beamforming link can fail due to radio hardware failure, wireless interference, radio misalignment, or external blockage.
- *Node Failures*: a node can be shut down by sudden rack power loss, or rack switch failure.
- *Correlated Failures*: a cluster of spatially correlated nodes, *e.g.* an entire row of racks, can fail concurrently due to planned maintenance or network upgrades.



(a) Link failure (b) Node failure

Figure 5: Fault-tolerant routing with 4 radios per rack. (a) A failed link can be recovered by a detour with 2 extra hops. (b) A rack failure requires antenna rotation to build new link (bold lines), and can be recovered by either 1 hop or a 3-hop detour.

Fault Detection. Fault recovery requires timely fault detection. It is particularly important for a transmitter to identify the true cause of the error, *i.e.* whether it is a node failure or a link failure, since fault recovery mechanisms for each are quite different. This detection is also non-trivial, since link failures and node failures result in the same outcome from the transmitter's point of view: losses of all link layer beacons and ACKs. For link failures, we focus on "long-term" wireless transmission failure, and rely on 802.11's DCF mechanism to address temporary failures due to interference.

We use explicit feedback to distinguish between different failures. If a receiver rack n (in link $m \Rightarrow n$) detects an abnormality, *e.g.* radio interface is down or no beacons received for some period, n sends a feedback packet to the transmitter m using standard Kautz routing. Kautz routing guarantees that the paths $m \Rightarrow n$ and $n \Rightarrow m$ are always disjoint, and the feedback packet will avoid the problematic radio. If m is up, receiving a feedback packet means its link to n has failed, and it triggers the link failure handling mechanism. Otherwise, if m detects a down interface and no feedback packets are received after some suitable period, then the transmitter can assume that the rack is down, and it needs to trigger a node-failure recovery mechanism.

Handling Link Failures. If a single link $m \Rightarrow n$ fails, we recover by forwarding traffic to n around the faulty link. Although Kautz graphs have good redundancy, the deterministic digit-shifting routing provides no redundancy. Thus we route forward using another outgoing link from m to another neighbor n'. By definition, n' differs from n in only the last digit. Therefore there must a forward link to n' from m', where m' is some node that differs from m in only the last digit. We can route in reverse on this link (from n' back to m'), then forward from m' to n, effectively using a 3 hop detour to circumvent the failure. We show a simple example of this using a base-4 overlay in Figure 5(a), where the red detour ($1234 \rightarrow 2341 \rightarrow 0234 \rightarrow 2340$) replaces a single failed link $1234 \rightarrow 2340$.

Handling Node Failures. A ToR switch failure or rack power failure represents a node failure in the Kautz graph. When a node n fails, all of its incoming and outgoing radio links also fail. Recovery requires repairing all links that route through the failed node, and involves retuning radios pointing to the failed node to new receivers.

For a Kautz network with degree d, we must ensure that each of the d links routing through the failed node n_f can reach their destinations. The high level intuition is that we pair up n_f's d incoming link radios with the destination radios of its d outgoing links. These radios pair up to form new links that bypass the failed n_f. We show an example in Figure 5(b), where node 2342 failed, and two of its incoming links from 0234 and 1234 are redirected to two of its outgoing neighbors. Since the new links cannot reach all of 2342's outgoing neighbors, it reaches the missing neighbors via one hop redirection, *e.g.* $0234 \rightarrow 3420 \rightarrow 1342 \rightarrow 3421$. While we only showed half of 2342's incoming links, the other links extrapolate in the same manner. Although we only show redirection from each incoming link to one outgoing neighbor, the link also reroutes to the two unpictured outgoing neighbors the same way, *i.e.* 0234 reroutes to 3422 and 3423. This allows us to maintain full connectivity.

When a node n_f fails, it creates a "hole" in the routing mesh that flows route around. Any flow not targeting n_f as its destination maintains its connectivity. However, routing around the hole introduces overhead. Therefore, we assume that if and when new nodes are added, they should first be assigned nodeIDs that allow them to fill existing holes in the network. Doing so restores the links before the failure, and eliminates the redirection required for fault recovery.

Handling Correlated Node Failures. Our hybrid assignment provides good robustness towards correlated node failures, by spatially spreading out nodes with closeby IDs. That is, nodes serving as detours in case one of them fails are guaranteed to be well-separated in the data center layout. This provides hard guarantee that detour paths remain available and maintain network connectivity after up to 55% correlated node failures (see results in §6).

Complexity. Our proposed fault handling mechanisms are low in computation complexity. Handling node failures require retuning antenna orientation, which introduces a small overhead when using horn antennas. As antenna arrays are becoming more sophisticated and available, we can remove this overhead by replacing horn antennas with antenna arrays that use instantaneous electronic beam switching. Overall, since the proposed mechanisms are of low complexity, Angora is easy to troubleshoot and repair.

Adding Redundancy. In our current design of Angora, the fault handling algorithms already provide very high reliability (see results in §6). To further enhance reliability, the data center administrators can add redundant links in Angora to improve its fault tolerance just like those proposed for wired networks, at the cost of increased infrastructure spending.

5. TESTBED EXPERIMENTS

Using off-the-shelf 60GHz radios, we implemented a "proof-of-concept" prototype of Angora. We use it to evaluate the suitability of 60GHz and our Angora design. We also validate 60GHz propagation/interference models, which we use to drive network simulations in §6. For all experiments, we used a metal reflector at a ceiling height of 4m.

Our experiments used two different 60GHz radios:

WiloCity radios. Our primary testbed consists of six pairs of Dell 6430u laptops and D5000 docks (Figure 6). Each has a WiloCity 60GHz radio chipset with a 2x8 antenna array, operating according to the IEEE 802.11ad standard for 60GHz. We found the low-cost 2x8 array creates a wide beam (nearly 40° in width), and compensated this by attaching a metal box to each device, emulating a horn antenna of 10° beamwidth. Our experiments confirm that this modification does not affect link throughput/latency.

HXI radios. Our second testbed includes a single pair of HXI Gigalink 6451 radios, the same hardware as two prior works [20, 44]. Each radio operates on a proprietary (non-802.11ad) configuration, has a horn antenna of 10° 3dB beamwidth and runs on a fixed 1.25Gbps rate.

5.1 Is 60GHz Suitable for Facilities Networks?

We set up 3D beamforming links to mimic data center transmissions in the presence of human movement, temperature variations and structural vibrations. By examining 60GHz link-level performance, we confirm its suitability as an alternative to wired links in a facilities network.

Range. Our measurements show that the HXI radio has a range of 42m at 0dBm transmit power and 144m at 10dBm power, sufficient for today's medium data centers (40mx50m, 320 racks, 12800 servers). The WiloCity radio has a shorter range of 22m, because its 2x8 antenna array's gain is at least 12 dB[7] lower than the horn antenna. It can support today's small data centers (20mx20m, 80 racks, 3200 servers).

Throughput & Latency. The WiloCity radio uses 802.11ad rate adaptation and its link rate (reported by the driver) decreases gracefully with the link distance from 3.85Gbps (at <1m) to 770Mbps (at 22m). The iperf TCP throughput is capped to 1Gbps, due to the laptop's 1Gbps Ethernet interface. The HXI link achieves a fixed 800Mbps TCP rate. For both radios, ping latency is less than 1ms.

Link Stability. We repeat iperf experiments once per second, and record link TCP throughput continuously for 3 weeks. Figure 7 shows the CDF of per-second throughput for both radios, indicating that both radio links are stable over time. This confirms the feasibility of using 60GHz links to build reliable, high-performance connections.

Interference vs. Angular Separation. We build two directional links using WiloCity hardware, and examine their link throughputs as we vary the 3D angular separation between them. Experimentation with multiple link configurations seen in medium-sized data centers [44] all led to the same conclusion. Figure 8 shows the normalized throughput degradation for two configurations, where the links are 8.4m or 12.5m long. 3D angular separation of <6° produces 30-70% throughput loss, which disappears completely once the angular separation reaches 12°. Finally, we obtained a similar result from HXI radios where 10° separation is sufficient.

5.2 Angora Microbenchmarks

Using the WiloCity testbed, we build Angora overlay paths to study path-level TCP performance. We focus on impact of path length and self-interference on each path, and interference between multiple paths. For evaluation, we use both iperf and a TCP file transfer program to emulate management tasks in data centers, *e.g.* controller pushing a 10KB update to a rack switch, or pulling a 1.3MB status update from a rack. To build multi-hop paths, we "bridge" the wired NIC and the 60GHz NIC on Dell laptops. Such "software" switching is CPU-bound and reduces TCP rate from the baseline 1Gbps down to 660Mbps.

Single-Path Performance. Most Angora paths are multi-hop, so a common concern is self-interference across hops. With 3D beamforming, this only occurs when receivers of two hops are closeby (on the same or neighboring racks), and their 3D angular separation is small. Here each link is bi-directional – reverse link carries TCP and MAC acks.

[7] In theory, such 12dB loss translates into 4x range reduction.

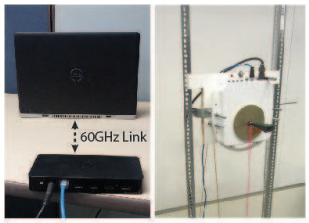


Figure 6: The two 60GHz radios.

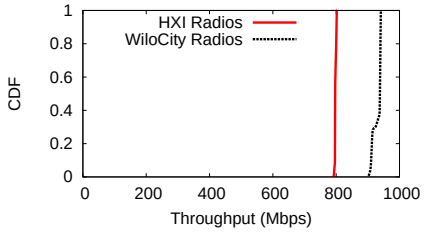


Figure 7: CDF of per-second TCP throughput of 60GHz links over 1 month.

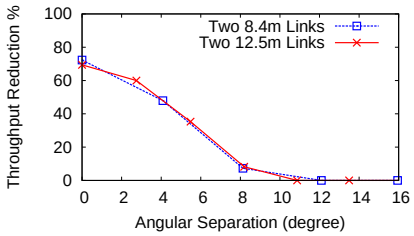
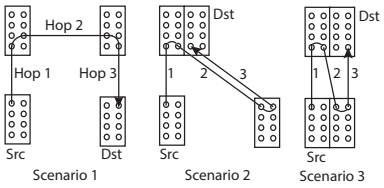


Figure 8: Link throughput degradation vs. the angular separation between two links.



(a) 3-hop overlay paths with different interference scenarios

Scenario	# of channels	TCP Thpt	10KB Msg. Latency	1.3MB Msg. Latency
1	1	654Mbps	3.1ms	30.8ms
2	1	173Mbps	4.8ms	84.9ms
	2	663Mbps	3.2ms	30.7ms
3	1	118Mbps	9.7ms	168.7ms
	2	413Mbps	4.6ms	48.3ms

(b) End-to-end results of 3 scenarios in (a)

Path Length	TCP Thpt	10KB Latency	1.3MB Latency
1 hop w/o bridging	940Mbps	1.3ms	19.0ms
1 hop	665Mbps	1.7ms	25.4ms
2 hops	662Mbps	2.5ms	30.5ms
3 hops	654Mbps	3.1ms	31.0ms
4 hops	665Mbps	3.5ms	35.9ms

(c) End-to-end results vs. path length

Figure 9: Angora single path performance. (a)(b) 3 single-path scenarios and their end-to-end performance. Self-interference exists but can be effectively suppressed via channel allocation. (c) Single path message delay scales gracefully with path length.

Figure 9 shows three 3-hop paths (no hops interfere, hop 2 and 3 interfere, hop 1, 2, 3 all interfere) and their end-to-end results from our testbed measurements. Specifically, when all hops use a single channel, path #2's self-interference reduces its throughput by 76% and increases message latency by 54%-175%. But with two channels, path #2 becomes interference-free, and path #3's loss reduces to only 30%.

Clearly the impact of self-interference is evident but can be effectively suppressed via channel allocation. This is only an issue when the path length exceeds the channel count (3 in 802.11ad) and when the channel assignment does not "spread" the channels evenly across hops. However, this happens rarely. For a medium-sized data center (320 racks), simulations show that when using random channel allocation only 0.72% of all paths experience self-interference using Kautz graphs (2.2% using a Random topology).

Next, we measure for each single path the impact of path length on end-to-end performance. Our experiments do not consider self-interference, since they appear in less than 1% of all paths. Figure 9(c) lists the throughput and average latency for 10KB or 1.3MB messages, and for reference the results for 1 hop paths without software bridging. For all these cases, the standard deviation of message latency is less than 10% of the average. These results show that the 660Mbps throughput cap from the software bridge also increases per-hop latency. Even so, Angora paths have small message latency: 3.5ms for 10KB messages, 35ms for 1.3MB messages. Message latency scales gracefully with path length.

Cross-Path Interference. Next we examine how interference affects concurrent paths. While Angora (Kautz+hybrid ID) nullifies interference among links on the same rack, a small number of disjoint but closeby paths can still interfere when running concurrently. Figure 10 shows a representative 2-path example extracted from Kautz and Random topologies assuming all links use the same channel. We implemented and evaluated these paths using our testbed. For Kautz, each path obtains 259Mbps TCP throughput (3.9ms latency for 10KB messages, 48.9ms for 1.3MB), while for Random, it reduces to 129Mbps (5.3ms latency for 10KB, 81.6ms for 1.3MB). This shows that cross-path interference does exist under heavy traffic and can lead to latency tails. However, its impact can be effectively reduced using channel allocation (based on our large-scale results in §6).

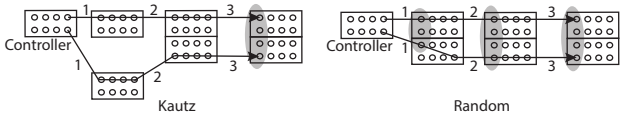


Figure 10: Examples on cross-path interference. Kautz guarantees that there is no cross-path interference within 2 hops from the controller. Random may have cross-path interference at any hop.

Note that Kautz graphs experience much less cross-path interference (thus much smaller latency tails) than Random topologies. This is because hybrid ID assignment ensures that links affected by cross-path interference are at least 2 hops away from any controller[8], putting a hard limit on the interference. For Random, interference can occur at any hop around a controller (see Figure 10), and affect other paths/flows sharing these links. The result is longer latency tails, later also seen from our large-scale simulations in §6.

6. LARGE-SCALE EVALUATION

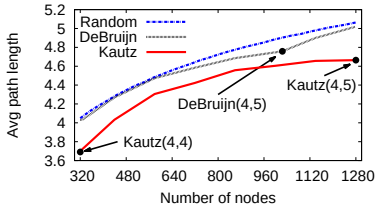
We perform detailed NS3 simulations to evaluate Angora at larger scales, focusing on its ability to deliver control messages with bounded latency and robustness against failures. We seek to understand how the topology choice (*i.e.* Kautz vs. Random) and hardware choice (*i.e.* horn antennas vs. antenna arrays) affect its performance.

Simulation Setup. We implement Angora in NS3, adapting the 60GHz flyways code [20] to include 3D beamforming radios (half-duplex), 802.11ad MAC (backoff and ACKs), overlay routing, and TCP. TCP ACKs are sent via the overlay path back to the source. We use existing data center designs [44]: each rack hosts 8 radios,

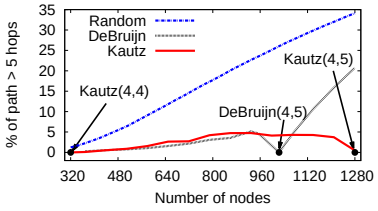
[8]As shown in Figure 10, with Kautz/hybrid ID assignment, the first hop destinations from each controller are well-separated, preventing cross-path interference within 2 hops from the controller.

Path Length	Overlay w/ 320 racks			Overlay w/ 1280 racks		
	Kautz	DeBruijn	Random	Kautz	DeBruijn	Random
< 4	26.0%	16.2%	23.5%	6.6%	4.2%	6.4%
4	74.0%	63.8%	42.8%	19.6%	11.6%	17.1%
5	0%	19.9%	32.3%	73.8%	63.4%	42.4%
6	0%	0.1%	1.4%	0%	20.6%	32.8%
>6	0%	0%	0%	0%	0.2%	1.3%

(a) Network sizes 320 and 1280 (complete Kautz graphs)



(b) Arbitrary network sizes



(c) Arbitrary network sizes

Figure 11: Angora's path length. (a) The path length distribution for data centers with 320 and 1280 racks. (b-c) Average path length and percentage of path lengths exceeding 5 hops for data centers of various sizes, representing *incomplete* Kautz graphs. The structured overlays (Kautz and DeBruijn) outperform the unstructured overlay (Random).

connected via a standard switch (6.4MB queue). Radio processing and switching delays are 10ns [6]) and 500ns [9]. We use a reflector at 4m height.

We configure the NS3 physical layer with the free-space propagation model, validated by our testbed experiments and prior work [20, 44]. Our simulations consider both horn antennas and antenna arrays By default, each radio is equipped with a standard horn antenna with 10° 3dB beamwidth. We derive detailed radiation patterns of the horn antennas according to the Kelleher's universal horn pattern [27]. We also verify that the Kelleher's model matches well with the horn antennas (10° 3dB beamwidth) used in our experiments, as well as that used in [20]. For antenna arrays, we calculate their radiation patterns following the standard rectangular array definitions from [21]. We set the maximum transmit power (10dBm) and rate table based on the 802.11ad specification [4]. Since the 802.11ad standard defines 3 orthogonal channels (2.16GHz each), we apply a simple greedy algorithm to preselect a channel for each link, prioritizing links directly connected to rack(s) housing controllers. We leave optimization of channel assignment to future work.

We consider data center layouts used by prior works [20, 44]: racks are grouped into 5×5 clusters; each cluster is a row of 10 racks with no inter-spacing; aisles separating the clusters are 3m (between columns) and 2.4m (between rows). We test data centers of size 320 racks to 1280 racks.

6.1 Path Length

We consider three overlay topologies: Kautz, de Bruijn and Random. For Random, we run 10000 rounds to obtain statistically significant results. Figure 11(a) lists the path length distribution for data centers of 320 and 1280 racks. For Kautz, they both lead to a complete graph, *i.e. Kautz* (4,4) and *Kautz* (4,5), respectively. As expected, the Kautz topology provides strict bounds on path length (4 and 5 respectively). In contrast, Random topologies have a longer tail: in 33% of cases, it leads to 1 more hop, and in 1.3% of the cases it leads to 2 more hops. This is consistent with prior work on randomized networks [36].

We also consider data centers of size between 320 and 1280, representing incomplete Kautz graphs and de Bruijn graphs. Figure 11(b)-(c) plot the average path length and the tail (the percentage of paths with more than 5 hops). In both cases, Kautz outperforms: its average path length grows gracefully with network size, and the ratio of long paths (>5 hops) is within 5%. These results validate our choice of Kautz graphs as the network topology, and the efficacy of our algorithms to support incomplete graphs.

6.2 Path Concurrency

To evaluate Angora's ability to support parallel flows, we randomly select M racks to communicate with one or multiple controllers. For each M, we run 2000 rounds with random rack and controller locations, and compute path concurrency as the portion of all paths able to run concurrently[9]. We experimented with three different traffic patterns: multiple racks to controller(s), controller(s) to multiple racks, and a random mix of the first two. Since they lead to similar conclusions, we only show the random mix results, which consistently have the lowest path concurrency of the three.

# of Flows	Overlay w/ 320 racks			Overlay w/ 480 racks		
	Kautz (hybrid ID)	Kautz (random ID)	Random	Kautz (hybrid ID)	Kautz (random ID)	Random
40	93%	90%	82%	88%	83%	77%
80	93%	89%	78%	84%	81%	75%
160	93%	88%	76%	81%	78%	71%

Table 1: The bottom 2% path concurrency, a single controller.

Topology choice	320 racks, 80 flows w/ varying # of controllers				
	1	2	4	6	8
Kautz (hybrid ID)	93%	88%	88%	85%	81%
Kautz (random ID)	89%	80%	80%	79%	79%
Random	78%	76%	68%	65%	60%

Table 2: The bottom 2% path concurrency, 1-8 controllers.

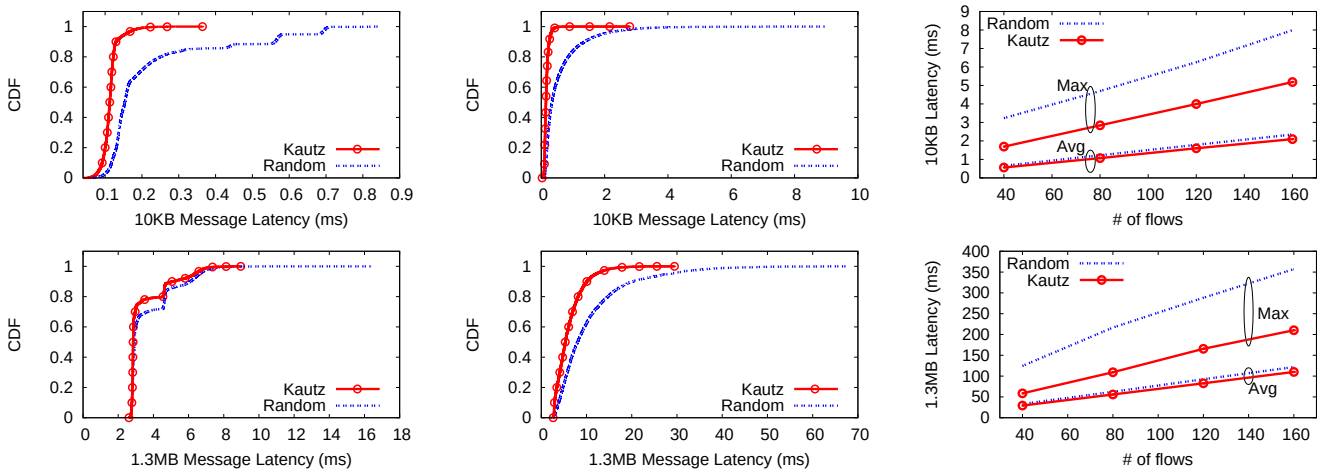
Table 1 lists the bottom 2%-percentile path concurrency (across 2000 rounds) when a single controller is present, for data centers of size 320 racks (a complete Kautz graph) and 480 racks (an incomplete Kautz graph). Our key observations are as follows. When using horn antennas, both Kautz and Random graphs maintain high path concurrency ($>70\%$) for the two data center sizes, even when 160 flows compete. Kautz shows a sizable advantage over Random, which can be attributed to two factors: reduced path length (thus less interference) and hybrid ID assignment that effectively scatters the directional links. Our hypothesis is confirmed by results of "Kautz with random IDs" in Table 1.

We obtain the same conclusion from results with multiple controllers (Table 2). Because distributing flows across multiple controllers creates more traffic hotspots and active wireless links, path concurrency decreases with more controllers. While the impact is significant for Random topologies (60% for 8 controllers), Kautz gracefully degrades from 93% for 1 controller to 81% for 8 controllers, again benefitting from a more controlled network structure.

6.3 Path Latency

Next we study a more practical question: *Can Angora provide reliable communication with bounded latency, even when multiple racks communicate with controller(s) in parallel?* For this we examine end-to-end latency of TCP traffic in a medium-sized data

[9]Multiple paths run *concurrently* if each link's SINR can support its target rate. The target rate is the aggregated flow rate of all the flows that share the link in absence of any interference.



(a) 10KB and 1.3MB single flows (b) 10KB and 1.3MB bursty flows (c) 10KB and 1.3MB synchronized flows

Figure 12: Angora's end-to-end TCP delay performance when carrying 10KB (top) and 1.3MB (bottom) messages. (a) CDF of single flow latency. (b) CDF of per-message latency of the two bursty flows. (c) The maximum and average per-message latency of multiple synchronized flows.

center (320 racks). In the absence of real control traffic traces, we consider three traffic patterns: single-flow, bursty multi-flow, and synchronized multi-flow.

Single-flow. At any given time, only a single rack communicates with the controller. We consider messages of size 10KB (*e.g.* SDN flow setup), 1.3MB (*e.g.* flow table and statistics [7, 12]) and 10MB (*e.g.* VM images).

Figure 12(a) shows the statistical distribution of per-message latency across all combination of rack and controller locations. We omit the 10MB figure due to space limit. We note that end-to-end latency is very low. For Kautz, maximum delay is bounded by 0.45ms for 10KB, 9ms for 1.3MB and 65ms for 10MB messages[10]. 90% of its flows experience delay less than 0.2ms, 5ms and 52ms respectively. We note there is a delay tail, because a small number of flows still experiences self-interference, which triggers MAC backoffs and occasional retransmissions. Kautz's more uniform results are due to a combination of shorter path lengths and stronger interference suppression.

Bursty Multi-flow. We consider two bursty scenarios where all racks send messages to a single controller over a period of 10 seconds. The first assumes each rack sends 10KB messages with an exponential distributed inter-arrival time of mean 15ms. The second increases message size to 1.3MB but reduces the mean of inter-arrival time to 500ms. We repeat each experiment for 100 rounds and randomize controller locations. Figure 12(b) shows that per-message latency is small for both Kautz and Random topologies, $<$9ms for 10KB messages and $<$70ms for 1.3MB. As before, the key difference between the two topologies is reduced variability (shorter tail, 3ms/30ms) for Kautz.

Synchronized Multi-flow. Multiple racks send or receive a single control message from controllers, and all flows start at the *same* time, creating heavy competition. Figure 12(c) shows latency of 10KB and 1.3MB messages with 40 to 160 flows. Even with 160 parallel flows, both Kautz and Random are able to offer good average latency results. Kautz again outperforms Random, with maximum delays as low as 50% of Random.

[10]The latency is lower than the WiloCity testbed result (§5.2) because we remove the bridging artifact and use a horn antenna.

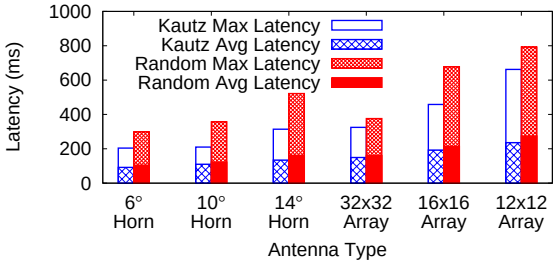
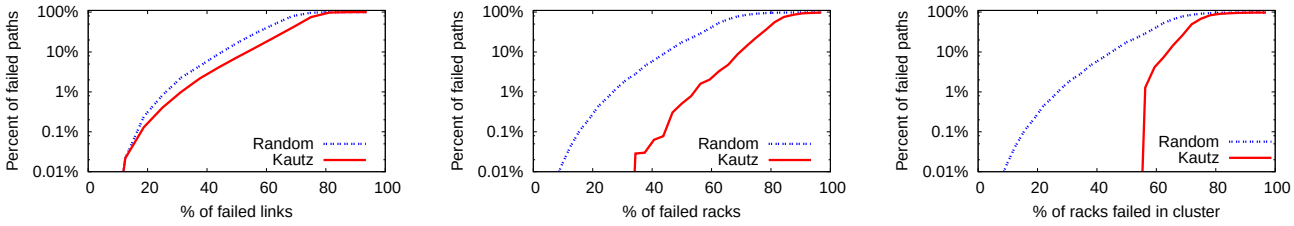


Figure 13: Impact of antenna choice: maximal and average per-message latency, 160 synchronized flows, 1.3MB messages.

Impact of Radio Hardware. To understand the impact of radio choice, we compare latency performance when using six different 60GHz antennas: horn antennas with 6°, 10° and 14° beamwidth which are typical commercial configurations, and antenna arrays of 32x32, 16x16 and 12x12 in size which have been prototyped [8, 30]. We repeat the above latency experiments and results of different traffic types lead to similar conclusions. We only show the result of 160 synchronized flows with 1.3MB messages, which represent the heaviest traffic load and flow competition.

Figure 13 shows the maximal and average per-message latency for Kautz and Random topologies. We make two key observations. *First*, antenna arrays lead to higher latency than horn antennas even though their main beam is narrower, *i.e.* 3.2°, 6.3°, and 8.5° respectively. This is because their side-beam emissions create extra interference that is harder to "suppress" via topology design. One can reduce side-beams by increasing antenna elements, *i.e.* from 12x12 to 32x32, at a higher cost. Another potential solution is to use interference nulling [29] to proactively cancel observed interference, which is an interesting open research direction.

Second, across all antenna configurations, Kautz consistently outperforms Random. As a result, a facilities network with Kautz/hybrid ID can meet the same latency requirements using cheaper hardware. For example, to pull 1.3MB route table within 500ms [12, 33], 16x16 arrays or 14° horn antennas should suffice for Kautz graphs, while Random requires 32x32 arrays or 10° horn antennas.



(a) Link failure (b) Node failure (c) Correlated node failure

Figure 14: Percentage of failed paths in Angora under different types of failures, using both Kautz and Random overlays.

Failure Percentage	5%	10%	15%	20%	25%
Rack Failure	2.0%	3.2%	4.2%	5.2%	6.3%
Link Failure	5.7%	11.2%	17.1%	23.5%	30.0%

Table 3: Average path length increases slightly during failures.

6.4 Fault Tolerance

We now evaluate Angora's robustness against failures. We create random failures by randomly disabling links and racks, and correlated failures by randomly removing consecutive rows of racks. We repeat each experiment with 1000 rounds, and examine how failures affect path connectivity. For Kautz, we apply recovery schemes described in §4.3. For Random, the fault recovery is done by re-identifying the shortest path in the current directed graph via global search. Figure 14 compares the percentage of failed paths under different failures. We see Angora is highly robust against all three types of failures. Using the Kautz overlay maintains 99% path availability for individual or correlated rack failure rates up to 50%, and for link failure rates up to 30%.

We also make two key observations. *First*, comparing the two graph structures, Kautz provides higher availability, especially for node failures. In particular, Kautz maintains perfect availability under correlated failures even when half of all racks are unavailable. This is because Kautz's hybrid ID assignment "spreads" out overlay links widely across racks, thus a path is unlikely to be colocated with its backup. *Second*, while Random leads to similar performance across all three types of failures, Kautz is more robust against node failures than link failures. The reason is that Kautz realigns radios to handle node failures, but not for link failures. We do not propose using radio realignment to recover from link failures, because searching for available radios to realign is a nontrivial task that may introduce considerable latency.

Latency Impact. Finally, we find that the latency cost of recovering from single link and rack faults is bounded (at most the latency of a 3-hop detour). Simulations show that average path length increases gracefully as the percentage of failure increases, as shown in Table 3. For example, The average path length only increases by 8% when 25% of all nodes fail, and 30% when 25% of links fail. Latency of detoured paths grow linearly with their path length, while existing flows in the neighborhood experience little impact.

7. RELATED WORK

Data Center Networking. Existing works focus on improving data plane performance, either by scheduling flows more efficiently (*e.g.* [7]) or by proposing new architectures (*e.g.* [9, 17, 36, 43]). In contrast, our work is to build a facilities network, a second network that differs significantly from the data plane in both characteristics and requirements.

60GHz in Data Centers. Recent works have utilized 60GHz links to augment data plane's bandwidth [20, 24, 25, 34, 41, 44]. Their key objective is to improve data plane bandwidth and address traffic hotspots. The 60GHz links are set up on-demand, thus requiring link coordination, antenna rotation, and/or centralized scheduling that introduce additional delay. Such extra delay makes these designs unsuitable for our targeted facilities network, which is highly delay sensitive but requires substantially lower bandwidth than the data plane. In contrast, the focus of Angora is to deliver management traffic via the facilities network with small bounded delay and high robustness. To this end, our solution, while supporting any-to-any rack communication, removes the need for link coordination/scheduling and antenna rotation that lead to considerable complexity and latency (up to 1s) in existing designs [44].

Another 60GHz in data centers proposal is to completely replace wired networks [35, 39] by 60GHz links. Unlike [35, 39] that require specialized server design and rack hardware, our design supports today's standard data center equipment.

SDN Control Plane. Active research has focused on SDN control plane designs, from operating systems to scalable designs [12, 28, 37, 42]. The reliable delivery of control traffic, however, has been often taken granted. Yet recent measurements show that control traffic delivery significantly affects network performance [12]. Our facilities network fills in this gap by delivering SDN control traffic and a wide variety of management traffic reliably in real-time.

8. CONCLUSION

We consider the problem of building an orthogonal facilities network as a core tool for managing data center networks. Our solution is Angora, a Kautz network built on 60GHz 3D beamforming links. Angora uses a small number of radios per rack to connect any pair of racks with a robust, latency-bounded path. We address multiple challenges including wireless interference, robustness to route and radio failures, and evaluate Angora using both experimental measurements and simulations. We believe that Angora is the first step towards the development of a robust and practical data center facilities network.

9. ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their feedback. This work is supported in part by by NSF grant CNS-1317153 and research gifts from Google and Cisco.

10. REFERENCES

[1] 802.11n: The end of ethernet? `http://www.arubanetworks.com/pdf/technology/whitepapers/wp_Burton_End_of_Ethernet.pdf`.

[2] Creating the data center of tomorrow. `http://csmedia.corning.com/CableSystems/%5CResource_Documents%5Carticles_rl%5CLAN-1225-EN.pdf`.

[3] Google data center images. `http://www.google.com/about/datacenters/gallery/#/all`.

[4] IEEE P802.11ad/D0.1: Enhancements for very high throughput in the 60 GHz band. TGad D0.1, May 2010.

[5] IEEE802.1Q - Virtual LANs. http://www.ieee802.org/1/pages/802.1Q.html.

[6] LightPointe Aire X-Stream 60GHz 1.25Gbps Wireless Bridges. http://www.lightpointe.com.

[7] AL-FARES, M., RADHAKRISHNAN, S., RAGHAVAN, B., HUANG, N., AND VAHDAT, A. Hedera: dynamic flow scheduling for data center networks. In *NSDI* (2010).

[8] AL HENAWY, M., AND SCHNEIDER, M. Planar antenna arrays at 60GHz realized on a new thermoplastic polymer substrate. In *EuCAP* (2010).

[9] ALIZADEH, M., KABBANI, A., EDSALL, T., PRABHAKAR, B., VAHDAT, A., AND YASUDA, M. Less is more: Trading a little bandwidth for ultra-low latency in the data center. In *NSDI* (2012).

[10] BENSON, T., ANAND, A., AKELLA, A., AND ZHANG, M. Microte: fine grained traffic engineering for data centers. In *CoNEXT* (2011).

[11] CASADO, M., FREEDMAN, M. J., PETTIT, J., LUO, J., MCKEOWN, N., AND SHENKER, S. Ethane: taking control of the enterprise. In *SIGCOMM* (2007).

[12] CURTIS, A. R., MOGUL, J. C., TOURRILHES, J., YALAGANDULA, P., SHARMA, P., AND BANERJEE, S. DevoFlow: Scaling flow management for high-performance networks. In *SIGCOMM* (2011).

[13] DE BRUIJN, N. G. A combinatorial problem. In *Koninkli- jke Nederlandse Akademie van Wetenschappen* (1946), vol. 49.

[14] DEAN, J., AND GHEMAWAT, S. MapReduce: simplified data processing on large clusters. *Commun. ACM* (Jan. 2008), 107–113.

[15] GREENBERG, A., HJALMTYSSON, G., MALTZ, D. A., MYERS, A., REXFORD, J., XIE, G., YAN, H., ZHAN, J., AND ZHANG, H. A clean slate 4D approach to network control and management. *CCR 35*, 5 (2005).

[16] GUMMADI, K., GUMMADI, R., GRIBBLE, S., RATNASAMY, S., SHENKER, S., AND STOICA, I. The impact of DHT routing geometry on resilience and proximity. In *Proc. of SIGCOMM* (2003).

[17] GUO, C., LU, G., LI, D., WU, H., ZHANG, X., SHI, Y., TIAN, C., ZHANG, Y., AND LU, S. BCube: a high performance, server-centric network architecture for modular data centers. In *SIGCOMM* (2009).

[18] GUO, D., LIU, Y., AND LI, X. BAKE: A balanced Kautz tree structure for peer-to-peer networks. In *INFOCOM* (2008).

[19] GUO, D., WU, J., CHEN, H., AND LUO, X. Moore: An extendable peer-to-peer network based on incomplete Kautz digraph with constant degree. In *INFOCOM* (2007).

[20] HALPERIN, D., KANDULA, S., PADHYE, J., BAHL, P., AND WETHERALL, D. Augmenting data center networks with multi-gigabit wireless links. In *SIGCOMM* (2011).

[21] HANSEN, R. C. *Phased array antennas*, 2nd ed. John Wiley & Sons, 2009.

[22] HILBERT, D. Ueber die stetige Abbildung einer Line auf ein Flächenstück. *Mathematische Annalen 38*, 3 (1891), 459–460.

[23] HOSS, R., AND LACY, E. *Fiber optics*. Prentice Hall, 1993.

[24] KATAYAMA, Y., TAKANO, K., KOHDA, Y., OHBA, N., AND NAKANO, D. Wireless data center networking with steered-beam mmwave links. In *WCNC* (2011).

[25] KATAYAMA, Y., YAMANE, T., KOHDA, Y., TAKANO, K., NAKANO, D., AND OHBA, N. Mimo link design strategy for wireless data center applications. In *WCNC* (2012).

[26] KAUTZ, W. Bounds on directed (d, k) graphs. *Theory of cellular logic networks and machines, AFCRL-68-0668 Final report* (1968).

[27] KELLEHER, K. *The Microwave Engineers' Handbook and Buyers' Guide*, 5th ed. New York: Horizon Press, 1964.

[28] KOPONEN, T., CASADO, M., GUDE, N., STRIBLING, J., POUTIEVSKI, L., ZHU, M., RAMANATHAN, R., IWATA, Y., INOUE, H., HAMA, T., AND SHENKER, S. Onix: a distributed control platform for large-scale production networks. In *OSDI* (2010).

[29] NIKOLAIDIS, G., ZHUSHI, A., JAMIESON, K., AND KARP, B. Cone of silence: adaptively nulling interferers in wireless networks. *SIGCOMM CCR* (2010).

[30] NISHI, S., AND TOKUDA, K. Development of mm-wave video transmission system-development of antenna. In *APMC* (2001).

[31] PEARSON, E. R. *Professional Fiber Optic Installation: The Essentials For Success*. Pearson Technologies Incorporated, 2011.

[32] PHEMIUS, K., AND THALES, M. B. Openflow: Why latency does matter. In *IFIP/IEEE IM* (2013).

[33] RAICIU, C., BARRE, S., PLUNTKE, C., GREENHALGH, A., WISCHIK, D., AND HANDLEY, M. Improving datacenter performance and robustness with multipath TCP. In *SIGCOMM* (2011).

[34] RANACHANDRAN, K., KOKKU, R., MAHINDRA, R., AND RANGARAJAN, S. 60GHz data-center networking: wireless => worryless? *NEC Technical Report* (2008).

[35] SHIN, J.-Y., SIRER, E. G., WEATHERSPOON, H., AND KIROVSKI, D. On the feasibility of completely wireless data centers. In *ANCS* (2012).

[36] SINGLA, A., HONG, C.-Y., POPA, L., AND GODFREY, P. B. Jellyfish: Networking data centers randomly. In *NSDI* (2012).

[37] TOOTOONCHIAN, A., AND GANJALI, Y. HyperFlow: a distributed control plane for OpenFlow. In *INM/WREN* (2010).

[38] VAHDAT, A., AI-FARES, M., FARRINGTON, N., MYSORE, R. N., PORTER, G., AND RADHAKRISHNAN, S. Scale-out networking in the data center. *Micro, IEEE 30*, 4 (2010), 29–41.

[39] VARDHAN, H., RYU, S.-R., BANERJEE, B., AND PRAKASH, R. 60ghz wireless links in data center networks. *Computer Networks 58* (January 2014), 192–205.

[40] WU, X., TURNER, D., CHEN, C.-C., MALTZ, D., YANG, X., YUAN, L., AND ZHANG, M. NetPilot: automating datacenter network failure mitigation. In *SIGCOMM* (2012).

[41] YAMANE, T., AND KATAYAMA, Y. An effective initialization of interference cancellation algorithms for distributed mimo systems in wireless datacenters. In *GLOBECOM* (2012).

[42] YU, M., REXFORD, J., FREEDMAN, M. J., AND WANG, J. Scalable flow-based networking with DIFANE. In *SIGCOMM* (2010).

[43] ZATS, D., DAS, T., MOHAN, P., BORTHAKUR, D., AND KATZ, R. DeTail: reducing the flow completion time tail in datacenter networks. In *SIGCOMM* (2012).

[44] ZHOU, X., ZHANG, Z., ZHU, Y., LI, Y., KUMAR, S., VAHDAT, A., ZHAO, B. Y., AND ZHENG, H. Mirror mirror on the ceiling: Flexible wireless links for data centers. In *SIGCOMM* (2012).

APPENDIX

The hybrid ID assignment maximizes angular separation between pairs of outgoing links and between pairs of incoming links on each rack.

PROOF. We provide a sketch of the proof due to space limitations. The proof consists of three steps. *First*, using geometry, we prove that for 3D beamforming, maximizing angular separation of a rack's outgoing (incoming) links is equivalent to maximizing the physical distance between their receivers (transmitters). For Kautz graphs, these are "sibling" racks whose IDs have the same middle-digits, but different first/last digit. Furthermore, it is easy to show that maximizing the physical distance between siblings is achieved by properly placing each group of d^2 racks sharing the same middle-digits into d^2 distinct regions.

Second, assuming the data center layout is a rectangle of size $L \times H$ ($L \leq H$), we prove that for $d = 4$, for any ID assignment, the minimum physical distance between any two siblings is upper-bounded by $L/2$. This is because each rack has $2d - 2$ (6 when $d = 4$) siblings that need to be separated. Consider the rack placed in the middle of the rectangle. There is not enough space to separate its incoming (outgoing) siblings by $L/2$. The same proof applies to $d = 3, 5, 6$ although the upper-bound may vary.

Finally, we prove that for $d = 4$, the hybrid ID assignment proposed in Section 4 achieves the optimal value of $L/2$ in terms of the minimum sibling separation. Similarly, for $d = 3, 5, 6$, we also found the corresponding hybrid ID assignments which achieve their corresponding upper-bounds. This concludes our proof. □

We now show that with the hybrid ID assignment, the angular separation between any two links $\geq 14^\circ$ in a typical rectangular data center layout.

Using the rack size and spacing described in [20, 44], a data center containing 320 racks (8 rows, 40 racks per row) is 30m×24m large. Assuming the ceiling is 4m high from the top of racks, it is easy to verify that the minimum angular separation in the whole data center is 14.6°. Furthermore, as long as the layout size and ceiling height scale proportionally, this minimum angular separation value will not change. For a fixed ceiling height, the larger the layout, the larger the angular separation. This means that the hybrid ID assignment scales well to larger data centers with a guarantee of $14^\circ+$ angular separation.