The Case for Decentralized Fallback Networks

Citation: Lynch, James, Liu, Ziqian, Li, Chenning, Ghobadi, Manya and Balakrishnan, Hari. 2024. "The Case for Decentralized Fallback Networks."

As Published: https://doi.org/10.1145/3696348.3696883

Publisher: ACM|The 23rd ACM Workshop on Hot Topics in Networks

Persistent URL: https://hdl.handle.net/1721.1/157895

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

The Case for Decentralized Fallback Networks

James Lynch, Ziqian Liu, Chenning Li, Manya Ghobadi, and Hari Balakrishnan
MIT Computer Science and Artificial Intelligence Lab

ABSTRACT

This paper argues that network and application delivery infrastructures have become highly centralized and are more vulnerable to attacks and disasters than is desirable. It proposes a research agenda for decentralized fallback networks and focuses on a key component—a city-scale decentralized network using existing Wi-Fi access points, which are deployed across almost all buildings in cities. It proposes a routing system that uses information about buildings from geospatial maps instead of traditional routing mechanisms to scale well to millions of Wi-Fi nodes.

CCS CONCEPTS

• **Networks → Mobile ad hoc networks**; **Peer-to-peer networks**; **Network design and planning algorithms**; **Network simulations**; **Network architectures**.

KEYWORDS

MANET, DFN, mesh networks, wireless networks

ACM Reference Format:
James Lynch, Ziqian Liu, Chenning Li, Manya Ghobadi, and Hari Balakrishnan, MIT Computer Science and Artificial Intelligence Lab. 2024. The Case for Decentralized Fallback Networks. In *The 23rd ACM Workshop on Hot Topics in Networks (HOTNETS '24), November 18–19, 2024, Irvine, CA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3696348.3696883

1 INTRODUCTION

The Internet has become highly centralized, both at the network infrastructure and application layers. Due to extensive Internet Service Provider (ISP) consolidation, a handful of companies control much of the wireline and cellular Internet infrastructure in any given country. Equally problematic is the centralization of the global application-delivery infrastructure, which relies heavily on cloud services.

In fact, the *physical* localization of network infrastructure, even when distributed across ISPs, is even more centralized. A few cities are critical meet-points for countrywide fiber connectivity, regardless of ISP [11]. Fiber conduits are shared between multiple ISPs and share the same fate [15]. Amplifier huts are vulnerable, in remote areas, and insecure [7]. And last but not least, Internet exchange points are clear targets at known locations [58].

Moreover, many applications today rely on centralized services delivered via a small number of cloud providers. Although the major cloud providers are geographically distributed, their datacenters are concentrated in a few locations based on access to inexpensive electricity [11]. In addition, recent research has shown that public cloud providers are vulnerable to cascading failures of the electricity grid [26], whose correct functioning itself relies on a robust network communications infrastructure (though not necessarily the public Internet).

As a result of this centralization and consolidation, we believe that both the network and application layers of the Internet may be more vulnerable to cyberattacks, natural disasters, bugs, and misconfigurations than before. To overcome these vulnerabilities, this paper proposes a program on *decentralized fallback networks (DFNs)*. During times of duress or attack, DFNs can be used to re-enable network applications, albeit at lower performance, but still capable of providing key functionality. The impact of this research on DFN will be to significantly improve the preparedness and resilience of network communications and applications.

We envision a comprehensive research agenda for DFNs comprising the following elements:

Network Infrastructure: Deploying entirely new infrastructure for DFNs is expensive, so leveraging existing networks to the extent possible is ideal. Is it possible to design large-scale DFNs using existing widely deployed Wi-Fi access points (APs) and mobile devices in urban areas to form city-scale Wi-Fi mesh networks? Specifically, we consider: (i) is there sufficient wireless node density, (ii) how do we scale routing to many millions of nodes, and (iii) how do we augment existing Wi-Fi infrastructure to bridge gaps in connectivity (e.g., rivers, parks, etc.)? Further, what role, if any, should technologies such as satellite networks serve as a component to fallback networks? Could these systems be used to connect between population centers in regions of a

---

HOTNETS '24, November 18–19, 2024, Irvine, CA, USA

ACM ISBN 979-8-4007-1272-2/24/11
https://doi.org/10.1145/3696348.3696883

country? We pose that DFNs are urban in scope; therefore, how do we form an inter-network of DFNs across regions?

Application delivery infrastructure: A useful operating principle is to guarantee application-level communication as long as there is a viable network path in the DFN. What features are required to enable existing applications to recover from lack of access to cloud servers and Internet services (e.g., DNS) during the fallback periods? Low-bandwidth applications such short peer-to-peer messaging, emergency broadcast messages, geospatial messaging, and payments should be enabled by these features.

Security: How do we ensure resilience to compromised nodes within the DFN? It is possible, even likely, that under cyberattacks some fraction of the nodes will be compromised. A successful routing protocol for a DFN should find a path between two nodes wishing to communicate if there exists a path that does not traverse a compromised node. Applications developed on a DFN must also be made secure against attack. At the application layer, we propose to use self-certifying names [42], deriving each identifier by hashing the entity's public key exchanged out-of-band. This approach ensures message and origin authenticity and confidentiality via encryption, both without the need for real-time access to centralized certificate authorities that may be unreachable.

With this backdrop, our goal in this paper is to focus on a narrower, but key, first step toward enabling DFNs. We describe the design and implementation of CityMesh, a city-scale DFN built using existing Wi-Fi APs. CityMesh focuses on one approach to addressing the fundamental network infrastructure challenges facing DFNs, and in doing so successfully demonstrates their near-term feasibility. We leave the other parts of the broader agenda described above to future work.

2 CITYMESH FEASIBILITY

CityMesh aims to enable a city-scale DFN that operates atop existing Wi-Fi infrastructure, perhaps with a modest addition of new APs, without using on any wireline or backhaul connections. CityMesh would enable communication between two nodes (e.g., smartphones) in a city. At a minimum, this approach requires that a sequence of AP-to-AP Wi-Fi hops exist between source and destination pairs. Our intuition is that there is already sufficient, overlapping residential and commercial Wi-Fi AP coverage in most cities to enable the creation of such a fallback network.

On the surface, it may seem that enabling intra-city communications isn't of much use because many of today's applications require significant Internet bandwidth and cloud access. But consider when a natural disaster knocks out network connectivity in a region. In disaster scenarios, the most common use of communication is to check on the safety of family and friends and update each other. Other uses are to look for emergency updates, find directions to a safer area, obtain access to essentials and to access a banking application for money. These are all low-bandwidth applications. We argue that it is possible to modify these applications to not rely on cloud servers.

Despite much prior work on mobile ad hoc networks (MANETs) and wireless mesh networks, scaling such networks to more than several thousand nodes remains an unsolved problem (see §5). One issue is that any routing protocol over wireless links that exchanges any form of keepalive or routing information is likely to run into scaling and reliability challenges. Thus we propose an approach where the nodes exchange no metadata about their existence, addresses, link state, etc.

We design CityMesh with a flat logical topology and reduce routing decisions to simply whether or not a node should rebroadcast a received CityMesh message. Rather than relying on geographic routing [27], which requires access to location information at reasonable fidelity and complex mechanisms to overcome dead ends, or the creation of bespoke coordinate systems among nodes exchanging local connectivity information [49], we exploit the layout of buildings in cities to route efficiently.

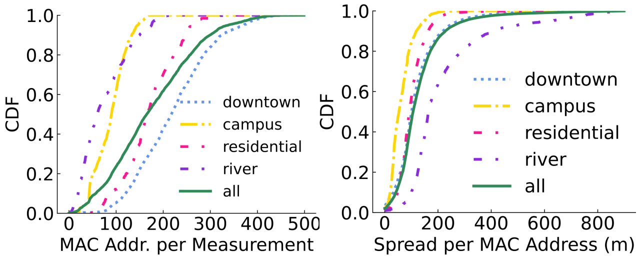
The core insight that enables CityMesh is that already deployed Wi-Fi APs are concentrated in residential and commercial buildings rather than areas outside building footprints. Additionally, we note that users, who are ultimately the source and destinations of messages, are most often inside or near buildings. This building-centric placement of devices participating in CityMesh allows a sending device to engage in source routing by predicting that APs in a city are well-deployed in buildings, without using any information from the network itself. Then, a device may select a *building route* that is likely to describe a region through which there are no gaps in AP coverage between the source and destination. Provided a path via buildings exists, it is likely that a path through the APs within those buildings also exists. Of course, such an AP path must comprise hops with a length smaller than the Wi-Fi communication range.

Exploiting city layout at scale is now possible with modern geospatial digital maps, resulting in the availability of detailed building footprint data through services such as OpenStreetMap [18], Google Maps, etc. In practice, today's devices can easily cache the data necessary for building routing in advance and continue to use this infrequently-updated data through the duration of an outage.

We note that during attacks or disasters, the supply of electricity might be unreliable, raising the question of how Wi-Fi APs might be powered. We acknowledge this issue, but note that when disasters occur, utilities are likely to restore power

Dataset	# Measurements	# Unique APs
downtown	2,691	26,532
campus	726	2,399
residential	461	10,333
river	550	4,794
all	4,428	40,158

Table 1: Summary of collected data for measurements.



(a) (b)

Figure 1: (a) CDF of the number of MAC addresses seen in each measurement. (b) CDF of the spread of locations where each MAC address was seen.

as soon as possible due to electricity's importance to healthcare, sanitation, and the supply of necessities. Moreover, we note that off-grid generators and battery backups are ubiquitous, particularly in regions where power outages are more frequent. Therefore, damaged telecommunications infrastructure is often likely to be the limiting factor to restoring network access during crises.

AP density. To support the claim that there is sufficient AP connectivity and distribution in cities, we sought to collect real-world data by conducting a measurement study in the Boston area.[1] We collected Wi-Fi AP beacon frames by walking or bicycling in four areas: downtown, in and around the MIT campus, a residential area, and along the banks of the Charles river. The first three areas are intended to show AP density, whereas the river measurements show inter-island connectivity. In each area, we use a 2.4 GHz wireless device (a Pineapple [17] or a TP-Link Wi-Fi router [59]) with a sampling frequency of 0.2–0.4 Hz to scan for APs. Each measurement contains a GPS location and a list of MAC addresses (BSSIDs) from the AP beacon frames. Table 1 summarizes our measurements in each area.

Figure 1a shows the cumulative distribution function (CDF) of the number of MAC addresses observed at each measurement. The median numbers in the worst case (river) and the best case (downtown) are 60 and 218, respectively. For each AP discovered, we calculate the *spread* of the locations where it was seen, which we define as the maximum distance between any two of the locations. The spread is an estimate of the diameter of the transmission region. Figure 1b shows the CDF of this metric. The median spread in the smallest case (campus) and the largest case (river) are 54 m and 168 m, respectively. The corresponding transmission radii are 27 m and 84 m, respectively. The substantial presence of Wi-Fi APs in most locations in this small-scale study suggests that cities may well have a sufficient infrastructure of APs for CityMesh to be practical.

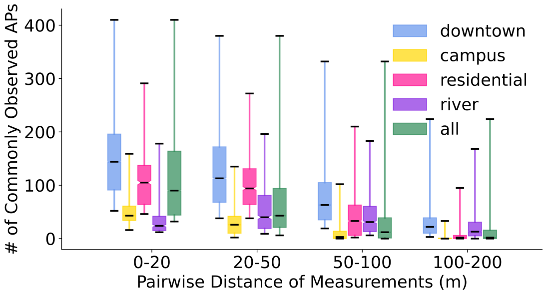


Figure 2: The distance between each pair of measurements and the number of APs observed by both measurement samples. The whiskers of each bar indicate the 10%, 25%, 50%, 75%, and 100% of values.

AP connectivity. To determine whether the APs can interconnect to form large connected meshes, we record the distance L for each *pair* of measurements and count the number of common APs observed at both locations. A larger L implies a larger transmission range of the AP r, which is greater than $L/2$. We bin the measurement pairs based on their distance L in our measurements. Figure 2 shows the distribution of the number of APs observed in common for each distance bin. Many APs observed in common are from locations 100 m apart, but we also observe a significant number of common APs beyond 100 m, particularly in the downtown area. This result indicates that a substantial number of APs are mutually visible and are likely to form a connected mesh when their distance is less than 100 m.

3 ROUTING VIA BUILDINGS

CityMesh presents itself to endpoints (smartphones and laptops) as a single Wi-Fi SSID, with which they can associate directly. The novel core concept in CityMesh is *building routing*, a form of source routing within the mesh network of APs using geospatial building maps of a city without any traditional distributed routing machinery.

Sending a message between users (say, Alice and Bob) involves 4 steps: postbox information exchange, route planning and encoding, transmission through the network, and message retrieval. Figure 3 shows this architecture.

---

[1] AP survey databases, like wigle.net [62], are sporadically collected via crowdsourcing and thus are non-uniform, and often lack precise locations.

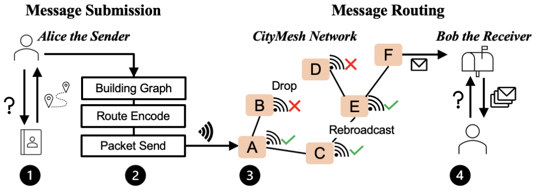


Figure 3: CityMesh's workflow: Alice, the sender, submits her messages to CityMesh's network, which routes the messages to Bob's postbox.

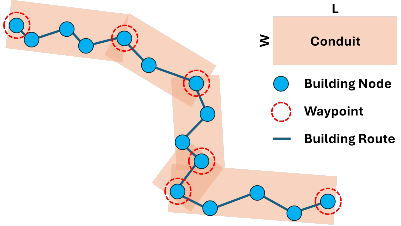


Figure 4: To compress the encoded building route in a CityMesh packet, our algorithm combines consecutive buildings that lie in approximately the same direction along the route into a single waypoint. We call the region defined between these waypoints (each segment of length L) and some parameter width W a *conduit*. APs that are within the conduit defined by some CityMesh packet's route rebroadcast that packet.

Step ❶. Acquiring the destination postbox. Before Alice can send a message to Bob through CityMesh, Bob must provide his postbox information to Alice. This exchange of information must occur out-of-band, potentially before a network outage event. As the address information is small, it could even fit within a QR code and be distributed physically. Bob's postbox information includes his unique public key and the building ID of the building that contains the desired postbox AP.

Step ❷. Generating the message. Given Bob's postbox information, Alice uses a form of source routing to compute the route for the message. This method relies on the stored city map information, which includes building footprint information extracted from a digital map such as OpenStreetMap (OSM) [18]. The source first generates a *building graph* from a map of the city, where each vertex represents a building, and edges represent AP connections between buildings that are likely to exist. To these edges, the routing method assigns weights based on the *cubed distance* (for reasons given below) between buildings and uses this map to compute a *building route* from Alice to Bob's postbox address. The densely deployed APs within buildings can route the message via multiple hops. Cubed-distance edge weights prioritize shorter edges for connectivity between buildings through their APs. The source adds this building route (a sequence of building IDs) to the header of the packet and compresses it to ensure that the packet header size is as small as possible.

This compression algorithm is central to CityMesh's routing. While reducing the building route length, it also reduces the precision of the specified route. Rather than specifying an explicit list of buildings that the packet should pass through, we instead specify a larger region it should follow. In doing so, we improve CityMesh's tolerance to mispredicted AP connectivity between buildings.

We construct this larger region out of a set of connected *conduits*: rectangles of length L and width W superimposed over the building route. Each conduit begins and ends on a *waypoint building*, which we select using the following algorithm: we place the starting edge of the first conduit on the centroid of the first building in the route. We then find the latest building in the route at which we can place the ending edge of the conduit and cover all buildings in the route that precede it. This building is a waypoint building. The building that satisfies this requirement is dependent on W, which we take to be a parameter of the algorithm (it should be comparable to the Wi-Fi transmission range, 50 m in our implementation). We start the next conduit at this waypoint and find its end, the following waypoint, using the same requirement as before. We repeat this process until we reach the last building in the route. The building IDs for the waypoint buildings are then encoded in the CityMesh packet. Figure 4 shows the behavior of this algorithm.

Step ❸. Routing through CityMesh. Once the route is encoded, Alice submits the message to CityMesh's network, which comprises APs running a small software agent. Individual APs use their copy of the building graph and the encoded route included the packet header to determine whether to rebroadcast the message. Only APs in buildings that fall within the geographic area of the conduits defined by the waypoint buildings in the route rebroadcast, while others ignore the packet. Each AP executes the following steps: upon receiving the packet, extract the waypoint buildings' unique IDs from the packet header and look up their geographic locations on the building map. Then, they reconstruct the conduit for each pair of consecutive waypoint buildings in the route using the predefined conduit width. Finally, the AP rebroadcasts the packets if and only if it is located within any of these connected conduits.

Step ❹. Retrieving the message. Upon reaching the destination, CityMesh caches the message in Bob's postbox (this means that APs must have the ability to store messages for a period of time). This postbox acts as a reliable intermediary for message storage and forwarding and also handles message integrity checks and decryption for Bob. Bob retrieves cached messages from the postbox periodically. The postbox may also implement push notifications for the immediate forwarding of urgent messages based on predefined user preferences. To enable these "push" notifications, Bob's postbox caches location updates from his device that it receives whenever his device checks for new messages.

We believe that this method can be implemented on commodity hardware routers with modest modifications. Most consumer-grade routers run the Linux kernel with a small-footprint userspace. This conventional software stack allows our proposed routing algorithm and message storage capabilities without hardware modification. Projects such as OpenWRT [45] provide community-supported replacement images for factory firmware for a wide range of Wi-Fi AP hardware devices.

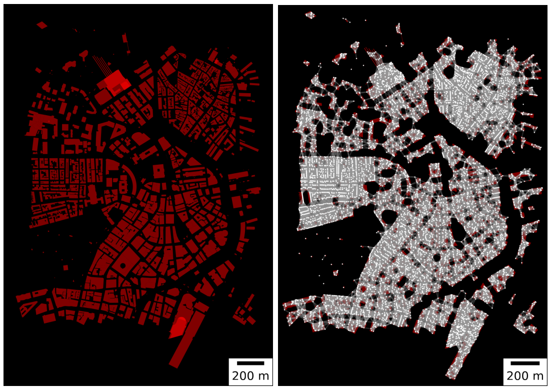
4 PRELIMINARY EVALUATION

Simulation design. To evaluate our proposed design, we implemented a simulator using Sympy [57] that takes real-world city building footprint data, assigns AP locations within buildings, and then attempts to route packets between pairs of APs across the city using the algorithm described in §3. The simulator does the following:

- Compiles building footprint data from OSM.
- Randomly places APs in a 2D plane, inside building footprints at a configurable AP density.
- Connects these APs into a graph where the inter-AP distance is below a configurable transmission range.
- Without using the AP graph, separately generates a CityMesh building graph from the footprint data that predicts inter-building connectivity given the simulation parameters of transmission range and AP density.
- Randomly selects a set of buildings as source-destination pairs.
- Simulates the proposed building routing algorithm to determine whether it is able to successfully deliver the packet from the source to the destination.

Performance. First, to complement the real-world measurements described in §2, we sought to verify that an idealized CityMesh network would have wide coverage of a city through our simulation.

Figure 5 shows a section of a downtown area as generated by this program using OSM. We randomly sampled 1000 unique pairs of buildings from the region and determined whether a path existed between the buildings via the AP graph. We repeated this process across multiple cities. We report results here for a transmission range of 50 m and (a relatively sparse) AP density of one AP per 200 m^2.



(a) (b)

Figure 5: Section of a downtown area. (a) shows building footprints in red. (b) shows the same region with APs randomly placed as white dots, and interconnected with gray lines where the distance between APs is less than 50 m. The AP density used to populate this graphic is 1 AP/200 m^2

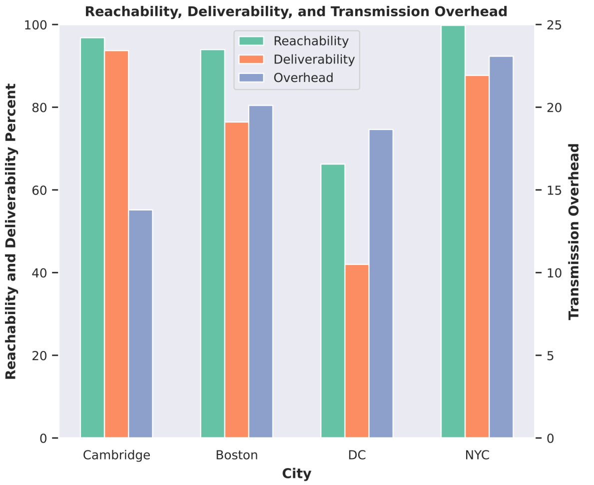


Figure 6: Reachability, deliverability, and transmission overhead between pairs of buildings or "routes." This simulated mesh uses a symmetric transmission range cutoff of 50 m and 1 AP/200 m^2 as the AP density. We test reachability against 1000 source/destination pairs. From the successful pairs, we test 50 source/destination pairs for deliverability using the full event-based simulation.

Figure 6 shows the fraction of building pairs that are "reachable" and given reachability, the "deliverability" for each city across a proposed CityMesh AP mesh. Reachability describes whether a source/destination building pair is connected through the AP graph. Deliverability assumes reachability, but additionally takes into account the performance of a particular routing algorithm that operates across the CityMesh network, describing whether the packet is successfully delivered to the destination. Additionally, we also calculate the transmission overhead for each simulation. We define this overhead as the ratio of the number of packet broadcasts generated by our algorithm in the simulation to the minimum number of transmissions necessary to reach from source to destination for the same realization of AP placement within the city. Note that the denominator here is the absolute best case as it does not account for link-layer retransmissions that will be necessary in any practical unicast routing protocol (tightening this estimate and optimizing the routing method are both important areas for future work).

Our evaluation suggests that given reasonable assumptions about AP distribution and connectivity within a city, the CityMesh algorithm can successfully deliver packets across many pairs of buildings for the tested cities, with most cities surveyed having high deliverability. In reviewing simulations in which packets do not reach the destination, we find that connectivity is occasionally interrupted by large features such as highways, parks, and bodies of water that can prevent communication under the assumptions of this simulation. These breaks in connectivity fracture some cities, like Washington D.C., into multiple islands of connectivity. Consistently, however, we find that large swaths of inhabited urban areas are densely connected across simulations. For the cases in which we have failed deliverability due to these connectivity gaps, we propose that the addition of a small number of well-placed APs would serve to bridge connectivity between these islands. Taller buildings with APs on higher floors would likely increase the transmission range and extend the connectivity of the network, a factor not reflected with the conservative transmission range assumptions made in these simulations.

Finally, the overheads are tolerable: in a typical city simulation, the median and 90%ile packet header for the compressed source route are 175 and 225 bits. The number of extra transmissions compared to the ideal unicast route is 13×, but that is because currently all the APs within a building rebroadcast, and there are other inefficiencies; we are confident that this overhead can be reduced with various improvements to the method.

5 RELATED WORK

Wi-Fi mesh networks. A Wi-Fi mesh consists of connected Wi-Fi devices that collaborate to route data to and from clients. Prior work includes RoofNet [3], MadMesh [5], WiFi-WiMAX-mesh [14], TFA urban network [8], MetroMesh [1], and DGP [9]. Wi-Fi mesh networks require that every mesh node track state and distribute control packets in the network for topology management, topological routing, path selection, and coordination. Scaling current mesh network protocols to a city scale with hundreds of thousands of Wi-Fi nodes using the protocols from these prior papers would result in excessive communication overhead between mesh nodes.

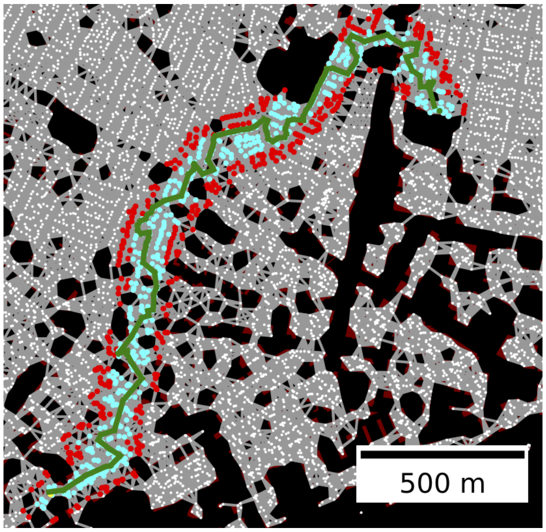


Figure 7: The results from a single simulation. The green line indicates the path selected by CityMesh's building route algorithm. Light blue dots are APs that fall within the rebroadcast conduit and that transmit the packet. Red dots show APs that receive the packet but do not rebroadcast as they fall outside of the conduit. In this simulation, the packet is successfully delivered to the destination building.

Mobile Ad hoc Networks (MANETs). MANETs have routing solutions for decentralized wireless networks with mobile nodes, but these protocols are not scalable to city-scales [6] due to the overhead of topology maintenance and route discovery. Proactive routing algorithms, such as DSDV [19], GSR (Global State Routing) [60], OLSR [22], and BATMAN advanced [30] require that each mobile node maintain a routing table. The size and update frequency of these tables increase proportionally with network size, making them impractical for city-scale networks. Reactive routing algorithms, like AODV [48], DSR [25], and TORA [47] update routing tables only on demand. However, these algorithms still require route maintenance and the distribution of control packets. Upon each route construction request, a burst of control packets will forward through the city-scale network, quickly wasting the bandwidth which should be reserved for data packet transmissions. Hybrid routing solutions (e.g., HSR [21] and ZRP [16]) face the same scalability issues.

Geographical routing. To improve the scalability of MANETs, prior work uses the geographical position of devices acquired from positioning services (e.g., GPS), and geographical routing algorithms to reduce the maintenance overhead of routing tables. Some previous work uses the hierarchical landmark system [61], GLS [38], and BLS [46] for destination position lookups. However, these approaches require that nodes in the network communicate with location servers to update and retrieve node locations. These location lookups consume valuable network bandwidth, which should be reserved for actual data packets in a peer-to-peer wireless network. Further, these lookups introduce additional latency, as the location servers are not necessarily close to the requesting nodes. Some previous work uses geographical routing algorithms (e.g., GPSR [28, 29], GOAFR [32], GOAFR+[31], GDSTR [35], PVEX [36], DREAM [2], and FACE [4, 54]) to navigate around geographical voids where no wireless devices are present. Upon encountering a void, these algorithms identify the nodes at the perimeter of the void, and each algorithm selects the appropriate neighboring nodes to forward the packet. Consequently, these algorithms degrade when locations are imprecise, as would occur within buildings. Additionally, each device shares its location with the one-hop neighbors by broadcasting beacon packets and must carefully tune the beaconing interval to prevent network flooding or providing outdated location information. Subsequent work [49, 51] mitigates the degradation from localization errors but induces extra computation or communication overhead. CityMesh does not require neighbor position tracking via beacon packets, making the system stateless in the traditional sense (it uses maps, of course).

Vehicular Ad hoc Networks (VANETs). Due to the highly mobile nature of vehicles and frequent reconnections, conventional geographical position-based approaches struggle to accurately track the position of neighbors at each node. Some approaches for building VANETs such as AGPSR [53], MM-GPSR [63], GPSR-L [50], AGF [43], and CBF [12] adapt established MANET geographical routing algorithms to highly mobile environments. Other works for VANETs exploit road maps and urban structures, including GSR (Geographic Source Routing) [39], GPCR [40], GPSRJ+ [34], A-STAR [52], and GyTAR [23, 24]. These techniques inform their routing decisions by leveraging the fact that cars are constrained to roads and their movement is predictable. Similarly, CityMesh observes that *people are often in buildings, and wireless devices are often near people.*

LoRa. LoRa networks enable wireless communication across several kilometers. Previous studies [10, 33, 41, 55] have proposed building city-scale LoRa mesh networks. While LoRa extends communication range, it requires specialized hardware and experiences significant interference during concurrent transmissions due to its MAC layer design [13] and may have challenges with scalability [37].

Satellite. Recent developments in satellite connectivity like Starlink [56] and the inclusion of satellite radios in phones [20] are enabling the use of satellite networks to supplement connectivity when there is no cellular or Wi-Fi access. We consider satellite fallback communications to be insufficient in the context of routing messages across a metropolitan area due to high hardware and service costs, restrictions in the number of concurrent users in an area, limited existing hardware support, and power constraints of mobile devices for frequent direct communication.

6 FUTURE WORK

Simulation. As we develop CityMesh, we would like to improve the fidelity of our simulations. Simulation tools such as ns-3 [44] are an excellent starting point as it boasts widespread in the investigation of MANETs. Scaling ns-3 to the number of AP nodes necessary to run a simulation that covers a metropolitan area is a major design challenge in extending these results.

Real-World Evaluation and Deployment. Even with a high-fidelity simulation, demonstrating that CityMesh will operate under real-world wireless conditions is essential to motivating further development. Physical network characteristics such as wireless channel congestion and interference from obstacles would need to be evaluated in our target deployment environment. To that end, building out a testbed of a to-scale mesh network, albeit with a smaller coverage area than our vision of CityMesh, would help design and validate the CityMesh protocol. Testing proof-of-concept applications on this testbed provides actionable feedback to guide further development of the protocol.

7 CONCLUSION

We propose CityMesh, an implementation of a DFN that utilizes existing Wi-Fi infrastructure in cities. CityMesh is able to scale by avoiding the bookkeeping of routing metadata in favor of utilizing the physical layout of cities to make informed routing decisions.

CityMesh is one city-scale implementation of a DFN, a network that re-enables connectivity across urban areas for circumstances under which centralized Internet access is interrupted. Developing this network model facilitates our exploration of the challenges facing real-world DFN implementation by addressing DFN's infrastructure, decentralized application design, and security. CityMesh lays the groundwork for designing DFNs that are appropriate for different scales and operational domains.

ACKNOWLEDGMENTS

We thank John Cheng and the HotNets reviewers for their valuable feedback. This work was supported by DARPA Contract HR001120C0191, FastNICs 4202290027, NSF SHF-2107244, NSF CAREER-2144766, NSF PPoSS-2217099, NSF CNS-2211382, NSF FuSe-TG-2235466, Sloan fellowship FG-2022-18504, as well as Jump 2.0 ACE and CUBIC centers sponsored by SRC and DARPA.

REFERENCES

[1] M. Afanasyev, T. Chen, G. M. Voelker, and A. C. Snoeren. Analysis of a mixed-use urban wifi network: When metropolitan becomes neapolitan. In *ACM IMC*, 2008.

[2] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward. A distance routing effect algorithm for mobility (DREAM). In *ACM MobiCom*, 1998.

[3] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and evaluation of an unplanned 802.11b mesh network. In *MobiCom*, 2005.

[4] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *ACM Discrete algorithms and methods for mobile computing and communications (DIALM)*, 1999.

[5] V. Brik, S. Rayanchu, S. Saha, S. Sen, V. Shrivastava, and S. Banerjee. A measurement study of a commercial-grade urban wifi mesh. In *ACM IMC*, 2008.

[6] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *MobiCom*, 1998.

[7] T. B. Buildings. Fiber huts explained. https://thermobond.com/fiber-huts/.

[8] J. Camp, J. Robinson, C. Steger, and E. Knightly. Measurement driven deployment of a two-tier urban mesh access network. In *ACM MobiSys*, 2006.

[9] K. Chebrolu, B. Raman, and S. Sen. Long-distance 802.11b links: performance measurements and experience. In *ACM MobiCom*, 2006.

[10] J. R. Cotrim and J. H. Kleinschmidt. LoRaWAN Mesh Networks: A Review and Classification of Multihop Communication. *Sensors*, 20(15):4273, 2020.

[11] R. Durairajan, P. Barford, J. Sommers, and W. Willinger. InterTubes: A study of the US long-haul fiber-optic infrastructure. In *SIGCOMM*, 2015.

[12] H. Füßler, H. Hartenstein, M. Mauve, W. Effelsberg, and J. Widmer. Contention-based forwarding for street scenarios. *International Workshop in Intelligent Transportation (WIT)*, 2004.

[13] A. Gamage, J. Liando, C. Gu, R. Tan, M. Li, and O. Seller. Lmac: Efficient carrier-sense multiple access for lora. *ACM Transactions on Sensor Networks*, 2023.

[14] N. Ghazisaidi, H. Kassaei, and M. S. Bohlooli. Integration of WiFi and WiMAX-Mesh Networks. In *IEEE International Conference on Advances in Mesh Networks (MESH)*, 2009.

[15] M. Ghobadi and R. Mahajan. Optical layer failures in a large backbone. In *Proceedings of the 2016 Internet Measurement Conference*, IMC '16, page 461–467, New York, NY, USA, 2016. Association for Computing Machinery.

[16] Z. Haas. The zone routing protocol (zrp) for ad hoc networks. Technical report, 1998.

[17] Hak5. Wifi pineapple. https://shop.hak5.org/products/wifi-pineapple, Retrieved by June 28th 2023.

[18] M. Haklay and P. Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18, 2008.

[19] G. He. Destination-sequenced distance vector (dsdv) protocol. *Networking Laboratory, Helsinki University of Technology*, 135:1–9, 2002.

[20] A. Inc. Emergency SOS via satellite on iPhone 14 and iPhone 14 Pro lineups made possible by $450 million Apple investment in US infrastructure. https://www.apple.com/newsroom/2022/11/emergency-sos-via-satellite-made-possible-by-450m-apple-investment/, 2022.

[21] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen. Scalable routing strategies for ad hoc wireless networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1369–1379, 1999.

[22] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *IEEE International Multi Topic Conference*, 2001.

[23] M. Jerbi, S.-M. Senouci, R. Meraihi, and Y. Ghamri-Doudane. An Improved Vehicular Ad Hoc Routing Protocol for City Environments. In *IEEE ICC*, 2007.

[24] M. Jerbi, S.-M. Senouci, T. Rasheed, and Y. Ghamri-Doudane. Towards efficient geographic routing in urban vehicular networks. *IEEE Transactions on Vehicular Technology*, 58(9):5048–5059, 2009.

[25] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile computing*, pages 153–181, 1996.

[26] S. A. Jyothi. Characterizing the role of power grids in internet resilience, 2023.

[27] B. Karp and H.-T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *ACM MobiCom*, 2000.

[28] B. Karp and H.-T. Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *ACM MobiCom*, 2000.

[29] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. Geographic routing made practical. In *USENIX NSDI*, 2005.

[30] K. Kiran, N. P. Kaushik, S. Sharath, P. D. Shenoy, K. R. Venugopal, and V. T. Prabhu. Experimental Evaluation of BATMAN and BATMAN-Adv Routing Protocols in a Mobile Testbed. In *IEEE Region 10 Conference*, 2018.

[31] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric ad-hoc routing: Of theory and practice. In *ACM PODC*, 2003.

[32] F. Kuhn, R. Wattenhofer, and A. Zollinger. Worst-case optimal and average-case efficient geometric ad-hoc routing. In *ACM MobiHoc*, 2003.

[33] H.-C. Lee and K.-H. Ke. Monitoring of Large-Area IoT Sensors Using a LoRa Wireless Mesh Network System: Design and Evaluation. *IEEE Transactions on Instrumentation and Measurement*, 67(9):2177–2187, 2018.

[34] K. C. Lee, J. Haerri, U. Lee, and M. Gerla. Enhanced Perimeter Routing for Geographic Forwarding Protocols in Urban Vehicular Scenarios. In *IEEE Globecom Workshops*, 2007.

[35] B. Leong, B. Liskov, and R. T. Morris. Geographic routing without planarization. In *NSDI*, 2006.

[36] B. Leong, S. Mitra, and B. Liskov. Path vector face routing: Geographic routing with local face information. In *IEEE International Conference on Network Protocols (ICNP)*, 2005.

[37] C. Li, X. Guo, L. Shangguan, Z. Cao, and K. Jamieson. {CurvingLoRa} to boost {LoRa} network throughput via concurrent transmission. In *USENIX NSDI*, 2022.

[38] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris. A scalable location service for geographic ad hoc routing. In *MobiCom*, 2000.

[39] C. Lochert, H. Hartenstein, J. Tian, H. Füßler, D. Hermann, and M. Mauve. A routing strategy for vehicular ad hoc networks in city environments. In *IEEE Intelligent Vehicles Symposium*, 2003.

[40] C. Lochert, M. Mauve, H. Füßler, and H. Hartenstein. Geographic routing in city scenarios. *ACM SIGMOBILE mobile computing and communications review*, 9(1):69–72, 2005.

[41] D. Lundell, A. Hedberg, C. Nyberg, and E. Fitzgerald. A Routing Protocol for LoRA Mesh Networks. In *WoWMoM*, 2018.

[42] D. Mazieres, M. Kaminsky, M. F. Kaashoek, and E. Witchel. Separating key management from file system security. In *SOSP*, 1999.

[43] V. Naumov, R. Baumann, and T. Gross. An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In *MobiHoc*, 2006.

[44] Nsnam. ns-3. https://www.nsnam.org/, 2024.

[45] Openwrt. Openwrt. https://openwrt.org/, 2024.

[46] J. Ortiz, C. R. Baker, D. Moon, R. Fonseca, and I. Stoica. Beacon location service: a location service for point-to-point routing in wireless sensor networks. In *ACM/IEEE IPSN*, 2007.

[47] V. Park and M. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *IEEE INFOCOM*, 1997.
[48] C. Perkins and E. Royer. Ad-hoc on-demand distance vector routing. In *IEEE Workshop on Mobile Computing Systems and Applications*, 1999.
[49] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without location information. In *ACM MobiCom*, 2003.
[50] S. A. Rao, M. Pai, M. Boussedjra, and J. Mouzna. Gpsr-l: Greedy perimeter stateless routing with lifetime for vanets. In *IEEE International Conference on ITS Telecommunications*, 2008.
[51] K. Seada, A. Helmy, and R. Govindan. On the effect of localization errors on geographic face routing in sensor networks. In *ACM/IEEE IPSN*, 2004.
[52] B.-C. Seet, G. Liu, B. Lee, C. Foh, K. Wong, and K.-K. Lee. A-star: A mobile ad hoc routing strategy for metropolis vehicular communications. In *NETWORKING*, 2004.
[53] A. Silva, K. N. Reza, and A. Oliveira. An adaptive gpsr routing protocol for vanets. In *IEEE International Symposium on Wireless Communication Systems (ISWCS)*, 2018.
[54] H. Singh. *Compass routing on geometric graphs.* University of Ottawa (Canada), 1999.
[55] J. M. Solé, R. P. Centelles, F. Freitag, and R. Meseguer. Implementation of a lora mesh library. *IEEE Access*, 10:113158–113171, 2022.
[56] SpaceX. Starlink. https://www.starlink.com/, 2024.
[57] Team SimPy. Simpy: Discrete event simulation for python. https://gitlab.com/team-simpy/simpy, 2023.
[58] TeleGeography. Map of internet exchange points. https://www.internetexchangemap.com/.
[59] TP-Link. Ac750 wireless dual band router. https://www.tp-link.com/us/home-networking/wifi-router/archer-c20/, Retrieved by June 28th 2023.
[60] Tsu-Wei Chen and M. Gerla. Global state routing: a new routing scheme for ad-hoc wireless networks. In *IEEE ICC*, 1998.
[61] P. F. Tsuchiya. The landmark hierarchy: a new hierarchy for routing in very large networks. *ACM SIGCOMM Computer Communication Review*, 18(4):35–42, 1988.
[62] WiGLE. WiGLE API. https://api.wigle.net/, Dec. 2022.
[63] X. Yang, M. Li, Z. Qian, and T. Di. Improvement of gpsr protocol in vehicular ad hoc network. *IEEE Access*, 6:39515–39524, 2018.