From Human Mobility to Data Mobility: Leveraging Spatiotemporal History in Device-to-Device Information Diffusion

Jonas Michel and Christine Julien
The University of Texas at Austin
Email: {jonasrmichel c.julien}@utexas.edu

***Abstract*—Supporting mobile applications in densely populated environments requires connecting mobile users and their devices with the surrounding digital landscape. Device-to-device communications (without the support of an infrastructure) will play a critical role in facilitating transparent access to proximate digital resources. A variety of approaches exist to support device-to-device dissemination, yet very few capitalize on the *contextual history* of disseminated data itself to distribute additional data. We introduce a fully decentralized model that captures the causal history of shared information across a lifetime of device-to-device propagation. Our approach enhances each transmitted message with a *spatiotemporal trajectory*, a portion of the message's contextual history, which provides a window into the time-varying state of the network and the overall spreading behavior of the message. We benchmark the performance of spatiotemporal trajectories at a global level and demonstrate the practical utility of our constructs for making intelligent distributed routing decisions through two use cases. The first illustrates how locally-available contextual history may be used to make inferences about remote nodes' knowledge and thus direct routing of future messages. The second example uses trajectories from different types of data to identify commonly co-located data and devices, which can then be used as substitutable routing targets.**

I. Introduction

The sheer density of connected devices in everyday environments is rapidly growing. Supporting emerging mobile applications in these very dense deployments requires connecting devices and their users to *hyper*-localized information. Such data is intimately attached to a very specific space and time. For many reasons, it is becoming apparent that pure device-to-device interactions will play a pivotal role in connecting users to this data. There are numerous mechanisms to support the nuts and bolts of device-to-device message exchange, even over multiple hops, and some are even "content-aware" (i.e., they distribute data based on its own semantics). However, few approaches tap into the *contextual history* of the shared data. We propose an approach whose key tenet is that knowing the context in which a piece of data was created and the contexts in which it has been shared over its lifetime can help determine the future contexts in which that data might be relevant.

Consider a scenario that illustrates the utility of such contextual history. At an outdoor festival with thousands of attendees, festival-goers wish to find nearby mobile vendors of particular food items, be alerted when friends are near, discover popular photos of recent performances, receive coupons from vendors, and be alerted about special events and limited offerings. In such a densely populated environment, connecting attendees' devices with the desired information may require device-to-device interactions: the demand for fixed infrastructure resources (e.g., cellular upload bandwidth) could greatly exceed available capacity [27]; round trip times to the "cloud" may be too slow [26]; excessive data costs or quotas may limit devices' abilities to use a fixed network; and devices' battery constraints may make short-range interactions far more advantageous than long-range ones [28]. As a message is opportunistically propagated between festival goers' devices it accumulates a contextual history comprising causal trajectories of the device-to-device paths it traverses over space and time. For example, as a user's query for "*vendors selling craft beer*" is disseminated using an epidemic protocol, we can represent the query's complete contextual history as a partially ordered list of the timestamped pairs of devices the query traverses. Knowing this history for one or more messages gives insight into the evolution of the time-varying network topology and indicates *how* messages tied to particular data have moved. Instead of capturing the mobility of devices or humans (as in existing approaches), these trajectories capture the mobility of data. While this movement depends on opportunities for message exchange governed by devices' contact patterns, it adds important and semantically rich information that can influence various aspects of information diffusion in device-to-device networks. Consider the following two examples, which show the potential for the use of these trajectories:

- A beer vendor distributing digital coupons for craft beer may use an algorithm that relies on trajectory information about previously disseminated coupons to infer where in the network his coupons have already reached, subsequently using network resources to target new areas of the network.
- A pretzel vendor may use overheard trajectory information to determine that beer and pretzel interest often overlap and subsequently decide to route his pretzel coupons towards beer interests, using the latter as a routing substitute for (potentially unknown) pretzel interests.

Similar scenarios are found in other domains (e.g., vehicular networks [18], smart cities [4], and emergency response [8]).

Problem Statement: The primary design challenge for device-to-device mobile applications is operating under the intermittency of connections that result from device mobility

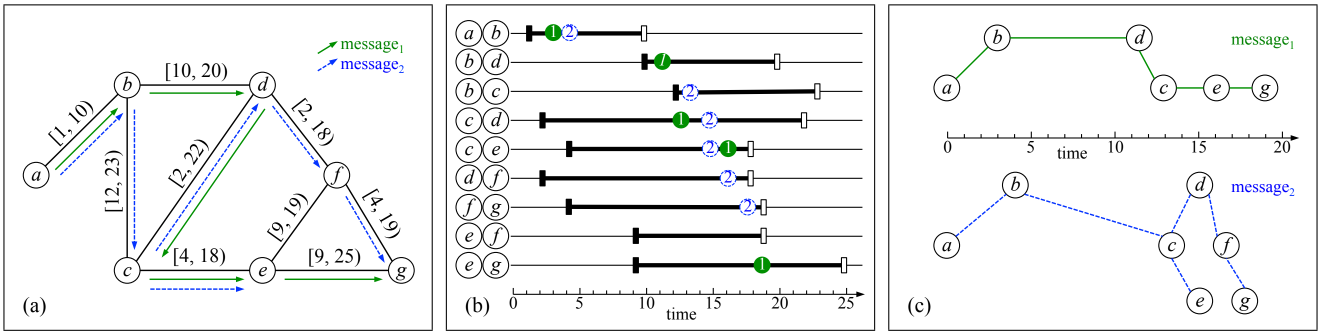


Fig. 1. Transmission graphs. Panel (a) shows an interval graph representing an opportunistic mobile network. Each solid black edge indicates 1-hop network proximity and is labeled with the time interval in which the contact occurred; solid green and dashed blue arrows indicate transmission of message$_1$ and message$_2$ (respectively). Panel (b) gives a more temporally-emphasized picture of the system, where a line corresponds to an edge in (a) and the active interval is bold. The position of a numbered circle in an interval indicates the reception time of a message over that connection. The transmission networks of the two messages are illustrated in (c) where a node's position along the x-axis indicates the reception time.

while building a completely distributed model. Formal and empirical studies of human mobility (e.g., [3], [22]) have driven the development of a wide spectrum of protocols that provide reliable and efficient dissemination of information within a dynamic network topology. While most of these approaches focus on human mobility (or the mobility of devices humans carry), we instead focus on *data mobility*, i.e., time-varying spatial and temporal properties that characterize the movement or spreading of a piece of data within an opportunistic network of mobile nodes. Analysis of real world human trajectory data sets has been exceptionally useful for understanding humans' spatiotemporal dynamics and interactions. Data trajectories, on the other hand, are subject to an entirely different set of barriers (e.g., physical, social, security) and can be very different from human paths. For example, consider Fig. 1, which visualizes the device-to-device propagation of two messages (indicated by the solid green and dashed blue lines) through a transmission graph representation [25] of an opportunistic network of mobile devices. The time-varying proximities of the devices, shown in Figs. 1a and 1b, are a product of (among other things) human mobility; as the humans carrying these devices move, the devices come into contact with one another. However, the causal trajectories of the two messages, shown in Fig. 1c, while enabled by device proximity and contact patterns, extend beyond the bounds of any single device's mobility—data mobility is rather a product of device interactions. In Fig. 1, for example, nodes a and g never come into physical contact (e.g., as a result of physical, social, or temporal boundaries), however both messages originating at a are eventually delivered to g. Harnessing the inferential value of a message's contextual lifetime directly within a highly mobile environment requires a distributed approach that can express the causal structure of message propagation.

Contributions: We introduce a completely distributed model that uses a novel *spatiotemporal trajectory* data type to capture a contextual history of device-to-device exchanges of data. Our model provides two complementary views on data mobility: a data-dependent view (i.e., what data is about) and a data-agnostic view (i.e., how data moves). For example, the transmission network in Fig. 1a may be viewed in a data-dependent fashion in terms of message$_1$'s trajectory (solid green lines), message$_2$'s trajectory (dashed blue lines), or a union of both. Alternatively, the network's time-varying edges may be considered in a data-agnostic view irrespective of the particular data that traversed those edges. Enriching transmitted data with its contextual history gives a mobile application local insight into the overall spreading behavior of shared data and the time-varying topology of the network. We benchmark the performance of our model's constructs at a global level using two empirical data sets. Finally, through two use cases, we demonstrate the practical utility of our model in improving opportunistic routing performance.

II. Model & Terminology

Our model is based on the *time-varying graph* (TVG) [2], which we extend to support spatiotemporal trajectories. While a graph is a natural fit for representing a fixed network, a *time-varying* graph is a natural means for representing data in highly-dynamic and infrastructure-less networks.

Assumptions. Our model relies exclusively on device-to-device communication. We assume nodes have unique identifiers and can detect the appearance and disappearance of incident edges; we assume these links are managed by an underlying discovery protocol. We also assume reliable message delivery provided by the underlying link protocol. In the remainder of the paper, therefore, we assume no lost messages, i.e., if node u believes it sent a message to node v, this message was received by v. The assumption of the use of device-to-device links is not itself unrealistic, especially in light of everyday users' acceptance of emerging technologies that support these links (e.g., BLE, WiFi-Direct) and applications that use them (e.g., FireChat[1]). The reliance on devices to contribute resources to forward messages for others requires incentives [29]; these are orthogonal to our model, and we assume that applications will either build in explicit (e.g., monetary) incentives or rely on implicit ones (e.g., the benefit to an the individual for using the application).

[1] http://www.opengarden.com

Time-Varying Graph. We represent a network of mobile nodes as a *time-varying graph* (TVG) $\mathcal{G} = (V, E, \mathcal{T}, \rho, \zeta, M)$. *V* is a set of mobile nodes making contact with each other over the *lifetime* ($\mathcal{T} \subseteq \mathbb{T}$) of the network. The temporal domain $\mathbb{T}$ corresponds to $\mathbb{R}^+$ (continuous-time). $E \subseteq V^2$ is the set of intermittently available (undirected) edges defined by the contact between nodes; $(x, y) \in E \Leftrightarrow x$ and *y* are in contact at least once in $\mathcal{T}$. For simplicity, we use undirected edges, presuming that two nodes in contact can both transmit to and receive from one another. In a real network this assumption may not hold (i.e., links may not be symmetric). Our model easily generalizes to a directed graph; the temporal domain creates its own level of direction. The *presence function*, $\rho : E \times \mathcal{T} \to \{0, 1\}$, indicates whether a given edge is available at a given time, and the *latency function*, $\zeta : E \times \mathcal{T} \to \mathbb{T}$, indicates the time it takes to propagate a message over a given edge at a given time. To simplify presentation, we use the constant ζ to indicate the latency for all edges and times. To signify the time-dependent availability of edge *e*, we use $\rho_{[t_1,t_2)}(e) = 1$, which indicates that $\forall t \in [t_1, t_2), \rho(e, t) = 1$. For example, in Fig. 2, $\rho_{[1,3)}(ab) = 1$. So far, this is the same definition of TVG as in [2]. Our model enhances this TVG by adding *M*, a set of triples of the form (m, e, t), where each triple signifies the transmission of message m on edge $e \in E$ at time t; to account for transmission times, all $(m, e, t) \in M$ must satisfy $\rho_{[t,t+\zeta)}(e) = 1$.

In our model, each node maintains a local TVG in which the identities the durations of each neighbor's contact are stored as *first-hand* knowledge. Each transmitted message carries the message's *contextual history*, which is extracted and locally stored by a receiving node as *second-hand* knowledge. We present the semantics of this contextual history later in this section. Local TVG maintenance is driven by operations triggered by the appearance and disappearance of incident edges and the reception of messages containing contextual history metadata. We take an entirely distributed approach; the global TVG $\mathcal{G}$ only exists in a virtual sense as a union of the local TVGs maintained by the nodes: $\mathcal{G} = \cup_{v \in V} \mathcal{G}_v$. Before we provide the details of the algorithm that maintains this first- and second-hand knowledge, we introduce the key terminology and fundamental concepts.

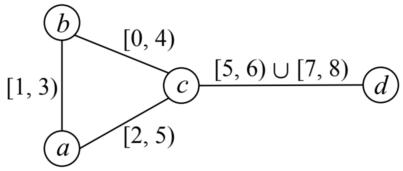


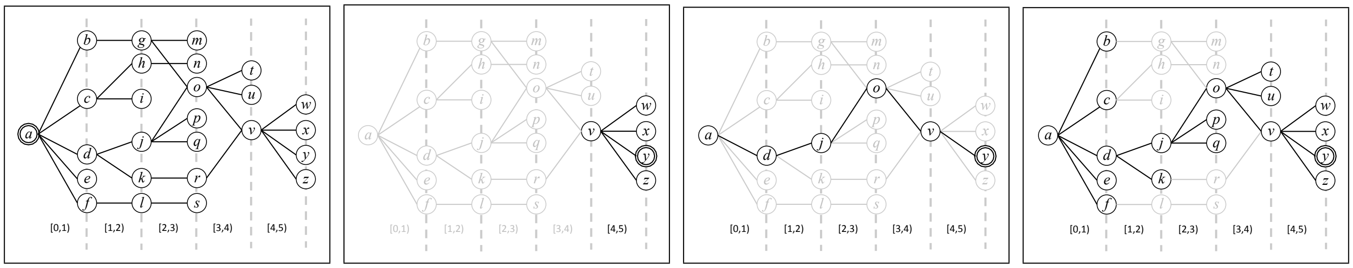
Fig. 2. A TVG $\mathcal{G}$ [2]. Edge labels represent time intervals during which edges are available, i.e., $\cup(t \in \mathcal{T} : \rho(e, t) = 1)$.

Journey. We capture temporal reachability [24], [30] by formalizing the notion of a *journey*. Informally, a journey represents a path through a network over time that a message *could* have followed. A sequence of tuples $\mathcal{J} = ((e_1, t_1), (e_2, t_2), \ldots, (e_k, t_k))$, where $e_1, e_2, \ldots, e_k$ is a walk in $\mathcal{G}$ and $t_i + \zeta \leq t_{i+1}$ for $1 \leq i < k$, is a journey in $\mathcal{G}$ if and only if $\rho_{[t_i,t_{i+\zeta}]}(e_i) = 1$. $\mathcal{J}_{(u,v)}$ is a path over time from node *u* to node *v*. We denote $\mathcal{J}^*_{\mathcal{G}}$ as the set of all journeys in $\mathcal{G}$ and $\mathcal{J}^*_{(u,v)} \subseteq \mathcal{J}^*_{\mathcal{G}}$ as those journeys starting at node *u* and ending at node *v*. If a journey exists from *u* to *v*, i.e., if $\mathcal{J}^*_{(u,v)} \neq \emptyset$, we say that a message from *u* can reach *v* at least once within the lifetime of $\mathcal{G}$, which we denote as $u \rightsquigarrow v$. However, this relationship is not symmetric. For example, $a \rightsquigarrow d$ via the journey $\mathcal{J}_{(a,d)} = ((ab, 1), (bc, 3), (cd, 7))$ in Fig. 2; however, no valid journey exists from *d* to *a*.

Transmission Network. While a journey represents potential communication, a *transmission network* represents the actual diffusion of a piece of information through the network, i.e., the set of journeys that define the causal propagation of a message in $\mathcal{G}$. We formally define the transmission network of m as $\mathcal{J}^*_{\mathcal{G}}(m) \subseteq \mathcal{J}^*_{\mathcal{G}}$, the set of all journeys along which *m* traveled. While a single journey represents a time-dependent path a message could take, a transmission network represents the set of all journeys a message actually took.

Spatiotemporal Trajectory. To enable local insight into a message's contextual history, we introduce a novel *spatiotemporal trajectory* data type. An outgoing message's spatiotemporal trajectory $T(m) \subseteq \mathcal{J}^*_{\mathcal{G}}(m)$ is updated based on a *strategy* before its transmission and attached as metadata to the message. We define three core update strategies; additional strategies could be derived and integrated based on application or domain requirements. A node receiving *m* extracts $T(m)$ and inserts the contents into the local TVG (as described as part of Algorithm 1 below). Conceptually, a spatiotemporal trajectory serves two purposes. First, $T(m)$ reveals a subset of nodes that have received *m* and when those nodes received *m*, which can be useful for an opportunistic routing protocol. Second, $T(m)$ provides a data-dependent view of the evolution of $\mathcal{G}$ (from *m*'s perspective); this can support applications such as computing a time-varying *vicinity* [23]. These two uses are complementary because they enable the causal propagation of *m* to be investigated in terms of the context of $\mathcal{G}$ or vice-versa. We explore two related use cases in Section IV.

Update Strategies. A node appends a spatiotemporal trajectory to each transmitted message. The attached trajectory includes a portion of the node's local view of the message's transmission network as well as a snapshot of the current network from the node's perspective. Exactly what is included is determined by the trajectory's *update strategy*. We introduce three strategies that capture *coverage* of the message's transmission network to varying degrees. Fig. 3 illustrates these strategies. Fig. 3a shows the entire transmission network. A TVG (e.g., as shown in Fig. 2) annotates each edge of the graph with a time interval associated with the relevant link. In Fig. 3, we assume that the edges are available in stages, from left to right across the figure. For example, all of the edges incident to *a* are available in the interval $[0, 1)$. This simplification is purely for ease of presenting the figure and the associated examples. The transmission network shows a message that originates at node *a* (at the far left of the figure); the other three panels show the trajectory information attached to the message when it is received by node *y*, using each of our core update strategies. We first describe our three update strategies, then we examine their relative tradeoffs.



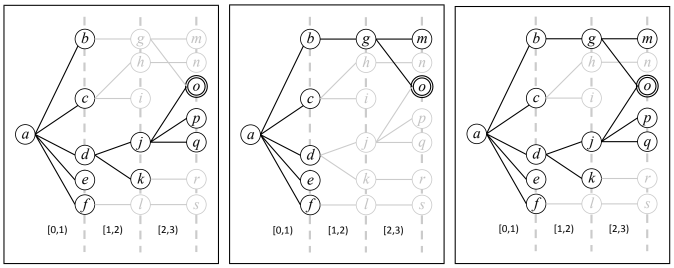
(a) Transmission graph (b) *Neighbors* (c) *Journey* (d) *Journey+*

Fig. 3. Spatiotemporal trajectory update strategies. Panel (a) shows a complete transmission graph for a message originating at node *a*, i.e., the given message traversed all of the edges shown in panel (a). Panels (b)–(d) illustrate the trajectory received at *y* (darkened nodes and edges) when updated using each strategy.

- ***Neighbors*:** Before transmitting *m* at time *t* to *k* neighboring nodes, the transmitting node v_{TX} generates a new spatiotemporal trajectory $T(m)$ containing edges between itself and all *k* neighbors that will receive *m*, i.e., $T(m) = \cup_{i \in k}((v_{TX}, v_{RX_i}), t)$. $T(m)$ is attached to *m* and transmitted to each of the *k* neighbors. This strategy produces a $T(m)$ strictly representing the first hand receivers (i.e., the "witnesses") of *m* at *t*. For instance, Fig. 3b shows that the trajectory received by node y contains information about all other neighbors of v to whom the same message was sent.
- ***Journey*:** Before transmitting *m* at time *t* to a neighboring node v_{RX}, the transmitting node v_{TX} updates *m*'s trajectory $T(m)$ (or generates one if $T(m) = \emptyset$) by appending an edge between itself and the receiver, i.e., $T'(m) = T(m) \cup ((v_{TX}, v_{RX}), t)$. This strategy produces a trajectory $T(m)$ that is precisely a journey of *m* in $\mathcal{G}$ (see Fig. 3c).
- ***Journey+*:** This strategy combines the first two. Before transmitting *m* at time *t* to *k* neighbors, the transmitting node v_{TX} updates *m*'s trajectory $T(m)$ by appending an edge between itself and each of the *k* receivers, i.e., $T'(m) = T(m) \cup_{i \in k} ((v_{TX}, v_{RX_i}), t)$. This strategy produces a journey of *m* enhanced with the receivers of *m* at every transmission along the journey's causal chain in $\mathcal{G}$ (see Fig. 3d).

When trajectories meet (i.e., when a node receives multiple copies of the same message via different causal paths) the knowledge in those trajectories can be *merged* at the receiver. Consider the example in Fig. 4, which takes a partial view of the transmission network in Fig. 3, now examining trajectories received at node *o*. By merging trajectories for a message *m*, a node can rebuild a more complete view of the transmission network. Because the trajectory data type is generic, this merging is possible even across trajectories created by different strategies or for different messages.

None of these update strategies reproduces the *exact* transmission network; in fact collecting such an omniscient, global view is very expensive (if not impossible) in a distributed way. As Fig. 3 demonstrates, the strategies produce trajectories that vary in their coverage of the global transmission network. *Neighbors* gives a very local view of the transmission; it provides good spatial coverage, but only at a single instant in time. It can be useful, for example, when y needs to know what other *nearby* devices have the same knowledge, which y can use, for example, to determine whether or not retransmitting a received message will be fruitful. The *Journey* strategy gives exactly the trajectory of the received message. It provides good temporal coverage but only captures the propagation along a single causal path. This can be useful, for example, when the message m "collects" information as it travels; the trajectory then indicates from where (and which devices) this information was collected. This is particularly insightful if different copies of m may have different semantic value, i.e., copies that follow different paths convey different information. Finally, *Journey+* combines the coverage strengths of the previous strategies but at the cost of added overhead. It conveys, to the maximum degree possible without additional coordination, m's knowledge about its distribution relative to the transmission network. Such a trajectory give applications knowlege about m's "coverage" of space and time, which may influence future decisions about (re)transmitting the same or similar information. Independent of m, these trajectories (and their unions, across multiple messages) give a general picture of the network's evolving connectivity over time, which can be used across applications and messages, i.e., in a data-agnostic way, to reason about message distribution, generally.



(a) First Trajectory (b) Second Trajectory (c) Merged Trajectory

Fig. 4. Merging trajectories.

Our approach maintains TVGs in a completely distributed way. Algorithm 1 gives the operations a node u uses to maintain its local TVG $\mathcal{G}_u$, including how the TVG is updated as trajectories are received. Initially, $\mathcal{G}_u$ only contains the node *u* (itself); E_u, ρ_u, and M_u are empty. When *u* comes into contact with a node *v*, *onContact*() is triggered, which ensures that $v \in V_u$ and $e = (u, v) \in E_u$ and then updates ρ_u to indicate the (temporally labeled) presence of *e*. When *v* moves out of contact, *onTimeout*() is triggered, which updates ρ_u to indicate that *e* is no longer present. Both *onContact*() and *onTimeout*() alter $\mathcal{G}_u$ based on *u*'s first-hand knowledge.

Algorithm 1: Local time-varying graph operations at mobile node *u* performed on *u*'s local TVG instance $\mathcal{G}_u$.

```
$\mathcal{G}_u \leftarrow (\{u\}, \emptyset, [now(), \infty), \emptyset, \zeta, \emptyset)$

onContact with a neighbor $v$ at time $t$:
$e = (u, v)$
$V_u \leftarrow V_u \cup v$
$E_u \leftarrow E_u \cup e$
$\rho_{u,[t,\infty)}(e) \leftarrow 1$

onTimeout of a neighbor $v$ at time $t$:
$e = (u, v)$
$\rho_{u,[t,\infty)}(e) \leftarrow 0$

onReception of a message $m$:
extract $T(m)$ from $m$
foreach tuple $(e = (i, j), t)$ in $T(m)$ do
    $V_u \leftarrow V_u \cup \{i, j\}$
    $E_u \leftarrow E_u \cup e$
    $M_u \leftarrow M_u \cup (m, e, t)$
    $\rho_{u,[t,t+\zeta)}(e) \leftarrow 1$

getTrajectory of a message $m$ in window $[t, t')$:
return $\cup\{w \in M : w.m = m \wedge t \leq w.t \leq (t' + \zeta)\}$
```

The injection of second-hand knowledge is performed by *onReception*(), which is triggered whenever a message *m* is received. This operation extracts *m*'s spatiotemporal trajectory $T(m)$ and inserts its contents into $\mathcal{G}_u$ tuple-by-tuple. Given trajectory tuples of the form $(e = (i, j), t)$, each tuple's nodes and edge are added to V_u and E_u (respectively), ρ_u is updated to indicate the presence of *e* for the duration of the transmission $[t, t + \zeta)$, and a tuple (m, e, t) is added to M_u indicating the transmission of *m* on edge *e* at time *t*. Finally, *u* may retrieve the spatiotemporal trajectory of message *m* within a time window using *getTrajectory*().

Device Mobility and Disconnection. It is worth briefly discussing the impacts of device mobility and intermittent connectivity on our model. While we focus on data mobility (as opposed to human or device mobility), data mobility occurs over and is constrained by the connections in a (highly dynamic) network of devices carried by mobile users. Our model's trajectories are therefore not representations of a static network but instead provide knowledge about how the network neighborhood has evolved, historically, over the ephemeral links among the mobile nodes. This knowledge can be accessed both in conjunction with specific application messages, e.g., to reason about the "coverage" of some application-level knowledge in the dynamic network, or in a data agnostic way, e.g., to reason about the changing communication network itself. A trajectory represents how a piece of data moved in the past; it is not a path that a subsequent piece of data could explicitly follow in the future. As our use cases will show, however, a trajectory or union of trajectories offer knowledge about data and connections in space and time that applications can use to influence future data dissemination actions.

III. Benchmarking

This section benchmarks the performance of our trajectory update strategies using two real world data sets of human proximity. Though not practically obtainable, a system's global TVG may be represented by the union of all nodes' local TVGs; this analysis examines the effectiveness of using our spatiotemporal trajectories as a tool for estimating characteristics of a complete transmission network (i.e., the global TVG).

A spatiotemporal trajectory $T(m)$ captures a view of the spatial and temporal history of *m*. A message's complete spatiotemporal history is given by its global transmission network $\mathcal{J}^*_{\mathcal{G}}(m)$, of which $T(m)$ is a subset. A node *u* that receives *m* initially has a local view of the global transmission network $\mathcal{J}^*_{\mathcal{G}_u}(m) = T(m)$. This local view may expand to include more of $\mathcal{J}^*_{\mathcal{G}}(m)$ if more copies of *m* are received (via different causal paths). Our trajectory update strategies produce trajectories with varying degrees of coverage of the transmission network; this benchmarking quantifies that coverage.

Experimental Setup. For these benchmarks, we employ two real world contact trace data sets of human proximity collected by the SocioPatterns project[2]. The SocioPatterns platform captures human proximity through the exchange of radio packets between RFID devices embedded in badges worn by humans. We use proximity data collected from two deployments, both of which are described and analyzed in [16] and [22]. The first deployment took place at the Science Gallery (SG) in Dublin and lasted almost three months (we use data from one day having 188 visitors). The second deployment tracked human proximity of about 100 volunteers at the Hypertext 2009 (HT09) conference in Turin. These deployments represent very different kinds of human behavior. In a museum, visitors spend a limited time on-site and generally follow a pre-defined path. At a conference, attendees stay on-site for most of the day and move at will between different parts of the conference venue. We chose these data sets because of their density, scale, and focus on human interactions. While these SocioPatterns data sets collect face-to-face human proximity, we use them as representations of more general human contact patterns. The networks we target are not limited to face-to-face proximity; existing device-to-device communication technologies can support connections across tens of meters. However, the fine-grained level of human behavior and interactions captured by the SocioPatterns deployments, which is the key driver of our evaluations, would be present in a network of human-carried devices connected by any wireless protocol.

We simulate device-to-device propagation of messages and their spatiotemporal trajectories using epidemic routing. We assume that each node in a contact trace represents a wireless device carried by a human and have each node maintain its own local TVG. For each simulation, we select a random

[2] http://www.sociopatterns.org/

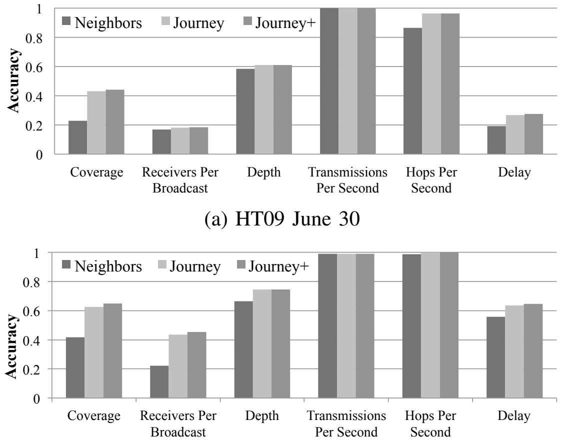
subset of the nodes to be *seeds*, which periodically inject data into the network. Every 60 seconds a seed transmits both a message and the message's initial spatiotemporal history (whose contents depend on the update strategy employed) to all other nodes in contact at transmit time. Any node that receives a message first stores the contextual history in the local TVG then performs the same steps as the seed: it updates the trajectory and propagates the message to all of its current contacts. Intermediate receivers do not buffer messages; if a receiving node does not have any active contacts at receive time, then it does not propagate the message. For the sake of these benchmarks and because our goal is not to evaluate epidemic routing but instead the capabilities of spatiotemporal trajectories, we do not impose any limit on the number of hops a message can take—device-to-device diffusion continues until all receivers of a message have already forwarded it (or attempted to). For simplicity, we assume the latency of all transmissions to be one second ($\zeta = 1$ s)[3].

Trajectories as an Estimation Tool. Our benchmark measurements investigate how well a node's partial (local) knowledge embedded in the local TVG functions as a tool for estimating global spreading dynamics and network characteristics (i.e., the global transmission network). We use five metrics to measure message propagation characteristics: *receivers per broadcast*, *transmission network depth*, *transmissions per second*, *hops per second*, and *inter-transmission delay*. When *u* receives *m* at time *t*, we compute the error between each metric as measured on $\mathcal{J}^*_{\mathcal{G}_u}(m,t)$ (*u*'s local view of *m*'s transmission network at *t*) and $\mathcal{J}^*_{\mathcal{G}}(m,t)$ (*m*'s global transmission network at *t*). We also measure *coverage*, or how well *u*'s local view of *m*'s transmission network covers the global transmission network, using the Jaccard similarity coefficient:

$$coverage(m,u,t) = \frac{|\mathcal{J}^*_{\mathcal{G}_u}(m,t) \cap \mathcal{J}^*_{\mathcal{G}}(m,t)|}{|\mathcal{J}^*_{\mathcal{G}_u}(m,t) \cup \mathcal{J}^*_{\mathcal{G}}(m,t)|}$$

We first investigate how well our update strategies support local estimation of global network characteristics. Fig. 5 shows the mean per-message average of the coverage and estimation accuracy ($1 - error$, relative to the global transmission network) achieved by each update strategy measured on one day of each SocioPatterns deployment. We use the mean per-message averages to account for varying network size. In all cases, half of the nodes act as seeds.

Not surprisingly, the *Neighbors* strategy achieves the poorest coverage of the global transmission network since it only captures one-hop contextual history. *Journey* and *Journey+* achieve much better coverage and more accurate estimations, since they both capture an entire causal chain of propagation from seed to destination. Interestingly, spatiotemporal trajectories generally produced more accurate estimations within the SG deployment (Fig. 5b) than in the HT09 deployment (Fig. 5a). We attribute this performance difference to the nature of the deployments. Nodes in the SG deployment (museum visitors) typically move through the venue within 35 minutes and rarely interact with other visitors entering the venue more than an hour after them [16]. Moreover, SG visitors generally spend *more* time in contact with *few* people. Message propagation is therefore restricted to groups of nodes that enter around the same time, which gives nodes in these cohesive groups more chances to acquire a message's spatiotemporal history. Contact activity in the HT09 deployment, on the other hand, is extremely bursty [22], with spikes of activity occurring during the lunch and coffee breaks. HT09 attendees do not follow a pre-defined path but rather move around freely, spending *less* time interacting with *more* people [16]. This diversity of interaction means that messages do not have the opportunity to propagate many hops and therefore possess less contextual history than in the SG deployment.



(a) HT09 June 30

(b) SG May 20

Fig. 5. Mean accuracy of per-message transmission network coverage and estimation metrics measured on nodes' partial transmission networks.

Though not shown due to space limitations, we also explored the coverage of each update strategy as a function of the topological distance a message traveled, and the results corroborate our conclusions. Because of contact diversity and burstiness, messages in HT09 generally do not propagate more than one or two hops from a seed and on average achieve a lower coverage of the global transmission network than in the SG deployment where nodes travel along similar spatiotemporal paths in cohesive groups. In our model, a message's spatial context is based on a perception of the surroundings—if that perception does not change, the message's "space" does not change. The spatiotemporal cohesiveness of the SG network means that, while nodes may be moving in physical space, their *context* changes slowly, which is why we observe better estimation accuracy in SG than in HT09.

In addition to capturing how well spatiotemporal trajectories capture the global transmission network, these benchmarks highlight potential uses of the trajectory information, for instance in helping applications determine what the connectivity and message propagation (and coverage) patterns are on a dynamic network. In the next section, we give two concrete examples of how applications can leverage this information.

[3] In a real world deployment, transmission time depends on payload size, the wireless protocol, available network bandwidth, etc. Our focus here is on the utility of spatiotemporal trajectories at the application layer. We leave investigations of the impacts of trajectory and data size as future work.

IV. Use Cases

This section demonstrates how spatiotemporal trajectories may be leveraged to draw concrete higher level inferences that improve the performance of opportunistic data dissemination. These use cases serve as examples of what *can* be done given knowledge of the contextual history of message propagation. They are by no means intended to indicate limits on the possibilities. The first use case takes a data agnostic view and attempts to use received spatiotemporal trajectories to build a (probabilistic) view of historical connectivity. Using this view, we show that an application can make inferences about to which other devices a message has likely already been delivered; the application can rely on these inferences to reduce the overhead of a simple message dissemination protocol without a substantial impact on the delivery ratio. In our second use case, we show how applications can form data-dependent views, inferring the likely co-location of two pieces of data and using that information to direct the search for one type towards known locations of the other.

Knowledge Inference. In this first use case, we assume a forwarding protocol that propagates messages to neighboring nodes based on some probabilistic parameter[4]. We demonstrate how such a protocol can use knowledge about the spatiotemporal history of message propagation to infer by which neighbors a message has likely already been received, thereby further reducing forwarding events. This case study relies on the observation that, in addition to indicating where and when data is spatiotemporally "about," trajectories provide a view into the network-wide state of knowledge of message transmissions. Given that an application knows something about the routing protocol responsible for disseminating data, a partially-complete trajectory can be supplemented with second-hand knowledge in the TVG to infer delivery likelihoods.

As a simple example, consider the scenario in Fig. 6 in which two trajectories, T_1 and T_2 (shorthand for $T(m_1)$ and $T(m_2)$) are received at *u* via different causal paths. Ignoring the temporal domain, from *u*'s perspective T_1 (solid gray arrows) traverses the upper walk *(a, b, u)*, and T_2 (solid black arrows) traverses the lower walk *(a, c, u)*. Node *b* receives both m_1 and m_2 from *a*. However, because of the probabilistic nature of forwarding, *b* only propagates m_1 to *u*, so the walk *(a, b, u)* is not included in T_1. Assuming *u* knows the underlying protocol's probability of forwarding (P_p), it can infer the likelihood that *b* is in trajectory T_2 (i.e., that *b* has received m_2) by using the spatiotemporal information in T_1. Since node *a* is in the intersection of T_1 and T_2, *u* knows that both m_1 and m_2 "passed through" *a*. If m_2 arrived at *a* before m_1 (or if *a* has no predecessor in T_1 or T_2), which *u* can directly infer by comparing the temporal entries of T_1 and T_2 containing *a*, then *u* knows there was a chance that *a* propagated m_2 to *b*. Under these circumstances the likelihood that *a* propagated m_2 to *b* is P_p times the topological length of the shortest known journey between *a* and *b* in *u*'s TVG.

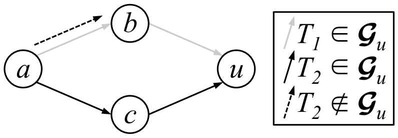


Fig. 6. Knowledge inference. Solid gray and black arrows indicate, respectively, trajectories T_1 and T_2 as known by node *u* (i.e., in *u*'s local TVG). The dashed black arrow is an element of T_2 that is *not* known by *u*, but which *u* can probabilistically infer by "stitching" together T_1 and T_2.

In implementing this use case, we demonstrate how such inferences of remote nodes' knowledge can be exploited to eliminate redundant overhead in opportunistic data dissemination. Through this use case, we seek to demonstrate how our trajectories can be used to impact network metrics. Similar network metrics have been addressed using other methods, for instance using node-to-node negotiation [13] or local machine learning over received data [31]. Our goal here is not to optimize redundancy reduction but to demonstrate potential uses of our spatiotemporal trajectories. We simulate device-to-device message propagation as in the previous section, with the addition of a probabilistic threshold, P_P associated with each forwarding event. In addition, before transmitting *m*, a node *u* uses local knowledge from its TVG to compute its confidence *c* that the destination *v* has already received *m*. If *c* exceeds a confidence threshold c_{thresh}, then *u* infers *v* has likely already received *m* and does *not* transmit *m* to *v*. Otherwise, *u* transmits *m* to *v* (subject to P_p), and *m*'s propagation continues at *v*. We measure the proportion of redundant transmissions ($\frac{|redundant|}{|total|}$) produced by the probabilistic forwarding protocol with and without knowledge inference. Additionally, for knowledge inference, we measure the achieved false positive rate (i.e., the proportion of incorrectly eliminated messages, $\frac{|incorrectly\ eliminated|}{|total|}$). We vary both the global propagation probability (P_p) and the knowledge inference confidence threshold (c_{thresh}).

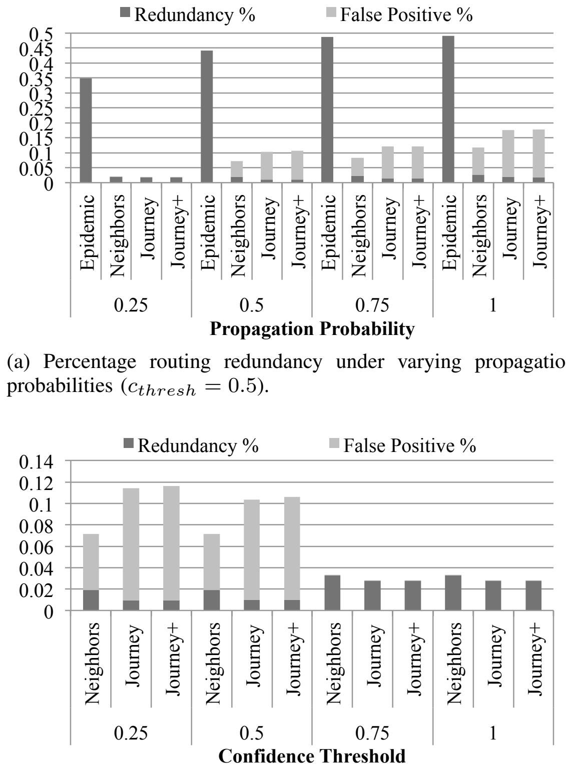
Fig. 7a illustrates the impact of varying P_p from 0.25 to 1 while keeping c_{thresh} fixed at 0.5. Knowledge inference supported by our update strategies eliminates on average 96% of redundant transmissions relative to the epidemic protocol. However, as P_p increases, the false positive rate (the proportion of non-redundant transmissions eliminated) also increases. In these cases, c_{thresh} is too low, resulting in overly-aggressive inference. This could be a problem for applications that require high delivery fidelity. However, for applications that can stand to sacrifice some portion of message deliveries or that operate in networks with limited bandwidth, a lower c_{thresh} setting could be a boon to performance.

Fig. 7b shows the effects of varying c_{thresh} with P_p of 0.5 (i.e., at each hop a message is propagated to half of the neighboring receivers). Since P_p is fixed, the redundancy of the protocol without inferences is the same (44%) for each c_{thresh}. As suspected, increasing c_{thresh} nearly eliminates false positives, but at the expense of additional redundancy when $c_{thresh} > 0.5$. Not much variation exists between the three update strategies, though the more expressive *Journey* and *Journey+* do provide some small reductions over the

[4] Such a situation is also representative of an unreliable wireless channel that may drop some relatively predictable proportion of transmitted packets.

Neighbor strategy. However, the drastic difference in false positive rates between $c_{thresh} \leq 0.5$ and $c_{thresh} > 0.5$ stands out. In fact, this jump occurs precisely when c_{thresh} exceeds P_p. This is an incredibly important feature of our knowledge inference—given that an application knows the global P_p, it can pick an appropriate c_{thresh} setting depending on how many false positives the application can tolerate.

Routing Substitute Inference. Our second case study relies on the semantics of the data with which context histories are associated. We demonstrate how applications can discover relationships between data items (e.g., that certain types of data tend to be co-located with one another) and then use these relationships to find related data. Many mobile social network routing mechanisms [17] leverage the heuristic that socially-related people tend to be regularly co-located [5]. Similarly, as a byproduct of regular co-location or of user interest, particular data may commonly occur at particular nodes. Assuming that data generated in similar spatiotemporal contexts will propagate along similar trajectories, we can detect data that are commonly co-located by measuring the degree to which their (known) trajectories overlap. For instance, in Fig. 1, *g* could infer the degree of co-location of m_1 and m_2 by overlaying the spatiotemporal trajectories received for each. Based on these observations, *g* can identify that the two messages were co-located (in both space and time) at *a*, *b*, *c*, *d*, and *e*. We refer to such related pieces of data as *data substitutes* based on the premise that finding one of the pieces of data is likely to lead to the other, related, piece. We can also detect the case that particular types of data tend to occur at particular nodes in the network, which we refer to as *node substitutes*.



(a) Percentage routing redundancy under varying propagation probabilities ($c_{thresh} = 0.5$).

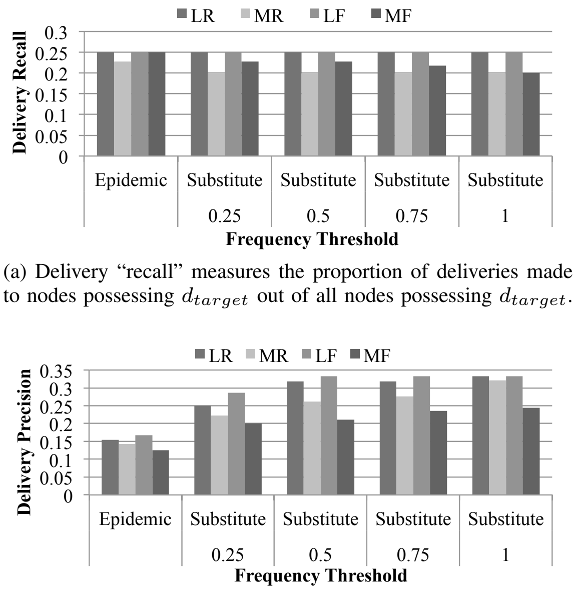
(b) Percentage routing redundancy under varying confidence thresholds ($P_p = 0.5$). Epidemic redundancy is 44%.

Fig. 7. Reduction of routing redundancies using knowledge inference in HT09 June 30. The same trends exist in the SG deployment.

In a distributed environment where global knowledge is not available, predictable regularities like these can be extremely valuable. Here, we demonstrate how such regularities may logically extend an opportunistic dissemination protocol by functioning as *substitutable routing targets*. A targeted data diffusion or query protocol may wish to deliver a message *m* to nodes possessing a particular type of target data d_{target}. For example, at the music festival a mobile beverage vendor may wish to deliver a digital coupon to attendees who have purchased salty snacks. Conversely, a thirsty attendee may want to query for digital data about nearby vendors selling cold drinks. Before propagating *m*, we can compute data and nodes that are frequently co-located with d_{target} (i.e., its *data* and *node substitutes*, respectively). During *m*'s diffusion, if *m* reaches a node where sufficient knowledge about d_{target} is lacking, we can instead use one or more of its substitutes to direct *m*'s continued propagation. Continuing the example, given knowledge that information about beer and salty snacks tend to be co-located, a search for a vendor selling cold drinks may alter its routing path based on contextual history associated with data about salty snacks.

To evaluate this use of our spatiotemporal trajectories, we implemented a routing algorithm that evaluates the effectiveness of using such routing substitutes to accomplish targeted data diffusion. We use four target data selection policies: most and least *recently* learned (M/LR) and most and least *frequently* learned (M/LF). The first two selection policies (recentness) target data that are different spatiotemporal distances from the diffusing node; the last two selection policies (frequency) target data that differ in availability. We defined a greedy data diffusion protocol that uses *only* substitutes to direct propagation. Initially, a seed is responsible for choosing the target data d_{target} per one of the selection policies. The goal is to deliver a message *m* to as many nodes possessing d_{target} as possible while minimizing the number of transmissions. Given d_{target}, a source node *u* uses only its local knowlege from its own TVG to compute d_{target}'s data and node substitutes (D_{sub} and N_{sub}, respectively) and their co-location frequencies. Node *u* then propagates *m* using only the substitutes with co-location frequencies exceeding a threshold f_{thresh} as routing targets ($D'_{sub} \subseteq D_{sub}$ and $N'_{sub} \subseteq N_{sub}$). When data substitutes are used, *u* only transmits *m* to a node *v* if *u*'s confidence that *v* has received at least one $d_{sub} \in D'_{sub}$ exceeds a threshold c_{thresh} (confidence is computed using knowledge inference). Likewise, for node substitutes, *u* only transmits *m* to *v* if *u*'s confidence that *v* has received any message from at least one $n_{sub} \in N'_{sub}$ exceeds c_{thresh}.

We simulate our greedy substitute-guided diffusion alongside an un-guided epidemic protocol with P_p and c_{thresh} both fixed at 0.5 and report results for *data substitutes* supported by the *Journey+* update strategy on a single day of the HT09 de-



(a) Delivery "recall" measures the proportion of deliveries made to nodes possessing d_{target} out of all nodes possessing d_{target}.

(b) Delivery "precision" captures the proportion of deliveries to nodes possessing d_{target} out of all diffusion deliveries made.

Fig. 8. Mean per-message average delivery "recall" and "precision" for targeted data diffusion using strictly *data substitutes* supported by the *Journey+* update strategy within the HT09 June 30 deployment.

ployment[5]. We compute two metrics: delivery *recall* measures the proportion of nodes possessing d_{target} that successfully received *m*; delivery *precision* captures the proportion of nodes possessing d_{target} that received *m* out of *all* the nodes that received *m*. We vary f_{thresh} from 0.25 to 1, interpreted as a percentile. For example, using *data substitutes* and an $f_{thresh} = 0.5$, only the top 50% most commonly co-located data are used as routing substitutes for d_{target}. An $f_{thresh} = 1$ is equivalent to our previous knowledge inference use case (i.e., no substitutes). Fig. 8 shows our results. Again we report the mean per-message average to weight our computed metrics by transmission network size.

Fig. 8a illustrates the delivery recall for varying values of f_{thresh}. Here, the un-guided epidemic protocol is a baseline, which is less than 1 simply because not all target nodes (i.e., nodes possessing d_{target}) are reachable. Impressively, substitutes achieve nearly the same delivery recall, which means that using strictly substitutes, we can deliver a message to almost as many target nodes as are physically possible. Delivery recall is unaffected by f_{thresh} under all d_{target} selection policies except for the *most frequent* policy, which decreases as f_{thresh} increases. This means that, for these settings, regardless of the *quality* of substitutes used, the same proportion of the target nodes is reached.

We next examine the delivery precision of our substitute-guided routing (Fig. 8b). Under all values of f_{thresh}, substitute-guided routing achieves better precision than un-guided epidemic routing, meaning it is more accurate per-transmission than epidemic routing. This is a byproduct of our knowledge inference, which is very effective at inferring whether an encountered node has received a piece of data or not. Intuitively, delivery precision increases with f_{thresh}; when inferential decisions are based on higher quality substitutes, more accurate inferences can be made. This case study demonstrates that detailed contextual history information carried by messages can improve the effectiveness of routing, specifically reducing the number of transmissions (i.e., overhead) while maintaining the overall performance (i.e., recall).

V. Related Work

Numerous approaches support disseminating data in a device-to-device fashion within networks of mobile nodes. Epidemic routing [6] mimics the spread of infectious agents (data packets) to susceptible hosts (mobile nodes). Many publish/subscribe mechanisms have been proposed and evaluated [5] and some are even "content aware" [7]. Other dissemination strategies provide high availability of data where it is spatiotemporally relevant [21]. TOTA [19] gives an application great flexibility by providing adaptive context-aware programming constructs that can be used to define reactive rules that govern how data diffuses. Still other methods attempt to exploit interplays between social networks, data diffusion, and mobility patterns by routing messages to nodes with high social centrality or regularity [15]. Despite this wide spectrum of device-to-device communication mechanisms, none directly capitalize on the contextual history (i.e., the causal structure) of transmitted data to distribute additional data.

Information diffusion [11] focuses on understanding and modeling causality in online social networks in order to anticipate popular topics, identify influential information spreaders, or optimize social marketing. Explanatory approaches [10] infer the underlying tree of influence representing who transmitted a piece of information to whom given only a set of nodes ordered by the times at which those nodes "learned" the information. On the other hand, predictive approaches [9] anticipate how a specific diffusion process will unfold on a given network by learning from past traces. Valuable lessons can be learned from these models. However, many of the techniques developed assume the underlying social network is globally accessible and either static or only minimally dynamic. These assumptions do not hold in the highly mobile and intermittently connected environments that we target.

Epidemiological models that capture how diseases spread through networks of humans (e.g., [14]) also relate to data movement in device-to-device communication networks. Disease spreading models have been used to probe the causal structure of data dissemination in networks of human proximity to uncover human behavioral patterns that aid in defining better routing strategies for device-to-device communication [16], [22]. The motivation of these works is very similar to ours; however, their focus is on post-hoc analysis over a global view of data's spreading history. We instead introduce a distributed model in which local information alone may be used to make *in situ* inferences about data's causal history.

[5] Performance results between *data* and *node substitutes* differed by no more than 5%. Trends were similar for the other update strategies and were consistent between the HT09 and the SG deployments.

T-CLOCKS [1] is a distributed tool that provides devices in delay tolerant networks with *temporal views* of other devices (i.e., a measure of the temporal and topological distance between nodes). Our trajectory data type may be thought of as a topological enrichment of the T-CLOCKS temporal view—a spatiotemporal trajectory provides a data-dependent temporal view in addition to the causal device-to-device propagation path of a piece of data. Similarly, *change awareness* [12] quantifies how much the latest information received from a node differs (in time) from the most up-to-date information being propagated by that node. Computing change awareness requires global knowledge of the network topology. Our model operates in a distributed fashion and captures time-dependent paths pieces of information take through the network, providing greater local insight into the contextual history of that information and the characteristics of the network.

VI. Conclusion and Future Work

Users and applications in emerging mobile networked environments require access to hyper-localized digital information. Device-to-device communications will play a crucial role in connecting devices with this proximately-available data. This paper introduced a distributed model that captures the contextual history of shared data across its lifetime of propagation and provides two complementary views: a data-dependent view, which reveals what data is about, and a data-agnostic view, which expresses how data moves. We benchmarked the performance of our model's constructs under varying network conditions using two real world data sets of human proximity. Finally, we showcased the practical utility of our model and its constructs for making distributed inference-based routing decisions. Future work will explore using our models to support deployed applications, such as the Gander search engine for the IoT [20], enabling further performance and overhead analyses. The privacy implications of collecting and sharing trajectories at varying granularities should also be explored and can potentially be leveraged to provided enhanced semantics. However, the use cases in this paper demonstrate that spatiotemporal contextual history can supplement an existing routing protocol to significantly reduce redundant transmissions and to precisely direct data towards target nodes. Future work could also explore whether our spatiotemporal trajectories could complement existing techniques to, for example, reduce network communication redundancy. However, the general purpose nature of our spatiotemporal trajectories allows them to be used by applications for multiple different purposes at the same time. Further, these use cases showcased that an entire historical *trajectory* (instead of just a source and destination) can provide additional useful information to applications to help them reduce the overhead or increase the quality of future communications.

Acknowledgments

This work was funded by the National Science Foundation (NSF), Grant #CNS-1218232 and a Google Research Award.

References

[1] A. Casteigts et al. Measuring temporal lags in delay-tolerant networks. In *IPDPS*, 2011.

[2] A. Casteigts et al. Time-varying graphs and dynamic networks. In *Ad-hoc, Mobile, and Wireless Networks*, pages 346–359. 2011.

[3] A. Chaintreau et al. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Trans. on Mobile Comp.*, 6(6):606–620, 2007.

[4] R.-I. Ciobanu et al. Big data platforms for the internet of things. In *Big Data and Internet of Things: A Roadmap for Smart Env.* 2014.

[5] P. Costa et al. Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. *IEEE J. on Selected Areas in Comm.*, 26(5), 2008.

[6] A. Datta, S. Quarteroni, and K. Aberer. Autonomous gossiping: A self-organizing epidemic algorithm for selective information dissemination in wireless mobile ad-hoc networks. In *Sem. of a Net. World. Sem. for Grid Datab.*, volume 3226 of *LNCS*, pages 126–143. 2004.

[7] D. Frey and G. Roman. Context-aware publish subscribe in mobile ad hoc networks. In *Coord. Models and Lang.* 2007.

[8] S. M. George et al. DistressNet: A wireless ad hoc and sensor network architecture for situation management in disaster response. *IEEE Comm. Mag.*, 48(3):128–136, 2010.

[9] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 12(3):211–223, 2001.

[10] M. Gomez Rodriguez, J. Leskovec, and B. Schölkopf. Structure and dynamics of information pathways in online media. In *WSDM*, 2013.

[11] A. Guille et al. Information diffusion in online social networks: A survey. *SIGMOD*, 42(1), 2013.

[12] X. F. Guo and M. C. Chan. Change awareness in opportunistic networks. In *MASS*, pages 365–373, 2013.

[13] W. Heinzelman et al. Adaptive protocols for information dissemination in wireless sensor networks. In *MobiCom*, pages 174–185, 1999.

[14] H. W. Hethcote. The mathematics of infectious diseases. *SIAM Review*, 42(4):599–653, 2000.

[15] P. Hui et al. Bubble rap: Social-based forwarding in delay-tolerant networks. *IEEE Trans. on Mobile Comp.*, 10(11):1576–1589, 2011.

[16] L. Isella et al. What's in a crowd? analysis of face-to-face behavioral networks. *J. of Theoretical Biology*, 271(1):166–180, 2011.

[17] N. Kayastha et al. Applications, architectures, and protocol design issues for mobile social networks: A survey. *IEEE*, 99(12):2130–2158, 2011.

[18] F. Li and Y. Wang. Routing in vehicular ad hoc networks: A survey. *IEEE Vehicular Tech. Mag.*, 2(2):12–22, 2007.

[19] M. Mamei, F. Zambonelli, and L. Leonardi. Tuples on the air: A middleware for context-aware computing in dynamic networks. In *ICDCS*, pages 342–347, 2003.

[20] J. Michel, C. Julien, and J. Payton. Gander: Mobile, pervasive search of the here and now in the here and now. *IEEE Internet of Things J.*, 1(5):483–496, 2014.

[21] J. Ott et al. Floating content: Information sharing in urban areas. In *PerCom*, pages 136–146, 2011.

[22] A. Panisson et al. On the dynamics of human proximity for data diffusion in ad-hoc networks. *Ad Hoc Networks*, 10(8), 2012.

[23] T. Phe-Neau et al. Examining vicinity dynamics in opportunistic networks. In *PM2HW2N*, pages 153–160, 2013.

[24] V. Rajamani et al. Inquiry and introspection for non-deterministic queries in mobile networks. In *FASE*, 2009.

[25] C. Riolo, J. Koopman, and J. Chick. Methods and measures for the description of epidemiological contact networks. *J. Urban Health*, 78(3):446–457, 2001.

[26] M. Satyanarayanan et al. The case for vm-based cloudlets in mobile computing. *Pervasive Computing*, 8(4):14–23, 2009.

[27] M. Z. Shafiq et al. A first look at cellular network performance during crowded events. In *SIGMETRICS*, pages 17–28, 2013.

[28] P. Simoens et al. Scalable crowd-sourcing of video from mobile devices. In *MobiSys*, pages 139–152, 2013.

[29] M. N. Tehrani, M. Uysal, and H. Yanikomeroglu. Device-to-device communication in 5G cellular networks: Challenges, solutions, and future directions. *IEEE Comm. Mag.*, 52(5):86–92, 2014.

[30] J. Whitbeck, M. Dias de Amorim, V. Conan, and J.-L. Guillaume. Temporal reachability graphs. In *MobiCom*, pages 377–388, 2012.

[31] B. Xu, O. Wolfson, and C. Naiman. Machine learning in disruption-tolerant MANETs. *ACM Trans. on Auto. and Adapt. Sys.*, 4(4), 2009.