Spring 2020

Towards efficacy and efficiency in sparse delay tolerant networks

Douglas John McGeehan

TOWARDS EFFICACY AND EFFICIENCY IN SPARSE DELAY TOLERANT NETWORKS

by

DOUGLAS JOHN MCGEEHAN

A DISSERTATION

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

2020

Approved by

Sanjay Madria, Advisor
Bruce McMillin
Venkata S. S. Nadendla
Yanjie Fu
Jagannathan Sarangapani

PUBLICATION DISSERTATION OPTION

This dissertation consists of the following three articles, of which two have been published and one is currently under peer review:

Paper I: Pages 37–67 have been published in the 36th IEEE International Conference on Distributed Computing Systems (ICDCS).

Paper II: Pages 68–124 have been accepted for publication to Distributed and Parallel Databases, a journal by Springer.

Paper III: Pages 125–165 has been submitted for publication in Wireless Networks, a journal by Springer.

ABSTRACT

The ubiquitous adoption of portable smart devices has enabled a new way of communication via Delay Tolerant Networks (DTNs), whereby messages are routed by the personal devices carried by ever-moving people. Although a DTN is a type of Mobile Ad Hoc Network (MANET), traditional MANET solutions are ill-equipped to accommodate message delivery in DTNs due to the dynamic and unpredictable nature of people's movements and their spatio-temporal sparsity. More so, such DTNs are susceptible to catastrophic congestion and are inherently chaotic and arduous. This manuscript proposes approaches to handle message delivery in notably sparse DTNs. First, the ChitChat system [69] employs the social interests of individuals participating in a DTN to accurately model multi-hop relationships and to make opportunistic routing decisions for interest-annotated messages. Second, the ChitChat system is hybridized [70] to consider both social context and geographic information for learning the social semantics of locations so as to identify worthwhile routing opportunities to destinations and areas of interest. Network density analyses of five real-world datasets is conducted to identify sparse datasets on which to conduct simulations, finding that commonly-used datasets in past DTN research are notably dense and well connected, and suggests two rarely used datasets are appropriate for research into sparse DTNs. Finally, the Catora system is proposed to address congestive-driven degradation of service in DTNs by accomplishing two simultaneous tasks: (i) expedite the delivery of higher quality messages by uniquely ordering messages for transfer and delivery, and (ii) avoid congestion through strategic buffer management and message removal. Through dataset-driven simulations, these systems are found to outperform the state-of-the-art, with ChitChat facilitating delivery in sparse DTNs and Catora unencumbered by congestive conditions.

ACKNOWLEDGMENTS

It is quite challenging to give due acknowledgement to all who aided me through this ardruous journey. Of course, I thank Dr. Sanjay Madria and Dr. Dan Lin, my academic advisors throughout my Ph.D., for their vision and guidance on which problems to research and the academic rigor needed to solve them. Beyond academics, the following people deserve recognition: my family, forever providing me unconditional love and support, a warm meal and an in-person visit when needed, and periodic reminders that the end is within sight, even when obscured by seemingly-impenetrable obstacles; my friends: Ashok, Ayan, Brett, Cameron, Dan, Gracie, Himanshu, Jenna, Jess, Jonathan, Katrina, Lauren, Michael, Shudip, and countless others for providing companionship, joyful memories, and an occassional escape from obligations; my pets: Linus, Lucy, Leland, and Isabel, for their warmth on cold nights and snuggles during emotional stress; Dr. Stephen Jackson, for giving me a thread of hope when I felt I had lost it all; Dr. Jeffrey Schramm, for his leadership and judgement-free ear; and all the DJs at KMNR 89.7 FM for the opportunity to lead their organization. No doubt, I have missed individuals who deserve acknowledgement. Just know that I feel indepted to you regardless of whether your name is listed here.

I dedicate this dissertation to the loving memory of my two dogs, Linus (2010–2016) and Lucy (2008–2019).

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

PAPER III

LIST OF TABLES

SECTION

1. INTRODUCTION

Smartphones have become a new accessory for many individuals, carried inside pockets and purses everywhere their owner goes. Forecasts predict that, by 2024, approximately 8.4 billion smartphones will be online, with an estimated 92% of the world's population having mobile broadband coverage to drive further adoptions of these devices [32]. This overwhelming distribution of wirelessly-connectable mobile devices has garnered interest from researchers seeking novel approaches to capitalize on their abundant availability. One particular area of interest investigates how to establish a network between only these smartphones, carried around by people, without relying on a connection to the Internet. Such a network has been interchangeably referred to by several names in research literature, such as a Delay/Disruption Tolerant Network [119, 120] (used in this report, abbreviated DTN), Mobile Opportunistic Network [3, 66, 111, 128], Intermittently Connected Networks [104], Pocket Switched Network [71, 108, 124], and so on. Within this network, the establishment of connections between nodes is dictated by people's mobility over time; when two nodes come within close proximity to each other, their devices are able to wirelessly connect. There are plenty of interesting applications that are possible in such an environment, as is detailed in the following subsections.

1.1. APPLICATIONS

1.1.1. Disaster Communications. Natural and man-made disasters can severely damage or completely destroy an area's communication infrastructure, preventing a timely and coordinated response from rescue personnel [6, 76, 89]. It takes time to roll out a temporary infrastructure – time which may be crucial in preventing further loss of life. With people already carrying smartphones, an emergency network could activate to immediately begin the retrieval and distribution of information. Examples of disaster-related information may pertain to severity of building damage, the location of isolated and injured people, or the contamination of air and water supplies.

1.1.2. Ad Hoc Battlefield Communications. An active battlefield is hardly a place to expect a secure and reliable communication infrastructure [4]. Strategic movements and enemy attacks require that any sort of communication infrastructure have protection, redundancy, and mobility to enable communication. Military vehicles, having electricity produced from diesel/gasoline engines, may act as mobile base stations to receive and transmit information about the situation at hand. Foot soldiers are able to move away from these vehicles to closer examine a point of interest, gather intelligence, and disseminate orders or warnings. By equipping soldiers with wirelessly-communication devices and visual, audio, and environment sensors, the turn-around between intelligence gathering and strategic response can be greatly reduced.

1.1.3. Offloading Traffic from Overloaded Networks. Locations where many people crowd together can overwhelm a wireless base station for that area. This can happen in both urban areas – e.g. San Francisco [115] – and rural areas – e.g. outdoor-oriented Music Festivals [51]. One option to alleviate such overloading is to provision more infrastructural resources to that area. This approach may not be economically feasible, though, due to the transient nature of such overloading events. In these areas, locally relevant information does not need Internet connectivity for successful distribution. Instead, information such as venue programming changes and emergency alerts can be distributed in a peer-to-peer fashion between the devices in that area [110, 122].

1.1.4. Floating Content. Geosocial network applications such as Tinder and Air-Drop provide users with content and social connection opportunities specifically relevant to their geographic location. However, these applications require an Internet connection to retrieve such content, and mobile devices must report their location to submit and retrieve content for their area. This approach opens up the possibility of location spoofing as a means of accessing content without the user being in the area. One way to maintain the locality and isolation of such information is to deploy it as *floating content*, where data is physically located in the location where it is relevant [28, 74]. Content can be stored in stationary dropboxes (e.g. a WiFi router equipped with adequate long-term storage) or virtual dropboxes, where the content is exchanged between and cached within devices entering and leaving an area. This method of content posting and retrieval permits devices to exchange content only when a direct connection can be made with a dropbox, thus requiring nearby proximity. The content is said to *float* in that particular area, adding a new aura of experience and entertainment to that area.

1.1.5. Network Connectivity for Developing Communities. In some environments, it is economically infeasible to provide an Internet connection through the installation of necessary infrastructure [37]. Such environments could be rugged and hostile settlements (e.g. camps leading to the peak of Mount Everest), widely distributed low-density population centers – e.g. scattered villages in rural South Africa [34] – along rivers for pollution monitoring [107] – and deep within the wilderness for animal tracking [99]. These areas may still benefit from Internet connectibility, even if the latency for delivery is orders of magnitude longer than what is available in more developed regions. Assuming a stream of people are arriving and leaving the area, each traveler can act as a carrier for content requests and delivery, thus providing connectivity at a significantly lower cost than dedicated infrastructure. In a sense, this approach would mimick a local library in terms of content requests and delivery delays, which was the source of knowledge for several millenia before the invention of the Internet.

1.2. CHALLENGES WITH MANET SOLUTIONS

Considering these applications, one may question whether solutions for Mobile Ad Hoc Networks (MANETs) could be adopted. A MANET is characterized as a set of communication-able nodes, each with its own ability of movement, deployed to serve some networking functionality that would otherwise be cost prohibitive for traditional infrastructure installation. Indeed, a Delay Tolerant Network is a specialization of the more generic MANET definition. However, the solutions proposed to provide MANET functionality are not practical in a DTN environment based on their implicit assumptions.

Consider, for example, on-demand routing protocols for MANETs – e.g. AODV [75], DSR [50]. When a message is generated at a source node for some destination, the first step of an on-demand routing protocol is to discover a path to the destination. This is accomplished in three phases: route request, route reply, and route maintenance. When there is no known path to the destination, the source will begin the process of discovering a route by contacting its neighbors, who contact their neighbors, and so on until the destination is located. Once the destination is found, a reply is generated to notify the sender of the route, following the reverse path of the request. This reply also provides a caching opportunity to all other nodes receiving the reply, who may record the path for later use. However, due to the mobility of participating nodes, the topology of the network may change, rendering some cached routes obsolete. When this happens, route maintenance occurs, allowing for the network to self-heal and for cached routes to be updated, albeit at the cost of additional overhead.

In order for route request and route reply to work, it is implicitly assumed that the path between the source and the destination will last long enough for (1) route discovery to reach the destination, (2) a route reply to backtrack back to the source node, and (3) the message to traverse through the path. Once it is known, the nodes along the path will receive the message and pass it along without any concern of long-term queuing/storage. This is a tough requirement to achieve for a network composed of smartphones carried by people.

The communication range of these devices is limited by their antenna length, government regulations on their transmission power, and the available battery capacity. In order for a DTN to be instantaneously connected requires a sufficient density of people. However, the mobility of these participants may rapidly change the topology of the network, rendering path discovery useless before any of its phases are complete. In reality, the network is more likely to be highly partitioned with many disjointed, small connected subgraphs scattered across the network's geographic area. As a result, messages may need to be stored until such a time as a bridge between connected subgraphs is formed. Simply put, past MANET solutions are crippled by the amplified mobility and network disconnections of a DTN.

Because of these challenges, solutions for a DTN must embrace and exploit the nature of human mobility, acknowledge the technological limitations of smartphones, and tolerate unpredictable, long delays for message delivery, and limited knowledge. This manuscript seeks to explore how researchers have dealt with such issues, provides an understanding of what remains to be addressed, and proposes our own solutions for accommodating message delivery in sparse DTNs.

1.3. CHARACTERISTICS OF DTNS

Delay Tolerant Networks are differentiable from other network architectures by the more frequent occurrence of disconnections between established channels of two nodes. Many other network types are often assumed to have some sufficient degree of stability and availability of communication channels between nodes; DTNs make no such assumption. Communication between nodes in a DTN may fluctuate more aggressively through time than what would be expected in other networks.

DTNs are derived from the more general Opportunistic Network (ON), which is itself derived from the more general Wireless Ad-hoc Network (WANET). The primary difference between DTNs and other ONs is the primary cause of connections and disconnections:

mobility. When two people move to become sufficiently close[1], a connection between their devices may be formed to open a channel for communication. Likewise, when they move away from each other, communication ceases. These connections and disconnections are *opportunistic*, implying that their formation and destruction has some degree of stochasticity and some degree of chaos, introducing uncertainty in predicting their occurrence. ONs are seperated from the more general WANETs by the degree to which uncertainty prevents connection predictability. The following two examples illustrates how a DTN may be different from some other type of ON/WANET.

Example 1: A network composed of polar-orbitting satellites and grounded receivers will have connections established due to a satellite passing through the portion of its orbit in the sky above the receiver. Disconnections occur due to the satellite progressing beyond the line-of-sight of the receiver or due to obstructions such as large buildings or underground tunnels. Thus, in order to accomplish network functionality, participating nodes should expect to store data for some period of time until the next connection will occur. Likewise, applications using this data should expect extended delays in its delivery. Although this network is a ON, it would not strictly be considered a DTN due to the availability of a contact plan – i.e. the guaranteed recurrence of connections at known intervals. The strong predictability of connections and disconnections allows for many important networking metrics to be calculated with increased certainty, such as the expected delay of data delivery, the transmission speed of the established channel, the theoretical upper bound on the channel's capacity, and an upper bound on necessary queuing storage.

Example 2: A network composed of wirelessly-connected mobile devices, carried by people, will have connections established due to two people coming within close proximity. Disconnections primarily occur when the two people move away from each other. Due to the unpredictable nature of human mobility, it is unknown when and for how long two nodes will be within communication range of each other. This prohibits a node from

[1]*Sufficiently close* would be defined by the wireless communication radius of both devices.

knowing how long it must store a message. Likewise, if a node does not know which other node it will connect with next, it also cannot know if a path to the destination will even exist to the destination. This network is an ON because it relies on opportunities to achieve network functionality, and further is considered a DTN due to the aforementioned uncertainties. When a connection opportunity arises, a message-holding node must determine if it will use this opportunity to forward its message to the encountered node.

These two examples highlight characteristics that can be used to differentiate one type of Delay Tolerant Network from another, as well as differentiating one type of Opportunistic Network from another. Example 1, for instance, has intrinsic predictability of when future connections will be formed. A schedule for message transmissions can be computed based on the satellite's orbital model, and resource allocation can be planned in advance of network deployment. By being able to predict future connections, the method of constructing delivery paths becomes a problem modeled by the quickest trans-shipment problem [33] or a mixed integer linear program [113, 135], which would allow traditional networking protocols to be trivially modified. Example 2, on the other hand, does not have such a luxury of predictibility due to human mobility. Fall [33] enumerates four types of DTNs with different characterizations of the causes of frequent interruptions: Terrestrial Mobile Networks, Exotic Media Networks, Military Ad-Hoc Networks, and Sensor/Actuator Networks. Each of these network types may be further classified by the predictability of interruptions (e.g. a Terrestrial Mobile Network of buses has symmetry and predictability due to fixed routes, whereas one of taxis has no guarantee of repetitive or predictable motion) and in the directionality of communication (e.g. military communications with a submarrine may only be one way due to its covert radio silence). In this manuscript, the focus is on reviewing recent advancements in Terrestrial Mobile Network characterised by opportunistic and unpredictable connections.

1.3.1. Temporal Model of a DTN. The traditional modeling of networks as graphs requires augmentation to accurately describe a DTN [16]. Instead of a graph $G = \langle V, E \rangle$ representing the network, a Time-Varying Graph (TVG) $\mathcal{G} = \langle V, E, \mathcal{T}, \rho, \zeta \rangle$ is required, where V is the set of nodes; $\mathcal{T} \subseteq \mathbb{T}$ is a timespan within the temporal domain $\mathbb{T}$ representing the lifetime of the network; E is the set of edges representing network connections that exist at some time instant during $\mathcal{T}$; $\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$ is a presence function that indicates whether a given edge exists at a given time instant; and $\zeta : E \times \mathcal{T} \rightarrow \mathbb{T}$ is a latency function that indicates the time duration needed to traverse the given edge starting at the given timestamp. The state of connectivity of the network depends on the locations of individuals at a given time. The set of edges $\{e \mid e \in E, \rho(e, t_0) = 1\}$, denoting the edges in the network at time t_0, may be considerably different when compared to the set of edges $\{e \mid e \in E, \rho(e, t_1) = 1\}$ at some other time $t_1 > t_0$. Between t_0 and t_1, people have moved, connections have been broken, and new connections have been formed. Thus, the topology of the network could be radically different at different periods of time.

To accomplish message delivery in a DTN, the definition of a path between two nodes u and v is often insufficient due to the scarsity of contemporaneous connections between nodes. Here, a path is defined as a sequence $P^{u,v} = (e_1 = \{u, w'\}, e_2, ..., e_\omega = \{w'', v\})$, such that:

- The nodes for each edge are nodes in the graph – i.e. $u, w', w'', v \in V$
- Each edge of the path is an edge in the graph – i.e. $\exists\, e_i \in E \;\forall\, e_i \in P^{u,v}$
- Each pair of consecutive edges shares a common node – i.e. $|e_i \cap e_{i+1}| = 1 \;\forall e_i \in P^{u,v}, 0 < i < \omega$

Rather, a *journey* connects two nodes through multiple hops that span time, and at each time instant the multi-hop path between the nodes is not necessarily connected. Formally, a journey $J^{u,v}$ in $\mathcal{G}$ is a sequence of timestamped edges

$$J^{u,v} = \left(\langle e_1 = \{u, w'\}, \mathcal{T}_1 = (t_1^s, t_1^e)\rangle, \langle e_2, \mathcal{T}_2\rangle, \ldots, \langle e_\omega, \mathcal{T}_\omega\rangle\right)$$

such that:

- The nodes u and v are on the first and last edges, respectively – i.e. $u \in e_1$, $v \in e_\omega$
- Each edge is connected during the defined time duration – i.e. $\rho(e_i, t) = 1 \;\forall\; t \in \mathcal{T}_i$
- An edge along the journey is open sometime after the previous edge – i.e. $t_{i+1}^s \geq t_i^s + \zeta(e_i, t_i^s)$ for all $0 < i < \omega$.

Let $m = \langle t_c, t_d, \texttt{payload}\rangle$ be a message, where $t_c \in \mathcal{T}$ is message's creation time, $t_k \in \mathcal{T}, t_d > t_c$ is the time when the message is no longer useful, and `payload` is the content of the message. A message m is deliverable from a node u to another node v through $\mathcal{G}$ if and only if there exists a journey $J^{u,v}$ such that the message is created before the first connection ends and expires after the last connection begins – i.e. $t_c < t_1^d$ and $t_d > t_k^s$.

1.3.2. Requirements for Successfully Delivery. In order for a message to pass through a DTN, each node must adopt the store-carry-forward paradigm of message delivery and make best-effort decisions for routing. In this paradigm, a node stores a message in its buffer, carries it as it moves about, and, upon encountering a neighbor, decides whether to forward the message. This process brings about several bottlenecks.

- **Buffer space:** How much buffer space is available in each node to carry these messages? This attribute depends on the size of each message, the frequency at which messages are created, the duration of time a message is carried until it is dropped, and the aggressiveness of nodes forwarding messages to encountered neighbors.

- **Channel capacity:** How much channel capacity is available when two nodes connect with each other? This attribute depends on the power consumption, duration of time two nodes are connected, the distance between two nodes, and the overhead of the protocol used for communication.

- **Battery capacity:** How much energy does a node have to dedicate to wireless transmissions? Smartphones are not solely used to relay messages between other devices, and it would otherwise be undesirable if a phone's battery was depleted in doing so.

- **Delivery delay:** How much time will the recipient(s) of a message have to wait until it is delivered? This depends on many complex factors, a few of which are the geographic distance between the source and the destination, the number of participants in the network, and the amount of time each relay must carry the message before forwarding it on further to the next worthwhile neighbor.

Several forwarding strategies have been adopted by proposed solutions for DTNs. Some schemes will simply push a message to all encountered neighbors not already carrying a message. These strategies are known as flooding schemes. Although one may think this approach would be able to provide the highest guarantee of delivery in a DTN, this approach runs this risk of triggering network congestion [49].

Another strategy is to have nodes make calculated decisions on to whom a message should be forwarded, whether it be single-copy forwarding (i.e. at most one copy of a message resides in the network at any time) or a multi-copy forwarding (i.e. when a node forwards a message to a neighbor, it retains a copy for future forwarding opportunities). Implicitly, this strategy requires a node to have some knowledge on which to base its decision. Without the availablility an Internet connection nor contemporaneous paths, knowledge of the global state of the network is unavailable to each node. This prohibits nodes from computing optimal routing plans; instead, the knowledge available to each node is limited to that it has learned and stored locally or that it can learn from its connected neighbors. However, this stored knowledge must be accurate to some degree. If this knowledge pertains to currently open connections in the network, it can quickly become stale due to frequent changes in network topology. If it pertains to predicting future connections, there is a chance the future connection will not occur. Thus, if a node is to rely on local knowledge within itself and its neighbors, knowledge recording mechanisms must be designed in such a way that can indicate whether an encountered neighbor is a worthwhile candidate. Section 2.1 details how recent proposals have designed such mechanisms.

1.4. PERFORMANCE METRICS

In order to evaluate the effectiveness of a proposed DTN routing scheme and make comparisons between other schemes, it is important to understand certain performance metrics. In their surveys on DTNs, Abdelkader et al [2], Cao and Sun [12], and Liu et al [65] describe three metrics (defined below) that are commonly used in DTN routing simulations: (1) delivery ratio; (2) delivery latency; and (3) overhead ratio. In addition to these three, hop counts have also been demonstrated as an important metric to optimize [1, 125]. These metrics are important for evaluating DTNs and many past proposals have used them to compare their contribution against other state-of-the-art approaches.

1.4.1. Delivery Ratio. When n messages are created for dissemination in a DTN, and r messages are successfully delivered, then the *delivery ratio* for the given experiment is $\frac{r}{n}$. A higher delivery ratio indicates more messages were able to reach their destination.

1.4.2. Hop Count. The *hop count*, also called the *path length*, of a delivered message is defined as the number of nodes that relayed a message over the first journey that reached the message's destination. For the r messages that are successfully delivered, the reported metric is often the average number of edges in the delivering journey for each. For single-copy forwarding schemes, the hop count of a delivered message is approximately proportional to the amount of power consumed to deliver the message. For multi-copy forwarding schemes, the propagation pattern forms a tree with the source as its root. The hop count for multi-copy forwarding schemes is the distance from the root node to the destination node, which only accounts for a subset of the number of forwards.

1.4.3. Delivery Delay. Assume that r messages are successfully delivered in a DTN. For each message m_i of the r delivered messages, let Δt_i be the duration of time between message m_i being created and it arriving at its destination. The *average delivery delay* for the experiment is defined as

$$\frac{1}{r}\sum_{i=1}^{r}\Delta t_i$$

Should all messages pass along their globally optimal paths of earliest arrival time, then the average delivery delay is optimal as it implies all messages were delivered as fast as possible within the simulated network.

1.4.4. Overhead Ratio. DTNs are derived from the more general Opportunistic Network (ON), which is itself derived from the more general Wireless Ad-hoc Network (MANET). For multi-copy forwarding schemes, the *overhead ratio* gives an idea of the resource costs that are needed to successfully deliver messages, and only applies to multi-copy strategies. When a message m is forwarded from a node s to its neighbor t, an additional copy of m is created, thus occupying additional buffer space and consuming power for transmission. These resources are non-renewable; regardless if m is successfully delivered, the resources consumed to disseminate m throughout the network have been consumed.

Assume that n messages are created in a DTN. For each message m_i of the n messages, let c_i be the number of times message m_i was forwarded through a channel. The *overhead ratio* for the experiment is defined as

$$\frac{1}{n}\sum_{i=1}^{n}c_i$$

An overhead ratio of 1 is optimal for minimizing resource consumption. It implies all delivered messages were delivered in only one hop on a channel between the source node and the destination node. Typically, an overhead ratio is expected to be $>$ 1, and a comparison between two routing algorithms, all other factors remaining equal, would have the router with the lesser overhead ratio being favored[2].

1.4.5. Caveats of Metric Interpretations. The above-mentioned metrics may conflict with one another when evaluating a DTN forwarding scheme. For instance, assuming the absense of network congestion, a flooding approach is able to achieve both an optimal delivery delay and delivery ratio in a DTN at the cost of increased overhead. Overhead can only be reduced by preventing some nodes from using an opportunistic connection with a neighbor to pass along a message. By placing restrictions on which connections to use, a lower delivery ratio and an increased delivery delay is to be expected. It is thus important that, when interpretting and comparing results, one metric should be allowed to vary while all other metrics are held stationary. For example, if two schemes are able to achieve the same delivery ratio, then the scheme with the lower overhead ratio or the lower average delivery delay is favored. Determining which performance metric is favorable, however, depends on the application specifications.

[2]Favoring the router that consumes less resources is typical, but not absolute. Some routing strategies are more favorable because of their reduced latencies or increased delivery ratios, albeit at the cost of consuming more resources.

2. LITERATURE REVIEW

2.1. PROPOSED SOLUTIONS TO DTN ROUTING

In this section, the state-of-the-art DTN solutions are categorized and briefly described. The focus here is to detail what types of knowledge has been demonstrated as useful for DTNs, and how this knowledge is gathered within the confines of a DTN to enable nodes to make informed forwarding decisions. One area of focus has looked into topics from social network analysis, such as centrality (Section 2.1.1) and community detection (Section 2.1.1). Another draws from online social networks, using a node's explicit friend list (Section 2.1.3) and social profiles (Section 2.1.4) to bootstrap network functionality without waiting for opportunistic training to complete.

2.1.1. Centrality. Centrality is a measurement tool originating from graph theory and social network analysis [23]. It provides a scalar measure of how important a node's presence is in its network. There are many flavors of centrality that have been proposed, each providing a different point of view of a node's importance. Degree centrality measures the number of direct links made to a particular node. In an opportunistic network, a node with a high degree centrality is one that is characterised with having contacted many other nodes, whereas one with a low degree centrality would be a characteristic of a recluse. Closeness centrality reflects the distance between a node and all other nodes in the network based on the shortest paths connecting the two. Nodes with a high closeness centrality are said to be able to contact other nodes quickly, while nodes with low closeness centrality must rely upon long paths in order to reach others in the network. Betweenness centrality for a node measures the number of times the node falls on the shortest paths between two other nodes in the network. When considering information flows within the network, a node's betweenness centrality indicates how influential that node is in facilitating the flow of information if it is assumed that information always flows through a shortest path.

A DTN-appropriate variation on the betweenness centrality is proposed by Zhang et al [128], which specifically targets a specific pair of nodes – e.g. a message carrier and the destination of the message – instead of just a single node. For a given pair of nodes s and d, the destination-aware betweeness centrality is calculated as the sum of ratios of the number of shortest journeys passing through s and ending at d over the total number of shortest paths ending at d.

Probabilistic centrality is investigated by Wu et al [118], where they propose the reachability probability centrality between two nodes. This metric considers the probability that two nodes are reachable over journeys of any length from 1 to k. This is built on the baseline probability of a direct contact, as calculated from ratio of the average contact rate between two nodes over the sum of averages across all pairs of nodes.

With many centrality metrics to draw upon, some researchers have questioned which serves a more effective role in facilitating DTN routing. Socievole et al. [98] investigated how degree centrality, eigenvector centrality, and egocentric betweenness centrality affected the performance of their proposed router. Their findings show that degree centrality provided better delivery ratios, followed closely by a properly tuned eigenvector centrality. Betweenness centrality, however, performed the worst out of their simulations. Although it performed worst, small networks have it performing better than large networks.

Degree centrality is calculated in BUBBLE Rap [42], where a message *bubbles* up through the network reaching more popular nodes until it enters the destination's community. Once there, the forwarding strategy shifts its focus from global degree centrality to community-centric degree centrality (i.e. degree centrality with nodes of that community). Hui et al. postulate that a node's popularity within its community is a more effective metric for reaching a message's destination than the node's global popularity when the node shares a community with the destination. In order for each node to know its popularity, each computes two centrality metrics for themselves: a local community-focused centrality and a global centrality.

Rango et al [83] extend this idea by computing node centrality using different inputs and adding time variability. This centrality metric, called the Fused Online and Offline Centrality (FOOC), is computed by considering both measurements on opportunistic contacts in the DTN as well as explicit social-ties founds in online social networks. This metric varies across certain time windows, allowing it to fluctuate based on changes in a node's sociability through time. Additionally, the influence of DTN and online contacts on a node's FOOC can be tweaked in real-time based solely on the node's local information. When a node recognizes that it is not meeting many others, it will put less emphasis on its online contacts and more emphasis on the nodes it directly meets.

Another approach to proposing a centrality suitable for DTNs is proposed by Zhou et al [136]. Similar to FOOC, the time-ordered cumulative neighboring relationship (TCNR) centrality is computed by analyzing a node's connectivity in the network at consecutive time-frames in the network's operation. However, it differs significantly by considering multi-hop journeys as part of its computation, whereas previous DTN-applied centrality metrics have only considered direct, single-hop encounters. The TCNRs of each node is calculated based on the average and variability of time durations between contacts with all other nodes in the network during specific time windows, whether the contacts be direct or through journeys. Each of the centralities for each time window is then aggregated to produce a single scalar value. Zhou et al propose three such aggregation methods: (i) averaging the centralities, (ii) a linear-decay weighing of the centralities, and (iii) an exponential-decay weighing of the centralities. The last two methods favor the centralities that occur earlier in the network, and thus offer a better chance to facilitate message delivery.

2.1.2. Communities. Communities are an artifact derived from people's desire to socialize [36], and are exploitable for DTN functionality [84]. They are characterised as a tightly-connected group of nodes in a network, with each community having more intracommunity connections than intercommunity connections. This leads to one node having more connections with members of his communities than with nodes outside of his communities. Additionally, a node will more likely have future connections with other members of his community than with members outside. This characteristic lends itself to assisting in the delivery of messages in DTNs; if a message destined for node d can be delivered to some individual who is a member of a community of d, then the possibility of a successful delivery increases. Indeed, having community information is useful for many DTN schemes, as has been demonstrated in recent investigations [10, 42, 59, 130]. When nodes are aware of communitiy formations and a destination's membership in these communities, they are able to decide to whom a message should be forwarded when contact opportunities arise. It's as simple as asking "which communities do you belong to?"

To exploit communities, nodes must first know about their existence, their boundaries, and their membership. BUBBLE Rap [42] accomplishes this by having nodes explicitly label themselves with one of their affiliated communities. When a message is created in a DTN, its header includes the destination's ID as well as the communities with which she is a member. This information might be unavailable, as was the case with BUBBLE Rap's performance evaluation. Centralized community detection was needed, after which labels were explicitly assigned to each node as input for their simulations. Even if people are aware that they are a member of some communities (e.g. a Volleyball team for their interdepartment company games), they may be unaware that they are part of other unconventional communities (e.g. a group of commuters taking the subway to work each morning). Detection of these communities must be performed based on the contact information accumulated by each node's participation in the network.

In [59], the SEBAR algorithm employs k-clique communities to distribute *social energy* to nodes, which in turn is used to make routing decisions. The social energy metric is dynamically computed based on the duration of a contact between two nodes, and some of the *energy*[1] generated is shared with the other nodes residing in their communities. It also exhibits decay in the form of a sliding window average of past and current energy values so as to maintain up-to-date accuracy on a node's social activity. When considering message forwarding, upon creation of a message, the source caps the number of permitted message copies, with the router halving this amount for each node that receives a copy. A two-phased routing strategy, similar to that of BUBBLE Rap, is adopted based on the location of the message at each juncture. If the message is outside of all of the destination's communities, the message carrying node forwards the message to its neighbor if the neighbor has a higher social energy. Once the message has reached a node that shares a community with the destination, the second phase only forwards to other nodes also within a community of the destination, specifically to those that are members of communities with higher social activity than the message carrier.

Zhang and Cao [130] consider the temporal relevance of community formation, with an emphasis on detecting the dividing line between multiple communities that are spatially and temporally correlated. They identify two scenarios that may hinder community detection in a time-varying network: false mixture, and false separation. A false mixture of two communities occurs when, through the process of detecting communities in a network, two logically separate communities are merged due to spatial and temporal overlap and common nodes. This may occur on a college campus, for example, when one class is dispersing and the next class is taking their seats. A false separation, on the other hand,

[1]The authors adopt the term *energy* as a quantity produced by a contact between nodes as an analog to the energy produced by particals colliding in partical physics. It is not to be confused with the impression that a node contact creates actual thermodynamic energy.

is the division of a ground-truth community due to weak member connections. The time-varying nature of DTNs makes community detection through traditional methods, requiring static graph representations of the network, susceptible to improperly divided communities.

To accomplish accurate community detection in the disconnected environment of a DTN, Zhang and Cao propose the Contact-burst-based Clustering Method (CCM) for the real-time detection of communities within a DTN. Since contacts are instantaneous (e.g. observing a beacon broadcast from a neighbor), contact observations are merged into a contact burst $T_B = [t_s, t_e]$, starting at time t_s and ending at time t_e, if each instantaneous contact is within λ time units of the next. Contact bursts are then merged to form clusters. Two contact bursts T_{b1} and T_{b2} are merged if (1) they share a common node, and (2) their temporal overlap, represented by the Jaccard similarity $S(T_{b1}, T_{b2})$ over their time durations, exceeds a threshold γ.

$$S(T_{b1}, T_{b2}) = \frac{|T_{b1} \cap T_{b2}|}{|T_{b1} \cup T_{b2}|}$$

The merging of clusters requires internode exchanging of cluster information, and continues until the largest similarity between all known clusters cannot exceed γ.

Chen and Lou [19] propose another two-phased Community Aware Routing (CAR) protocol that considers expected encounter rates of nodes and the time-to-live (TTL) of the messages they carry. While a message is outside of the destination's communities, nodes rapidly try to spread copies of the messages to others such that the new recipient is in a community not yet carrying the message. With each forwarding, the number of replicas that are allocated to the new recipient is devised proportional to the expected number of community contacts of the two nodes within the remaining life of the message. Once the message has arrived within a community containing the destination, a strategy similar to BUBBLE Rap pushes the message copies to nodes with more frequent intra-community contacts. Unlike past systems, though, the system has a contingency for when a node can no longer make additional message copies – i.e. the total number of permissible message copies has been reached in the network. When this occurs, the single copy of the message is forwarded to whichever node has a higher likelihood of encountering the destination or one of its communities.

With community memberships and time occurrences known with these proposed systems, the periodic recurrence of communities can further be detected to assist with message delivery. If a node can accurately predict the formation of a community within a future time frame, this node can then decide if forwarding a message to another community is worthwhile. When considering a message's destination, the node carrying the message is aware of the destination's communities. The node is also aware of the membership lists of these communities and the time in which they meet and ajourn. This information lends itself to calculating the relay capability of one community to another community based on the shared membership between those communities. With this information, a node meeting another node in adjacent communities can decide to forward its message if the communities of the neighboring node have higher relay capabilities to the destination than the current node's.

These solutions, however, require either centralized computation of community membership and knowledge of predictable node behavior, or a lot of data overhead in order to detect communities in a real-time and opportunistic manner. This overhead adds to an already constrained network environment and leads to the possibility that some nodes may be unable to access up-to-date information for their forwarding decisions. To address this, Bulut et al. [10] propose a community detection and routing system that does not require network-wide distribution of community knowledge. Instead, nodes only need to exchange information with their connected neighbors.

Nodes deployed with Friendship Based Routing [10] construct their communities by considering elapsed time between contacts with other nodes in the network. For each node u, its friendship community F_u is defined as the set of nodes $\{v_1, v_2, \ldots\}$ that meet two conditions for set membership. The first condition considers the social pressure metric

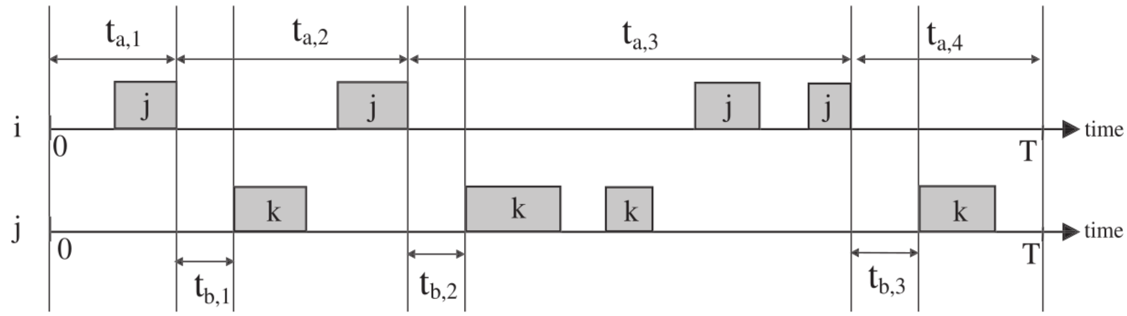


Figure 2.1. Illustrative example for calculating the Relative Social Pressure Metric (RSPM) from [10] based on historic contact information. In this example, node j would be calculating $\mathrm{RSPM}_{i,k|j}$ based on the interactions with i that immediately precede interactions with k.

(SPM) of v relative to u, denoted

$$\mathrm{SPM}_{u,v} = \frac{1}{T}\int_{t=0}^{T} f(t)dt$$

where T is the time frame in which contacts are considered, and $f(t)$ is how much estimated time remains until node u and v will next make contact. If $\frac{1}{\mathrm{SPM}_{u,v}} > \tau$ for some threshold τ, then v is in F_u. The second condition considers 2-hop contacts. If node v is a friend of u, and w is a friend of v, then node w and u are friends through v. To capture this, relative social pressure metrics (RSPM) are used, representing the average expected delay of message delivery if a message came from u, passed through v, and was destined for w.

$$\mathrm{RSPM}_{u,w|v} = \frac{1}{T}\sum_{x=1}^{n}\int_{0}^{t_{a,x}} (t_{b,x} + t_{a,x} - t)dt$$

Here, n is the number of times a (v, w)-contact immediately follows a (u, v)-contact during a time frame T. Refer to Figure 2.1 for a visual example of $t_{a,x}$ and $t_{b,x}$. Of these consecutive-contacts, $t_{a,x}$ is the duration of time between the x-th and $(x + 1)$-th (v, w) contacts, and $t_{b,x}$ is the time between the (u, v) end and the x-th (v, w) beginning for the x-th pair. For w to be considered an indirect close friend of u, and thus included in F_u, both $\frac{1}{RSPM_{u,w|v}} > \tau$ and

$\frac{1}{SPM_{u,v}} > \tau$ must be satisfied.

$$F_u = \left\{v|\frac{1}{\mathrm{SPM}_{u,v}} > \tau\right\} \cup \left\{w|\frac{1}{\mathrm{RSPM}_{u,w|v}} > \tau \text{ and } \frac{1}{\mathrm{SPM}_{u,v}} > \tau\right\}$$

With these communities known within each node, a node u carrying a message destined for node d will hand off its message to a neighbor node v if and only if (1) v and d are friends (*i.e.* $d \in F_v$) and (2) v is a closer friend to d (*i.e.* $\frac{1}{\mathrm{SPM}_{u,d}} < \max\left(\frac{1}{\mathrm{SPM}_{v,d}}, \frac{1}{\mathrm{RSPM}_{v,d|*}}\right)$ for any RSPMs involving v as the source friend).

When timely delivery of messages is important, the time in which a community assembles is crucial for making community-based routing decisions. The above-mentioned Friendship Based Routing [10] did indeed divide historic training data by the time frame in which they occurred. Each friendship communitity is thus time-dependent and stored for later querying. Thus, a node carrying a message can ask a connected neighbor which communities they are members of, and make a forwarding decision knowing the receiving node will likely meet the destination before the message expires.

2.1.3. Friendship. As social beings, people form strong friendships that result in regular encounters and longer times spent within close proximity. These traits are desirable when messages are being forwarded in a DTN. If a node is friends with the destination of a message, which is carried by someone they encounter, the exchange of the message to the destination's friend would intuitively strengthen the possibility of successful delivery. Several investigations have acknowledged this, leading to attention placed on how well friendship graphs improve DTN message forwarding. For instance, Socievole et al. [97] found the degree centrality of an individual's Facebook profile is highly correlated with their degree centrality from physical proximity. This finding suggests that a DTN could be bootstrapped for immediate routing decisions by using an online social network to calculate degree centralities of participating nodes. Their findings also suggested that betweenness centrality, eigenvector centrality, and closeness centrality had little to no correlation between the social networks. Thus, these metrics would need to be calculated within the confines of the DTN in order to be used.

In [98], an analysis of multiple social network graphs is performed to determine which social network graph has promising influence on successful forwarding. Similar to the motivation of [97], the goal in their analysis is to rely less on long-term learning of social patterns through opportunistic connections, and instead bootstrap the network by using social information that exists in other social networks. Their analysis considers two networks: the social graph formed from the Facebook friend lists of participating nodes, and a graph formed from the nodes' explicit interests. Their findings suggest that centrality measurements from the opportunistic network are still needed, but the addition of other social network graphs improves successful delivery when compared to relying solely on centrality measurements. Specifically, the explicit interests of participating nodes can be used to predict future contacts in the network. Likewise, a very social node who has a message's destination as an explicitly listed friend is very likely to meet with that destination. Centrality is still found to be the most important factor in making opportunistic forwarding decisions, but complementing centrality with other social network information boostes a node's decision making abilities.

Kim and Han [54] propose the Multi-path Routing for Heterogeneous-sized data (MRH) router that considers both the size of the messages being forwarded and the type of relationships between encountered nodes: whether they are friends or just opportunistic encounters. Friendship is determined based on the length of the contact; contacts that are too short are considered pass-by contacts, suggesting the encounter was not between two friends; conversely, contacts that are sufficiently long are considered to be between friends. Given these contact durations, nodes will compute and update three utility scores over time: a node's social tie score with a particular node is calculated by the average multi-hop minimum social contact duration between the two nodes; a node's social popularity score is calculated by the average of all friendship-based contact durations of the node; and a node's non-social popularity score is calculated by the average of all pass-by contact durations. Using these, the MRH router will forward smaller messages to pass-by contacts that have a higher non-social popularity with others. For larger messages, a contact between friends will result in a message being forwarded if the neighbor has either a higher social tie to the destination or, both having equal social ties, if their social popularity is higher.

2.1.4. Social Interests and Profiles. Apart from a friendship graph, online social networks also offer other details forming a *social profile* of a particular individual: where they live, what topics interest them, their favorite music, etc. The intuition behind using these social profiles is based on the notion that people will tend to meet others with shared interests more often than they would with dissimilar interests. These personal details have been investigated in a few works to see if they improve a node's ability to make forwarding decisions in a DTN. Primarily, when a message-carrying node encounters another node, a comparison between the two's social profiles is performed. This comparison would then result in the message-carrying node deciding if the encountered node should receive the message.

In [114], the social tie strength between nodes is considered in a two-phased broadcast scheme with the objective of decreasing delivery latency without incurring flooding-based resource consumption. During the first phase, called *Weak tie-driven forwarding*, nodes who act primarily as bridges between communities (who are identified based on their weak social ties) spread a message to other bridges. When message-holding nodes estimate that the message has sufficiently been seeded across the network, the scheme switches into the second phase, called *Strong tie-driven forwarding*, where message holders forward the message to very popular nodes so as to quickly spread the message to all nodes within its community.

The social tie strengths used in [114] are calculated for each node-pair based on a linear combination of the number of contacts between the two nodes and their social profile similarities. The contact numbers and the social profile information are recorded and exchanged during a training phase of the network. Social profile similarities are defined as the number of differences between two individuals in specified categories, such as favorite food, home town, employer, etc. For example, a node *u* with the social profile ⟨*female, Paris*⟩ and a node *v* with the social profile ⟨*female, London*⟩ have a social distance of 1, whereas *u* and another node *w* with the social profile ⟨*male, New York City*⟩ would have a social distance of 2.

In SANE [71], unicasting and interest-casting is accomplished through the usage of social profile comparison. In order to represent an individual's social profile, Mei et al adopt a vector representation $\langle v_0, v_1, \cdots, v_m \rangle$, where, among m unique social descriptors, the value v_k represents a binary weight to which the kth descriptor describes the individual. For instance, consider Bob, who lives in San Francisco and likes Chinese food, and Linda, who lives in Boston but doesn't like Chinese food. The adopted approach to represent their social profile would first start with assigning indices to each descriptor. In the example's case, the index 0 would be assigned to *Lives in San Francisco*, 1 to *Lives in Boston*, and 2 to *Likes Chinese Food*. Then, Bob's social profile could be represented as $\langle 1, 0, 0 \rangle$, and Linda's social profile as $\langle 0, 1, 1 \rangle$.

The forwarding decision adopted in SANE is based around how similar an encountered node's social profile is to the topics that describe the message under consideration. Similar to each participating node having a social profile of the form $\langle v_0, v_1, \cdots, v_m \rangle$, so too does each message have metadata of the form $\langle v_0, v_1, \cdots, v_m \rangle$. When a node u, carrying a message d, meets a node v, u calculates the cosine similarity the social profile of v and the message metadata of d, which is the cosine of the angle formed by the two vectors if they were vectors in an m-dimensional space. If this angle is larger than some minimum threshold, u will forward the message to v. For the interest-casting approach, where many nodes in the network wish to receive messages of topics that interest them, the cosine similarity would represent how interested the individual is in the message. In the unicasting approach, the metadata of a message is the social profile of the message's destination. Thus, if an encountered node has a high similarity value to the message, it is predicted that this individual is more likely to meet up with the destination than some other individual with a much lower similarity.

The Social Identity-aware Opportunistic Router (SIaOR) [112] considers both the social profiles of nodes and their social influences when making routing decisions. Social profiles are generated not solely from a user's online social media profile, but rather are dynamically generated based on the metadata of the content they produce, such as that stemming from object recognition in photos or significant keywords in messages. Social influence is computed using the nodes' encounter ratio to each other relative to all of the encounters for a particular node and their trust degree – the ratio of the number of messages a node forwards from its neighbor over the number of messages it received from its neighbor. When nodes encounter one another, they update their social influences, either growing or decaying depending on the new trust degree toward their neighbor, and compute a routing utility relative to each carried message as a power-law weighted product of the potential recipient's social similarity to the message's destination and their own social influence on their neighbor. Should the neighbor have a higher utility score for the message, the carrier will forward a copy.

2.1.5. Geographic Routing. Beyond social network based approaches, geographic information has been investigated as a means for DTN routing. Generally speaking, geographic routing through a network focuses on the specific locations of sources, intermediate relays, and destinations [56]. Of course, this requires that this knowledge be available. Within the confines of a DTN, however, a node carrying a message destined for a destination may be unable to look up their location when an opportunistic contact occurs, and any a priori information on the destination's whereabouts may be inaccurate due to their unobservable movements. Zhang et al [129] propose the Mobility Prediction-based Routing (MPR) protocol, which models the probability that a node will move from one area to another, and stay within that region for a given period of time, by a time-homogeneous semi-Markov process. Using this process, a message-carrying node will forward a message to the encountered neighbor with the highest probability of encountering the message's destination for as much time as is needed to successfully transfer the message. MobiT [123] is proposed for trajectory-based routing in Vehicular DTNs, where it is assumed that there exists some infrastructure for message delivery such as moving service vehicles and static road-side units (RSUs). The service vehicles collect the trajectories of participating vehicles in the network and centralize them at RSUs, which then use the trajectories to make routing schedules for messages. Messages are delivered through the network over participating vehicles in such a way as to deliver messages to RSUs located where a destination is traveling to before the destination arrives.

Many of the state-of-the-art works for DTN-based geographic routing have, in one way or another, extended the Spray-and-Wait (SaW) algorithm [100]. The most common characteristics of these routers is that the number of message copies that are permitted in the network is capped at some pre-determined quantity, after which no further copies are permitted to be created, and the routing algorithms operate in two phases: the first phase consists of the *spraying* of messages to as many nodes as possible, and the second phase consisting of discriminatory forwarding of the remaining message copy to qualified nodes, dropping the message from the sender's buffer, and traversing through the network until either the message arrives at its destination or the message is dropped, either to make room for more important messages or to delete expired ones. Zhang et al [127] propose the Speed Adaptive Multi-Copy Routing (SAMCR) algorithm that follows this paradigm for Vehicular DTNs (VDTNs) with sparsely-available, statically located road-side units (RSUs). SAMCR considers an encountered neighboring node to be qualified if the node has a greater relative velocity. Other works, like those that follow, consider geographic information beyond simple velocity.

Cao et al [13] propose the Geographic-based Spray-and-Relay (GSaR) router to use historic locations, velocities, and encounter rates of nodes. Each node is assumped to have initial historic information pertaining to location, velocity, and encounter rates of some nodes throughout the network. This information bootstraps the network, and is progressively updated and further disseminated to others opportunistically. When a node comes in contact with a neighbor, it will iterate over its messages in an order dependent on whether location information of the router is available: in its absence, the order is based on the age and number of remaining replicas available for the message (younger messages with more remaining replicas are favored), and routing operates exactly as specified in Spray-and-Wait. Should the location information be available, the message ordering is based on the likelihood that the traveling encountered neighbor will encounter the destination, based on projections of movement, before the message expires (higher likelihood results in higher priority). To compute this, the message carrying node computes the possible range of movement that the destination for each message could have traveled since its location was last recorded, given the historic speed of the destination. Then, given this range and the traveling direction of the encountered neighbor, half of the permissible message replicas will be handed off if the neighbor will be within this range faster than the message holder.

Similar to GSaR, the Location-Aided Controlled Spraying (LACS) router is proposed by Hang et al as another extension on Spray-and-Wait that considers location information [39]. Again, it is a two-phased router split between a *spraying* phase followed by a decisive single-copy forwarding strategy. When a message carrying node encounters another node, the expected remaining throughput of the channel is computed based on the other node's relative velocity, channel transmission speed, and the distance between the two nodes. Using this, messages are iterated over, with one being skipped if it cannot be transmitted during the contact's remaining time. If a message can be transmitted, the node's router decides if it will be transmitted. For messages with remaining copies left, transmission is carried out, with half of the messages being transmitted if the nodes are traveling in opposite directions (in a binary-spraying manner) or only one message copy if the nodes are traveling in a similar direction (single-copy relaying manner). However, when only one message copy remains, it would be relayed to the neighbor only if the neighbor is expected to encounter the destination sooner than the carrier. This is computed based on a semi-Markov model that is used to predict the location of the destination node, along with the velocity of the two nodes.

2.2. SIMULATING DTNS

To demonstrate the effectiveness of a DTN forwarding scheme, many researchers have simulated a DTN and gathered statistics from the simulations, such as those outlined in Section 1.4. These simulations require input datasets that describe the time-evolving nature of the network topology. This can come from two sources: real-world datasets or synthesized datasets. In this section, available datasets and models for synthesizing the data needed to simulate a DTN are listed and described.

2.2.1. Contact Traces. When it comes to characterising real-world datasets usable for DTN simulation, the majority of datasets available are contact traces. A contact trace is a dataset that includes records pertaining to when, and between whom, a contact has occurred. These datasets have typically been produced by dispensing a wirelessly-enabled device to many individuals that periodically broadcast a beacon to all neighboring devices, who in turn reply to announce their presence. The contact traces available have been generated through neighbor detections using the Bluetooth protocol, which has an intended range of 10 meters for most mobile phones and Bluetooth accessories [116].

One of the most widely used datasets is the MIT Reality Mining Dataset [29, 31]. In the Reality Mining dataset, data from approximately 100 participants, a mixture of students and faculty, was collected over a nine month period. Each participant was given a Bluetooth-equipped smartphone with special software to regularly collect contextual information. This data included proximity of nearby Bluetooth devices, phone application usage, call and text message logs, the ID of the nearest cellular tower and the status of the phone, each collected once ever 6 minutes. Additionally, each participant was surveyed for information pertaining to perceived friendships, research group affiliation, personal attributes, academic status, their home's neighborhood, and details on their lifestyles.

The Haggle Project [92] produced four contact trace datasets from different environments, three of which are applicable for simulating a DTN: the Cambridge dataset, and the INFOCOM 2005 and 2006 datasets. The records in these datasets were gathered from Bluetooth-equipped iMotes that recorded the MAC addresses of the devices responding to a node's periodic beacon. Each of these devices is said to have an approximate wireless range of 30 meters, with some more powerful iMotes having around 100 meters. In addition to having mobile iMotes carried by individuals, this project also installed stationary iMotes in points of interest such as within local pubs, commercial areas, and often frequented areas within a conference venue.

A recent shift in the data collected in these studies has bundled in social profile and online social networks with the proximity datasets. Kim and Gerla [53] generate a contact trace using two types of information commonly available from online social network services (in their case, Instagram): friendship lists, and geographic locations. First, the geolocations of 80 users was gathered from their posts to the social media site to build mobility patterns of each user. Then, for each user, the list of their followers is collected, with each follower receiving their own mobility pattern. From this collection of mobility patterns, a contact trace was generated considering two factors: a contact occurred if two users (i) one user was a follower of the other, and (ii) they both visited locations that were in close proximity to each other. The INFOCOM 2006 dataset [17], as part of the series of Haggle Project datasets, includes some basic social profiles of its participants, including the languages they speak, their current affiliations, the school in which they completed their studies, and a list of academic topics in which they are interested. The SocialBlueConn dataset [14] provides proximity data, Facebook friendship relations, and self-declared interests for 15 students at the University of Calabria over 7 days. The SIGCOMM 2009 dataset [78, 80, 81] contains the social profiles of the participants, along with providing a platform for users to create and disseminate messages over the DTN. These datasets have helped researchers evaluate DTN routing schemes to consider other factors influencing predictable social connections.

It should be noted that these datasets exhibit some flaws in that the results extracted from simulations employing them may not generalizable. This comes from the datasets' small sizes and the lack of diversity of the participants. Consider the INFOCOM 2005 and 2006 datasets and the SIGCOMM 2009 dataset. These datasets are composed of the attendees of an academic conference over the short period of 3 to 4 days. The mobility of these individuals is limited between the few closely-located rooms where talks are held and the corridors, which are often the congregation area for many attendees especially during coffee breaks. This environment results in a high density of people, thus bringing to question how partitioned the network is at any given time. Additionally, with every attendee having primarily academic backgrounds, there is little diversity in their social characteristics. So too is this a flaw of the MIT Reality Mining dataset. These issues encourage the exploration of other methods for simulating DTNs that offer controllable network parameters and diversity in their participant population.

2.2.2. Synthetic Mobility Models. The limited availability of real-world datasets, as well as the potential bias intrinsic to those available, limits the generality of results observed in proposed DTN routers. Most of the above mentioned contact traces have a small number of participants, all of whom serve various roles in an academic environment. Additionally, with fixed datasets brings the lack of controllable parameters in the simulated environment. Thus, synthesis of datasets is also appropriate to widen the available expirements that can be conducted. Sensitivity analysis cannot be conducted using these datasets to observe, for instance, the effect of the scaling up the number of participants on the successful delivery of messages.

Many simulations on DTN performance have relied upon simple mobility models for participating nodes. Rhee *et al.* [85] argues that many of these mobility models, such as Random Waypoint, Random Walk, and Brownian Motion, do not accurately reflect true human mobility even on an aggregated statistical level. Through their analysis of fine-grained GPS trajectories from over a hundred volunteers, they find the flight-time, pause-time, and intercontact-time distributions of these trajectories show similarities to those seen in spatially-bounded Levy walks. They then suggest the use of a Levy-walk based mobility model as a means of simulated node movements in a MANET and DTN environment. Levy walks are also used by [10]. Here, the nodes would have an ordered list of other communities that they would visit. They would travel from one community to another with some random speed, and when inside the community would meander about using a random walk, thus mimicking a Levy walk. When a randomly selected time passed, they would move on to the next community. They also incorporate a probability that a node would deviate from their itinerary and visit some other community. Once the deviation from their itinerary is complete, the node would return to their normal path. However, Bulut et al. acknowledge their suggestion lacks the spatial-temporal characteristics and social contexts that influence actual human movement.

To address these short-comings, the Small Worlds in Motion (SWIM) mobility model [57] has been proposed to mimick mobility patterns in people. The SWIM model is designed on the intuition that people consider two conflicting factors when deciding where to go: the location's distance from home, and the location's popularity. To calculate a node's movement, a two-dimensional plane is split up into equal-sized squares representing locations of interest. Each node has assigned to it a home location, and each node computes a weight for all other locations in the plane that is used as the probability of visiting that location. This weight depends on the location's distance and the location's popularity as observed by the node during its most recent visit to that location. Each person has their own preference, either preferring to travel to popular locations regardless of distance, or preferring to stay local, which is built into a node's calculation of each location's weight. When a node chooses a location, it travels at a speed proportional to its distance and, upon arriving, calculates the waiting time via a truncated power-law distribution. This waiting time reflects the tendancy for people to spend a lot of time at a few locations, and short periods of time at many locations.

One shortcoming of the SWIM mobility model is that it appears to reflect movement without a strict schedule. During the weekends, people may be more free to choose their next locations, but during the work week a node is more likely to travel from home to work. This shortcoming of SWIM is addressed in the Home-cell Community-based Mobility Model (HCMM) [9], which is designed mimick the cyclical, *home-to-work-to-activity-to-home* nature of human mobility. This model is similar to the SWIM mobility model with the exception that one location other than a node's home has a high probability of being visited, and being stayed at for a long duration. The probability of another location being the next selected waypoint is nonuniform, dependent on the distance to that location from the current location as well as a node's social connection to that location.

2.2.3. GPS Trajectory Datasets. Another type of dataset useful for DTN simulations are those providing the geographic trajectories of individuals and vehicles. This type of dataset is characterised by timestamped location records, where each record in a trajectory lists who is where – their geographic location, such as a GPS latitude and longitude – at what time. By going over the trajectories of many individuals, a contact trace can be constructed based on the proximity of individuals at a given time. When two individuals are within some radius of each other, a contact would occur between them.

Microsoft's GeoLife GPS trajectory dataset [132, 133, 134] is one such dataset. In this dataset, the locations of 182 users over a period of five years was recorded by GPS recievers carried as the participants traveled in the city of Beijing, China. This dataset contains 17,621 trajectories covering 1,282,951 kilometers over a period of 50,176 person-hours, with 91.5% of the trajectories having sequential records every 1 to 5 seconds or every 5 to 10 meters [131]. Thus, this dataset provides a fine granularity in time and spatial movements.

Nokia's Mobile Data Challenge (MDC) [55, 58] dataset is another dataset providing GPS trajectories of users. It similarly captures the trajectories of 185 individuals over a period of a year and a half in the city of Lausanne, Switzerland. Considering a trajectory to be a consecutive sequence of locations of a particular user where no two consecutive points are more than 10 minutes apart, the MDC dataset provides 761,463 trajectories covering 1,795,349 kilometers over 46.5 person-years. The participants were recruited via snowball sampling in two phases: the first phase consisted of an initial set of volunteers, and the second phase added individuals with a social association to those in the first phase. Ultimately, the resulting dataset is comprised of 38% women and 62% men, with roughly two-thirds falling within the 22-33 year-old age group [58]. In addition to GPS trajectories, the dataset also provides rich metadata such as the participants' demographics, call and text message records, phone book and calendar entries, mobile application usage, and observed Bluetooth encounters.

2.3. UNADDRESSED PROBLEMS

Most of the proposed works discussed have focused on exploiting gathered knowledge in deciding when messages should be forwarded. This knowledge is either pre-loaded on a node at the ignition of a DTN or is gradually learned over time as a node encounters others and exchanges information. The information gathered are primarily artifacts of the social constructs of the people carrying each device. Based on the comparisons conducted on these new solutions and on solutions designed for the more general DTN, exploting social properties is shown to improve performance more than relying solely on opportunistic contact metrics (e.g. contact frequency, intercontact time, and contact duration). However, there remains some challenges that have remained unaddressed in the field of DTN functionality.

2.3.1. The Unaddressed Challenges of a Sparse DTN. The datasets described in Section 2.2.1 provide a glimpse into the real-world social interactions of the people contributing contact records. These datasets have been the primary driver of evaluations and comparisons of many new DTN solutions. However, the results extracted from simulations using these datasets are not generalizable. This comes from the datasets' small sizes and the lack of diversity of the participants. With the INFOCOM 2005 and 2006 datasets and the SIGCOMM 2009 dataset, the population of participants is small and homogeneous in their professions, and the connectivity of the resulting network may be quite high with their activities being confined to a conference venue. In other words, there is concern of the density of these networks.

A notable challenge that has seemingly been untested is the effects of network sparsity. In this context, sparsity refers to the extremely low occurrence of contacts between pairs of nodes over time. This may be attributed to short-range wireless communication, low participation of individuals relative to an area's population density, or a wide geographical distribution of participating individuals. For instance, the GeoLife dataset [132, 133, 134] exhibits sparsity due to its participants being widely scattered across the city of Beijing. Even though Beijing is an area with a high population density, those individuals who provided their trajectories are often too far away from even the closest neighbor to form a connection with one another. To the best of our knowledge, only one previous study [72] has test their DTN routing system using the GeoLife dataset.

2.3.2. Congestion and Network Degredation. Another challenge is the handling of network congestion within a DTN. Here, congestion is considered the full consumption of resources that ultimately leads to higher delivery latencies, lower successful deliveries, and more wasted resources on relaying messages that otherwise would not be successfully delivered [94, 96, 103]. Most works have assumed the sufficient availability of resources such that congestion is never experienced. There are numerous factors that lead to congestion, some of which are typical in sparse networks: slow transmissions, shorter-range communication, small storage buffers, larger message sizes, higher-frequency message creation, and fewer nodes available to relay messages to name a few. I postulate that there exists a horizon in the space of network properties that defines a boundary between normal and degraded performance. As of now, it remains unidentified.

The possibility of congestion arising warrants the investigation of strategies to avert or properly tolerate it. One such strategy is to only forward messages to others that have a high likelihood of reaching their destinations through the selected relays. As was discussed earlier in this section, many systems have been proposed for this. Be that as it may, these proposals have not explicitly been evaluated under congested conditions. Other strategies may be adopted to augment said systems. Buffer management can be employed to alleviate degradation through strategically removing messages from full buffers, thus making space for those that are predicted to be deliverable. Replication control can prevent saturation in storage buffers and transmission channels by capping the number of message copies that can exist at any time in the network. For messages that exhibit redundancy – not in the terms of redundant message copies, but in terms of exceedingly similar information being created by independent sources – a redundant message dropping or consolidation/aggregation system can reduce the amount of space occupied and the amount of bandwidth consumed for that information to be delivered.

PAPER

I. CHITCHAT: AN EFFECTIVE MESSAGE DELIVERY METHOD IN SPARSE POCKET-SWITCHED NETWORKS

D. McGeehan, D. Lin, S. Madria

ABSTRACT

The ubiquitous adoption of portable smart devices has enabled a new way of communication via Pocket Switched Networks (PSN), whereby messages are routed by personal devices inside the pockets of ever-moving people. PSNs provide opportunities for various interesting applications such as location-based social networking, geolocal advertising, and military missions in active battlefields where the central communication tower is unavailable. One key challenge of the successful roll-out of PSN applications is the difficulty of achieving high message delivery ratio due to the dynamic nature of moving people and spatial-temporal sparsity in such networks. In this paper, we propose a novel message routing approach, called ChitChat, which exploits users' direct and transient social interests via discriminatory gossiping to penetrate messages deeper into the network. Our approach enables message carriers to make opportunistic and distributed routing decisions based on the likelihood a potential message receiver will meet individuals that have a high chance to forward the message to the destination. Our experimental results have demonstrated that our approach achieves higher delivery ratios against the two more recent state-of-the-art algorithms, while maintaining a lower communication overhead against flooding and reducing the amount of time messages remain idle in buffers.

1. INTRODUCTION

The ubiquitous adoption of portable smart devices (e.g., smart phones and smart watches), along with the provisioning of technology to allow long-range device-to-device communication as proposed in emerging 5G specifications [5], provides an opportunity for exploring Pocket Switched Networks (PSN) [6, 13, 17]. A PSN is a type of Delay Tolerant Network (DTN) in which messages are routed by personal devices carried inside the pockets of ever-moving people. Figure 1 portrays this type of network environment. As people move about, their mobile devices connect with one another when they are within wireless communication range, and data packets may be exchanged while the connection lasts [7, 17]. The challenge that lies in effectively utilizing these mobile devices as message disseminators results from the volatility and sparsity in network connectivity, primarily due to limited resources, low node densities, long inter-contact times between nodes, and security restrictions on message passing.

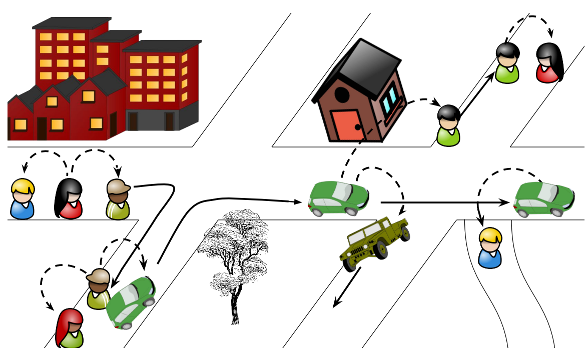


Figure 1. An example of a Pocket Switched Network. Solid arrows indicate movement of device holders, and dashed lines indicate established connections and message forwarding.

For example, soldiers in an active battlefield carrying wireless-equipped tablets can act as nodes in a temporary ad-hoc network in case the primary communication network is destroyed. However, the ad-hoc network may suffer extreme congestion due to the high velocity of information and the frequent disconnections between exchanging devices.

Each connection can only transport a finite quantity of information, and thus to prevent catastrophic congestion, a message should only be routed through a connection if it will likely reach its intended destination because of the connection.

Another example considers a commercial application of PSNs, whereby businesses broadcast sale advertisements and coupons to patrons passing near their storefronts. However, individuals interested in the business's products or services may not frequently visit the store's location. Users are willing to accept and distribute these advertisements to their friends and neighbors to increase the chance they will receive the ones they are interested in, but sophisticated forwarding strategies must be adopted so as to not burden the users with depleted battery life and not disengage them with copious amounts of irrelevant information.

One key challenge of PSNs is the difficulty of achieving high message delivery ratios with finite lifetimes [6, 13, 18]. We believe this challenge is exacerbated in networks characterized by spatial-temporal sparsity. Spatial-temporal sparsity refers to the sparse connections among participating nodes in a PSN primarily due to (1) the geographic distances between each other, (2) the duration of time between connection opportunities, and (3) security or non-cooperative nature of nodes carrying messages. Moreover, in a PSN, there is no prior knowledge as to when connection opportunities will arise for messages to make the next hop along their journey. When an opportunity does occur, it is also challenging to make the optimal decision on whether to forward messages. Numerous constraints limit how many messages to forward, including varying bandwidth, limited buffers within each device, and restricted lifetime of messages.

Although there have been many previous algorithms for data dissemination in PSNs [1, 2, 4, 6, 7, 10, 11, 13, 17, 18, 23], none of them tackles the sparsity challenge that exists in a real world PSN environment. Existing works assume high node densities in a PSN such as a concentrated meeting environment [3, 15, 16], which limits their adoptions in a wider range of applications.

In this paper, we propose a novel message routing approach, called ChitChat, which is designed to conquer the spatial-temporal sparsity problem in PSNs. Our approach leverages an important characteristic of PSNs, which is that the devices are carried by people whose movements are guided by their social interests, roles, and responsibilities. Our proposed ChitChat system implements sophisticated strategies for nodes (mobile devices carried by people) to exchange, aggregate, store, and disseminate the social profiles of encountered individuals, allowing individuals to utilize this increasingly rich information to judge whether a current opportunistic channel has high likelihood to lead to a successful message journey.

More specifically, when two nodes connect with each other, they *chitchat* to exchange the following two kinds of information: (i) *direct social interests*, which is the metadata, or the set of keywords, that describe the encountered node's interests, roles, and responsibilities (e.g. social interests such as “photography" and “gourmet cooking", or role-specific metadata such as “MANET researcher", “military intelligence officer"); and (ii) *transient social relationships*, which is aggregated information of the social interests of the people that the node has encountered before. Our approach allows the social interests to be dynamically expanded, refined and aggregated in real-time into the richer transient social relationships so as to capture multi-hop relationships.

For example, if Alice meets Bob reliably, and Bob meets Janet reliably, then Alice's transient social relationships should be influenced by Janet's even if Alice and Janet have never met and have no common interests. To more precisely model the weight of multi-hop relationships, we have carefully designed a temporal growth and decay model that ensures the freshness of these transient social relationships while also maintaining an upper bound on storage complexity. Our experimental results on a real world dataset verify that the use of such multi-hop relationship modeling helps achieve significant improvement on message routing in sparsely-connected PSNs.

The contributions of this paper are summarized as follows:

- We propose the ChitChat routing algorithm to address the impact of spatial-temporal sparsity on message delivery in PSNs. Our approach is able to facilitate nodes in a PSN to make intelligent decisions on which nodes to forward a message copy, even in situations where the journey between the source and destination has high social network distance.
- We propose a novel way of modeling and maintaining multi-hop transient social relationships by taking into account the time durations of connections and disconnections. This model maintains the accuracy and freshness of social relationship information residing within each node, building a solid foundation for making informed message routing decisions to overcome the constrained connectivity in sparsely-connected PSNs. It also permits users to change their social interests through time while still participating in the network.
- We evaluate the performance of the ChitChat protocol by using a real dataset, the GeoLife dataset [20, 21, 22], which provides diverse and fine-grained human trajectories over a period of five years. We compare the performance of ChitChat against two recent state-of-the-art PSN algorithms [10, 13], which have been selected as they perform better than many others. The experimental results demonstrate that our proposed ChitChat achieves better message delivery ratios without flooding-level communication overhead.

The rest of the paper is organized as follows. Section 2 presents a formal definition of the Pocket Switched Network environment, along with our proposed ChitChat protocol. Section 3 reports the experimental results. Section 4 reviews existing message routing protocols in PSNs. Finally, Section 5 concludes the paper.

2. CHITCHAT ARCHITECTURE AND ROUTING PROTOCOLS

In this section, we first present the problem statement, and then provide an overview of our proposed ChitChat system. After that, we elaborate the detailed algorithms that reside within the ChitChat system.

2.1. PROBLEM STATEMENT

In this work, we consider the following characteristics of a PSN. Nodes in a PSN are users who have smart pocket devices that are equipped with the ChitChat system, which automatically connects to other devices that are within the communication range. Each user has his/her own social profile as defined in Definition 1 and 2. Accordingly, when a user sends a message, the message is annotated with appropriate metadata keywords (a subset of all social interests) that describe the topic or content of the message. Figure 1 provides a diagram of how a PSN operates.

Definition 1. (Social Interest). A social interest is represented as SI⟨SID, kw⟩, where *SID* is a unique ID to identify the social interest and *kw* is the keyword that describes the content of this social interest.

Definition 2. (Social Profile). Let u be a user in a PSN and SP_u be his/her social profile. The social profile of user u is a set of social interests, i.e., $\mathrm{SP}_u\langle \mathrm{SID}_1, \mathrm{SID}_2, \cdots, \mathrm{SID}_k\rangle$.

For example, suppose that there are three types of social interests: $SI_1\langle$001, ‘hiking’$\rangle$, $SI_2\langle$002, ‘photography’$\rangle$, $SI_3\langle$003, ‘gourmet cooking’$\rangle$. If Alice’s social interests include ‘hiking’ and ‘gourmet cooking’, her social profile can be represented as $\mathrm{SP}_{alice}\langle$001, 003$\rangle$. When Alice visits her favorite outdoor equipment store and learns of their coupons and sales, she may pass along these coupons to her friends when they next meet.

In considering the military application of PSNs, a patrolling soldier may become aware of the movement of enemy tanks and the location of ammunition stockpiles. In this situation, the social interest would be ‘ammunition location’ and ‘tank movement’, and a message can be reported back to Ground Intelligence Officers with expressed roles for tracking ammunition stockpiles and tank movements.

The proposed ChitChat system supports the following unicast and multicast scenarios in a PSN environment.

- *Unicast*: A sender sends a message to a designated message receiver. In this case, the sender knows the message receiver’s ID and annotates the message to reflect content the recipient would be interested in.

- *Multicast*: A sender sends a message to a group of users whose social interests match a message’s metadata keywords. In this case, the sender does not need to know the message receivers’ IDs.

Messages are distributed in a multi-copy fashion. When a node connects with a neighbor, the node may create a replica of a message he/she is carrying and forward that replica to the neighbor. The sending node retains a copy of the message, only deleting it when its time to live (TTL) has expired, when the node successfully delivers a copy of the message to the destination, or when its buffer is saturated and room is needed for other messages with better delivery potential.

2.2. AN OVERVIEW OF THE CHITCHAT SYSTEM

Our ChitChat system consists of two major components with associated storage buffers as shown in Figure 2: (i) Real-time Transient Social Relationship (RTSR) modeling; and (ii) Message Routing. The overall data flow in the ChitChat-equipped PSN is as follows. When two users come within communication range, the ChitChat system will first invoke the RTSR module. The RTSR module will automatically exchange the two users’ current

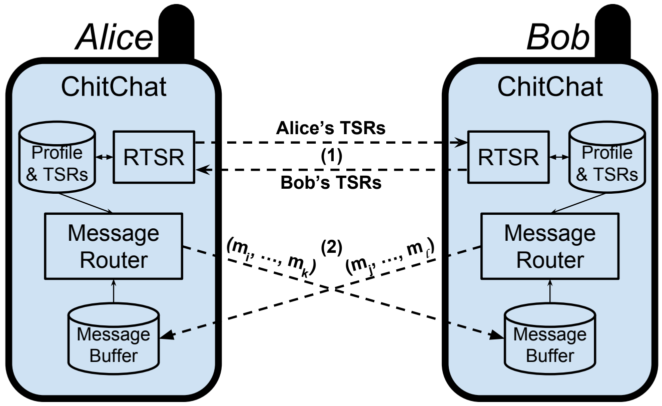


Figure 2. Architectural diagram of the ChitChat system within each node. The ChitChat system consists of two components, each backed by an associated datastore: the Real-time Transient Social Relationships, and the Message Router. Here, Alice and Bob connect with one another, initiating the ChitChat system. (1) Alice exchanges her Transient Social Relationships (TSRs) with Bob, and vice versa. Once this chitchat is completed, (2) Alice decides which messages (if any) to forward to Bob and builds a bundle out of her messages $\langle m_i, \cdots, m_k \rangle$. Likewise, Bob builds a bundle out of his messages $\langle m_j, \cdots, m_l \rangle$. These bundles are then forwarded to each other, and added to the recipient's message buffer.

Transient Social Relationships (TSRs), resulting in an adjustment in their TSRs based on a growth-decay model as presented in Section 2.3. Then, the ChitChat system will invoke the message routing component to exchange a selected subset of messages carried by the two users based on the analysis results of their revised TSRs.

The Real-time Transient Social Relationship modeling aims to represent the evolution of each user's social interests impacted by the people that they encounter. The message routing can then select better message forwarders based on their Transient Social Relationships. For example, Alice is interested in gourmet cooking. She may have friends with the same interest, who also have friends with the same interest, and so on. It would then be promising to ask Alice to help forward messages (e.g. recipes, photographs, sales on special knives, etc) tagged with 'gourmet cooking' so that the messages will have a higher likelihood of reaching its designated receiver(s) with the same interest. A more complicated case is the following. Although Alice is not interested in 'photography', she may reliably encounter people in the network who have this interest. In this case, it may also be helpful to improve message delivery by having Alice as an intermediate carrier for messages tagged with 'photography'. In what follows, we elaborate the algorithms drivings these forwarding decisions.

2.3. REAL-TIME TRANSIENT SOCIAL RELATIONSHIP MODELING

As aforementioned, the Real-time Transient Social Relationship (RTSR) modeling function is invoked when two users are within communication range in a PSN. The two users will exchange their current transient social relationships (TSRs) as defined in Definition 3 and then adjust it to reflect their most recent multi-hop social relationships with other users in the network.

Definition 3. (Transient Social Relationship). The transient social relationship of a user u at timestamp t is represented as $\mathrm{TSR}_u\langle(\mathrm{SID}_1,\ w(SID_1, t)),\ (SID_2, w(SID_2, t)),\ ...,\ (SID_k, w(SID_k, t))\rangle$, where SID_i is the ID of a social interest and $w(SID_i, t)$ ranges from 0 to 1 that indicates the weight of the social interest SID_i at time t.

We now present how to obtain the social interests for the Transient Social Relationships (TSRs) and how to compute their weight. At the beginning of the whole network (denoted as timestamp t_0), each user's TSRs are the same as his/her social profile with the weight of each TSR set to the medium weight of 0.5. For example, given Alice's social profile $SP_{alice}\langle 001, 003\rangle$, her initial TSRs at timestamp t_0 will be $TSR_{alice}\langle(001, 0.5), (003, 0.5)\rangle$.

Later, when users encounter each other, they conduct the following three steps to adjust their TSRs: (1) each one computes his/her own latest TSRs based on a decay model; (2) exchange their TSRs; (3) each one computes the growth of his/her TSRs based on a growth model. Without loss of generality, we consider u and compute its current TSRs as follows.

Suppose that user u enters the communication range of some users at the same timestamp t_s where $t_s - t_0 \geq 1$. The ChitChat system will first compute the current weight of each social interest in each user's TSRs by using the decay function defined in Equation 1.

$$w_u(\mathrm{SID}_i, t_s) = \begin{cases} \dfrac{w_u(\mathrm{SID}_i, t_{d,i})}{\beta \cdot (t_s - t_{d,i})}, & \\ & \text{if } \mathrm{SID}_i \notin SP_u; \\ \dfrac{(w_u(\mathrm{SID}_i, t_{d,i}) - 0.5)}{\beta \cdot (t_s - t_{d,i})} + 0.5, & \\ & \text{if } \mathrm{SID}_i \in SP_u. \end{cases} \tag{1}$$

The intuition behind the above decay function is that the longer a user is disconnected to those holding a positive weight for the social interest SID_i, the less likely this person will be able to successfully deliver a message with this social interest. Specifically, Equation 1 considers two cases: (i) $\mathrm{SID}_i \notin SP_u$ means the social interest is not part of the user's social profile; and (ii) $\mathrm{SID}_i \in SP_u$ means the social interest is in the user's social profile. The decay equation ensures that the weight of social interests from the user's social profile will never decrease below 0.5. In both cases, $t_{d,i}$ denotes the latest timestamp that the user was connected with some user with a positive weight for SID_i, implying that at time $(t_{d,i} + 1)$ they disconnected. The time difference $(t_s - t_{d,i})$ is a positive integer representing the number of seconds that has passed since u was last in contact with another user with a positive TSR weight for SID_i. The longer the user remains disconnected from users with SID_i, the lower the TSR weight of SID_i will be, whereby $1/\big(\beta \cdot (t_s - t_{d,i})\big)$ is the factor by which the weight undergoes decay. The parameter β is introduced to adjust the speed of decay such that $\beta \geq 1$. It is worth noting that in the case when a user remains connected to someone with a positively-weighted TSR for SID_i, the value of $t_{d,i}$ is equal to t_s. In such case, there is no need to compute the decay for this social interest since it has been continuously reinforced up to t_s.

At timestamp $t_s + 1$, user u will exchange his/her positively-weighted TSRs, consisting of only the social interests with positive weights, with the users $v_1, \cdots, v_k$ who newly connected to u at time t_s. Likewise, each neighbor $v_1, \cdots, v_k$ will exchange their TSRs with u. They will not exchange TSRs again for the duration of their uninterrupted connection. Here, the timestamp $t_s + 1$ ensures that only users who stay in contact with each other for at least one time unit will be considered during the social interest growth phase. Then, the weight of each social interest SID_i in u's TSRs will be modeled as a function of the current timestamp t_c (s.t. $t_c > t_s$) according to Equation 2.

$$w_u(\mathrm{SID}_i, t_c) = \min\{1, w_u(\mathrm{SID}_i, t_s) + \Delta\} \tag{2}$$

$$\Delta = \sum_{v \in \mathcal{v}} \frac{w_v(\mathrm{SID}_i, t_s) \cdot (t_c - t_s)}{\psi_{i,u,v}} \tag{3}$$

In Equation 3, $w_u(\mathrm{SID}_i, t_s)$ denotes the weight of SID_i in u's TSR at time t_s; $\mathcal{v}$ denotes the set of users in the communication range of u at time t_c, and $w_v(\mathrm{SID}_i, t_s)$ denotes the TSR weight of a user $v \in \mathcal{v}$ at time t_s when they start their interaction. The min function ensures that the growth would not exceed the upper bound. The growth function takes into account three factors: (i) the users' social interest weight (i.e., w_u, w_v) at the beginning of their interactions; (ii) the duration of the interaction, i.e., $(t_c - t_s)$; and (iii) the appropriate growth dampening factor $\psi_{i,u,v}$ which, as detailed below, is dependent on whether SID_i is a social profile resident or an induced TSR of both u and v. Regarding the first two factors, the higher the TSR weight and the longer the users remain in contact increases the growth of the corresponding TSR weight, i.e., the more likely that a message annotated with that social interest will be delivered. As for the third factor $\psi_{i,u,v}$, its value is determined based on the residency of the social interest SID_i in u's and v's social profile. In particular, we identify the following six cases in descending order of their impact on the social relationships in a PSN, i.e., the movement of users in PSN is driven by their social interests, or users with similar social interests may gather together more often.

- $\psi_{i,u,v} = 1$: This refers to the case when the social interest SID_i is a direct social interest of both u and v, meaning SID_i is in both of their social profiles (i.e. $\text{SID}_i \in \text{SP}_u$ and $\text{SID}_i \in \text{SP}_v$). In such case, u's TSR for SID_i will have no dampening due to $\psi_{i,u,v}$, obtaining the most growth to reflect the high likelihood they can forward messages annotated with SID_i.

- $\psi_{i,u,v} = 2$: This refers to the case when SID_i is in user u's social profile but not user v's. Rather it is a TSR that user v obtained when interacting with others.

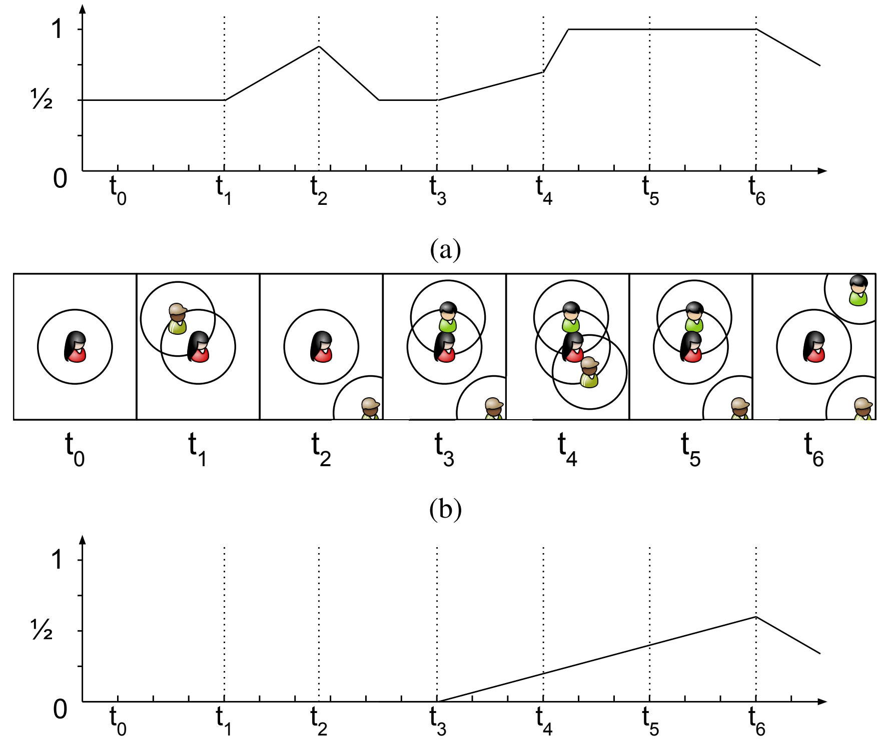
- $\psi_{i,u,v} = 3$: This refers to the case when SID_i is not user u's social profile, but is in user v's. Since SID_i is not a direct social interest for user u, the growth of its TSR weight is less than previous cases.

- $\psi_{i,u,v} = 4$: This refers to the case when SID_i is neither user u's nor v's social profile. Both u and v obtained this TSR from encountering others. This means there is some chance for user u and v to meet the people with this social interest, but the chance may be small since it is not these two users' direct interest.

- $\psi_{i,u,v} = 5$: This refers to the case when SID_i is in user v's social profile and is a weightless TSR in u (i.e. $w_u(\text{SID}_i, t_s) = 0$). In such case, user u will expand its TSR by including this new social interest with a relatively low initial weight.

- $\psi_{i,u,v} = 6$: This refers to the case when SID_i exists only as an induced TSR of user v (i.e. $\text{SID}_i \notin \text{SP}_u \cup \text{SP}_v$, $w_u(\text{SID}_i, t_s) = 0$, and $w_v(\text{SID}_i, t_s) > 0$). This is the weakest case, yielding the least likelihood of message delivery for messages with this interest, and hence it is given the highest dampener.

To have a better understanding of the algorithm, let us step through an example as shown in Figure 3.



(c)

Figure 3. Illustrative example for calculating a Transient Social Relationship (TSR) given connections and disconnections. (a) Alice's TSR corresponding to the interest 'hiking'. She has this TSR as part of her social profile, and thus its weight cannot go below 0.5. (b) This diagram demonstrates the connections between Alice, Bob, and Carl that influence the TSR weights in Figure 3a and 3c. (c) Alice's TSR corresponding to the interest 'photography'. She is not directly interested in traveling as it is not in her social profile. However, she interacts with people directly interested in it, and thus has a TSR with weight between 0 and 1.

Alice starts off with a social profile $SP_{alice}\langle 001, 003\rangle$ to express her interest in hiking and gourmet cooking. Thus, her TSR (shown in Figure 3a) begins at 0.5. She meets Bob at time t_1, who also has 'hiking' in his social profile (i.e. $\mathrm{SP}_{bob}\langle 001\rangle$), which causes her 'hiking' TSR to grow according to Equations 2 and 3, with the dampening factor $\psi_{001,alice,bob} = 1$. At time t_2, Bob disconnects from Alice, resulting in her 'photography' TSR to begin decaying according to Equation 1. Enough time passes for her TSR to fall back to 0.5. Then, at time t_3, Alice encounters Carl, who is interested in hiking and photography

Table 1. Table of Symbol Descriptions

Symbol	Description
u	The node executing the algorithm
t_c	Current timestamp of algorithm execution
$t_{s,v}$	The timestamp when v most recently connected to u
$t_{d,i}$	The most recent timestamp when u was connected to a neighbor with non-zero TSR for SID_i
$\mathcal{V}$	All nodes connected to u at time t_c
$\mathcal{V}_{t_c}$	Nodes who newly formed a connection to u at time t_c
v	Some connected neighbor of u
SID_i	The unique identifier for the ith social interest
SP_u	The social profile of node u, defined in Def. 2
$w_u(\text{SID}_i, t)$	User u's TSR weight for social interest SID_i at time t
TSR_u	The Transient Social Relationships for user u (Def. 3)
β	Adjustable parameter to control rate of decay
$\psi_{i,u,v}$	Growth rate dampener, defined by Def. 3

(i.e. $SP_{carl}\langle 001, 002\rangle$). This causes Alice's TSR for 'hiking' (Figure 3a) to undergo growth with a dampener of $\psi_{001,alice,carl} = 1$. Likewise, a TSR for 'photography' is induced in Alice (Figure 3c) with a growth dampener of $\psi_{002,alice,carl} = 5$. At time t_4, Alice reconnects to Bob, causing a spike in the growth rate of her 'hiking' TSR. Eventually, it maxes out with a weight of 1. At time t_5, Bob disconnects from Alice. However, since Alice is still connected to Carl, her TSR for 'hiking' and 'photography' does not undergo decay. Only when Carl disconnects at time t_6 do both TSRs undergo decay.

Algorithm 1 summarizes the main steps of the Real-time Transient Social Relationship modeling.

Our modeling ensures the following properties.

Theorem 1. *Let u be a user and TSR_u be the user's Transient Social Relationships. For any $SID_i \in TSR_u$ and any timestamp t such that $t > t_0$, then $0 \leq w_u(SID_i, t) \leq 1$.*

Theorem 2. *Each node has a storage complexity of $O(N(b + \log N))$ for TSRs.*

Due to constricted space limitations, the full proofs for Theorems 1 and 2 are provided in [12].

Algorithm 1 Real-time Transient Social Relationship Modeling: A new connection is established between node u and nodes $v_{t_c} = \{v_1, \cdots, v_k\}$ at time t_c.

procedure CONNECT(u, v_{t_c})
▷ Decay TSRs according to Eq. 1
  for each $\mathrm{SID}_i \in \mathrm{SP}_u$ **do**
    if $t_{d,i} == t_c$ **then** ▷ No decay to $w_u(\mathrm{SID}_i, t_c)$
      $w_u(\mathrm{SID}_i, t_c) = w_u(\mathrm{SID}_i, t_{d,i})$
    else if $\mathrm{SID}_i \in \mathrm{SP}_u$ **then**
      $w_u(\mathrm{SID}_i, t_c) = \dfrac{w_u(\mathrm{SID}_i, t_{d,i}) - 0.5}{\beta \cdot (t_c - t_{d,i})} + 0.5$
    else
      $w_u(\mathrm{SID}_i, t_c) = \dfrac{w_u(\mathrm{SID}_i, t_{d,i})}{\beta \cdot (t_c - t_{d,i})}$
    end if
  end for
▷ Exchange current TSRs with new neighbors.
  for each node $v \in v_{t_c}$ **do**
    Send TSR_u to v
    Receive TSR_v from v
    $t_{s,v} = t_c$
  end for
▷ Grow TSRs according to Eq. 2
  for each $\mathrm{SID}_i \in$ all received and cached TSRs **do**
    $\Delta = \sum_{v \in v} \dfrac{w_v(SID_i, t_{s,v}) \cdot (t_c - t_{s,v})}{\psi_{i,u,v}}$
    $w_u(\mathrm{SID}_i, t_c + 1) = \min\{1, w_u(\mathrm{SID}_i, t_c) + \Delta\}$
  end for
end procedure

2.4. CHITCHAT ROUTING PROTOCOL

The message forwarding phase occurs after TSRs are updated. Since each node may hold a set of interest-tagged messages, and each node has limited message storage and forwarding capabilities, it is critical to determine which message to forward to which neighboring node so that the overall delivery rate is not penalized. To achieve this, we propose the following routing protocol (shown in Algorithm 2) that leverages the knowledge carried by the newly updated TSRs.

The routing protocol takes as input a user u who wants to forward a message msg that has social interests $msg.\mathtt{SIDs} = \langle \mathrm{SID}_1, \cdots, \mathrm{SID}_m \rangle$, and a set of users $\nu = \langle v_1, \cdots, v_n \rangle$ who are within user u's communication range and just completed exchanging TSRs with u. First, u will check if the designated message receiver (in the unicast case) is among the neighboring nodes. If so, u will pass the message only to the receiver and moves on to other messages. Otherwise, the protocol will proceed as follows. User u will rule out those neighboring nodes whose message buffer are full, and thus are unable to receive any more messages. For each remaining neighbor $v_k \in \nu$, user u will select all of v_k's TSRs that match msg's social interests, and then compute the sum of their weights (line 9 in the algorithm). Note that u will use the neighbors' TSRs as they were after their decay, but before their growth, to avoid recursive consideration. These TSRs have already been obtained by u during the RTSR phase (i.e. Algorithm 1) and no more communication is needed here.

Next, user u will compare the sum of the weights of the message's social interests (denoted as S_u) in its own TSRs (line 8) with that of the neighbor's (denoted as S_{v_k}). If $S_{v_k} > S_u$, the neighbor v_k may be more interested in this message or a more viable carrier for other interested users, resulting in user u forwarding the message to v_k.

Since we are considering a sparsely-connected network, it is possible that user u has very few neighbors and none of the neighbors' S_{v_k} is greater than S_u. In such case, so as to not penalize the message delivery rate, user u will still try to forward the message to the neighbor with the highest S_{v_k}, if all the following conditions are met:

- User u's buffer is full. That means if u does not forward the message msg now, u may not be able to take in new messages even if they are of great interest to u.

Algorithm 2 ChitChat Routing Algorithm: Node u decides to which connected neighbors in v to forward the message msg at time t

procedure SENDMESSAGE(u, msg, v)
    for $v_k \in v$ **do**
        if msg.`destination` == v_k **then**
            Forward msg to v_k
            u.`messages` = u.`messages` − msg
        end if
        if $msg \notin v_k$.`messages` **then**
            $S_u = \sum_{\mathrm{SID}_i \in msg.\mathtt{SIDs}} w_u(\mathrm{SID}_i, t)$
            $S_{v_k} = \sum_{\mathrm{SID}_i \in msg.\mathtt{SIDs}} w_{v_k}(\mathrm{SID}_i, t)$
            if $S_{v_k} > S_u$ **then**
                Forward msg to v_k
            end if
        end if
    end for
end procedure

- The time before msg expires is equal to its estimated delivery time, as shown in the following Equation 4, where t_{exp}^{msg} is the message's expiration time and $E(t_{del}^{msg})$ is the estimated delivery time based on statistic information. In this case, if u does not forward the message, the message will likely never reach its destination. Note that this condition is only needed in the unicast scenario.

$$t_{exp}^{msg} - t_c = E(t_{del}^{msg}) \tag{4}$$

- S_{v_k} in the selected neighbor must not be zero, implying this neighbor has at least some interest or delivery capability for this message. Otherwise, it could be a waste of energy and buffer space to forward this message to a user with no interest in it.

This edge case is considered only after message forwarding of Algorithm 2 results in no messages being forwarded.

3. PERFORMANCE EVALUATION

In this section, we first introduce the experimental settings and then report the experimental results.

All experiments were conducted in the ONE simulator [9] version 1.5.1 RC2. We compare our ChitChat system with one benchmark algorithm (i.e., Epidemic [18]), and two recent related works SEDUM [10] and SANE [13].

In the experiments, we evaluate the performance of each algorithm by using the GeoLife GPS trajectory dataset [20, 21, 22]. The GeoLife dataset consists of the trajectories of 182 users over a period of five years in the city of Beijing, China. Specifically, the dataset contains 17,621 trajectories covering 1,282,951 kilometers over a period of 50,176 person-hours [19]. Each trajectory consists of a temporal sequence of latitude and longitude points, recording how a user moved during the period of time their GPS device was active.

Although the GeoLife dataset provides the best available dataset for unbiased movements of people, each day over the entire five year period provides only trajectories for up to 28 users. Each of these users are scattered across the city of Beijing and rarely come within close proximity to each other. In order to form a network to evaluate and compare the aforementioned routing protocols, trajectories from the same person occurring on different days were gathered together and treated as unique individuals. For example, if Alice contributed a GPS trajectory for Monday, Tuesday, and Friday, then the modified GeoLife dataset used for these experiments would present three users moving about on the same day: AliceMonday, AliceTuesday, and AliceFriday. This view changes the GeoLife dataset from spanning five years with 182 unique participants into a dataset spanning one day (24 hours) with 9,797 unique trajectories. From this, we then isolate the trajectories consisting of at least two hours of contiguous recordings and at least 500 unique location records, which constitutes approximately 63.2% of the unique trajectories (6,193 out of 9,797 unique trajectories). The resulting dataset provides a node density of approximately

Table 2. Default Experimental Settings

Configuration	Default Experimentation Values
Number of Participants	2,000
Freq. of Location Reporting	1/5 sec^{-1}
Pool of Social Interest Keywords	200
No. of Defined Social Interests	25 per node
Transmission speed	250 kBps
Transmission radius	300 meters
Buffer capacity	500 MB
Message TTL	10 hours
Freq. of Message Creation	1/60 sec^{-1}
Message size	1 MB
Simulated time	24 hours
SANE Relay Threshold	0.25
SANE Message Replicas	Unbounded
SEDUM Epoch Duration	1 sec
SEDUM Weight Constant	0.2
SEDUM Message Replicas	Unbounded

16 to 20 nodes per square kilometer throughout the day. These modifications of the dataset have no effect on the movements of the individuals, and thus preserves the intrinsic natural human mobility properties.

It is worth mentioning that many past works have used spatially dense datasets for experimental evaluation. Additionally, these datasets are typically constrained to a special event at a specific location and not sufficient to represent general human daily mobility. For example, the INFOCOM 2006 [16] dataset has 76 participants moving within an academic conference venue of approximately only 80 meters by 40 meters, resulting in the density as high as 20,000 people per square kilometer. Likewise, the SIGCOMM 2009 [15] dataset has similar number of participants, density, and conference venue locality. In these datasets, the participants attending the conference are away from their hometowns, thus influencing their habitual movements. The MIT Reality Mining dataset[3] does not provide the information needed to compute node density, but with only a cohort of college students and faculty contributing data, there is concern that their movements might be biased towards academic life. It is because of these reasons that we chose the GeoLife dataset in our experiments. The GeoLife dataset records a broader range of human mobility during a much longer time period, thus better representing users' movement in their socially habitual manner.

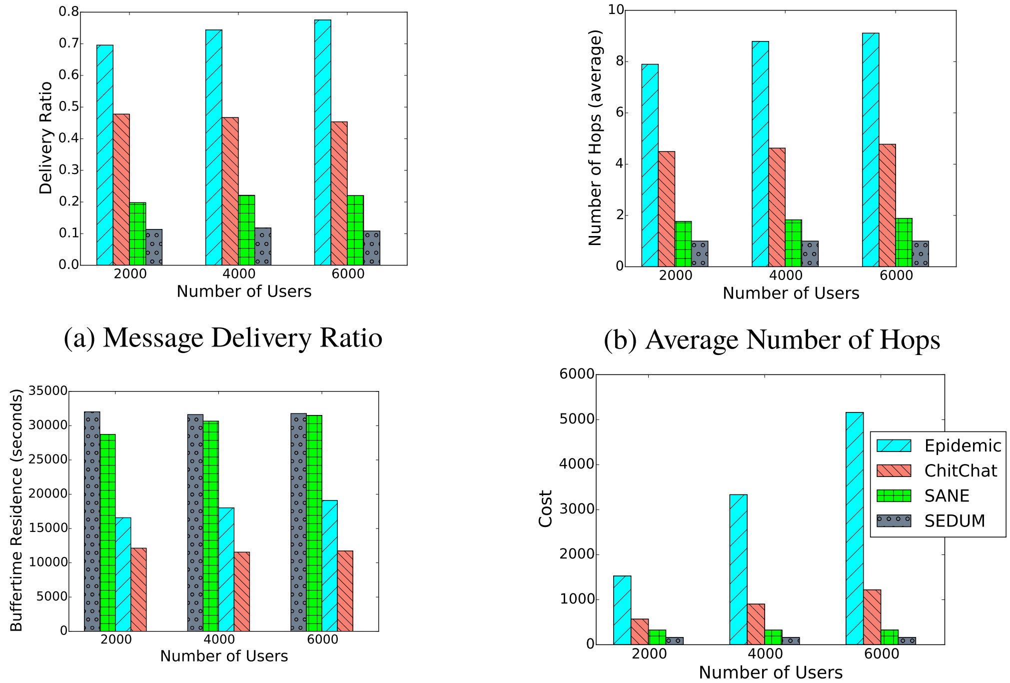
Since the GeoLife dataset contains no social profiles of its participating users, we generate a set of social interests per user selected uniformly out of 200 predefined social interests such as 'gourmet cooking', 'hiking', 'photographing', etc. To evaluate the routing protocols, we also randomly generate messages in the network as follows. For each message, we randomly select its sender and receiver among users, and keep the pairs for which there is at least one shared social interest. Then, we randomly select a subset of social interests from the chosen destination to attach to the message. The total simulation time covers 24 hours of GPS trajectory replay, and one message is generated every 60 seconds.

To compare the performance of each chosen algorithm, we adopt the following performance metrics: (i) message delivery ratio; (ii) the average number of hops to deliver a message; (iii) the average time messages resided in each node's buffer before being deleted; and (iv) the average replication overhead[2]. In what follows, we report the performance results of the unicast versions of these protocols due to the lack of multicast simulation in the ONE simulator. Unless otherwise stated, the settings in Table 2 were used across all experiments conducted. We used the default settings for SANE and SEDUM as recommended in [13] and [10], respectively.

3.1. EFFECT OF THE NUMBER OF USERS IN THE NETWORK

In the first round of experiments, we evaluate the routing performance of all the algorithms by varying the total number of users in the network from 2,000 to 6,000 with corresponding user density ranging from 5 to 20 per square kilometers, respectively. This round of simulations investigates the behavior and performance of each algorithm under

[2]The average replication overhead is defined in the prepackaged MessageStatsReport class of the ONE simulator [9]. It is calculated as $c = \frac{r-d}{d}$, where r is the total number of message transmissions that occurred in the simulation, and d is the number of messages that were successfully delivered to its destination.



(a) Message Delivery Ratio

(b) Average Number of Hops

(c) Buffertime Occupancy

(d) Overhead Costs

Figure 4. Effect of user participation on performance. The legend in the last graph applies to all.

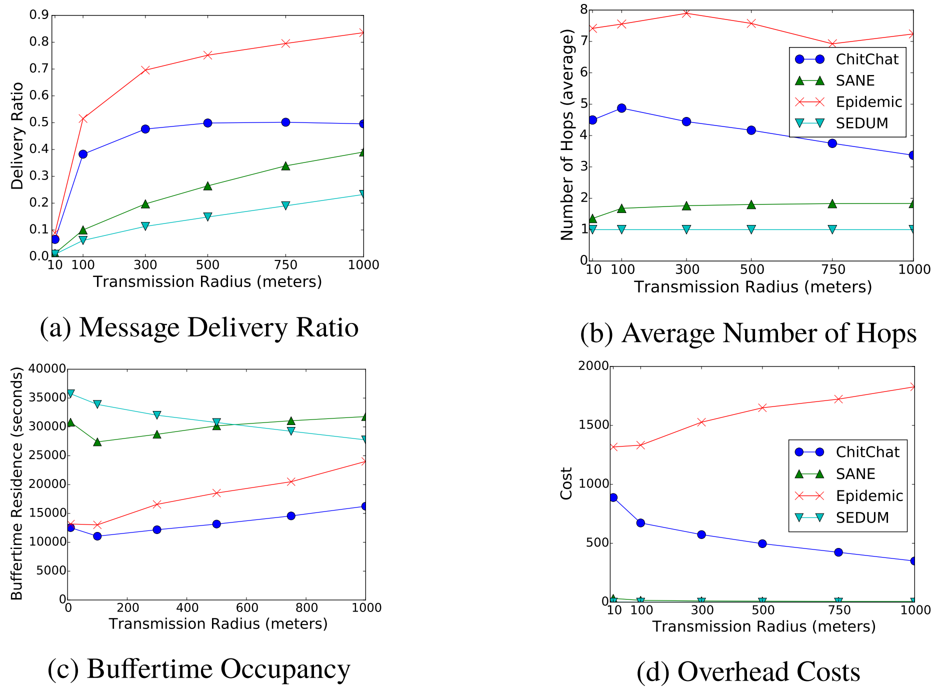
varying user sparsity. Intuitively, a decrease in the number of participants results in a lower user density, thus varying the network's sparsity. Figure 4 reports the results. It is not surprising to see that Epidemic achieves the highest delivery ratio since it floods messages to every node encountered. Our proposed ChitChat is the second best, achieving 2.5 times higher delivery ratio than SANE and more than 5 times higher delivery ratio than SEDUM at a fraction of the cost of Epidemic flooding. This is because ChitChat effectively utilizes the Transient Social Relationships of encountered nodes to identify the nodes that have higher likelihood of delivering messages. However, SANE considers only direct social interests between two users and SEDUM considers only the duration of interaction time, which are not effective enough in sparse networks.

Regarding the number of hops to deliver a message (Figure 4(b)), although SANE and SEDUM require the smallest number of hops, it does not mean they operate with more efficiency. Rather, the low hop counts are attributable to their ability to only deliver to nearby nodes, failing to deliver to nearly 80% of the created messages. The buffer-space efficiency of the algorithms can be observed from Figure 4(c), which shows the time a message needs to be kept in a node's buffer. Our proposed ChitChat has the shortest buffer occupancy time for each message. This again indicates that ChitChat makes intelligent message forwarding decisions so that message buffers do not needlessly fill, which could lead to catastrophic congestion that might prevent the delivery of messages with better delivery likelihoods. More importantly, ChitChat also has very low communication overhead compared to Epidemic as shown in Figure 4(d). Both SANE and SEDUM have low communication overhead too because their algorithms are designed to route messages which can be reached within a few hops in a dense network. The sparsity of the network causes their algorithm to rarely forward messages before they expire.

3.2. EFFECT ON TRANSMISSION RANGE

In the second round of simulations, we evaluate the routing performance in a subset of 2,000 users by varying the communication range from 10 meters to 1,000 meters[3]. Similar to the simulations in Section 3.1, the motivation for this round of simulations is to observe the effect of variable network sparsity on the behavior and performance of each protocol. As shown in Figure 5(a), the delivery ratio of all approaches increases with the communication range because expanded communication ranges yield better likelihoods to find proper forwarding nodes. Epidemic still achieves the highest delivery ratio and our proposed ChitChat is the second best in all cases. It is interesting to see that ChitChat has a relatively stable performance when the communication range grows quite large (more

[3]It is worth noting that 1,000 meters of communication range may not be possible in a real PSN, and is considered here to test the extreme behavior of all approaches.



(a) Message Delivery Ratio

(b) Average Number of Hops

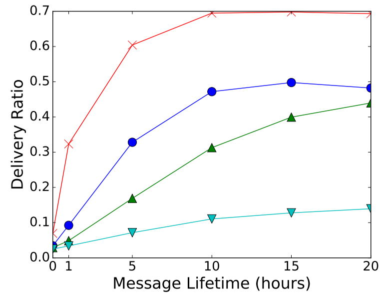
(c) Buffertime Occupancy

(d) Overhead Costs

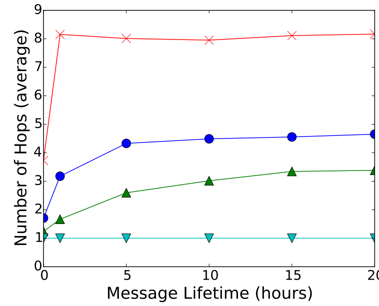
Figure 5. Effect of transmission radius on performance. The legend in the last graph applies to all.

than 500 meters) while SEDUM and SANE's delivery ratios continue to improve. This indeed indicates the effectiveness of the Transient Social Relationship modeling in the ChitChat. The transient social relationships already consider the social impact of users residing multiple hops away and hence the unrealistic expansion of communication ranges is not necessary. In other words, ChitChat is more suitable to PSN applications than existing works, where communication range among devices are typically limited to small proximities.

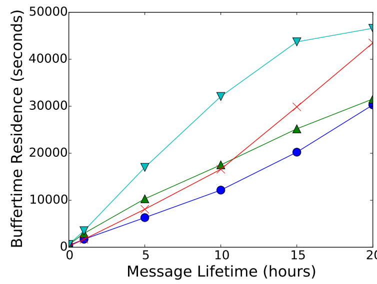
As for the average number of hops per message, we observe that ChitChat requires fewer number of hops with the increase of the communication range. The reason is straightforward: when communication ranges are large, multi-hop paths become shorter because fewer intermediate nodes are needed to reach a destination. However, SANE and SEDUM have constantly low number of hops (1 to 3) which is due to their ability to



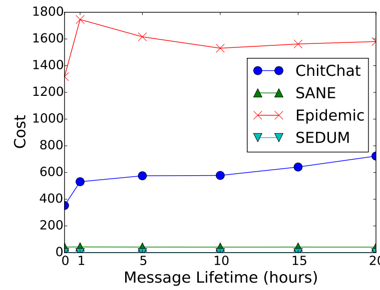
(a) Message Delivery Ratio



(b) Average Number of Hops



(c) Buffertime Occupancy



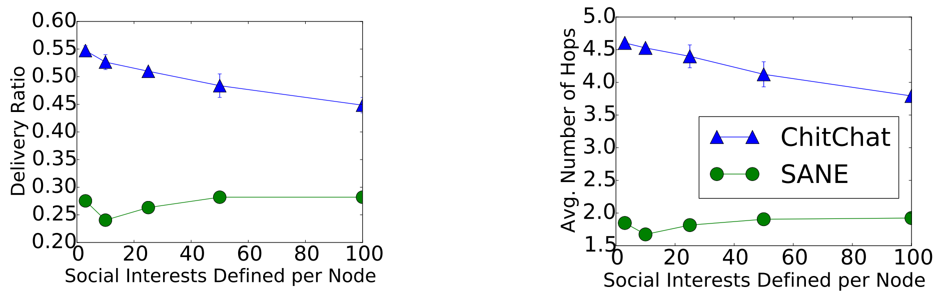
(d) Overhead Costs

Figure 6. Effect of message lifetime on performance. The legend in the last graph applies to all.

only deliver messages that are not far from the destination. As for the buffer time and communication overhead, our ChitChat again achieves the shortest buffer time and low communication overhead. The reasons are similar to that for the first round of experiments.

3.3. EFFECT OF MESSAGE LIFETIME

Next, we examine how message lifetimes affects the performance of each algorithm. We vary the messages' lifetimes from 1 hour to 20 hours. As shown in Figure 6(a), the delivery ratio of SANE and SEDUM continues to increase with the increase of message lifetimes, while both Epidemic and ChitChat reach plateaued performance early on. This is because Epidemic floods messages to all neighbors and the delivery ratio will not change if there is no path between a sender and its destination, as is common in sparse PSNs, no matter how long the messages are kept in the buffer. Similar for ChitChat, its plateaued



(a) Message Delivery Ratio

(b) Average Number of Hops

Figure 7. Effect of the number of social interests per user.

performance implies ChitChat is making well-informed forwarding decisions. However, the increase of the delivery ratio with the message lifetime in SANE and SEDUM reflect their limitations on identifying promising forwarding nodes. When message lifetimes are short, SANE and SEDUM are unable to find the right nodes to forward the messages. In Figure 6(b), we can also see that SANE and SEDUM require more hops per message when their delivery ratio increases (i.e. more messages with longer paths have been delivered), which conforms to our explanation in previous experiments. In addition, the buffer occupancy time is growing with message lifetimes as expected, and ChitChat's cost still remains 1/3 of Epidemic in all cases.

3.4. EFFECT OF SOCIAL INTERESTS DISTRIBUTION

Finally, we evaluate the effect of nodes' social interests declarations on the performance of the only two approaches that use the social interests: ChitChat and SANE. Specifically, we vary the number of social interests associated with each user from 5 to 100 out of a predefined 200 unique social interests. As shown in Figure 7, the increase of the social interests per user does not affect SANE much whereas it slightly degrades the performance of the ChitChat. This is because SANE considers only the social interests of two interacting users, but ChitChat takes into account the social interests belonging to users multiple hops away. In the ChitChat system, the more social interests per user, the larger number of social interests will be held in the Transient Social Relationships by each user, which in turn complicates forwarding decisions. However, it is worth noting that even in the extreme and almost unrealistic case where each user has 100 unique social interests, ChitChat still outperforms SANE by delivering two times more messages.

4. RELATED WORK

Many unique PSN routing algorithms have reported in past works. Epidemic routing [18] provides an upper bound for the number of successfully delivered messages as well as a lower bound on the delivery delay if one assumes light traffic or infinitely fast data transmission and unlimited buffer sizes. This is not a reasonable assumption, however, as past work [8] has shown that assuming finite transmission speeds and buffer sizes can result in network degradation when using Epidemic. Thus, for the sake of reducing the resources consumed, any routing mechanism that is deployed needs to identify if a connected neighbor is a worthwhile candidate for message forwarding, and if so, which messages should be forwarded.

Initial pioneers in the field of PSN routers have been quite simple, yet effective at delivering messages at a fraction of the cost of Epidemic-level flooding. PRoPHET [11] makes intermediate message forwarding decisions based on the observed probability that an intermediate node will meet with the destination, and maintains the freshness of these probabilities through the use of a weighted, convex combination of past and current probabilities. BUBBLE Rap [6] has messages *bubble up* through the network to higher-centrality nodes, reaching more popular nodes until it enters the destination's community. Once there, the forwarding strategy shifts its focus from global centrality to community-centric centrality, i.e., centrality with nodes of that community. Hui et al. reckon a node's popularity within its community is more effective at reaching the destination than the node's global popularity when the node shares a community with the destination.

Whereas PRoPHET calculates contact probabilities based on the rate of contacts per time period, the SEDUM router [10] expresses a similar metric by using continuous contact durations between any two nodes during a time period. This metric is then used to make message forwarding decisions between a message carrier and an intermediate node. The message carrier forwards a message to another node if the recipient has a higher utility with the destination than the current carrier. Utilities are also transitively spread throughout the network in an opportunistic manner, only being passed between nodes when connections are established. This permits an intermediate node to have a high utility with another node even if the two nodes never directly meet. Rather, the two nodes meet indirectly through one or more intermediate relays.

Whereas the foundational work on PSN routers has focused on probabilistic meetings between two nodes, a recent shift focuses on exploiting social artifacts. The SANE algorithm [13] requires each individual to hold a binary string of equal length that can be translated into the set of unique keywords that describe the user's interests. When two individuals meet, a message carrier computes the cosine similarity of its social interest vector to its neighbor's social interests, and decides to forward the message if the similarity exceeds a predefined threshold. To the best of our knowledge, this work is the first to investigate the affect of social interests as a decisive characteristic in message forwarding in sparse PSNs, and is thus offers a fair comparison for our proposed work.

These past works have limitations on successfully delivering messages in spatial-temporal sparse environments, where users are sparsely distributed throughout a geographic area or few members of that area are participating in the PSN. The evaluations of past works have primarily been conducted on very dense datasets, such as those showing the contact traces of academic attendees of a conference [15, 16] or students' movements through a college campus [3, 14]. Sparse networks have not been investigated extensively, such as is present in metropolitan environments where a small subset of the population is participating in the network. This makes it unlikely that short paths exist between a source and destination.

Rather, geographically distant individuals are connected by long multi-hop paths, with many intermediate nodes, spanning long periods of time. With probabilistic routing [10, 11], the chances of an intermediate node having ever contacted the destination is very slim under this scenario. What is needed for a network with these characteristics is a routing mechanism that can successfully percolate social relationship information throughout the network, thus permitting routing decisions to occur in a sparsely-connected network.

5. CONCLUSION

In this paper, we present a novel routing protocol, the ChitChat system, for sparsely-connected Pocket Switched Networks (PSNs), which exist in many real applications. Our proposed ChitChat system successfully models multi-hop social relationships via a novel decay-growth model and enables each participating node to make informed decisions during message routing. Our experimental study with a real world dataset demonstrates the superiority of our proposed approach compared to recent existing efforts, in that our ChitChat system achieves high delivery ratio with much lower communication overhead and shorter buffer occupancies.

As part of our future work, we have multiple of avenues to investigate. Of primary interest is to evaluate how well ChitChat performs when people's social profiles exhibit various distributions and geographic correlations of interests. Additionally, we will also evaluate the effectiveness of ChitChat's multicasting capabilities to broaden its adoption in more potential applications. Finally, it would be interesting to investigate the real-time augmentation of message metadata annotations. Such an application would greatly assist battlefield reconnaissance and intelligence gathering by speeding up the turn around between raw field data to rich intelligence acquisition, thus facilitating faster turn around in wartime strategies.

ACKNOWLEDGEMENTS

This work was funded by the Department of Education (P200A120110).

REFERENCES

[1] Cabaniss, R., Vulli, S. S., and Madria, S., 'Social group detection based routing in delay tolerant networks,' Wireless Networks, 2013, **19**(8), pp. 1979–1993, ISSN 1022-0038, doi:10.1007/s11276-013-0580-2.

[2] Daly, E. M. and Haahr, M., 'Social network analysis for routing in disconnected delay-tolerant manets,' in 'Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing,' MobiHoc '07, ACM, New York, NY, USA, ISBN 978-1-59593-684-4, 2007 pp. 32–40, doi:10.1145/1288107.1288113.

[3] Eagle, N., Pentland, A. S., and Lazer, D., 'Inferring friendship network structure by using mobile phone data,' Proceedings of the National Academy of Sciences, 2009, **106**(36), pp. 15274–15278, doi:10.1073/pnas.0900282106.

[4] Gao, W., Cao, G., La Porta, T., and Han, J., 'On exploiting transient social contact patterns for data forwarding in delay-tolerant networks,' IEEE Transactions on Mobile Computing, 2013, **12**(1), pp. 151–165, ISSN 1536-1233, doi:10.1109/TMC.2011.249.

[5] Hattachi, R. E. and Erfanian, J., 'Ngmn 5g white paper,' Technical Report 1.0, Next Generation Mobile Networks Ltd, 2015.

[6] Hui, P., Crowcroft, J., and Yoneki, E., 'BUBBLE rap: Social-based forwarding in delay-tolerant networks,' IEEE Transactions on Mobile Computing, 2011, **10**(11), pp. 1576–1589, ISSN 1536-1233, doi:10.1109/TMC.2010.246.

[7] Jain, S., Fall, K., and Patra, R., 'Routing in a delay tolerant network,' in 'Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications,' SIGCOMM '04, ACM, New York, NY, USA, ISBN 1-58113-862-8, 2004 pp. 145–158, doi:10.1145/1015467.1015484.

[8] Jindal, A. and Psounis, K., 'Performance analysis of epidemic routing under contention,' in 'Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing,' IWCMC '06, ACM, New York, NY, USA, ISBN 1-59593-306-9, 2006 pp. 539–544, doi:10.1145/1143549.1143657.

[9] Keränen, A., Ott, J., and Kärkkäinen, T., 'The ONE Simulator for DTN Protocol Evaluation,' in 'Proceedings of the 2nd International Conference on Simulation Tools and Techniques,' SIMUTools '09, ICST, New York, NY, USA, ISBN 978-963-9799-45-5, 2009 .

[10] Li, Z. and Shen, H., 'Sedum: Exploiting social networks in utility-based distributed routing for dtns,' IEEE Transactions on Computers, 2013, **62**(1), pp. 83–97, ISSN 0018-9340, doi:10.1109/TC.2011.232.

[11] Lindgren, A., Doria, A., and Schelén, O., 'Probabilistic routing in intermittently connected networks,' in 'Service Assurance with Partial and Intermittent Resources,' volume 3126 of *Lecture Notes in Computer Science*, pp. 239–254, Springer Berlin Heidelberg, ISBN 978-3-540-22567-6, 2004.

[12] McGeehan, D., Lin, D., and Madria, S., 'Theorems and Proofs of Bounded Storage Complexity in the ChitChat PSN Routing System,' Technical report, Missouri University of Science and Technology, Department of Computer Science, 2015.

[13] Mei, A., Morabito, G., Santi, P., and Stefa, J., 'Social-aware stateless routingin pocket switched networks,' IEEE Transactions on Parallel and Distributed Systems, 2015, **26**(1), pp. 252–261, ISSN 1045-9219, doi:10.1109/TPDS.2014.2307857.

[14] Nahrstedt, K. and Vu, L., 'CRAWDAD dataset uiuc/uim (v. 2012-01-24),' Downloaded from http://crawdad.org/uiuc/uim/20120124, 2012, doi:10.15783/C7HG6D.

[15] Pietilainen, A.-K. and Diot, C., 'CRAWDAD data set thlab/sigcomm2009 (v. 2012-07-15),' Downloaded from http://crawdad.org/thlab/sigcomm2009/, 2012.

[16] Scott, J., Gass, R., Crowcroft, J., Hui, P., Diot, C., and Chaintreau, A., 'CRAWDAD dataset cambridge/haggle (v. 2009-05-29),' Downloaded from http://crawdad.org/cambridge/haggle/20090529, 2009, doi:10.15783/C70011.

[17] Spyropoulos, T., Turletti, T., and Obraczka, K., 'Routing in delay-tolerant networks comprising heterogeneous node populations,' IEEE Transactions on Mobile Computing, 2009, **8**(8), pp. 1132–1147, ISSN 1536-1233, doi:10.1109/TMC.2008.172.

[18] Vahdat, A. and Becker, D., 'Epidemic routing for partially-connected ad hoc networks,' Technical Report CS-20000, Duke University, 2000.

[19] Zheng, Y. and Fu, H., *Geolife GPS trajectory dataset - User Guide*, 2011.

[20] Zheng, Y., Li, Q., Chen, Y., Xie, X., and Ma, W.-Y., 'Understanding mobility based on gps data,' in 'Proceedings of the 10th International Conference on Ubiquitous Computing,' UbiComp '08, ACM, New York, NY, USA, ISBN 978-1-60558-136-1, 2008 pp. 312–321, doi:10.1145/1409635.1409677.

[21] Zheng, Y., Xie, X., and Ma, W.-Y., 'Geolife: A collaborative social networking service among user, location and trajectory,' IEEE Data Engineering Bulletin, 2010.

[22] Zheng, Y., Zhang, L., Xie, X., and Ma, W.-Y., 'Mining interesting locations and travel sequences from gps trajectories,' in 'Proceedings of the 18th International Conference on World Wide Web,' WWW '09, ACM, New York, NY, USA, ISBN 978-1-60558-487-4, 2009 pp. 791–800, doi:10.1145/1526709.1526816.

[23] Zhu, Y., Xu, B., Shi, X., and Wang, Y., 'A survey of social-based routing in delay tolerant networks: Positive and negative social effects,' IEEE Communications Surveys Tutorials, 2013, **15**(1), pp. 387–401, ISSN 1553-877X, doi: 10.1109/SURV.2012.032612.00004.

II. EFFECTIVE SOCIAL-CONTEXT BASED MESSAGE DELIVERY USING CHITCHAT IN SPARSE DELAY TOLERANT NETWORKS

D. McGeehan, S. Madria, D. Lin

ABSTRACT

Delay Tolerant Networks (DTNs) have garnered much interest with the wide-spread adoption of portable smart devices capable of wirelessly connecting with one another, thus enabling the formation of a network for opportunistic data dissemination. This type of network is useful in a variety of applications where other form of network communication strategies are unavailable, such as an on-the-ground tactical military network in an active battlefield or an emergency network formed immediately after a catastrophic disaster. DTNs also provide opportunities for various other interesting applications such as location-based social networking, interests-based data dissemination, and geolocal advertising. One persistent challenge for DTNs is achieving sufficient message delivery due to the dynamic, unpredictable, and opportunistic nature of inter-device connections; this challenge is exacerbated when such connections are sparsely available. In this paper, a novel social-context based message routing system, called ChitChat, is proposed with the focus on message delivery through sparsely-connected DTNs. ChitChat is a hybrid geographic/data-centric routing system designed to exploit each user's social (or mission) interests to opportunistically learn of multi-hop paths through the network, and to derive the social semantics of geographic locations using user travel itineraries and multi-hop social relationships. In turn, this information is used to make distributed routing decisions based on the likelihood an encountered node will connect with others capable of successfully delivering a message. An analysis of network sparsity is conducted against five real-world datasets. Through sim-

ulations using the two highest-sparsity real-world datasets, ChitChat is capable of achieving more successful deliveries against three recent state-of-the-art DTN routing schemes while incurring lower costs against flooding.

1. INTRODUCTION

The ubiquitous adoption of portable smart devices (e.g., smart phones and smart watches), along with the provisioning of technology to allow now long-range device-to-device communication (such as using radio communication) provides an opportunity for exploring Delay Tolerant Networks (DTN) [1, 12, 25, 32] – a type of Mobile Ad-Hoc Network in which messages are routed by personal devices carried by ever-moving people in a highly intermittent environment like an active battlefield or a region impacted by a disaster. Figure 1 portrays this type of network environment where, as people move, their mobile devices connect with one another when they are within wireless communication range, and data packets may be exchanged while the connection lasts [13, 32]. The challenge that lies in effectively utilizing these mobile devices as message disseminators results from the volatility and sparsity in wireless network connectivity, primarily due to limited resources (i.e., battery and storage), low node densities, long inter-contact times between nodes, and security restrictions on message passing.

For example, soldiers in an active battlefield carrying wireless-equipped tablets can act as nodes in a temporary ad-hoc network in case primary communications are destroyed. However, the ad-hoc network may suffer extreme congestion due to the high velocity of information and the frequent disconnections between exchanging devices. Each connection can only transport a finite quantity of information, and thus to prevent catastrophic congestion, a message should only be routed through a connection if it will likely reach its intended destination because of the connection.

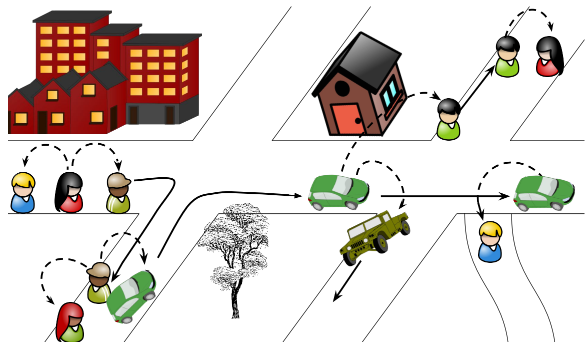


Figure 1. An example of a Delay Tolerant Network. Solid arrows indicate movement of device holders, and dashed lines indicate established connections and message forwarding.

Another example considers a commercial application of DTNs, whereby businesses broadcast sale advertisements and coupons to patrons passing near their storefronts [34]. However, individuals interested in the business's products or services may not frequently visit the store's location. Users are willing to accept and distribute these advertisements to their friends and neighbors to increase the chance they will receive the ones they are interested in, but sophisticated forwarding strategies must be adopted so as to not burden the users with depleted battery life and to not disengage them with copious amounts of irrelevant information.

One key challenge of DTNs is the difficulty of achieving high message delivery ratios with finite lifetimes [12, 25, 33]. This challenge is exacerbated in networks characterized by spatial-temporal sparsity. Spatial-temporal sparsity refers to the sparse connections among participating nodes in a DTN primarily due to (1) the geographical distances between each other, (2) the duration of time between connection opportunities, and (3) security or non-cooperative nature of nodes carrying messages. Moreover, in a DTN, there is no prior knowledge as to when connection opportunities will arise for messages to make the next hop along their journey. When an opportunity does occur, it is also challenging to make the optimal decision on whether to forward messages. Numerous constraints limit how many messages to forward, including varying bandwidth, limited buffers within each device, and restricted lifetime of messages.

Although there have been many previous algorithms for data dissemination in DTNs [4, 7, 10, 12, 13, 19, 21, 22, 25, 32, 33, 35, 39], none of them tackle the sparsity challenge that exists in a real world DTN environment. Existing works implicitly assume high node densities in a DTN such as a concentrated meeting environments [9, 27, 30] or other social gatherings, with evaluations of such systems relying solely on synthetic or real-world datasets exhibiting this property. This limits their adoptions in a wider range of applications.

In this paper, we propose a novel message routing approach, called ChitChat, which is designed to conquer the spatial-temporal sparsity problem in DTNs. Our approach leverages an important characteristic of DTNs: the devices are carried by people, whose movements are guided by their social interests, roles, and responsibilities. Our proposed ChitChat system implements sophisticated strategies for nodes (mobile devices carried by people) to exchange, aggregate, store, and disseminate the social profiles of encountered individuals along with opportunistically learning about the geographic social landscape of their environment in a distributed, decentralized manner. This allows individuals to utilize increasingly rich information to judge whether a current opportunistic channel has high likelihood to lead to a successful message journey.

More specifically, when two nodes connect with each other, they *chitchat* to exchange the following information: (i) *direct social interests*, which is the metadata, or the set of keywords, that describe the encountered node's interests (social or mission related), roles, and responsibilities (e.g. social interests such as "photography" and "hiking", or role–specific metadata such as "firefighter", "military intelligence officer"); (ii) *transient social relationships*, which is aggregated information of the social interests of the people that the node has encountered before; and (iii) *geographic social heatmaps*, which describe various locations of interest in terms of the social interests of visiting individuals. Our approach allows the social interests to be dynamically expanded, refined, and aggregated in real-time into richer transient social relationships so as to capture multi-hop relationships, and permits the learning of social semantics of various locations and/or of nodes, both in ways that accommodate the constraints in DTNs.

For example, if Alice and Bob connect reliably, and Bob and Janet connect reliably, then Alice's transient social relationships should include Janet's social interests even if Alice and Janet have never connected earlier and have no common interests. To more precisely model the weight of multi-hop relationships, we have carefully designed a temporal growth and decay model that ensures the freshness of these transient social relationships while also maintaining an upper bound on storage complexity. Alice is also able to learn more about the social semantics of her environment from both Bob and Janet, even if she has never visited some locations therein. Locations visited by nodes are annotated with social interests and are stored on a node's local map. When nodes come in contact with one another, they exchange maps containing both their own visited locations and the locations visited by previously encountered nodes to enrich and expand their own knowledge of locations in their environment.

The contributions of this paper are summarized as follows:

- We propose the ChitChat routing algorithm to address the impact of spatial-temporal sparsity on message delivery in DTNs. Our approach is able to facilitate nodes in a DTN to make intelligent decisions on which nodes to forward a message copy, even in situations where the journey between the source and destination has high social network distance.

- We propose a novel way of modeling and maintaining multi-hop transient social relationships by taking into account the time durations of connections and disconnections. This model maintains the accuracy and freshness of social relationship information residing within each node, building a solid foundation for making informed message routing decisions to overcome the constrained connectivity in sparsely-connected DTNs. It also permits users to change their social interests through time while still participating in the network.

- We propose a novel method for learning the social semantics of various locations through the use of node itineraries - the paths that they travel between locations of interest - and the transient social relationships of people visiting locations along these itineraries. This approach provides valuable knowledge for nodes to decide if an encountered node can deliver messages to locations commonly visited by interested individuals, even if the encountered node has no interest in such messages.

- We analyze the network densities of five real world datasets: the INFOCOM 2006 dataset [30], the SIGCOMM 2009 dataset [27], the MIT Reality Mining dataset [9], Microsoft's Geolife dataset [36, 37, 38], and Nokia's Mobile Data Challenge dataset [17, 18]. To the best of our knowledge, this analysis is the first attempt to quantify and compare the sparsity of these datasets as a means of determining whether their use in simulations is appropriate for research into DTNs. Our findings suggest that INFOCOM 2006 and MIT Reality Mining, both widely used to conduct DTN simulations, are quite dense compared to the other three, which have rarely been used. This suggests that handling sparse DTNs is an unaddressed field of research.

- As a result of the network density analysis, we evaluate the performance of the ChitChat protocol using two real world datasets exhibiting the highest sparsity, Microsoft's GeoLife dataset and Nokia's Mobile Data Challenge dataset, both providing diverse and fine-grained human trajectories over time periods spanning multiple years. We compare the performance of ChitChat against three recent state-of-the-art DTN algorithms [19, 21, 25] that have been selected based on their improved performance over their predecessors. The experimental results demonstrate that our proposed ChitChat achieves better message delivery ratios without incurring flooding-level communication overhead. Thus, we conclude that multi-hop relationship modeling and location semantics help achieve significant improvements on message routing in sparsely-connected DTNs.

The rest of the paper is organized as follows. Section 2 presents a formal definition of the Delay Tolerant Network environment, along with defining our proposed ChitChat system. Section 3 presents the network density analysis of the five real world datasets, followed by the evaluation of simulation results using the two most sparse datasets. Section 4 reviews existing message routing protocols in DTNs. Finally, Section concludes the paper.

2. CHITCHAT ARCHITECTURE AND ROUTING PROTOCOLS

In this section, we provide a formal discussion of the ChitChat system. We first present preliminary definitions of entities used by the proposed ChitChat system in Section 2.1, followed by a discussion on the modules and algorithms of this system. Our ChitChat system consists of three major components with associated storage buffers as shown in Figure 2: (i) Opportunistic Geographic Social Heatmap (OGSH) modeling; (ii) Opportunistic Transient Social Relationship (OTSR) modeling; and (iii) Message Routing. Each of these modeling approaches is to be appropriate for the limitations and restrictions of DTNs. When two users come within communication range, the ChitChat system will first invoke the OGSH module to exchange the two users' current Geographic Social Heatmaps (GSHs) as defined in Section 2.2. Upon receiving its neighbor's GSH, a node will merge it into its own. Following this, each node calls their OTRS module to update, exchange, and merge their current Transient Social Relationships (TSRs). Section 2.3 defines the procedures for this process, and proves there is an upper bound to the storage needed for this subsystem. Finally, the ChitChat system will invoke the message routing component to exchange a selected subset of messages carried by the two users based on the analysis of their revised TSRs and GSHs, as defined in Section 2.4.

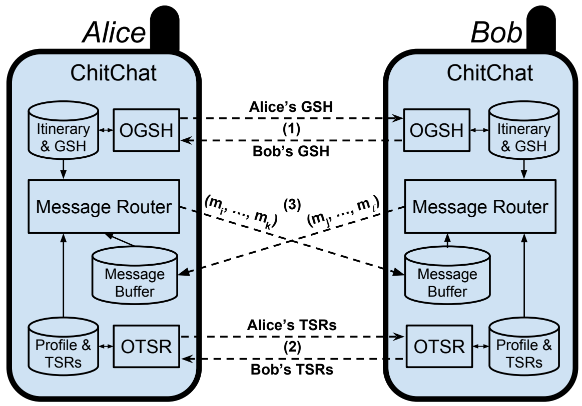


Figure 2. Architectural diagram of the ChitChat system within each node, employing Opportunistic Geographic Social Heatmaps, Opportunistic Transient Social Relationships, and the Message Router. Here, Alice and Bob connect with one another, initiating the ChitChat system. Alice exchanges (1) her Geographic Social Heatmap (GSH, see Definition 8) and (2) her Transient Social Relationships (TSRs, see Definition 6) with Bob, and vice versa. Once this *chitchat* is completed, (3) Alice decides which messages (if any) to forward to Bob and builds a bundle of these messages $\langle m_i, \ldots, m_k \rangle$. Likewise, Bob builds a bundle of his messages $\langle m_j, \ldots, m_l \rangle$ to send. These bundles are then forwarded to each other, and added to the recipient's message buffer.

2.1. PRELIMINARY DEFINITIONS

In this work, we consider the following characteristics of a DTN. Nodes in a DTN are users who have devices like smart phones equipped with the ChitChat system, which automatically connects to other devices that are within communication range, using such technologies as Bluetooth, WiFi Direct, or Google Nearby. Each user has his/her own social profile (see Definition 4 and 5), itinerary (see Definition 7), and geographic social heatmap (see Definition 8). The messages that a user carries are each annotated with appropriate metadata keywords (a subset of all social interests) that describe the topic or content of a particular message. Figure 1 provides a diagram of how a DTN operates.

Definition 4. (Social Interest). A social interest is represented as a tuple $\langle \mathrm{SID}, \mathrm{kw} \rangle$, where SID is a unique ID to identify the social interest and kw is the keyword that describes the content of this social interest.

Definition 5. (Social Profile). Let u be a node in a DTN and SP_u be its social profile. The social profile of u is a set of social interests, i.e., $\mathrm{SP}_u = \{\mathrm{SID}_1, \mathrm{SID}_2, \ldots, \mathrm{SID}_k\}$.

For example, suppose there are three types of social interests: $\langle 001, \text{'hiking'}\rangle$, $\langle 002, \text{'photography'}\rangle$, and $\langle 003, \text{'firefighter'}\rangle$. If Alice's social interests include 'hiking' and 'photography', her social profile can be represented as $\mathrm{SP}_{alice} = \{001, 002\}$. When Alice visits her favorite outdoor equipment store and learns of their coupons and sales, she may pass along these coupons to her friends when they next meet.

The social context in which the ChitChat is deployed need not necessarily be strictly recreational interests. In considering the battlefield application of DTNs, a patrolling soldier may become aware of the movement of enemy tanks and the location of ammunition stockpiles. In this situation, the social interest (i.e., mission interest) could be 'ammunition location' and 'tank movement', and a message would be reported back to Ground Intelligence Officers with expressed roles for tracking ammunition stockpiles and tank movements. For a disaster deployment of a DTN, social interests such as 'flooded road', 'gas leak', and 'stranded civilians' could be deployed.

Definition 6. (Transient Social Relationship). The Transient Social Relationships of a node u at timestamp t is represented as

$$\mathrm{TSR}_u = \langle\langle \mathrm{SID}_1, w_u(\mathrm{SID}_1, t)\rangle, \langle \mathrm{SID}_2, w_u(\mathrm{SID}_2, t)\rangle, \ldots, \langle \mathrm{SID}_k, w_u(\mathrm{SID}_k, t)\rangle\rangle$$

where SID_i is the ID of a social interest and $w_u(\mathrm{SID}_i, t)$ is the weight of the social interest SID_i for user u at time t such that $0 \leq w_u(\mathrm{SID}_i, t) \leq 1$.

Whereas a node's Social Profile indicates the personal social interests of the node, its Transient Social Relationships (TSRs) act as indicators of whether this node could reliably forward messages having content relevant to a particular social interest to other nodes sharing that interest. It is intended to offer insights on the ever-evolving multi-hop relationships, whether direct or indirect, that nodes have with each other.

Continuing with the example following Definition 5, recall that Alice's social profile is $\mathrm{SP}_{alice} = \{001, 002\}$, representing her interests in hiking and photography. Further, assume Alice spends more time with her photography friends than her hiking friends, and is married to a firefigher who often invites his coworkers into their home. Although she is not personally interested in firefighting, her close association with individuals interested in 'firefighter' induces her TSR for 'firefighting'. Specifically, Alice's TSRs at some timestamp could be $\mathrm{TSR}_{alice} = \langle\langle 001, 0.65\rangle, \langle 002, 0.95\rangle, \langle 003, 0.325\rangle\rangle$. Her direct interest in photography and hiking leads to the relevant TSRs having higher weights than that of 'firefighter'. Furthermore, since she spends more time with her photography friends than her hiking friends, her 'photography' TSR weight is higher than that for 'hiking'. Finally, since she lives with her firefighter husband, and his coworkers often visit, her 'firefighter' TSR weight is moderately above 0. The exact formula for calculating TSR weights is defined in Equations 1– 3 of Section 2.3.

Definition 7. (Itinerary). An itinerary of a node u is a sequence of tuples

$$I = (\langle x_0, y_0, t_0\rangle, \langle x_1, y_1, t_1\rangle, \ldots)$$

where x_i, y_i represents the ith longitude and latitude, respectively, of the itinerary and t_i represents the time when u will be at the ith location.

Suppose Alice is traveling from her home (located at $(0, 0)$) at 8:00AM to work (located at $(2, 4)$) by 8:20AM. Following her normal daily commute, her itinerary would be $(\langle 0, 0, 8{:}00\text{AM}\rangle, \langle 1, 0, 8{:}10\text{AM}\rangle, \langle 1, 4, 8{:}15\text{AM}\rangle, \langle 2, 4, 8{:}20\text{AM}\rangle)$, with the middle two tuples representing intermediate locations along the way (e.g. stoplights, subway stations). It is assumed this itinerary is readily available from such offline sources as a mobile application providing directions to some destination or from historic traveling patterns.

Definition 8. (Geographic Social Heatmap). A geographic social heatmap is a set of social staypoints $G = \{S_i, S_j, \ldots\}$, where a social staypoint $S = \langle x, y, C \rangle$ is a location (x, y) annotated with a set of weighted social IDs $C = \{\langle \mathrm{SID}_i, w_i \rangle, \langle \mathrm{SID}_j, w_j \rangle, \ldots\}$.

A geographic social heatmap expresses a node's understanding of the social landscape surrounding it, and is constructed based on the social profiles and itineraries of an individual node and the heatmaps of the nodes it opportunistically encounters. In the running example on Alice, her geographic social heatmap would contain the social staypoints $\langle 0, 0, \{\langle 001, 1.0 \rangle, \langle 003, 1.0 \rangle\} \rangle$ and $\langle 2, 4, \{\langle 001, 1.0 \rangle, \langle 003, 1.0 \rangle\} \rangle$ - i.e., there is a social staypoint for each endpoint of Alice's itinerary, $(0, 0)$ and $(2, 4)$, with each staypoint annotated with Alice's entire social profile with weights equal to 1.0 – $\{\langle 001, 1.0 \rangle, \langle 003, 1.0 \rangle\}$. This is only a simple example, and doesn't consider the influence of other encountered nodes on the contents of Alice's geographic social heatmap. The next section provides a more thorough explanation of this concept.

The ChitChat system operates on the initial condition that participating nodes have defined a social profile with their direct social interests. From these, the Transient Social Relationships and Geographic Social Heatmaps of each node are initialized from a cold-start (i.e., having no TSRs or GSHs defined) of the system. Through their encounters with others, TSRs and GSHs are exchanged to evolve those of each node to more accurately represent their social relationships and the location semantics of their environment. As a requirement for this system to operate, a node must voluntarily consume their own resources and share their TSRs and GSHs with one another, and are motivated to do so in exchange of benefiting from the network's functionality – i.e. a participating node would receive relevant messages through the network and will have its own messages relayed through the aid of others. Nodes that do not share their social information or resources do not benefit from the network, as a key requirement for ChitChat's routing algorithm (see Algorithm 6) is the knowledge of the potential relay's TSRs and itinerary strength.

2.2. OPPORTUNISTIC GEOGRAPHIC SOCIAL HEATMAP MODELING

The OGSH modeling aims to provide each node with a view of the social landscape of their surroundings, and likewise can be used to select better message forwarders among a node's encounters. For example, Alice is traveling to her favorite Thai restaurant in a plaza that also houses a toy store and a candy store. Since she has visited this restaurant before, her geographic social heatmap would contain a social staypoint for the plaza with SIDs for 'toys', 'candy', and 'discount clothing'. When her device connects to Bob's device, they exchange their GSHs. Referencing Alice's GSH, Bob could decide it is promising to forward a message about candy to Alice since she will be in the general vicinity of an area frequented by candy lovers. Algorithm 3 defines the procedure for opportunistically constructing and expanding each node's GSH.

The intuition behind this opportunistic modeling of GSHs stems primarily from the disconnected nature of DTNs. While there exist online sources for the social semantics of various locations (e.g., Google Places, Foursquare City Guide), an Internet connection is required to query these services. This is likely unavailable within a DTN, and as such this paper assumes its absence. Secondarily, these services only provide semantics of locations, not necessarily the people who visit these locations. For example, a hospital would have social IDs such as for 'medicine', 'surgery', and 'nurse'. However, if a significant number of individuals visit with a shared interest of 'baseball', then the hospital might be a worthwhile destination for messages about baseball. The aforementioned online services do not provide this type of associative semantics.

In Algorithm 3, two nodes u, v are connected to one another and intend on enhancing their own GSH by learning from their neighbor. First, u obtains its current itinerary, GSH, and TSR (lines 2–4). From its itinerary I, u constructs two social staypoints, S_0 and S_1, representing the origin and destination of u's current travel, respectively, and annotates them

Algorithm 3 Opportunistically merge two Geographic Social Heatmaps when a connection occurs between nodes u and v.

```
procedure MergeGSH(u, v)
    I ← u.getCurrentItinerary ()
    G_u ← u.getGSH ()
    TSR_u ← u.getTSR ()
                                ▹ Construct social staypoints for the itinerary endpoints
    S_0 = ⟨I.startPoint.x, I.startPoint.y, TSR_u⟩
    S_1 = ⟨I.endPoint.x, I.endPoint.y, TSR_u⟩
                                ▹ Integrate itinerary endpoints to GSH
    MergeStaypointIntoGSH (S_0, G_u)
    MergeStaypointIntoGSH (S_1, G_u)
    G_u → v                     ▹ Send the updated GSH to v
                                ▹ Merge v's GSH into u's GSH
    G_v ← v.getGSH ()
    for each S ∈ G_v do
        MergeStaypointIntoGSH (S, G_u)
    end for
end procedure

procedure MergeStaypointsIntoGSH(S, G)
                                ▹ Grab the staypoint in G that is nearest in location to S.
    S_nn = G.nearestNeighbor (S)
    if distanceBetween (S, S_nn) > d_max then
                                ▹ If the staypoints are sufficiently close, merge them
        S′ ← new SocialStaypoint
        S′.x ← 1/2 · (S_nn.x + S.x)
        S′.y ← 1/2 · (S_nn.y + S.y)
        S′.C ← {⟨SID_i, w′_i⟩ | SID_i ∈ S_nn.C ∪ S.C, w′_i ← max (w_i(S), w_i(S_nn))}
                                ▹ Replace the existing staypoint with the merged one
        G.remove (S_nn)
        G.add (S′)
    else
                                ▹ If the staypoints are far apart, add the new staypoint
        G.add (S)
    end if
end procedure
```

with u's TSRs (lines 5 and 6). These staypoints are then merged into u's GSH using the MergeStaypointIntoGSH function (lines 7 and 8), explained later. u sends the updated GSH to v (line 9), and then receives and merges v's GSH into its own GSH (lines 10–13).

The purpose for annotating S_0 and S_1 with TSR_u and then merging them into G_u is to bootstrap a node's initial, empty GSH and to inform individuals encountered in the future that a person (in this case, u), with the weighted social interests in TSR_u, is a visitor of those locations. Additionally it alleviates the requirement of u to explicitly divulge its current itinerary to its neighbor. u passes along its GSH G_u to v, who merges G_u into G_v (through its own execution of Algorithm 3), thus opportunistically learning about the types of people who visit various locations defined in G_u. In the future, when v connects with another person w, w will transitively know that a user with social interests of TSR_u were connectable at the locations of S_0 and S_1.

The MergeStaypointIntoGSH function operates to expand a user's GSH G with the given staypoint S by either adding S to G (line 26) or aggregating the location and the weighted social IDs of S with that of a pre-existing nearby staypoint (lines 17–24). The purpose for doing so is two-fold. First, less storage space and lower computational time for staypoint lookups is required for a set composed of merged staypoints in comparison to that of the original set. Second, and more importantly, the merging of a location-cohesive set of staypoints and their weighted social interests provides a summary of the types of people that visit the connectable vicinity a social staypoint. These individuals likely have diverse yet partially overlapping interests, and thus their presence within the connectable vicinity of a staypoint should be reflected in the GSH. To accommodate this, a new staypoint S that is within $d_{\texttt{max}}$ meters of an existing staypoint S_{nn} are merged. The merged staypoint S' is located halfway between the two staypoints S and S_{nn} (lines 20 and 21), and has the weighted social interests of both staypoints, with each social interest's weight being that which is larger (line 22). The pre-existing social staypoint is then replaced by the merged social staypoint (lines 23 and 24).

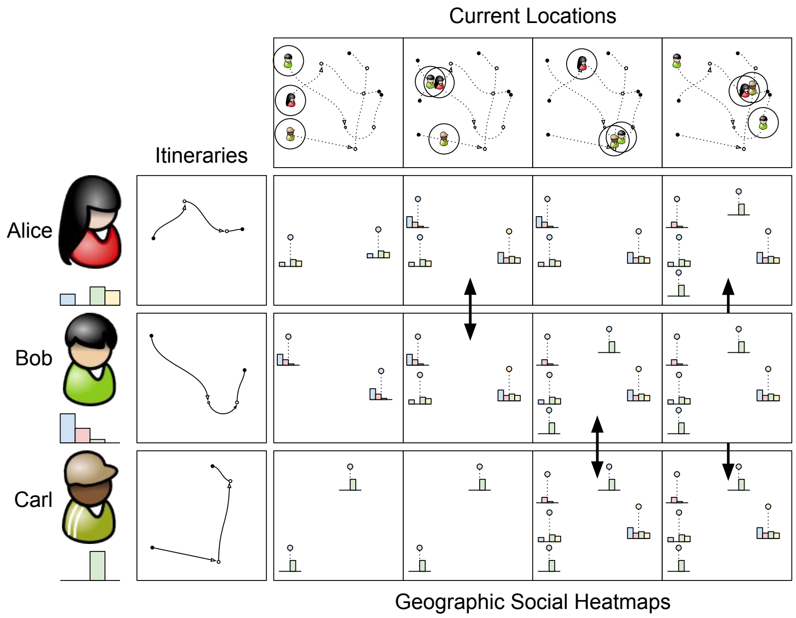


Figure 3. Illustrative example of opportunistic Geographic Social Heatmap (GSH) merging. Alice, Bob, and Carl know their itineraries, TSRs, and initial GSHs. Through encountering one another, a more thorough view of the social landscape is learned through exchanging and merging GSHs with one another.

Figure 3 illustrates an example of opportunistic modeling of GSHs among three people: Alice, Bob, and Carl. Initially, each of them know their current itinerary and TSRs, and initialize their GSHs to contain two social staypoints: one for the starting point of their own itinerary, and one for the ending point. Both social staypoints are annotated with their own TSRs. Once in travel, Alice first encounters Bob, and each of them exchange their GSH and merge the other's into their own. Since both Alice and Bob are traveling to a location very close to one another, the pre-existing social staypoints are merged together, creating a new social staypoint midway between the two original ones and with the larger of the two's weighted social interests. In essence, when Alice encounters Bob, both of their GSHs are syncronized. Alice and Bob continue their journeys, disconnecting from one another. Shortly thereafter, Bob encounters Carl. Again, both Bob and Carl exchange GSHs and merge them. Carl is now aware of Alice's influence on the social staypoint representing the end of her itinerary, albeit without directly contacting her. Bob and Carl continue along their journeys. The final scenario of this example has Alice and Carl directly meeting. They exchange and merge GSHs, and then continue onward. Ultimately, in this scenario, all three have syncronized GSHs.

2.3. OPPORTUNISTIC TRANSIENT SOCIAL RELATIONSHIP MODELING

The Opportunistic Transient Social Relationship (OTSR) modeling aims to represent the evolution of each user's social interests as impacted by the people that they encounter. Message routing can then select better message forwarders based on their TSRs. For example, Alice is interested in gourmet cooking. She may have friends with the same interest, who also have friends with the same interest, and so on. It would then be promising to ask Alice to help forward messages tagged with 'hiking' (e.g. routes, photographs, sales on special equipment, etc) so that the messages will have a higher likelihood of reaching their designated receiver(s) with the same interest. Alternatively, Alice may not be interested in 'photography', but she reliably encounters people who are. In this case, message delivery may be improved by having Alice as an intermediate carrier for messages tagged with 'photography'.

Much like the opportunistic modeling of Geographic Social Heatmaps in Section 2.2, the opportunistic modeling of Transient Social Relationships is primarily motivated by the unavailability of an Internet connection to query online sources for social interests of individuals and their friends. Another motivating factor is the capturing of associations between nodes based on physical proximity. Indeed, services such as Facebook offer individuals the ability to describe their interests and to infer their social relationships from a user's social graph. However, they do not provide information about physical proximity; two individuals with a social connection may live on opposite sides of the world and never meet one another. Additionally, these services miss out on the proximity between strangers who may not necessarily socially interact with each other, such as a group of commuters on a subway every morning. Regardless of the social connection between individuals, their close proximity offers opportunities for their mobile devices to transfer messages and gather information to aid in the effectiveness of message delivery.

As aforementioned, the OTSR modeling function is invoked when two users are within communication range in a DTN. The two users will exchange their current transient social relationships (TSRs) as defined earlier in Definition 6 and then adjust it to reflect their most recent multi-hop social relationships with other users in the network.

We now discuss the procedure for obtaining the social interests for the TSRs and how to compute their weight. Upon the network's initialization (denoted as timestamp t_0), each user's TSRs are the same as his/her social profile, with the weight of each TSR set to the medium weight of 0.5.

$$w_u(\mathrm{SID}_i, t_0) = \begin{cases} 0 & \text{if } \mathrm{SID}_i \notin \mathrm{SP}_u; \\ 0.5 & \text{if } \mathrm{SID}_i \in \mathrm{SP}_u. \end{cases} \tag{1}$$

For example, given Alice's social profile $\mathrm{SP}_{\mathrm{alice}} = \langle 001, 003 \rangle$, her initial TSRs at timestamp t_0 will be $\mathrm{TSR}_{\mathrm{alice}} = \langle\langle 001, 0.5\rangle, \langle 003, 0.5\rangle\rangle$.

Later, when users encounter each other, they conduct the following three steps to adjust their TSRs: (1) each one computes his/her own latest TSRs based on a decay model; (2) exchange their TSRs; (3) each one computes the growth of his/her TSRs based on a growth model. Without loss of generality, we consider u and compute its current TSRs as follows.

Suppose that user u enters the communication range of some users at timestamp t_s where $t_s - t_0 \geq 1$. The ChitChat system will first compute the current weight of each social interest in each user's TSRs by using the decay function defined in Equation 2.

$$w_u(\mathrm{SID}_i, t_s) = \begin{cases} \dfrac{w_u(\mathrm{SID}_i, t_{d,i})}{\beta \cdot (t_s - t_{d,i})}, & \\ \qquad\qquad\qquad \text{if } \mathrm{SID}_i \notin \mathrm{SP}_u; \\ \dfrac{\big(w_u(\mathrm{SID}_i, t_{d,i}) - 0.5\big)}{\beta \cdot (t_s - t_{d,i})} + 0.5, & \\ \qquad\qquad\qquad \text{if } \mathrm{SID}_i \in \mathrm{SP}_u. \end{cases} \tag{2}$$

The intuition behind the above decay function is that the longer a user is disconnected to those holding a positive weight for the social interest SID_i, the less likely this person will be able to successfully deliver a message with this social interest. Specifically, Equation 2 considers two cases: (i) $\mathrm{SID}_i \notin \mathrm{SP}_u$ means the social interest is not part of the user's social profile; and (ii) $\mathrm{SID}_i \in \mathrm{SP}_u$ means the social interest is in the user's social profile. The decay equation ensures that the weight of social interests from the user's social profile will never decrease below 0.5. In both cases, $t_{d,i}$ denotes the latest timestamp that the user was connected with some user with a positive weight for SID_i, implying that at time $(t_{d,i} + 1)$ they disconnected. The time difference $(t_s - t_{d,i})$ is a positive integer representing the number of seconds that has passed since u was last in contact with another user with a positive TSR weight for SID_i. The longer the user remains disconnected from users with SID_i, the lower the TSR weight of SID_i will be, whereby $1/\big(\beta \cdot (t_s - t_{d,i})\big)$ is the factor by which the weight undergoes decay. The parameter β is introduced to adjust the speed of decay such that $\beta \geq 1$. It is worth noting that in the case when a user remains connected to someone with a positively-weighted TSR for SID_i, the value of $t_{d,i}$ is equal to t_s. In such case, decay is not computed as it has been continuously reinforced up to t_s.

Lemma 1. *If a social interest is in a node's social profile, and its TSR weight at the last time of growth is between 0.5 and 1, then the decayed TSR weight is bounded between 0.5 and the previous value. Formally, if $SID_i \in SP_u$ and $0.5 \leq w_u(SID_i, t_{d,i}) \leq 1$, then $0.5 \leq w_u(SID_i, t) \leq w_u(SID_i, t_{d,i})$, $\forall t \geq t_{d,i} \geq t_0$ after decay (Eq. 2).*

Proof.

Base: At time t_0, $t_{d,i} = t_0$ and $w_u(\mathrm{SID}_i, t_0) = 0.5$ by Equation 1.

Inductive: Assume $t_{d,i} > t_0$ and $0.5 \leq w_u(\mathrm{SID}_i, t_{d,i}) \leq 1$. Let $t \geq t_{d,i}$ be the time a connection is established with u, kicking off the TSR decay. If $t = t_{d,i}$, then no decay will occur. Thus, assume $t - t_{d,i} \geq 1$.

$$w_u(\mathrm{SID}_i, t) = \frac{(w_u(\mathrm{SID}_i, t_{d,i}) - 0.5)}{\beta \cdot (t - t_{d,i})} + 0.5 \qquad \text{by Eq. 2}$$

$$0.5 \leq w_u(\mathrm{SID}_i, t_{d,i}) \leq 1$$

$$\implies 0 \leq w_u(\mathrm{SID}_i, t_{d,i}) - 0.5 \leq 0.5$$

$$\beta \geq 1 \text{ and } t - t_{d,i} \geq 1 \implies \beta \cdot (t - t_{d,i}) \geq 1$$

$$\implies 0 \leq \frac{w_u(\mathrm{SID}_i, t_{d,i}) - 0.5}{\beta \cdot (t - t_0)} \leq w_u(\mathrm{SID}_i, t_{d,i}) - 0.5$$

$$\implies 0.5 \leq \frac{(w_u(\mathrm{SID}_i, t_{d,i}) - 0.5)}{\beta \cdot (t - t_0)} + 0.5 \leq w_u(\mathrm{SID}_i, t_{d,i})$$

$$\therefore 0.5 \leq w_u(\mathrm{SID}_i, t) \leq w_u(\mathrm{SID}_i, t_{d,i})$$

□

Lemma 2. *If a social interest is not in a node's social profile, and its TSR weight at the last time of growth is between 0 and 1, then the decayed TSR weight is bounded between 0 and the previous value. Formally, if* $SID_i \notin SP_u$ *and* $0 \leq w_u(SID_i, t_{d,i}) \leq 1$*, then* $0 \leq w_u(SID_i, t) \leq w_u(SID_i, t_{d,i})$, $\forall t \geq t_{d,i} \geq t_0$ *after decay (Eq. 2).*

Proof.

Base: At time t_0, $t_{d,i} = t_0$ and $w_u(\mathrm{SID}_i, t_0) = 0$ by Equation 1.

Inductive: Assume $t_{d,i} > t_0$ and $0 \leq w_u(\text{SID}_i, t_{d,i}) \leq 1$. Let $t \geq t_{d,i}$ be the time a connection is established with u, kicking off the TSR decay. If $t = t_{d,i}$, then no decay will occur. Thus, assume $t - t_{d,i} \geq 1$.

$$w_u(\text{SID}_i, t) = \frac{w_u(\text{SID}_i, t_{d,i})}{\beta \cdot (t - t_{d,i})} \qquad \text{by Eq. 2}$$

$$\beta \geq 1 \text{ and } t - t_{d,i} \geq 1 \implies \beta \cdot (t - t_{d,i}) \geq 1$$

$$0 \leq w_u(\text{SID}_i, t_{d,i}) \leq 1 \text{ and } \beta \cdot (t - t_{d,i}) \geq 1$$

$$\implies 0 \leq \frac{w_u(\text{SID}_i, t_{d,i})}{\beta \cdot (t - t_{d,i})} \leq w_u(\text{SID}_i, t_{d,i})$$

$$\therefore 0 \leq w_u(\text{SID}_i, t) \leq w_u(\text{SID}_i, t_{d,i})$$

□

Corollary 1. *If a TSR weight is between 0 and 1 after its most recent growth, then the decayed TSR weight is bounded between 0 and the previous value. Formally, if* $0 \leq w_u(SID_i, t_{d,i}) \leq 1$, *then* $0 \leq w_u(SID_i, t) \leq w_u(SID_i, t_{d,i})$, $\forall t \geq t_{d,i} \geq t_0$ *after decay (Eq. 2).*

Proof. Corollary 1 trivially follows Lemma 1 and 2. □

At timestamp $t_s + 1$, user u will exchange his/her positively-weighted TSRs, consisting of only the social interests with positive weights, with the users $v_1, \ldots, v_k$ who newly connected to u at time t_s. Likewise, each neighbor $v_1, \ldots, v_k$ will exchange their TSRs with u. They will not exchange TSRs again for the duration of their uninterrupted connection. Here, the timestamp $t_s + 1$ ensures that only users who stay in contact with each other for at least one time unit will be considered during the social interest growth phase. Then, the weight of each social interest SID_i in u's TSRs will be modeled as a function of the current timestamp t_c (s.t. $t_c > t_s$) according to Equation 3.

$$w_u(\text{SID}_i, t_c) = \min\{1, w_u(\text{SID}_i, t_s) + \Delta\} \tag{3}$$

$$\Delta = \sum_{v \in \mathcal{v}} \frac{w_v(\mathrm{SID}_i, t_s) \cdot (t_c - t_s)}{\psi_{i,u,v}} \tag{4}$$

In Equations 3 and 4, $w_u(\mathrm{SID}_i, t_s)$ denotes the weight of SID_i in u's TSR at time t_s, $\mathcal{v}$ denotes the set of users in the communication range of u at time t_c, and $w_v(\mathrm{SID}_i, t_s)$ denotes the TSR weight of a user $v \in \mathcal{v}$ at time t_s when they start their interaction. The min function ensures that the growth would not exceed the upper bound. The growth function takes into account three factors: (i) the users' social interest weight (i.e., w_u, w_v) at the beginning of their interactions; (ii) the duration of the interaction, i.e., $(t_c - t_s)$; and (iii) the appropriate growth dampening factor $\psi_{i,u,v}$ which, as detailed below, is dependent on whether SID_i is a social profile resident or an induced TSR of both u and v. Regarding the first two factors, the higher the TSR weight and the longer the users remain in contact increases the growth of the corresponding TSR weight, i.e., the more likely that a message annotated with that social interest will be delivered. As for the third factor $\psi_{i,u,v}$, its value is determined based on the residency of the social interest SID_i in u's and v's social profile. In particular, we identify the following six cases in descending order of their impact on the social relationships in a DTN, i.e., the movement of users in DTN is driven by their social interests, or users with similar social interests may gather together more often.

- $\psi_{i,u,v} = 1$: This refers to the case when the social interest SID_i is a direct social interest of both u and v, meaning SID_i is in both of their social profiles (i.e. $\mathrm{SID}_i \in \mathrm{SP}_u$ and $\mathrm{SID}_i \in \mathrm{SP}_v$). In such case, u's TSR for SID_i will have no dampening due to $\psi_{i,u,v}$, obtaining the most growth to reflect the high likelihood they can forward messages annotated with SID_i.

- $\psi_{i,u,v} = 2$: This refers to the case when SID_i is in user u's social profile but not user v's. Rather it is a TSR that user v obtained when interacting with others.

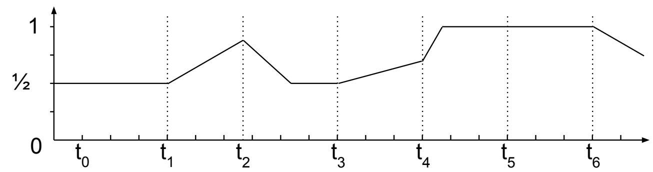
- $\psi_{i,u,v} = 3$: This refers to the case when SID_i is not user u's social profile, but is in user v's. Since SID_i is not a direct social interest for user u, the growth of its TSR weight is less than previous cases.

- $\psi_{i,u,v} = 4$: This refers to the case when SID_i is neither user u's nor v's social profile. Both u and v obtained this TSR from encountering others. This means there is some chance for user u and v to meet the people with this social interest, but the chance may be small since it is not these two users' direct interest.

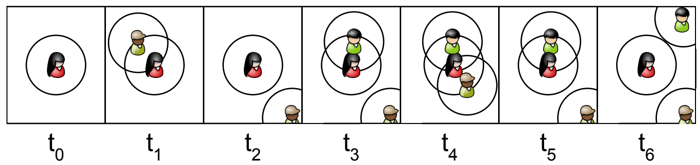
- $\psi_{i,u,v} = 5$: This refers to the case when SID_i is in user v's social profile and is a weightless TSR in u (i.e. $w_u(\text{SID}_i, t_s) = 0$). In such case, user u will expand its TSR by including this new social interest with a relatively low initial weight.

- $\psi_{i,u,v} = 6$: This refers to the case when SID_i exists only as an induced TSR of user v (i.e. $\text{SID}_i \notin \text{SP}_u \cup \text{SP}_v$, $w_u(\text{SID}_i, t_s) = 0$, and $w_v(\text{SID}_i, t_s) > 0$). This is the weakest case, yielding the least likelihood of message delivery for messages with this interest, and hence it is given the highest dampener.

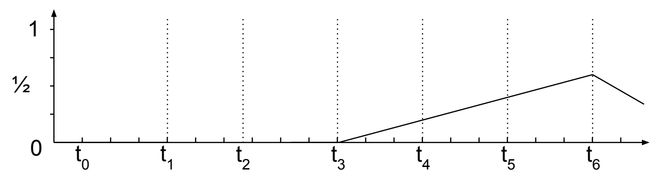
Algorithm 4 summarizes the main steps for the modeling Opportunistic Transient Social Relationships. To have a better understanding of the algorithm, let us step through an example as shown in Figure 4. Alice starts off with a social profile $SP_{alice}\langle 001, 003\rangle$ to express her interest in hiking and gourmet cooking. Thus, her TSR (shown in Figure 4a) begins at 0.5. She meets Bob at time t_1, who also has 'hiking' in his social profile (i.e. $\text{SP}_{bob}\langle 001\rangle$), which causes her 'hiking' TSR to grow according to Equations 3 and 4, with the dampening factor $\psi_{001,alice,bob} = 1$. At time t_2, Bob disconnects from Alice, resulting in her 'photography' TSR to begin decaying according to Equation 2. Enough time passes for her TSR to fall back to 0.5. Then, at time t_3, Alice encounters Carl, who is interested in hiking and photography (i.e. $SP_{carl}\langle 001, 002\rangle$). This causes Alice's TSR for 'hiking' (Figure 4a) to undergo growth with a dampener of $\psi_{001,alice,carl} = 1$. Likewise, a TSR for 'photography' is induced in Alice (Figure 4c) with a growth dampener of $\psi_{002,alice,carl} = 5$. At time t_4, Alice reconnects to Bob, causing a spike in the growth rate of her 'hiking'



(a) Alice's TSR corresponding to the interest 'hiking'. She has this TSR as part of her social profile, and thus its weight cannot go below 0.5.



(b) This diagram demonstrates the connections between Alice, Bob, and Carl that influence the TSR weights in Figure 4a and 4c.



(c) Alice's TSR corresponding to the interest 'photography'. She is not directly interested in traveling as it is not in her social profile. However, she interacts with people directly interested in it, and thus has a TSR with weight between 0 and 1.

Figure 4. An example of the growth and decay of a Transient Social Relationship caused by connections and disconnections with others.

TSR. Eventually, it maxes out with a weight of 1. At time t_5, Bob disconnects from Alice. However, since Alice is still connected to Carl, her TSR for 'hiking' and 'photography' does not undergo decay. Only when Carl disconnects at time t_6 do both TSRs undergo decay.

Lemma 3. *If the TSR weights for two nodes u and v are bounded between 0 and 1, then applying growth to u's TSR weight bounds it between its previous weight and 1. Formally, if Equation 3 is computed assuming* $t_c - t_s \geq 1$, $0 \leq w_u(SID_i, t_s) \leq 1$ *and* $0 \leq w_v(SID_i, t_s) \leq 1 \forall v \in v$, *then* $w_u(SID_i, t_s) \leq w_u(SID_i, t_c) \leq 1$.

Algorithm 4 Execute the OTSR module to decay and then grow the node u's TSRs based on the current time t_c and u's neighbors $v_{t_c} = \{v_1, \cdots, v_k\}$.

procedure UPDATETSRS(u, v_{t_c}, t_c)

▷ Decay TSRs according to Eq. 2

    for each $\mathrm{SID}_i \in \mathrm{TSR}_u$ **do**

        if $t_{d,i} == t_c$ **then** ▷ No decay to $w_u(\mathrm{SID}_i, t_c)$

            $w_u(\mathrm{SID}_i, t_c) = w_u(\mathrm{SID}_i, t_{d,i})$

        else if $\mathrm{SID}_i \in \mathrm{SP}_u$ **then**

            $w_u(\mathrm{SID}_i, t_c) = \dfrac{w_u(\mathrm{SID}_i, t_{d,i}) - 0.5}{\beta \cdot (t_c - t_{d,i})} + 0.5$

        else

            $w_u(\mathrm{SID}_i, t_c) = \dfrac{w_u(\mathrm{SID}_i, t_{d,i})}{\beta \cdot (t_c - t_{d,i})}$

        end if

    end for

▷ Exchange current TSRs with new neighbors.

    for each node $v \in v_{t_c}$ **do**

        Send TSR_u to v

        Receive TSR_v from v

        $t_{s,v} = t_c$

    end for

▷ Grow TSRs according to Eq. 3

    for each $\mathrm{SID}_i \in$ all received and cached TSRs **do**

        $\Delta = \sum_{v \in v} \dfrac{w_v(SID_i, t_{s,v}) \cdot (t_c - t_{s,v})}{\psi_{i,u,v}}$

        $w_u(\mathrm{SID}_i, t_c + 1) = \min\{1, w_u(\mathrm{SID}_i, t_c) + \Delta\}$

    end for

end procedure

Proof.

$$w_u(\mathrm{SID}_i, t_c) = \min\{1, w_u(\mathrm{SID}_i, t_s) + \Delta\} \quad \text{by Equation 3}$$

$$\Delta = \sum_{v \in \mathcal{V}} \frac{w_v(\mathrm{SID}_i, t_s) \cdot (t_c - t_s)}{\psi_{i,u,v}} \quad \text{by Equation 4}$$

$$\psi_{i,u,v} \geq 1 \text{ and } t_c - t_s \geq 1 \text{ and } 0 \leq w_v(\mathrm{SID}_i, t_s) \leq 1$$

$$\implies \sum_{v \in \mathcal{V}} \frac{w_v(\mathrm{SID}_i, t_s) \cdot (t_c - t_s)}{\psi_{i,u,v}} \geq 0 \implies \Delta \geq 0$$

$$\implies w_u(\mathrm{SID}_i, t_s) \leq w_u(\mathrm{SID}_i, t_s) + \Delta$$

$$w_u(\mathrm{SID}_i, t_s) + \Delta \leq \min\{1, w_u(\mathrm{SID}_i, t_s) + \Delta\} \leq 1$$

$$\therefore w_u(\mathrm{SID}_i, t_s) \leq w_u(\mathrm{SID}_i, t_c) \leq 1$$

□

Our modeling ensures the following properties.

Theorem 3. *Let u be a user and TSR_u be the user's Transient Social Relationships. The TSR weight for any social interest is bounded between 0 and 1 at all times after network initialization. Formally, for any $SID_i \in TSR_u$ and any timestamp t such that $t > t_0$, then $0 \leq w_u(SID_i, t) \leq 1$.*

Proof.

Base: Assume that at time $t > t_0$, $t_{d,i} = t_0$, implying that user u has made its first connection to its neighbors $\mathcal{V}_t = \langle v_1, \cdots, v_k \rangle$. Thus, by Equation 1, its TSR weights are either 0 or 0.5 for social interests either absent from or within its social profile – i.e. $w_u(\mathrm{SID}_i, t_0) = 0$ or $w_u(\mathrm{SID}_i, t_0) = 0.5$. With this, Corrolary 1 guarantees that, after decay is applied (lines 2 to 10 in Algorithm 5), the TSR weight $w_u(\mathrm{SID}_i, t)$ is lower-bounded by 0 and upper-bounded by the previous weight value – i.e. $0 \leq w_u(\mathrm{SID}_i, t) \leq w_u(\mathrm{SID}_i, t_0)$. Then, the TSRs of the neighboring users in $\mathcal{V}_t$ are used to induce growth in $w_u(\mathrm{SID}_i, t+1)$, according to lines 16 to 19 in Algorithm 5. Here, Lemma 3 guarantees that $w_u(\mathrm{SID}_i, t+1)$ is lower bounded by the previous (decayed) value, and upper bounded by 1 – i.e. $w_u(\mathrm{SID}_i, t) \leq w_u(\mathrm{SID}_i, t+1) \leq 1$.

Because of this, and with Lemma 1 and Lemma 2 lower-bounding a TSR by 0, we know that $0 \leq w_u(\mathrm{SID}_i, t+1) \leq 1$. Therefore, after the first initiation of Algorithm 5, the final value of a TSR weight is bounded between 0 and 1 – i.e. $0 \leq w_u(\mathrm{SID}_i, t+1) \leq 1$. This would also be the case for every other connected neighbor in v_t.

Inductive: At time t ($t > t_0$), assume $t_{d,i} > t_0$, implying that user u has made previous connections that have induced decay and growth. At time t, user u connects with its neighbors $v_t = \langle v_1, \cdots, v_k \rangle$, where we assume both u and its neighbors' TSR weights are bounded between 0 and 1 – i.e. $0 \leq w_u(\mathrm{SID}_i, t_{d,i}) \leq 1$ and $0 \leq w_v(\mathrm{SID}_i, t_{d,i}) \leq 1$ for each connected neighbor $v \in v_t$. Then, Corrolary 1 guarantees that, after decay is applied (lines 2 to 10 in Algorithm 5), the value of $w_u(\mathrm{SID}_i, t)$ is lower-bounded by 0 and upper-bounded by the previous weight value – i.e. $0 \leq w_u(\mathrm{SID}_i, t) \leq w_u(\mathrm{SID}_i, t_{d,i})$. Likewise, Lemma 3 guarantees that $w_u(\mathrm{SID}_i, t+1)$ is lower bounded by the previous (decayed) value, and upper bounded by 1 – i.e. $w_u(\mathrm{SID}_i, t) \leq w_u(\mathrm{SID}_i, t+1) \leq 1$. Therefore, after Algorithm 5 is applied, the TSR weight is bounded between 0 and 1 – i.e. $0 \leq w_u(\mathrm{SID}_i, t+1) \leq 1$. This would also be the case for every other connected neighbor in v_t. □

Theorem 4. *Each node has a storage complexity of $O(N(log(N+1)+b))$ for TSRs.*

Proof. There are N unique social interests that users may define in their social profiles. Each social interest ID requires $log(N+1)$ bits of storage. Each TSR weight is bounded between 0 and 1 as proven in Theorem 3. The storage complexity of a TSR weight is thus defined by the bit-precision b allocated for the weights. Thus, each TSR id-weight pair occupies $log(N+1)+b$ bits of storage.

The TSR weights of each node are aggregates of the TSRs of the nodes one has encountered. If all N social interests are defined throughout the social profiles across all nodes, and if a user connects to all other nodes participating in the PSN, then the user will have TSRs for all N social interests, consuming $O(N(log(N+1)+b))$ bits of storage. □

With nodes sharing information that may be sensitive with one another, one concern of this system's employment is the preservation of the nodes' privacy. This topic is out of the scope of this paper, but is left as future work in Section 5.1.

2.4. CHITCHAT ROUTING PROTOCOL

At a timestamp t_c when a node is connected to others, it will commence its GSH and TSR updates based on its current neighbors. Afterward, it will attempt to forward messages as deemed appropriate by its routing algorithm. These steps are defined in Algorithm 5, with the specifics of the routing algorithm defined in Algorithm 6.

Algorithm 5 A new connection is established between node u and nodes $v_{t_c} = \{v_1, \cdots, v_k\}$ at time t_c, initiating the Opportunistic Geographic Social Heatmap and Opportunistic Transient Social Relationship modules.

procedure CONNECT(u, v_{t_c}, t_c)
▷ Update u's GSH according to Alg. 3
    for each $v \in v_{t_c}$ **do**
        MERGEGSH (u, v)
    end for
▷ Exchange current TSRs with new neighbors.
    UPDATETSRS (u, v_{t_c}, t_c)
▷ Consider u's messages for relaying to neighbors.
    for each msg carried by u **do**
        ATTEMPTMESSAGEFORWARD (u, msg, v_{t_c})
    end for
end procedure

The ChitChat system supports the following unicast and multicast scenarios in a DTN environment.

- *Unicast*: A sender sends a message to a designated message receiver. In this case, the sender knows the message receiver's ID and annotates the message to reflect content the recipient would be interested in.

- *Multicast*: A sender sends a message to a group of users whose social interests match a message's metadata keywords. In this case, the sender does not need to know the message receivers' IDs.

Messages are distributed in a multi-copy fashion. When a node connects with a neighbor, the node may create a replica of a message it is carrying and forward that replica to the neighbor. The sending node retains a copy of the message, only deleting it when its time to live (TTL) has expired, when the node successfully delivers a copy of the message to the destination, or when its buffer is saturated and room is needed for other messages with better delivery potential.

The message forwarding phase occurs after GSHs and TSRs are updated. Since each node may hold a set of interest-tagged messages, and each node has limited message storage and forwarding capabilities, it is critical to determine which message to forward to which neighboring node so that the overall delivery rate is not penalized. To achieve this, we propose the following routing protocol in Algorithm 6 that leverages the knowledge carried by the newly updated GSHs and TSRs.

The routing protocol takes as input a node u that wants to forward a message m that has social interests $msg.\texttt{SIDs} = \{\mathrm{SID}_1, \ldots, \mathrm{SID}_m\}$, and a set of nodes $v = \{v_1, \ldots, v_n\}$ that are within u's communication range and just completed exchanging GSHs and TSRs with u. First, u will check if the designated message receiver (in the unicast case) is among the neighboring nodes. If so, u will immediately pass the message only to the receiver and move on to other messages. Otherwise, the protocol proceeds as follows. Node u will rule out those neighboring nodes whose message buffer are full, and thus are unable to receive any more messages, and those who already hold m in their buffers. For each remaining neighbor $v_k \in v$, node u computes v_k's TSR strength $S_{\mathrm{TSR}}(v_k)$ (line 8) and itinerary strength $S_I(v_k)$ (line 13), relative to the message of interest, and compares them with its own. If either of v_k's TSR strength or itinerary strength is greater than u's, then the message will be forwarded.

Algorithm 6 ChitChat Routing Algorithm: Node u decides which connected neighbors in v to forward the message msg to at time t.

procedure $\textsc{AttemptMessageForward}(u, msg, v)$
    for $v_k \in v$ **do**
        if $msg.\texttt{destination} == v_k$ **then**
            Forward msg to v_k
            $u.\texttt{messages} = u.\texttt{messages} - msg$
        else if $msg \notin v_k.\texttt{messages}$ **then**
            $S_{\mathrm{TSR}}(u) = \sum_{\mathrm{SID}_i \in msg.\texttt{SIDs}} w_u(\mathrm{SID}_i, t)$
            $S_{\mathrm{TSR}}(v_k) = \sum_{\mathrm{SID}_i \in msg.\texttt{SIDs}} w_{v_k}(\mathrm{SID}_i, t)$
            if $S_{\mathrm{TSR}}(v_k) > S_{\mathrm{TSR}}(u)$ **then**
                Forward m to v_k
            end if
            $S_I(u) \leftarrow \textsc{ComputeItineraryStrength}\,(u, msg)$
            $S_I(v_k) \leftarrow \textsc{ComputeItineraryStrength}\,(v_k, msg)$
            if $S_I(v_k) > S_I(u)$ **then**
                Forward m to v_k
            end if
        end if
    end for
end procedure
procedure $\textsc{ComputeItineraryStrength}(u, m)$
    $S_I(u) \leftarrow 0$
    $G \leftarrow u.\textsc{get}\mathrm{GSH}\,()$
    $I \leftarrow u.\textsc{getCurrentItinerary}\,()$
    $I_{t_c} \leftarrow \{\langle x_i, y_i, t_i\rangle \mid \langle x_i, y_i, t_i\rangle \in I, t_i > t_c\}$
    for $S \in I_{t_c}$ **do**
        for $S_{nn} \in G.\textsc{nearestNeighbors}\,(S, 3)$ **do**
            $d \leftarrow \textsc{max}\,(1, \textsc{distanceBetween}\,(S, S_{nn}))$
            for $\mathrm{SID}_i \in msg.\texttt{SIDs}$ **do**
                $S_I(u) \leftarrow S_I(u) + S_{nn}.\textsc{getSocialWeight}\,(\mathrm{SID}_i)\,/d^2$
            end for
        end for
    end for
    return $S_I(u)$
end procedure

Both a node's TSR strength and itinerary strength relative to a message are measurements of a node's potential capability to pass along the message to interested parties. The TSR strength is the sum of the node's TSR weights corresponding to the social tags of the message (lines 7 and 8). Recalling back to Section 2.3, a high weight value of one particular TSR of a node reflects that node's consistency in encountering others with a similarly high TSR weight. Thus, a message carrier encountering a node with a higher TSR strength for a message would do well to forward the message.

The itinerary strength of a node to a message is a conceptually different measurement of a node's delivery capability. Whereas the TSR strength is constructed based on node contacts, the itinerary strength is constructed based on node locations. Pulling from a node's current itinerary and opportunistically-learned Geographic Social Heatmap, the itinerary strength is the distance-weighted sum of the social weights for the areas through which a node will soon be traveling, only pertaining to those social weights that correspond to social tags of the message. The ComputeItineraryStrength procedure in Algorithm 6 defines the steps in calculating a node's itinerary strength relative to a given message. When a node u is determining whether to forward a message m to its neighbor v, both u and v will compute their own itinerary strengths relative to m, and v will share its strength with u (line 13). This definition permits the computation of a node's itinerary strength without requiring the divulging of its current itinerary to its neighbor.

Without loss of generality, let u be the node computing its own Itinerary weight. u initializes its weight $S_I(u)$ to 0 and grabs a reference to its current GSH (lines 21 and 22), followed by obtaining the points along its itinerary that have not been visited yet (lines 23–24). For each of these remaining points in the itinerary, the three social staypoints defined in u's GSH that are nearest to the itinerary point are obtained and iterated over (line 26). The subset of social weights on these staypoints, corresponding to the tags on the message, are then weighted by the inverse-squared distance between the itinerary point and the social staypoint and added to $S_I(u)$ (lines 26–31). Execution ends with the computed itinerary strength being returned (line 33).

The inverse-squared distance weighting of the social weights is necessary due to the opportunistic nature of a node's GSH. A node does not have global knowledge of the exact social markup of every location in the DTN's coverage area. Rather, this information is fragmented and learned opportunistically. In order to compute an arbitrary location's social weights, the three nearest social staypoints are obtained from a node's GSH, and their weights are used to induce the social weights of the queried location. If a social staypoint is very close, then the inverse-squared distance weight will be higher than those further away. Thus, the social weights of closer staypoints will have greater influence on the queried location than those of further staypoints, more accurately estimating the social markup of a given region as best as is permissible.

3. PERFORMANCE EVALUATION

In this section, we first introduce the experimental settings that were used for testing and comparing the ChitChat system with other state-of-the-art systems. Then, we present the network density analysis on five real-world datasets that are useful for simulating DTNs. Finally, we report the results of our simulations.

All experiments were conducted in the ONE simulator [16] version 1.6.0. We compare our ChitChat system with one benchmark algorithm (i.e., Epidemic [33]), and three recent related works: SEDUM [21], SANE [25], and SEBAR [19]. Additionally, the proposed implementation of ChitChat in [24], upon which this paper's ChitChat is based and expands, is compared for observing improvements on performance.

In the experiments, we evaluate the performance of each algorithm by using two real GPS trajectory datasets: Microsoft's GeoLife GPS trajectory dataset [36, 37, 38] and Nokia's Mobile Data Challenge dataset [17, 18]. Within both datasets are GPS trajectories consisting of timestamped sequences of latitude and longitude points, recording how a user moved during the period of time their smart device was active. The GeoLife dataset consists of the trajectories of 182 users intermittently over a period of five years in the city of Beijing, China. The Mobile Data Challenge (MDC) dataset similarly captures the trajectories of 185 users over a period of a year and a half in the city of Lausanne, Switzerland. Defining a trajectory to be a consecutive sequence of locations of some user where no two consecutive points are more than 10 minutes apart, the Geolife dataset provides 41,543 trajectories covering 1,282,951 kilometers over 2.8 person-years; the MDC dataset provides 761,463 trajectories covering 1,795,349 kilometers over 46.5 person-years.

These datasets were further processed so as to permit the simulation of a network with varying participation of a city's population. Compared to Beijing's population of approximately 21.5 million and Lausanne's population of approximately 138 thousand individuals, the number of participants present in these datasets is significantly small. Each of these users were scattered across the cities and rarely come within close proximity to each other. In order to form a network to evaluate and compare the aforementioned routing protocols, trajectories from the same person occurring on different days were gathered together and treated as unique individuals. For example, if Alice contributed a GPS trajectory for Monday, Tuesday, and Friday, then the processed datasets used for these experiments would present three users moving about on the same day: AliceMonday, AliceTuesday, and AliceFriday. This view, which we call the *daily slicing* of the datasets, changes the number of participants in each dataset that are available for the simulations: the processed GeoLife dataset presents 11,149 unique individuals, and the processed MDC dataset presents 41,931 unique individuals. From this, we then isolate the trajectories consisting of at least two hours of contiguous recordings and at least 500 unique location records that occur within the core metropolitan area of each city. This approach produces datasets that constitute approximately 60% of the unique individuals from the Geolife dataset (6,656 out of 11,149 unique trajectories) and approximately 14.9% from the MDC dataset

(6,259 out of 41,931). The resulting datasets provide a node density of approximately 16 to 20 nodes per square kilometer throughout the day. These modifications to the datasets preserve the statistical properties of human mobility based on the findings in [2] and [29]. In [2], Barbosa et. al. survey the literature on human mobility models and cite multiple studies that observe human mobility exhibiting the statistical properties of Lévy walks. Further, Rhee et. al. performed a similar daily slicing on four GPS trajectory datasets and found the statistical properties of Lévy walks were preserved in the processed datasets [29].

Since the GeoLife and the MDC datasets contain no social profiles of their participating users, social profiles are randomly generated from a set of 200 predefined social interests with SID values of 1, 2, ..., 200. Since the ChitChat system does not consider the semantic meaning of social interests, the keywords associated with each social interest are simply "1", "2", ..., "200". Social profiles of each node are generated by uniformly selecting k social interests with replacement out of the pool of 200, where $k = 25$ unless otherwise stated in discussion. A particular node has the same k social interests in their profile for all simulations in which they participate where k remains constant across simulations. Each user's Transient Social Relationships are initialized from their social profiles, with each weight being initialized following Equation 1. TSR weight decay and growth is conducted throughout the simulation following Equations 2 and 3–4, respectively, as defined by Algorithm 4. We also randomly generate messages in the network as follows. For each message, we randomly select its sender and receiver among the participants, and keep the pairs for which there is at least one shared social interest. Then, we randomly select a subset of social interests from the chosen destination to attach to the message. The total simulation time covers 24 hours of GPS trajectory replay, and one message is generated every 60 seconds.

To compare the performance of each chosen algorithm, we adopt the following performance metrics: (i) message delivery ratio – the ratio of number of messages delivered to the number of messages created; (ii) the average network penetration for messages – the average number of hops to deliver each message; (iii) the network's throughput – the number of messages that can be delivered per hour; (iv) the total number of forwarding attempts made, reflecting the amount of power each system used; and (v) the average overhead cost ratio. The average overhead cost ratio is defined in the prepackaged MessageStatsReport class of the ONE simulator [16], which is the number of copies of all messages that were necessary to deliver the number of messages that were delivered. It is calculated as $c = \frac{r-d}{d}$, where r is the total number of message transmissions that occurred in the simulation, and d is the number of messages that were successfully delivered to its destination. Whereas message delivery ratio, network throughput, and overhead cost are employed for measuring a DTN's quality of service [1], also observing the network penetration and number of forwarding attempts allow for a deeper insight on each system's operations in achieving their quality of service. These performance metrics are evaluated when certain control parameters are varied, as has been previously used in past DTN research [1]: the number of users in the network (i.e., node count), the transmission range of these nodes, and the message lifetimes (i.e., time-to-live). Additionally, the length and diversity of social profile sizes was varied to observe any changes in system performance.

In what follows, as others, we report the performance results of the unicast versions of these protocols due to the lack of multicast simulation in the ONE simulator. Unless otherwise stated, the settings in Table 1 were used across all experiments conducted. We used the default settings for SANE, SEDUM, and SEBAR as recommended in [25], [21], and [19], respectively, with the exception of the maximum number of message replicas, which we permit to be unbounded to fairly compare ChitChat with these other systems.

3.1. SPARSITY ANALYSIS OF REAL-WORLD DATASETS

It is worth mentioning that many past works have used spatially dense datasets for experimental evaluation. Additionally, these datasets are typically constrained to a special event at a specific location and not sufficient to represent general human daily mobility. For example, the INFOCOM 2006 [30] dataset has 76 participants moving within an academic

Table 1. Default Experimental Settings

Configuration	Default Experimentation Values
Number of Participants	2,000 nodes
Freq. of Location Reporting	1/5 $\sec^{-1}$
Social Profile Sizes	25 per node
Available Social Interests	200
Transmission speed	250 kBps
Transmission radius	50 meters
Buffer capacity	500 MB
Message TTL	10 hours
Freq. of Message Creation	1/60 $\sec^{-1}$
Message Size	1 MB
Simulated time	24 hours
SANE Relay Threshold	0.25
SANE Message Replicas	Unbounded
SEDUM Epoch Duration	1 minute
SEDUM Weight Constant	0.2
SEDUM Message Replicas	Unbounded
SEBAR τ Parameter	1 minute
SEBAR p Parameter	0.9
SEBAR ξ Parameter	0.2
SEBAR Community Overlap	14
SEBAR Community Duration	30 mins
SEBAR Message Replicas	Unbounded

conference venue of approximately only 80 meters by 40 meters, resulting in the density as high as 20,000 people per square kilometer. Likewise, the SIGCOMM 2009 [27] dataset has similar number of participants, density, and conference venue locality. In these datasets, the participants attending the conference are away from their hometowns, thus influencing their habitual movements. The MIT Reality Mining dataset [9] does not provide the information needed to compute node density, but with only a cohort of college students and faculty contributing data, there is concern that their movements might be biased towards academic life.

To better understand the sparsity of these datasets, we analyzed the temporal reachability of all five previously mentioned datasets. At a high level, this analysis quantifies the the density of a time-evolving network by measuring the percentage of node pairs that are connected through fragmented, multi-hop paths. Our analysis adopts the formal foundations proposed in [3, 5].

First, we cover some preliminary formal definitions based on those proposed in [5]. A Delay Tolerant Network may be cast as a Time-Varying Graph (TVG) $\mathcal{G} = (V, E, \mathcal{T}, \rho, \zeta)$, where V is the set of nodes, $\mathcal{T} \subseteq \mathbb{T}$ is a timespan within the temporal domain $\mathbb{T}$ representing the lifetime of the network, E is the set of edges representing network connections that exist at some time instant during $\mathcal{T}$, $\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$ is a presence function that indicates whether a given edge exists at a given time instant, and $\zeta : E \times \mathcal{T} \rightarrow \mathbb{T}$ is a latency function that indicates the time duration needed to traverse the given edge starting at the given timestamp. A *journey* connects two nodes through multiple hops, where at each time instant the multi-hop path between the nodes is not necessarily connected. Formally, a journey $J^{u,v}$ in $\mathcal{G}$ is a sequence of timestamped edges $\left(\langle e_1 = \{u, w'\}, \mathcal{T}_1 = (t_1^s, t_1^e)\rangle, \langle e_2, \mathcal{T}_2\rangle, ..., \langle e_k, \mathcal{T}_k\rangle\right)$ such that $u \in e_1$, $v \in e_k$, $\rho(e_i, t) = 1 \; \forall \, t \in \mathcal{T}_i$ and $t_{i+1}^s \geq t_i^s + \zeta(e_i, t_i^s)$ for all $0 < i < k$. Let $m = \langle t_c, t_d, \texttt{payload}\rangle$ be a message, where $t_c \in \mathcal{T}$ is message's creation time, $t_k \in \mathcal{T}, t_d > t_c$ is the time when the message is no longer useful, and `payload` is the content of the message. A message m is deliverable from a node u to another node v through $\mathcal{G}$ if and only if there exists a journey $J^{u,v}$ such that $t_c < t_1^d$ and $t_d > t_k^s$ – i.e., the message is created before the first connection ends and expires after the last connection begins.

We next define the density of a TVG as proposed in [3]. Given a TVG $\mathcal{G} = (V, E, \mathcal{T}, \rho, \zeta)$ and some timespan $\mathcal{T}' = [t_i, t_j] \subseteq \mathcal{T}$, the *transitive closure* of $\mathcal{G}$ during $\mathcal{T}'$ is $C_{\mathcal{T}'} = (\mathcal{V}_{\mathcal{T}'}, \mathcal{J}_{\mathcal{T}'})$, where $\mathcal{V}_{\mathcal{T}'} \subseteq V$ is the set of nodes that were active in $\mathcal{G}$ during any time $t \in \mathcal{T}'$, and $\mathcal{J}_{\mathcal{T}'} = \{(u, v) \mid \exists \, J^{u,v} \text{ s.t. } t_1^s > t_i, t_k^d < t_j\}$ where t_1^s and t_k^d are the starting and ending time of the journey, respectively. Essentially, the transitive closure of a TVG during a certain timespan is the set of journeys existing, and the nodes participating, in the

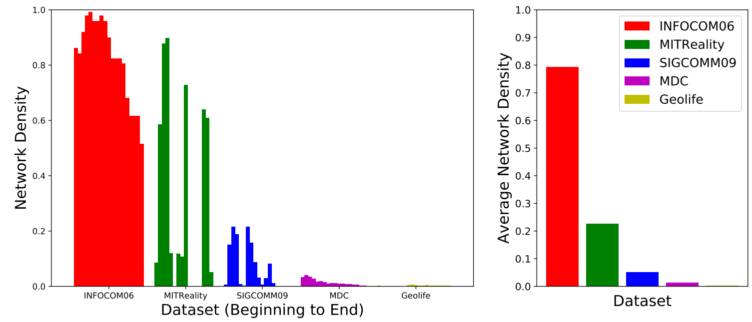
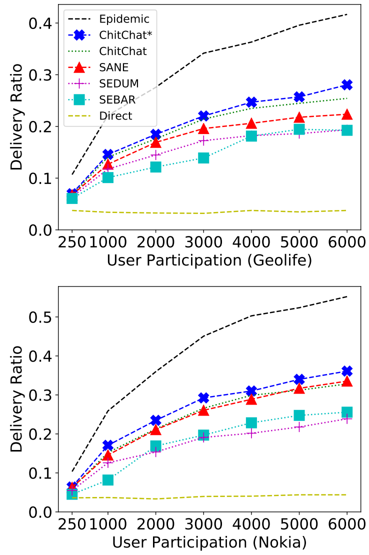


Figure 5. Network densities of the DTNs formed by five datasets: INFOCOM06, MIT Reality Mining, SIGCOMM09, Microsoft's Geolife, Nokia's MDC. On the left, network density is plotted over the duration of each dataset. Each data point represents the number of the journeys existing in a 20-minute span of the network, starting some time after the beginning of the datset, over the size of a complete network formed by all active nodes. The right plot shows the average network density across the entire dataset.

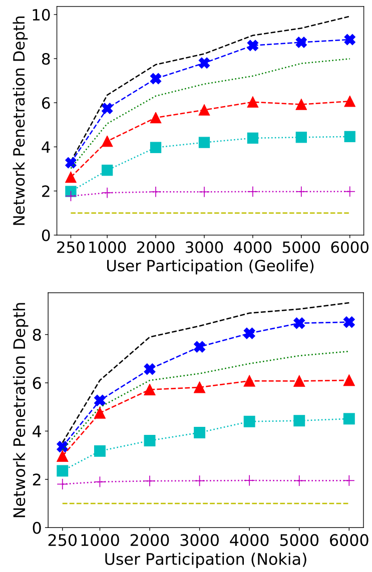
network during that timespan. The *density* of the transitive closure $C = (\mathcal{V}, \mathcal{J})$ is $\frac{|\mathcal{J}|}{|\mathcal{V}|(|\mathcal{V}|-1)}$ – i.e., the proportion of the node pairs that are connectable through journeys over the size of a complete graph composed of active nodes.

Given these definitions, Figure 5 displays the density of the networks formed by the five datasets described previously. The INFOCOM 2006 dataset, SIGCOMM 2009 dataset, and the MIT Reality Mining datasets all exhibited connections based on Bluetooth proximity; thus, if a node detected the presence of another, it can be inferred that the two nodes were within 10 meters of each other. For the Geolife and MDC datasets, we used the daily-sliced versions of each dataset with a random sampling of 2,000 users, and connections between nodes were inferred when two nodes were within 50 meters of one another.

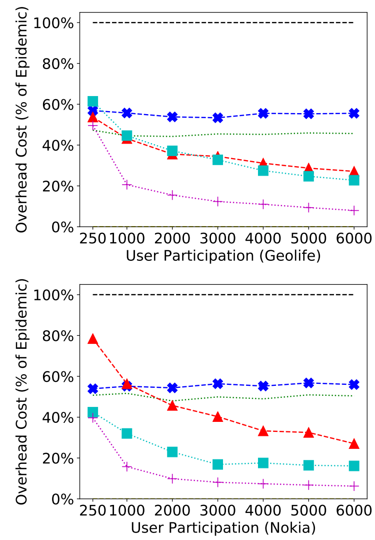
In Figure 5, each datapoint represents the number of journeys that exist in a 20-minute span of the network, starting from various times after the beginning, over the size of the network were it completely connected. It is apparent that the INFOCOM 2006 dataset has the highest density across time, with an average network density of 79.3%. The second highest is MIT Reality Mining dataset at certain times, but overall it exhibited an average network density of 22.7%. The two least dense (i.e., most sparse) datasets are Nokia's MDC



(a) Message Delivery Ratio



(b) Average Number of Hops



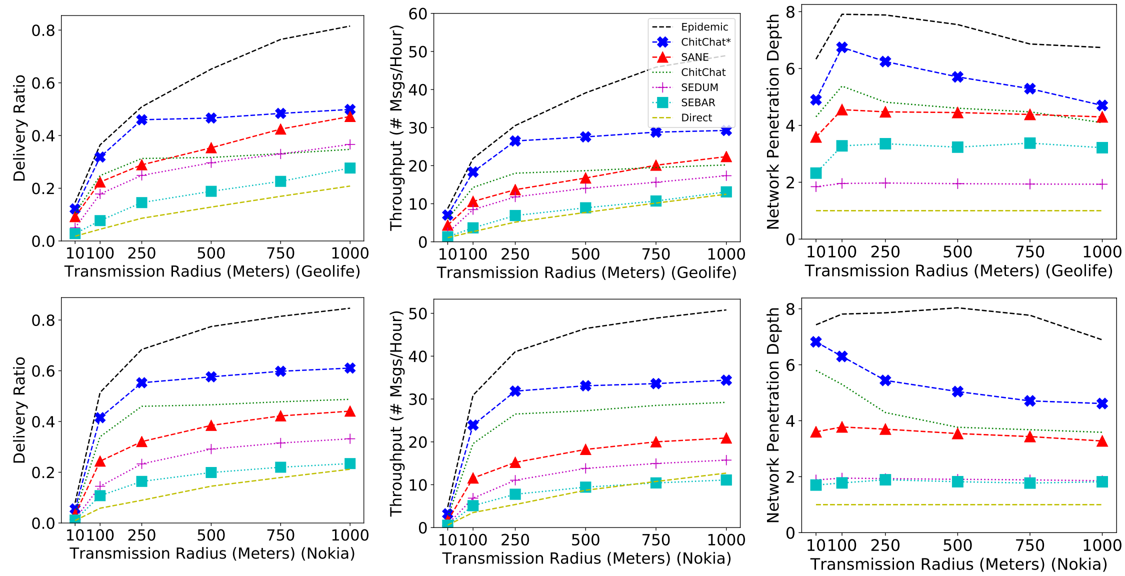
(c) Overhead Costs relative to Epidemic

Figure 6. Effect of user participation on performance. The top figures use the Geolife dataset, while the bottom figures use the MDC datasets. Note that ChitChat* is this paper's proposed system, whereas ChitChat refers to the preliminary implementation proposed in [24].

and Microsoft's Geolife, with average densities of 1.3% and 0.3% respectively. It is because of these properties that we chose the GeoLife and MDC datasets in our experiments. These datasets record a broader range of human mobility during a much longer time period, thus better representing users' movement in their socially habitual manner, and are not biased toward academic life as is the case for the other three datasets.

3.2. EFFECT OF THE NUMBER OF USERS IN THE NETWORK

In the first round of experiments, we evaluate the routing performance of all algorithms by varying the total number of users in the network from 250 to 6,000. This round of simulations investigates the behavior and performance of each algorithm under varying user sparsity. Intuitively, a decrease in the number of participants results in a lower user density, thus influencing the network's sparsity as fewer nodes come sufficiently close to each other



(a) Message Delivery Ratio (b) Network Throughput (c) Average Number of Hops

Figure 7. Effect of transmission radius on performance. The top figures use the Geolife dataset, while the bottom figures use the MDC datasets. The legend in the top-right graph applies to all graphs. Note that ChitChat* is this paper's proposed system, whereas ChitChat refers to the preliminary implementation proposed in [24].

to form connections. Figure 6 reports the results of these simulations. It is not surprising to see that Epidemic achieves the highest delivery ratio since it floods messages to every node encountered. This results in messages reaching deeper into the network and increasing the overall resource consumption to operate the network. Direct Delivery achieves the lowest delivery ratio since message forwards do not occur between nodes unless a message's destination is encountered directly. In turn, Direct Delivery has the lowest network penetration and overhead costs, and intuitively the highest latency.

Our proposed ChitChat is the best performing system besides Epidemic in terms of successful deliveries by achieving up to 64.3% of Epidemic's performance for the MDC-based network and up to 68% for the Geolife-based network. As for overhead costs, ChitChat consistently requires approximately half of the costs associated with Epidemic. This is due to ChitChat selectively choosing which messages to send to which encountered nodes, rather than indiscriminately forwarding all to every neighbor, thus resulting in a reduction in overhead and resource consumption. The selection process for message forwarding is able to choose, in a best effort manner considering the constraints of a DTN, better nodes to receive the messages. When compared to the preliminary verion of ChitChat proposed in [24], which did not consider node itineraries nor geographic social heatmaps, the improved ChitChat exhibits increases in its delivery performance of up to 12.2% (MDC) and up to 10.3% (Geolife) while increasing overhead by up to 13.3% (MDC) and up to 25.1% (Geolife). Thus, the introduction of node itineraries and opportunistically-learned geographic social heatmaps offers an improvement in delivery performance, albeit at the cost of additional overhead. ChitChat's successful deliveries exceeds that of the other three routing algorithms: SEBAR (up to 2x more successful deliveries for MDC and up to 58.4% more for Geolife), SANE (up to 17.1% more for MDC and up to 25.5% more for Geolife), and SEDUM (up to 56.1% more for MDC and 45% more for Geolife). SANE considers only direct social interests between two users and SEDUM considers only the duration of interaction time, which are less effective in sparser networks.

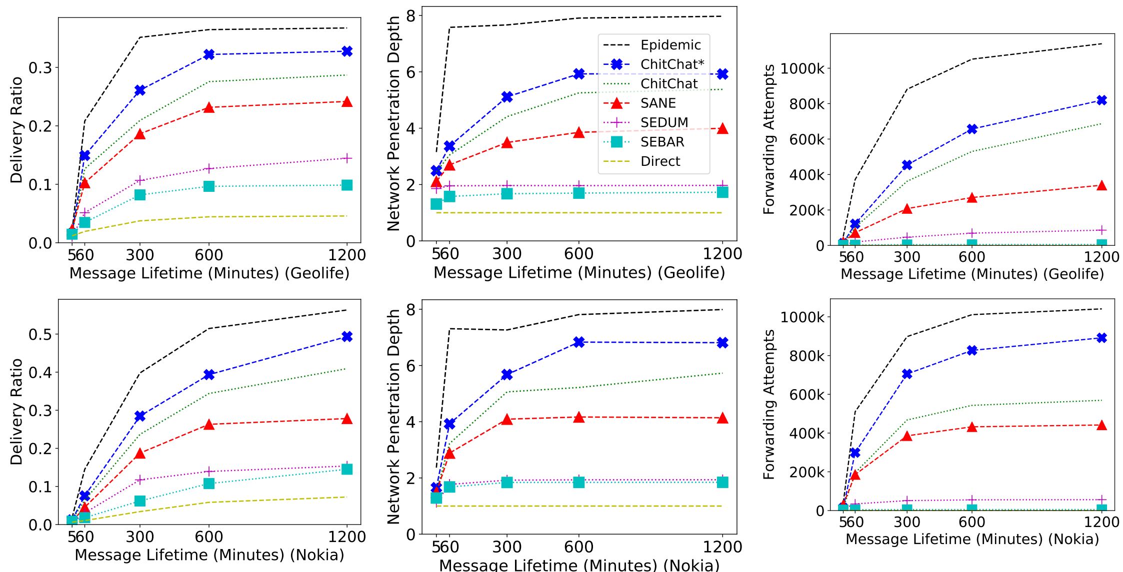
Regarding the number of hops to deliver a message (Figure 6(b)) and the resource costs (Figure 6(c)), a lower value does not necessarily imply efficient operation of a router. Although SEDUM, SANE, SEBAR are shown to require lower overhead and fewer hops, and their costs decrease as the number of users increases, it does not mean they operate with more efficiency. Rather, the low hop counts observed for SANE, SEDUM, and SEBAR are attributable to their ability to only deliver to nearby nodes, failing to deliver messages that are destined for nodes residing deeper in the network. Similarly, the lower overhead is attributable to them not attempting many message forwards, which is detremental to message delivery within a sparse DTN.

3.3. EFFECT ON TRANSMISSION RANGE

In the second round of simulations, we evaluate the routing performance in a subset of 2,000 users by varying the communication range from 10 meters to 1 kilometer[4]. Similar to the simulations in Section 3.2, the motivation for this round of simulations is to observe the effect of variable network sparsity on the behavior and performance of each protocol. As shown in Figure 7(a), the delivery ratio of all systems increases with the communication range as expanded communication ranges yield increasing likelihoods of finding proper forwarding nodes. Epidemic still achieves the highest delivery ratio, and our proposed ChitChat is the second best in all cases. It is interesting to see that, for both datasets, the performance of ChitChat stabilizes when the communication range reaches 250 meters, while all others continue to grow. This indicates the effectiveness of the modeling of Transient Social Relationships and Geographic Social Heatmap in the ChitChat system, and suggests that the need to further improve communication ranges is not necessary to boost performance when employing ChitChat. This is also apparent when observing the throughput of the network – the number of messages that are delivered per time unit – as is depicted in Figure 7(b). With ChitChat, achieving a certain target throughput requires a smaller radius of communication when compared to the other systems. This, in turn, indicates that less power would be needed for wireless transmissions, which is a critical and finite resource for battery-powered smart devices.

As for the average number of hops per message in Figure 7(c), we observe that ChitChat is able to reach deeper into the network than others, and the depth it needs to reach decreases with the increase of the communication range. The reason is straightforward: when communication ranges are large, multi-hop paths become shorter as fewer intermediate nodes are needed to reach a destination. However, SANE, SEDUM, and SEBAR do not exhibit this decreasing behavior, but rather remain consistent. This suggests they

[4]It is worth noting that 1 kilometer of communication range may not be possible in a real DTN, and is considered here to test the behavior and performance of all approaches under extreme network conditions.



(a) Message Delivery Ratio (b) Average Number of Hops (c) Forwarding Attempts

Figure 8. Effect of message lifetime on performance.

reuse the same network paths for delivery, and are unable to capitalize on newly-available edges to more efficiently deliver messages. Conversely, ChitChat's modeling of multi-hop relationships allows it to exploit these new edges to find shorter journeys to a message's destination.

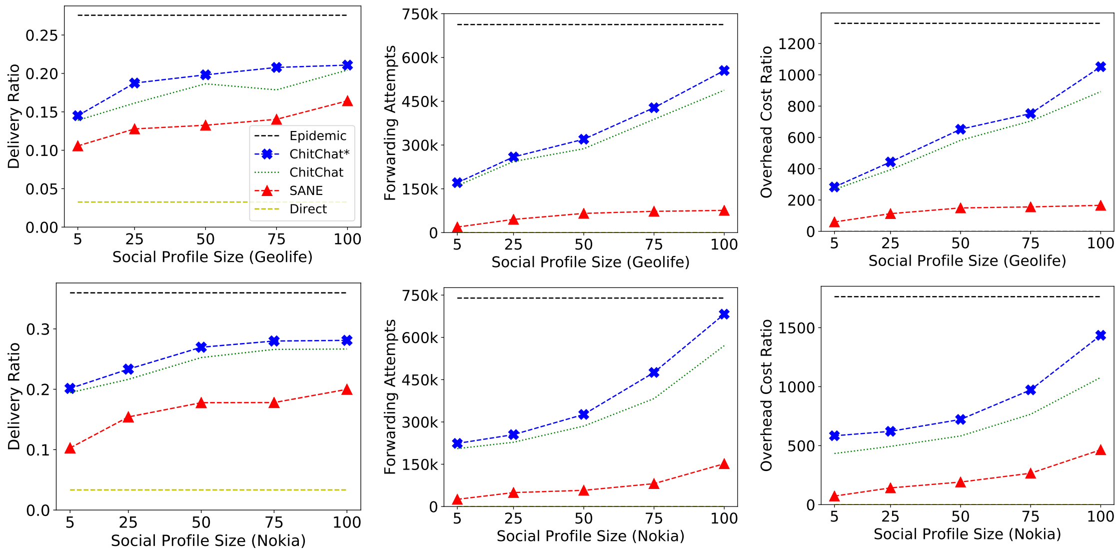
In Figure 7(c), the throughput of the network is plotted against the changing of transmission ranges. Similar to the stabilizing depicted in Figure 7(a), the ChitChat system is able to reach a throughput that is higher than all other systems without demanding a larger communication range.

3.4. EFFECT OF MESSAGE LIFETIME

Next, we examine how message lifetimes affect the performance and behavior of each algorithm. Figures 8(a)–(c) shows the results of this suite of simulations as the message lifetimes range from 5 minutes to 20 hours. This suite of simulations emulates the reachability of nodes within the network given that a certain duration of time is provided to reach them, and thus offers a glimpse of the degree of *delay tolerance* that DTN applications must adopt to operate.

A common pattern in these graphs is a plateauing of the dependent variables as message lifetimes are increased. As is evident from the message penetrations by Epidemic (Figure 8(b)), starting at one hour, messages cannot reach beyond 8-hops from the source on average. This is a property of the networks induced by the two datasets: at any given time, the number of new encounters that a node makes stagnates after about an hour. Waiting any longer does not bring new encounters that would otherwise offer opportunities to reach other disconnected parts of the network. Rather, messages are only deliverable within a node's local neighborhood, and for timespans longer than one hour, the set of nodes that are encountered by a given node remains consistent.

This property appears to harm the performance of SANE, SEDUM, and SEBAR. As depicted in Figure 8(b), SEBAR and SEDUM are unable to send messages beyond two hops into both networks after one hour, and SANE is unable to send messages beyond four hops. However, ChitChat's penetration of the network continues to increase as a message's lifetime extends up to 10 hours, and in turn its successful deliveries also increase. This is due to the dynamic growth and decay of ChitChat's TSR modeling, and the introduction of opportunistically-learned GSHs also leads to improvements above ChitChat's predecessor. ChitChat's dynamic TSRs and GSHs allow a node to recognize that an encountered neighbor, who may not have appeared qualified to receive a message during an earlier encounter, has become qualified because of its new itinerary or its other encounters during the elapsed timeframe. This is apparent in Figure 8(c), which plots the number of forwarding attempts made by all nodes given the message lifetimes. Although nodes are encountering many of the same nodes over and over again, both versions of ChitChat continue to make more forwarding attempts as messages lifespans increase. SANE, SEDUM, and SEBAR do not; their forwarding attempts stagnate after about five hours, as does their number of



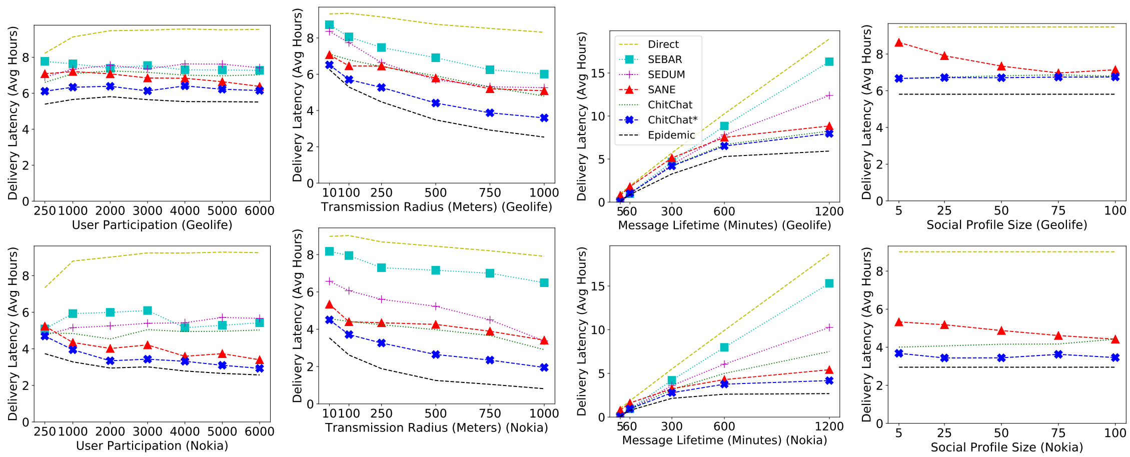
(a) Message Delivery Ratio (b) Forwarding Attempts (c) Network Overhead

Figure 9. Effect of social profile sizes on performance.

successful deliveries. SANE relies on static social profiles to make its routing decisions; if an encountered neighbor isn't qualified once, it won't be qualified later. The same goes for SEDUM and SEBAR.

3.5. EFFECT OF SOCIAL INTERESTS DISTRIBUTION

Finally, we evaluate the effect of nodes' social interests declarations on the performance of the two systems that use the social interests: ChitChat and SANE. Specifically, we vary the number of social interests associated with each user from 5 to 100 out of a pre-defined set of 200 social interests. The motivation for this is to consider the scenario where users restrict the number of social interests they share, perhaps due to privacy concerns, or the population of participants exhibits rather homogeneous social interests, perhaps due to these systems being deployed to a specific target audience with many shared interests. As shown in Figure 9(a), ChitChat exceeds SANE in successful deliveries in all cases, with increases in deliveries as social profile sizes increase. However, with ChitChat these



(a) Participation (b) Transmission Radius (c) Message Lifetime (d) Social Profile Size

Figure 10. Influence of various factors on delivery latency.

increases to successful deliveries taper out at larger social profile sizes, even though the number of forwarding attempts (Figure 9(b)) and overhead costs of the network (Figure 9(c)) grows more rapidly. These results are due to social profile homogeneity: the total number of available social interests remains constant, but as the number of social interests per node increases, so too do their social profiles become more homogeneous as nodes share more social interests in common. In turn, the selectivity of ChitChat to make intelligent routing decisions goes into decline as it observes more nodes that appear qualified to deliver messages. This is apparent from the increases in forwarding attempts and overhead costs. In other words, these findings suggest that ChitChat consumes more resources when the diversity and distinctiveness of the network's participants declines.

3.6. DELIVERY LATENCIES

Message delivery latency – the amount of time between a message's creation and its successful delivery – is an important factor for DTNs in determining how *tolerant* applications must be to *delays*. As such, this section explores the latencies observed in Figures 10(a)–(d) for the simulations described before. The Epidemic scheme and the

Direct Delivery scheme obtain the lowest and highest latencies, respectively, and offer referential bounds on the performance of more sophisticated systems. Observing these, the Geolife-induced network exhibits higher and tighter bounds on latency than the MDC-induced network. This is attributable to the sparsity of these networks, as was discussed in Section 3.1; Geolife has a greater degree of sparsity than MDC, thus leading to messages taking longer to traverse the network to reach their destinations.

As depicted in these figures, the ChitChat system is able to obtain a lower latency than all other non-flooding systems, with some scenarios having the average difference between ChitChat and Epidemic being as low as 20 minutes. This achievement is primarily due to the journeys through which messages were delivered. ChitChat's employment of Transient Social Relationships, node itineraries, and Geographic Social Heatmaps results in it taking advantage of certain connections that other systems may ignore, including those occurring earlier in a message's lifetime. By sending a message through these connections, there is a higher likelihood the message will reach its destination earlier.

4. RELATED WORK

4.1. ROUTING STRATEGIES FOR DTNS

Many unique DTN routing algorithms have reported in past works. Epidemic routing [33] provides an upper bound for the number of successfully delivered messages as well as a lower bound on the delivery delay if one assumes light traffic or infinitely fast data transmission and unlimited buffer sizes. This is not a reasonable assumption, however, as past work [15] has shown that imposing finite transmission speeds and buffer sizes can result in network degradation when using Epidemic. Thus, for the sake of reducing the resources consumed, any routing mechanism that is deployed needs to identify if a connected neighbor is a worthwhile candidate for message forwarding, and if so, which messages should be forwarded.

Spray and Wait [31] accomplishes controlled overhead costs by imposing a hard limit on the number of message copies that are permitted throughout the network. It quickly distributes a strict limit of message replicas to intermediate nodes and then waits for one of them to reach the destination and deliver the message. PRoPHET [22] makes intermediate message forwarding decisions based on the observed probability that an intermediate node will meet with the destination, and maintains the freshness of these probabilities through the use of a weighted, convex combination of past and current probabilities. BUBBLE Rap [12] has messages *bubble up* through the network to higher-centrality nodes, reaching more popular nodes until it enters the destination's community. Once there, the forwarding strategy shifts its focus from global centrality to community-centric centrality, i.e., centrality with nodes of that community. Hui et al. reckon a node's popularity within its community is more effective at reaching the destination than the node's global popularity when the node shares a community with the destination.

Recent developments have adopted other metrics based on those previously mentioned for choosing intermediate message relays. Whereas PRoPHET calculates contact probabilities based on the rate of contacts per time period, the SEDUM router [21] expresses a similar metric by using continuous contact durations between any two nodes during a time period. The message carrier forwards a message to another node if the recipient has a higher utility with the destination than the current carrier. Utilities are also transitively spread throughout the network in an opportunistic manner, only being passed between nodes when connections are established. This permits an intermediate node to have a high utility with another node even if the two nodes never directly meet. Rather, the two nodes meet indirectly through one or more intermediate relays. Expanding on BUBBLE Rap's use of centrality, the DAS router [35] utilizes the destination-aware betweenness centrality to route messages, and further uses this metric to dynamically cap the permissible number of copies of these messages. Encountered nodes that reside on fewer multi-hop shortest-paths to a message's destination are permitted to relay fewer copies of the message than those that have more multi-hop paths, thus curbing message delivery overhead.

Another recent shift in DTN research focuses on exploiting social artifacts. Community membership and detection has been the primary focus of several works [4, 10, 12, 19], whereby forwarding to a node may occur if the node is within a community shared with the destination. In [19], the SEBAR algorithm adopts the social energy routing metric that grows and decays by direct encounters between nodes and is influenced by the duration of these contacts and the social activity of the nodes' communities. When two nodes encounter one another, a quantity of social energy is generated. This energy is then partially applied directly to the node and partially distributed to nodes within its communities. Social energies undergo decay in a similar manner as adopted by PRoPHET. When considering message forwarding, the number of copies of each message is capped and halved at each forwarding, and a two-phased routing strategy is adopted similar to SEDUM. While a message is outside of all of the destination's communities, a node carrying the message considers its neighbor's social energy; if the neighbor has a higher social energy than the message carrier, the message will be forwarded. Once the message has reached a node within a community of the destination, the second phase only forwards to other nodes also within a community of the destination, specifically to those that are members of communities with higher social activity than the message carrier.

The SANE algorithm [25] requires each individual to hold a binary string of equal length that can be translated into the set of unique keywords that describe the user's interests. When two individuals meet, a message carrier computes the cosine similarity of its social interest vector to its neighbor's social interests, and decides to forward the message if the similarity exceeds a predefined threshold. To the best of our knowledge, [25] is the first to investigate the utility of social interests as a decisive measure in message forwarding in DTNs, and thus offers a fair comparison for our proposed work.

In our previous study [24], the modeling and dissemination of transient social relationships served as an effective means to facilitate message delivery in sparse DTNs. However, it did not consider whether participants were traveling to locations that may benefit from receiving certain messages. The addition of location information has been demonstrated as an effective factor for DTN routing [11], with routing decisions being made with the consideration of where people are traveling to in the near future. In this work, we extend our previous study [24] to augment the ChitChat system by considering both a node's transient social relationships and their current travel itinerary, and propose a distributed method for learning the social semantics of locations within the network's geographic area of operation.

4.2. INTRINSIC DENSITY OF PREVIOUSLY EVALUATED DATASETS

These past works have limitations on successfully delivering messages in spatial-temporal sparse environments, where users are sparsely distributed throughout a geographic area or few members of that area are participating in the DTN. The evaluations of past works have primarily been conducted on either dense real-world datasets, such as those showing the contact traces of academic attendees of a conference [27, 30] or students' movements through a college campus [9, 26], or synthetic datasets ranging in their degree of sparsity. As analyzed in Section 3.1, the confines and constituents of these datasets produce networks that have a significantly high population density and frequent contacts between participants, resulting in higher temporal reachability of the networks. Table 2 summarizes the datasets used by the surveyed DTN routers and the resulting densities of the dataset populations. Of note, most strategies are evaluated against the academic conference-based datasets [10, 12, 19, 23, 25, 28] that are confidened to conference venues with the highest density of nodes. Five strategies employ the MIT Reality Mining dataset [4, 7, 10, 12, 19] consisting of students and faculty on a college campus, where students of the same discipline are likely to encounter one another through their daily course schedules. Those systems

Table 2. Summary of Dataset Usage and Density of DTN Routers

Router	Dataset	Nodes	Network Area	Node Density
Epidemic [33]	Synthetic	50	1500m × 300m (0.45 km^2)	≈ 111/km^2
PRoPHET [22]	Synthetic	50	1500m × 300m (0.45 km^2)	≈ 111/km^2
	Synthetic	60	3000m × 1500m (4.5 km^2)	≈ 13/km^2
SimBet [7]	MIT Reality	100	Unknown (≈ 0.672 km^2)*	≈ 149/km^2
BUBBLERap [12]	INFOCOM06	76	80m × 40m (0.0032 km^2)*	≈ 23750/km^2
	MIT Reality	100	Unknown (≈ 0.672 km^2)*	≈ 149/km^2
MobiClique [28]	SIGCOMM09	76	82m × 69m (≈ 0.0057 km^2)*	≈ 13333/km^2
DSG-N^2 [4]	MIT Reality	100	Unknown (≈ 0.672 km^2)*	≈ 149/km^2
Gao et. al. [10]	INFOCOM06	76	80m × 40m (0.0032 km^2)*	≈ 23750/km^2
	MIT Reality	100	Unknown (≈ 0.672 km^2)*	≈ 149/km^2
SEDUM [21]	Synthetic	150	2000m × 2000m (4 km^2)	38/km^2
SANE [25]	INFOCOM06	76	80m × 40m (0.0032 km^2)*	≈ 23750/km^2
	Synthetic	200	1000m × 1000m (1 km^2)	200/km^2
Machado et. al. [23]	SIGCOMM09	76	82m × 69m (≈ 0.0057 km^2)*	≈ 13333/km^2
SEBAR [19]	INFOCOM06	76	80m × 40m (0.0032 km^2)*	≈ 23750/km^2
	MIT Reality	100	Unknown (≈ 0.672 km^2)*	≈ 149/km^2
DAS [35]	Synthetic	128	Unknown extent/area	N/A

*Dimensions and area of network activity is estimated based on the dimensions of the conference venue or college campus.

employing synthesized networks [21, 22, 25, 33] exhibit varying degrees of sparsity, but do not present statistical properties that match human mobility. Specifically, the models used to synthesize the movement of nodes follow various random walk strategies such as Random Waypoint (RWP) [13, 22, 33] and models derived from it, such as Manhattan Walks [21] and community-based mobility models [22, 25]. RWP-based movement models have been shown to favor more frequent pairwise contacts between nodes over those found in real-world datasets, resulting in lower routing delays and higher throughput by an order of magnetude [29]. Thus, even though some of these exhibit a similar degree of sparsity as is focused on in this paper, the statistical properties of the employed models remains inconsistent with those found in real-world datasets and may not exhibit movement patterns that are representative to socially-motivated people.

As such, sparse networks formed by natural human mobility have not been investigated extensively, such as is present in metropolitan environments where a small subset of the population is participating in the network. This makes it unlikely that short paths exist between a source and destination. Rather, geographically distant individuals are connected by long multi-hop paths, with many intermediate nodes, spanning long periods of time. With probabilistic routing [21, 22], the chances of an intermediate node having ever contacted the destination is very slim under this scenario. Community-based routing [4, 10, 12] may suffer in a similar manner if an intermediate node does not share any communities with a message's destination. What is needed is a routing mechanism that can successfully percolate social relationship information throughout the network, thus permitting routing decisions to occur in a sparsely-connected network.

5. CONCLUSION AND FUTURE WORK

In this paper, we present a novel social-context aware routing protocol, the ChitChat system, for sparsely-connected Delay Tolerant Networks (DTNs), which exist in many real applications like battlefield or disaster response for timely situational-awareness. Our proposed ChitChat system successfully models multi-hop social relationships via a novel decay-growth model, introduces a method for opportunistically learning the social semantics of locations using node itineraries, and enables each participating node to make informed decisions during message routing. Our analysis of the network densities of five real world datasets identified the two most widely used datasets for DTN research form considerably dense and connected networks, which suggests the problem of network sparsity in DTNs has not been previously addressed. Our experimental study using ONE simulator with two sparse real world datasets demonstrates the superiority of our proposed approach compared to recent existing efforts, in that our ChitChat system achieves higher successful deliveries with reduced communication overhead (thus reduced energy usage).

As part of our future work, we have multiple of avenues to investigate. Of primary interest is to evaluate how well ChitChat performs when people's social profiles exhibit various distributions and geographic correlations of interests. With a recent shift in research toward DTN security, topics such as trust, incentives, fault detection, and provenance have inevitable problems that require investigating [6, 8, 14, 20] for applications in a battlefield environment. As a future work, ChitChat may be extended with attribute and role-based access policies for more effective data dissemination in battlefield and disaster management environment. Another topic of interest is to investigate the real-time augmentation of message metadata annotations. Such an application would greatly assist battlefield reconnaissance and intelligence gathering by speeding up the turn-around between raw field data to rich intelligence acquisition, thus facilitating faster turn around in wartime strategies.

5.1. PRIVACY OF USER SOCIAL INTERESTS

Finally, privacy preservation of a node's TSRs and the semantic information they represent is of great interest for those who wish to participate in the DTN without divulging their associations (either directly through their social interests or indirectly through their frequent encounters with others) to certain topics. The ChitChat system intrinsically offers some degree of privacy through its representation of social interests and the exchange of information. First, TSRs offer a degree of privacy through the uncertainty of whether a TSR is a direct social interest of a node or if it was induced by others. For example, assume Alice is an HIV-negative doctor who tends to the care of HIV-positive patients but she is does not express it as an interest in her social profile. Further, in her free time she is very engaged in photography (i.e., she has a direct social interest in it). If Bob encounters Alice, he receives her TSRs showing high weights for both 'photography' and 'HIV+'. Following Algorithm 4, a node's TSRs are created either from a node's direct social interest or from the TSRs of nodes it has previously encountered. Thus, Bob cannot confidently deduce whether Alice is HIV-positive, nor can he deduce whether she is directly interested in photography. All Bob knows is that there is a strong association between Alice and these keywords. Another intrinsic property of TSRs is that they do not necessarily have semantic meaning that nodes can derive and understand. As shown in Definitions 6, social interests are represented in a node's TSRs by a unique identifier (e.g., a hexadecimal string, an integer, etc) and not by a semantic keyword. In certain scenarios, nodes need not know the association between an social interest identifier (SID) and its associated keyword. For instance, a sensitive message may be encrypted and annotated with encrypted keywords that only authorized recipients can decrypt. These encrypted keywords would be used as the SIDs for their TSRs that are distributed through the network, with unauthorized network participants being able to aid in the distribution of the encrypted messages without learning of their contents. Further investigation into privacy preservation in the ChitChat system is left for future consideration.

Although the ChitChat system has some intrinsic properties that aid in preserving user privacy, further research is needed to demonstrate its effectiveness against attacks and propose stronger solutions. Topics such as homomorphic encryption and secure multiparty computation may augment the ChitChat system to eliminate these problems.

ACKNOWLEDGEMENT

Portions of the research in this paper used the MDC Database made available by Idiap Research Institute, Switzerland and owned by Nokia.

REFERENCES

[1] Abdelkader, T., Naik, K., Nayak, A., Goel, N., and Srivastava, V., ‘A performance comparison of delay-tolerant network routing protocols,’ IEEE Network, 2016, **30**(2), pp. 46–53, ISSN 0890-8044, doi:10.1109/MNET.2016.7437024.

[2] Barbosa, H., Barthelemy, M., Ghoshal, G., James, C. R., Lenormand, M., Louail, T., Menezes, R., Ramasco, J. J., Simini, F., and Tomasini, M., 'Human mobility: Models and applications,' Physics Reports, 2018, **734**, pp. 1 – 74, ISSN 0370-1573, doi:https://doi.org/10.1016/j.physrep.2018.01.001, human mobility: Models and applications.

[3] Bedogni, L., Fiore, M., and Glacet, C., 'Temporal reachability in vehicular networks,' in 'IEEE INFOCOM 2018 - IEEE Conference on Computer Communications,' 2018 pp. 81–89, doi:10.1109/INFOCOM.2018.8486393.

[4] Cabaniss, R., Vulli, S. S., and Madria, S., 'Social group detection based routing in delay tolerant networks,' Wireless Networks, 2013, **19**(8), pp. 1979–1993, ISSN 1022-0038, doi:10.1007/s11276-013-0580-2.

[5] Casteigts, A., Flocchini, P., Quattrociocchi, W., and Santoro, N., 'Time-varying graphs and dynamic networks,' International Journal of Parallel, Emergent and Distributed Systems, 2012, **27**(5), pp. 387–408, doi:10.1080/17445760.2012.668546.

[6] Cho, J. and Chen, I., 'Provest: Provenance-based trust model for delay tolerant networks,' IEEE Transactions on Dependable and Secure Computing, 2018, **15**(1), pp. 151–165, ISSN 1545-5971, doi:10.1109/TDSC.2016.2530705.

[7] Daly, E. M. and Haahr, M., 'Social network analysis for routing in disconnected delay-tolerant manets,' in 'Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing,' MobiHoc '07, ACM, New York, NY, USA, ISBN 978-1-59593-684-4, 2007 pp. 32–40, doi:10.1145/1288107.1288113.

[8] Datta, S., Madria, S., Milligan, J., and Linderman, M., 'Secure information forwarding through fragmentation in delay-tolerant networks,' in '2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS),' ISSN 2575-8462, 2018 pp. 93–102, doi:10.1109/SRDS.2018.00020.

[9] Eagle, N., Pentland, A. S., and Lazer, D., 'Inferring friendship network structure by using mobile phone data,' Proceedings of the National Academy of Sciences, 2009, **106**(36), pp. 15274–15278, doi:10.1073/pnas.0900282106.

[10] Gao, W., Cao, G., La Porta, T., and Han, J., 'On exploiting transient social contact patterns for data forwarding in delay-tolerant networks,' IEEE Transactions on Mobile Computing, 2013, **12**(1), pp. 151–165, ISSN 1536-1233, doi:10.1109/TMC.2011.249.

[11] Guo, H., Wang, X., Cheng, H., and Huang, M., 'A location aided controlled spraying routing algorithm for delay tolerant networks,' Ad Hoc Networks, 2017, **66**, pp. 16 – 25, ISSN 1570-8705, doi:https://doi.org/10.1016/j.adhoc.2017.08.005.

[12] Hui, P., Crowcroft, J., and Yoneki, E., 'Bubble rap: Social-based forwarding in delay-tolerant networks,' IEEE Transactions on Mobile Computing, 2011, **10**(11), pp. 1576–1589, ISSN 1536-1233, doi:10.1109/TMC.2010.246.

[13] Jain, S., Fall, K., and Patra, R., 'Routing in a delay tolerant network,' in 'Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications,' SIGCOMM '04, ACM, New York, NY, USA, ISBN 1-58113-862-8, 2004 pp. 145–158, doi:10.1145/1015467.1015484.

[14] Jethawa, H. and Madria, S., 'Reputation and credit based incentive mechanism for data-centric message delivery in dtns,' in '2018 19th IEEE International Conference on Mobile Data Management (MDM),' ISSN 2375-0324, 2018 pp. 207–216, doi: 10.1109/MDM.2018.00038.

[15] Jindal, A. and Psounis, K., 'Performance analysis of epidemic routing under contention,' in 'Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing,' IWCMC '06, ACM, New York, NY, USA, ISBN 1-59593-306-9, 2006 pp. 539–544, doi:10.1145/1143549.1143657.

[16] Keränen, A., Ott, J., and Kärkkäinen, T., 'The ONE Simulator for DTN Protocol Evaluation,' in 'Proceedings of the 2nd International Conference on Simulation Tools and Techniques,' SIMUTools '09, ICST, New York, NY, USA, ISBN 978-963-9799-45-5, 2009 .

[17] Kiukkonen, N., Blom, J., Dousse, O., Gatica-Perez, D., and Laurila, J., 'Towards rich mobile phone datasets: Lausanne data collection campaign,' in 'Proceedings of the 2010 ACM International Conference on Pervasive Services,' 2010 .

[18] Laurila, J., Gatica-Perez, D., Aad, I., Blom, J., Bornet, O., Do, T.-M.-T., Dousse, O., Eberle, J., and Miettinen, M., 'The mobile data challenge: Big data for mobile computing research,' in 'Pervasive Computing,' 2012 .

[19] Li, F., Jiang, H., Li, H., Cheng, Y., and Wang, Y., 'Sebar: Social-energy-based routing for mobile social delay-tolerant networks,' IEEE Transactions on Vehicular Technology, 2017, **66**(8), pp. 7195–7206, ISSN 0018-9545, doi:10.1109/TVT.2017.2653843.

[20] Li, W., Galluccio, L., Bassi, F., and Kieffer, M., 'Distributed faulty node detection in delay tolerant networks: Design and analysis,' IEEE Transactions on Mobile Computing, 2018, **17**(4), pp. 831–844, ISSN 1536-1233, doi:10.1109/TMC.2017.2743703.

[21] Li, Z. and Shen, H., 'Sedum: Exploiting social networks in utility-based distributed routing for dtns,' IEEE Transactions on Computers, 2013, **62**(1), pp. 83–97, ISSN 0018-9340, doi:10.1109/TC.2011.232.

[22] Lindgren, A., Doria, A., and Schelén, O., 'Probabilistic routing in intermittently connected networks,' in 'Service Assurance with Partial and Intermittent Resources,' volume 3126 of *Lecture Notes in Computer Science*, pp. 239–254, Springer Berlin Heidelberg, ISBN 978-3-540-22567-6, 2004.

[23] Machado, K., Boukerche, A., de Melo, P. O. V., Cerqueira, E., and Loureiro, A. A., 'Pervasive forwarding mechanism for mobile social networks,' Computer Networks, 2016, **111**, pp. 6 – 16, ISSN 1389-1286, doi: https://doi.org/10.1016/j.comnet.2016.08.022, cyber-physical systems for Mobile Opportunistic Networking in Proximity (MNP).

[24] McGeehan, D., Lin, D., and Madria, S., 'Chitchat: An effective message delivery method in sparse pocket-switched networks,' in '2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS),' IEEE Computer Society, Los Alamitos, CA, USA, ISSN 1063-6927, 2016 pp. 457–466, doi: 10.1109/ICDCS.2016.69.

[25] Mei, A., Morabito, G., Santi, P., and Stefa, J., 'Social-aware stateless routingin pocket switched networks,' IEEE Transactions on Parallel and Distributed Systems, 2015, **26**(1), pp. 252–261, ISSN 1045-9219, doi:10.1109/TPDS.2014.2307857.

[26] Nahrstedt, K. and Vu, L., 'CRAWDAD dataset uiuc/uim (v. 2012-01-24),' Downloaded from http://crawdad.org/uiuc/uim/20120124, 2012, doi:10.15783/C7HG6D.

[27] Pietilainen, A.-K. and Diot, C., 'CRAWDAD data set thlab/sigcomm2009 (v. 2012-07-15),' Downloaded from http://crawdad.org/thlab/sigcomm2009/, 2012.

[28] Pietilänen, A.-K. and Diot, C., 'Dissemination in opportunistic social networks: The role of temporal communities,' in 'Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing,' MobiHoc '12, ACM, New York, NY, USA, ISBN 978-1-4503-1281-3, 2012 pp. 165–174, doi: 10.1145/2248371.2248396.

[29] Rhee, I., Shin, M., Hong, S., Lee, K., Kim, S. J., and Chong, S., 'On the levy-walk nature of human mobility,' IEEE/ACM Transactions on Networking, 2011, **19**(3), pp. 630–643, doi:10.1109/TNET.2011.2120618.

[30] Scott, J., Gass, R., Crowcroft, J., Hui, P., Diot, C., and Chaintreau, A., 'CRAWDAD dataset cambridge/haggle (v. 2009-05-29),' Downloaded from http://crawdad.org/cambridge/haggle/20090529, 2009, doi:10.15783/C70011.

[31] Spyropoulos, T., Psounis, K., and Raghavendra, C. S., 'Spray and wait: An efficient routing scheme for intermittently connected mobile networks,' in 'Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking,' WDTN '05, ACM, New York, NY, USA, ISBN 1-59593-026-4, 2005 pp. 252–259, doi: 10.1145/1080139.1080143.

[32] Spyropoulos, T., Turletti, T., and Obraczka, K., 'Routing in delay-tolerant networks comprising heterogeneous node populations,' IEEE Transactions on Mobile Computing, 2009, **8**(8), pp. 1132–1147, ISSN 1536-1233, doi:10.1109/TMC.2008.172.

[33] Vahdat, A. and Becker, D., 'Epidemic routing for partially-connected ad hoc networks,' Technical Report CS-20000, Duke University, 2000.

[34] Zhang, B., Teng, J., Bai, X., Yang, Z., and Xuan, D., 'P^3-coupon: A probabilistic system for prompt and privacy-preserving electronic coupon distribution,' in '2011 IEEE International Conference on Pervasive Computing and Communications (PerCom),' 2011 pp. 93–101, doi:10.1109/PERCOM.2011.5767599.

[35] Zhang, J., Huang, H., Yang, C., Liu, J., Fan, Y., and Yang, G., 'Destination-aware metric based social routing for mobile opportunistic networks,' Wireless Networks, 2019, ISSN 1572-8196, doi:10.1007/s11276-018-01907-2.

[36] Zheng, Y., Li, Q., Chen, Y., Xie, X., and Ma, W.-Y., 'Understanding mobility based on gps data,' in 'Proceedings of the 10th International Conference on Ubiquitous Computing,' UbiComp '08, ACM, New York, NY, USA, ISBN 978-1-60558-136-1, 2008 pp. 312–321, doi:10.1145/1409635.1409677.

[37] Zheng, Y., Xie, X., and Ma, W.-Y., 'Geolife: A collaborative social networking service among user, location and trajectory,' IEEE Data Engineering Bulletin, 2010.

[38] Zheng, Y., Zhang, L., Xie, X., and Ma, W.-Y., 'Mining interesting locations and travel sequences from gps trajectories,' in 'Proceedings of the 18th International Conference on World Wide Web,' WWW '09, ACM, New York, NY, USA, ISBN 978-1-60558-487-4, 2009 pp. 791–800, doi:10.1145/1526709.1526816.

[39] Zhu, Y., Xu, B., Shi, X., and Wang, Y., 'A survey of social-based routing in delay tolerant networks: Positive and negative social effects,' IEEE Communications Surveys Tutorials, 2013, **15**(1), pp. 387–401, ISSN 1553-877X, doi: 10.1109/SURV.2012.032612.00004.

III. CATORA: CONGESTION AVOIDANCE THROUGH TRANSMISSION ORDERING AND RESOURCE AWARENESS IN DELAY TOLERANT NETWORKS

D. McGeehan, S. Madria

ABSTRACT

The proliferation of wireless mobile devices encourages research into their employment to form Delay Tolerant Networks (DTN) for such applications as disaster response, military communications, and crowdsourcing. Within a DTN, messages are exchanged between nodes following a store-carry-forward paradigm, which is notably susceptible to congestion and can lead to a crippling in network performance. A DTN's time-dynamic topology departs from traditional network definitions in its unpredictable and volatile nature, thus prohibiting the effective adoption of traditional network solutions to this problem. In this paper, the Catora system is proposed as a multi-copy message exchange and buffer management system designed to both aid in the delivery of prioritized messages and mitigate congestion and its degradation. Operating around the distinct ordering of messages for transfer, delivery, and deletion, Catora propagates messages so as to balance their dissemination, hasten the delivery of high priority messages, and avoid congestion through strategic buffer management. Simulations using two real-world datasets demonstrate Catora's capability to quickly deliver more messages at reduced overhead costs when compared to benchmarks and the state-of-the-art, even when the network suffers from congestion.

1. INTRODUCTION

As of the third quarter of 2018, at least 4.74 billion smartphones were in use globally, with annual forecasts for 2019 to 2023 projecting an estimated 1.39 to 1.54 additional billion smartphones will be sold each year [8, 27]. With these wireless mobile devices becoming ubiquitous throughout the world, now is an opportunity to investigate how to piggyback from their pervasiveness to form ad-hoc networks for applications such as disaster relief and coordination [25, 32], battlefield coordination [4], and safe and efficient transportation [10, 20]. The challenge of effectively employing these devices results from the volatility and sparsity of interconnectivity, primarily due to mobility, isolation, noisy wireless communication, and limited resources. Thus, message delivery in such a network must adopt a store-carry-forward approach, where devices store messages in their buffers, carry them as they move, and forward them to opportunistically encountered devices. This network is called a Delay Tolerant Network, and nodes therein may have no knowledge as to when encounters will occur [14]. When encounters do occur, the unpredictable duration of the connections, variable channel capacities, and finite buffer spaces require devices to adopt sophisticated strategies in order to choose which messages are to be exchanged and cached.

It is assumed that messages are annotated with metadata to describe their payloads, and that nodes introduce messages into a DTN for delivery that otherwise cannot be delivered through more stable networks. For instance, a node with a connection to external networks may retrieve messages from them to serve the requests from other nodes in the DTN [32], or a node may retrieve requests from a crowdsourcing platform to disseminate to nodes who can respond to these requests [6]. Nodes may also create messages by observing real world objects or events, such as photos of damaged buildings, or video recordings of important or dangerous events.

Previous algorithms for message delivery in DTNs have been proposed with a focus on probabilistic node encounters [18], bounded resource consumption [26, 31], and exploiting graph-theoretic and social properties of the network [11, 17]. However, these approaches take on a node-centric approach to message routing, ignoring the contents of a message and its priority and importance to their intended destination. Another challenge is the degradation of service due to congestion. While traditional networks have many strategies to detect and mitigate congestion, these cannot be readily adopted primarily due to the fleeting and unpredictable nature of a DTN's topology [9, 29]. DTNs are notably prone to congestion; with messages being stored for longer periods of time, node buffers are liable to overflow should there be too many incoming messages relative to those outgoing. When congestion does occur, messages must either be rejected or dropped, both of which may impact successful deliveries and latencies if not done so strategically.

As such, in this paper, we propose the Congestion Avoidance through Transmission Ordering and Resource Awareness (Catora) system as a multi-copy message exchange protocol and buffer management system that is both message centric and congestion aware. Three utility metrics are proposed as a means of ordering messages into three virtual sorted queues for the tasks of message relaying, buffer management, and message delivery. In the transfer virtual queue, messages are organized so that their dissemination tree expands in a bredth-first manner, thus ensuring wide and balanced dissemination to increase the chances of final delivery without uncontrolled resource consumption. In the event of network congestion, the buffer management virtual queue drops messages that exhibit both low utility-to-size ratios and high dissemination status as a means of freeing up buffer space without dropping messages unlikely to be delivered. When a message destination is encountered, messages destined for the node are pulled from the delivery virtual queue such that those with high utility densities are delivered early through the unpredictable connection, thus insuring faster delivery of higher priority messages. Information on successful deliveries is then opportunistically disseminated throughout the network to allow nodes to remove delivered messages from their buffers, which in turn aids in the system's congestion avoidance.

This paper's contributions are summarized as follows:

- Preliminary simulations are conducted on three DTNs, one synthetic and two induced from real-world datasets, to analyze the effects of congestion on the performance of five well-known DTN routers. These simulations identify key network metrics that can be used to diagnose congestion, revealing that an increase in message drops and a decrease in buffer occupancy times of messages are signs of higher congestive conditions. Regardless of which router is employed, increasing the load on a DTN results in a drop in the quality of service in terms of fewer successful deliveries and higher latencies.

- The Catora system is proposed to mitigate the effects of congestion and more appropriately consume resources for prioritized message delivery in a DTN. Three new utility metrics are proposed to construct three virtual message queues for the tasks of message relaying, buffer management, and message delivery, promoting balanced message dissemination, strategic congestion mitigation, and fast delivery of high-priority messages. In addition, Catora tracks and disseminates information opportunistically to allow for a delivered message's immediate removal from node buffers, thus reducing the likelihood of congestion by eliminating unneeded resource consumption.

- Through simulations driven by two real-world datasets, Catora is evaluated against three benchmark systems and two state-of-the-art congestion controlling systems under varying levels of congestion. Results demonstrate that Catora outperforms both congestion-crippled Epidemic and the state-of-the-art in terms of more successful deliveries, lower latencies, and lower resource consumption. Additionally, analysis of the symptoms of congestion indicate it is more capable at preventing and controlling congestion than other systems susceptible to it.

The subsequent parts of this paper are organized as follows. Section 2 summerizes related work into DTN congestion and summarizes the contributions of both pivotal and recent research in the field. Section 3 discusses the DTN architecture, and details the constituents of and challenges to the problem of message delivery in a DTN. Section 4 conducts a analysis on the causes and effects of congestion through preliminary simulations. Section 5 proposes the Catora system, first by formally defining the metrics employed by the virtual sorted message queues followed by the algorithms employing them to drive message routing and buffer management. Section 6 evaluates Catora in comparison to benchmarks and two state-of-the-art DTN congestion control systems using two real-world datasets. Finally, Section 7 concludes the paper.

2. RELATED WORK

In this section, both pivotal and recent related work is reviewed within the realm of DTN congestion. These works have focused on tasks of modeling congestion as well as proposing systems to control and avoid congestion using strategies such as message replication control, ensuring fair resource consumption, and strategic buffer management.

2.1. MODELING CONGESTION

To better understand the factors that influence congestion in DTNs and the consequences faced when congestion is experienced, diverse modeling techniques have been proposed in recent literature. Birrane [1] adopted a contact-graph based approach to model congestion, focusing on predicting the capacity of transmission channels as a means to diverting traffic along paths with sufficient capacity. While the network operates, nodes are tasked to opportunistically learn the contact graph from information in the headers of messages passing through them, inform downstream nodes of new knowledge about the contact graph known locally to the node, and provide feedback to upstream nodes to improve on path selection.

Another goal of congestion modeling is to derive formulas for certain metrics useful in DTN performance. Silva et al [30] adopted percolation theory to model congestion, whereby nodes that have sufficient buffer space are used to detect journeys (time-ordered paths) through the network. Their model allows for the derivation of the probability that a journey exists, which in turn defines the probability that the message can be delivered. In doing so, their formulation permits the derivation of the average delivery latency, the average buffer occupancy, and the average duration of time that nodes were congested. Sandulescu et al [26] propose an online, holistic framework for determining lower and upper bounds on the achievable throughput – i.e. the number of bits that can be delivered over a defined time period – between any two nodes exhibiting random mobility. Their formulations depend on the mobility of the nodes, the adopted protocol for message dissemination, and the distribution of resources available to each node. Through analytical and simulation-driven analysis, it is observed that resource exhaustion and a degradation in message delivery is induced by an increasing amount of data being sent through the network.

2.2. CONGESTION AVOIDANCE AND CONTROL

Apart from modeling congestion, some proposals have investigated strategies to avoid and control congestion. This has been motivated by observed shortcomings of prior work. Early works implicitly rely on nodes with high network centralities to relay messages, as these nodes are capable of contacting many other nodes who are either message destinations or are able to reach them [5, 11, 18]. These strategies, however, are naive to their increased resource consumption and the possibility of introducing bottlenecks and congestion. Intuitively, these nodes will use more of their resources in relaying messages than those with lower centralities. When the rate of messages flowing through the network increases beyond their capacity, congestion occurs with overflowing buffers and saturated connections.

To respond to this, some strategies have controlled message replication in a DTN. Each message is permitted to be copied some number of times with each replication being recorded in the headers of both the replicated message and the replica. A well known strategy is Spray-and-Wait [31], which permits messages to be *sprayed* to other nodes until the header value reaches 1. After this, the message must *wait* until a direct encounter with its destination occurs. Jain and Chawla [13] propose a similar system to differentiate types of traffic flowing through a DTN, with contributions to the strategic ordering of messages for transmission.

Another approach for congestion control emphasizes fair resource consumption. One such work is CAFé [9], which proposes two metrics on a DTN's nodes, receptiveness and retentiveness, to measure the degree to which a node has been utilized for forwarding in terms of their consumed channel capacity and their buffer occupancy, respectively. Using these metrics, a forwarding heuristic is computed on a message being considered by a potential relay and used to route traffic away from paths that are at risk of congestion, favoring those with ample buffer space and lower relaying delays.

2.3. BUFFER MANAGEMENT

Another strategy for controlling congestion is to use sophisticated methods for managing congested buffers. Systems emphasizing this focus are tasked with choosing which messages should be dropped in order to make room for other incoming messages. Davis et al [7] pioneered the field with the proposal of four message dropping policies to be carried out when node buffers become overloaded: drop-random, drop-least-recently received, drop oldest, and drop-least-encountered. The drop-least-encountered policy sorts messages based on a node's encounters with the messages' destinations; destinations that are encountered frequently will have their message higher in order. Given this ordering of messages, a congested node drops those messages with the lowest order, representing destinations that the node is least likely to encounter, until enough space is available to accommodate new messages.

Many strategies that manage buffers focus on averting overflows from flooding-based routing schemes. An alternative approach is to focus on discriminatory routing schemes, which under certain circumstances are favorable due to their reduced resource consumption via fewer transmissions and lower buffer occupancy. Lindgren and Phanse [19] consider this approach and propose dropping policies to complement the PRoPHET routing protocol [18]. Particularly, five dropping policies are defined: First-In-First-Out (FIFO) drops messages from a buffer if they arrived earlier than others within the buffer, motivated by the observation that a message occupying a buffer for a long duration will likely have more opportunities to be forwarded by others than those freshly arriving; Most-Forwarded-First (MOFO) drops messages that have been forwarded more than others based on the observation that these messages have a higher probability of delivery because of their higher replication status, enabling messages with fewer replicas more time to infiltrate deeper into the network; Most-Favorable-First (MOPR) maintains a metric for each message called its FP value that is increased at each forward by the message's delivery probability, and drops messages that have the highest value for the metric to enable messages with lower FP values a change to reach nodes with higher delivery probabilities to their destinations; Short-Lifetime-First (SHLI) drops messages that are the closest to expire on the observation that if the destination has not been encountered earlier, it is increasingly unlikely to be encountered before the message expires; finally, the Least-Probable-First (LEPR) policy drops messages that have the lowest delivery probabilities, favoring messages that are more likely to reach their destinations through the carrying node.

Beyond simple-metric-based buffer policies, Iranmanesh [12] propose a joint buffer management and routing protocol similar to Spray-and-Wait [31] called QM-EBRP, intended to avert buffer overflow congestion. In QM-EBRP, a multi-objective utility function is proposed that considers a message's upper limit on the number of permitted replicas, remaining lifetime, and the node's encounter history in estimating the probability of delivery and expected delivery delay. Then, the number of replicas that a node allocates to a potential relay, along with a more restricted time to live, is varied to observe the influence on the estimated metrics. Messages exhibiting a higher rate of change to these values are prioritized higher for replication, while those exhibiting a lower rate of change are dropped when congestion is imminent. The Fairness-Aware Message Forwarding (FAMF) system, proposed by Roy et al [25], is designed to avoid an asymmetric favoring of certain destinations to others when delivering messages in a congested DTN, that results in some destinations receiving more messages them than others. To this end, a metric is computed by considering a node's contact probability with a message's destination and the probability of the message being dropped before delivery. This metric is employed in both deciding which message to forward to which neighbor as well as choosing which messages to drop from a congested buffer. It should be noted, though, that these systems rely on the predictability of the occurrence and duration of an encounter with between nodes, both of which may not be readily available or reliably estimated in an opportunistic network due to the uncertainties of node mobility.

Apart from dropping messages, the strategy proposed by Wu et al [35] adopts a collaborative caching technique to free up space from a full buffer for new messages without unnecessarily dropping messages and eliminating their potential delivery. To this end, the ICMT system tracks and opportunistically disseminates the contact history of all nodes throughout the network so as to formulate the probability of contact between any two pairs and assign priorities to messages based on the contact probability with its destination.

When a buffer lacks the capacity to store new messages, the node will attempt to broadcast and then drop its lower-priority messages to its neighbors to make space. When there are no neighbors to broadcast to, only then will messages be dropped without relocation.

3. FORMAL DEFINITIONS AND ARCHITECTURE

Delay Tolerant Networks depart from classical network architectures due to the network's unpredictable, time-dynamic topology, which disconnects the network's nodes from each other and complicates predictions of when and between whom connections will arise. The remainder of this section is dedicated to formalizing this network and the requirements for message delivery.

3.1. DELAY TOLERANT NETWORK ARCHITECTURE

The primary constituents of a DTN are nodes capable of introducing messages into the network, storing them for extended periods of time, and propagating copies to other nodes when conditions permit. The necessary conditions for propagation are (1) the node currently holding the message (a *message carrier*, or simply *carrier*) has an open communication channel with some other nodes (the carrier's *encountered neighbors*, or simply *neighbors*); (2) at least one neighbor is an *interested recipient* by either expressing interest in receiving the message or by being (one of) the message's *destination(s)*; (3) the channel has sufficient capacity (i.e. it is open long enough and has a high enough transmission speed in bits per second) through which to transmit the message successfully; and (4) the interested recipient has enough buffer space to store the message.

Definition 9. Delay Tolerant Network*:* *Let $G = (V, E)$ be a graph, where V is the set of nodes in the network and $E \subseteq V^2$ is the set of edges between nodes, where an edge between nodes u and v is denoted by (u, v). A Delay Tolerant Network (DTN) underlined by G is defined by $\mathcal{G} = (V, E, \mathcal{L}, \mathcal{S}, \mathcal{B})$, where the network lifetime $\mathcal{L} = [t_\varnothing, t_\Omega)$ specifies the time* *span in which the network is operating as defined by a starting time* $t_\varnothing$ *and an ending time* t_Ω*; the transmission speed* $\mathcal{S} : E \times \mathcal{L} \to \mathbb{R}_{\geq 0}$ *defines the number of bits per second a node can exchange with a neighbor; the buffer size* $\mathcal{B} : V \to \mathbb{N}_{\geq 0}$ *specifies the bit-size of a node's message buffer.*

Nodes are classified by three non-exclusive types: message sources, message carriers, and message destinations. *Message sources* introduce messages into the network for delivery to their destinations, either by introducing the messages from external networks or by creating new messages through observing objects and events in the real world. In this work, it is assumed that messages introduced into the network are truthful and do not intentionally introduce misinformation. *Message carriers* are the nodes that receive messages from other nodes, either other carriers or message sources. A message carrier is motivated to receive and store other nodes' messages in its buffer, and to expend energy in transmitting those messages to nodes encountered later on, so that it will have its own messages received, stored, and delivered by other nodes. *Message destinations* are nodes interested in the payload of messages on various topics, or they may be nodes with a connection to other networks acting as gateways between these networks and the DTN.

3.2. MESSAGES

An annotated message, herein referred to as a *message*, is denoted by the tuple $m = (\pi, A, H)$, where the *payload* π is the main content of the message (e.g. a photograph, a report), the *metadata* A forms a set of attributes that annotate π, and the *header* H tracks network-specific properties related to the message. A message is considered to have a *utility*, as defined by Defintion 10.

Definition 10. Message utility*: The utility of a message $m = (\pi, A, H)$ is denoted by $u(m)$, and is calculated as the sum of the precisions of the attributes in A:*

$$u(m) = \sum_{a \in A} \rho(a) \tag{1}$$

where $\rho(a)$ is the precision *of attribute $a \in A$ such that* $0 \leq \rho(a) \leq 1$.

Since messages may be created by nodes in response to observations, the annotations that a node makes for a message may have varying precision. Multiple messages may be created covering the same observation by different nodes, each having a unique vantage point and observing details differently. Definition 10 accomodates this through attribute precision, where $\rho(a) = 0$ is considered the least precise and $\rho(a) = 1$ the most precise.

The header of a message, denoted H, tracks various network-specific properties of the message. In this paper, it is assumed to track at least two properties: the path this message has traveled from the source, and the number of known copies of the message that exist in the network. These properties are updated as follows.

3.2.1. Message Path.

Definition 11. Message Path*: For a message $m = (\pi, A, H)$, the path of m is denoted as $m.\mathtt{path}$ and is defined as a sequence of nodes $(n_0, n_1, \ldots, u)$, where n_0 is the message's creator, n_i received a copy from n_{i-1} for $i > 0$, and u is the node carrying the specific copy of m under inspection. The length of a message's path is denoted as $|m|_{path}$.*

Upon creation by a node n_0, a message's path is simply defined as a singleton sequence (n_0). When a copy of the message is relayed from some node u to some other node v, the path of the relayed message is appended with v, while the copy residing with u is unchanged.

3.2.2. Message Copy Count.

Definition 12. Message Copy Count*: For a message $m = (\pi, A, H)$, the copy count of m is defined in the message's header H as a positive integer denoted as $|m|_{copies}$, where $|m|_{copies} > 0$.*

Upon creation by a node, a message's copy count is equal to 1. The process of updating this value is as follows. Let m_u reside in node u's buffer, and let m_v be a copy of m_u that is transferred from u to v. Upon completion of this transfer, the copy counts for both messages $|m_u|_{\texttt{copies}}$ and $|m_u|_{\texttt{copies}}$ are incremented and equal.

For example, assume a message m is created by n_0 at time t_0. At time $t_1 > t_0$, a copy m' is made from m and transferred to node u. Later, at time $t_2 > t_1$, a copy m'' is made from m and transferred to node v. At time t_0, $|m|_{\texttt{copies}} = 1$; at time t_1, $|m|_{\texttt{copies}} = \left|m'\right|_{\texttt{copies}} = 2$; finally, at time t_2, $|m|_{\texttt{copies}} = \left|m'\right|_{\texttt{copies}} = 3$ and $\left|m''\right|_{\texttt{copies}} = 2$. The inconsistency at t_2 is due to u not being notified that another copy of m_{n_0} being made for v. Further, if no other copies are made from m and m', but copies are made from m'', the values of $|m|_{\texttt{copies}}$ and $\left|m'\right|_{\texttt{copies}}$ would remain equal to 3 while $\left|m''\right|_{\texttt{copies}}$ would continue to increment in accordance to Definition 12.

3.3. CHALLENGES OF MESSAGE DELIVERY IN DTNS

Successful delivery of messages to their destination(s) is challenging due to the sparse and time-varying topology of a DTN, resulting in the absence of contemporaneous paths between a message carrier and the messages' destinations at the time a carrier receives a message [2, 14]. Instead, message delivery relies on time-ordered paths through which nodes propagate a message to encountered neighbors, one hop at a time, until a message reaches its destination. In between node encounters, messages are cached in each carrier's buffer.

Within some time interval in $\mathcal{L}$, there may exist multiple time-ordered paths between a pair of nodes through which a message may travel. The existence of a time-ordered path is a necessary, but not sufficient, condition for message delivery. For successful delivery to occur, each node along a time-ordered path must successfully receive messages from its predecessor, store them without dropping, and propagate them others. Acceptance of a message requires resources that permit message delivery, such as sufficient channel capacity between encountered nodes and sufficient buffer space to cache messages. These are competitive resources, as other messages will also be consuming them while in transit and within buffers.

If a node knows when encounters will occur between pairs of nodes, then a time-ordered path can be pre-determined. This may not always be achievable, as is assumed in this paper. Nodes move autonomously, and the inaccessibility of knowledge oracles [14], from which information is obtainable on the network's topology or the nodes' locations, complicates reliable predictions. Thus, message delivery is done in a best-effort, store-carry-forward manner: when two nodes encounter one another, their task is to decide if they want to receive a subset of their neighbors' messages, and if so, which subset. Should a node's buffer be too full to receive a message, the node must also decide if it will reject the incoming message or drop some of its carried messages to make room.

4. CONTRIBUTING FACTORS TO CONGESTION

Congestion within a traditional network is characterised by a node having an arrival rate of messages that exceeds its departure rate, whereby messages begin to pool up at the node [29]. Should this continue, the queue that temporarily stores the backlogged messages will overflow, requiring some messages to be dropped in order to continue receiving new ones. When a message is dropped, it potentially may no longer exist in the network, thus effecting the network's delivery metrics. This characterization of congestion may also occur within DTNs, but instead of a seemingly uninterrupted flow of messages passing through a routing path (that which is perceived in traditional networks), the messages in a DTN progressively move forward in the lock-gate manner of the store-carry-forward paradigm [2, 14]. As a result, the arrival rate and departure rate of messages is bursty in nature; messages only arrive to, and depart from, a node when it opportunistically connects with another neighbor.

One important distinction for DTN strategies is whether a message departure results in more space available in a node's buffer. Those that adopt single-copy routing free up space when a node transfers a message because it deletes its copy after it has successfully transferred. Alternatively, for multi-copy routing strategies, this does not necessarily occur. Rather, a node may retain a copy of a message that it hands off. This intuitively results in more messages residing within the buffer of a node. When it becomes full, it must then decide whether to reject new messages, or accept them at the sacrifice of dropping some of those it currently carries [15].

Ultimately, when congestion occurs, the observable symptoms include fewer successful deliveries, increased message drops, and higher delivery latencies [24, 34]. Various factors have been shown to contribute to the occurrence of congestion. In [34], Wang et al found that the rate at which messages enter the network has a dramatic influence on the ultimate probability of successful deliveries; too many messages degrades network deliveries as it induces congestion rapidly, preventing deliverable messages from traversing far into the network before being quickly dropped to make room for newer messages. Xia et al [36] found message creation rate also affects delivery delay; when more messages are created than the network can handle, longer latencies in delivery occur. Silva et al [30] found that increasing the space available within buffers results in shorter durations in which nodes are congested. Additionally, larger buffers also result in messages staying within them for longer on average, as the availability of more space alleviates the need to drop messages quickly.

Below, we observe the symptoms of congestion when the frequency of message creation varies from one every second to one every 100 seconds. To measure the quality-of-service of the network, the observed metrics are the number of delivered messages and the average latency for delivery. To better understand congestion, the average buffer occupancy time – i.e. the average amount of time each message resided within a buffer before it is dropped – and the number of dropped messages is measured.

4.1. SIMULATION SETUP

The simulations in this section employ three datasets – one synthetic, and two real-world – that are used to simulate a DTN, and are conducted using the Opportunistic Network Environment (ONE) simulator [16]. For the synthetic dataset, 44 nodes operate within a 4.5km by 3.4km area centered on Helsinki, Finland. Their movement is dictated by the Random Waypoint mobility model taking the shortest path between two points following available roadways. The set of nodes is partitioned into two groups: the first represents 40 pedestrians moving at walking speed; the second represents 4 vehicles moving between 10 to 50 km/h. Pedestrian nodes are able to wirelessly communicate with all other nodes when their proximity is within 10 meters at a speed of 250 kbps, simulating the connectivity of Bluetooth, whereas vehicular nodes employ two wireless technologies, one for pedestrian communication (Bluetooth) and the other for vehicle-to-vehicle (V2V) communication within 1 km at 10 Mbps. Pedestrians have 100 MB of buffer space to dedicate to message storage; vehicles have 2 GB.

The two real-world datasets are the SIGCOMM 2009 contact trace dataset [21, 22, 23] and the INFOCOM 2006 contact trace dataset [3, 28]. For the SIGCOMM 2009 dataset, the Bluetooth encounters between smartphones carried by 76 attendees of the SIGCOMM 2009 conference were recorded over the course of five days. Similarly for the INFOCOM 2006 dataset, 98 iMote devices were dispersed at the INFOCOM 2006 conference: 20

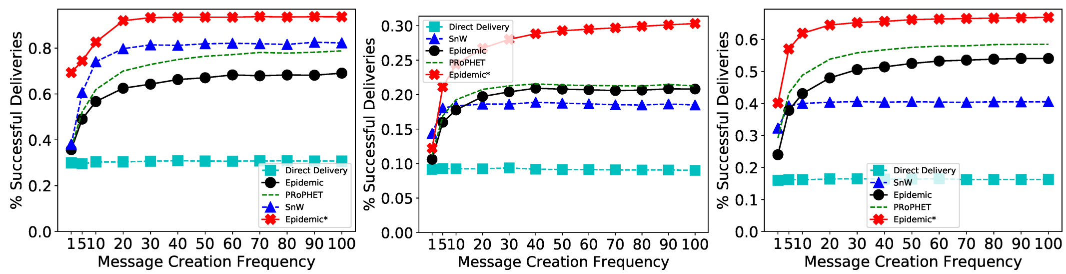
were statically located at various points of interest, and the other 78 were distributed to conference attendees. Bluetooth encounters between the devices were recorded over four days.

In the three simulated networks, each encounter offers the participating nodes the opportunity to exchange messages throughout its duration. The bandwidth of connections is set to a constant bitrate of 250 kbps. Each node is configured to have 100 MB of storage to allocate to messages, each being 100 KB in size and having a lifespan of 5 hours. Messages are randomly generated throughout the timespan of the simulation, with the source and destination being uniformly selected from the participating nodes.

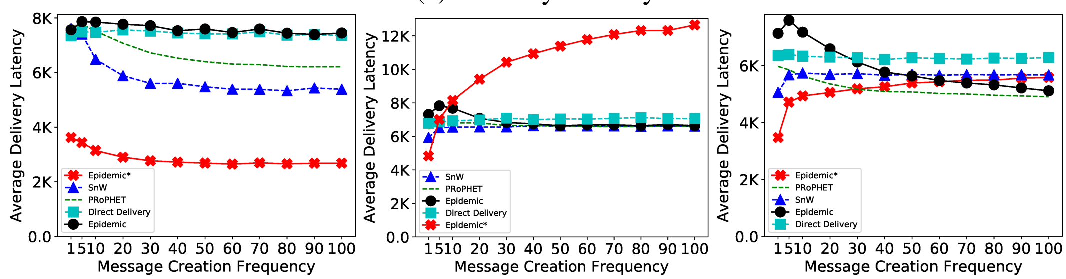
Five well-known DTN routers were adopted by all nodes to perform message routing: Spray and Wait [31], Direct Delivery, and two versions of Epidemic [33], PRoPHET [18]. The two Epidemic routers are differentiated by whether they adhere to the constraints of a DTN: one version complies by recognizing that connections have finite bandwidth, nodes have finite buffer space, and knowledge of a message's delivery is only known to nodes opportunistically; the other, labeled Epidemic*, enables all nodes to instantly drop a message once it has been successfully delivered and dismisses the restrictions of message sizes, permitting all messages carried by a node to instantly be exchanged during an encounter. This is motivated so as to observe if the performance of the former Epidemic is greatly impacted by congestion compared to other routers, and to witness the best possible performance of a DTN using Epidemic*.

4.2. QUALITY OF SERVICE DEGRADATION

To better understand the effect of congestion on a range of networks, we investigate the delivery ratio and delivery latency of messages in the three simulated networks. Figure 1 presents these results, with Figure 1a displaying the percentage of successfully delivered messages and Figure 1b displaying the average latency in delivery of messages. The most striking effect of congestion is the performance of Epidemic relative to the other routers.



(a) Delivery Latency

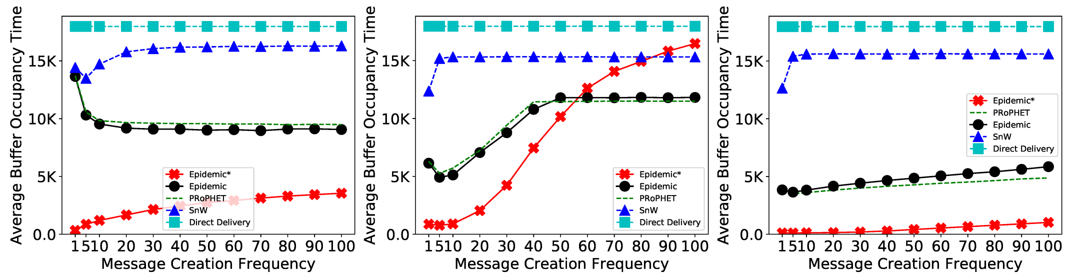


(b) Delivery Ratio

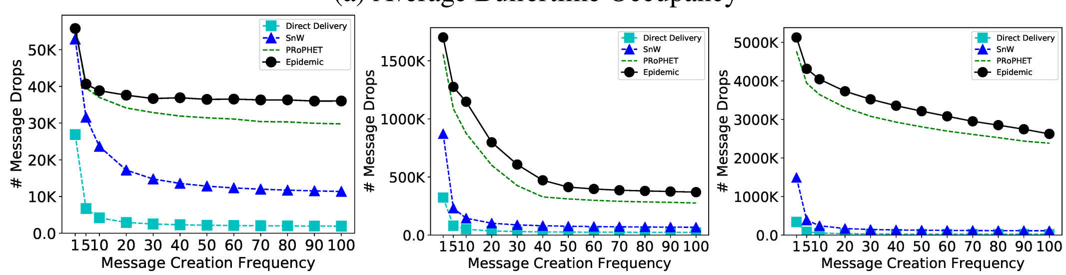
Figure 1. Quality of Service for Synthetic (left), SIGCOMM 2009 (middle), and INFOCOM 2006 (right)

In many past investigations on DTNs, Epidemic has typically been shown to perform the best in terms of achieving the highest delivery ratio while also obtaining the lowest delivery latencies. While Epidemic*, the version that ignores DTN constraints, exhibits these properties, the more DTN-compliant Epidemic is observed to perform worse than most of the other strategies. In the synthetic network, PRoPHET and Spray and Wait are able to deliver a higher percentage of messages than Epidemic, with only Direct Delivery having lower delivery ratio. So too does this affect the delivery latencies: PRoPHET and Spray and Wait both have lower latencies than Epidemic in most cases. Similar observations are found in the networks formed by the SIGCOMM 2009 and INFOCOM 2006 datasets, although Spray and Wait's performance dips below Epidemic for higher message creation intervals.

Another observation is the rapid decline in delivery ratios along with a spike in latencies as the message creation interval approaches 1. At lower message creation intervals, messages are created in higher frequencies, resulting in more messages passing through the network and residing within node buffers. Inflection points in these graphs suggest the point



(a) Average Buffertime Occupancy



(b) Dropped Message Count

Figure 2. Congestion Symptoms for Synthetic (left), SIGCOMM 2009 (middle), and INFOCOM 2006 (right)

at which congestion begins to degrade performance. In the synthetic network, this inflection point occurs at the 20-second interval; for the real-world networks, at the 10-second interval. In the next section, the potential causes for this degradation are analyzed and identified.

4.3. ANALYSIS OF CONGESTION

Intuitively, if messages reside within buffers for extended periods of time, the risk of a buffer becoming full increases with the rate of message creation. Longer-duration buffer occupancies are attributable to a number of factors, such as longer message lifetimes, larger buffer sizes, and smaller message sizes [30]. The trigger that brings an end to a message's occupancy is the dropping of the message from a node's buffer, brought about from a message being successfully delivered (and subsequently no longer needed) by the node, the message expiring, or an overflow in the buffer. When a buffer overflows, the only recourse is to either reject incoming messages or drop enough carried messages from the buffer to make space. Thus, when a message is dropped, the amount of time it resided in the buffer from which it was dropped is defined. In Figure 2, the symptoms of buffer-overflow congestion is investigated against the varying of message creation intervals and changes in two measurements are observed: the average time in which messages resided in node buffers, and the number of message drops that occurred over the simulation's timespan.

As depicted in Figure 2a, Direct Delivery and Spray and Wait exhibit the longest buffer occupancies in most cases, while PRoPHET and both versions of Epidemic show the lowest. With Epidemic*, it is rather straight forward: when a message is delivered, all nodes simultaneously drop that message from their buffer, knowing to do so through immediately accessible global knowledge. The difference between Direct Delivery / Spray and Wait and Epidemic / PRoPHET is due to their restrictive versus generous replication of messages, respectively. Direct Delivery does not replicate messages at all; rather, a message creator will only transmit a message to its destination directly. Spray and Wait permits replication, but restricts the number of times it is permitted, after which only single-copy forwarding may be carried out. This bounds the number of copies of a message may reside in the network, thus restricting the number of buffers the message occupies at a given time. Epidemic and PRoPHET do not bound the number of message copies, thus leading to more buffers being occupied.

5. THE CATORA SYSTEM

In this section, the constituents of the Catora system are formally defined. Catora's operation revolves around the ordering of messages based on an intended action, such as delivering a message to its final destination, relaying a message to a new carrier, or dropping a message from a congested buffer. Three metrics are proposed for these various tasks so as to impose an order on a collection of messages in the form of virtual sorted message queues: a delivery utility, a transfer utility, and a buffer utility. The delivery utility (Section 5.1) orders a collection of messages that are immediately deliverable into a delivery virtual queue

Table 1. Catora Symbology

Symbol	Summary
u, v	Nodes encountering one another in a DTN
m	A message
$\lvert m\rvert_{\texttt{bytes}}$	The size of message m in bytes
$\lvert m\rvert_{\texttt{path}}$	The length of m's path (Def. 11)
$\lvert m\rvert_{\texttt{copies}}$	The known copy count of m (Def. 12)
$\texttt{dest}(m)$	The destination node of m
B_u	The message storage buffer of node u
$\lvert B_u\rvert_{\texttt{free}}$	The number of bytes freely available in u's buffer
$u(m)$	The utility of message m (Def. 10)
$u_d(m)$	The delivery utility of message m (Def. 13)
$u_t(m)$	The transfer utility of message m (Def. 14)
$u_b(m)$	The buffer utility of message m (Def. 15)
$M_{u,v}$	u's delivery virtual queue for messages to v (Def. 13)
Q_u	u's transfer virtual queue (Def. 14)
D_u^*	u's buffer management virtual queue (Def. 15)

so as to ensure higher utility-dense messages are delivered before the connection drops. The transfer utility (Section 5.2) orders messages into a relay virtual queue such that those with a higher transfer utility are transmitted to neighbors earlier than those with lower. Finally, the buffer utility (Section 5.3) orders messages into a buffer management virtual queue such that those with low utilities and high replication will be selected first for removal when congestion occurs. The algorithms followed by the Catora system are defined in Section 5.4 and describe the procedures that are followed when nodes connect, disconnect, complete a message transfer, and manage an overflowing buffer. Table 1 provides a summary of this paper's adopted symbology.

5.1. MESSAGE DELIVERY ORDERING

Assume a connection is established between a message carrier u and another node v, and u is carrying messages destined for v. With the connection's duration being unpredictable, the Catora system will immediately attempt to deliver messages destined for v in the descending order imposed by the delivery utility.

Definition 13. Delivery Utility*: The delivery utility of a message* $m = (\pi, A, H)$ *is denoted by* $u_d(m)$*, and is calculated as the quotient of m's utility and m's size:*

$$u_d(m) = u(m) / |m|_{bytes} \tag{2}$$

where $|m|_{bytes}$ *is the storage size of m in bytes. The delivery virtual queue between u and v, denoted as* $M_{u,v}$*, is in descending order defined by Equation 2.*

Since the connection's duration is unknown, this ordering attempts to deliver messages with high utility density relative to size before the connection drops. Per Definition 10, assuming two messages have the same message utility, the smaller message would be transferred before the larger one. This is done so as to hasten the delivery of as many messages as possible without potentially wasting a connection on the transfer of a large message that is aborted due to premature connection drop.

5.2. MESSAGE TRANSFER ORDERING

Assume a connection is established between a message carrier u and another node v, and u is carrying messages that are not currently carried by v. Because the duration of the connection is unknown, the Catora system will add these messages to a transfer queue in descending order of their transfer utility.

Definition 14. Transfer Utility*: The transfer utility of a message $m = (\pi, A, H)$ is denoted by $u_t(m)$, and is calculated as the quotient of m's utility and the product of the m's size and m's current path length:*

$$u_t(m) = u(m) / \left(|m|_{bytes} \cdot |m|_{path}\right) \tag{3}$$

where $|m|_{path}$ is the message path as defined by Definition 11. The transfer virtual queue of u, denoted as Q_u, is in descending order defined by Equation 3.

Similar to the delivery utility in Definition 13, the transfer utility favors messages with high utility densities relative to their size. However, it also considers the dispersion of the message in the network by incorporating the message's current path length $|m|_{\mathtt{path}}$, and strives to spread messages in a bredth-first manner. The intuition behind Equation 3 is that messages that have not traveled far from their source are less likely to reach their destination than messages that have, seeing as far traveling messages reside with more nodes, any of whom may come into contact with the destination in the near future.

5.3. STORAGE BUFFER MANAGEMENT

When a node's storage buffer becomes too full to receive new messages, messages within the node's buffer must be dropped so as to continue receiving new ones. The manner in which messages are selected for dropping is decided by the buffer utility, whereupon a node orders its carried messages and drops enough of the lowest-utility messages to make room for an incoming message.

Definition 15. Buffer Utility*: The buffer utility of a message $m = (\pi, A, H)$ is denoted by $u_b(m)$, and is calculated as the quotient of m's utility and the product of the m's size and m's known copy count:*

$$u_b(m) = u(m) / \left(|m|_{bytes} \cdot |m|_{copies}\right) \tag{4}$$

where $|m|_{\texttt{copies}}$ is m's copy count as defined by Definition 12, representing the number of copies that are known by u to exist in the network. The buffer management virtual queue of u, denoted as D_u^*, is in ascending order defined by Equation 4.

The buffer utility of a message is similar to the transfer utility defined in Definition 15 in that it favors messages with higher utility densities relative to their size in conjunction with lower dispersion in the network. Those messages that both have lower utility densities and have been replicated more across the network are likely be to dropped. However, instead of measuring the distance the message has traveled from its creator, it uses the known number of copies of the message that exist in the network.

Algorithm 7 A channel is open between node u and node v at time $t_{\in}\mathcal{L}$.

procedure CONNECT(u, v)
▹ Opportunistic learning of delivered messages
  Let $D_v \leftarrow$ delivered messages known to v
  $D_u \leftarrow D_u \cup D_v$
  $B_u \leftarrow B_u \setminus D_v$
  $Q_u \leftarrow \{\langle m, v'\rangle | \langle m, v'\rangle \in Q_u, m \notin D_v\}$
▹ Transfer deliverable messages
  Let $M_{u,v} \leftarrow \{m | m \in B_u, \texttt{dest}(m) = v\}$
  while $M_{u,v} \neq \emptyset$ **do**
    Let $m^* \leftarrow \underset{m \in M_{u,v}}{\operatorname{argmax}} \{u_d(m)\}$
    Send $m^* \Longrightarrow v$ ▹ Send highest delivery utility message
    $M_{u,v} \leftarrow M_{u,v} \setminus \{m^*\}$
    $B_u \leftarrow B_u \setminus \{m^*\}$
    $D_u \leftarrow D_u \cup \{m^*\}$
    $D_v \leftarrow D_v \cup \{m^*\}$
  end while
▹ Enqueue future transfers
  Let $Q_{u \to v} \leftarrow \{\langle m, v\rangle | m \in B_u, \texttt{dest}(m) \neq v, v.\texttt{wants}(m)\}$
  $Q_u \leftarrow Q_u \cup Q_{u \to v}$
  if $Q_u \neq \emptyset$ **then** ▹ Send message with highest transfer utility
    Let $\langle m^*, v^*\rangle \leftarrow \underset{\langle m, v'\rangle \in Q_u}{\operatorname{argmax}} \{u_t(m)\}$
    Send $m^* \Longrightarrow v^*$
  end if
end procedure

5.4. THE CATORA PROTOCOL

At a high level, the Catora protocol is conceptually composed of four subsystems that control the node's behavior when a connection is established, a message has been transferred, a node's buffer has insufficient space to receive a message, and when a connection drops. Each of these subsystems is defined in detail below.

5.4.1. Behavior At Connection Establishment. Assume nodes u and v connect at some time t. At a high level, these nodes carry out the following tasks:

1. Both v and u notify each other of the messages known to have been delivered and immediately drop those messages that are carried

2. u immediately begins transferring any messages it carries that are destined for v according to the order of the delivery virtual queue (Definition 13)

3. u iterates over its carried messages and queues up a future transfer for each message that v expresses interest in relaying in order of the relay virtual queue (Definition 14)

Algorithm 7 formally defines the process between two newly-connected nodes u and v, and is executed by both nodes. First, v informs u of the message IDs it knows have been successfully delivered (line 2), either by itself or learned from others, and incorporates that knowledge into its own (line 3). If u carries any of these messages, it removes them from its buffer (line 4) and cancels any associated queued transfer (line 5). Following this, u then begins transferring any messages it carries that are destined for v for final delivery (lines 6–14). The order in which messages are sent is dictated by the delivery virtual queue of each message, as defined in Definition 13, where the message with the highest delivery utility is chosen first. Once the message has been successfully delivered, the message is removed from u's delivery queue (line 10) and buffer (line 11), and marked as delivered by both nodes (lines 12 and 13). This continues until all deliverable messages have been transferred or when the connection drops.

Concurrently, u also iterates over its other messages to determine which will be relayed to v in the future (lines 16–20). u isolates the messages that v wants (line 15), as dictated by the routing algorithm employed by the node. These messages are then added to u's transfer queue (line 16). If there are messages in u's transfer queue, the one with the highest transfer utility, as dictated by Equation 3, is chosen (line 18) and the transfer to the intended recipient begins (line 19).

5.4.2. Behavior At Message Transfer. With the execution of Algorithm 7, message transfers may be underway. When a transfer for a message $m = (\pi, A, H)$ is completed, another procedure is called for book-keeping and starting the next transfer as defined in Algorithm 8. At a high level:

1. If there is insufficient space to store m, then Algorithm 9 is called to drop messages and make room
2. m is added to v's buffer with updates to appropriate header values
3. u begins its next queued transfer (if any)

Algorithm 8 A transfer of message $m = (\pi, A, H)$ was completed from u to v.

procedure MESSAGETRANSFERRED(m, u, v)
    ▷ Buffer management (see Alg. 9)
    if $|m|_{\texttt{bytes}}$ exceeds free space in B_v **then**
        Call MANAGEBUFFER (v, m)
    end if
    ▷ Finalize message receipt
    Increment $|m|_{\texttt{copies}}$ for both u and v.
    Append v to $m.\texttt{path}$ for only v's copy
    $B_v \leftarrow B_v \cup \{m\}$
    $Q_u \leftarrow Q_u \setminus \{\langle m, v \rangle\}$
    ▷ Begin next transfer
    if $Q_u \neq \emptyset$ **then**
        Let $\langle m^*, v^* \rangle \leftarrow \underset{\langle m, v' \rangle \in Q_u}{\operatorname{argmax}} \{u_t(m)\}$
        Send $m^* \Longrightarrow v^*$
    end if
end procedure

The first step of Algorithm 8 is to manage v's buffer if there is insufficient space for m (lines 2–4) according to Algorithm 9. Then, m's transfer is finalized by updating the copy count for both v's and u's copy (line 5), appending v to the path of the received copy of m (line 6), adding m to v's buffer (line 7), and removing the completed transfer from u's transfer queue (line 8). Finally, the next transfer in u's transfer queue is started if there remains transfers to begin (lines 9–12), whereby the next message with the highest transfer utility is selected (line 10) and sent to its intended recipient (line 11).

5.4.3. Behavior At Buffer Congestion. As a message is being transferred, the receiver stores the received message fragments in a separate buffer until the transfer is complete, after which the message will be moved to the receiver's message buffer. If there is insufficient space in the buffer to store the message due to buffer congestion, then the Catora system will remove a sufficient quantity of messages in order from the buffer management virtual queue, as defined in Definition 15, such that enough space becomes available. Algorithm 9 formally defines these steps.

Algorithm 9 Drop messages from B_v to provide sufficient space for storing m. Let $|B_v|_{\texttt{free}}$ be B_v's free space (bytes).

1: **procedure** ManageBuffer(v, m)
2:     Let $D_v^* \leftarrow \left\{m' | m' \in B_v, \left|m'\right|_{\texttt{copies}} > 1, \left|m'\right|_{\texttt{path}} > 1\right\}$
3:     **while** $|B_v|_{\texttt{free}} < |m|_{\texttt{bytes}}$ **do**
4:         Let $m^* \leftarrow \underset{m' \in D_v^*}{\operatorname{argmin}} \{u_b(m)\}$
5:         $Q_v \leftarrow \left\{\langle m', v'\rangle | \langle m', v'\rangle \in Q_v, m' \neq m^*\right\}$
6:         $D_v^* \leftarrow D_v^* \setminus \left\{m^*\right\}$
7:         $B_v \leftarrow B_v \setminus \left\{m^*\right\}$
8:     **end while**
9: **end procedure**

In Algorithm 9, line 2 starts by constructing the buffer management queue to contain a subset of v's buffer that can be deleted, denoted as D_v^*, consisting of messages that have been relayed at least once – i.e. messages created by v that have not been relayed will not be deleted. These messages are considered *deletable* because they have been replicated and thus still have a chance for delivery through other nodes if removed from v's buffer. In lines 3 through 8, this set is iterated over, with each iteration grabbing the message with the minimum buffer utility, as defined by Equation 4, from the buffer management queue (line 4). Since the message will be removed from v's buffer, any queued transfers containing this message are terminated (line 5). Then, the message is removed from both the buffer management queue (line 6) and v's buffer (line 7). This ends when the freespace available in v's buffer becomes sufficiently large to store m (line 3).

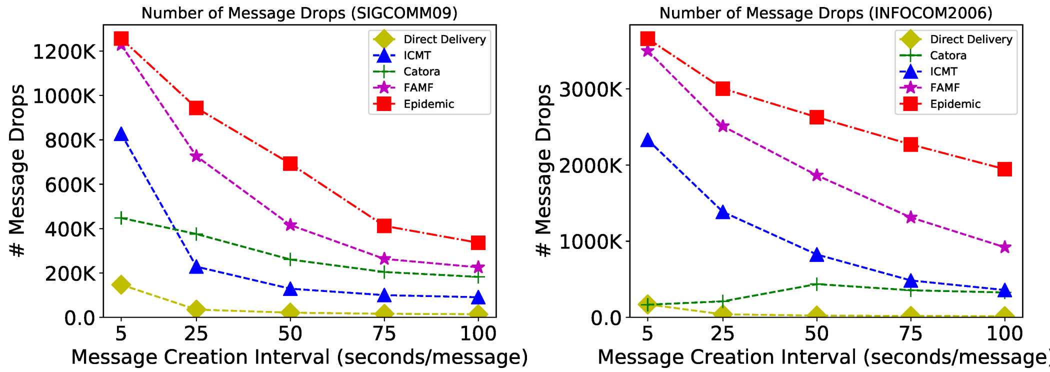
5.4.4. Behavior At Connection Drop. Because it is assumed that the duration of a connection is unknown, a message transfer may be aborted should the transfer's connection drop before completion. Since the Catora system enqueues transfers between nodes at the time of their established connection (via Algorithm 7), those transfers are dequeued upon disconnection. Algorithm 10 formally defines these steps, removing the associated transfers from both u's transfer queue (line 2) and v's transfer queue (line 3).

Algorithm 10 An existing channel between node u and node v is closed.

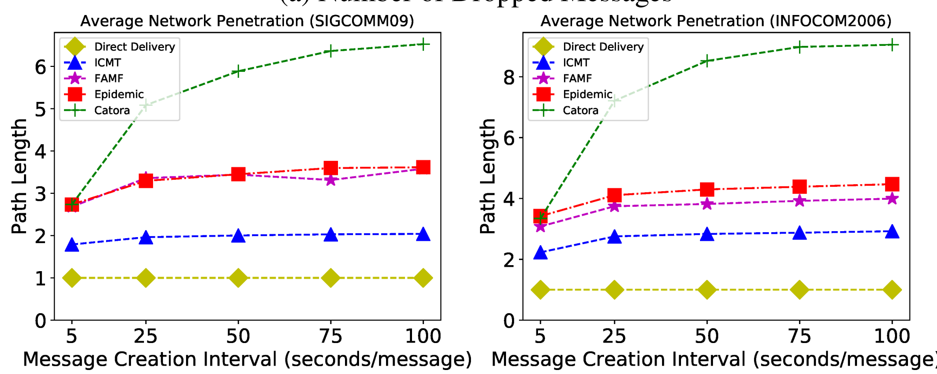
1: **procedure** DISCONNECT(u, v)
2: $\quad Q_u \leftarrow \{\langle m, v' \rangle | \langle m, v' \rangle \in Q_u, v' \neq v\}$
3: $\quad Q_v \leftarrow \{\langle m, v' \rangle | \langle m, v' \rangle \in Q_v, v' \neq u\}$
4: **end procedure**

6. PERFORMANCE EVALUATION

To evaluate the Catora system, a suite of simulations was conducted to compare its performance to that of various benchmark systems and relevant state-of-the-art systems, described below. For benchmarking, three systems were employed: the Direct Delivery router, and two versions of the Epidemic router. The Direct Delivery router restricts message relaying to only when a message's destination is encountered by the message's creator. Thus, it provides a lower-bound on delivery probability and resource consumption and an upper-bound on latency. Similar to the setup in Section 4.1, the two versions of Epidemic are differentiated by whether they are susceptible to congestion. Traditional



(a) Number of Dropped Messages



(b) Average Network Penetration

Figure 3. Congestion Symptoms for SIGCOMM 2009 (left) and INFOCOM 2006 (right)

Epidemic is restricted by buffer space and connection capacity when relaying messages, and thus congestion leads to increasingly dropped messages and aborted relays. The other version, labeled Epidemic*, disreguards all such restrictions and transfers all carried messages to a neighbor when a connection occurs. A message is only dropped when it has been successfully delivered, at which point every node is instantly notified of the successful delivery and drops their copy. As such, Epidemic* is immune to congestion, and provides an upper-bound on delivery probability and resource consumption, and a lower-bound on latency. Epidemic, on the other hand, does not necessarily provide such bounds, but offers a practical reference system for comparison.

Table 2. Catora Simulation Configurations

Configuration	SIGCOMM 2009	INFOCOM 2006
Number of Participants	76 nodes	98 nodes
Simulated time	5 days	4 days
Update Interval	1 second	
Transmission Speed	250 kBps	
Buffer capacity	100 MB	
Message Lifetime	5 hours	
Message Creation Interval	5–100 seconds	
Message Size	25 KB – 1 MB	
Message Attribute Count	20	
Created Messages per Burst	Up to 10	

For comparing against the state-of-the-art, two congestion-mitigating systems are used: the ICMT system [35] and the FAMF system [25]. These systems provide mitigation strategies for controlling congestion, and have been shown to outperform previously proposed systems such as Spray-and-Wait [31], PRoPHET [18], and BUBBLE Rap [11].

Two datasets were used to simulate DTNs: the SIGCOMM 2009 contact trace dataset [21, 22, 23] and the INFOCOM 2006 contact trace dataset [3, 28]. Both datasets are described in detail in Section 4.1. Message creation was conducted in such a way as to introduce redundancy in messages and to induce varying levels of congestion. To simulate this redundancy, a burst of messages were created by a random subset of nodes on a given topic at the same time, with each message being destined for some randomly selected destination. To induce congestion, the rate at which these message bursts occurred was varied between once every 5 seconds (highly congested) to once every 100 seconds (marginal congestion). Each message of a given burst was given a random number of attributes, with the attribute precisions being selected randomly between 0 and 1. Table 2 defines these and other variables used for the simulations.

6.1. CONGESTION AVOIDANCE

The first part of this evaluation examines Catora's ability to tolerate and control congestion in the network. Figure 3 illustrates the impact of the message creation interval on two metrics tied to congestion: message drops and the average network penetration of messages. When a node's buffer becomes full, it must drop carried messages in order to continue creating new messages or receive relayed messages. As observed in Section 4.3, the occurrence of many drops throughout the network is a sign that conditions are becoming congestive. Because Direct Delivery only relays messages directly to destinations, it exhibits the fewest drops and single-hop penetration – thus, it serves as a lower-bound. We omit the performance of Epidemic* for visual clarity due to it producing measurements that are orders of magnitude higher than other systems.

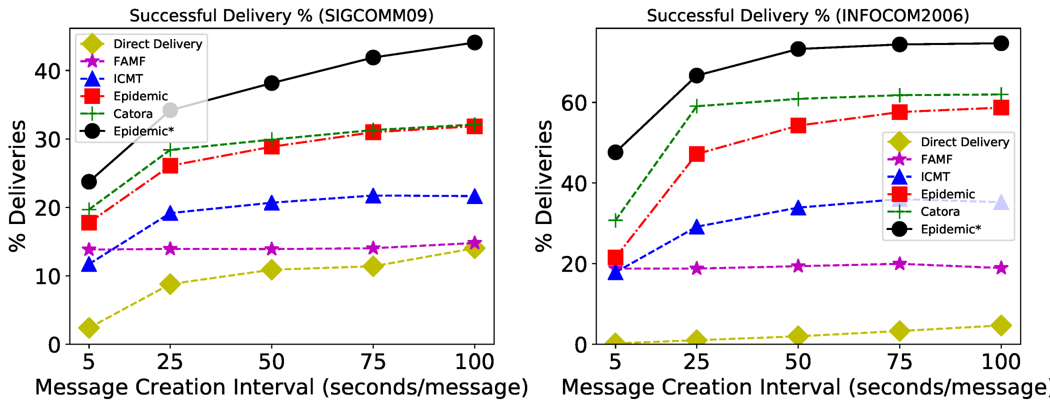
As is illustrated in Figure 3a, the number of message drops increases with the increased frequency of message creation – as more messages are created, buffers fill up quickly and systems must drop more frequently. A side effect of these drops is that messages are unable to penetrate very far into the network, which in turn limits the network's capability to deliver certain messages. Figure 3b illustrates this, showing that under greater congestive conditions, messages on average are only able to travel shorter distances into the network.

In comparison to other systems, these figures indicate that Catora is able to reduce the symptoms of congestion from which a network suffers by performing fewer drops and penetrating messages deeper into the network. In the INFOCOM 2006 network, it outperforms the state-of-the-art and Epidemic by having the fewest drops. In the SIGCOMM 2009 network, its performance is best under significant congestion, only being beaten by the FAMF system at lower congestion. These improvements come from several design factors of the Catora system. First, the delivery notification feedback subsystem opportunistically informs nodes of when a carried message was successfully delivered, thus enabling these nodes to remove the message from their buffers. This has multiple benefits: buffer space is freed to allow for receiving new messages without requiring premature drops, and connections are used to propagate messages that have not yet been delivered.

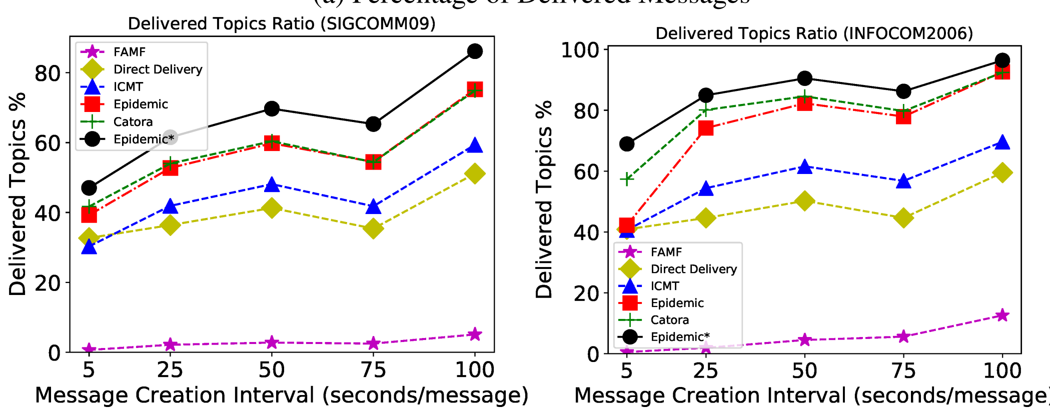
6.2. QUALITY OF SERVICE

Next, the quality of service achieved by these services is evaluated in terms of the percentage of successfully delivered messages (Figure 4a), the percentage of unique topics that were successfully delivered (Figure 4b), and the latency in message delivery (Figure 4c). Overall, these figures illustrate the degradation that occurs with congestion: a lower delivery percentage of both messages and topics occurs when more messages are created (and thus more messages being dropped), and the latency decreases because the messages requiring more time are dropped before delivery.

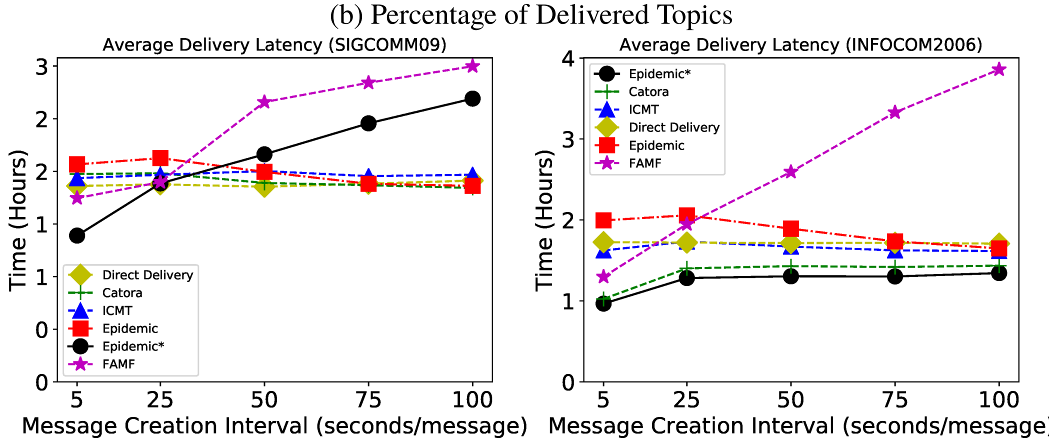
Even with this degradation, Catora is demonstrated to suffer the least of all systems. In Figure 4a, it is able to deliver the highest percentage of messages, even beating Epidemic that is suffering from congestion. This is also observed with the percentage of delivered topics, although Catora's performance is approximately equal to Epidemic in the SIGCOMM 2009 network. With regard to latency, the INFOCOM 2006 network shows Catora achieves a low latency approximately equal to the lower-bound benchmark demonstrated by Epidemic*. The SIGCOMM 2009 network is inconclusive, as most systems are demonstrated to achieve between 1 and 2 hours latency. The improvement to Catora's performance compared to other systems is due to two factors: its capability to avert congestion, and its transmission ordering. As discussed in Section 6.1, Catora's buffer management dampens the effects of congestion by properly freeing up space for receiving more messages and reallocating channel capacity toward more beneficial relaying. The transmission ordering subsystem then takes the available channel capacity and allocates it to messages in order of their delivery or



(a) Percentage of Delivered Messages



(b) Percentage of Delivered Topics



(c) Delivery Latency

Figure 4. Quality of Service for SIGCOMM 2009 (left) and INFOCOM 2006 (right)

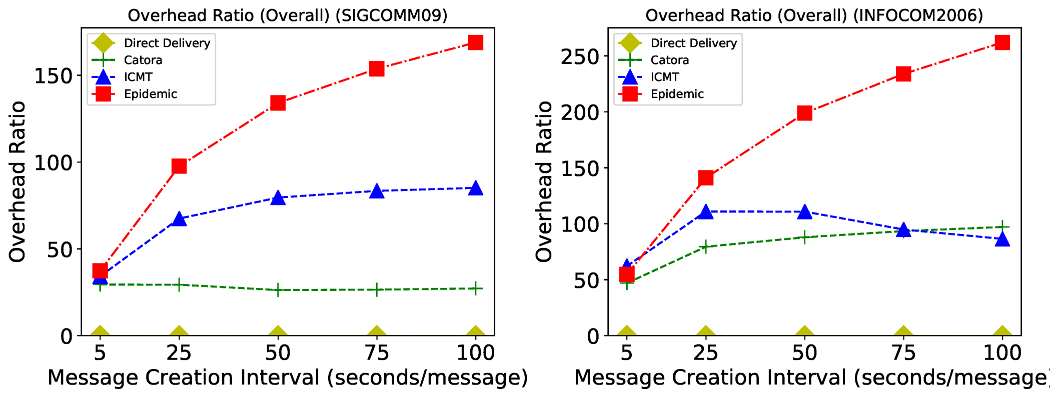
transfer utility. With intermediate nodes receiving higher-quality, non-redundant messages while the channel is up, the likelihood of these messages being relayed further into the network and ultimately reaching their destinations also increases.

6.3. RESOURCE CONSUMPTION

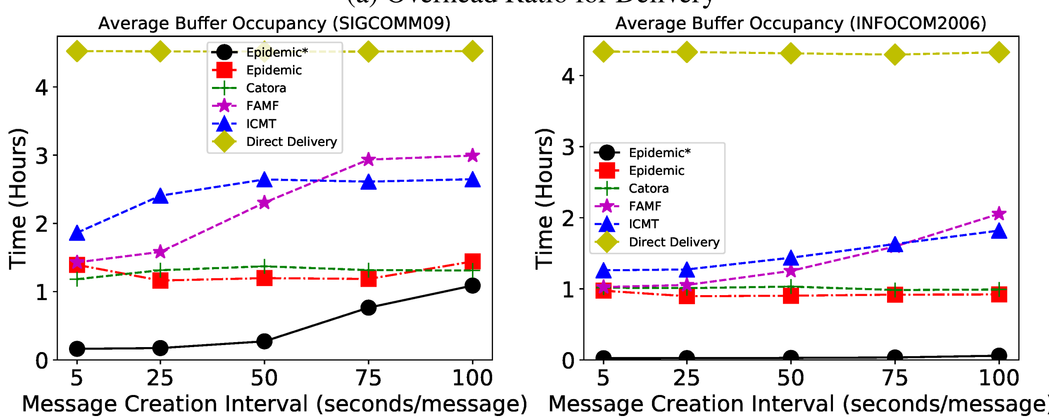
Finally, the consumption of resources is analyzed in terms of the overhead ratio required to deliver messages (Figure 5a, representing the number of message replicas created for each delivered message), the average time in which messages occupied buffers (Figure 5b, representing buffer consumption), and the network-wide energy consumption (Figure 5c).

When facing congestion, it is desirable to observe lower buffer occupancy times as it reflects that messages are not lingering within buffers for too long and not obstructing newer messages from traveling through the network. So, too, is it desirable to observe lower overhead ratios – if too many copies of a message linger too long in the network, both buffer space and channel capacity is consumed so as to further propagate those messages, potentially worsening congestion. With energy consumption, the finite availability of battery power limits the operating lifetime of participating nodes, and congestive conditions can lead to futile consumption of energy for messages that are replicated but ultimately dropped.

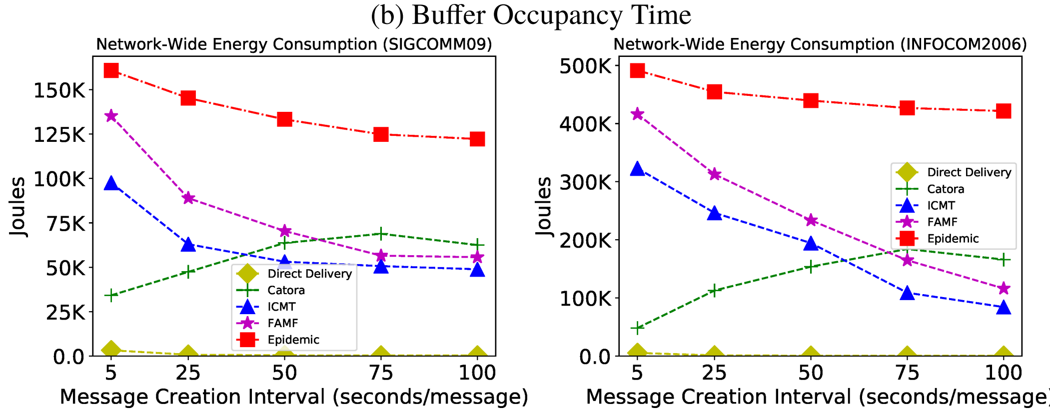
Figure 5a demonstrates Catora's ability to reduce overhead in accomplishing its improved quality of service. The results for the overhead of Epidemic* and FAMF were orders of magnitude larger than other systems, and thus are omitted for visual clarity. Overall, when under congestion, Catora consumes the least overhead of all systems. In the SIGCOMM 2009 network, Catora's overhead is lower than that of all other systems, and remains approximately constant through varying degrees of congestion while the other systems show significant changes. As for the INFOCOM 2006 network, although the ICMT



(a) Overhead Ratio for Delivery



(b) Buffer Occupancy Time



(c) Network-wide Energy Consumption

Figure 5. Resource Consumption for SIGCOMM 2009 (left) and INFOCOM 2006 (right)

system undergoes lower overhead under light to no congestion, Catora's overhead is lower when under heavier congestion. Ultimately, this suggests that networks deploying Catora need not factor in potential levels of congestion when determining energy accomodations.

As for buffer occupancy, Figure 5b shows Catora achieves a low average buffer occupancy messages in line with Epidemic, beating out FAMF and ICMT at all levels of simulated congestion. This is primarily due to Catora's buffer management system. During congestion, messages are appropriately removed (from being delivered or expired) or dropped (from being redundant or low quality), thus preventing them from wastefully consuming any more resources without significantly impacting the network's quality of service. Epidemic simply drops the oldest messages in a buffer to make space for new messages, which impacts its successful deliveries comparatively (see Figure 4a and Figure 4b). This indicates that Catora's strategic buffer management is beneficial to networks that may be susceptible to congestion.

Referencing [35] for the energy consumed during each message transfer (0.125 joules per transfer), Figure 5c depicts the network-wide energy consumption for each of the evaluated systems, with Catora being shown as ideal for scenarios involving higher congestion. Considering both the systems' energy consumption and their achieved delivery probability, as depicted in Figure 4a, Catora's energy consumption is more efficient in its ability to deliver a higher percentage of messages, while other systems consume more energy without such improvements. Of particular interest is that Catora's energy consumption is lower when under congestive conditions than when not. This is due to Catora's design to strategically manage a node's buffer only when necessary so as to continue message propagation, along with its employment of delivery notification feedback to remove delivered messages from node buffers. Both of these strategies reduce the presence of messages in the network, which in turn eliminates the consumption of energy that would otherwise be used to continue propagating poor-quality or already-delivered messages. When conditions are less congested, the Catora buffer management subsystem is called less frequently, resulting in more message copies persisting in the network and propagating further. While Catora consumes more energy than ICMT and FAMF under lesser congestion, this is the cost of achieving more successful deliveries and lower latencies.

7. CONCLUSION AND FUTURE WORK

The Catora system is proposed as a solution to combat congestion and achieve prioritized message delivery in DTNs. Catora's design and procedures revolve around the newly proposed utility density metric of each message, depending on this metric and network-specific properties of messages to construct ordered virtual message queues for the tasks of message relaying, buffer management during bouts of congestion, and message delivery. In doing so, the dissemination of messages through a DTN is balanced and proportional to a message's priority. Simulations from two real-world datasets demonstrate that Catora is capable of outperforming two state-of-the-art congestion control systems with regard to preventing and controlling congestion and improving quality of service – higher delivery ratios and lower latencies – while reducing resource consumption. Even when congestion levels begin to impact the performance of Epidemic, Catora successfully curbs the symptoms of congestion and beats its performance.

REFERENCES

[1] Birrane, E. J., 'Congestion modeling in graph-routed delay tolerant networks with predictive capacity consumption,' in '2013 IEEE Global Communications Conference (GLOBECOM),' ISSN 1930-529X, 2013 pp. 3016–3022, doi: 10.1109/GLOCOM.2013.6831534.

[2] Casteigts, A., Flocchini, P., Mans, B., and Santoro, N., 'Measuring Temporal Lags in Delay-Tolerant Networks,' IEEE Transactions on Computers, 2014, **63**(2), pp. 397–410, ISSN 0018-9340, doi:10.1109/TC.2012.208.

[3] Chaintreau, A., Hui, P., Scott, J., Gass, R., Crowcroft, J., and Diot, C., 'Impact of human mobility on opportunistic forwarding algorithms,' IEEE Transactions on Mobile Computing, 2007, **6**(6), pp. 606–620, (previously published in the Proceedings of IEEE INFOCOM 2006).

[4] Cho, J. and Chen, I., 'Provest: Provenance-based trust model for delay tolerant networks,' IEEE Transactions on Dependable and Secure Computing, 2018, **15**(1), pp. 151–165, ISSN 1545-5971, doi:10.1109/TDSC.2016.2530705.

[5] Daly, E. M. and Haahr, M., 'Social network analysis for information flow in disconnected delay-tolerant manets,' IEEE Transactions on Mobile Computing, 2009, **8**(5), pp. 606–621, ISSN 1536-1233, doi:10.1109/TMC.2008.161.

[6] Datta, S. and Madria, S., 'Efficient photo crowdsourcing in delay-tolerant networks with evolving pois,' in '2019 20th IEEE International Conference on Mobile Data Management (MDM),' ISSN 2375-0324, 2019 pp. 150–159, doi:10.1109/MDM.2019.00-62.

[7] Davis, J. A., Fagg, A. H., and Levine, B. N., 'Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks,' in 'Proceedings Fifth International Symposium on Wearable Computers,' ISSN 1530-0811, 2001 pp. 141–148, doi:10.1109/ISWC.2001.962117.

[8] Ericsson, 'Ericsson mobility report special edition: World economic forum,' Technical report, World Economic Forum, 2019.

[9] Grundy, A. and Radenkovic, M., 'Promoting congestion control in opportunistic networks,' in '2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications,' ISSN 2160-4886, 2010 pp. 324–330, doi:10.1109/WIMOB.2010.5645048.

[10] Hsu, Y. and Hu, C., 'Enhanced buffer management for data delivery to multiple destinations in dtns,' IEEE Transactions on Vehicular Technology, 2016, **65**(10), pp. 8735–8739, ISSN 0018-9545, doi:10.1109/TVT.2015.2503431.

[11] Hui, P., Crowcroft, J., and Yoneki, E., 'BUBBLE rap: Social-based forwarding in delay-tolerant networks,' IEEE Transactions on Mobile Computing, 2011, **10**(11), pp. 1576–1589, ISSN 1536-1233, doi:10.1109/TMC.2010.246.

[12] Iranmanesh, S., 'A novel queue management policy for delay-tolerant networks,' EURASIP Journal on Wireless Communications and Networking, 2016, **2016**(1), p. 88, ISSN 1687-1499, doi:10.1186/s13638-016-0576-6.

[13] Jain, S. and Chawla, M., 'A fuzzy logic based buffer management scheme with traffic differentiation support for delay tolerant networks,' Telecommunication Systems, 2018, **68**(2), pp. 319–335, ISSN 1572-9451, doi:10.1007/s11235-017-0394-0.

[14] Jain, S., Fall, K., and Patra, R., 'Routing in a delay tolerant network,' in 'Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications,' SIGCOMM '04, ACM, New York, NY, USA, ISBN 1-58113-862-8, 2004 pp. 145–158, doi:10.1145/1015467.1015484.

[15] Jiang, G., Chen, J., and Shen, Y., 'Delivery ratio- and buffered time-constrained: Multicasting for delay tolerant networks,' Journal of Network and Computer Applications, 2014, **44**, pp. 92 – 105, ISSN 1084-8045, doi: https://doi.org/10.1016/j.jnca.2014.05.004.

[16] Keränen, A., Ott, J., and Kärkkäinen, T., 'The ONE Simulator for DTN Protocol Evaluation,' in 'Proceedings of the 2nd International Conference on Simulation Tools and Techniques,' SIMUTools '09, ICST, New York, NY, USA, ISBN 978-963-9799-45-5, 2009 .

[17] Liaqat, H. B., Ali, A., Qadir, J., Bashir, A. K., Bilal, M., and Majeed, F., 'Socially-aware congestion control in ad-hoc networks: Current status and the way forward,' Future Generation Computer Systems, 2019, **97**, pp. 634 – 660, ISSN 0167-739X, doi:https://doi.org/10.1016/j.future.2019.02.017.

[18] Lindgren, A., Doria, A., and Schelén, O., 'Probabilistic routing in intermittently connected networks,' in 'Service Assurance with Partial and Intermittent Resources,' volume 3126 of *Lecture Notes in Computer Science*, pp. 239–254, Springer Berlin Heidelberg, ISBN 978-3-540-22567-6, 2004.

[19] Lindgren, A. and Phanse, K. S., 'Evaluation of queueing policies and forwarding strategies for routing in intermittently connected networks,' in '2006 1st International Conference on Communication Systems Software Middleware,' 2006 pp. 1–10, doi: 10.1109/COMSWA.2006.1665196.

[20] Pham, T. N. D., Yeo, C. K., Yanai, N., and Fujiwara, T., 'Detecting flooding attack and accommodating burst traffic in delay-tolerant networks,' IEEE Transactions on Vehicular Technology, 2018, **67**(1), pp. 795–808, ISSN 0018-9545, doi: 10.1109/TVT.2017.2748345.

[21] Pietilainen, A.-K. and Diot, C., 'CRAWDAD dataset thlab/sigcomm2009 (v. 2012-07-15),' Downloaded from http://crawdad.org/thlab/sigcomm2009/20120715, 2012, doi:10.15783/C70P42.

[22] Pietiläinen, A.-K. and Diot, C., 'Dissemination in opportunistic social networks: The role of temporal communities,' in 'MobiHoc'12: Proceedings of the 13th International Symposium on Mobile Ad Hoc Networking and Computing,' 2012 .

[23] Pietiläinen, A.-K., Oliver, E., LeBrun, J., Varghese, G., and Diot, C., 'Mobiclique: Middleware for mobile social networking,' in 'WOSN'09: Proceedings of ACM SIGCOMM Workshop on Online Social Networks,' 2009 .

[24] Roy, A., Acharya, T., and Dasbit, S., 'Quality of service in delay tolerant networks: A survey,' Computer Networks, 2018, **130**, pp. 121 – 133, ISSN 1389-1286, doi: https://doi.org/10.1016/j.comnet.2017.11.010.

[25] Roy, A., Acharya, T., and Dasbit, S., 'Fairness in message delivery in delay tolerant networks,' Wirel. Netw., 2019, **25**(4), pp. 2129–2142, ISSN 1022-0038, doi: 10.1007/s11276-018-1802-4.

[26] Sandulescu, G., Niruntasukrat, A., and Charnsripinyo, C., 'Guaranteed capacity bounds in intermittently-connected networks: A resource-aware, holistic evaluation,' Computer Communications, 2015, **59**, pp. 12 – 23, ISSN 0140-3664, doi: https://doi.org/10.1016/j.comcom.2014.12.008.

[27] Scarsella, A. and Stofega, W., 'Worldwide smartphone forecast, 2019–2023,' Worldwide Smartphone Forecast, 2019.

[28] Scott, J., Gass, R., Crowcroft, J., Hui, P., Diot, C., and Chaintreau, A., 'CRAWDAD dataset cambridge/haggle (v. 2009-05-29),' Downloaded from http://crawdad.org/cambridge/haggle/20090529, 2009, doi:10.15783/C70011.

[29] Silva, A. P., Burleigh, S., Hirata, C. M., and Obraczka, K., 'A survey on congestion control for delay and disruption tolerant networks,' Ad Hoc Networks, 2015, **25**, pp. 480 – 494, ISSN 1570-8705, doi:https://doi.org/10.1016/j.adhoc.2014.07.032, new Research Challenges in Mobile, Opportunistic and Delay-Tolerant Networks Energy-Aware Data Centers: Architecture, Infrastructure, and Communication.

[30] Silva, A. P., Hilário, M. R., Hirata, C. M., and Obraczka, K., 'A percolation-based approach to model dtn congestion control,' in '2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems,' 2015 pp. 100–108, doi: 10.1109/MASS.2015.91.

[31] Spyropoulos, T., Psounis, K., and Raghavendra, C. S., 'Spray and wait: An efficient routing scheme for intermittently connected mobile networks,' in 'Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking,' WDTN '05, ACM, New York, NY, USA, ISBN 1-59593-026-4, 2005 pp. 252–259, doi: 10.1145/1080139.1080143.

[32] Uchida, N., Kuga, H., Ishida, T., and Shibata, Y., 'Implementation of wifi p2p based dtn routing and gateway for disaster information networks,' in L. Barolli, F. Xhafa, N. Javaid, and T. Enokido, editors, 'Innovative Mobile and Internet Services in Ubiquitous Computing,' Springer International Publishing, Cham, ISBN 978-3-319-93554-6, 2019 pp. 139–146.

[33] Vahdat, A. and Becker, D., 'Epidemic routing for partially-connected ad hoc networks,' Technical Report CS-20000, Duke University, 2000.

[34] Wang, K., Guo, H., Shu, L., and Liu, B., 'An improved congestion control algorithm based on social awareness in delay tolerant networks,' in '2014 IEEE International Conference on Communications (ICC),' ISSN 1550-3607, 2014 pp. 1773–1777, doi: 10.1109/ICC.2014.6883579.

[35] Wu, J., Chen, Z., and Zhao, M., 'Information cache management and data transmission algorithm in opportunistic social networks,' Wirel. Netw., 2019, **25**(6), pp. 2977–2988, ISSN 1022-0038, doi:10.1007/s11276-018-1691-6.

[36] Xia, F., Liaqat, H. B., Ahmed, A. M., Liu, L., Ma, J., Huang, R., and Tolba, A., 'User popularity-based packet scheduling for congestion control in ad-hoc social networks,' Journal of Computer and System Sciences, 2016, **82**(1, Part B), pp. 93–112, ISSN 0022-0000, doi:https://doi.org/10.1016/j.jcss.2015.07.002, mobile Social Networking and computing in Proximity (MSNP): A Multi-disciplinary Inspired Approach.

SECTION

3. CONCLUSION

The challenges of sparsity and congestion in Delay Tolerant Networks have been seldom addressed in previous research, and have been the primary motivators for the proposals presented in this document. In acknowledging that these characteristics may exist in a real-world instance of a DTN, it becomes prudent to consider them as potentially obstructing factors against message delivery. Two systems are proposed and evaluated to combat these problems for social-context-based DTNs: the ChitChat system for sparse DTNs, and the Catora system for congested DTNs.

The ChitChat system operates on the basis of social profiles, the set of keywords representing an individual's interests, as defined by each node in the DTN. These are employed to construct data structures that aid in the message delivery through a sparse DTN. First, a node's Transient Social Relationships associates a keyword with a weight to represent the node's ability to reach others with a direct interest in the keyword, either through direct contacts or indirect paths Second, the Geographic Social Heatmap provides information on location semantics for various points of interest in a region through the use of Social Staypoints, a collection of key-weight pairs tied to a location that permits nodes to identify junction points for relaying messages to interested parties. These structures are opportunistically built from a cold-start of the network (as is permissible for DTNs), dynamically evolve to reflect changes in the network, and are employed in a decentralized manner by the ChitChat routing algorithm to choose next-hop routing for messages when encounters occur between nodes. Through simulations, ChitChat is shown to outperform three state-of-the-art DTN systems in the presence of significant network sparsity.

Adjacent to sparsity, the Catora system focuses on DTN congestion as the primary obstacle to address, acknowledging that energy and storage resources are finitely available and transfer opportunities are unpredictable in terms of occurrence and duration. Catora's operations revolve around the strategic ordering of messages based on various metrics depending on the action to be performed: message relaying, message delivery, and buffer management. For message relaying, messages are transferred to neighbors in order of the transfer virtual queue so as to expand a message's dissemination tree in a bredth-first manner, thus insuring wide and balanced dissemination and increase delivery likelihood. Messages that may be immediately delivered to their destination are ordered by the delivery virtual queue so as to expedite the delivery of high utility messages before a connection drops. Congestion is controlled and avoided by two strategies: first, the message delivery feedback system opportunistically drops delivered messages from the buffers of nodes carrying them; second, the buffer management virtual queue orders messages so as to drop bloated messages that have received higher dissemination over others. Both strategies free up buffer space to allow other message flows to continue and halt the consumption of resources for messages that no longer need it. Catora was evaluated against three benchmark systems and two state-of-the-art congestion avoidance systems, and found to outperform in scenarios susceptible to congestion in terms of resource consumption, successful delivery, and latency.

4. FUTURE WORK

Working towards effective message delivery within a sparse DTN still remains a challenging problem, and there exists many potential strategies that could be proposed and investigated. While my previous contributions mend some holes in the tapestry of knowledge, noticeable gaps persist.

4.1. PRIVACY

One glaring assumption for ChitChat's operation is the willingness of nodes to share their social interests for the aid of accurate message routing. These social interests may represent private and sensitive information. To give an example, assume Alice works in a hospital treating HIV-positive patients, and in her free time is very engaged in photography (i.e., showing a direct social interest in it). If Bob encounters Alice, he will receive Alice's TSRs showing high weights for both 'photography' and 'HIV+'. Although the ChitChat system offers some privacy by not explicitly revealing the direct social interests of users, it is not foolproof, and the rigorous analysis of its inherent privacy protection is beyond the scope of the published work. Briefly, some strategies can be investigated, such as the employment of encryption methods (e.g., homomorphic encryption, secure multiparty computation) to hide the semantics of social interests and messages, or the substitution of sensitive keywords with others that are semantically related but more vague/less sensitive (e.g., 'HIV+' could be substituted with 'auto-immune diseases'). The impact on performance could then be analyzed against variations on the degree to which induced vagueness.

4.2. INCENTIVIZATION

A fundamental assumption of both Catora and ChitChat is that nodes are willing to expend resources for others so that they will receive the benefits of having their own messages relayed and delivered. Essentially, both systems function on a quid-pro-quo arrangement. This may not hold in real-world deployments, however, as some individuals may not need a DTN to send and receive messages, and thus are not willing to participate. Incentivizing participation is another open problem in the realm of DTN research that would provide contributions towards an increased effectiveness in DTN deployments.

4.3. MACHINE LEARNING

Techniques from machine learning may also be of interest for improving DTN performance, but their disconnected nature inhibits traditional techniques, requiring significant quantities of data, from being adopted. The application of certain machine learning techniques that are adapted for a DTN's opportunistic nature may be applied to a number of challenges touched upon in this manuscript. For instance, the successful delivery of messages through certain individuals, or alternatively the repeated failure of deliveries therein, could lend itself to being used in an iterative feedback-based learning system for next-hop node selection. Another possible avenue is the adoption of association rule learning for identifying social interests that are frequently associated with those that are known. For example, if Alice is interested in 'outdoor activities', it may be inferred that she is also interested in 'hiking' or 'kayaking' regardless of whether she specifies these interests. Alternatively, Alice's interest in a vegetarian implies she is uninterested in meat-based cuisine, and thus it would not be prudent to send meat-related messages to her. The foreseeable primary challenge of these strategies lies in maintaining accuracy of these learning engines and aggregating information from others encountered in the DTN.

REFERENCES

[1] Abdelkader, T., Naik, K., and Nayak, A., ‘Choosing the objective of optimal routing protocols in delay tolerant networks,’ in ‘Computer Engineering Conference (ICENCO), 2010 International,’ 2010 pp. 16–21, doi: 10.1109/ICENCO.2010.5720418.

[2] Abdelkader, T., Naik, K., Nayak, A., Goel, N., and Srivastava, V., ‘A performance comparison of delay-tolerant network routing protocols,’ IEEE Network, 2016, **30**(2), pp. 46–53, ISSN 0890-8044, doi:10.1109/MNET.2016.7437024.

[3] Alic, K. and Svigelj, A., ‘A one-hop opportunistic network coding algorithm for wireless mesh networks,’ Wireless Networks, 2018, **24**(4), pp. 1007–1018, ISSN 1572-8196, doi:10.1007/s11276-016-1384-y.

[4] Amin, R., Ripplinger, D., Mehta, D., and Cheng, B. N., ‘Design considerations in applying disruption tolerant networking to tactical edge networks,’ 2015, **53**(10), pp. 32–38, ISSN 0163-6804, doi:10.1109/MCOM.2015.7295460.

[5] Barbosa, H., Barthelemy, M., Ghoshal, G., James, C. R., Lenormand, M., Louail, T., Menezes, R., Ramasco, J. J., Simini, F., and Tomasini, M., ‘Human mobility: Models and applications,’ Physics Reports, 2018, **734**, pp. 1 – 74, ISSN 0370-1573, doi:https://doi.org/10.1016/j.physrep.2018.01.001, human mobility: Models and applications.

[6] Basu, S., Roy, S., Bandyopadhyay, S., and Das Bit, S., ‘A utility driven post disaster emergency resource allocation system using dtn,’ IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018, pp. 1–13, ISSN 2168-2216, doi: 10.1109/TSMC.2018.2813008.

[7] Bedogni, L., Fiore, M., and Glacet, C., ‘Temporal reachability in vehicular networks,’ in ‘IEEE INFOCOM 2018 - IEEE Conference on Computer Communications,’ 2018 pp. 81–89, doi:10.1109/INFOCOM.2018.8486393.

[8] Birrane, E. J., ‘Congestion modeling in graph-routed delay tolerant networks with predictive capacity consumption,’ in ‘2013 IEEE Global Communications Conference (GLOBECOM),’ ISSN 1930-529X, 2013 pp. 3016–3022, doi: 10.1109/GLOCOM.2013.6831534.

[9] Boldrini, C. and Passarella, A., ‘HCMM: Modelling spatial and temporal properties of human mobility driven by usersâĂŹ social relationships,’ Computer Communications, 2010, **33**(9), pp. 1056–1074, ISSN 0140-3664, doi: 10.1016/j.comcom.2010.01.013.

[10] Bulut, E. and Szymanski, B., 'Exploiting friendship relations for efficient routing in mobile social networks,' IEEE Transactions on Parallel and Distributed Systems, 2012, **23**(12), pp. 2254–2265, ISSN 1045-9219, doi:10.1109/TPDS.2012.83.

[11] Cabaniss, R., Vulli, S. S., and Madria, S., 'Social group detection based routing in delay tolerant networks,' Wireless Networks, 2013, **19**(8), pp. 1979–1993, ISSN 1022-0038, doi:10.1007/s11276-013-0580-2.

[12] Cao, Y. and Sun, Z., 'Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges,' 2013, **15**(2), pp. 654–677, ISSN 1553-877X, doi: 10.1109/SURV.2012.042512.00053.

[13] Cao, Y., Sun, Z., Wang, N., Riaz, M., Cruickshank, H., and Liu, X., 'Geographic-based spray-and-relay (gsar): An efficient routing scheme for dtns,' IEEE Transactions on Vehicular Technology, 2015, **64**(4), pp. 1548–1564, ISSN 0018-9545, doi:10.1109/TVT.2014.2331395.

[14] Caputo, A., Socievole, A., and Rango, F. D., 'CRAWDAD dataset unical/socialblueconn (v. 2015-02-08),' Downloaded from http://crawdad.org/unical/socialblueconn/20150208, 2015, doi:10.15783/C7GK5R.

[15] Casteigts, A., Flocchini, P., Mans, B., and Santoro, N., 'Measuring Temporal Lags in Delay-Tolerant Networks,' IEEE Transactions on Computers, 2014, **63**(2), pp. 397–410, ISSN 0018-9340, doi:10.1109/TC.2012.208.

[16] Casteigts, A., Flocchini, P., Quattrociocchi, W., and Santoro, N., 'Time-varying graphs and dynamic networks,' International Journal of Parallel, Emergent and Distributed Systems, 2012, **27**(5), pp. 387–408, doi:10.1080/17445760.2012.668546.

[17] Chaintreau, A., Hui, P., Scott, J., Gass, R., Crowcroft, J., and Diot, C., 'Impact of human mobility on opportunistic forwarding algorithms,' IEEE Transactions on Mobile Computing, 2007, **6**(6), pp. 606–620, (previously published in the Proceedings of IEEE INFOCOM 2006).

[18] Chaintreau, A., Hui, P., Scott, J., Gass, R., Crowcroft, J., and Diot, C., 'Impact of human mobility on opportunistic forwarding algorithms,' IEEE Transactions on Mobile Computing, 2007, **6**(6), pp. 606–620, (previously published in the Proceedings of IEEE INFOCOM 2006).

[19] Chen, H. and Lou, W., 'Contact expectation based routing for delay tolerant networks,' Ad Hoc Networks, 2016, **36**, pp. 244 – 257, ISSN 1570-8705, doi: https://doi.org/10.1016/j.adhoc.2015.07.017.

[20] Cho, J. and Chen, I., 'Provest: Provenance-based trust model for delay tolerant networks,' IEEE Transactions on Dependable and Secure Computing, 2018, **15**(1), pp. 151–165, ISSN 1545-5971, doi:10.1109/TDSC.2016.2530705.

[21] Cho, J. and Chen, I., 'Provest: Provenance-based trust model for delay tolerant networks,' IEEE Transactions on Dependable and Secure Computing, 2018, **15**(1), pp. 151–165, ISSN 1545-5971, doi:10.1109/TDSC.2016.2530705.

[22] Daly, E. M. and Haahr, M., 'Social network analysis for routing in disconnected delay-tolerant manets,' in 'Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing,' MobiHoc '07, ACM, New York, NY, USA, ISBN 978-1-59593-684-4, 2007 pp. 32–40, doi:10.1145/1288107.1288113.

[23] Daly, E. M. and Haahr, M., 'Social network analysis for information flow in disconnected delay-tolerant MANETs,' 2009, **8**(5), pp. 606–621, ISSN 1536-1233, doi:10.1109/TMC.2008.161.

[24] Daly, E. M. and Haahr, M., 'Social network analysis for information flow in disconnected delay-tolerant manets,' IEEE Transactions on Mobile Computing, 2009, **8**(5), pp. 606–621, ISSN 1536-1233, doi:10.1109/TMC.2008.161.

[25] Datta, S. and Madria, S., 'Efficient photo crowdsourcing in delay-tolerant networks with evolving pois,' in '2019 20th IEEE International Conference on Mobile Data Management (MDM),' ISSN 2375-0324, 2019 pp. 150–159, doi:10.1109/MDM.2019.00-62.

[26] Datta, S., Madria, S., Milligan, J., and Linderman, M., 'Secure information forwarding through fragmentation in delay-tolerant networks,' in '2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS),' ISSN 2575-8462, 2018 pp. 93–102, doi:10.1109/SRDS.2018.00020.

[27] Davis, J. A., Fagg, A. H., and Levine, B. N., 'Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks,' in 'Proceedings Fifth International Symposium on Wearable Computers,' ISSN 1530-0811, 2001 pp. 141–148, doi:10.1109/ISWC.2001.962117.

[28] Desta, M. S., Hyytiä, E., Ott, J., and Kangasharju, J., 'Characterizing content sharing properties for mobile users in open city squares,' in '2013 10th Annual Conference on Wireless On-demand Network Systems and Services (WONS),' 2013 pp. 147–154, doi:10.1109/WONS.2013.6578340.

[29] Eagle, N. and Pentland, A. S., 'CRAWDAD dataset mit/reality (v. 2005-07-01),' Downloaded from http://crawdad.org/mit/reality/20050701, 2005, doi:10.15783/C71S31.

[30] Eagle, N., Pentland, A. S., and Lazer, D., 'Inferring friendship network structure by using mobile phone data,' Proceedings of the National Academy of Sciences, 2009, **106**(36), pp. 15274–15278, doi:10.1073/pnas.0900282106.

[31] Eagle, N. and (Sandy) Pentland, A., 'Reality mining: sensing complex social systems,' Personal and Ubiquitous Computing, 2005, **10**(4), pp. 255–268, ISSN 1617-4917, doi:10.1007/s00779-005-0046-3.

[32] Ericsson, 'Ericsson mobility report special edition: World economic forum,' Technical report, World Economic Forum, 2019.

[33] Fall, K., 'A delay-tolerant network architecture for challenged internets,' in 'Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications,' SIGCOMM '03, ACM, New York, NY, USA, ISBN 1-58113-735-4, 2003 pp. 27–34, doi:10.1145/863955.863960.

[34] Galati, A., Sazonovs, A., Olivares, M., Mangold, S., and Gross, T. R., 'Delay tolerant networking for the socio-economic development in rural south africa,' in O. Gaggi, P. Manzoni, C. Palazzi, A. Bujari, and J. M. Marquez-Barja, editors, 'Smart Objects and Technologies for Social Good,' Springer International Publishing, Cham, ISBN 978-3-319-61949-1, 2017 pp. 195–202.

[35] Gao, W., Cao, G., La Porta, T., and Han, J., 'On exploiting transient social contact patterns for data forwarding in delay-tolerant networks,' IEEE Transactions on Mobile Computing, 2013, **12**(1), pp. 151–165, ISSN 1536-1233, doi: 10.1109/TMC.2011.249.

[36] Girvan, M. and Newman, M. E. J., 'Community structure in social and biological networks,' 2002, **99**(12), pp. 7821–7826, ISSN 0027-8424, 1091-6490, doi: 10.1073/pnas.122653799.

[37] Grasic, S. and Lindgren, A., 'Revisiting a remote village scenario and its DTN routing objective,' 2014, **48**, pp. 133–140, ISSN 0140-3664, doi: 10.1016/j.comcom.2014.04.003.

[38] Grundy, A. and Radenkovic, M., 'Promoting congestion control in opportunistic networks,' in '2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications,' ISSN 2160-4886, 2010 pp. 324–330, doi:10.1109/WIMOB.2010.5645048.

[39] Guo, H., Wang, X., Cheng, H., and Huang, M., 'A location aided controlled spraying routing algorithm for delay tolerant networks,' Ad Hoc Networks, 2017, **66**, pp. 16 – 25, ISSN 1570-8705, doi:https://doi.org/10.1016/j.adhoc.2017.08.005.

[40] Hattachi, R. E. and Erfanian, J., 'Ngmn 5g white paper,' Technical Report 1.0, Next Generation Mobile Networks Ltd, 2015.

[41] Hsu, Y. and Hu, C., 'Enhanced buffer management for data delivery to multiple destinations in dtns,' IEEE Transactions on Vehicular Technology, 2016, **65**(10), pp. 8735–8739, ISSN 0018-9545, doi:10.1109/TVT.2015.2503431.

[42] Hui, P., Crowcroft, J., and Yoneki, E., 'BUBBLE rap: Social-based forwarding in delay-tolerant networks,' IEEE Transactions on Mobile Computing, 2011, **10**(11), pp. 1576–1589, ISSN 1536-1233, doi:10.1109/TMC.2010.246.

[43] Hui, P., Crowcroft, J., and Yoneki, E., 'Bubble rap: Social-based forwarding in delay-tolerant networks,' IEEE Transactions on Mobile Computing, 2011, **10**(11), pp. 1576–1589, ISSN 1536-1233, doi:10.1109/TMC.2010.246.

[44] Iranmanesh, S., 'A novel queue management policy for delay-tolerant networks,' EURASIP Journal on Wireless Communications and Networking, 2016, **2016**(1), p. 88, ISSN 1687-1499, doi:10.1186/s13638-016-0576-6.

[45] Jain, S. and Chawla, M., 'A fuzzy logic based buffer management scheme with traffic differentiation support for delay tolerant networks,' Telecommunication Systems, 2018, **68**(2), pp. 319–335, ISSN 1572-9451, doi:10.1007/s11235-017-0394-0.

[46] Jain, S., Fall, K., and Patra, R., 'Routing in a delay tolerant network,' in 'Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications,' SIGCOMM '04, ACM, New York, NY, USA, ISBN 1-58113-862-8, 2004 pp. 145–158, doi:10.1145/1015467.1015484.

[47] Jethawa, H. and Madria, S., 'Reputation and credit based incentive mechanism for data-centric message delivery in dtns,' in '2018 19th IEEE International Conference on Mobile Data Management (MDM),' ISSN 2375-0324, 2018 pp. 207–216, doi: 10.1109/MDM.2018.00038.

[48] Jiang, G., Chen, J., and Shen, Y., 'Delivery ratio- and buffered time-constrained: Multicasting for delay tolerant networks,' Journal of Network and Computer Applications, 2014, **44**, pp. 92 – 105, ISSN 1084-8045, doi: https://doi.org/10.1016/j.jnca.2014.05.004.

[49] Jindal, A. and Psounis, K., 'Performance analysis of epidemic routing under contention,' in 'Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing,' IWCMC '06, ACM, New York, NY, USA, ISBN 1-59593-306-9, 2006 pp. 539–544, doi:10.1145/1143549.1143657.

[50] Johnson, D. B. and Maltz, D. A., 'Dynamic source routing in ad hoc wireless networks,' in 'Mobile Computing,' Kluwer Academic Publishers, 1996 pp. 153–181.

[51] Jurgensen, J., 'Concert crowds flounder in digital dead zones,' 2014, ISSN 0099-9660.

[52] Keränen, A., Ott, J., and Kärkkäinen, T., 'The ONE Simulator for DTN Protocol Evaluation,' in 'Proceedings of the 2nd International Conference on Simulation Tools and Techniques,' SIMUTools '09, ICST, New York, NY, USA, ISBN 978-963-9799-45-5, 2009 .

[53] Kim, S. and Gerla, M., 'Socio-geo: Social network routing protocol in delay tolerant networks,' in '2016 International Conference on Computing, Networking and Communications (ICNC),' 2016 pp. 1–6, doi:10.1109/ICCNC.2016.7923427.

[54] Kim, S.-H. and Han, S.-J., ‘Distinguishing social contacts and pass-by contacts in dtn routing,’ Telecommunication Systems, 2018, **68**(4), pp. 669–685, ISSN 1572-9451, doi:10.1007/s11235-017-0416-y.

[55] Kiukkonen, N., Blom, J., Dousse, O., Gatica-Perez, D., and Laurila, J., ‘Towards rich mobile phone datasets: Lausanne data collection campaign,’ in ‘Proceedings of the 2010 ACM International Conference on Pervasive Services,’ 2010 .

[56] Ko, Y. and Vaidya, N. H., ‘Location-aided routing (lar) in mobile ad hoc networks,’ Wireless Networks, 2000, **6**(4), pp. 307–321, ISSN 1572-8196, doi: 10.1023/A:1019106118419.

[57] Kosta, S., Mei, A., and Stefa, J., ‘Large-scale synthetic social mobile networks with SWIM,’ IEEE Transactions on Mobile Computing, 2014, **13**(1), pp. 116–129, ISSN 1536-1233, doi:10.1109/TMC.2012.229.

[58] Laurila, J., Gatica-Perez, D., Aad, I., Blom, J., Bornet, O., Do, T.-M.-T., Dousse, O., Eberle, J., and Miettinen, M., ‘The mobile data challenge: Big data for mobile computing research,’ in ‘Pervasive Computing,’ 2012 .

[59] Li, F., Jiang, H., Li, H., Cheng, Y., and Wang, Y., ‘Sebar: Social-energy-based routing for mobile social delay-tolerant networks,’ IEEE Transactions on Vehicular Technology, 2017, **66**(8), pp. 7195–7206, ISSN 0018-9545, doi:10.1109/TVT.2017.2653843.

[60] Li, W., Galluccio, L., Bassi, F., and Kieffer, M., ‘Distributed faulty node detection in delay tolerant networks: Design and analysis,’ IEEE Transactions on Mobile Computing, 2018, **17**(4), pp. 831–844, ISSN 1536-1233, doi:10.1109/TMC.2017.2743703.

[61] Li, Z. and Shen, H., ‘Sedum: Exploiting social networks in utility-based distributed routing for dtns,’ IEEE Transactions on Computers, 2013, **62**(1), pp. 83–97, ISSN 0018-9340, doi:10.1109/TC.2011.232.

[62] Liaqat, H. B., Ali, A., Qadir, J., Bashir, A. K., Bilal, M., and Majeed, F., ‘Socially-aware congestion control in ad-hoc networks: Current status and the way forward,’ Future Generation Computer Systems, 2019, **97**, pp. 634 – 660, ISSN 0167-739X, doi:https://doi.org/10.1016/j.future.2019.02.017.

[63] Lindgren, A., Doria, A., and Schelén, O., ‘Probabilistic routing in intermittently connected networks,’ in ‘Service Assurance with Partial and Intermittent Resources,’ volume 3126 of *Lecture Notes in Computer Science*, pp. 239–254, Springer Berlin Heidelberg, ISBN 978-3-540-22567-6, 2004.

[64] Lindgren, A. and Phanse, K. S., ‘Evaluation of queueing policies and forwarding strategies for routing in intermittently connected networks,’ in ‘2006 1st International Conference on Communication Systems Software Middleware,’ 2006 pp. 1–10, doi: 10.1109/COMSWA.2006.1665196.

[65] Liu, M., Yang, Y., and Qin, Z., 'A survey of routing protocols and simulations in delay-tolerant networks,' in Y. Cheng, D. Y. Eun, Z. Qin, M. Song, and K. Xing, editors, 'Wireless Algorithms, Systems, and Applications,' Number 6843 in Lecture Notes in Computer Science, pp. 243–253, Springer Berlin Heidelberg, ISBN 978-3-642-23489-7 978-3-642-23490-3, 2011, DOI: 10.1007/978-3-642-23490-3_22.

[66] Lu, Z., Sun, X., and La Porta, T., 'Cooperative data offload in opportunistic networks: From mobile devices to infrastructure,' IEEE/ACM Trans. Netw., 2017, **25**(6), pp. 3382–3395, ISSN 1063-6692, doi:10.1109/TNET.2017.2747621.

[67] Machado, K., Boukerche, A., de Melo, P. O. V., Cerqueira, E., and Loureiro, A. A., 'Pervasive forwarding mechanism for mobile social networks,' Computer Networks, 2016, **111**, pp. 6 – 16, ISSN 1389-1286, doi: https://doi.org/10.1016/j.comnet.2016.08.022, cyber-physical systems for Mobile Opportunistic Networking in Proximity (MNP).

[68] McGeehan, D., Lin, D., and Madria, S., 'Theorems and Proofs of Bounded Storage Complexity in the ChitChat PSN Routing System,' Technical report, Missouri University of Science and Technology, Department of Computer Science, 2015.

[69] McGeehan, D., Lin, D., and Madria, S., 'Chitchat: An effective message delivery method in sparse pocket-switched networks,' in '2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS),' IEEE Computer Society, Los Alamitos, CA, USA, ISSN 1063-6927, 2016 pp. 457–466, doi: 10.1109/ICDCS.2016.69.

[70] McGeehan, D., Madria, S., and Lin, D., 'Effective social-context based message delivery using chitchat in sparse delay tolerant networks,' Distributed and Parallel Databases, 2019, ISSN 1573-7578, doi:10.1007/s10619-019-07274-x.

[71] Mei, A., Morabito, G., Santi, P., and Stefa, J., 'Social-aware stateless routingin pocket switched networks,' IEEE Transactions on Parallel and Distributed Systems, 2015, **26**(1), pp. 252–261, ISSN 1045-9219, doi:10.1109/TPDS.2014.2307857.

[72] MongiovÃň, M., Singh, A. K., Yan, X., Zong, B., and Psounis, K., 'Efficient multicasting for delay tolerant networks using graph indexing,' in '2012 Proceedings IEEE INFOCOM,' 2012 pp. 1386–1394, doi:10.1109/INFCOM.2012.6195503.

[73] Nahrstedt, K. and Vu, L., 'CRAWDAD dataset uiuc/uim (v. 2012-01-24),' Downloaded from http://crawdad.org/uiuc/uim/20120124, 2012, doi:10.15783/C7HG6D.

[74] Ott, J., Hyytiä, E., Lassila, P., Vaegs, T., and Kangasharju, J., 'Floating content: Information sharing in urban areas,' in '2011 IEEE International Conference on Pervasive Computing and Communications (PerCom),' 2011 pp. 136–146, doi: 10.1109/PERCOM.2011.5767578.

[75] Perkins, C. E. and Royer, E. M., 'Ad-hoc on-demand distance vector routing,' in 'Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on,' 1999 pp. 90–100, doi:10.1109/MCSA.1999.749281.

[76] Petre, A.-C., Chilipirea, C., and Dobre, C., ‘Delay tolerant networks for disaster scenarios,’ in ‘Resource Management in Mobile Computing Environments,’ Number 3 in Modeling and Optimization in Science and Technologies, pp. 3–24, Springer International Publishing, ISBN 978-3-319-06703-2 978-3-319-06704-9, 2014, DOI: 10.1007/978-3-319-06704-9_1.

[77] Pham, T. N. D., Yeo, C. K., Yanai, N., and Fujiwara, T., ‘Detecting flooding attack and accommodating burst traffic in delay-tolerant networks,’ IEEE Transactions on Vehicular Technology, 2018, **67**(1), pp. 795–808, ISSN 0018-9545, doi: 10.1109/TVT.2017.2748345.

[78] Pietilainen, A.-K. and Diot, C., ‘CRAWDAD data set thlab/sigcomm2009 (v. 2012-07-15),’ Downloaded from http://crawdad.org/thlab/sigcomm2009/, 2012.

[79] Pietilainen, A.-K. and Diot, C., ‘CRAWDAD dataset thlab/sigcomm2009 (v. 2012-07-15),’ Downloaded from http://crawdad.org/thlab/sigcomm2009/20120715, 2012, doi:10.15783/C70P42.

[80] Pietiläinen, A.-K. and Diot, C., ‘Dissemination in opportunistic social networks: The role of temporal communities,’ in ‘MobiHoc’12: Proceedings of the 13th International Symposium on Mobile Ad Hoc Networking and Computing,’ 2012 .

[81] Pietiläinen, A.-K., Oliver, E., LeBrun, J., Varghese, G., and Diot, C., ‘Mobiclique: Middleware for mobile social networking,’ in ‘WOSN’09: Proceedings of ACM SIGCOMM Workshop on Online Social Networks,’ 2009 .

[82] Pietilänen, A.-K. and Diot, C., ‘Dissemination in opportunistic social networks: The role of temporal communities,’ in ‘Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing,’ MobiHoc ’12, ACM, New York, NY, USA, ISBN 978-1-4503-1281-3, 2012 pp. 165–174, doi:10.1145/2248371.2248396.

[83] Rango, F. D., Socievole, A., and Marano, S., ‘Exploiting online and offline activity-based metrics for opportunistic forwarding,’ Wireless Networks, 2014, **21**(4), pp. 1163–1179, ISSN 1022-0038, 1572-8196, doi:10.1007/s11276-014-0842-7.

[84] Rasul, K., Chowdhury, S. A., Makaroff, D., and Stanley, K., ‘Community-based forwarding for low-capacity pocket switched networks,’ in ‘Proceedings of the 17th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems,’ MSWiM ’14, ACM, ISBN 978-1-4503-3030-5, 2014 pp. 249–257, doi:10.1145/2641798.2641801.

[85] Rhee, I., Shin, M., Hong, S., Lee, K., Kim, S. J., and Chong, S., ‘On the levy-walk nature of human mobility,’ IEEE/ACM Trans. Netw., 2011, **19**(3), pp. 630–643, ISSN 1063-6692, doi:10.1109/TNET.2011.2120618.

[86] Rhee, I., Shin, M., Hong, S., Lee, K., Kim, S. J., and Chong, S., ‘On the levy-walk nature of human mobility,’ IEEE/ACM Transactions on Networking, 2011, **19**(3), pp. 630–643, doi:10.1109/TNET.2011.2120618.

[87] Roy, A., Acharya, T., and Dasbit, S., 'Quality of service in delay tolerant networks: A survey,' Computer Networks, 2018, **130**, pp. 121 – 133, ISSN 1389-1286, doi: https://doi.org/10.1016/j.comnet.2017.11.010.

[88] Roy, A., Acharya, T., and Dasbit, S., 'Fairness in message delivery in delay tolerant networks,' Wirel. Netw., 2019, **25**(4), pp. 2129–2142, ISSN 1022-0038, doi: 10.1007/s11276-018-1802-4.

[89] Saha, S., Sushovan, Sheldekar, A., C, R. J., Mukherjee, A., and Nandi, S., 'Post disaster management using delay tolerant network,' in A. Özcan, J. Zizka, and D. Nagamalai, editors, 'Recent Trends in Wireless and Mobile Networks,' Number 162 in Communications in Computer and Information Science, pp. 170–184, Springer Berlin Heidelberg, ISBN 978-3-642-21936-8 978-3-642-21937-5, 2011, DOI: 10.1007/978-3-642-21937-5_16.

[90] Sandulescu, G., Niruntasukrat, A., and Charnsripinyo, C., 'Guaranteed capacity bounds in intermittently-connected networks: A resource-aware, holistic evaluation,' Computer Communications, 2015, **59**, pp. 12 – 23, ISSN 0140-3664, doi: https://doi.org/10.1016/j.comcom.2014.12.008.

[91] Scarsella, A. and Stofega, W., 'Worldwide smartphone forecast, 2019–2023,' Worldwide Smartphone Forecast, 2019.

[92] Scott, J., Gass, R., Crowcroft, J., Hui, P., Diot, C., and Chaintreau, A., 'CRAWDAD dataset cambridge/haggle (v. 2009-05-29),' Downloaded from http://crawdad.org/cambridge/haggle/20090529, 2009, doi:10.15783/C70011.

[93] Silva, A. P., Burleigh, S., Hirata, C. M., and Obraczka, K., 'A survey on congestion control for delay and disruption tolerant networks,' Ad Hoc Networks, 2015, **25**, pp. 480 – 494, ISSN 1570-8705, doi:https://doi.org/10.1016/j.adhoc.2014.07.032, new Research Challenges in Mobile, Opportunistic and Delay-Tolerant Networks Energy-Aware Data Centers: Architecture, Infrastructure, and Communication.

[94] Silva, A. P., Hilário, M. R., Hirata, C. M., and Obraczka, K., 'A percolation-based approach to model dtn congestion control,' in '2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems,' 2015 pp. 100–108, doi:10.1109/MASS.2015.91.

[95] Silva, A. P., Hilário, M. R., Hirata, C. M., and Obraczka, K., 'A percolation-based approach to model dtn congestion control,' in '2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems,' 2015 pp. 100–108, doi:10.1109/MASS.2015.91.

[96] Silva, A. P., Obraczka, K., Burleigh, S., and Hirata, C. M., 'Smart congestion control for delay- and disruption tolerant networks,' in '2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON),' 2016 pp. 1–9, doi:10.1109/SAHCN.2016.7733018.

[97] Socievole, A., Caputo, A., and Marano, S., 'Multi-layer sociality in opportunistic networks: An extensive analysis of online and offline node social behaviors,' in '2015 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS),' 2015 pp. 1–8, doi:10.1109/SPECTS.2015.7285280.

[98] Socievole, A., Yoneki, E., De Rango, F., and Crowcroft, J., 'ML-SOR: Message routing using multi-layer social networks in opportunistic communications,' Computer Networks, 2015, **81**, pp. 201–219, ISSN 1389-1286, doi:10.1016/j.comnet.2015.02.016.

[99] Sommer, P., Kusy, B., Valencia, P., Dungavell, R., and Jurdak, R., 'Delay-tolerant networking for long-term animal tracking,' IEEE Internet Computing, 2018, **22**(1), pp. 62–72, ISSN 1089-7801, doi:10.1109/MIC.2018.011581518.

[100] Spyropoulos, T., Psounis, K., and Raghavendra, C. S., 'Spray and wait: An efficient routing scheme for intermittently connected mobile networks,' in 'Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking,' WDTN '05, ACM, New York, NY, USA, ISBN 1-59593-026-4, 2005 pp. 252–259, doi:10.1145/1080139.1080143.

[101] Spyropoulos, T., Psounis, K., and Raghavendra, C. S., 'Spray and wait: An efficient routing scheme for intermittently connected mobile networks,' in 'Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking,' WDTN '05, ACM, New York, NY, USA, ISBN 1-59593-026-4, 2005 pp. 252–259, doi:10.1145/1080139.1080143.

[102] Spyropoulos, T., Turletti, T., and Obraczka, K., 'Routing in delay-tolerant networks comprising heterogeneous node populations,' IEEE Transactions on Mobile Computing, 2009, **8**(8), pp. 1132–1147, ISSN 1536-1233, doi:10.1109/TMC.2008.172.

[103] Stewart, M., Kannan, R., Dvir, A., and Krishnamachari, B., 'Caspar: Congestion avoidance shortest path routing for delay tolerant networks,' International Journal of Distributed Sensor Networks, 2017, **13**(11), p. 1550147717741264, doi:10.1177/1550147717741264.

[104] Suthaputchakun, C. and Sun, Z., 'Multihop broadcast protocol in intermittently connected vehicular networks,' IEEE Transactions on Aerospace and Electronic Systems, 2018, **54**(2), pp. 616–628, ISSN 0018-9251, doi:10.1109/TAES.2017.2761140.

[105] Uchida, N., Kuga, H., Ishida, T., and Shibata, Y., 'Implementation of wifi p2p based dtn routing and gateway for disaster information networks,' in L. Barolli, F. Xhafa, N. Javaid, and T. Enokido, editors, 'Innovative Mobile and Internet Services in Ubiquitous Computing,' Springer International Publishing, Cham, ISBN 978-3-319-93554-6, 2019 pp. 139–146.

[106] Vahdat, A. and Becker, D., 'Epidemic routing for partially-connected ad hoc networks,' Technical Report CS-20000, Duke University, 2000.

[107] Velásquez-Villada, C. and Donoso, Y., 'Delay/disruption tolerant network-based message forwarding for a river pollution monitoring wireless sensor network application,' Sensors, 2016, **16**(4), ISSN 1424-8220, doi:10.3390/s16040436.

[108] Wang, E., Yang, Y., Wu, J., and Liu, W., 'Phone-to-phone communication utilizing wifi hotspot in energy-constrained pocket switched networks,' IEEE Transactions on Vehicular Technology, 2016, **65**(10), pp. 8578–8590, ISSN 0018-9545, doi: 10.1109/TVT.2015.2514098.

[109] Wang, K., Guo, H., Shu, L., and Liu, B., 'An improved congestion control algorithm based on social awareness in delay tolerant networks,' in '2014 IEEE International Conference on Communications (ICC),' ISSN 1550-3607, 2014 pp. 1773–1777, doi:10.1109/ICC.2014.6883579.

[110] Wang, N. and Wu, J., 'Optimal cellular traffic offloading through opportunistic mobile networks by data partitioning,' in '2018 IEEE International Conference on Communications (ICC),' ISSN 1938-1883, 2018 pp. 1–6, doi:10.1109/ICC.2018.8422387.

[111] Wang, N. and Wu, J., 'Optimal data partitioning and forwarding in opportunistic mobile networks,' in '2018 IEEE Wireless Communications and Networking Conference (WCNC),' ISSN 1558-2612, 2018 pp. 1–6, doi:10.1109/WCNC.2018.8376979.

[112] Wang, R., Wang, X., Hao, F., Zhang, L., Liu, S., Wang, L., and Lin, Y., 'Social identity-aware opportunistic routing in mobile social networks,' Transactions on Emerging Telecommunications Technologies, 2018, **29**(5), p. e3297, doi: 10.1002/ett.3297, e3297 ett.3297.

[113] Wang, Y., Sheng, M., Zhuang, W., Zhang, S., Zhang, N., Liu, R., and Li, J., 'Multi-resource coordinate scheduling for earth observation in space information networks,' IEEE Journal on Selected Areas in Communications, 2018, **36**(2), pp. 268–279, ISSN 0733-8716, doi:10.1109/JSAC.2018.2804045.

[114] Wang, Y. and Wu, J., 'Social-tie-based information dissemination in mobile opportunistic social networks,' in 'World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a,' 2013 pp. 1–6, doi:10.1109/WoWMoM.2013.6583411.

[115] Weiss, T. R., 'In emergencies, can cell phone network overload be prevented?' 2007.

[116] Wright, J., 'Dispelling common bluetooth misconceptions,' Technical report, SANS Technology Institute Security Laboratory, 2017.

[117] Wu, J., Chen, Z., and Zhao, M., 'Information cache management and data transmission algorithm in opportunistic social networks,' Wirel. Netw., 2019, **25**(6), pp. 2977–2988, ISSN 1022-0038, doi:10.1007/s11276-018-1691-6.

[118] Wu, J., Wang, J., Liu, L., Tanha, M., and Pan, J., 'A data forwarding scheme with reachable probability centrality in dtns,' in '2016 IEEE Wireless Communications and Networking Conference,' ISSN 1558-2612, 2016 pp. 1–6, doi: 10.1109/WCNC.2016.7565160.

[119] Wu, Y., Deng, S., and Huang, H., 'Control of message transmission in delay/disruption tolerant network,' IEEE Transactions on Computational Social Systems, 2018, **5**(1), pp. 132–143, ISSN 2329-924X, doi:10.1109/TCSS.2017.2776322.

[120] Wu, Y., Wang, Y., Hu, W., Zhang, X., and Cao, G., 'Resource-aware photo crowdsourcing through disruption tolerant networks,' in '2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS),' ISSN 1063-6927, 2016 pp. 374–383, doi:10.1109/ICDCS.2016.18.

[121] Xia, F., Liaqat, H. B., Ahmed, A. M., Liu, L., Ma, J., Huang, R., and Tolba, A., 'User popularity-based packet scheduling for congestion control in ad-hoc social networks,' Journal of Computer and System Sciences, 2016, **82**(1, Part B), pp. 93–112, ISSN 0022-0000, doi:https://doi.org/10.1016/j.jcss.2015.07.002, mobile Social Networking and computing in Proximity (MSNP): A Multi-disciplinary Inspired Approach.

[122] Xu, D., Li, Y., Chen, X., Li, J., Hui, P., Chen, S., and Crowcroft, J., 'A survey of opportunistic offloading,' IEEE Communications Surveys Tutorials, 2018, **20**(3), pp. 2198–2236, ISSN 1553-877X, doi:10.1109/COMST.2018.2808242.

[123] Yan, L., Shen, H., and Chen, K., 'Mobit: Distributed and congestion-resilient trajectory-based routing for vehicular delay tolerant networks,' IEEE/ACM Transactions on Networking, 2018, **26**(3), pp. 1078–1091, ISSN 1063-6692, doi: 10.1109/TNET.2018.2812169.

[124] Yang, S., Yeo, C. K., and Lee, F. B. S., 'Cooperative duty cycling for energy-efficient contact discovery in pocket switched networks,' IEEE Transactions on Vehicular Technology, 2013, **62**(4), pp. 1815–1826, ISSN 0018-9545, doi: 10.1109/TVT.2012.2227865.

[125] Yuan, P., Fan, L., Liu, P., and Tang, S., 'Recent progress in routing protocols of mobile opportunistic networks: A clear taxonomy, analysis and evaluation,' 2016, **62**, pp. 163–170, ISSN 1084-8045, doi:10.1016/j.jnca.2016.01.006.

[126] Zhang, B., Teng, J., Bai, X., Yang, Z., and Xuan, D., 'P^3-coupon: A probabilistic system for prompt and privacy-preserving electronic coupon distribution,' in '2011 IEEE International Conference on Pervasive Computing and Communications (PerCom),' 2011 pp. 93–101, doi:10.1109/PERCOM.2011.5767599.

[127] Zhang, F., Thiyagalingam, J., Kirubarajan, T., and Xu, S., 'Speed-adaptive multi-copy routing for vehicular delay tolerant networks,' Future Generation Computer Systems, 2019, **94**, pp. 392 – 407, ISSN 0167-739X, doi: https://doi.org/10.1016/j.future.2018.12.006.

[128] Zhang, J., Huang, H., Yang, C., Liu, J., Fan, Y., and Yang, G., 'Destination-aware metric based social routing for mobile opportunistic networks,' Wireless Networks, 2019, ISSN 1572-8196, doi:10.1007/s11276-018-01907-2.

[129] Zhang, L., Cai, Z., Lu, J., and Wang, X., 'Mobility-aware routing in delay tolerant networks,' Personal and Ubiquitous Computing, 2015, **19**(7), pp. 1111–1123, ISSN 1617-4917, doi:10.1007/s00779-015-0880-x.

[130] Zhang, X. and Cao, G., 'Transient community detection and its application to data forwarding in delay tolerant networks,' in '2013 21st IEEE International Conference on Network Protocols (ICNP),' 2013 pp. 1–10, doi:10.1109/ICNP.2013.6733594.

[131] Zheng, Y. and Fu, H., *Geolife GPS trajectory dataset - User Guide*, 2011.

[132] Zheng, Y., Li, Q., Chen, Y., Xie, X., and Ma, W.-Y., 'Understanding mobility based on gps data,' in 'Proceedings of the 10th International Conference on Ubiquitous Computing,' UbiComp '08, ACM, New York, NY, USA, ISBN 978-1-60558-136-1, 2008 pp. 312–321, doi:10.1145/1409635.1409677.

[133] Zheng, Y., Xie, X., and Ma, W.-Y., 'Geolife: A collaborative social networking service among user, location and trajectory,' IEEE Data Engineering Bulletin, 2010.

[134] Zheng, Y., Zhang, L., Xie, X., and Ma, W.-Y., 'Mining interesting locations and travel sequences from gps trajectories,' in 'Proceedings of the 18th International Conference on World Wide Web,' WWW '09, ACM, New York, NY, USA, ISBN 978-1-60558-487-4, 2009 pp. 791–800, doi:10.1145/1526709.1526816.

[135] Zhou, D., Sheng, M., Wang, X., Xu, C., Liu, R., and Li, J., 'Mission aware contact plan design in resource-limited small satellite networks,' IEEE Transactions on Communications, 2017, **65**(6), pp. 2451–2466, ISSN 0090-6778, doi:10.1109/TCOMM.2017.2685383.

[136] Zhou, H., Ruan, M., Zhu, C., Leung, V. C. M., Xu, S., and Huang, C., 'A time-ordered aggregation model-based centrality metric for mobile social networks,' IEEE Access, 2018, **6**, pp. 25588–25599, ISSN 2169-3536, doi:10.1109/ACCESS.2018.2831247.

[137] Zhu, Y., Xu, B., Shi, X., and Wang, Y., 'A survey of social-based routing in delay tolerant networks: Positive and negative social effects,' IEEE Communications Surveys Tutorials, 2013, **15**(1), pp. 387–401, ISSN 1553-877X, doi:10.1109/SURV.2012.032612.00004.

VITA

Douglas John McGeehan was born in Jefferson City, Missouri. In attending the Missouri University of Science and Technology, he earned his Bachelor's degree in Computer Science in May 2013, and continued into the Ph.D. program in Computer Science by joining the Wireless to Cloud Computer Laboratory through a Department of Education GAANN fellowship. Under the advisement of Dr. Sanjay Madria and the co-mentorship of Dr. Dan Lin, he focused on tackling open problems in Delay Tolerant Networks, and investigated strategies stemming from graph theory, distributed machine learning, social networking theory, and control systems engineering. His studies as a Ph.D. student brought him to present in such conferences as the International Symposium on Reliable Distributed Systems (SRDS) and the IEEE International Conference on Distributed Computing Systems (ICDCS), along with conducting research and development with the Air Force Research Laboratory under the Summer Faculty Fellowship Program. Beyond his research, he earned numerous awards and held various leadership roles during his Ph.D. studies; from the Computer Science department, he received both the Mentor Award and Ambassador Award in 2014 and both the Outstanding Graduate Teaching Assistant Award and the Leadership Award in 2018; he served as the Nepal Earthquake Fundraising Director in May 2015, successfully raising over $6,000 over two days for disaster relief; and for the KMNR 89.7 FM student radio station, he served as the Chief Engineer and Station Manager during the 2017–18 and 2018–19 school years, respectively. In May 2020, he received his Ph.D. degree in Computer Science from the Missouri University of Science and Technology.