Quantifying Unlinkability in Multi-hop Wireless Networks

Victoria Ursula Manfredi[*,a], Cameron Donnay Hill[a]

[a] Department of Mathematics and Computer Science, Wesleyan University, 265 Church St., Middletown, CT, USA 06459

Abstract

Consider a multi-hop wireless network in which devices act as anonymizing routers. Even if devices anonymize their link transmissions, an adversary may still be able to infer key information by observing the traffic patterns in the network. In this work, we quantify how well an adversary can infer unlinkability, that is, the probability that different pairs of devices are communicating, from anonymized link transmissions. We first propose a metric to compute unlinkability using a Kalman-filter based adversary. Using this metric, we then evaluate how different network characteristics impact unlinkability. We assume that devices do not reorder packets to mix traffic and thereby increase unlinkability. Instead, we show that traffic mixing is still possible due to the use of multi-hop routing and broadcast transmissions, with the amount of mixing dependent on the network characteristics. In simulation, we find that i) for unicast links, as network connectivity increases unlinkability decreases, while for broadcast links, as connectivity increases unlinkability increases, ii) link dynamics tend to increase unlinkability with unicast links but decrease unlinkability with broadcast links, iii) well-connected topologies, particularly with broadcast links, achieve the same level of unlinkability with fewer transmissions per packet delivered, iv) a lattice topology has consistently good unlinkability in different scenarios, and v) heterogeneous network traffic gives higher unlinkability and better anonymization efficiency than uniform traffic, even when the average rate of traffic is the same.

Key words: Wireless networks, multi-hop routing, anonymous communication, unlinkability
2010 MSC: 00-01, 99-00

1. Introduction

Rather than relying on fixed infrastructure like Internet routers or cell towers to relay traffic, in a multi-hop wireless network devices relay traffic for each other in a peer-to-peer fashion. Lack of infrastructure not only makes multi-hop wireless networks easier to deploy, it also increases privacy. For instance, devices can avoid communication over infrastructure that may be monitored [1, 2], and users can better control the distribution of their data by ensuring that any collected data is stored locally.

Consider then a multi-hop wireless network in which devices act as anonymizing routers. Even if devices anonymize their link transmissions an adversary may still be able to infer important information by observing the traffic patterns in the network, such as which pairs of devices are communicating. This is problematic since in many multi-hop wireless networks, different devices have different roles (e.g., sources vs. sinks in a sensor network) and some devices are more critical to network functionality (e.g., a military commander) than others. If an adversary can identify such devices it can prevent important information from reaching its destination.

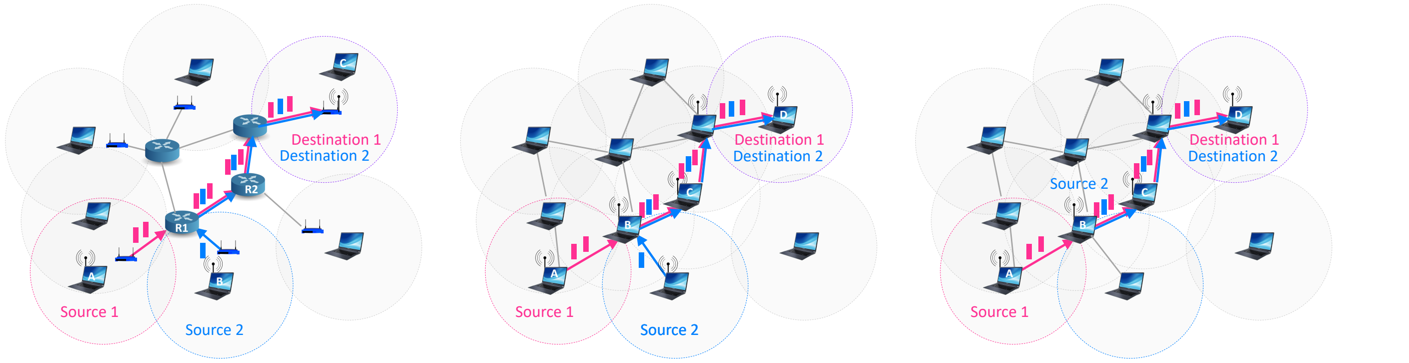
Given this network scenario, our goal is to quantify what impacts how well an adversary can infer *unlinkability* [3], that is, the probability that different pairs of devices are communicating (see §2.1), given the anonymized link transmissions. We assume that the devices in the multi-hop wireless networks we consider do not mix (i.e., reorder) traffic, unlike a mix network [4]. Instead, we hypothesize that traffic mixing is still possible due to the use of multi-hop routing and broadcast transmissions (see Fig. 1 and §2.2). The amount of traffic mixing that is possible should depend on the flows present, the network connectivity, the link dynamics, and the routing strategy. It is these network characteristics whose influence on traffic mixing and thus unlinkability that we investigate in this work.

To quantify unlinkability, we assume a global adversary that passively eavesdrops on the anonymized packet transmissions on each link. The adversary uses these transmissions to compute a probability distribution over the possible communicating pairs of devices. We formulate the adversary as a Kalman filter to compute this distribution and derive an unlinkability metric. We then introduce the idea of *anonymization efficiency* to quantify the efficiency of unlinkable communication in different network scenarios.

In simulation, we confirm that traffic mixing does occur even when devices themselves do not mix traffic. We show that i) for unicast links, as network connectivity increases unlinkability decreases, while for broadcast links, as connectivity increases unlinkability increases, ii) link dynamics tend to increase unlinkability with unicast links but decrease unlinkability with broadcast links, iii) well-connected topologies, particularly with broadcast links, achieve the same level of unlinkabil-

[*] Corresponding author

Email addresses: vumanfredi@wesleyan.edu (Victoria Ursula Manfredi), cdhill@wesleyan.edu (Cameron Donnay Hill)



(a) **Internet routing and flow interleaving.** On the Internet, devices are only either end-hosts or routers, and only end-hosts are sources or destinations of traffic. Flows from different end-hosts may cross at a router, but incoming traffic at the router matches outgoing traffic. E.g., traffic from Source 1 and Source 2 cross at Router R1, but the incoming and outgoing traffic at R1 is the same.

(b) **Multi-hop routing and flow interleaving.** In a multi-hop wireless network, devices are both end-hosts and routers, and so any device may be the source or destination of traffic as well as a router. Like on the Internet, flows from different devices may cross at a device operating as a router. E.g., traffic from Source 1 and Source 2 cross at Device B.

(c) **Multi-hop routing and packet interleaving.** Only with multi-hop routing will packet interleaving happen at a device. For instance, Device B is both a source of packets (for Source 2) and forwarder of packets (for Source 1). Consequently, the incoming packets at Device B are different than the outgoing packets, due to interleaving of Source 1 and Source 2 packets.

Figure 1: Illustration of how multi-hop routing supports packet mixing. A flow is a set of packets sent from a source to a destination over a sequence of links (i.e., path). Two-way communication requires two flows, one in each direction.

ity with fewer transmissions per packet delivered, iv) a lattice topology has consistently good unlinkability in different scenarios, and v) heterogeneous traffic gives higher unlinkability and better anonymization efficiency than uniform traffic, even when the average rate of traffic is the same.

The rest of this paper is structured as follows. In §2, we explain how traffic mixing can happen in multi-hop wireless networks. In §3 we review related work. In §4, we describe our Kalman filter adversary. In §5, we show how we use our Kalman filter adversary to derive an unlinkability metric and propose the idea of anonymization efficiency. In §6, we evaluate our unlinkability metric in simulation. Finally, in §7, we summarize our contributions.

2. Background

2.1. *Computing Unlinkability*

In this work, we focus on multi-hop wireless networks in which devices act as anonymizing routers. To anonymize transmissions, devices re-encrypt [5] packets at the network layer, and set link layer addresses in such a way as to hide the intended next hop of a packet yet still allow this hop to process the packet. We assume devices do not mix traffic, but, as we shall see in §2.2 and quantify in this paper, traffic mixing can still happen.

In the anonymity literature, the adversary's goal is to compute the *unlinkability* of a packet's source with its destination [3]. Unlinkability is also known as relationship or source-destination anonymity. To enable unlinkable communication, Chaum [4] proposed mix nodes that re-order and re-encrypt the messages passing through them to hide the message paths, and the idea of onion routing used in Tor [6], where messages are encrypted multiple times, each layer of encryption corresponding to the next hop to which the message is to be forwarded. In mix networks, mixing of messages at nodes is done to decorrelate input traffic from output traffic. When mixing is not done (e.g., as in Tor to reduce user latency), timing attacks can potentially be used [7, 8] to accurately correlate a message's source with its destination.

In the network tomography literature, the problem of traffic matrix inference [9, 10, 11] is similar to that of unlinkability but does not consider explicit obfuscation of traffic patterns. Additionally, such inference usually considers aggregated traffic, and assumes it is possible to periodically obtain the true traffic matrix at some cost, which is useful for training an inference algorithm.

In this work, we assume a global adversary uses the packet transmissions it passively observes over links to compute a probability distribution, i.e., the *flow distribution*, over the possible flows, see Fig. 1. Because this adversary cannot parse any packet header or payload data it does not know which flows are present. Assuming a passive adversary actually makes our problem harder, not easier, since our goal is to be the adversary and compute unlinkability, rather than to design mechanisms to increase unlinkability.

2.2. *What impacts traffic mixing?*

We assume that the devices in the multi-hop wireless networks we consider do not mix traffic. For instance, if traffic is rare or high delays are problematic, it may be infeasible for devices to wait for sufficient packets in their queues so that the packets can be reordered. Instead, we conjecture that traffic mixing is still possible due to the network features below. Our focus in this work is specifically on the impact of multi-hop routing, broadcast transmissions, network connectivity, link dynamics, and traffic heterogeneity on traffic mixing.

2.2.1. *Multi-hop routing*

Due to multi-hop routing, every device may be the source or destination of a flow, and hence packet, even though that device may also forward packets on flows to or from other devices. Thus, not every packet entering a device will leave it, and every packet leaving a device may or may not have been sourced by the device. We call this packet interleaving, see Fig. 1(c). Because of packet interleaving, the adversary must consider all possible devices along a path as possible sources and destinations of traffic. Flow interleaving, when two flows cross at a device, see Fig. 1(b), increases packet interleaving.

2.2.2. *Wireless links*

The MAC protocol used to access a wireless link typically has a component introducing random delays. Because wireless transmissions interfere, a device may attempt (typically up to 7) re-transmissions of the same packet at random backoff times, interleaved with transmissions from other devices, to cope with collisions. Thus, the dwelling time of packets at devices is variable.

2.2.3. *Broadcast transmissions*

Wireless transmissions may be unicast (unidirectional) or broadcast (omnidirectional). When a wireless device transmits a packet using a broadcast transmission, all devices within range receive the transmission, not just the intended recipient. A receiving device then determines whether it is the intended recipient by checking the packet destination address. Thus, to an outside observer, which neighbor device is the intended recipient may be unclear, assuming no control traffic such as acknowledgements are sent upon receipt.

2.2.4. *Network connectivity*

Well-connected topologies should support higher traffic mixing, due the flow of traffic on the topology itself, and thus unlinkability. In our simulations in §6, we measure network connectivity using *algebraic connectivity,* λ_2, which is defined as the second-smallest eigenvalue of the normalized Laplacian matrix of a graph [12]. The larger the value of λ_2, the more well-connected is the graph.

2.2.5. *Link dynamics*

Due to wireless interference, fading, or mobility the network connectivity may change, and consequently, the pattern of wireless transmissions observed by an adversary may change, even if the underlying set of flows stays the same.

2.2.6. *Multiple packet copies*

Flow correlation attacks typically assume a single copy of a packet. To cope with link dynamics, a multi-hop routing strategy may transmit multiple packet copies.

2.2.7. *Traffic heterogeneity*

Different flows may have different characteristics, such as packet arrival rate, source-destination distance, and duration. The amount of traffic heterogeneity affects how well an adversary is able to infer unlinkability, and may even make it easier to infer that some pairs of devices are communicating, while simultaneously making it harder to infer that other pairs of devices are communicating. For instance, consider a scenario in which one flow has a high arrival rate, while all others have low arrival rates.

3. Related Work

Existing unlinkability metrics [13, 14, 15, 16, 17, 18] are not suitable for our work, as they do not give a straightforward way to compute unlinkability for arbitrary network scenarios or consider multi-hop routing or link dynamics. Other works have designed protocols for unlinkable [19, 20, 21, 22] and anonymous [23, 24, 25, 26, 18] communication for multi-hop wireless networks, but do not give us a way to compute unlinkability. This motivates our derivation of a new metric in Sections 4 and 5 based on a Kalman filter adversary.

Works [27, 28] on inferring unlinkability for anonymous wireless and mobile ad hoc networks correlate and aggregate link layer frames into traffic matrices over time. In comparison, we focus on performing inference at the network layer and build a statistical model that explicitly incorporates adversary knowledge and lets us quantify unlinkability in many different network scenarios.

Work [29] similar in spirit to ours but in mix networks builds a probabilistic model to infer unlinkability using user selected mix path lengths and mixing strategies to compute the model probabilities. Due to computational constraints they focus on smaller static networks and consider up to 10 mixes. In comparison, our Kalman filter model allows us to more directly incorporate different multi-hop routing strategies, as well as consider the impact of different network characteristics, including link dynamics. While we are also affected by computational constraints, we look at networks with up to 25 devices.

In our work, we use a biased random walk routing strategy to limit control overhead and handle topology changes, see §4.1.4. This strategy lets us further quantify the benefits of anonymous broadcast [30] on unlinkability, and the additional impact of multi-hop routing. Other works [25, 31, 32] on anonymous communication have also considered a random walk routing strategy, but here our focus is not to design a new anonymous routing strategy, but to understand how routing randomness impacts unlinkability.

In comparison to works on traffic matrix inference [9, 10, 11], not only do we consider traffic obfuscation and multi-hop routing, we also focus on inferring individual flows in multi-hop wireless networks with potentially dynamic topologies. We also expand on our work in [33] to consider other traffic characteristics here: we evaluate the impact of traffic heterogeneity on unlinkability and add both throughput and queue length analysis. While our model has similarities with the Kalman filter based approach of [11], those authors operate under the assumption that their model can be initialized using the true traffic matrix and instead their goal is to track how traffic in this traffic matrix changes over time.

Works examining the impact of network topology in the context of anonymity primarily focus on identifying which mix topologies are more vulnerable to attack or enable faster mixing [6, 34, 35, 36, 31] or the interplay of mix connectivity with dummy packets for padding [35]. Of particular interest to us are works identifying well-connected topologies like expander graphs [34, 36], and scale-free and small-world topologies [36] as being mix topologies that support efficient mixing in terms of message path lengths. Here, however, our focus is to understand not just the impact of connectivity on unlinkability but also the impact of other network characteristics.

Recent mix network implementations [2] ensure unlinkability even when both the entry and exit nodes are controlled by an adversary, unlike Tor [6]. Other work on mix networks [37] looks at adding noise to protect against traffic analysis. Mix networks as well as the Tor onion routing overlay are typically constructed using end-hosts, which can be both sources and destinations of traffic as well as traffic relays, but these implementations still rely on the Internet to route traffic between relays.

When Tor onion routing is implemented at the network layer [38] and mixes are instead high-speed routers, the mix network comprises only routers. Our model could thus be viewed as a multi-hop wireless network in which every device is both a wireless Tor node operating at the network layer and a possible source or destination of traffic. In our scenarios, though, not only can the number of relays be much larger than the three used in Tor, the next relay to use can change depending on network dynamics and multi-hop routing, while broadcast wireless transmissions further protect against traffic analysis.

4. Kalman Filters for Flow Inference

We now overview how we use a Kalman filter [39, 40] to obtain the flow distribution. Computing the flow distribution is generally a computationally intensive task. The primary reason why we use a Kalman filter to model our states and observations with continuous rather than discrete random variables (like in a hidden Markov model) is to make our computations more efficient. Our goal, however, is not to propose Kalman filters as a real-time adversary for flow inference, but instead make meaningful comparisons of the unlinkability of different network scenarios.

4.1. Kalman Filter

Kalman filters originated in the target tracking literature and assume the true location (state) of a tracked object is unobservable (hidden) and modeled as a Gaussian random variable. Noisy observations of the true state are assumed to be available and are also modeled as a Gaussian random variable. The Kalman filter update equations are thus given as follows.

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{w}_t \tag{1}$$

$$\mathbf{y}_t = \mathbf{B}\mathbf{x}_t + \mathbf{v}_t \tag{2}$$

Let the initial state be $\mathbf{x}_0 \sim \mathcal{N}[\mu, \mathbf{\Sigma}]$. Then the state $\mathbf{x}_{t+1}$ is a linear function of $\mathbf{x}_t$ plus some Gaussian noise $\mathbf{w}_t \sim \mathcal{N}[\mathbf{0}, \mathbf{Q}]$. The observation $\mathbf{y}_t$ is similarly a linear function of the state $\mathbf{x}_t$ plus some Gaussian noise $\mathbf{v}_t \sim \mathcal{N}[\mathbf{0}, \mathbf{R}]$. The transition matrix $\mathbf{A}$ transforms the current state $\mathbf{x}_t$ to the next state $\mathbf{x}_{t+1}$. The observation matrix $\mathbf{B}$ transforms the current state $\mathbf{x}_t$ to the current observation, $\mathbf{y}_t$. When the assumptions of linearity and Gaussian noise are true, the Kalman filter is an optimal estimator of the state.

In the rest of this section we describe how we set-up a Kalman filter to solve the flow inference problem.

4.1.1. States $\mathbf{x}_0$, $\mathbf{x}_t$ and Covariances $\mathbf{\Sigma}$, $\mathbf{Q}$

We model a multi-hop wireless network as a graph, $G = (V, E)$, where $N = |V|$ is the number of devices and E is the set of links. In a network with N devices, there are at most N^2 possible flows including those whose source and destination are the same device. Since which flows are present is unknown, we model all possible flows. We include the possibility of self-flows as this gives more flexibility to the model estimation: for instance, self-flows could correspond to cover traffic or to devices holding onto packets for extended periods of time.

We define the state $\mathbf{x}_t$ to be a vector of length $2N^2$. The first N^2 states represent the total traffic on each of the N^2 possible flows up to time t. The next N^2 states represent the traffic arrivals on each flow. While it would be natural to have the state additionally model the total traffic at each device for each flow at each timestep, we do not do this since it makes inference intractable as the state space size increases to $2N^3$ from $2N^2$.

We set each entry of the initial state vector, μ, to $1/2N^2$. We set the $2N^2 \times 2N^2$ initial covariance matrix, $\mathbf{\Sigma}$, to the identity matrix times 0.1. We set the $2N^2 \times 2N^2$ covariance matrix, $\mathbf{Q}$, to the identity matrix.

4.1.2. Observation $\mathbf{y}_t$ and Covariance $\mathbf{R}$

In a network with N devices, there are at most N^2 possible links including self-links. While we assume which links are present in the network is known, which links exist or have traffic on them may change over time, and so we must model all links. We include self-links as these could model cover transmissions or delaying of transmissions.

We define the observation $\mathbf{y}_t$ to be a vector of length N^2 representing for each link, the total traffic transmitted up to time t. We consider both unicast and broadcast wireless links. For broadcast links, we assume all unicast links incident to a device are activated during packet transmission, and so all dimensions of $\mathbf{y}_t$ corresponding to those links will have traffic on them. We set the $N^2 \times N^2$ observation covariance matrix, $\mathbf{R}$, to the identity matrix.

4.1.3. Transition Matrix, A

The transition matrix $\mathbf{A}$ is of size $2N^2 \times 2N^2$ and maps states from one timestep to the next. We assume traffic from one flow never switches to another flow, and that traffic currently on a flow accumulates over time. We set the entries of $\mathbf{A}$ as follow, where $src(i)$ indicates the source device for flow i and $dst(i)$ indicates the destination device for flow i.

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{if } i = j \\ 1 & \text{if } j = N^2 + i \text{ and } src(i) \neq dst(i) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

The first N^2 elements in $\mathbf{x}_t$ keep track of the total traffic on each flow, while the next N^2 elements in $\mathbf{x}_t$ keep track of the new traffic arrivals on each flow. Intuitively, when $\mathbf{A}$ is multiplied with $\mathbf{x}_t$, the result is the following. The total traffic on each flow in $\mathbf{x}_t$ is multiplied with the diagonal elements of $\mathbf{A}$ while the new traffic on each flow is multiplied with the elements above the main diagonal of $\mathbf{A}$. The results are then summed together, giving the new total traffic on each flow.

4.1.4. Observation Matrix, B

The matrix $\mathbf{B}$ is of size $N^2 \times 2N^2$, where rows are the total traffic sent over each link and columns are the total traffic and arrivals on each flow. We set the entries of $\mathbf{B}$ directly given assumptions about i) the multi-hop routing strategy in use and ii) the network connectivity.

In our simulations in §6, we use a ϕ-randomized routing strategy that forwards packets to the next device on the shortest path (or stays at the current device) with probability ϕ and forwards packets to a random neighbor device with probability $1-\phi$. A packet's path terminates once it reaches its destination. Setting $\phi = 1$ gives shortest path routing and lets us quantify how much unlinkability exists even when devices do not themselves mix traffic. Setting $\phi = 0$ gives a random walk and lets us quantify the unlinkability gained due to randomness in the routing strategy. Essentially the time to deliver a packet is the first passage time from the source to destination for a random walk parameterized by ϕ. If the packet reaches its destination before mixing has occurred, then unlinkability will not be maximized. If the packet reaches its destination after mixing has occurred, then unlinkability will be maximized but will possibly use more transmissions than necessary.

When setting the entries of $\mathbf{B}$, however, the adversary assumes only shortest path routing is used and has no knowledge of ϕ. The adversary does, however, know the true network topology. Let $src(j)$ be the source device of flow j and let $dst(j)$ be the destination device. Let $snd(i)$ be the sending device on link i and let $rcv(i)$ be the receiving device. $Nbr(k)$ indicates the set of neighbor devices for device k and $R\{src : dst\}$ indicates the set of devices comprising the shortest route between a source device, src, and destination device, dst. We then set the entries of $\mathbf{B}$ as follows.

$$\mathbf{B}_{ij} = \begin{cases} 1, & \text{if } snd(i) \neq dst(j),\ rcv(i) \in Nbr(snd(i)), \\ & \text{and } rcv(i) \in R\{src(j) : dst(j)\} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Intuitively, for each possible sending device $snd(i)$, $\mathbf{B}$ gives the next hop receiving device $rcv(i)$ for packets on flow j.

4.2. Flow Inference

The Kalman filter lets us compute the maximum likelihood estimates of the flow state given the observed link transmissions, assuming the linear Gaussian assumptions hold.

Given a Kalman filter with its parameters set for a network scenario and a sequence of observations $\mathbf{y}_{1:T}$, we can recursively compute the probability distribution $P(\mathbf{x}_t|\mathbf{y}_{1:t})$. This distribution remains Gaussian and the computation remains tractable even with many observations. Let $\bar{\mu}_F$ be the mean of this distribution and let μ_F be the vector containing the first N^2 values of $\bar{\mu}_F$, corresponding to the estimates of the total traffic on the N^2 possible flows. We use μ_F to derive a probability distribution over flows, P_F.

We perform one post-processing step: any entries in μ_F that are negative are set to zero, since negative amounts of traffic on a flow are not feasible. We conjecture several reasons for the presence of negative entries. First, from inspecting the values of μ_F, negative entries seem to arise in part for flows that don't exist but that are sub-flows of flows that do exist, and so may capture the removal of traffic at one device and the transfer to another device. Second, our problem formulation is unlikely to strictly satisfy the linear, Gaussian assumptions of the Kalman filter and so negative entries may be a consequence of numerical approximations.

We compute the flow distribution P_F as follows, where $\mu_F(i)$ is the total traffic on flow i and $P_F(i)$ is the probability that the ith flow had traffic.

$$P_F(i) = \frac{\mu_F(i)}{\sum_{j=1}^{N^2} \mu_F(j)} \tag{5}$$

As the network size increases, the probability mass is more finely dispersed over the possible states. We thus renormalize P_F focusing on the most likely states. To identify these states, we sort the probabilities and find the value, min_F, at index $2F$ in the sorted list, where F is the actual number of flows in the network. We set all probabilities less than min_F to zero and renormalize P_F. Knowing F is strictly not necessary and any cutoff point could be used.

5. Quantifying Unlinkability

Regardless of the adversary model, computing unlinkability for a given network scenario is computationally hard, given the large space of possibilities and limited adversary information. Consequently, some kind of probabilistic model is necessary. Here, we describe a new metric based on our Kalman filter adversary.

5.1. Unlinkability Metric

We derive an unlinkability metric, U, by computing the total variation distance between the flow distribution P_F and the true distribution, P_T. Total variation distance has range $[0, 1]$ and so U also has range $[0, 1]$. We use total variation distance rather than the Kullback-Liebler (KL) divergence, because the KL-divergence is not a true metric (e.g., the distance from P_F to P_T could be different than the distance from P_T to P_F), and our goal is a metric we can use to compare unlinkability in many different network scenarios.

$$U = \frac{1}{2}\sum_{i=1}^{N^2}|P_F(i) - P_T(i)| \tag{6}$$

We obtain a bound on the maximum unlinkability, U_{max}, by computing U when P_F is set to the uniform random distribution, but excluding those flows for which the source and destination are the same device. While we considered self-flows in the Kalman filter computation, we do not consider them here since we are interested only in how many "real" flows are correctly inferred. U_{max} can be viewed as a bound on the worst performance of an intelligent adversary, but not the absolute maximum unlinkability achievable, which would be achieved when all probability weight is put on flows that are not present. In our experiments, the U_{max} values are typically in the range of 0.9 to 1. Because U_{max} can be less than one, for clarity, we show the normalized unlinkability in our results computed as follows.

$$U_{norm} = \frac{U}{U_{max}} \tag{7}$$

If $U_{norm} > 1$, this indicates that the adversary's flow inference is worse than uniformly random guessing.

Our use of normalization here is to provide a bound on the performance of our adversary and give additional insight when comparing the adversary's performance in different network scenarios. In practice, normalizing the unlinkability by the performance of a uniform random adversary may not always be useful, and a more intelligent adversary could be used. For instance, if there are few flows in the network, then the adversary's random guessing could be restricted to consider only those devices that forward any traffic for any flow.

5.2. Anonymization Efficiency

A network's characteristics impact both unlinkability and the total link transmissions used to deliver traffic. For instance, while additional transmissions from using a longer path to route traffic from source to destination increases unlinkability, it requires using network capacity above what is minimally required to deliver the traffic over the shortest path. Alternatively, while having every device retransmit every packet over a broadcast link would maximize unlinkability, it would also be inefficient.

We would thus like to quantify the gains in unlinkability at the cost of transmissions. To do this, we introduce a metric we call *anonymization efficiency*, E, computed as follows, where D_{tx} is the total packets transmitted and D_{dv} is the total packets delivered.

$$E = \frac{U_{norm}}{D_{tx}/D_{dv}} \tag{8}$$

When computing E in our simulations, we assume infinite capacities on links and infinite queues at devices. We focus solely on data traffic since the amount of control traffic generated by a routing strategy is strategy specific and a variable and hard to optimize constraint. Instead, we assume that there is no control traffic present, neither to set up routes nor any acknowledgements that might be sent in response to received data packets. It is true that knowledge of control traffic should increase an adversary's ability to accurately estimate the flow distribution and consequently decrease unlinkability. However, our interest in this work is not purely the absolute value of unlinkability or anonymization efficiency, but how unlinkability changes in different network scenarios.

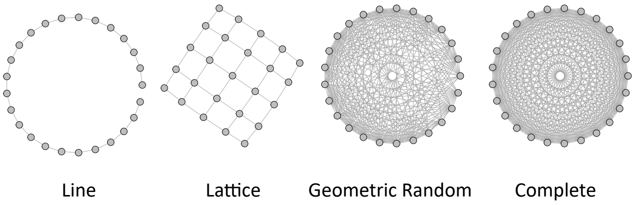


Figure 2: Network topologies used in simulations.

6. Evaluation

Our simulations are done in R and run using the MIT SuperCloud and Lincoln Laboratory Supercomputing Center [41]. We use the FKF (Fast Kalman Filter) package [42] as our Kalman filter implementation. We next describe our simulation set-up and then overview our simulation results.

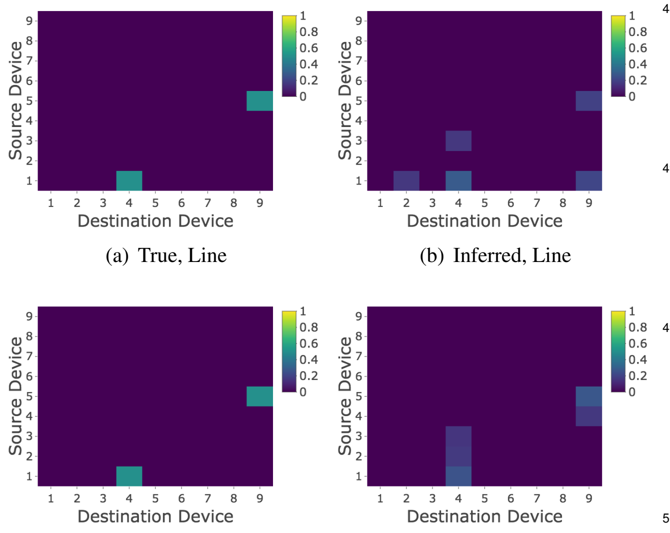
6.1. Methodology

6.1.1. Network topology

We assume the adversary knows the network topology and whether unicast or broadcast links are present. As shown in Fig. 2, we consider four network topologies, with $N = 25$: i) line, ii) 4-degree lattice, iii) geometric random graph, and iv) complete graph. To generate a geometric random graph, points are randomly placed in a unit square. Then any points within a given transmission radius are connected; we use relatively large radii of 0.6 and 0.85 to ensure a connected graph.

6.1.2. Link dynamics

We consider scenarios where devices are stationary but the links present may change. We use a 2-state Markov model for the link dynamics: links are i.i.d. and stay up from one timestep to the next with probability p (and transition from up to down with probability $1 - p$) and stay down with probability q (and transition from down to up with probability $1 - q$). We initialize the up or down state of each possible link according to the steady-state probability that a link is up for the 2-state Markov model, $\pi = (1-q)/(2-p-q)$; on each timestep, we then update the state of each link according to the model. The steady-state probability, π, tells us on average how much time a link spends up, despite any transitions down that might have occurred.



(a) True, Line

(b) Inferred, Line

(c) True, Lattice

(d) Inferred, Lattice

Figure 3: Examples of true (P_T) vs. inferred (P_F) flow distributions in line and lattice topologies for $N = 9$. Results are for static ($p = 1$ and $q = 0$) unicast links, $\phi = 1$, uniform traffic rate $\lambda = 0.5$, and $T = 200$. The y-axes indicate the 9 possible source devices and the x-axes indicate the 9 possible destination devices. The value at a square (x, y) represents the (inferred) proportion of total network traffic sent from source y to destination x. In these plots, traffic comprises 2 flows: from source 1 to destination 4, and from source 5 to destination 9. Using these distributions and Eq. 7, we get $U_{norm} = 0.526$ for the line topology, and $U_{norm} = 0.477$ for the lattice topology.

For both unicast and broadcast links, if a link that was present disappears, no packets can be sent over that link. Since we model a broadcast link as comprising a set of unicast links, we assume the adversary can tell when any of the individual unicast links disappears. We also assume, however, that the adversary does not know the probabilities with which links change state.

6.1.3. Routing Strategy

We use the ϕ randomized multi-hop routing strategy that we introduced in §4.1.4. The adversary knows that a shortest path-based routing strategy is being used but does not know the value of ϕ. If an estimate of ϕ were known to the adversary, this could be accounted for by changing how $\mathbf{B}$ is set. Note that the link dynamics do not change the shortest paths in the network and so do not affect $\mathbf{B}$. This is because links are i.i.d and which links will be up or down over time cannot be predicted.

6.1.4. Medium access control

We assume discrete time and that the duration of a timestep is long enough for every device in the network to transmit one packet.

6.1.5. Traffic generation

We randomly choose F flows with replacement and simulate the flows for T timesteps. We assume no control traffic is generated. We consider two traffic models: i) uniform traffic and ii) heterogeneous traffic. To generate uniform traffic, we model the number of data packets that arrive on each flow using a Poisson process with rate λ. In this model, while the traffic generated on each flow is random, the rate of traffic is the same for all flows. To generate heterogeneous traffic, we again model the number of data packets that arrive on each flow using a Poisson process but now the arrival rate on average for all flows is $\bar{\lambda}$, and we uniformly randomly assign each flow a rate in the range $[0, 2\bar{\lambda}]$.

We assume that the adversary does not know the packet arrival rates, λ and $\bar{\lambda}$, nor the number of flows present, nor which subsets of devices comprise sources or destinations of flows. If such information were known to the adversary, it could be used to improve how the initial state of the Kalman filter model is set.

6.1.6. Adversary model.

We assume that the adversary is able to start observing a network when it is “turned on”, so that the adversary sees all initial traffic on all flows. But we also assume that the adversary does not necessarily observe when the network is turned off, so that it is possible that the adversary does not observe some packets delivered for some flows. Equivalently, we could assume that the adversary starts observing a network after it has been in operation for some time, but is then able to observe all traffic until the network is turned off.

When an adversary is not able to observe all packets delivered for a flow (or is not able to observe the initial forwards of some packets on a flow), these incompletely delivered packets will potentially serve as a kind of cover traffic, due to the partially travelled paths that they add to the adversary’s observations. Similarly, the extra packets sent due to increasing routing randomness, ϕ, can also be considered as cover traffic. In §6.2.7 and §6.2.8, which respectively look at the throughput and queue lengths for the main network scenarios that we consider, we explore how and when not yet delivered packets can serve as cover traffic.

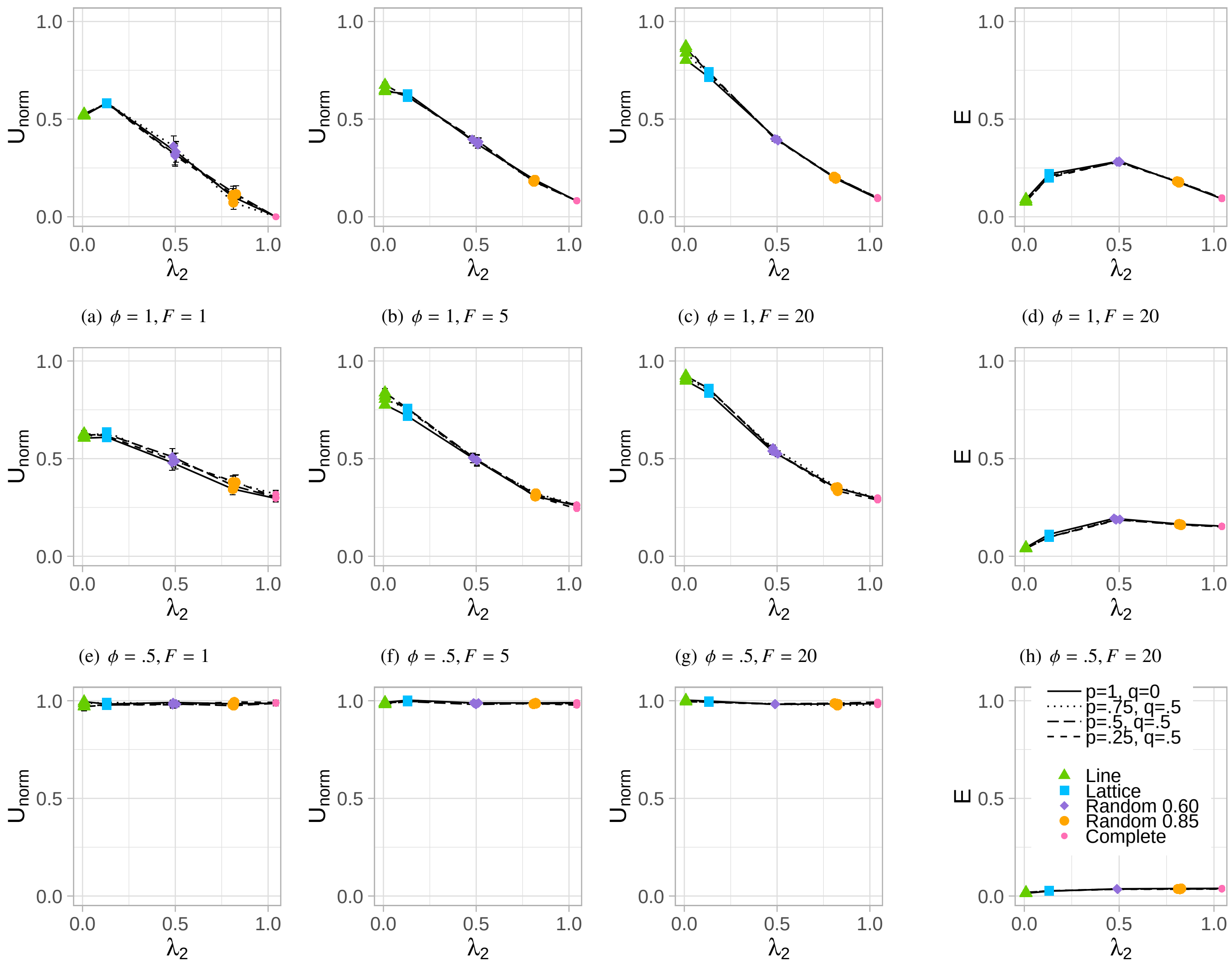
6.2. What impacts unlinkability?

Our results are summarized in Figs. 3 to 8. Fig. 3 shows examples of true and inferred flow distributions and the associated U_{norm} values to provide intuition about how we compute unlinkability. Figs. 4 and 5 show how unlinkability changes for different network scenarios. Fig. 6 shows the throughput for the corresponding scenarios in Figs. 4 and 5, while Fig. 7 shows unlinkability as a function of queue length for these and other scenarios. Finally, while Figs. 3 to 7 focus on network scenarios with uniform traffic, Fig. 8 shows how unlinkability changes when traffic is heterogeneous.

6.2.1. Impact of number of flows

Figs. 4 and 5 plot unlinkability as a function of algebraic connectivity, λ_2. Each simulation is executed for $T = 200$ timesteps. For each topology, we run 100 simulations, choosing different sets of F flows randomly with replacement. We show 95% confidence intervals, with the points colored according to the associated network topology. Different lines indicate different settings for the probability that links stay up, p, or down, q.

In Fig. 4, for unicast links, uniform traffic rate $\lambda = 0.1$, and no link dynamics (i.e., the solid line where $p = 1, q = 0$),



(a) $\phi = 1, F = 1$ (b) $\phi = 1, F = 5$ (c) $\phi = 1, F = 20$ (d) $\phi = 1, F = 20$

(e) $\phi = .5, F = 1$ (f) $\phi = .5, F = 5$ (g) $\phi = .5, F = 20$ (h) $\phi = .5, F = 20$

(i) $\phi = 0, F = 1$ (j) $\phi = 0, F = 5$ (k) $\phi = 0, F = 20$ (l) $\phi = 0, F = 20$

Figure 4: Network scenarios with unicast links and uniform traffic rate of $\lambda = 0.1$. Plots show unlinkability, U_{norm}, and anonymization efficiency, E, as a function of algebraic connectivity, λ_2, for $N = 25$.

we see that for a given value of routing randomness ϕ, as the number of flows, F, increases, unlinkability increases. This is because having more flows provides more opportunities for flows to cross paths. In Fig. 5, for broadcast links, we see similar behaviour.

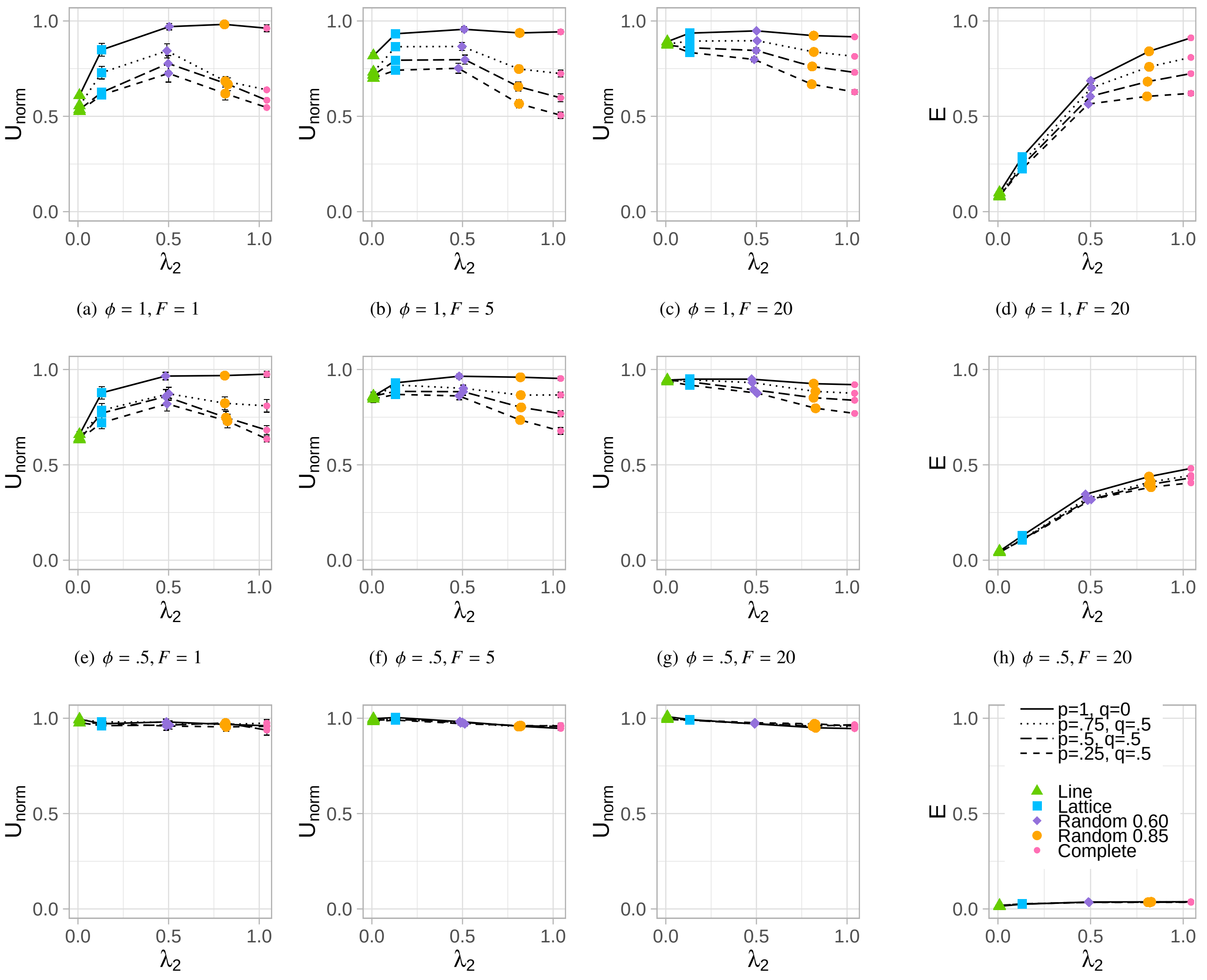
It is important to note that, for $F = 1$, unicast links, and $\phi = 1$, that for all topologies except the complete graph, unlinkability *is not* 0, even though no explicit attempt is made to increase traffic mixing. This is due to multi-hop routing: the adversary must consider that the possible intermediate hops on the path between a potential source and destination might themselves be potential sources and destinations. With unicast links, unlinkability is greater than zero as long as there are shortest paths in the topology that are more than 1 hop long. In the case of the complete graph topology, the unlinkability *is* 0 because all shortest paths are only 1 hop long, so there are no potential intermediate sources and destinations, and so no mixing.

6.2.2. *Impact of routing randomness*

In Figs. 4 and 5, as routing randomness increases (that is, ϕ decreases from 1 to 0), unlinkability generally increases, regardless of the number of flows, link type, or topology. This is in part because all possible sub-paths along a path must be considered as potential flows due to multi-hop routing. For instance, with increased routing randomness, it takes longer for any packet to reach its destination, since the packet must pass through more intermediate devices that must be considered potential sources and destinations. Regardless of link type, but depending on the number of flows and network topology, less routing randomness is needed to achieve the same level of unlinkability.

6.2.3. *Impact of link type*

Comparing Figs. 4 and 5 shows that unicast scenarios generally have lower unlinkability than broadcast scenarios. For



(a) $\phi = 1, F = 1$

(b) $\phi = 1, F = 5$

(c) $\phi = 1, F = 20$

(d) $\phi = 1, F = 20$

(e) $\phi = .5, F = 1$

(f) $\phi = .5, F = 5$

(g) $\phi = .5, F = 20$

(h) $\phi = .5, F = 20$

(i) $\phi = 0, F = 1$

(j) $\phi = 0, F = 5$

(k) $\phi = 0, F = 20$

(l) $\phi = 0, F = 20$

Figure 5: Network scenarios with broadcast links and uniform traffic rate of $\lambda = 0.1$. Plots show unlinkability, U_{norm}, and anonymization efficiency, E, as a function of algebraic connectivity, λ_2, for $N = 25$.

$\phi = 0$, i.e., maximum routing randomness, there is little difference in unlinkability between unicast vs. broadcast links.

However, for each link type there is a split based on network connectivity. For lower connectivity topologies like the line and lattice, regardless of whether unicast or broadcast links are used, high unlinkability is possible only when there are many randomly chosen flows or routing is via a highly random walk. For high connectivity topologies like the geometric random graph or complete graph, high unlinkability is only possible when either broadcast links are used or routing is via a highly random walk.

Thus, broadcast transmissions are primarily beneficial when the topology is well-connected. If the topology is not well-connected then broadcast is not sufficient by itself: instead, many randomly chosen flows or very random walks are needed. While other work has shown the power of broadcast transmissions [30], here we see with multi-hop routing that the benefits of broadcast are dependent on additional characteristics of the network scenario.

6.2.4. *Impact of network topology*

In Fig. 4, for unicast links and no link dynamics (i.e., $p = 1, q = 0$), as algebraic connectivity λ_2 increases, unlinkability generally decreases except when $\phi = 0$. This is because when connectivity is higher, paths are shorter, and so there are fewer devices potentially involved in flows, and so there are fewer possible flows to consider which lowers unlinkability. When connectivity is lower, so paths are longer, there are more devices potentially involved in flows which means there are more flows to consider which increases unlinkability. Note that for the geometric random topologies, the mean λ_2 is plotted, since each simulation is for a different randomly generated topology.

Conversely, in Fig. 5, for broadcast links and no link dynamics, as λ_2 increases, unlinkability increases except when $\phi = 0$ or $F = 20$. Now, each broadcast increases the number of devices that receive a transmission, thus increasing the number of possible flows that must be considered. For networks with higher connectivity, each broadcast reaches more devices, increasing unlinkability. For networks with lower connectivity, even though each broadcast reaches fewer devices, flows must use longer paths, which means more devices are still reachable and must be considered in possible flows, thereby increasing unlinkability.

6.2.5. Impact of link dynamics

The dotted and dashed lines in Figs. 4 and 5 are for scenarios when links are dynamically changing.

In Fig. 4, using the uniform traffic rate of $\lambda = 0.1$, we see that unicast link dynamics have minimal impact on unlinkability. In other experiments that we have done (see [33] as well other results not shown), we have used the much higher uniform traffic rates of $\lambda = 0.5$ and 0.75, and observed that as unicast link dynamics increase, the result is somewhat higher unlinkability compared to when there are no link dynamics, with somewhat more significant increases when connectivity is low. Essentially, when the network is sufficiently sparsely connected or there is sufficient network traffic, then unicast link dynamics may prevent a device from making a transmission, and as a result, can change the probabilities that packets are destined to that device or its neighbors. Thus, for unicast links, link dynamics do increase unlinkability, but only when there is sufficient network congestion, either due to traffic or sparseness of the network topology.

Conversely, in Fig. 5, broadcast link dynamics result in smaller decreases in unlinkability (relative to static links) when connectivity is low and larger decreases in unlinkability (relative to static links) when connectivity is high. Essentially, when a given device makes repeated broadcast transmissions over time, if different subsets of the component unicast links to the device's neighbors are down due to link dynamics, then this can give information about which neighbor a transmission is intended for. When the network is sparsely connected (and there is sufficient traffic), the broadcast scenario becomes similar to the unicast scenario, with link dynamics potentially preventing a device from making any transmission.

6.2.6. Anonymization efficiency

The last columns of Figs. 4 and 5 plot anonymization efficiency as a function of algebraic connectivity, λ_2, for unicast and broadcast links respectively. We show only the anonymization efficiency results for $F = 20$ since the $F = 1$ and $F = 5$ results are very similar.

In Fig. 4 for unicast links, anonymization efficiency is highest for the lattice and random graph topologies, except when routing randomness is maximized with $\phi = 0$. The lattice, however, achieves significantly higher unlinkability than do the random graph topologies. We conjecture that in terms of efficiency, the lattice best trades-off having paths that are not too short so that intermediate hops must be considered as potential sources and destinations, with having paths that are not too long and thereby incurring too many transmissions.

In Fig. 5 for broadcast links, anonymization efficiency increases as λ_2 increases, except when routing randomness is maximized with $\phi = 0$. Generally, anonymization efficiency is higher for broadcast links than for unicast links, except when $\phi = 0$.

6.2.7. Throughput

The partially travelled paths of packets that have not yet been delivered can potentially be viewed as generating cover traffic and thereby increasing unlinkability. To understand if and when this occurs, Fig. 6 shows the percentage of packets delivered for the different scenarios in Figs. 4 and 5. Because unicast or broadcast link type only affects what the adversary observes, not which devices store and forward packets, Fig. 6 shows only the results for the unicast link scenarios, as the broadcast link results are essentially identical.

Consider first those scenarios in Fig. 6 in which there is no routing randomness (i.e., $\phi = 1$) and all packets are delivered. In Figs. 6 (a) to (c) this is true for all but the line topology. Despite this, we still observe in Figs. 4 (a) to (c) and 5 (a) to (c) that unlinkability can be very different for different scenarios. Hence, even without cover traffic, unlinkability is not necessarily zero and depends on the network topology, traffic flows, and link type, combined with the use of multi-hop routing.

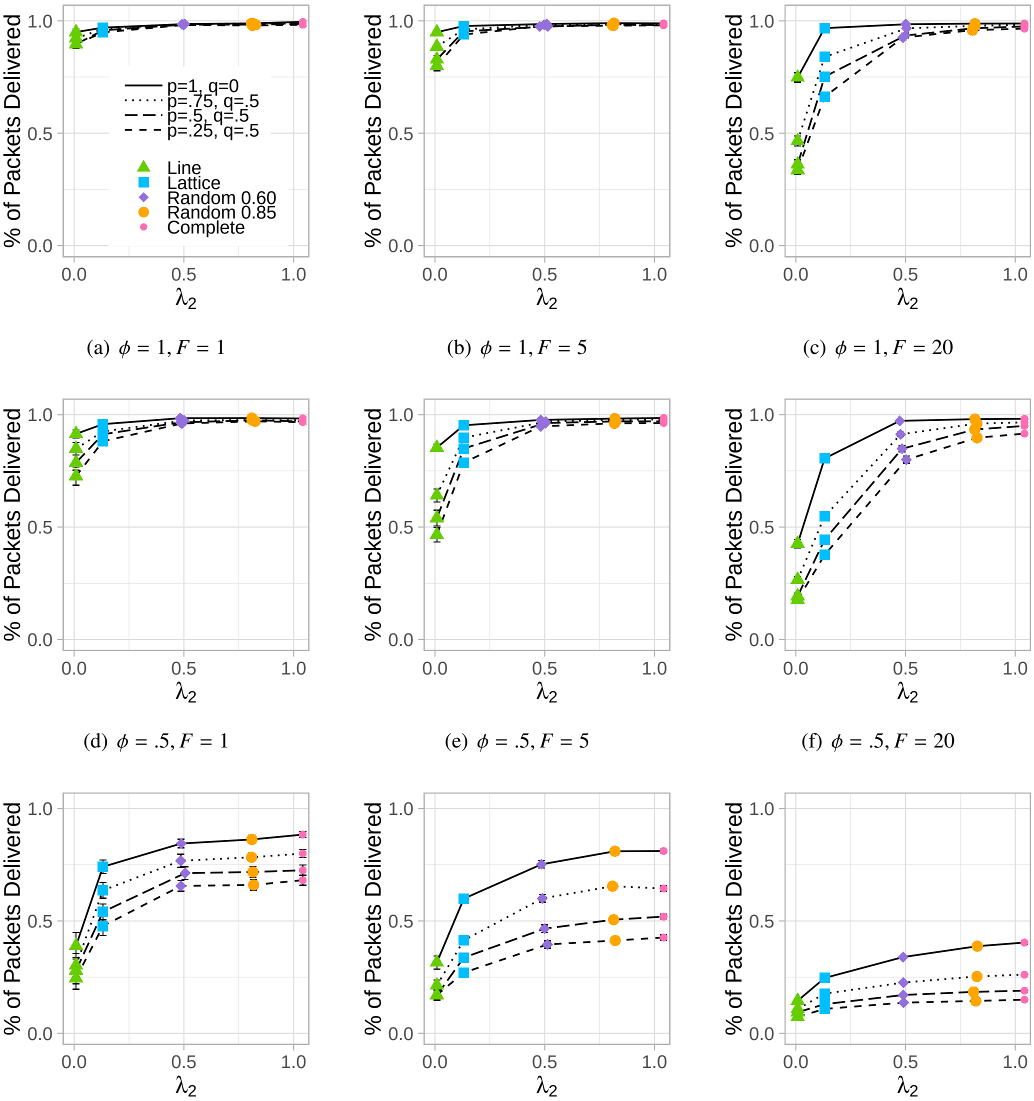
Now consider those scenarios in Fig. 6 for which there is some routing random randomness (i.e., $\phi = 0.5$) and all packets are delivered. In Figs. 6 (d) to (f) this is true for all but the line and lattice topologies. The increase in unlinkability we observe in Figs. 4 (e) to (g) and Figs. 5 (e) to (g) we can thus attribute purely to the increased routing randomness contributing "cover."

Finally, consider those scenarios in Fig. 6 for which there is maximal random randomness (i.e., $\phi = 0$): for this scenario, due to the increased traffic congestion, not all packets can be delivered, and the percentage of packets delivered varies significantly depending on the number of flows, topology, and link dynamics, all of which impact congestion. The increase in unlinkability we observe in Figs. 4(i) to (k) and 5 (i) to (k) for all of these scenarios we can thus attribute to not just routing randomness contributing cover but also potentially due to incompletely observed packet paths.

6.2.8. Queue lengths

A common feature of mix networks [4] is to wait for a sufficient number of packets to arrive at a device, so that they can be reordered. A further benefit of this waiting is that it introduces delays between when a packet arrives at a device and when it leaves. Here, we explore the possibility that longer queue lengths (due to more congestion), indirectly increase unlinkability due to the delays that they introduce, as well as provide more opportunities for mixing due to multi-hop routing.

In Fig. 7, we plot unlinkability as a function of average queue length. The left column of Fig. 7 shows results for unicast links while the right column of Fig. 7 shows results for broadcast



(a) $\phi = 1, F = 1$

(b) $\phi = 1, F = 5$

(c) $\phi = 1, F = 20$

(d) $\phi = .5, F = 1$

(e) $\phi = .5, F = 5$

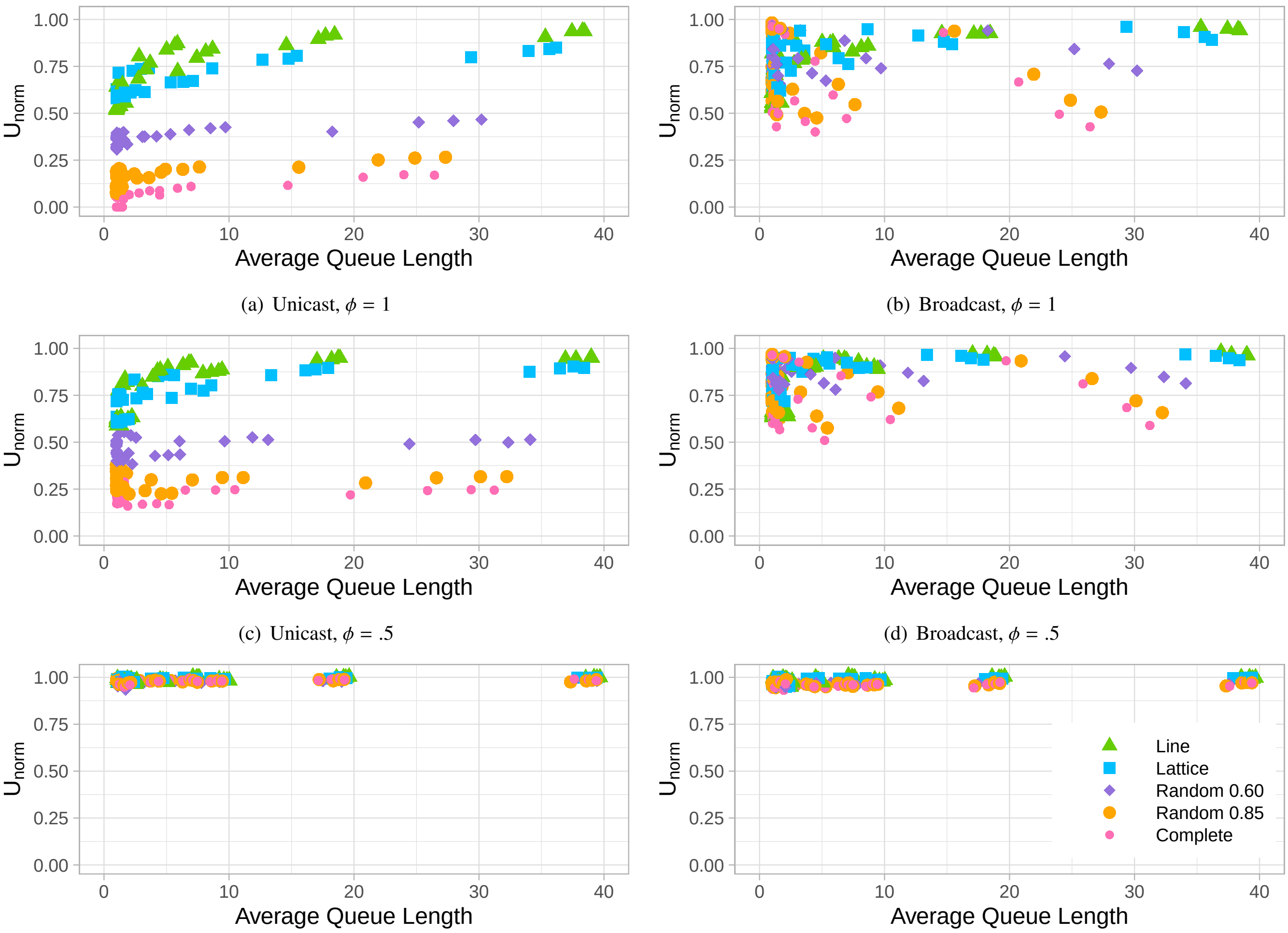
(f) $\phi = .5, F = 20$

(g) $\phi = 0, F = 1$

(h) $\phi = 0, F = 5$

(i) $\phi = 0, F = 20$

Figure 6: Network scenarios with unicast links and uniform traffic rate of $\lambda = 0.1$. Plots show the percentage of packets delivered as a function of algebraic connectivity, λ_2, for $N = 25$. Because link type only affects what the adversary observes, not which devices ultimately store and forward packets, the corresponding plots for broadcast links are essentially identical to these plots and thus not shown.



(a) Unicast, $\phi = 1$

(b) Broadcast, $\phi = 1$

(c) Unicast, $\phi = .5$

(d) Broadcast, $\phi = .5$

(e) Unicast, $\phi = .0$

(f) Broadcast, $\phi = .0$

Figure 7: Plots show unlinkability U_{norm} as a function of average queue length for $N = 25$. Each plot overlays points for all flow values $F = 1, 5, 20$, link dynamics p and q, and uniform traffic rates, $\lambda = 0.1, 0.25, 0.5$.

links. To get a better sampling of the space, in each plot we overlay points for all flow values ($F = 1, 5, 20$), link dynamics ($p = 1, 0.75, 0.5, 0.25$ and $q = 0, 0.5$) and uniform traffic rates ($\lambda = 0.1, 0.25, 0.5$).

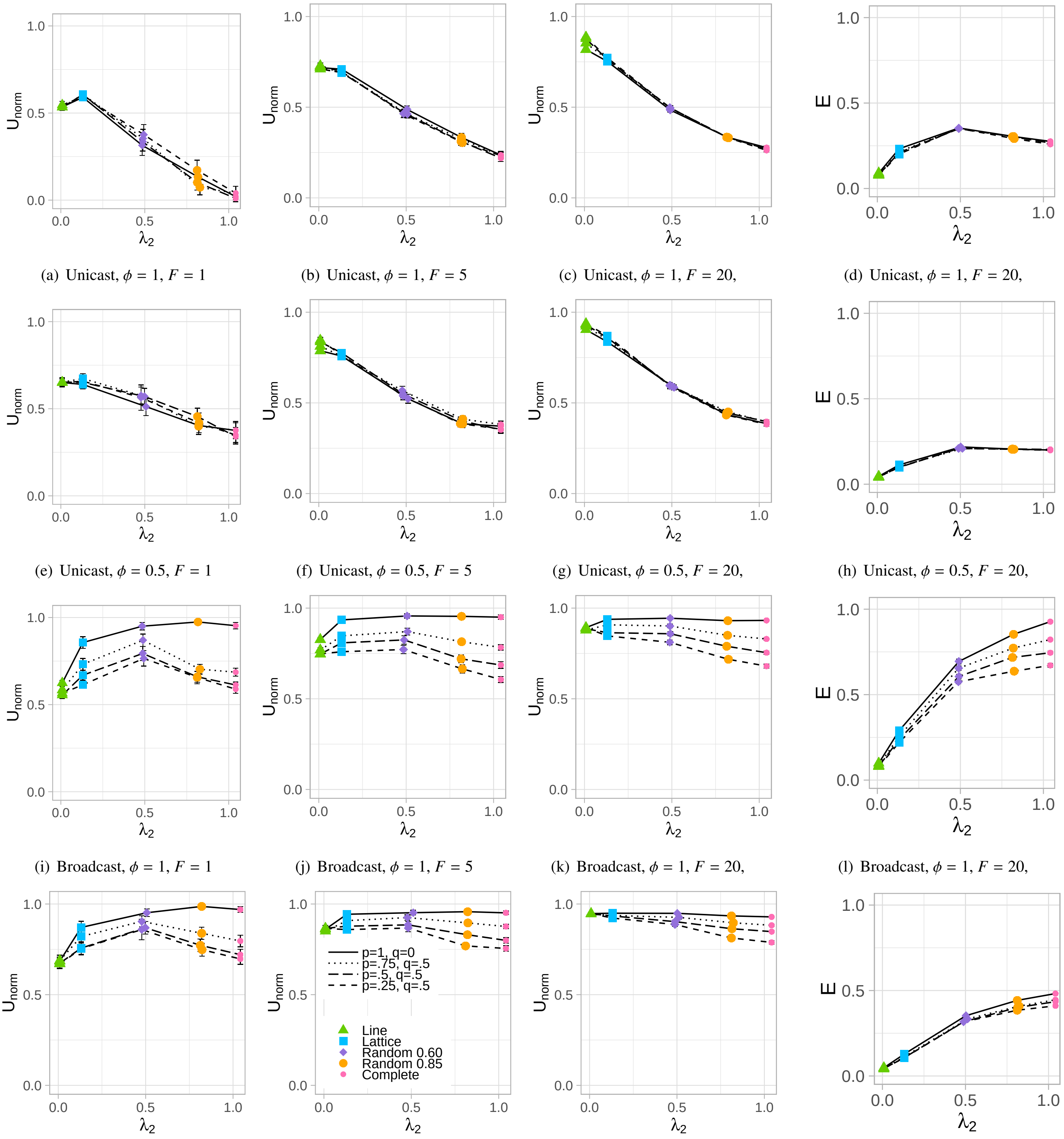
First, consider the unicast link scenarios in the left column of Fig. 7. Except for when routing randomness is maximized ($\phi = 0$), we see a clear trend: as queue length increases, unlinkability increases. And those topologies that are most affected by congestion, due to their sparse connectivity, the line and the lattice, show the largest gains in unlinkability as average queue length increases.

Now, consider the broadcast link scenarios in the right column of Fig. 7. For the line and lattice points, we again see generally increasing gains in unlinkability as average queue length increases. For the other three topologies, however, this is not the case. A more careful examination of these results indicate that the topologies for which unlinkability significantly decreases as average queue length increases are those for which the topology is well-connected and the addition of link dynamics ($p \neq 1$ and $q \neq 0$) thus decreases connectivity and unlinkability.

For both unicast and broadcast links in Fig. 7, however, we observe that even when the queue length is zero, unlinkability is frequently not zero, and so, like in §6.2.7, it is not purely traffic congestion that is increasing unlinkability. We also observe that most of the gains in unlinkability as a function of queue size happen with small queue sizes, such as going from empty queues to queues with one or a few packets in them. Once the network is too congested, there are likely limited unlinkability gains from sending more traffic. Instead, a more promising approach to increase unlinkability would be to change the characteristics of the traffic flows themselves.

6.2.9. *Impact of heterogeneous traffic*

So far, our simulations have focused on uniform traffic scenarios, in which all flows generate traffic at the same rate λ. We now consider heterogeneous traffic scenarios, in which flows on average generate traffic at rate $\bar{\lambda}$ but individual flows draw rates from the range $[0, 2\lambda]$. Fig. 8 shows unlinkability as a function of algebraic connectivity for $\bar{\lambda} = 0.1$. We do not show results



(a) Unicast, $\phi = 1$, $F = 1$

(b) Unicast, $\phi = 1$, $F = 5$

(c) Unicast, $\phi = 1$, $F = 20$,

(d) Unicast, $\phi = 1$, $F = 20$,

(e) Unicast, $\phi = 0.5$, $F = 1$

(f) Unicast, $\phi = 0.5$, $F = 5$

(g) Unicast, $\phi = 0.5$, $F = 20$,

(h) Unicast, $\phi = 0.5$, $F = 20$,

(i) Broadcast, $\phi = 1$, $F = 1$

(j) Broadcast, $\phi = 1$, $F = 5$

(k) Broadcast, $\phi = 1$, $F = 20$,

(l) Broadcast, $\phi = 1$, $F = 20$,

(m) Broadcast, $\phi = 0.5$, $F = 1$

(n) Broadcast, $\phi = 0.5$, $F = 5$

(o) Broadcast, $\phi = 0.5$, $F = 20$,

(p) Broadcast, $\phi = 0.5$, $F = 20$,

Figure 8: Network scenarios with heterogeneous traffic rate of $\bar{\lambda} = 0.1$. Plots show unlinkability, U_{norm}, and anonymization efficiency, E, as a function of algebraic connectivity, λ_2, for $N = 25$.

for $\phi = 0$ as unlinkability is close to one for all scenarios.

Figs. 8(a) to (h) focus on unicast links and heterogeneous traffic. Compared with Figs. 4(a) to (h) which uses a uniform traffic rate of $\lambda = 0.1$ and unicast links, we see that a heterogeneous traffic rate of $\bar{\lambda} = 0.1$ does increase unlinkability and anonymization efficiency, except when there is only one flow. When $F = 1$, we see somewhat noisier results than in Fig. 4, as would be expected due to the higher variability in the traffic on any given flow.

Figs. 8(i) to (p) focus on broadcast links and heterogeneous traffic. Compared with Figs. 5(a) to (h) which uses a uniform traffic rate of $\lambda = 0.1$ and broadcast links, we again see that heterogeneous traffic does increase unlinkability, even when the average amount of traffic is the same.

While not shown, the percentage of packets delivered for heterogeneous traffic is essentially the same as for uniform traffic (shown in Fig. 6). This indicates that it is primarily the distribution of the traffic, rather than some consequence of increased congestion that is responsible for the increased unlinkability when traffic is heterogeneous rather than uniform.

6.3. *Discussion*

Our results in §6.2 confirm that traffic mixing is possible from multi-hop routing even when devices themselves do not reorder traffic. Our results also give insight into how best to control the network structure on which unlinkable communication protocols might run. That is, depending on the network characteristics, it may not always be necessary to delay and mix traffic at devices in order to increase unlinkability. We next discuss these insights in more detail.

6.3.1. *How to design unlinkable networks?*

We can divide the network characteristics we consider into two groups: i) characteristics of the network topology, such as connectivity, link type, and link dynamics, and ii) characteristics of the network traffic, such as routing randomness and traffic diversity (which is influenced by the number of flows and the heterogeneity of packet arrival rates on those flows). The interplay between the network topology and the network traffic together give rise to the traffic patterns from which unlinkability is computed. There is thus some flexibility in choosing how to maximize unlinkability in any given network depending on which characteristics we are able to vary.

Network connectivity. Our results show that the lattice most consistently supports high unlinkability (with U_{norm} never less than about 0.6) regardless of the other network characteristics. This suggests that when it is possible to control network connectivity, a lattice is a good target topology when other network conditions are unknown.

Link type. Our results show significant differences between unicast and broadcast links. Sparsely connected topologies improve unlinkability in unicast scenarios but degrade unlinkability in broadcast scenarios. This suggests that depending on the link type, controlling the topology to be more or less sparse may be beneficial.

Link dynamics. Our results show small improvements in unlinkability in sparsely connected topologies for both unicast and broadcast scenarios, and large decreases in unlinkability for well-connected topologies for broadcast scenarios. This suggests that artificial link dynamics could be beneficial in some limited sparse scenarios if that can be achieved without introducing too much congestion, while additional mechanisms to increase unlinkability should be incorporated in well-connected broadcast scenarios. It would be worthwhile, however, to explore the impact of other kinds of link dynamics, particularly those due to device mobility. For instance, delay tolerant networks have broadcast links that are mostly down (large q and small p link dynamics), and frequently use multi-copy protocols for forwarding packets, and so might be expected to have good unlinkability.

We would expect that the impact of link quality on unlinkability would be similar to what we found for link dynamics. In our simulations, we explored the impact of one particular kind of link dynamics on unlinkability, by using a 2-state Markov model. For this Markov model, the steady-state probability that a link is up is given by π (and down is given by $1 - \pi$), defined in §6.1.2. For instance, for the $p = 0.75$ and $q = 0.5$ simulations, we have $\pi = 0.67$, indicating that taking link dynamics into consideration, links are up 67% of the time and down 33% of the time. Thus, we can view our 2-state Markov model as also a model of link quality, where links are up with probability π and down with probability $1 - \pi$.

Routing randomness. Our results show that routing randomization was consistently helpful at increasing unlinkability. In practice, for scenarios with low unlinkability, rather than routing all flows by (mostly) random walks there may be benefits to a more fine-grained approach. For instance, when there are few flows, routing can be done by a random walk. As the number of flows increases, only a subset of flows need be routed randomly. An alternative approach would be to add an additional set of cover traffic flows to the network that are long-lived and routed randomly, with the number of cover traffic flows changing as some function of the number of real flows in the network. Initial experiments (not shown) indicated that the addition of a random walk flow can increase unlinkability, but only when the network is not already overwhelmed with traffic: i.e., there must be sufficient network capacity for random walk packets to be forwarded in the network, and not just waiting in queues. This is true more generally: if the network is already overwhelmed with traffic, then increasing the number of flows will not increase unlinkability.

Traffic diversity. We found that increasing traffic diversity was helpful at increasing unlinkability, even when the amount of traffic and number of flows stayed the same and routing randomness was fixed. This suggests that being able to quantify the amount of traffic diversity at any given time may be beneficial, as periods of low traffic diversity could be identified, and artificial traffic diversity injected into the network, such as by delaying packets to artificially increase traffic burstiness, or by adding additional cover flows with more diverse traffic arrivals.

6.3.2. Which network scenarios are most challenging?

We found that the most challenging network topologies were those at the extremes of connectivity. Lines or very sparsely connected topologies are challenging because of their susceptibility to traffic congestion, making it hard to increase unlinkability significantly without overwhelming the network with traffic. Possible congestion points in the network topology, however, allow the possibility of opportunistically reordering of packets in queues when queue lengths happen to be greater than zero. Conversely, complete graphs or very well-connected topologies are challenging, particularly for unicast or dynamic links, because packets are able to take very short paths to reach the destination, reducing opportunities for the network to itself mix and re-order traffic. These short paths, however, mean more traffic can be accommodated in the network overall, allowing more cover traffic to be used to increase unlinkability.

We found that the most challenging traffic scenarios were those with few flows. But as with well-connected topologies, this also provides an opportunity. When there are few flows, there is more likely to be available bandwidth to support cover traffic, such as via routing randomness or artificially injected traffic diversity. We hypothesize that there is some minimum amount of total traffic and traffic diversity necessary to achieve a given amount of unlinkability.

In §6.2, our most challenging network scenarios overall were those that combined challenging topologies with challenging traffic, such as a unicast fully-connected network with a single flow, or a broadcast line network with a single flow. For both of these scenarios, the small amounts of traffic present in the network should allow for the use of cover traffic to increase unlinkability. Other potentially challenging scenarios are networks that transition between very different topologies or very different traffic scenarios, as might be found in a mobile network.

6.4. Scalability

We chose to model states and observations as multivariate Gaussian random variables to reduce the number of dimensions. The Kalman filter implementation we use, FKF [42], was chosen for its ability to work with large state spaces. The programming language R, however, itself has a maximum vector length and array dimension limit of $2^{31} - 1$. Experimentally, we have found that the largest networks for which we have been able to construct a Kalman filter and simulate before hitting this limit have been for $N = 64$ devices. For $N = 64$, the $\mathbf{A}$ matrix is of size 8192×8192 and cannot be represented sparsely due to the Kalman filter computations. Since we require $18,000$ simulations to obtain the data to make our plots (from 5 topologies times 3 values of ϕ times 3 values of F times 4 sets of p and q values times 100 simulations for statistical significance). Thus, due to the memory and simulation time required to run simulations with larger N values, even using cloud resources, the largest N value that we simulate in this work is $N = 25$.

Our goal, however, is not an algorithm to run in real-time for large networks, but instead to quantify what impacts unlinkability in multi-hop wireless networks which are typically smaller in size. While we consider small networks, they still give insight, such as how to prevent poor unlinkability subnetworks in large networks.

7. Conclusions

In this work, we have quantified the unlinkability achievable when traffic mixing is due to multi-hop routing and broadcast transmissions, rather than mixing at individual devices. To do this, we formulated a Kalman filter adversary who passively observes all packet transmissions that occur in a multi-hop wireless network in which devices also act as anonymizing routers. The adversary uses these transmissions to compute a probability distribution over the possible flows present in the network. From this flow distribution we derived an unlinkability metric that we analyzed in simulation. We showed that i) for unicast links, as network connectivity increases, unlinkability decreases since less traffic mixing is possible, while for broadcast links as connectivity increases unlinkability increases, ii) link dynamics tend to increase unlinkability with unicast links but decrease unlinkability with broadcast links, iii) well-connected topologies, particularly with broadcast links, are able to achieve the same level of unlinkability with fewer transmissions per packet delivered, iv) a lattice topology has consistently good unlinkability in different network scenarios, and v) heterogeneous traffic gives higher unlinkability and better anonymization efficiency than uniform traffic, even when the average rate of traffic is the same.

In future work, we would like to scale our simulations by either approximating the Kalman filter state space or using approximate inference methods. The applicability of non-linear Kalman filters as well as particle filters would also merit investigation. We would also like to explore the impact of more realistic physical and link layers as well as mobility models, which would enable analysis and inference of performance metrics like throughput and delay. Finally, we would like to consider adversaries that may only have partial information about the network topology, as well as active adversaries who are able to intelligently jam transmissions to decrease unlinkability. Our long-term goal is to devise unlinkable communication protocols using our insights.

8. Acknowledgements

The authors are grateful to Danny Krizanc for many helpful discussions, and thank Amir Herzberg and Bing Wang for helpful feedback on the paper. The authors also thank the anonymous reviewers for their helpful comments. The authors acknowledge the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing HPC and consultation re-

sources that have contributed to the research results reported within this paper.

References

[1] Open Garden, Firechat Messaging App, https://en.wikipedia.org/wiki/FireChat (2019).

[2] Validity Labs, HOPR Messaging App, https://hopr.network/ (2020).

[3] A. Pfitzmann, M. Hansen, Terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management, internet draft (expired), https://tools.ietf.org/id/draft-hansen-privacy-terminology-00.html (2010).

[4] D. L. Chaum, Untraceable electronic mail, return addresses, and digital pseudonyms, Commun. ACM 24 (2) (Feb. 1981).

[5] P. Golle, M. Jakobsson, A. Juels, P. Syverson, Universal re-encryption for mixnets, in: Topics in Cryptology – CT-RSA 2004, 2004, pp. 163–178.

[6] R. Dingledine, V. Shmatikov, P. F. Syverson, Synchronous batching: From cascades to free routes, in: PETS, Vol. 4, Springer, 2004.

[7] B. N. Levine, M. K. Reiter, C. Wang, M. Wright, Timing attacks in low-latency mix systems, in: International Conference on Financial Cryptography, Springer, 2004, pp. 251–265.

[8] Y. Zhu, X. Fu, B. Graham, R. Bettati, W. Zhao, Correlation-based traffic analysis attacks on anonymity networks, IEEE Transactions on Parallel and Distributed Systems 21 (7) (2009) 954–967.

[9] Y. Vardi, Network tomography: Estimating source-destination traffic intensities from link data, Journal of the American statistical association 91 (433) (1996) 365–377.

[10] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, C. Diot, Traffic matrix estimation: Existing techniques and new directions, in: ACM SIGCOMM Computer Communication Review, Vol. 32, ACM, 2002, pp. 161–174.

[11] A. Soule, K. Salamatian, A. Nucci, N. Taft, Traffic matrix tracking using kalman filters, ACM SIGMETRICS Performance Evaluation Review 33 (3) (2005) 24–31.

[12] F. R. Chung, Lectures on spectral graph theory, CBMS Lectures, Fresno 6 (1996) 17–21.

[13] S. Köpsell, S. Steinbrecher, Modeling unlinkability, in: Proceedings of the Third Workshop on Privacy Enhancing Technologies, 2003.

[14] V. Shmatikov, M.-H. Wang, Measuring relationship anonymity in mix networks, in: Proceedings of the 5th ACM workshop on Privacy in electronic society, ACM, 2006, pp. 59–62.

[15] L. Fischer, S. Katzenbeisser, C. Eckert, Measuring unlinkability revisited, in: ACM workshop on Privacy in the electronic society, 2008.

[16] D. Huang, Unlinkability measure for ieee 802.11 based manets, IEEE Transactions on Wireless Communications 7 (3) (2008) 1025–1034.

[17] M. E. G. Moe, Quantification of anonymity for mobile ad hoc networks, Electronic Notes in Theoretical Computer Science 244 (2009) 95–107.

[18] V. Mohanty, D. Moliya, C. Hota, M. Rajarajan, Secure anonymous routing for manets using distributed dynamic random path selection, in: Pacific-Asia Workshop on Intelligence and Security Informatics, Springer, 2010, pp. 65–72.

[19] C. Rackoff, D. Simon, Cryptographic defense against traffic analysis, in: STOC, 1993.

[20] A. Beimel, S. Dolev, Buses for anonymous message delivery., Journal of Cryptology 16 (1) (2003).

[21] T. Hayajneh, R. Doomun, P. Krishnamurthy, D. Tipper, Source destination obfuscation in wireless ad hoc networks, Security and Communication Networks 4 (8) (2011) 888–901.

[22] R. Berman, A. Fiat, M. Gomulkiewicz, M. Konowski, M. Kutylowski, T. Levinboim, A. Ta-Shma, Provable unlinkability against traffic analysis with low message overhead, Journal of Cryptology 28 (2015) 623–640.

[23] J. Kong, X. Hong, Anodr: anonymous on demand routing with untraceable routes for mobile ad-hoc networks, in: ACM International symposium on Mobile ad hoc networking & computing, 2003.

[24] S. Seys, B. Preneel, ARM: Anonymous routing protocol for mobile ad hoc networks, in: International Conference on Advanced Information Networking and Applications-Volume 1 (AINA'06), 2006.

[25] J. Deng, R. Han, S. Mishra, Decorrelating wireless sensor network traffic to inhibit traffic analysis attacks, Pervasive and Mobile Computing (2006).

[26] M. Blaze, J. Ioannidis, A. D. Keromytis, T. G. Malkin, A. Rubin, Anonymity in wireless broadcast networks (2009).

[27] Y. Liu, R. Zhang, J. Shi, Y. Zhang, Traffic inference in anonymous manets, in: IEEE SECON, 2010.

[28] Y. Qin, D. Huang, B. Li, STARS: A statistical traffic pattern discovery system for manets, IEEE Transactions on Dependable and Secure Computing 11 (2013).

[29] C. Troncoso, G. Danezis, The bayesian traffic analysis of mix networks, in: ACM conference on Computer and communications security, ACM, 2009, pp. 369–379.

[30] F. Stajano, R. Anderson, The cocaine auction protocol: On the power of anonymous broadcast, in: International Workshop on Information Hiding, 1999.

[31] P. Mittal, N. Borisov, Shadowwalker: peer-to-peer anonymous communication using redundant structured topologies, in: ACM conference on Computer and communications security, 2009.

[32] P. Mittal, M. Wright, N. Borisov, Pisces: Anonymous communication using social networks, arXiv preprint arXiv:1208.6326 (2012).

[33] V. Manfredi, C. Donnay Hill, Quantifying unlinkability in multi-hop wireless networks, in: Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, 2020, pp. 73–82.

[34] G. Danezis, Mix-networks with restricted routes, in: International Workshop on Privacy Enhancing Technologies, 2003, pp. 1–17.

[35] C. Diaz, S. Murdoch, C. Troncoso, Impact of network topology on anonymity and overhead in low-latency anonymity networks, in: Privacy Enhancing Technologies, 2010.

[36] S. Nagaraja, Anonymity in the wild: Mixes on unstructured networks, in: International workshop on privacy enhancing technologies, 2007, pp. 254–271.

[37] J. Van Den Hooff, D. Lazar, M. Zaharia, N. Zeldovich, Vuvuzela: Scalable private messaging resistant to traffic analysis, in: Symposium on Operating Systems Principles, 2015.

[38] C. Chen, D. E. Asoni, D. Barrera, G. Danezis, A. Perrig, Hornet: High-speed onion routing at the network layer, in: ACM SIGSAC Conference on Computer and Communications Security, 2015.

[39] R. E. Kalman, A new approach to linear filtering and prediction problems, Transactions of the ASME–Journal of Basic Engineering 82 (Series D) (1960) 35–45.

[40] G. Welch, G. Bishop, An introduction to the Kalman filter, Tech. Rep. TR95-041, U of North Carolina at Chapel Hill, Dept. of Computer Science (1995).

[41] A. Reuther, J. Kepner, C. Byun, S. Samsi, W. Arcand, D. Bestor, B. Bergeron, V. Gadepally, M. Houle, M. Hubbell, et al., Interactive supercomputing on 40,000 cores for machine learning and data analysis, in: 2018 IEEE High Performance extreme Computing Conference (HPEC), IEEE, 2018, pp. 1–6.

[42] D. Luethi, P. Erb, S. Otziger, FKF: Fast Kalman Filter. R package version 0.1.5, https://cran.r-project.org/web/packages/FKF/index.html (2018).