Don't Shoot the Messenger: Localization Prevention of Satellite Internet Users

David Koisser
Technical University of Darmstadt
david.koisser@trust.tu-darmstadt.de

Richard Mitev
Technical University of Darmstadt
richard.mitev@trust.tu-darmstadt.de

Marco Chilese
Technical University of Darmstadt
marco.chilese@trust.tu-darmstadt.de

Ahmad-Reza Sadeghi
Technical University of Darmstadt
ahmad.sadeghi@trust.tu-darmstadt.de

***Abstract*—Satellite Internet plays an increasingly important role in geopolitical conflicts. This notion was affirmed in the Ukrainian conflict escalating at the beginning of 2022, with the large-scale deployment of the Starlink satellite Internet service which consequently demonstrated the strategic importance of a free flow of information. Aside from military use, many citizens publish sensitive information on social media platforms to influence the public narrative. However, the use of satellite communication has proven to be dangerous, as the signals can be monitored by other satellites and used to triangulate the source on the ground. Unfortunately, the targeted killings of journalists have shown this threat to be effective. While the increasing deployment of satellite Internet systems gives citizens an unprecedented mouthpiece in conflicts, protecting them against localization is an unaddressed problem.**

To address this threat, we present AnonSat, a novel scheme to protect satellite Internet users from triangulation. AnonSat works with cheap off-the-shelf devices, leveraging long-range wireless communication to span a local network among satellite base stations. This allows rerouting users' communication to other satellite base stations, some distance away from each user, thus, preventing their localization. AnonSat is designed for easy deployment and usability, which we demonstrate with a prototype implementation. Our large-scale network simulations using real-world data sets show the effectiveness of AnonSat in various practical settings.

1. Introduction

The Internet has fundamentally changed the way conflicts unfold and are perceived on the global stage. A crucial aspect is the growing role of social media with civilians sharing, publishing, and forwarding information, including sensitive strategic data. The term *hybrid warfare* [1] alludes to the increasing focus on information warfare, such as disinformation campaigns. For example, allegedly, journalists were specifically and lethally targeted in both the second Chechen war and the Syrian civil war, to control the public narrative [2]. Today, individual citizens can leverage social media to reach a global audience, further escalating the dynamics of (dis-)information.

This new paradigm is particularly visible in the Ukrainian conflict that escalated in February 2022. United Nations appointed independent rights experts warned that journalists in Ukraine are targeted and in danger [3], with 15 journalists confirmed to have been killed in Ukraine in 2022 [4]. Another perspective on the conflict claims Ukraine and its citizens got the upper hand in the so-called *social media war* [5]. Indeed, the U.S. government recognized the importance of a free flow of information among Ukrainians. After spending millions of dollars to fund the widespread deployment of Starlink satellite Internet terminals and service in Ukraine [6], the service quickly reached over 150 000 users shortly after deployment [7]. While a large share of Starlink's usage in Ukraine seems to be military, the Starlink smartphone app was downloaded over 806 000 from Ukraine, making it the most downloaded app at the time [8]. As a response, Russia has started a barrage of cyber and jamming attacks on Starlink since [9], yet they were repelled [10]. While the precedent of Starlink's satellite Internet in a conflict is still ongoing at the time of writing, the E.U. already announced a deal to deploy its own satellite Internet system [11].

Clearly, satellite-based Internet has a significant impact, as citizens and journalists gain the ability to freely share, e.g., sensitive information, evidence of war crimes, or timely warnings. However, openly sharing information also carries great risks, and thus, many social media companies have added additional security measures to protect Ukrainian users, along with some guidelines to minimize risks [12]. Moreover, there is a specific danger when using satellite-based communication, especially when used to publish sensitive information. As satellite signals can be monitored by virtually anyone in the sky and space above, satellite uplink communications can also be used to geolocate their users on the ground by triangulating their signals. One example is the killings of two American journalists [13] and another a missile strike on the leader of the Chechen republic [14]. In both cases it is assumed that the attacks were possible by

tracing satellite phones. While official confirmation of such state-backed attacks is rare, it was shown that techniques to geolocate transmitters by satellites (target tracking) are practical [15]. After a security researcher gained traction with a tweet warning Ukrainian Starlink users potentially being geolocated and becoming targets [16], the CEO of Starlink's company issued a public warning to Ukrainian Starlink users [17].

Considering the scale at which a satellite-based Internet can be monitored, and worse, individual users triangulated to get their physical position, it is an important aspect to protect citizens against this threat while preserving this novel and free flow of information during conflicts. However, there are no works to properly address this issue in a practical manner. On the one hand, using typical Internet encryption (i.e., TLS) on the communication channel is insufficient. Eavesdropping on satellite communication targets the actual physical medium, whereas TLS is a high-level protocol not designed to provide anonymity. The Tor network aims to fix this problem for the traditional Internet. However, our case has a crucial difference, as satellite communications can be monitored by any satellite and, worse, triangulated to geolocate the user. For example, numerous attacks on Tor assume an adversary can do *entry point* monitoring [18], [19]. While typically an ambitious position for the adversary, with a satellite-based Internet, it becomes quite straightforward, as the connection is first sent to the satellite before reaching the Tor network. Other attacks on Tor assume the adversary can monitor both the *entry* and *exit point* [20], [21], [22]. However, if the adversary aims to prevent a user from publicly sharing information, it may anticipate and monitor popular social media sites, such as Twitter, for the exit point.

Prior works on *location privacy* in mesh and wireless sensor networks have different shortcomings [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42]. For example, some works have impractical assumptions for our purposes and others induce significant overheads. There are also works that aim to establish a reliable network in case the existing infrastructure fails, called *emergency networks* [43], [44], [45], [46], [47], [48]. These approaches are typically based on specialized hardware, such as vehicles equipped with bulky communication equipment and even flying vehicles, and custom network protocols to facilitate basic communication, making them quite impractical for our purposes. There are also emergency networks working with satellites [49], [50], [51]; yet, they do not consider protection against triangulation. We will discuss the related work more thoroughly in Section 8.

In summary, the remote monitoring and the possibility to triangulate satellite Internet users is a global threat to the new-found free flow of information by citizens. To the best of our knowledge, there is no existing system that prevents geolocating satellite Internet users. In this paper, we present *AnonSat* to close this gap.

Goals & Contributions: Our primary goal is to hide the geographic position of satellite Internet users in case their connection is being triangulated. AnonSat works by leveraging long-range wireless communication to span a simple network among satellite base stations. Our system is agnostic with respect to the used wireless communication technology. Leveraging this local network, a client's WAN connection is routed to another randomly selected satellite base station, which acts as a delegate to do the actual connection uplink to the satellite. Further, the targeted satellite base station is regularly changed to avoid tracing back a long-lasting connection.

Our secondary goal is to focus on the accessibility of our system. Therefore, AnonSat is designed to work with cheap and simple devices, and thus, it can be deployed with widely available hardware and does not require impractical extensions on the user's device, such as a specific app or radio device. Further, we aim at the usage of popular Internet services, like Twitter or WhatsApp, refraining from custom network protocols. AnonSat effectively protects users from being geolocated and becoming targeted.

Our main contributions include:

- AnonSat is the first scheme to address the triangulation of satellite Internet users by rerouting connections to more distant satellite base stations. We introduce two security parameters that adjust the selection of routes to avoid geolocating a user over time.
- We derive requirements for wireless communication technologies and give an overview of possible candidates that can be used to enable AnonSat. Similarly, due to our aim to design a practical system, we discuss numerous approaches, such that AnonSat can access typical Internet services without needing to install custom software or hardware on the user's device.
- We implemented a proof of concept demonstrating the feasibility of AnonSat, leveraging a cheap Raspberry Pi equipped with a LoRa shield for the local network.
- We further developed a large-scale simulation using real-world data sets to evaluate key aspects of AnonSat in different environments, such as effective distances from the user or the use of more powerful wireless technologies.

2. System Model

Our system model consists of the following entities: A **network** is a collection of *gateways*, *clients*, and satellites providing internet access. Each **gateway** is equipped with a base station, i.e., means of providing access to the Internet via satellite communication, a radio transmitter to connect to the *local network*, and two WiFi access points to provide access to the *WiFi Network*. A **local network** is spanned among the *gateways* via their radio transmitter capable of communicating with each other. A **WiFi network** is a direct connection between *clients* and *gateways* separated into two WiFi access points. One is a connection to the gateway's WAN, directly uplinking to the satellite internet. The other

provides a secure WiFi connection via our AnonSat system. **Clients** are simple end-user devices, such as smartphones, which aim to establish a secured WAN connection to send sensitive data. For this, *clients* simply connect to the secure *WiFi Network* provided by a close-by *gateway*.

Note, for simplicity, we assume all *gateways* have a base station providing Internet access, even though in a real-world scenario simple relay nodes for the *local network* may also be deployed. We further assume that standard means of Internet access are impaired or even unavailable entirely, and thus clients need to rely on satellite Internet.

Furthermore, *gateways* are equipped with certificates to identify each other and to establish secret key pairs to enable symmetric encryption between any two *gateways*. We assume the *gateways* can trust each other's certificates. For example, in the real world, this could be realized via exchanging certificates via direct contacts in combination with a Web-of-Trust approach.

2.1. Adversary Model and Assumptions

The adversary $\mathcal{A}$ has the goal of geolocating a specific client. To do this, $\mathcal{A}$ has a range of satellites deployed, which can eavesdrop on the Internet communication between base stations and the receiving satellite. We assume that $\mathcal{A}$ is able to correlate the communication data to identify a specific client. Further, $\mathcal{A}$ is capable of triangulating the sending base station via the mentioned satellites, which was shown to be practical [15]. Thus, as the client has to use the closest base station via a close-range WiFi connection, $\mathcal{A}$ can infer the client's approximate geographic position. However, as our system aims to reroute the client's connection to another gateway, simply triangulating the base station is not enough to geolocate the client.

We assume $\mathcal{A}$ is not capable of establishing a holistic view of the local network. To achieve this, $\mathcal{A}$ would need to monitor a significant number of local network connections, which also requires prolonged physical proximity in a multitude of locations. This contradicts the scenario we are targeting, i.e., an active conflict zone, as widespread deployment of eavesdropping devices is infeasible. However, $\mathcal{A}$ is able to intercept individual messages sent between gateways. This assumption is comparable to the Tor network, which can also be broken by an adversary with a global view of the network; yet, in practice, this is hard to achieve. We further argue due to the nature of the limited range of each node (discussed in Section 4.1), the local network cannot be surveilled by satellites to attain a global view.

We will thoroughly discuss several possible local attacks in Section 7, including jamming attacks that deal with similar assumptions. Nevertheless, our system focuses on preventing remote and globally applicable triangulation via satellites.

Further, we assume the WiFi connection between the client and gateway cannot be intercepted by $\mathcal{A}$ due to its close-range nature. We also assume $\mathcal{A}$ aims to minimize any collateral damage, as this might have grave political repercussions [52], [53]. Finally, we further assume $\mathcal{A}$ cannot forge digital signatures or break symmetric encryption.

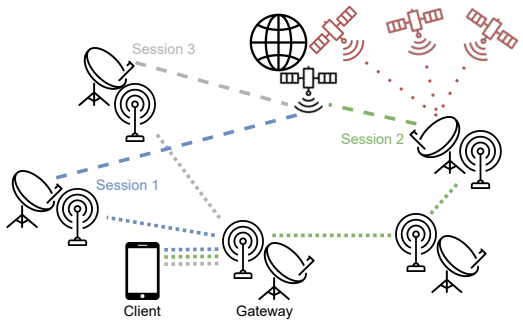

Figure 1. An example setup of AnonSat with five gateways and one client.

2.2. Requirements

To formalize the setting outlined in the Introduction, we aim to design a secure satellite Internet scheme with the following requirements:

R.1 *Prevent geolocating base station:* The scheme shall prevent $\mathcal{A}$ from geolocating the client. More specifically, the gateway uplinking traffic to the WAN shall not indicate the geographic position of the client. For example, as a client has to use the closest gateway, if this gateway uplinks the client's traffic to the WAN, $\mathcal{A}$ may assume the client is very close.

R.2 *Prevent local geolocation leakage*: In addition to requirement R.1, $\mathcal{A}$ may intercept individual messages in the local network traffic and trace back the actually used gateway by the client. Thus, the scheme shall further prevent geolocation leakage in terms of the local network.

R.3 *Internet compatibility:* The use of, e.g., simplified network protocols may greatly increase the performance of the scheme. However, this implies that most common Internet services are not accessible, and thus, the scheme shall be compatible with most Internet services.

R.4 *Out-of-the-box for clients:* From the client's point of view, the scheme shall impose minimal requirements on the client. For example, a client may need to unexpectedly and urgently send some sensitive data using a smartphone. In such a case, requiring the client to install an additional app or even an additional hardware device is impractical.

3. AnonSat Design

In this section, we focus on the general design of AnonSat. However, as our focus is designing a practical system (cf. Section 2.2), we will discuss essential technical aspects in Section 4. Figure 1 illustrates a simplified setup of AnonSat with five gateways, each with a base station to connect to the Internet and a local radio transmitter to communicate with other gateways. The client uses a

smartphone to upload sensitive data to a gateway in close proximity, called the *origin*. For example, the client may try to publish an incriminating picture on Twitter. Instead of directly forwarding the data over the satellite link, the gateway randomly selects an output gateway. In *Session 1* in the figure, the origin will then transmit the client's data over the local radio connection to the output gateway, which in turn will do the actual satellite transmission. Thus, if the output gateway is identified and triangulated by $\mathcal{A}$, the actual geographic position of the client stays hidden. After a chosen amount of time, the origin gateway will select a new output gateway for this client's connection in *Session 2*. This new output gateway is only reachable via an intermediate gateway, and thus, establishes a 2-hop connection. As seen in *Session 3*, the origin will continue to change the client's output gateway regularly while the connection lasts.

Changing Gateways. While we assume $\mathcal{A}$ cannot eavesdrop on the entire local gateway network (cf. Section 2.1), $\mathcal{A}$ may be able to intercept individual messages. As a first step, all communication between the gateways is per-hop encrypted. Thus, a forwarding gateway will receive an encrypted message, decrypt it, encrypt it with the key established with the next-hop gateway, and forward it. However, if $\mathcal{A}$ intercepts enough messages in relation to a specific client, $\mathcal{A}$ may be able to trace the origin gateway, and thus, geolocate the client eventually. Therefore, we introduce the security parameter $gateway\_timeout$, which defines a timeframe. When a client starts a connection and the origin gateway selects a random output gateway, a timer is started. After $gateway\_timeout$ time, the origin will select a new output gateway. Choosing a proper value for $gateway\_timeout$ depends on the bandwidth and delay of the local gateway network. Note that $gateway\_timeout$ may also define a message count instead of a timeframe. If $gateway\_timeout$ is set too high, $\mathcal{A}$ has an increasing chance to trace back the origin. Setting $gateway\_timeout$ too low may lead to technical problems with the client's connection. We will discuss this further in Section 4.4. Note, AnonSat's goal is not to make all traffic indistinguishable from unobjectionable traffic, as current approaches, such as Dummy Data Sources create non-negligible overhead (we discuss this in Section 8).

Distance to Origin. The selection of the output gateways has some crucial implications as well. If the origin only selects very close gateways, then the clients are not adequately protected. If the output gateways are too far away, this would require many hops between the origin and the output gateways, negatively affecting the performance of AnonSat. Thus, we introduce the security parameter max_hops, which defines how many hops away the output gateway shall be from the origin. Increasing this parameter increases the average distance from the client (i.e., the origin gateway) to the output gateway, and thus, increasing the protection of the client. However, increasing max_hops will negatively affect the performance of the client's connection.

Output Selection Bias. While simply routing the client's traffic to output gateways max_hops hops away is fine for individual short sessions, we need to consider a crucial aspect for longer sessions. Suppose we have a uniformly spread network of gateways and a client that needs a long-lived connection to, e.g., upload many pictures. In this case, many gateway changes happen over time and the selected gateways will eventually be selected in all directions from the origin. Simply put, $\mathcal{A}$ can observe these changes and will be able to draw a circle containing all output gateways. The gateway closest to the centroid of this circle is most likely the origin gateway; thus, endangering the client. To counteract this effect, we additionally introduce a selection bias for the output gateway. For each client, the origin gateway will generate a random *direction* and *weight* bias. When selecting a new output gateway, the origin will prefer random output gateways in the given direction and with the assigned weight. In a practical context, the direction can simply be a bias for selecting the next neighbor gateway via an index without the need to consider the gateways' geolocations. This way, the centroid of the mentioned circle shifts to a random direction, and thus, cannot be used to trace the origin. The selected biases will be preserved for each client.

Additionally, this allows us to change the role of the max_hops security parameter. Instead of always selecting an output gateway exactly max_hops hops away, we can select a range of hops between 0 and max_hops. This does not weaken the previously discussed selection bias. However, the range improves the client's network performance on average, as some output gateways may be only one hop away and fewer hops mean better performance for the client. Note that it is important for the origin to select itself as the output gateway. Otherwise, $\mathcal{A}$ can simply identify the origin by checking which gateway never sends.

4. Technical Considerations

After we described the theoretical design in Section 3, it is crucial to consider the technical challenges of AnonSat to satisfy requirements R.3 and R.4. Therefore, this section will discuss key practical aspects to implement our theoretical design. Note, we focus on available techniques or techniques currently in deployment, i.e., we will not address recent research approaches, as these might take many more years until ready for deployment. Our focus is on techniques usable in a practical deployment now or in the near future.

4.1. Wireless Communication Technology

As AnonSat relies on a local network between gateways, this section discusses possible options for wireless communication technologies. Fortunately, due to the prevalence of the Internet of Things (IoT), there have been many proposals and advancements in recent years to enable remote IoT devices to connect directly to the Internet or via a mesh network. We will leverage these advancements for AnonSat.

Table 1 shows an overview of the technologies we considered. Note, this table is not comprehensive of all available technologies, as we carefully selected technologies we deem

TABLE 1. OVERVIEW OF LONG-RANGE RADIO TRANSMISSION TECHNOLOGIES APPLICABLE TO ANONSAT.

Name	Maximum Data Rate	Range (urban)	Range (rural)	Unlicensed Frequency Bands
LoRa Sub-GHz [54]	27 kbps 50 kbps (FSK)	5 km	15 km	✓
LoRa 2.4 GHz	250 kbps 1 Mbps (FLRC)	1 km	n/a	✓
LTE-M Cat-M1 [55]	1 Mbps 4 Mbps (Cat-M2)	1 km	10 km	✗
NB-IoT Cat-NB2 [56]	200 kbps	1 km	10 km	✗
DASH7 [54]	166 kbps	5 km	n/a	✓
Weightless-W [54]	10 Mbps	5 km	n/a	✗

applicable to AnonSat. For example, while Sigfox is a mature wireless technology already deployed in many regions around the world, it only supports 100 bps data rates [56], which is too slow to be meaningfully used to connect to traditional Internet services (violating requirement R.3).

One may also consider repurposing existing infrastructure, such as the cellular network for mobile Internet. However, in a conflict, existing infrastructure (e.g., cell towers) is unreliable or may not be functional at all, making satellite Internet necessary in the first place. In addition, repurposing this infrastructure for a disaster network is not feasible as cellular networks do not have mesh network capabilities and access to the hardware is limited.

For the *Maximum Data Rate*, we also considered alternatives, like the Fast Long Range Communication (FLRC) for Lora 2.4 GHz, which uses demodulation and error correction techniques for a significantly improved data rate [57]. Further, the last column of Table 1 shows if the respective technology uses unlicensed frequency bands. For example, Lora supports a variety of different sub-GHz bands, which are publicly usable in the respective region, e.g., North America allocates different bands than Europe [56]. While the other technologies use licensed bands, we expect that legal restrictions play a lesser role in the settings applicable to AnonSat or may even be officially lifted.

4.2. MTU Size Mismatch

One technical hurdle with a significant practical impact is the Maximum Transmission Unit (MTU), which defines the maximum amount of bytes sent in a single network layer transaction. Considering our scenario targeting the Internet, typically the MTU of Ethernet is used (1500 bytes). The hurdle arises when using a wireless communication technology, which only supports a smaller MTU. For example, LoRa restricts the MTU to 255 bytes, which leads to problems with Internet servers expecting a large MTU, such as extra negotiation steps for smaller messages and spurious retransmissions. For example, in our preliminary tests using a small MTU, we saw significant delays with TLS handshakes, even to a point in which some servers simply declined the session entirely. Another downside of a small MTU is the overhead of the headers, such as the IP and TCP protocols. With a small MTU, the headers consume a significant share of the bandwidth.

Therefore, a mechanism is required to split Ethernet MTUs received by the client to fit them into, e.g., multiple LoRa frames for forwarding them over the local network. These kinds of operations are usually implemented by the Operative System's (OS) kernel. An example of LPWAN is IEEE 802.15.4 (*Low-rate Wireless Personal Area Network*) [58] over IPv6, documented as 6LoWPAN in RFC 8930 [59], which specifies how to forward 6LoWPAN fragments over a multi-hop network. The implementation is already available in the Linux Kernel [60] for IEEE 802.15.4 compliant devices. This could benefit AnonSat greatly, as it was demonstrated that a proper fragmentation strategy can lead to significant performance improvements [61]. Yet, not all wireless technologies fit this standard. For example, there are discrete definitions of header compression and segmentation for LoRa in RFC 9011 [62]; yet, there are no implementations so far.

4.3. Slow TCP Connections

The Transmission Control Protocol (TCP) and how servers handle TCP sessions are usually optimized for fast and reliable connections nowadays, as this is the most common scenario. However, for AnonSat this is not the case. Indeed, in our case, the connections are slow and potentially lossy, leading to problems with many TCP deployments. The most relevant one is the way retransmissions are handled, as typical deployments are optimized to maximize data rates for fast connections. One aspect is the Retransmit Timeout (RTO), which is typically set quite low, such that the server can react quickly to network congestion, which is also identified with timeouts. If the acknowledgement for a packet is not received by the server in time, to optimize data rates for typical connections, the server will quickly do a retransmission. This results in many duplicate, and thus, unnecessary retransmissions with slow connections called *spurious* retransmissions, quickly saturating the connection. Another problem to consider is the loss of packets, especially when using one of the wireless communication technologies (cf. Section 4.1). For example, today's TCP deployments bundle the transmission of multiple resources over a single connection to achieve a form of concurrency. However, when a packet is lost for one resource, this delays all of the resources, resulting in even more retransmissions.

Unfortunately, the most effective TCP settings to avoid these issues are controlled by the endpoints, such as retransmission timeouts or the used congestion control algorithm. Thus, in AnonSat, we cannot change these settings, as we can only influence the gateways. An exception to this is the `txqueuelen` property for each network interface in Linux. Settings this to very low values (e.g., 1) prevents the client to send many packets in a short time, which will exacerbate the mentioned retransmission problems. Another setting to consider is setting TCP's window size, influencing how much data is bundled for each acknowledgment. Setting

this to a low value further helps to avoid retransmission problems.

Another approach to counteract these problems is optimizations for the used wireless communication technologies in a multi-hop setup. A local retransmission scheme may be used among the gateways, essentially dealing with packet losses on the local gateway network level. There are different strategies for such local retransmissions [63]. Handling retransmissions locally would circumvent many of the issues with TCP packet losses.

Furthermore, a more sustainable solution is being deployed at the time of writing and will likely be widely available in the near future. QUIC [64] is a network protocol introduced by Google as a more performant and flexible alternative to TCP. Among other functionalities, it leverages UDP with merged handshakes, custom congestion controls, loss detection, and retransmission algorithms [65]. This allows QUIC to handle slow connections with high latency with much better performance than TCP. For example, Google published a large-scale performance study on QUIC, which showed that QUIC can improve latency by over 30% compared to TCP with large round-trip-times in a common Internet scenario [66]. Thus, QUIC would likely have a significant performance impact for our purposes.

Practically speaking, QUIC is already a reality, as many companies are already adopting it. For example, the company Meta already deployed it for most of its applications [67].

4.4. Changing Routing Paths

As described in Section 3, AnonSat needs to reroute packets through different nodes. Typically, TCP sessions stay alive until the end of the communication and it is not possible to dynamically change the stream's endpoints, i.e., the source or destination IP. However, changing the output gateway implies a change in the endpoint IP; thus, we need to adopt a strategy for establishing a new endpoint.

A possible way for achieving this goal is using the Reset (RST) flag in the TCP header. When this flag is set, the receiver will close the TCP session immediately. The use of the RST flag is also used for malicious applications, such as the so-called "TCP Reset Attack" [68]. Here an attacker forges TCP packets with a set RST flag in order to maliciously interrupt or disturb the Internet connection. However, for our purposes, the origin gateway can leverage the RST flag to stop a TCP session and force the endpoints (i.e., the client and the Internet server) to start a new session with a different endpoint, i.e., another output gateway selected by the origin. Indeed, for AnonSat, this comes with the additional overhead of establishing a new TCP session regularly. Thus, when using this approach the $gateway\_timeout$ parameter should not be set too low.

However, this is exactly what the Multipath TCP (MPTCP) protocol sets out to do in a more elegant way. The protocol was originally documented in 2013 by RFC 6824 [69] and later updated with RFC 8684 [70]. This protocol allows a single TCP session to take different routing paths and use different endpoints without interruption. As widespread support is relevant, MPTCP is already implemented in the Linux Kernel [71] and, e.g., Apple's iOS uses it to be able to quickly switch between a WiFi and cellular connection [72].

Another elegant way is provided by the aforementioned QUIC protocol. While there is also a multipath extension for it in the making [73], similar to MPTCP, the original QUIC protocol already supports *Connection Migration* [64]. Here, each session is assigned a connection ID, which allows the connection to survive endpoint address changes.

Furthermore, depending on the used underlying protocols for the local network, e.g., IPv4 or LoRaWAN, the client needs additional protection. Namely, the output gateway should execute a Network Address Translation (NAT) on the connection, translating its own address to the origin. Otherwise, the address of the origin gateway may leak, and thus, potentially reveal the client's geographic position.

4.5. Node Discovery

The local gateway network acts as a mesh network, and thus, needs protocols for discovering nodes and establishing routing tables for later network message routing. For our purposes, a link state routing protocol works well. While there are many different approaches to these types of protocols, we expect the local gateway network to only observe a limited degree of dynamics, as opposed to, e.g., a mobile ad-hoc network. The latter typically requires more advanced techniques for routing and node discovery. Thus, the Optimized Link State Routing Protocol, defined in RFC 3626 [74], or its successor [75] is sufficient for our approach. Here, neighbors exchange information about their neighbors to build simple routing tables, which show the shortest path to any node in the network.

Further, there are also protocols to dynamically optimize the radio settings between nodes. For example, there is the *Adaptive Data Rate* optimization between LoRa nodes [76]. These protocols may be used to further optimize the data rate for the local gateway network.

4.6. Leakage in Presence of Malicious Nodes

In the mesh and wireless sensor network research area for location privacy (we give an overview of them in Section 8), there are approaches additionally considering defenses against compromised nodes [77]. Many approaches like *Network Coding* and *In network location anonymization* are designed to protect against an attacker with a *global* view, which is infeasible in our discussed scenario (cf. Section 2.1). These approaches introduce many new requirements and overheads, making them inapplicable for our scenario. However, approaches considering a *local* adversary are not applicable due to their assumptions. One type of approach assumes a hierarchical structure among the nodes in the network, in which certain nodes are trusted and cannot be compromised [30], [78], [36]. Due to the nature of our targeted scenario, assuming only some gateways to

be trusted is not practical in AnonSat. A different approach is to use end-to-end encryption between source and destination, such that a potentially compromised intermediary node cannot know the endpoints of a message [79]. However, this requires that all possible endpoint pairs have pre-shared keys deployed, which is infeasible in our scenario. Another approach can only hide the destination of the connection, not the source [27]. Yet, protecting the source, i.e., the origin gateway, is the main goal of AnonSat. A different approach protects against compromised nodes, yet relies on the assumption that intermediary nodes cannot be too close to the source [80]. We deem this assumption too strict to be practical for AnonSat.

The main concern for AnonSat regarding compromises is malicious intermediary gateways, which can extract the origin gateway and thus target the client. Onion routing [81] is an effective method to hide the origin gateway from compromised intermediaries. No intermediary hop can know its position in the path, as a packet's amount of onion shells is unknown. This is due to AnonSat randomly choosing a path length between 0 and max_hops. Additionally, onion routing can also be used in wireless networks [82], where, when mapped to our approach, no node (except for the first and last) is able to learn the source of a message.

5. Prototype Implementation

To demonstrate AnonSat, we describe the implementation of our Proof of Concept in this section. While we have shown advanced techniques in Section 4, many of them lack either proper wide-ranging support or applicable implementations. At this point in time, we consider the integration of these techniques as a significant engineering effort and out of scope for this research work. Note, however, there is potential to significantly improve the practical performance of AnonSat. We will give justifications for the choice of the respective techniques in the following.

Scenario. For our prototype, we deployed five off-the-shelf consumer devices. For the Internet connection, we deployed a Starlink dish [83] on the roof of our building. We used three gateways deployed as Raspberry Pi 3 Model B+ [84] (called *RaspberryPi1*, *RaspberryPi2* and *RaspberryPi3*), equipped with a sub-GHz Lora SX1262 868M shield [85]. Finally, we used a simple Android phone as the client. *RaspberryPi3* acts as a proper gateway, as it is connected to the Starlink router, which is connected to the dish. All Raspberry Pis are interconnected via Lora and provide a WiFi access point for the client to connect. Figure 2 shows the client connected to the origin's secure WiFi access point as well as the output gateway of our test setup.

We have chosen to use sub-GHz Lora due to its use of unlicensed frequency bands. Another aspect was the availability of development kits that include proper integration for both hardware (e.g., connection to our Raspberries) and software (e.g., driver). Note, we have also evaluated using the Lora 2.4 GHz shield SX1280Z3DSFGW1 [86]; yet, we found that the provided implementation by the manufacturer[1] is impractical, as it has limitations that severely limit the effective data rates, even below the Lora sub-GHz shield mentioned above. For example, the implementation demands a 50 ms sleep after each packet is sent, meaning a 1500 bytes MTU split into 7 Lora packets already implies a 350 ms delay before considering the actual transmission time.

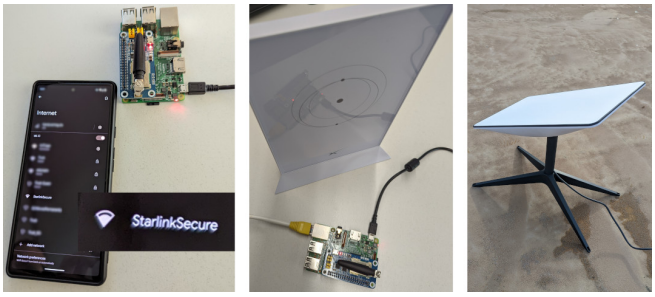

Figure 2. Our test setup. The left picture shows the client with the origin gateway, the center picture shows the output gateway connected to the Starlink router, and the right picture shows the Starlink dish that is connected to the Starlink router.

Implementation. To send IP packets over LoRa, we utilize the open source software `tncattach`[2] which creates and sets up network interfaces to translate IP to LoRa and back. Unfortunately, the implementation has no support for packet fragmentation and reassembly, and therefore, only utilizes the maximum LoRa packet size (236 bytes excluding header overhead), leading to the problem discussed in Section 4.2. To speed up the translation (and avoid package retransmissions) between Ethernet (typical MTU of 1500 bytes) and LoRa, we added simple fragmentation support to `tncattach`. The more advanced techniques for fragmentation and header compression implemented in the Linux kernel [60] are not applicable to LoRa, and the LoRa-specific specification [62] has no implementation.

In terms of routing, we pre-deploy our routing tables for simplicity, as opposed to implementing an advanced protocol, as described in Section 4.4. For example, *RaspberryPi1* sets up a Wi-Fi Access Point with its own subnet and forwards all packages to the LoRa interface using another subnet utilizing NAT. *RaspberryPi2* receives these packets and forwards them to *RaspberryPi3*. *RaspberryPi3* is then forwarding them to the Starlink router connected via Ethernet, utilizing NAT again. By providing every device its own static IP address, the operating system takes over tasks such as device discovery and routing.

As the origin gateway, *RaspberryPi1* is in charge of setting up routes for the client's traffic to keep track of the duration of each link. As described in Section 4.4, we chose to break TCP connections using a Reset Attack. We utilized `scapy`[3] to listen to TCP packets traveling on *RaspberryPi1's* interfaces. As the WiFi Access Point utilizes DHCP, a list of connected devices is known, including the connec-

1. https://github.com/Lora-net/gateway_2g4_hal
2. https://github.com/markqvist/tncattach
3. https://scapy.net/

TABLE 2. AVERAGE RTT AND PACKET LOSS FOR PINGING 8.8.8.8 WITH 100 64 BYTES ICMP.

LoRa Hops	RTT	Packet Loss
0	49.111 ms	0%
1	157.938 ms	3%
2	211.786 ms	4%

TABLE 3. AVERAGE TIME FOR UPLOADING AND DOWNLOADING IMAGES OF DIFFERENT SIZES USING 2 LORA HOPS.

Image Size	Upload	Download
50 kB	65.84 s	60.40 s
100 kB	137.84 s	117.80 s
150 kB	171.92 s	223.20 s
200 kB	269.31 s	276.80 s

tion time; thus, we can listen to established TCP connections of these devices using a packet filter. If a connection is kept alive for too long (i.e., longer than $gateway\_timeout$), the origin gateway creates a TCP packet with a set RST flag and injects it into the packet flow in both directions, killing the connection. Afterwards, the origin can establish a new route for this client.

Further, to counteract the TCP retransmission problems outlined in Section 4.3, we set the `txqueuelen` to 1 and TCP's TX buffer to exactly one packet. This mitigated many spurious retransmissions created because of delayed ACKs.

6. Evaluation

In this section, we evaluate AnonSat. On the one hand, we show real-world results of our prototype setup. On the other hand, we show the large-scale performance in different settings based on a network simulator run on real-world data sets.

6.1. Prototype

Recall Section 2.1, the goal of our approach is to hinder $\mathcal{A}$ to localize a client, e.g., publishing incriminating information. Usually, such information is published in the form of text or images. Therefore, to measure the performance of our prototype, we considered three use cases for the client in our setup. One is simply sending messages via the popular WhatsApp messenger. Another was to send out tweets via the Twitter Lite app. The third is to publish an image. For accurate measurement numbers, we used pings as well as downloaded and uploaded small images via the client.

Round-Trip-Time (RTT). We measured the RTT using a standard ping via zero, one and two LoRa hops to a server on the internet via the Starlink connection. Note that using Starlink alone already adds around 50 ms of delay. We send 100 pings and averaged the results. The results are depicted in Table 2.

Upload & Download. We implemented our own API service to upload images using REST over TLS using HTTP/2. We used a POST form request to upload a picture of size between 50 kB and 200 kB. These numbers correspond to the lower and upper bounds of typical mobile applications image compression (e.g., WhatsApp), which we measured with common photographs. Similarly, we downloaded the files using `wget`. We repeated this experiment 10 times and averaged the results. The results are shown in Table 3. Note, while we employed both fragmentation for the LoRa packets and optimized the TCP settings, as discussed in Section 5, we still observed a significant amount of spurious retransmissions.

Naturally, AnonSat incurs an overhead. Securely uploading a high-resolution photo using WhatsApp (i.e., 150 kB) takes 171.92 s. However, we argue it is reasonable considering the alternative of being either localized or not publishing at all.

6.2. Simulation

To measure the large-scale performance of AnonSat, we implemented a network simulation. In the following, we describe our evaluation setup by first describing how we simulated the local wireless network, presenting the used real-world data sets, and how the network simulation works. Afterwards, we present our results showing how the security parameter max_hops affects the distance from the position of a client, the delays to establish a TLS session in different settings, and finally, the practical data rates.

6.2.1. Local Wireless Network. While we were restricted to Lora with the sub-GHz frequencies, the simulator allows us to simulate more powerful wireless technologies. Informed by the data shown in Table 1, we selected the following combinations of assumed ranges and data rates between nodes:

1) 5 km @50 kbps (Lora Sub-GHz)
2) 5 km @166 kbps (DASH7)
3) 1 km @1 Mbps (Lora 2.4GHz & LTE-M Cat-M1)
4) 1 km @4 Mbps (LTE-M Cat-M2)

We excluded NB-IoT due to its low performance compared to the other technologies on our list, which is mostly due to its low-power requirement. We further excluded Weightless-W, even though its impressive data rates, as we could not establish the readiness of the technology. For example, unlike the other technologies, we could not find any purchasable devices equipped with Weightless-W components or any practical demonstrators.

Thus, for our simulation we assume, e.g., a range of 5 km between nodes with a maximum data rate of 50 kbps, simulating Sub-GHz Lora. However, the maximum data rate is practically not achievable, especially at longer ranges. While there are some practical measurements regarding this phenomenon, we found the used evaluation setups (e.g., obstructions or radio settings) and results vary significantly[4].

4. For example, one work measured around 10 kbps [87] while another measured double the data rate [88] for similar settings.

TABLE 4. URBAN WIFI HOTSPOT DATASETS USED FOR SIMULATION. *Close* REFERS TO THE NUMBER OF RECORDS AFTER FILTERING OUT TOO CLOSE RECORDS. *CC* REFERS TO THE NUMBER OF RECORDS CONTAINED IN THE LARGEST CONNECTED COMPONENT WHEN CONNECTING THE GRAPH WITH THE GIVEN RANGE.

Data Set	Total	Close	CC 5 km	CC 1 km
Hong Kong [89]	5441	874	866	332
New York City [90]	3319	765	753	602
Rhein-Neckar [91]	1338	551	530	30
Brisbane [92]	347	97	95	38
Paris [93]	277	182	181	178
Adelaide [94]	272	51	51	51
Leeds [95]	236	169	163	83
Linz [96]	124	35	34	29

For our simulation, we approximate the *log-distance path loss model* as a simple logarithmic function over the distance between nodes $r = e^{-2d}$. d is the distance between two nodes as a relative distance $[0, 1]$ with respect to the maximum range. The resulting data rate r is also relative $[0, 1]$ to the maximum data rate. Thus, the maximum data rate is only achievable if two nodes are right next to each other, while two nodes that are far away, e.g., close to the maximum range, have a severely reduced data rate with only a small fraction of the maximum data rate.

6.2.2. Data Sets. As far as we are aware, there are no representative data for gateway positions for the settings we target. Nevertheless, we found public data sets on various cities' WiFi hotspots to be a good approximation for an urban environment. Table 4 lists all of the data sets we used, each containing the geographic positions of WiFi hotspots for cities of different sizes. Figure 3 shows renderings of the Hong Kong, New York City, and Rhein-Neckar data sets.

However, we had to filter these data sets to fit our needs for the simulation, due to the following two problems. For one, these data sets contain points that are very close together, which is to be expected, e.g., in the city center, there will be a large number of shops or similar with a high density of hotspots. As such a dense concentration of gateways is not representative in our settings, we filtered out nodes that are closer than 200 m from each other. In Table 4, the number of nodes left after this filtering step is shown as *Close*. The second problem is that our assumed maximum range leads to a disconnected graph, as some sub-graphs may not be in range for another sub-graph. Thus, we constructed a graph of nodes from the data sets with the respective maximum range and found the set of connected components in the overall graph. Finally, we chose the largest connected component for each data set as the actual set of nodes for our simulation. In Table 4, this is shown as *CC 5 km* and *CC 1 km* for a range of 5 km and 1 km respectively. Note, for some data sets the 1 km range limitation creates a very small network, such as Rhein-Neckar, which covers a large area, but many of the nodes are further than 1 km away from each other. Therefore, this results in a significant reduction of the size, if the used connected network, e.g., Rhein-Neckar *CC 1 km* only has ∼5.4% nodes left relative to *Close*. However, note that Rhein-Neckar is an exceptional outlier, due to the data set spanning relatively few nodes over almost 100 km. The other data sets with higher reductions comprise of dense clusters that are far from one another, e.g., see Figure 3 for Hong Kong. In such a scenario, AnonSat could be applied individually for each cluster. Generally, a restriction true for all physical wireless networks is that a more densely packed network results in a holistically better connectivity among nodes [97]. Therefore, we stress that, if there are no alternatives to using satellite Internet (cf. Section 2), even a suboptimal network setup benefits from our approach.

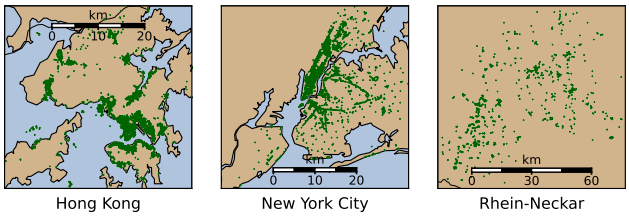

Figure 3. Renderings of the three largest data sets shown in Table 4. Each green dot is one geographic position.

6.2.3. Network Simulator. To implement the network simulation, we used the OMNeT++ 6.0 network simulator [98]. We load the processed data sets, as described in Section 6.2.2, as the individual nodes into the simulation. These nodes are then connected, if they are in range of each other, with a data rate calculated at initialization, as described in section 6.2.1. We further use the provided routing component in OMNeT++ to route individual messages as well as to ensure max_hops is satisfied when randomly selecting gateways by the clients.

For the actual simulation, we assign each client to a random gateway as its origin. Each client will then proceed to execute the following steps:

1) The client's gateway will randomly select an output gateway less than max_hops hops away.
2) The client will send a `TCP SYN` message out to simulate establishing a connection. This message is routed to the output gateway.
3) After the output gateway received each message, we simulate a 100 ms delay for the WAN server to answer[5].
4) The output gateway will send a TCP `SYN-ACK` message back to the client's gateway.
5) The client will answer this with a TCP `ACK` and TLS `ClientHello` combined message, as is a common optimization practice of the Internet.
6) The simulated server will answer this with a TLS `ServerHello` message; thus, establishing the TLS connection when the client receives it.
7) The client will then start sending a 200 kB data package (the upper bound for WhatsApp image compression)

5. We based this number on the upper average of 50 ms delay with Starlink and some additional processing time.

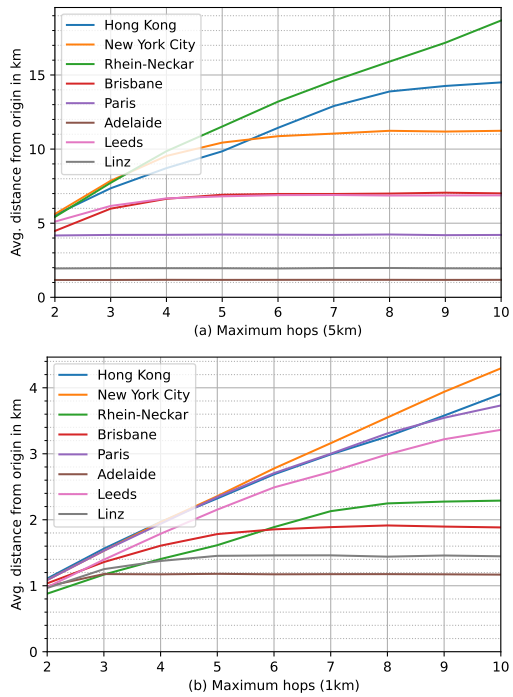

Figure 4. Graphs showing the average distance from a chosen output gateway to the origin for different max_hops over all data sets with (a) 5 km and (b) 1 km range.

divided into multiple messages, i.e., according to the MTU sizes.

The client will repeat these steps, selecting a new output gateway each time, until the simulation ends after a simulated hour. Put simply, each client has a fixed origin and will constantly send images with changing output gateways. For each parameter combination and data set, we execute 30 runs with different random seeds. While quite a simple setup, this allows us to approximately measure TLS session delays, the effective data rates in the network, and the interactions of both, e.g., TLS session delays of clients over a gateway currently busy sending many data messages.

6.2.4. Distance to Origin. To evaluate the key security parameter max_hops, we measured the average distance from the client's gateway (origin) to the output gateway. For this, we employed a simplified simulation over our data sets (cf. Table 4) to get a large sample size of 10 000. For each sample, we select a random origin with a random output gateway less than max_hops away and measured the actual distance from the origin via their geographic position. Further, we executed this simulation with differently set max_hops. The results of this simulation are shown in Figure 4. Wide-ranging data sets in terms of the overall covered area by the nodes, like Hong Kong or Rhein-Neckar, show a nearly linear increase in distance with an increase of max_hops. We discuss this more thoroughly in Section A.

6.2.5. TLS Session Delay. To measure the performance of AnonSat, we focus on two aspects: delays and data rate. However, as the primary goal of AnonSat is security, we focus on practical applications. Namely, for delay we measure the average delay it takes to establish a TLS session. As most Internet services today are based on these sessions, this is a more practical number than measuring simple round-trip delays, especially as we want to see the effects of concurrent data transfers and session establishments in the network.

Figure 5 shows our measurements. Noticeable between the slower networks (a) & (b) and the faster networks (c) & (d) is the effect of max_hops with many clients. With max_hops = 5 congestion of the gateways affects the delay significantly. With a range of 5 km (a) & (b), the two data sets Adelaide and Linz show a much better performance, due to all nodes being packed closely together in a small area, which also results in much better data rates on average. Note that there is an implicit tradeoff, as the low average distances between nodes naturally affects the *distance to origin*, as shown in Section 6.2.4. A similar effect can be observed for the 1 km range measurements (c) & (d). Data sets with closely packed nodes in general, like New York City, show much better performance than data sets with spread-out nodes, like Brisbane.

6.2.6. Practical Data Rates. In a practical scenario, the data rates may depend on multiple TLS sessions first, which would heavily reduce effective the data rates when simply calculating overall sent bytes divided by time. Thus, we decided to measure the time it takes to upload a 200 kB sized image as a practical and isolated example.

Figure 6 shows our measurements. Generally, the observation made in Section 6.2.5 regarding the effect of the density of nodes on data rates applies here as well. However, while the delays imply a logarithmic trend with a growing number of clients, we can clearly see the effects of many clients sending data and the resulting congestion in the network. Particularly noticeable are the measurements for Paris with a 5 km range. The increase in the number of clients has an especially detrimental effect on the transmission speed. We believe this is due to the unique and dense spread of nodes in the data set, creating some *bottleneck* nodes that serve multiple connections simultaneously. This effect disappears with a 1 km range, as fewer nodes can connect to these bottleneck nodes.

7. Anonymity & Security

In this section, we first analyze the practical anonymity guarantees of AnonSat and the security of AnonSat.

7.1. Anonymity

We quantify and evaluate the anonymity provided by AnonSat on the data sets described in Section 6.2.2. Table 5 summarizes our analysis for each data set for both the 5 km and 1 km cases, with max_hops = 3 and max_hops = 5,

respectively. In the following, we will explain our metrics and evaluate our results.

The first metric we evaluated is the traditional *Anonymity Set* size as defined by Chaum [99]. The idea is that one's anonymity can be quantified by the set of parties from which one is not distinguishable. Thus, $\mathcal{A}$ ideally does not know, which of the parties in the set is the actual target. In our case, we counted all gateways reachable by an output gateway, i.e., less than max_hops away, as these are the potential alternatives among the origin gateway. We considered all gateways in our data sets for both the average and the gateway with the lowest number of reachable gateways in Table 5. While a dense data set like New York City has a high average, the less dense Rhein-Neckar set has a comparably low average. When looking at the minima, i.e., the least connected gateways, some data sets contain remote nodes with few reachable nodes.

Nevertheless, just counting the possible alternatives is not a complete metric, as the probabilities to be the origin among the set of reachable gateways may not be uniform. A way to model this is to consider the entropy of the anonymity set based on implicit information on the network and its nodes leveraged by $\mathcal{A}$ [100], e.g., the reachability of each node throughout the graph. In our case, similar to our Anonymity Set metric, we look at all reachable gateways from an output gateway. In addition, we consider the number of paths between each reachable gateway and the output gateway, as a gateway with more paths to the output gateway is more likely to be the origin than gateways with fewer. With this entropy, we are able to calculate the *Effective Set* size for the anonymity set (see Table 5). Concretely, we calculate per reachable gateway g the number of paths to the output gateway divided by all possible paths from any reachable node to the output gateway as p_g (c.f. [100]). With this, we can calculate the effective anonymity set for all g in the reachable gateway set for an output gateway:

$$-\sum p_g \log_2(p_g)$$

We considered all gateways in our data sets to get both the average and the minimum effective anonymity set. Compared to the uniform anonymity set, data sets that are more clustered in terms of gateway distribution have a significantly lower effective set size (e.g., Rhein-Neckar 5 km with a $\sim 39\%$ reduction) than data sets that are well-connected (e.g., Paris 5 km with a $\sim 4\%$ reduction). These numbers demonstrate that different data sets with different degrees of connectivity among the gateways result in varying degrees of anonymity, which needs to be considered in practical deployment. However, the average effective anonymity set sizes show the efficacy of AnonSat.

Finally, we measured the average number of paths between any two gateways as *Node2node Paths* within max_hops distance. This gives a hint at the general connectivity among the nodes. To further refine this metric, we also counted all *Unique Paths*, i.e., the number of paths that do not share any common gateways in their route. This indicates how reliable AnonSat is when individual gateways fail, i.e., the number of alternative paths. In Table 5, we can examine that in more dense networks with many connections, like Paris 5 km, there are many alternative paths on average. Contrarily, if there are few connections between nodes, like Paris 1 km, then there are few alternative paths on average. Generally, a higher number shows a more resilient network against gateway failures.

7.2. Security

The adversary $\mathcal{A}$ aims to geolocate a specific client by identifying the origin gateway. $\mathcal{A}$ may use the following strategies to accomplish this: (1) $\mathcal{A}$ assumes the used output gateway is close enough and target it instead, (2) $\mathcal{A}$ uses individual intercepted messages from the local network to trace the origin, and (3) $\mathcal{A}$ monitors the gateways for extended periods to collect data pointing to the origin.

Strategy (1) is prevented in our system by rerouting communication away from the origin gateway used by the client. In Section 6.2.4, we analyze the effective distances achieved on real-world data sets. Nevertheless, due to our *Changing Gateway* approach (cf. Section 3), eventually, the actual origin gateway is used for the satellite uplink. Thus, $\mathcal{A}$ may simply target each gateway and eventually be successful. However, this would effectively result in a large-scale attack, e.g., in an urban context, targeting the entire city. As stated in our adversary model (Section 2.1), we deem this undesirable for $\mathcal{A}$. Note that this also applies to $\mathcal{A}$ taking kinetic measures against gateways in subregions of the network. As described in Table 5 (column *Unique Paths*), every network has at least two unique paths between any two nodes, i.e., alternative paths that do not share any nodes. This demonstrates the network's resilience against forceful disconnection, even in the presence of a costly, wide-ranging attack on many gateways instead of targeting individual ones. We deem this undesirable, as $\mathcal{A}$ tries to limit its attacks as much as possible. Further, in case $\mathcal{A}$ captures extensive territory, resulting in few gateways in the region, we assume that clients will not remain in the area and transmit sensitive data.

For the second strategy (2), $\mathcal{A}$ is prevented from learning any information about the route with the end-to-end encryption employed by the gateways. Yet, $\mathcal{A}$ may intercept numerous local gateway messages over time and eventually be able to correlate the route from the output gateway back to the origin. Our system prevents this by regularly changing the output gateway (cf. Section 3). The effectiveness of this approach is dependent on the gateway distribution, which we evaluate in Section 6.2.4 on our real-world data sets.

$\mathcal{A}$ may monitor all used output gateways by a client to infer the origin. The success of this strategy (3) is prevented by the *Selection Bias*, as described in Section 3, which shifts the centroid of all used gateways over time away from the origin.

In the following, we discuss additional local attacks, motivating our adversary model.

Total Local Network Monitoring. In Section 2.1, we assume $\mathcal{A}$ is unable to monitor the entire gateway network

TABLE 5. ANONYMITY METRICS FOR ALL DATA SETS FOR BOTH THE 5 KM AND 1 KM CASES.

	5 km Average Anonymity Set	5 km Minimum Anonymity Set	5 km Average Effective Set	5 km Minimum Effective Set	5 km Average Node2node Paths	5 km Average Unique Paths	1 km Average Anonymity Set	1 km Minimum Anonymity Set	1 km Average Effective Set	1 km Minimum Effective Set	1 km Average Node2node Paths	1 km Average Unique Paths
Hong Kong	417.3	12	278.0	9.1	6 080.3	54.2	111.8	26	60.1	21.8	5 147.9	5.7
New York City	523.7	13	330.0	12.6	6 415.1	59.9	92.7	8	47.7	8.0	3 734.4	4.2
Rhein-Neckar	98.5	11	60.4	7.5	193.1	8.9	18.3	13	12.2	9.1	116.2	2.3
Brisbane	83.7	17	61.9	14.9	862.9	20.2	35.3	30	26.7	20.8	116.5	4.3
Paris	180.0	179	172.7	162.4	10 096.9	93.8	78.4	12	43.0	9.4	332.2	2.7
Adelaide	50.0	50	50.0	50.0	2 402.0	50.0	50.0	50	44.7	43.7	2 045.5	13.4
Leeds	149.7	49	121.4	39.5	1 880.1	35.6	37.9	10	22.5	8.4	61.2	2.1
Linz	33.0	33	33.0	33.0	959.8	31.6	37.9	26	20.7	19.2	834.3	4.6

to establish a holistic view of the local network. $\mathcal{A}$ would aim to trace back routes from the output gateway back to the origin. $\mathcal{A}$ would need to get an extensive coverage of the gateway network. Practically speaking, to achieve this, $\mathcal{A}$ needs to deploy numerous devices, which must be widely spread in close proximity to the gateways and potentially deployed over long periods, as it is unknown where and when the client may become active. Considering the potential scale of an active conflict, we deem this strategy infeasible.

Jamming. $\mathcal{A}$ may try to use jamming to interrupt the client's ability to communicate. There are three types of jamming to consider. One type is targeting the satellites, e.g., with a high-power ground-based jamming signal directed at individual satellites. However, with over 3 000 Starlink satellites deployed [101] at the time of writing, this strategy does not scale well. Further, according to reports, Starlink used specialized firmware updates to withstand numerous jamming attacks [10].

An additional type of jamming is adversarial satellites jamming ground stations. Theoretically, a signal can be considered jammed if the Signal to Noise Ratio (SNR) at the receiver is 1. To achieve this the jamming signal must be received with at least the same power as the benign signal [102]. The power of electromagnetic signals decays quadratically with distance. Thus, a satellite targeting a ground station would need a jamming signal powerful enough to cover the vast distances in space. As satellites are significantly limited in terms of power, we deem this strategy infeasible.

Another strategy is local jamming ground-to-ground, as modern jammers may cover a wide area. However, depending on the scale of the gateway network, $\mathcal{A}$ would need to either deploy many jammers or target the client's general area, which might be unknown. In case $\mathcal{A}$ *is* able to jam the client's gateway, this is difficult to circumvent; however, in this case, $\mathcal{A}$ is not able to geolocate the client.

Malicious Gateways. $\mathcal{A}$ may try to deploy malicious gateways. However, we deem this strategy to be unviable. Similar to the holistic monitoring of the gateway network, $\mathcal{A}$ would need to deploy or compromise a plethora of devices spread throughout the gateway network to reliably trace clients. Individual gateways may reveal the client's route if the client actually routes over them. However, unlike *overlay* networks, in which an adversary can remotely create new nodes, $\mathcal{A}$ must control *physical* gateways in advantageous geographic locations to reliably target clients. Thus, such an elaborate strategy requires physical access, resulting in low probabilities of success. To protect against leakage of the origin gateway by intermediary malicious nodes, approaches such as onion routing can be deployed to AnonSat, as described in Section 4.6.

8. Related Work

To the best of our knowledge, our approach is the first work to address the triangulation of satellite Internet users in critical circumstances. Thus, we could not find directly related work for our purposes. However, one related research topic is providing *location privacy* in mesh and wireless sensor networks. Another related research topic is *emergency communication networks*, which, similarly to AnonSat, often employ mesh-like network topologies and are designed to work under exceptional situations.

Location Privacy in Mesh and Wireless Sensor Networks. In these networks, due to their physical properties, communication is vulnerable to tracking and monitoring. If vulnerable participants utilize such a network, keeping the (geographic) position undisclosed becomes essential. Approaches for location privacy can be grouped into eleven categories [77].

The high-level idea of *Random Walk*-based approaches is to direct packets to traverse a network through a random path to a sink (e.g., base station). This makes the path of a packet unpredictable to an adversary attempting local traffic analysis [23], [24]. *Geographic Routing* is similar to Random Walk, yet utilizes the physical location information of nodes for more efficient routing of packets towards the sink. For location privacy, these approaches additionally leverage pseudonyms, reputation, and a fixed set of intermediary nodes creating a mix subnetwork [25], [26]. Both these types of approaches assume a *backtrack* or *hunter* adversary model, in which the adversary starts close to the sink, traces each sent-out message back to the next hop, and this is repeated until the adversary eventually arrives at the source. However, this is incompatible with our assumptions, as all

nodes in AnonSat are sinks, we regularly switch the sink, and a backtracking adversary is unlikely in a conflict zone.

In *Delay*-based solutions, nodes store incoming packets and transmit them after a random period of time, disrupting the chronological order of the packets. This also modifies the traffic pattern, rendering it difficult for a local adversary to trace the origin of the traffic [27], [28]. *Limiting node detectability* temporarily throttles or disables the transmission power of nodes, making it harder for an adversary to receive packets [29], [30]. *Network Coding* uses homomorphic encryption at intermediary nodes to hide traffic flows. After receiving the aggregated data, the sink is then able to reverse the encryption process [31], [32]. However, these three classes of solutions add significant delays to the network that adversely affect low-latency networks, such as AnonSat, especially when the aim is to be compatible with the Internet (cf., Section 4.3).

Dummy Data Sources generate authentic-looking dummy traffic to obscure the authentic traffic. The objective is to prevent an adversary from differentiating between genuine and fabricated traffic [33], [34]. *Cyclic Entrapment* confuses potential adversaries by routing the traffic between nodes with cyclical patterns [35], [36]. *Separate Path Routing* splits data into multiple packets, which will be sent over multiple, non-intersecting paths to the sink. Therefore, the local adversary is only able to capture part of the data [37]. Similarly to introducing delays, these three approaches induce large traffic overheads to the network (e.g., dummy traffic or additional retransmissions). Thus, they are incompatible with the goals of AnonSat (cf., Section 2.2).

Cross Layer Routing utilizes multiple OSI layers to hide information from adversaries [38]. Yet, this is based on the assumption that the adversary may not see certain OSI layers, which we deem impractical. Approaches focusing on *Wireless Mesh Networks* specifically, usually assume a hierarchical network with base stations, mesh routers, and mesh clients. Numerous security features, including location privacy, rely on pseudonyms in the form of public key certificates, either directly distributed by an authority [39] or are self-generated with a domain authority backing it [40]. *In network location anonymization* utilizes hierarchical (e.g., clusters of nodes) pseudonyms or aggregation of traffic to hide the source of the traffic [41], [42]. Both the Wireless Mesh Networks and In network location anonymization approaches assume a hierarchical trust structure in the network, which is not feasible in AnonSat's local network as, e.g., all gateways are set up by citizens and trusted equally.

Emergency Communication Networks. These networks are designed to provide reliable communication during an emergency when other communication infrastructures fail. The research community has since proposed many approaches for establishing a network in case of natural disasters, conflicts, or any adverse situation that prevents typical access to the Internet. In particular, Portmann *et al*. [43] defined the fundamental characteristics an emergency network must have in its design: *privacy, data integrity, authentication, and access control*. The most common emergency networks are considering the use of Locally Deployed Resource Units (LDRU) (e.g., base stations of cellular networks), satellites, ad-hoc networks, or a combination of them. Usually, portable devices span a network, while only a subset of them are actually capable of external means of communication (e.g., satellites) [103], [104].

To establish emergency networks, many recent works focus on using Unmanned Aerial Vehicles (UAVs) [105]. Numerous works in this area leverage *drones* to establish the emergency network. One proposal is to directly leverage the drones as base stations [44]. Other works use drones to span a mesh network back to a static base station. Proposed wireless communication technologies for the mesh range from leveraging WiFi [45], LTE [46], 5G [106], and LoRa [47]. Besides the use of drones, the application of aerostatic balloons found space in the context of emergency networks. One approach is to build an ad-hoc network based on IEEE 802.11j between the ballons [107]. Another approach is to span a multihop WiFi backbone from one area to a satellite-based base station via zeppelin-like balloons [108], while another extends this approach to a mesh network [109]. Besides all the possible proposed designs, multiple recent works are proposing several optimizations, including load balancing between the units [48], [110].

A practical concern is that drones exhibit limited fly times, and thus, coverage. Further, the individual hardware required (i.e., the drones and balloons) is expensive; yet, hundreds or even thousands of units are necessary to operate in an emergency. Moreover, drones and aerostatic balloons are subject to weather conditions (e.g., cannot easily fly in a storm), which is heavily limiting their operative scenario.

Other works focus on deploying an ad-hoc mesh network between base stations that are then put into communication with satellites. Zhou *et al*. [49] develop such a network based on WiFi 2.4 GHz for the network backbone and 5.8 GHz for data transmission. However, due to the limited range of WiFi, this approach requires the devices to be quite close to each other and does not scale well to cover a wide area. Instead, Iapichino *et al*. [50] propose a hybrid system where equipped vehicles (Vehicle Communication Gateways) establish a connection with satellites and users can connect to these mobile gateways. Similarly, Patricelli *et al*. [51] are proposing a MOBSAT access point (that has to be carried by car or helicopter), which provides high-speed data connection to the users through WLAN and WiMAX through GEO satellites. Nevertheless, these approaches assume that numerous, specially equipped vehicles are prepared and ready for use.

In contrast, the target scenario of AnonSat is novel in the context of emergency networks. The continuous expansion of satellite Internet services allows for the deployment of comparably cheap access points. This enables to provide widespread access to many deployed satellite base stations. Therefore, the mentioned works above are not taking the unique problems of this scenario into account. As outlined in the Introduction, AnonSat specifically aims to protect its users from triangulation. Furthermore, our focus is on the accessibility of the design system. Thus, AnonSat does not

require any additional application or hardware installed on the user's device and the gateways are easy to deploy.

9. Conclusion

In this work, we presented AnonSat, the first scheme to address the triangulation of satellite Internet users, and thus, protect them from being targeted. To achieve this, AnonSat leverages a local wireless communication technology to span a network between the gateways, rerouting a client's connection away from the origin gateway. Additionally, AnonSat regularly changes the output gateway for each client to avoid detection of long-lasting connections. We thoroughly discussed different technical aspects, meeting our defined requirements to make AnonSat usable with both existing Internet protocols and widely available hardware. We implemented a prototype demonstrating AnonSat's feasibility and evaluated its performance via network simulation on various real-world data sets.

Acknowledgment

This work was supported by the European Space Operations Centre with the Networking/Partnering Initiative.

References

[1] F. G. Hoffman, *Conflict in the 21st century: The rise of hybrid wars*. Potomac Institute for Policy Studies Arlington, 2007.

[2] Foreign Policy Magazine, "Kill the Messenger," https://foreignpolicy.com/2012/03/03/kill-the-messenger/, 2012.

[3] United Nations News, "Ukraine: Journalists targeted and in danger, warn top rights experts," https://news.un.org/en/story/2022/05/1117462, 2022.

[4] Committee to Protect Journalists, "Journalists and Media Workers Killed in Ukraine," https://cpj.org/data/killed/europe/ukraine/, 2022.

[5] British Broadcasting Corporation, "How Ukraine is winning the social media war," https://www.bbc.com/news/world-europe-63272202, 2022.

[6] The Washington Post, "U.S. quietly paying millions to send Starlink terminals to Ukraine, contrary to SpaceX claims," https://www.washingtonpost.com/politics/2022/04/08/us-quietly-paying-millions-send-starlink-terminals-ukraine-contrary-spacexs-claims/, 2022.

[7] Daily Mail, "SpaceX Starlink internet service has 150,000 daily users in Ukraine as citizens of war-torn parts of the country fight to stay connected, government official reveals," https://www.dailymail.co.uk/sciencetech/article-10781461/SpaceX-Starlink-150-000-daily-users-Ukraine-just-five-weeks-activated.html, 2022.

[8] TeslaRati, "Starlink broke top 100 most downloaded iPhone apps on Wednesday," https://www.teslarati.com/starlink-top-100-iphone-apps-wednesday/, 2022.

[9] Space.com, "Elon Musk says Russia is ramping up cyberattacks on SpaceX's Starlink systems in Ukraine," https://www.space.com/starlink-russian-cyberattacks-ramp-up-efforts-elon-musk, 2022.

[10] Breaking Defense, "SpaceX beating Russian jamming attack was 'eyewatering': DoD official," https://breakingdefense.com/2022/04/spacex-beating-russian-jamming-attack-was-eyewatering-dod-official/, 2022.

[11] Reuters, "EU secures deal on satellite internet system," https://www.reuters.com/business/aerospace-defense/eu-secures-deal-satellite-internet-system-2022-11-17/, 2022.

[12] The Washington Post, "Social media companies push Ukrainian users to add safeguards," https://www.washingtonpost.com/technology/2022/02/26/protecting-identity-socialmedia/, 2022.

[13] Electronic Frontier Foundation, "Satphones, Syria, and Surveillance," https://www.eff.org/deeplinks/2012/02/satphones-syria-and-surveillance, 2012.

[14] Radio Free Europe/Radio Liberty, "10th Anniversary Of Chechen Leader's Death Noted," https://www.rferl.org/a/1067831.html, 2006.

[15] A. Elgamoudi, H. Benzerrouk, G. A. Elango, and R. Landry Jr, "A survey for recent techniques and algorithms of geolocation and target tracking in wireless and satellite systems," *Applied Sciences*, vol. 11, no. 13, p. 6079, 2021.

[16] Wayback Machine / Twitter, "Archived Tweet of John Scott-Railton," https://web.archive.org/web/20220301105339/twitter.com/jsrailton/status/1497745011932286979, 2022.

[17] New York Post, "Elon Musk warns Starlink users in Ukraine could be Russian targets," https://nypost.com/2022/03/04/elon-musk-warns-starlink-users-in-could-be-russian-targets/, 2022.

[18] S. J. Murdoch and P. Zieliński, "Sampled traffic analysis by internet-exchange-level adversaries," in *International workshop on privacy enhancing technologies*. Springer, 2007, pp. 167–183.

[19] M. Yang, X. Gu, Z. Ling, C. Yin, and J. Luo, "An active de-anonymizing attack against tor web traffic," *Tsinghua Science and Technology*, vol. 22, no. 6, pp. 702–713, 2017.

[20] K. Bauer, D. Grunwald, and D. Sicker, "Predicting tor path compromise by exit port," in *2009 IEEE 28th International Performance Computing and Communications Conference*. IEEE, 2009, pp. 384–387.

[21] S. Le Blond, P. Manils, A. Chaabane, M. A. Kaafar, C. Castelluccia, A. Legout, and W. Dabbous, "One bad apple spoils the bunch: Exploiting P2P applications to trace and profile tor users," in *4th USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET 11)*, 2011.

[22] F. Palmieri, "A distributed flow correlation attack to anonymizing overlay networks based on wavelet multi-resolution analysis," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2271–2284, 2019.

[23] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, "Enhancing source-location privacy in sensor network routing," in *25th IEEE international conference on distributed computing systems (ICDCS'05)*. IEEE, 2005, pp. 599–608.

[24] Y. Xi, L. Schwiebert, and W. Shi, "Preserving source location privacy in monitoring-based wireless sensor networks," in *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium*. IEEE, 2006, pp. 8–pp.

[25] R. A. Shaikh, H. Jameel, B. J. d'Auriol, H. Lee, S. Lee, and Y.-J. Song, "Achieving network level privacy in wireless sensor networks," *Sensors*, vol. 10, no. 3, pp. 1447–1472, 2010.

[26] Y. Li and J. Ren, "Preserving source-location privacy in wireless sensor networks," in *2009 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*. IEEE, 2009, pp. 1–9.

[27] X. Hong, P. Wang, J. Kong, Q. Zheng *et al.*, "Effective probabilistic approach protecting sensor traffic," in *MILCOM 2005-2005 IEEE Military Communications Conference*. IEEE, 2005, pp. 169–175.

[28] P. Kamat, W. Xu, W. Trappe, and Y. Zhang, "Temporal privacy in wireless sensor networks: Theory and practice," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 4, pp. 1–24, 2009.

[29] R. Rios and J. Lopez, "Exploiting context-awareness to enhance source-location privacy in wireless sensor networks," *The Computer Journal*, vol. 54, no. 10, pp. 1603–1615, 2011.

[30] R. El-Badry, A. Sultan, and M. Youssef, "Hyberloc: providing physical layer location privacy in hybrid sensor networks," in *2010 IEEE International Conference on Communications*. IEEE, 2010, pp. 1–5.

[31] Y. Fan, Y. Jiang, H. Zhu, and X. Shen, "An efficient privacy-preserving scheme against traffic analysis attacks in network coding," in *IEEE INFOCOM 2009*. IEEE, 2009, pp. 2213–2221.

[32] Y. Fan, J. Chen, X. Lin, and X. Shen, "Preventing traffic explosion and achieving source unobservability in multi-hop wireless networks using network coding," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*. IEEE, 2010, pp. 1–5.

[33] Y. Yang, M. Shao, S. Zhu, and G. Cao, "Towards statistically strong source anonymity for sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 9, no. 3, pp. 1–23, 2013.

[34] K. Mehta, D. Liu, and M. Wright, "Location privacy in sensor networks against a global eavesdropper," in *2007 IEEE International Conference on Network Protocols*. IEEE, 2007, pp. 314–323.

[35] Y. Ouyang, X. Le, G. Chen, J. Ford, and F. Makedon, "Entrapping adversaries for source protection in sensor networks," in *2006 International symposium on a world of wireless, mobile and multimedia networks (WoWMoM'06)*. IEEE, 2006, pp. 10–pp.

[36] L. Kazatzopoulos, C. Delakouridis, G. F. Marias, and P. Georgiadis, "ihide: Hiding sources of information in wsns," in *Second International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU'06)*. IEEE, 2006, pp. 8–pp.

[37] H. Wang, B. Sheng, and Q. Li, "Privacy-aware routing in sensor networks," *Computer Networks*, vol. 53, no. 9, pp. 1512–1529, 2009.

[38] M. Shao, W. Hu, S. Zhu, G. Cao, S. Krishnamurth, and T. La Porta, "Cross-layer enhanced source location privacy in sensor networks," in *2009 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*. IEEE, 2009, pp. 1–9.

[39] Y. Zhang and Y. Fang, "Arsa: An attack-resilient security architecture for multihop wireless mesh networks," *IEEE Journal on Selected areas in communications*, vol. 24, no. 10, pp. 1916–1928, 2006.

[40] J. Sun, C. Zhang, Y. Zhang, and Y. Fang, "Sat: A security architecture achieving anonymity and traceability in wireless mesh networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 2, pp. 295–307, 2010.

[41] S. Misra and G. Xue, "Efficient anonymity schemes for clustered wireless sensor networks," *International Journal of Sensor Networks*, vol. 1, no. 1-2, pp. 50–63, 2006.

[42] X. Luo, X. Ji, and M.-S. Park, "Location privacy against traffic analysis attacks in wireless sensor networks," in *2010 International Conference on Information Science and Applications*. IEEE, 2010, pp. 1–6.

[43] M. Portmann and A. A. Pirzada, "Wireless mesh networks for public safety and crisis management applications," *IEEE Internet computing*, vol. 12, no. 1, pp. 18–25, 2008.

[44] N. Zhao, W. Lu, M. Sheng, Y. Chen, J. Tang, F. R. Yu, and K.-K. Wong, "Uav-assisted emergency networks in disasters," *IEEE Wireless Communications*, vol. 26, no. 1, pp. 45–51, 2019.

[45] K. G. Panda, S. Das, D. Sen, and W. Arif, "Design and deployment of uav-aided post-disaster emergency network," *IEEE Access*, vol. 7, pp. 102 985–102 999, 2019.

[46] M. Deruyck, J. Wyckmans, W. Joseph, and L. Martens, "Designing uav-aided emergency networks for large-scale disaster scenarios," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, pp. 1–12, 2018.

[47] M. Pan, C. Chen, X. Yin, and Z. Huang, "Uav-aided emergency environmental monitoring in infrastructure-less areas: Lora mesh networking approach," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2918–2932, 2021.

[48] N. Lin, Y. Liu, L. Zhao, D. O. Wu, and Y. Wang, "An adaptive uav deployment scheme for emergency networking," *IEEE Transactions on Wireless Communications*, vol. 21, no. 4, pp. 2383–2398, 2021.

[49] J. Zhou, C. Zhou, Y. Kang, and S. Tu, "Integrated satellite-ground post-disaster emergency communication networking technology," *Natural Hazards Research*, vol. 1, no. 1, pp. 4–10, 2021.

[50] G. Iapichino, C. Bonnet, O. del Rio Herrero, C. Baudoin, and I. Buret, "Advanced hybrid satellite and terrestrial system architecture for emergency mobile communications," in *26th international communications satellite systems conference (ICSSC)*, 2008.

[51] F. Patricelli, J. E. Beakley, A. Carnevale, M. Tarabochia, and D. K. Von Lubitz, "Disaster management and mitigation: the telecommunications infrastructure," *Disasters*, vol. 33, no. 1, pp. 23–37, 2009.

[52] AP News, "US hits Russia with 'war crimes' sanctions, Europe following," https://apnews.com/article/russia-ukraine-kyiv-business-european-commission-united-kingdom-acb86730120a1230b9eb95c3ebdded77, 2012.

[53] Reuters, "ICC judges issue arrest warrant for Putin over war crimes in Ukraine," https://www.reuters.com/world/europe/icc-judges-issue-arrest-warrant-against-putin-over-alleged-war-crimes-2023-03-17/, 2023.

[54] B. Foubert and N. Mitton, "Long-range wireless radio technologies: A survey," *Future internet*, vol. 12, no. 1, p. 13, 2020.

[55] Amazon AWS, "LTE-M," https://docs.aws.amazon.com/whitepapers/latest/implementing-lpwan-solutions-with-aws/lte-m.html, 2022.

[56] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of lpwan technologies for large-scale iot deployment," *ICT express*, vol. 5, no. 1, pp. 1–7, 2019.

[57] IMST GmbH, "High range with LoRa® on worldwide 2.4 GHz band," https://wireless-solutions.de/blog/2020/07/24/im282a-high-range-with-lora-on-worldwide-2-4-ghz-band/, 2020.

[58] IEEE Computer Society, "Ieee standard for low-rate wireless networks," 5 2020, IEEE 802.15.4.

[59] B. Watteyne, Thubert, "On forwarding 6lowpan fragments over a multi-hop ipv6 network," 11 2020, RFC 8930.

[60] The Linux kernel development community, "IEEE 802.15.4 Developer's Guide," https://www.kernel.org/doc/html/latest/networking/ieee802154.html, 2022.

[61] A. Bruniaux, R.-A. Koutsiamanis, G. Z. Papadopoulos, and N. Montavont, "Defragmenting the 6lowpan fragmentation landscape: A performance evaluation," *Sensors*, vol. 21, no. 5, p. 1711, 2021.

[62] P. Gimenez, "Static context header compression and fragmentation (schc) over lorawan," 4 2021, RFC 9011.

[63] H. She, Z. Lu, A. Jantsch, D. Zhou, and L.-R. Zheng, "Analytical evaluation of retransmission schemes in wireless sensor networks," in *VTC Spring 2009-IEEE 69th Vehicular Technology Conference*. IEEE, 2009, pp. 1–5.

[64] J. Iyengar and M. Thomson, "Quic: A udp-based multiplexed and secure transport," Internet Requests for Comments, RFC Editor, RFC 9000, May 2021.

[65] J. Iyengar and I. Swett, "Quic loss detection and congestion control," Internet Requests for Comments, RFC Editor, RFC 9002, May 2021.

[66] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar *et al.*, "The quic transport protocol: Design and internet-scale deployment," in *Proceedings of the conference of the ACM special interest group on data communication*, 2017, pp. 183–196.

[67] Engineering at Meta, "How Facebook is bringing QUIC to billions," https://engineering.fb.com/2020/10/21/networking-traffic/how-facebook-is-bringing-quic-to-billions/, 2022.

[68] P. Watson, "Slipping in the window: Tcp reset attacks," *Presentation at*, 2004.

[69] Ford, Raiciu, Handley, Bonaventure, "Tcp extensions for multipath operation with multiple addresses," 1 2013, RFC 6824.

[70] Ford, Raiciu, Handley, Bonaventure, Paasch, "Tcp extensions for multipath operation with multiple addresses," 3 2020, RFC 8684.

[71] Christoph Paasch, Fabien Duchêne and Gregory Detal, "Multi-Path TCP - Linux Kernel implementation," www.multipath-tcp.org/, 2022.

[72] Apple Support, "Use Multipath TCP to create backup connections for iOS," https://support.apple.com/en-us/HT201373, 2022.

[73] Y. Liu, Y. Ma, Q. D. Coninck, O. Bonaventure, C. Huitema, and M. Kühlewind, "Multipath Extension for QUIC," Internet Engineering Task Force, Internet-Draft draft-ietf-quic-multipath-03, 2022.

[74] Clausen, Jacquet, "Optimized link state routing protocol (OLSR)," 10 2003, RFC 3626.

[75] Clausen, Dearlove, Jacquet, Herberg, "The optimized link state routing protocol version 2," 4 2014, RFC 7181.

[76] R. Kufakunesu, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on adaptive data rate optimization in lorawan: Recent solutions and major challenges," *Sensors*, vol. 20, no. 18, p. 5044, 2020.

[77] M. Conti, J. Willemsen, and B. Crispo, "Providing source location privacy in wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1238–1280, 2013.

[78] R. El-Badry, M. Youssef, and M. Eltoweissy, "Hidden anchor: Providing physical layer location privacy in hybrid wireless sensor networks," in *2009 3rd International Conference on New Technologies, Mobility and Security*. IEEE, 2009, pp. 1–5.

[79] J.-P. Sheu, J.-R. Jiang, and C. Tu, "Anonymous path routing in wireless sensor networks," in *2008 IEEE International Conference on Communications*. IEEE, 2008, pp. 2728–2734.

[80] L. Lightfoot, Y. Li, and J. Ren, "Preserving source-location privacy in wireless sensor network using star routing," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*. IEEE, 2010, pp. 1–5.

[81] M. G. Reed, P. F. Syverson, and D. M. Goldschlag, "Anonymous connections and onion routing," *IEEE Journal on Selected areas in Communications*, vol. 16, no. 4, pp. 482–494, 1998.

[82] A. El Mougy and S. Sameh, "Preserving privacy in wireless sensor networks using onion routing," in *2018 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, 2018, pp. 1–6.

[83] Starlink, "Starlink Specifications," https://www.starlink.com/specifications, 2022.

[84] Raspberry, "Raspberry Pi 3 Model B+," https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/, 2022.

[85] Waveshare, "SX1262 868M LoRa HAT," https://www.waveshare.com/wiki/SX1262_868M_LoRa_HAT, 2022.

[86] Semtech, "LoRa® Reference Design for 2.4GHz," https://www.semtech.com/products/wireless-rf/lora-core/sx1280zxxxxgw1, 2022.

[87] J. Petäjäjärvi, K. Mikhaylov, M. Pettissalo, J. Janhunen, and J. Iinatti, "Performance of a low-power wide-area network based on lora technology: Doppler robustness, scalability, and coverage," *International Journal of Distributed Sensor Networks*, vol. 13, no. 3, p. 1550147717699412, 2017.

[88] M. Swain, D. Zimon, R. Singh, M. F. Hashmi, M. Rashid, and S. Hakak, "Lora-lbo: an experimental analysis of lora link budget optimization in custom build iot test bed for agriculture 4.0," *Agronomy*, vol. 11, no. 5, p. 820, 2021.

[89] data.goc.hk, "Information on Wi-Fi.HK locations," https://data.gov.hk/en-data/dataset/hk-ogcio-ogcio_hp-wi-fi-hk-locations, 2022.

[90] NYC OpenData, "NYC Wi-Fi Hotspot Locations," https://data.cityofnewyork.us/City-Government/NYC-Wi-Fi-Hotspot-Locations/yjub-udmw, 2022.

[91] GovData, "Freifunk Rhein-Neckar," https://www.govdata.de/web/guest/daten/-/details/freifunk-rhein-neckar, 2022.

[92] data.gov.au, "Wireless hotspot locations - Libraries, Parks and Public spaces," https://data.gov.au/dataset/ds-brisbane-17fb3724-ecfc-4802-8f16-62839fb73fc0/details, 2022.

[93] Paris Data, "Paris Wi-Fi - Sites disposant du service," https://parisdata.opendatasoft.com/explore/dataset/sites-disposant-du-service-paris-wi-fi/information/?disjunctive.cp&disjunctive.etat2, 2022.

[94] data.gov.au, "AdelaideFree Wi-Fi Access Point Locations," https://data.gov.au/dataset/ds-sa-8b8040ac-c71c-4c9b-b361-be84b4cd3dc6/details, 2022.

[95] data.europe.eu, "Public access free WiFi," https://data.europa.eu/data/datasets/public-access-free-wifi/?locale=de, 2020.

[96] data.gv.at, "Hotspot - Standorte (Linz)," https://www.data.gv.at/katalog/dataset/stadt-linz_hotspotstandorte, 2022.

[97] T. ANDREW S and W. DAVID J, "Computer networks fifth edition," 2011.

[98] OMNeT++, "OMNeT++ Discrete Event Simulator," https://hub.packtpub.com/iot-forensics-security-connected-world/, 2018.

[99] D. Chaum, "The dining cryptographers problem," *Journal of Cryptology*, vol. 1, pp. 65–75, 1988.

[100] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity," in *Privacy Enhancing Technologies: Second International Workshop, PET 2002 San Francisco, CA, USA, April 14–15, 2002 Revised Papers 2*. Springer, 2003, pp. 41–53.

[101] Jonathan's Space Pages, "Starlink Launch Statistics," https://planet4589.org/space/con/star/stats.html, 2022.

[102] A. Mpitziopoulos and D. Gavalas, "An effective defensive node against jamming attacks in sensor networks," *Security and Communication Networks*, vol. 2, no. 2, pp. 145–163, 2009.

[103] P. D. Pradeep, B. A. Kumar *et al.*, "A survey of emergency communication network architectures," *International Journal of u-and e-Service, Science and Technology*, vol. 8, no. 4, pp. 61–68, 2015.

[104] V. Y. Kishorbhai and N. N. Vasantbhai, "Aon: a survey on emergency communication systems during a catastrophic disaster," *Procedia computer science*, vol. 115, pp. 838–845, 2017.

[105] S. Debnath, W. Arif, S. Roy, S. Baishya, and D. Sen, "A comprehensive survey of emergency communication network and management," *Wireless Personal Communications*, pp. 1–47, 2021.

[106] Y. Gao, J. Cao, P. Wang, J. Yin, M. He, M. Zhao, M. Peng, S. Hu, Y. Sun, J. Wang *et al.*, "Intelligent uav based flexible 5g emergency networks: Field trial and system level results," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 138–143.

[107] Y. Shibata, Y. Sato, N. Ogasawara, and G. Chiba, "A disaster information system by ballooned wireless adhoc network," in *2009 international conference on complex, intelligent and software intensive systems*. IEEE, 2009, pp. 299–304.

[108] H. Suzuki, Y. Kaneko, K. Mase, S. Yamazaki, and H. Makino, "An ad hoc network in the sky, skymesh, for large-scale disaster recovery," in *IEEE vehicular technology conference*. IEEE, 2006, pp. 1–5.

[109] H. Okada, H. Oka, and K. Mase, "Network construction management for emergency communication system skymesh in large scale disaster," in *2012 IEEE Globecom Workshops*. IEEE, 2012, pp. 875–880.

[110] H. Niu, X. Zhao, and J. Li, "3d location and resource allocation optimization for uav-enabled emergency networks under statistical qos constraint," *IEEE Access*, vol. 9, pp. 41 566–41 576, 2021.

Appendix A.
Distance To Origin Evaluation

The data sets with an assumed 5 km maximum range between nodes in Figure 4 (a) shows the limitations of the different data sets. In contrast, very densely populated sets that do not cover a lot of area, like Adelaide or Paris, show no increase in distance. This effect is primarily dependent on the set's covered area. For example, if we roughly draw a circle around the nodes provided by each data set, we get a diameter of ∼5 km for Adelaide and ∼15 km for Paris, while Hong Kong has ∼50 km and Rhein-Neckar even ∼100 km. Data sets, like New York City (∼30 km) or Brisbane (∼20 km) that cover a medium-sized area, show a diminishing return in terms of an increased max_hops. We also found that some sets have more uniformly spread nodes over the covered area, while others have a decreasing density from the center to the borders of the covered area. For example, the more uniformly dense Paris reaches the limit sooner than the unevenly dense Brisbane. With the Adelaide data set, we cannot get past 5 km and when considering the random selection of nodes with a more dense center, our average distance from the origin is naturally quite low.

Figure 4 (b) shows similar measurements for the 1 km range case. We can see the effects analogously to the 5 km results, just with a significantly reduced distance from the origin. Note that some data sets, especially Rhein-Neckar, have severely reduced node count when considering a connected network with a 1 km range (cf. Section 6.2.2). The effect of this can be seen in this graph. Overall, there is an inherent trade-off for setting the max_hops parameter, as setting it higher leads to an increased distance; yet, more hops for the communication will lead to more delays and overhead in the network. Our measurements show that it is crucial to take the underlying spread and density of nodes into account when choosing max_hops. For the following simulation results, we assumed max_hops = 3 for the 5 km range networks and max_hops = 5 for the 1 km range networks.

Appendix B.
Result Graphs

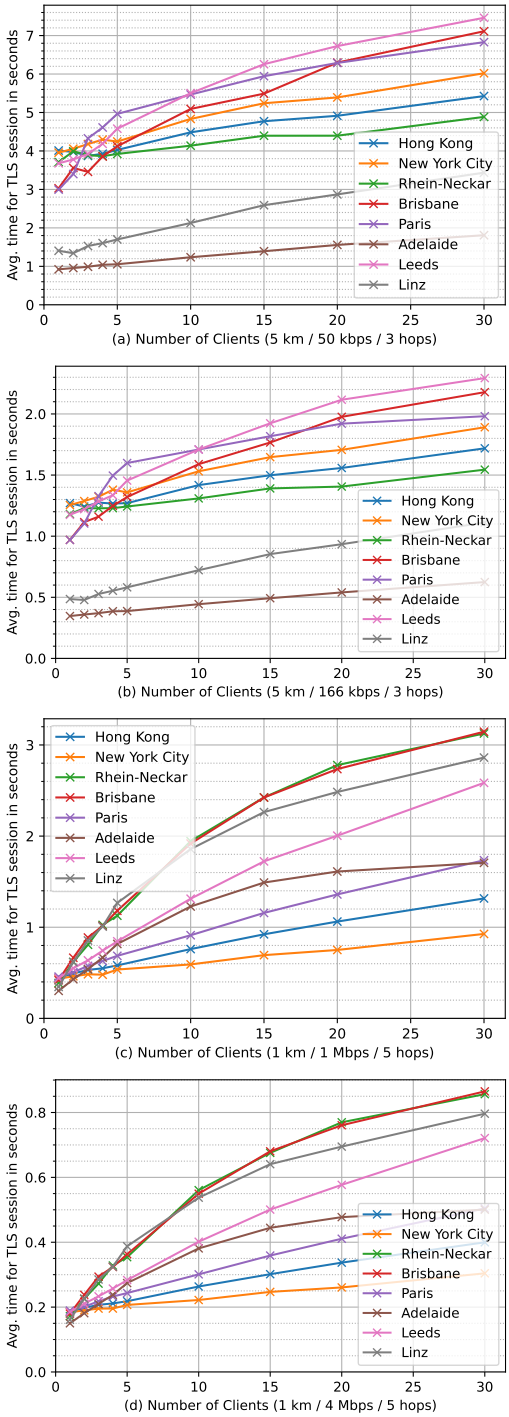

Figure 5. Graphs showing the measurements of TLS session delay in different network types for all data sets.

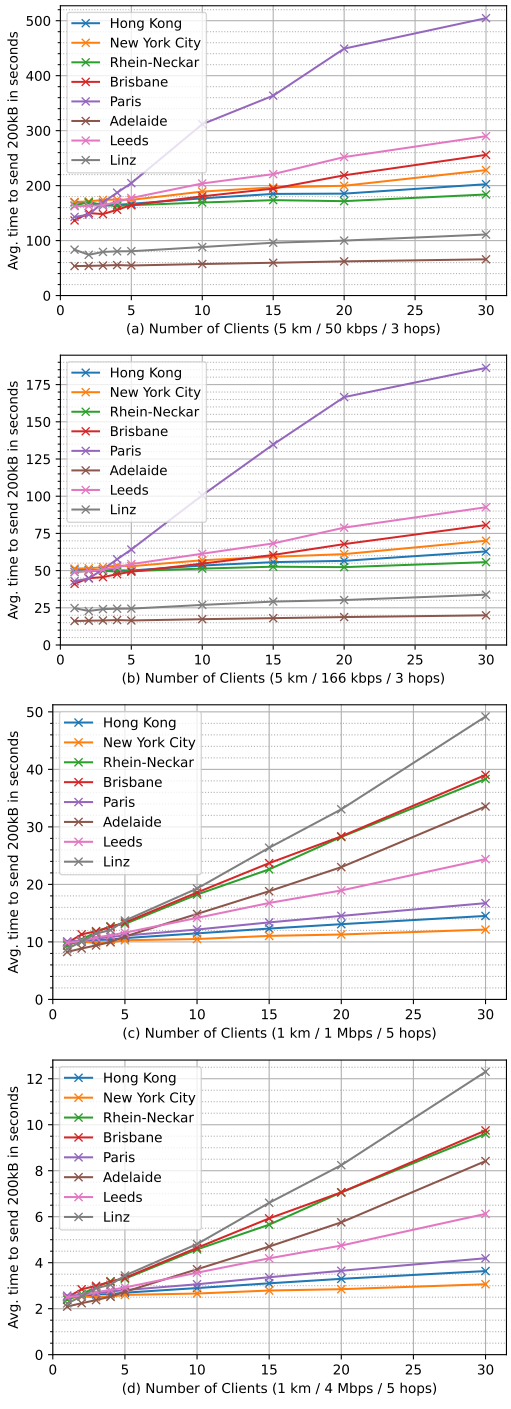

Figure 6. Graphs showing the measurements of length to upload 200 kB image in different network types for all data sets.