Properties and Impact of Vicinity in Mobile Opportunistic Networks

Tiphaine Phe-Neau

Université Pierre et Marie Curie
UPMC Sorbonne Universités

École doctorale

Informatique, Télécommunications et Électronique (Paris)

Thèse de Doctorat

pour obtenir le grade de

Docteur de l'Université Pierre et Marie Curie

présentée par

Tiphaine Phe-Neau

Propriétés et impact du voisinage dans les réseaux mobiles opportunistes

à soutenir le 23 janvier 2014 devant le jury composé de:

André-Luc Beylot	*Rapporteur*	Professeur, ENSEEIHT
Emmanuel Lochin	*Rapporteur*	Professeur, ISAE
Aline Carneiro Viana	*Examinateur*	Chargée de Recherche, INRIA
Vania Conan	*Examinateur*	Ingénieur, Thales Communications & Security
Anne Fladenmuller	*Examinateur*	Maître de Conférence, UPMC Sorbonne Universités
Vincent Gauthier	*Examinateur*	Maître de Conférence, Telecom SudParis
Marcelo Dias de Amorim	*Directeur*	Directeur de recherche, CNRS and UPMC Sorbonne Universités

Université Pierre et Marie Curie
UPMC Sorbonne Universités

Doctoral school

Computer Science, Telecommunications, and Electronics (Paris)

PhD Dissertation

submitted for the degree of

Doctor of Science of the Université Pierre et Marie Curie

presented by

Tiphaine Phe-Neau

Properties and Impact of Vicinity in Mobile Opportunistic Networks

to be defended on the 23^{rd} of January 2014 with the committee :

André-Luc Beylot	*Reviewer*	Professor, ENSEEIHT
Emmanuel Lochin	*Reviewer*	Professor, ISAE
Aline Carneiro Viana	*Examiner*	Researcher, INRIA
Vania Conan	*Examiner*	Researcher, Thales Communications & Security
Anne Fladenmuller	*Examiner*	Associate Professor, UPMC Sorbonne Universités
Vincent Gauthier	*Examiner*	Associate Professor, Telecom SudParis
Marcelo Dias de Amorim	*Advisor*	Research Director, CNRS and UPMC Sorbonne Universités

Résumé

Notre décennie a connu une augmentation spectaculaire du taux d'équipement en appareils mobiles tels les smartphones, ordinateurs portables, tablettes multimedia ou consoles de jeux. En ce qui concerne les smartphones, quasi 50% des ménages en sont actuellement pourvus. Tous ces équipements nous accompagnent dans notre vie de tous les jours et surtout lors de nos trajets quotidiens. Au delà de l'aspect pratique de ces appareils, tous ces nouveaux gadgets alliant haute capacité de calcul et mémoire embarquée grandissante transforment les nomades urbains en de puissants vecteurs d'information. L'information circule avec nous et peut être diffusée à partir de nous. Afin de profiter de ces nouveaux vecteurs de diffusions, la communauté scientifique commença à définir de nouveaux types de réseaux de communications laissant de plus grands degrés de liberté aux communications. Ainsi les réseaux opportunistes ou réseaux tolérant aux interruptions (DTN) permettent d'utiliser ces nouveaux vecteurs de transmission car leur fonctionnement-même tolère une extrême latence dans les communications et bénéficie de la mobilité des usagers.

Cependant avant de pouvoir profiter de toutes les capacités des DTN, la communauté scientifique doit tout d'abord se pencher sur la compréhension de ce nouveau paradigme. De nombreuses études se sont donc attelées à la tâche en essayant de comprendre les interactions dans les réseaux ainsi que l'évolution de leur structure topologique. D'autres se sont plutôt focalisées sur la compréhension des phénomènes de contact et d'intercontact s'y déroulant. De nombreuses propriétés concernant les réseaux DTN en général sont maintenant reconnues, mais les relations entre un noeud du réseau et celle de son voisinage proche ne semblent pas encore avoir été passées au crible. Ainsi, dans la plupart des études que nous avons pu observer, la présence de noeuds voisins proches mais pas directement liés par le contact est souvent ignorée. Dans cette thèse, nous tentons de montrer à quel point considérer les noeuds à proximité mais pas forcément en contact peut nous aider à améliorer les performances des DTN.

En identifiant et analysant le paradoxe binaire dans les DTN, nous montrons que les caractérisations actuelles basées sur la notion binaire de contact/intercontact ne sont pas suffisantes pour bénéficier de toutes les possibilités de transmission dans les DTN. Dans un réseau DTN, deux noeuds peuvent ne pas être en contact direct mais être tout de même lié par un chemin de longueur 2 ainsi, ils seraient tout de même proche. Cependant la vision binaire actuelle ne permet pas de garder cette information. Afin de mieux comprendre cette notion de proximité dans les DTN, nous proposons une définition formelle du voisinage pour les réseaux opportunistes nommée le "κ-vicinity". Nous étudions les caractérisations temporelles du κ-vicinity dans différents jeux de données et montrons que cette nouvelle caractérisation est liée à la précédente, issue de la vision binaire de contact/intercontact.

Ensuite, nous nous sommes concentré sur l'organisation interne du κ-vicinity afin de comprendre les mouvements de voisinage qui y règnent. Nous avons créé le Vicinity Motion qui est un analyseur permettant d'obtenir automatiquement un modèle markovien du κ-vicinity à partir de n'importe quelle trace de contact. Le Vicinity Motion modélise les distances entre les noeuds en tant qu'état et les transitions entre états comme la probabilité de passer directement d'une distance à une autre lorsque deux noeuds appartiennent au κ-vicinity. Nous avons pu extraire trois mouvements principaux dans les κ-vicinity: la naissance, la mort et les mouvements séquentiels. Grâce aux valeurs obtenues avec le Vicinity Motion, nous avons pu créer un générateur synthétique de mouvements de proximité nommé TiGeR.

Enfin, nous nous sommes posé la question de la prévisibilité des distances entre deux noeuds situé dans le κ-vicinity. En utilisant le savoir emmagasiné dans le Vicinity Motion et ses probabilités de transition, nous avons mis au point une heuristique permettant de prédire les futures distances entre deux noeuds. La particularité de notre heuristique est qu'elle fournit deux distances possibles pour les prochains intervalles considérés et que cette méthode peut s'étendre au n^{ime} intervalle suivant.

Toutes nos analyses ont pour but de démontrer les améliorations que peut apporter la notion de voisinage dans la compréhension et l'utilisation des réseaux opportunistes.

Abstract

The market of mobile devices, such as smartphones, tablets, game stations, or laptops has exponentially grown over the latest years. In 2013, when considering smartphones only, the worldwide penetration ratio is already around 50%. These devices have the necessary CPU and memory capacities to create, send, and forward information on the go. When people carry such equipments along their daily commuting, they become mobile information vectors. They are able to carry, send, or receive information whenever they meet each other. The networking paradigm using such information vectors is known as disruption-tolerant networks (DTN) or opportunistic networks.

Before being able to exploit all the capacities of DTN, we need to better understand their fundamental characteristics and potentials. Many inspiring studies focused on characterizing network structures as well as node specific properties like degree or betweenness centralities. Other analyses concentrated their efforts toward contact and intercontact characterization. Even if some properties are quite well known today, the relationship between a given node and its *vicinity* has not been thoroughly studied yet. For instance, the presence of nearby neighbors including nodes without direct contacts is often neglected in opportunistic approaches. In this thesis, we show how this closeness notion is a key ingredient to improve opportunistic network forwarding.

We begin by identifying and investigating the *binary assertion* issue in opportunistic networks. We notice how most DTNs mainly analyze nodes that are in contact (at a topological 1-hop distance). This vision implies that all nodes that are not in contact, are in intercontact. Nevertheless, when two nodes are not in contact, this does not mean that they are topologically far away from one another. For instance, a 2- or 3-hop path may link them. Following this chain of thoughts, we propose a formal definition of vicinities in DTNs called "κ-vicinity" and study the new resulting "contact/intercontact" temporal characterization. We show how extended temporal distributions differ from previous binary distributions.

Then, we examine the internal organization of vicinities using the asynchronous vicinity motion (AVM) framework. Asynchronous vicinity motion models pairwise vicinity movements as a chain. Each chain state indicates the shortest distance between nodes. Each state is linked to the others by a transitional probability. We highlight movement types such as birth, death, and sequential moves. We analyze a number of their characteristics and extract vicinity usage directions for mobile networks. Based on the vicinity motion outputs and extracted directions, we build TiGeR (a synthetic TImeline GEneratoR) that simulates how pairs of nodes interact within their vicinities. Vicinity motion and TiGeR are able to take into account various types of networks to generate synthetic vicinity behaviors following similar patterns. Both modules will be available in the Vicinity package that we provide on our dedicated website: http://vicinity.lip6.fr.

Finally, we inquire about the possibilities of vicinity distance prediction. We expose a vicinity motion-based heuristic for pairwise shortest distance forecasting. For this part, we also define a synchronous vicinity motion model (SVM) which is time-aware and analyzes datasets every τ seconds instead of following network dynamics like AVM. We find that our heuristics perform quite well with performances up to 99% for the synchronous vicinity motion-based scheme and around 40% for the asynchronous one. We must note that these measures are enhanced by the fact that our heuristic often predicts infinite pairwise distances (i.e., pure intercontact) and most of the datasets we observe are mainly disconnected. Still, they are interesting indicators of whether two nodes are likely to be close in the future.

Keywords: Disruption-tolerant networks, opportunistic networks, vicinity, k-contact, k-intercontact, contact, intercontact.

Contents

Chapter 1

Introduction

Computer networks have mutated a lot since their first appearance in the 1950's. In 1969, the Advanced Research Projects Agency Network (ARPANET) displayed the first characteristics of networks we currently know. ARPANET used packet switching, the TCP/IP protocol, and wired connections. A natural subsequent next step was to remove physical wires and replace them using wireless technologies. The most popular standard in use now being IEEE 802.11 (Wi-Fi). There are other well known standards such as 2G, 3G, 4G, NFC, ZigBee, or Bluetooth. Nowadays, all these technologies are often embedded in small portable devices like laptops, tablets, smartphones, or game stations (Playstation PS Vita, Nintendo 3DS). Most people carry at least one of these devices everyday along their daily commuting. Such devices enable people to carry, store and forward data wherever they go, transforming them into potential oblivious data carriers. The opportunistic network paradigm embraces such original information vectors and uses them in order to carry data.

This dissertation presents an alternative vision of opportunistic networks using the notion of *vicinity*. In this chapter, we first present the original characteristics of mobile opportunistic networks as well as most challenges they face in Section 1.1. Then in Section 1.2, we present an example motivating our original approach to opportunistic network characterization. In Section 1.3, we highlight the problems we tackle in our work. Finally in Section 1.4, we present our contributions together with the outline of this dissertation.

1.1 Opportunistic Networks: Characteristics and Challenges

This all started high in the sky with satellite networks and the idea of an Interplanetary Internet [1]. To work with such type of networks is clearly different from working with our common wired or Wi-Fi networks. Compared to most systems, interplanetary networks do bear unusual features. For instance, topological distance between spatial nodes is often of thousands of kilometers inducing long delays between the emission and reception of a signal. These long delays are considered faulty in usual networks. However in this case, it is a natural part of their functioning.

The opportunistic networking research area gained attention in 2003 when Kevin

Figure 1.1: DTN/Opportunistic networks substrate of use: urban connectivity.

Fall formalized the characteristics of a "delay-tolerant network architecture" for challenged networks [2]. By bringing a back-to-earth vision to challenged networks, Fall sparked a lot of interests in our fellow scientists. By lifting a few technical constraints, our community could extend wireless networks use to such challenged networks. This new paradigm could be applicable to urban areas with urban nomads always carrying connected devices (laptops, smartphones, etc.) When two or more of these devices are close enough, they have potential connectivity and transmitting powers thanks to their embedded technologies. Transmitting information hop-by-hop between these moving devices becomes theoretically possible. Such challenged networks are called "disruption-tolerant networks" (DTN) or "opportunistic networks".[1] In this thesis, we focus on urban human-driven networks (networks between user-carried devices in a city setting, see Fig. 1.1).

Opportunistic networks rely on device's "short" range connectivity (currently Bluetooth, NFC or Wi-Fi Direct) to transmit data. Therefore, nodes can transmit data only when they are close enough i.e., in *contact*. Fig. 1.2 presents an example of a 4-node DTN. At time t_1, A and B as well as C and D are connected. Next at t_2, they form a chain and finally at t_3, B moves away leaving A, C, and D fully connected. Any disruption-tolerant protocol should cope with such a connected/disconnected scenario.

[1] We will alternatively be using these two terms in this dissertation. One may also encounter them under the names: pocket switched networks (PSN), dynamic networks or intermittently connected mobile networks (ICMN).

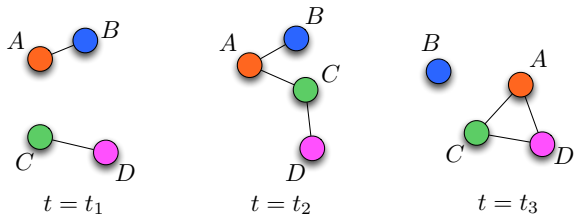


Figure 1.2: An example of disruption-tolerant/opportunistic network.

In order to reach the destination, nodes use a hop-by-hop "store, carry, and forward" scheme. Their main functioning differences compared to classic networks are as follows:

- **High latency.** Between the source and the destination, there may be several store and forward processes occurring after different nodes collocation. This can take quite some time between actually sending a message and receiving it. This high latency would not work with traditional networking paradigms. As an example, the TCP protocol would have to be completely retuned to fit this high latency.

- **Disconnection periods.** One of the challenges of DTN is to tolerate complete disconnection periods in their process. Any previous networking paradigm considered a disconnection in their session to be an error and had to resume the process at best or to restart it completely at worst. In DTN, nodes do not need to have an explicit connection or session from the sender to the receiver. This not-necessarily-connected property of DTN offers much more freedom in network utilization than any other existing paradigm.

- **Mobility.** Previous network paradigms or other research fields deem mobility as an opponent to good functioning. Here, a very interesting feature of human-driven opportunistic networks is the mobility of the nodes. Everyday people commute to work and travel long distances carrying connected devices.[2] Whenever they are close to someone, they can transmit data and during travel times, commuters encounter many persons that can act as potential data carriers. They may even encounter some people regularly without realizing it. This regular encounter phenomenon is called "the familiar stranger" and can be use for opportunistic forwarding [3]. Mobility allows us benefiting from these opportunistic encounters while previous networks did not. However, mobility is not the same for everyone nor is it a random process, therefore, its implications are not that obvious.

- **Limited longevity. Limited resources. Low data rates.** These properties stem from the devices in use in DTN. Portable devices like smartphones, laptop,

[2]On average, commuting takes 25 minutes un the US but around 1.7 million Americans are "mega-commuters" and have commuting duration of 90 minutes or more and of length at least 80 kilometers.

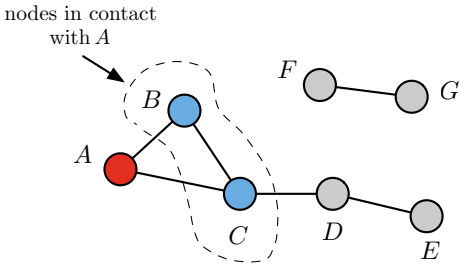


Figure 1.3: A snapshot of a 7-node network at instant t. A is in contact with B and C while nodes D, E, F, and G are beyond contact. From A's point of view, most opportunistic approaches only use the knowledge that B and C are nearby (direct connectivity knowledge). If A wanted to transmit something to E, who would it send it to? B or C. With this current vision, A cannot easily determine who to send the message to.

or game stations work on batteries and have a limited daily lifespan. The more applications we use, the less batteries last. So, finding the accurate power mode for opportunistic networking is an important issue. Limited resources also come from the devices capacities, but as technology progresses, we find smartphones almost as powerful as desktops and the main constraint resides in the embedded wireless technology. However, if we consider opportunistic networks between vehicles for example, restrictions over battery life, memory resources, or data rates are weaker.

- **Security.** For mobile networks, security always remains an issue. As long as a device stores data to forward it and respects the content's privacy, a node has no idea of what it stores. Therefore the security issue here is as hard to deal with as it is with any other mobile network.

1.2 Motivating Example

Let us observe an example of opportunistic networks. In most opportunistic approaches, decisions are made depending on what happens around a given node, more precisely which nodes are in "contact" with it. This notion of contact is fundamental in DTN as we will see throughout this manuscript.

Fig. 1.3 presents a snapshot of an opportunistic network with 7 nodes. At instant t, when we observe the situation through node A's point of view, A has two nodes in contact with it (namely B and C). All the remaining nodes from D to F are beyond contact. All opportunistic decisions concerning A are made according to A's current states and both B and C information. Still, we observe that A has an end-to-end path to nodes D and E so their behavior could and should influence A's forwarding decisions. For instance, A could easily transfer data to both D and E, or probe their encounters history to decide whether they are "good" carriers.

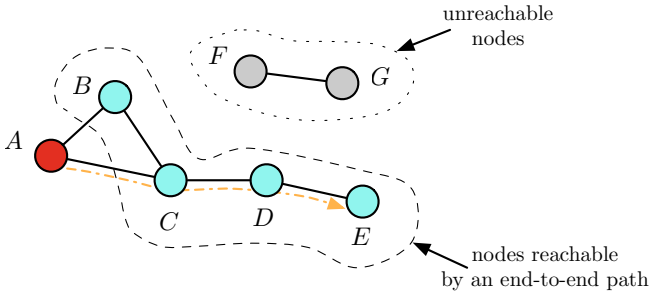


Figure 1.4: The same 7-node network snapshot but with a neighborhood-wise vision for node A. Here, A would take into account all nodes to whom it has a contemporaneous path instead of only considering nodes in contact. If A wanted to send a message to E, it would use C as a next hop because C belongs to the shortest path between C and E at that moment.

Now consider that A wants to send a message to E. A would only know that B and C are currently in contact and, at most, that D is reachable. A would more likely only know that it is in contact with B and C. So who should A use as next hop carrier? Using only contact knowledge, it is hard to tell as B and C seem to have the same properties to E – not in contact. But if A considered not only contact but the knowledge of all its nearby nodes and neighbors, A would certainly send to C to minimize the path length between A and E – 3 hops instead of 4 (see Fig. 1.4). This example shows that the current approach of considering only nodes in contacts in DTN may hamper a node's vision of the situation and prevents it from taking optimal decisions network-wise.

1.3 Problem Statement

In this dissertation, we question the current opportunistic network vision in use. We address two main issues.

Question 1: Is the current DTN vision enough? As we have shown in the previous example, considering only contacts in DTN overlooks many closeby opportunities. So, does ignoring neighbors beyond contacts harm DTN usage? The historical contact notion in DTN remains important as it settles direct communication possibilities. Nevertheless, this notion does not limit the capabilities of communication to nodes at a one-hop distance. If there are neighbors in contact, there may be others nodes closeby at a 2- or 3^+-hop distance. In real life, when one is seated on a bus, she may be close to 3 or 4 other commuters but that does not mean she cannot see other people a little further away or that she cannot speak to them. All these bus passengers are potential information carriers and their importance should not be neglected. When they arrive at their bus stop, they move to different places and this non random human-driven mobility expands the range of possible hop-by-hop communications. The more we can spread

a message, the more people we can reach, and the higher the probability of reaching a given destination in shorter delays.

Question 2: Why not considering a proximity notion in DTN? Noticing the flaws of the mere contact vision in DTN. We wonder how node proximity has been treated until now in opportunistic networks. A few intents at using proximity and resulting end-to-end paths of length strictly greater than 1 hop exists but none of them actually defined a notion of close vicinity in DTN [4, 5, 6, 7]. To the best of our knowledge, this is the first time a study formalizes the notion of vicinity in DTN. With a formalized vision of vicinities, we can start characterizing its behavior and the possibilities of communications beyond simple contacts and answer questions like: how marginal are such transmission possibilities? What is the behavior of these vicinities at a network level? Are there specific links between a node and members of its vicinity? What are the properties of vicinities in DTN? Replying to these interrogations can help opportunistic networks understanding on many stages. Plus, understanding vicinity movements may improve opportunistic knowledge bootstrapping in DTN protocols.

1.4 Contributions of this Thesis

In this dissertation, we answer the previously raised questions through the following contributions.

Contribution 1: Uncovering the Properties of Intercontacts in DTN. We first wonder if the DTN vision limited to nodes in contact is enough to benefit from DTN's original properties. We begin by showing how sub-optimal such a vision is and then formalize the concept of vicinity in opportunistic networks. To the best of our knowledge, our work is the first to ever expose a precise definition for vicinity in DTN namely *κ-vicinity*. κ-vicinity is a node-centered definition for vicinities in DTN. A node's κ-vicinity is the set of neighbors located within κ-hops from it. Note that we use connectivity graphs to derive our notion of vicinity. We also defined *κ-contact* and *κ-intercontact* notions to fit the pairwise relationships between two nodes and analyze their temporal distributions. We also analyze the internal topological properties of κ-vicinities to understand neighbors disposition. Finally, we observe the improvements of vicinity annexation in a simple forwarding protocol (Chapter 3).

Our main contributions with this regard are as follows:

- We identify the binary assertion issue in opportunistic networks where most opportunistic networks approaches consider only nodes in contact and neglect other nearby nodes.
- We defined close vicinity in DTN with the "κ-vicinity" and the related connectivity notions of "κ-contact" and "κ-intercontact".

- We show how κ-contact intervals distributions depend on node-centered density with behavior changing whenever the density is "sparse" or "dense".
- We assess the tradeoff between additional vicinity knowledge and monitoring costs and conclude that a κ between 3 and 4 is enough for the type of datasets we consider.
- By using the 2-vicinity instead of the usual contact vision in opportunistic networks, we can improve by 80% a given opportunistic performance metric.

Contribution 2: Digging into the Vicinity Dynamics of Mobile Opportunistic Networks. We analyze the internal κ-vicinity dynamics of opportunistic networks. We chose to model the node-centered vicinity behavior with a chain process using exact pairwise distances values as states. We call this model asynchronous vicinity motion. We observe the transitional probabilities between any given states to understand vicinity movements. We identify two different chain types and three main movement patterns namely *birth*, *death* and *sequential* moves in κ-vicinities. We observe their repartition in different datasets and how they can be leveraged in opportunistic networks. We analyze vicinity behaviors using *timelines* which is the sequence of shortest distances between pairs of nodes in the network. To ease the analysis of vicinities in various scenarios, we implemented a Python module to automatically perform all the aforementioned vicinity dynamics analyses. After extracting vicinity movements characteristics with asynchronous vicinity motion, we manage to recreate synthetic timelines exhibiting chosen datasets properties using the TiGeR generator (Chapter 4).

In this part, we make the following contributions:

- We conceive asynchronous vicinity motion, a model for internal dynamics in κ-vicinities.
- We identify three main vicinity motion patterns "birth, death, and sequential" movements and some of their properties: death rates remain stable independently of states, birth patterns are alike independently of the observed datasets and by considering death and sequential movements, for some states they represent more than 80% of all outgoing movements.
- We propose a pairwise vicinity behavior generator and its open access implementation included in the Vicinity package we provide.[3]

Contribution 3: Predicting Vicinity Dynamics. In our latest contribution, we investigate how predictability applies in the κ-vicinity and how accurately vicinity motion

[3]More information in the Vicinity package here: http://vicinity.lip6.fr.

captures the network operation. We create a heuristic based on transitional probabilities of vicinity motion to predict a pair of preferential pairwise distances between nodes for the future steps. In asynchronous vicinity motion, steps have various durations. Since our heuristic predicts pairwise distances for future steps, having constant duration steps may bring a more powerful forecasting. Therefore, we present the synchronous vicinity motion, a time-aware variant of vicinity motion. Synchronous vicinity motion samples its surroundings every τ seconds. A step is now of constant duration τ seconds here. We analyze the performances of asynchronous and synchronous vicinity motion (AVM and SVM) heuristics and show their prediction power in terms of pairwise distances not only for the next considered interval but also intervals further away in time (Chapter 5).

Our main findings are:

- The definition of a time-aware vicinity motion called synchronous vicinity motion.
- The possibility of inferring future κ-vicinity distances using Markov chains transient state analysis with our heuristic that can predict pairwise distances up to a 99% accuracy with the SVM model and 40% with AVM.
- A comparison of full duration and partial knowledge forecasting performance showing how vicinity motion is able to sensibly capture network behavior even with shorter sensing durations.

As a final note, it is important to underline that the observations we make depend of the datasets we use. We believe that similar observations can be made in other settings where mobility is influenced by human behavior. To enable such verification, we provide an implementation of our main analyses in the Python Vicinity package available at the following address: http://vicinity.lip6.fr.

Chapter 2

Related Work and Datasets

Contents

2.1 Related Work

As the notion of vicinity is quite new for opportunistic networks, we position our work according to different points of view.

2.1.1 Contact and intercontact vision

Since MANET and wired networks, our community used contacts between nodes to understand any transmission possibilities. Hence, understanding DTN based on network contacts is the most instinctive characterization. In DTN, we consider that a contact occurs when two nodes are within each other's wireless communication range. In our work, we consider that links are bidirectional. Vahdat and Becker studied the impact of wireless ranges on message delivery [8]. Chaintreau et al. used contact history to derive efficient transmission possibilities [9]. Yoneki et al. focused on the significance of meeting times in contact neworks and exposed how important network contact structures are [10]. Hui et al. investigated contact patterns to infer people's affinities and likeliness of meeting [11, 12]. Hossmann et al. analyzed contacts considering contact graphs abstraction with both aggregated durations and contact frequency [13]. Contact

is important in DTN as it allows direct transmission between nodes. But the original parameter in DTN is the intercontact notion. In MANET and wired networks, a connectivity disruption used to be a fault, now; it is part of the regular network functioning.

The intercontact notion indicates the state between two consecutive contacts. There are two main definitions for intercontact. The inter-any-contact notion meaning the time between two subsequent contacts independently of the identities of the neighbors. The second definition – the pairwise intercontact time – treats a specific pair of nodes. It describes the time between two sequential contacts for a given pair of nodes. Understanding intercontact distributions help us see when nodes will next be able to transmit data to other devices. Leguay et al. performed a thorough pairwise intercontact distributions analysis for three experimental datasets. They found that distributions fitted either log-normal or exponential laws [14]. Chaintreau et al. showed that pairwise intercontact times followed power law distributions over a specific time range [15]. In another study, Karagiannis et al. found that pairwise intercontacts fit diptych distributions – power law followed by exponential decay [16]. Passarella et al. exposed several intercontact properties of opportunistic networks and they assessed that aggregate intercontact times were not the same as pairwise intercontact times [17, 18]. So far, we are only aware of one initiative concerning an implementation of a connectivity generator handling contact intervals properties [19].

Contact and intercontact understanding is crucial in DTN and has been well investigated. However, we must note that the way we sample the network greatly influences the quality of intercontact observations we observe [20]. These probing issues are very important since they may change what we sense in a network. Anyway, none of these visions considered nodes beyond simple contact. This binary notion may not be enough to really benefit from the DTN paradigm.

2.1.2 DTN characterization and end-to-end connectivity usage

Beyond contact and intercontact, we observed different attempts to characterize DTN behavior on the whole. We have seen inspiring intents on linking DTN behavior with network density. Borel et al. produced a theoretical classification for wireless and mobile networks based on their behavior and composition [21]. Similarly, Whitbeck et al. connected the previous classification to nodes mobility and density considerations [22]. In his latest work, Heimlicher proved the importance of connected components in mobile wireless networks [23]. A temporal vision can also be applied to understand DTN. Chaintreau et al provided the first evidence of the small world phenomenon in opportunistic networks [9]. Scellato et al. also managed to exhibit different DTN temporal properties using time series [24]. Other researchers like Casteigts et al. or Whitbeck et al. decided to pursue DTN connectivity understanding using time-varying and tempo-

ral reachability graphs. These graphs are very helpful in order to recognize temporal patterns [25, 26]. Panisson et al. confronted both temporal and topological opportunistic network characterization and showed the importance of both properties in DTN understanding [27].

In DTN, researchers have found various ways to leverage a node's neighborhood. Some techniques choose to use the social behavior of the participants. As in a city people tend to cluster into communities around different points of interests, Ött et al presented a protocol leveraging end-to-end and multi-hop DTN paths [28]. End-to-end paths occur among connected components whereas DTN ones happen between these temporary components. Sarafijanovic-Djukic et al. made a similar observation in the VANET environment [4]. Later, Heimlicher and Salamatian laid their study over the groundwork that mobile wireless networks tend to have connected crowds [6, 23]. The main punch line for all these studies is: for each node, there are immediate neighborhood structures to use. Other analyses preferred another point of view towards the vicinity in DTN. Instead of considering a node's instantaneous vicinity, they considered a node's "reachable" vicinity in terms of nodes we can reach during a given time window. Chaintreau et al. analyzed spatio-temporal clusters diameter in a network [9]. Tang et al. focused on the nature of these spatio-temporal paths to better understand how to use them [29]. Whitbeck et al. proposed an interesting way to capture a node's reachable vicinity through a new graph type [26]. Similar principles have also been considered even in other contexts, such as wireless mesh networks [30].

The closest work to ours concerning end-to-end connectivity utilization has been done by Gao et al. [7, 31]. Their notion of transient contacts reckons the κ-contact notion developed in Chapter 3, Definition 2. However, their transient contact notion occurs over a certain period of time while κ-contact is an immediate notion translating the actual instantaneous end-to-end connectivity possibilities.

2.1.3 Routing techniques

To benefit from opportunistic network resulting properties, researchers designed several approaches relying on various parameters. The most naive approaches used opportunistic flooding to achieve highest delivery ratio but to the cost of redundant messaging [8]. This technique was closely followed by attempts of using localized and copy-aware routing approaches. Spray-and-wait and its sister protocols maintain a fixed number of copies to be sent around the network [32, 33]. They localize their flooding by spraying their copies to contacts having a high likeliness of meeting the copy destination.

In order to use upcoming contacts between nodes, Lindgren et al. bet on a probabilistic likeliness-of-meeting measure based on nodes contact history in the PROPHET forwarding protocol [34]. Instead of randomly distributing their content to any nodes coming in contact, nodes compute transitive probabilities of meeting the destination

and only forward it to nodes with high destination meeting values. Nelson et al. also showed that monitoring contact is really helpful in predicting future encounters and achieving interesting delivery rates [35]. The intuition of using nodes encounter history is also present in routing protocols based on social measures.

Human-driven opportunistic networks do not have random spatial movements or meeting patterns. Therefore, using social measures to characterize nodes behavior in a dataset to influence forwarding decision looks promising. In BUBBLE Rap, Hui et al. analyzed networks contact properties and extracted different existing communities. Based on detected communities, BUBBLE Rap computes centrality measures to determine the next-hop to forward to [11]. Bigwood et al. analyzed the use of two types of social networks "self-reported" and "detected" to determine that they do not impact the delivery ratio of resulting DTN protocol [36]. The PeopleRank approach by Mtibaa et al. ranked network nodes using weighted social measures to forward data and exhibited good performances [37]. In their work, Gaito et al. exposed how short contacts deemed unworthy of interest at the beginning are, in fact, social indicators and enable effective routing opportunities [38, 39]. Li et al. considered participants social unwillingness to cooperate into forwarding with the SSAR protocol [40]. Zhu et al. recapitulate most approaches in their latest survey concerning social-based routing in DTN [41].

Yet, none of the aforementioned opportunistic strategies really leverage a node's current vicinity to specifically direct their transmissions. In an interesting work, Vojnovic and Proutiere showed how a simple and lazy hop-limited flooding protocol achieves almost optimal performances [42]. In their latest paper, Diana et al. applied a similar vicinity notion to satellite communications [43]. By leveraging neighboring stations, their proposal allowed valuable routing performance gains. This encourages the utilization of κ-vicinity knowledge (our hop-based vicinity definition) in opportunistic routing protocols. Another interesting approach is to mix different network paradigm in order to get the best of different worlds like wireless sensor networks and challenged satellite networks [44].

2.1.4 Mobility models

In order to understand and characterize mobile opportunistic networks, our community relied on contact traces collected during real-life measurements. However, due to the lack of extensive realistic traces, researchers started creating synthetic mobility models and generated the corresponding contact traces. By creating different mobility models, they emulated human behaviors to test protocols they designed in specific settings [45, 46, 47, 48, 49]. Other studies moved towards accurate connectivity rebuilding based on real-life contact traces [50]. All these analyses removed the correlation between nodes behavior and their surrounding. With the vicinity motion patterns, we try to bridge the gap between these points. Now, more than understanding the whole network

movements, we can see the patterns followed in a node's vicinity.

There is also an intuitive discrimination for DTN characterization between day and night time behaviors. DTN we consider in our studies are reflection of human social behaviors. The aforementioned distributions are different according to the time of the week or the day one may consider. Gaito et al. showed that patterns during working time is different from night time behaviors [38]. Ekman et al. produced a specific mobility model exhibiting these specific differences [51]. Considering vehicular mobility models and the corresponding contact traces would also be very interesting in our case, yet it is beyond the scope of this thesis [52].

An effort has already been made to directly generate contact traces based on contact and intercontact analyses [19]. Yet, it lacked the vicinity notion we bring with the vicinity motion patterns. With TiGeR, our pairwise vicinity behavior generator, we provide a mean to obtain realistic vicinity behavior for pairs of nodes through timelines. Timelines indicate time intervals with the actual shortest distance (up to distance κ) between two nodes. Timelines include contact periods (at a 1-hop distance), therefore we can extract contact intervals for pairs of nodes.

2.1.5 Prediction in DTNs

The prediction aspect in opportunistic networks is one of the most promising research direction. Apart from contact and intercontact patterns, if we are able to foretell when two nodes will next be close to each other, we can schedule our next transmission and tune our opportunistic decisions accordingly.

The social underlay of human-driven opportunistic network is key in predicting future behavior. Many studies analyzed this social nature in order to understand how nodes come to meet each other [53, 54, 55, 56]. Following the human nature and their tendency to have regular moving patterns [3], Boc et al. managed to deduce repetitive behavior on a weekly basis [57]. Scellato et al. managed to use location information to predict link apparition between nodes [58]. The prediction of link appearance is an interesting issue in networking. Liben-Nowell and Kleinberg laid an inspiring study concerning link-prediction in social networks [59]. They tried out various ways of predicting future links using metrics based on node's neighborhood, paths and other "meta-approaches" on quite static datasets. However, their best predictor only achieves 16% correctness on its predictions. This shows how hard and demanding the prediction issue can be even for not highly dynamic datasets. Wang et al. also studied the link-prediction problem but on a mobile phone calls dataset [60]. They achieve interesting performances but their dataset is also quite stable and runs over a long period of time. Predicting in DTNs often occurs with a shorter time span and using sparse node-centered knowledge. Zayani et al. used a tensor-based prediction model to forecast future encounters based on social metrics reflecting localized network stability [61, 62].

They obtained inspiring results in both centralized and distributed deployment.

Song et al. proved the high predictability of human mobility with a 93% potential predictability using another mobile phone call dataset [63]. Lu et al. even achieved a 95% predictability value of user mobility in a similar dataset and exposed practical possibilities of achieving such a high degree of performance with a real algorithm [64]. Predictions in DTNs seem to be an open and appealing subject. Still, most link-prediction studies use quite stable and static datasets that are different from most scenario we observe in DTN. Some studies even question the possibility of really foretelling events in opportunistic networks. Hossmann et al. highlighted the difficulty of using complex network analysis and resulting social measures into DTN analysis [65]. Using the RECAST classifier to screen out nodes with random relationships and keep nodes with real social ties could improve results on this topic [66]. Given the highly dynamic nature of DTN, finding accurate sampling and aggregation methods as well as fitting network metrics is a real issue. Nikolopoulos et al. even question the accuracy of usual social metrics when they are applied on opportunistic networks [67].

Concerning DTN in general, there has been and there currently is a constant effort from our community. We reviewed some very interesting DTN analyses in the previous sections. However, most of these approaches maintain a very simplistic vision of opportunistic networks by neglecting nodes beyond one-hop distance that renders a biased DTN vision that is not totally able to benefit from DTN features.

2.2 Datasets

For this dissertation we confront our analyses to several datasets, both synthetic and real-life based. We provide both the main related study and the downloading links when available for each of the datasets.

2.2.1 Connectivity assumptions

In our datasets, we assumed bidirectional links meaning that when a link is present in synthetic datasets between two nodes, both of them can send data to one another. Most of real-world datasets used devices like the iMote (see Fig. 2.1) to gather experimental data. iMotes embed Bluetooth chipsets that are able to alternatively send or receive presence beacons. In real-world measurements, we deemed that if a node is able to detect another, the reciprocal property is true.

The symmetric vision of links in DTN is questionable due to the reality of transmission technologies yet this assumption is necessary in our work since it renders contact graphs symmetric.

Figure 2.1: iMotes: these devices were used in most of the real-life based measurements. They embed a Bluetooth chip alternatively collecting and sending probes in their vicinity. When an iMote detects a probe, it collects the sender's ID.

2.2.2 Real-world datasets

- **Infocom05** measurement was held during a 5-day conference in 2005 [15, 68]. 41 attendees carried iMotes collecting information about other iMotes nearby within a 10 m wireless range. We study a 12-hour interval bearing the highest networking activity. Each iMote probes its environment every 120 seconds. *Infocom05* represents a professional meeting framework.

- **Sigcomm09/Sig09** counted 76 attendees with dedicated smartphones probing their surroundings during 5 days [69, 70]. Smartphones sensed their surroundings using Bluetooth every 120 seconds. *Sigcomm09* is another example of a professional meeting scene. In our work, we also use the sole first day of the *Sigcomm09* dataset and name it *Sigcomm09-d1* or *Sig09-d1*.

- **Rollernet** had 61 participants measuring their mutual connectivity with iMotes while they where riding a dominical rollerblading tour during 3 hours in Paris [71, 72]. Researchers set devices to send beacons every 15 seconds. These measurements show a specific sport gathering scenario.

- **Unimi** is a dataset captured by students, faculty members, and staff from the University of Milano in 2008 [73, 74]. They involved 48 persons with special devices probing their neighborhood every second. *Unimi* provides a scholar and working environment scenario.

- **Shopping** used 25 dedicated devices in a shopping mall over 6 days [75, 76]. Galati and Greenhalgh gave 25 devices to shop owners and planted 8 others at

Table 2.1: Datasets characteristics.

Dataset	#	*Length*	*Probing*	*Range*	*Type*
Infocom05	41	12h	120s	10m	Conference
Sigcomm09/Sig09	76	4 days	120s	10m	Conference
Rollernet	61	1h30	15s	10m	Sport
Shopping	25	6 days	120s	10m	Mall
Unimi	50	19 days	1s	10m	Scholar
Stanford	200	1h	20s	3m	Scholar
RT (S)	20	9h	1s	10m	Work
Community (S)	50	9h	1s	10m	City

various locations in the mall. Devices performed neighborhood discovery every 120 seconds. *Shopping* reflects the working day routine of shop owners as well as some of their customers.

- **Stanford** has 789 persons in an American high school carrying TelosB motes – detecting contacts up to a 3 m range [77, 78]. Salathé et al. gave Motes to students, teachers and staff members for around two weeks. We focus on a subset of 200 participants. TelosB motes send beacons every 20 seconds. *Stanford* expresses a settings with a majority of teenagers who have a tendency to dwell in groups of interests.

2.2.3 Synthetic datasets

- **RT** uses the Random Trip mobility model which corrects flaws from the Random Waypoint model [79, 45, 80]. We sampled the behavior of 20 nodes following this model on a surface of 50 x 60 m^2 with speed between 0 and 7 m/s and a 10 m range.

- **Community** is a social-based mobility model [47, 81]. It tends to colocate socially-related nodes in specific locations at the same time like groups of friends would do. We simulated 50 nodes with a 10 m wireless range on a 1,500 x 2,500 m^2 plane during 9 hours.

We recapitulate the main datasets characteristics in Table 2.1. # indicates the number of nodes in the datasets, *Length* indicates its duration, *Probing* presents the probing intervals, *Range* provides the probing devices wireless range, and *Type* presents the scenario type. In the following, we use some of the presented datasets to introduce and illustrate our vicinity concepts.

Chapter 3

Uncovering Vicinity Properties of Intercontacts in DTNs

Contents

As a descendent of historical networks where the notion of contacts and direct connectivity is prominent, the first characterization of opportunistic networks focused on contacts between nodes and their resulting intercontact periods. The notion of contact has been well investigated years before [8, 10, 11]. However, the intercontact notion is quite unexplored. The first sensible approach was to consider intercontact as the complementary of contact. This assumption was maintained in the context of a number of interesting studies but it may be too shallow to correctly reflect the underlying network topology.

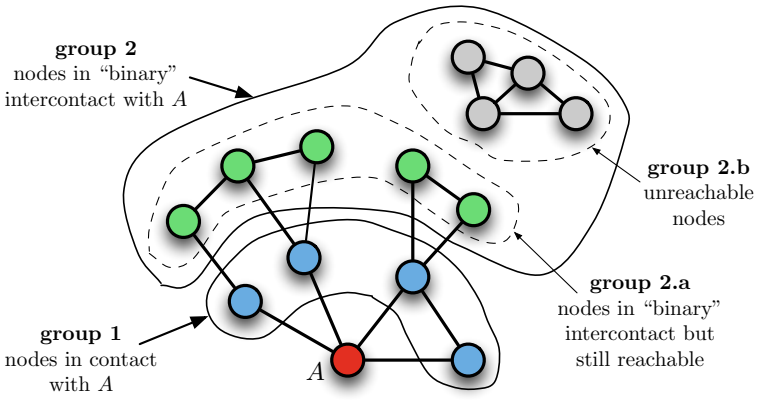
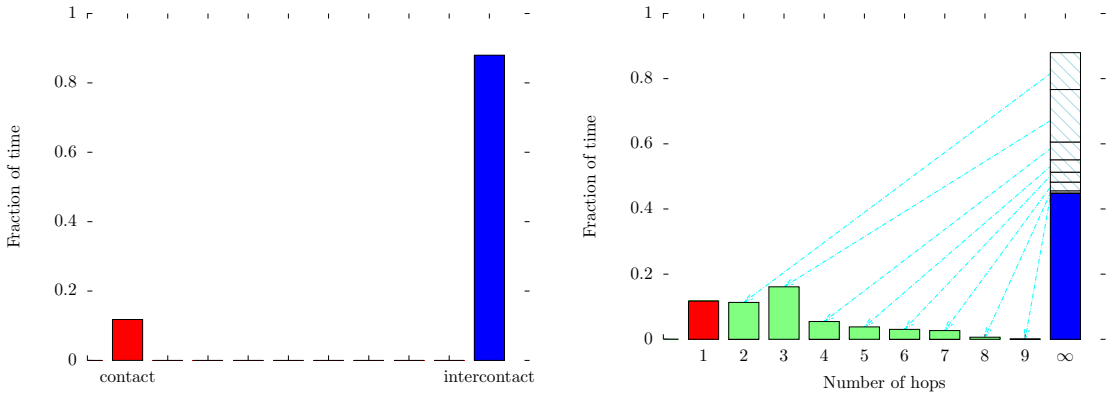


Figure 3.1: Motivating example. From node A's point of view, we see that nodes in "group 1" are in contact. Using the usual binary vision, we conclude that all other nodes are in intercontact (i.e., in "group 2"). However, nodes in "group 2.a" are essentially different from nodes in "group 2.b". A has end-to-end paths toward the first and no paths at all to the latter.

In Fig. 3.1, we represent a network snapshot illustrating our concerns. This figure represents a network where nodes in "group 1" are in contact with A (i.e., they are within A's direct communication range). In the usual *binary vision*, all remaining nodes (group 2) are, by definition, in intercontact. Still, we notice that there is a fundamental difference among nodes in "group 2". None of the nodes in "group 2.a" are in contact with A; nevertheless, they do have a contemporaneous path to A. On the other hand, nodes in "group 2.b" do not have any path to A. In opportunistic networking where we need to gather as much knowledge as possible to achieve efficient communication standards, deeming both cases of intercontact under the same definition results in a waste of information. Suppose A needs to send a message to one of the nodes in group 2.a. In such a situation, most DTN approaches infer the impossibility of exchanging messages via multi-hop paths and often calls for a "wait" period until it meets the destination or find someone else that knows the destination better (based on some other criterion). With the binary vision, A does not know that the destination is nearby, and may miss an opportunity to communicate if, for example, the destination moves after some time to group 2.b. By denying the inherent ad hoc network part in DTN, we cannot pull the best of both worlds. Noticing that contemporaneous paths may exist between nodes is important. Neglecting such closeby possibilities is a waste of connectivity assets in DTN.

In the next sections, we highlight the "binary assertion issue", which is the base of our questioning. We propose a formal definition of vicinity for DTN with the κ-vicinity and the related notions of κ-contact and κ-intercontact in opportunistic networks and analyze their distributions using datasets described in Chapter 2. Then, we study the κ-vicinity itself by analyzing its composition. Finally, we expose the advantages of vicinity utilization in a simple opportunistic protocol.



(a) Binary assertion

(b) Extended vision

Figure 3.2: Example of time-distance distribution from the *RT* dataset. In Fig. 3.2(a), nodes spend 10% the time in contact (1-hop). With the binary vision, we then consider that nodes spend around 90% of their time in intercontact. Fig. 3.2(b) shows that in reality, they dwell at a distance 2 for around 10%, at a distance 3 for 16%. Real intercontact deprived of multi-hop path represents only 50% of the time (∞).

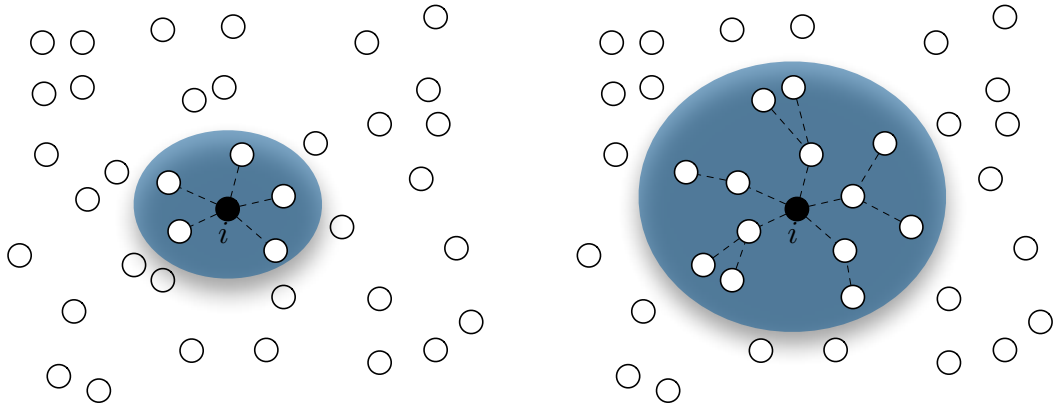
3.1 The Binary Assertion Issue

Considering the notion of intercontact as the mere binary complementary vision of contact is understandable. The leading property of historical networks has always been the "contact" between nodes. But in challenged networks such as DTNs, we have to get the most of every situation and surrounding assets. In Fig. 3.1, we observed that there were unused pairwise connectivity between nodes. The traditional contact vision misuses end-to-end "not-in-contact" connectivity. The *binary assertion issue*, where we ignore end-to-end connectivity beyond one hop, brings an interesting interrogation:

how pervasive are these hidden communication possibilities?

To understand the problem, let us show an example for a given pair of nodes using the *RT* dataset. We compare the amount of time they spend in contact and in intercontact and plot the results in Fig. 3.2. In Fig. 3.2(a), we observe that nodes spend around 10% of their time in contact and around 90% in binary intercontact. If we consider the extended vision (see Fig. 3.2(b)), for the same pair of node, we realize that they spend around 10% of their time at a 2-hop distance, 18% at 3-hops, 5% at a 4-hop distance, etc. The true time they spend without any path to one another is only around 50% of the experiment duration. The binary assertion hides 40% of the time where these two nodes have a path connecting them.

By simplifying our network vision, the binary assertion reduces the information we get from our surroundings. More than just limiting our vision, it prevents us from



(a) 1-vicinity (b) 2-vicinity

Figure 3.3: κ-vicinity illustration. Node i's $\{1, 2\}$-vicinity at a given time t.

leveraging our environments and performing simple yet efficient closeby end-to-end transmissions. To be able to use and characterize the closeby topological paths, we first define the notion of vicinity in DTN.

3.2 The Notion of Vicinity

To the best of our knowledge, this is the first time the notion of vicinity has ever been formalized in DTN. To understand the extended transmission possibilities in opportunistic networks, the first issue is to provide a formal definition of what the notion of "nearness" means in DTNs.

3.2.1 Vicinity definition for opportunistic networks

To formalize the vicinity notion in DTN, we choose to use a node-centered point of view. The κ-vicinity notion brings an ego definition to DTNs and also adds a hop-based discrimination [82, 83, 84]. This differentiation helps us limit our vision according to our needs as well as identifying neighbor properties. We discriminate a node i's vicinity according to the number of hops between i and its surrounding neighbors. We use the instantaneous connectivity graph between nodes to compute pairwise shortest paths. This connectivity graph illustrates the current network state and what is immediately useable.

Definition 1 ***κ-vicinity.*** *The κ-vicinity $\mathcal{V}_\kappa^i$ of node i is the set of all nodes with shortest paths of length at most κ hops from i.*

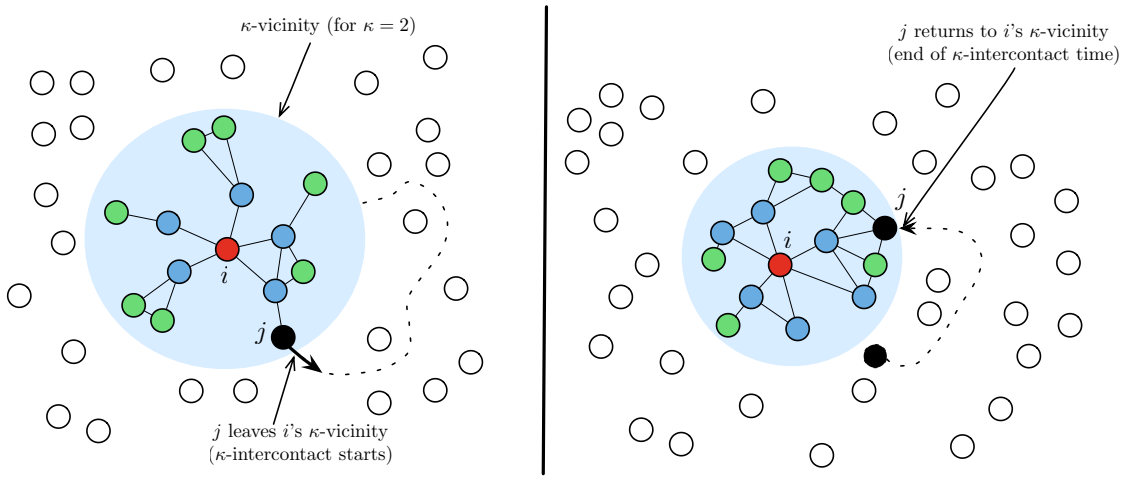


Figure 3.4: Node i's κ-vicinity and the κ-intercontact phenomenon. For the sake of clarity, we only display i's connectivity links within the κ-vicinity.

Clearly, $\mathcal{V}^i_{\kappa-1} \subset \mathcal{V}^i_\kappa$. In Fig. 3.3, we illustrate the node i's 1-vicinity and 2-vicinity at instant t. This is an interesting point of view for opportunistic networks because it extends a node's knowledge to immediately useable communication opportunities. The κ-vicinity empowers a node's reach in the network [85].

Vicinity knowledge may come from different techniques. For instance, we can use link state protocols to gather information about a node's connected component. There are many ways to do so, but they all are costlier than getting information from contacts only. The tradeoff between getting vicinity information and its additional costs may be a reason not to use κ-vicinity. However, we provide a solution to this tradeoff in Sections 3.4 and 3.5 by suggesting that monitoring the $\{3, 4\}$-vicinity is enough to get most events in a node's surroundings for the datasets we consider in our analyses.

The κ-vicinity defines a node's neighborhood, i.e. its new *zone* in the network. To characterize this zone's relationships to node i like the *contact* and *intercontact* notions previously, we must define some temporal measures relating to time neighbors spend in the zone and time outside the zone, namely "κ-contact" and "κ-intercontact". We maintain a pairwise definition for these measures and assume that connectivity is bidirectional.

Definition 2 ***κ-contact.*** *Two nodes are in κ-contact when they dwell within each other's κ-vicinity, with $\kappa \in \mathbb{N}^*$. More formally, two nodes i and j are in κ-contact when $\{i \in \mathcal{V}^j_\kappa\} = \{j \in \mathcal{V}^i_\kappa\}$. In other words, a contemporaneous path of length at most κ hops i and j.*

We also need to grasp the intercontact observations for our vicinity viewpoint. The literature definition of mere intercontact is when two nodes are not in contact. There-

fore, we consider κ-intercontact when two nodes are not in κ-contact. These are complementary notions. Another way to see it is as follows: if node i maintains knowledge about its κ-vicinity, it is in κ-intercontact with any node beyond its κ-vicinity. In Fig. 3.4, node j leaves i's κ-vicinity and then gets back some time later, characterizing a κ-intercontact interval.

Definition 3 ***κ-intercontact.*** *Two nodes are in κ-intercontact while they do not belong to each other's κ-vicinity. Formally speaking, two nodes i and j are in κ-intercontact when $\{i \notin \mathcal{V}_\kappa^j\}$ or $\{j \notin \mathcal{V}_\kappa^i\}$ or there is no path of length κ or less linking i and j.*

Note that 1-contact matches the contact notion and 1-intercontact corresponds to usual binary intercontact. Our zone point of view integrates previous binary network vision and extends it via nearby nodes. For dense networks, maintaining the binary vision may result in a great loss of closeby transmission possibilities.

3.2.2 Missed transmission possibilities with binary assertion

To quantify how many end-to-end transmission opportunities the binary assertion misses, we present what we call aggregated network sociostructures in Fig. 3.5 [84]. For each real life-based dataset, we plotted (in layered mode) the number of connected pairs for each shortest distance. The bottom layer symbolizes the amount of pair of nodes in contact. Layer 2 shows the amount of pairs connected via a 2-hop path, layer 3 represents connection via a 3-hop path, and so on. Each sociostructure layer of value ≥ 2 represents pair of nodes linked by end-to-end paths longer than 1 hop. Recall that the binary assertion does not recognize such relations.

In Fig. 3.5(a), for *Infocom05*, we observe several density peaks of connected pairs. Being a conference-based measurement, these peaks indicate morning arrivals, lunch, the afternoon break, and end of sessions. During high density peaks, an unexpected observation is how pairs connected by 2 hops overcome contact opportunities. Places with high density ignite transmission possibilities beyond mere contact. For *Sigcomm09*'s sociostructure, we focus in the first density peak of this conference dataset (see Fig. 3.5(b)). The number of pairs in contact remains non null during the observation. At some point, see 10,000 seconds, pairs linked by two or more (2^+) hops represent more than four times the number of pairs currently in contact. As a result, in such a scenario, 2^+*-hop transmissions should be more helpful than mere contact transmissions* or pure DTN techniques.

In Fig. 3.5(c), we witness *Rollernet*'s accordion phenomenon, i.e., the stretching and shrinking of the crowd due to urban obstacles preventing the crowd from moving forward [71]. *Rollernet* has a dynamic setting with a compulsory path. Nodes do not

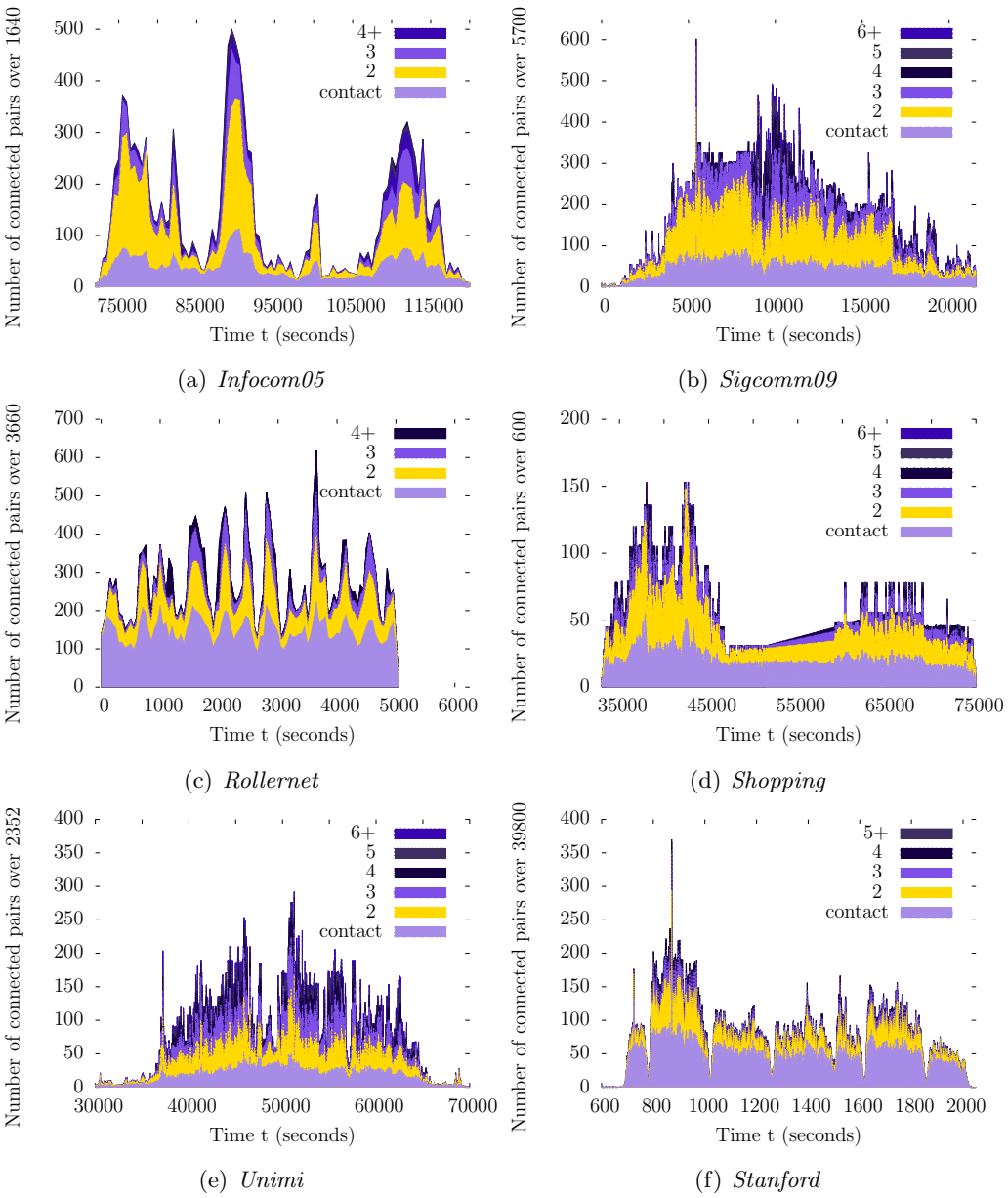


Figure 3.5: Datasets sociostructures presenting the amount of pairs connected by contacts, 2-hop paths, 3-hop and so on in a layered mode according to time. We notice the omnipresence of pairs connected by 2^+ hops. They often overcome the possibilities offered by contact only (bottom layer). As a result, contact opportunities only represent a minor part of all end-to-end opportunities between two nodes. The binary assertion overlooks these possibilities by blending all nodes in intercontact under a unique concept.

have as much movement liberty as they have in other datasets. Contacts are prominent in *Rollernet*, still, we observe many pairs of nodes connected by 2^+-hops.

For *Shopping* and *Unimi* (Fig. 3.5(d) and Fig. 3.5(e)), we present their first day sociostructure as this pattern repeats itself every other day. Once more, we notice the omnipresence of pairs connected at a 2-hop distance and how they may overcome contact opportunities. In the *Unimi* dataset, we notice even more pairs connected by 3^+-hops. They all display a social density igniting linking nodes beyond simple contact.

The *Stanford* sociostructure from Fig. 3.5(f) occurs during a school day and shows the different groups found in high school. The majority of students stay close enough to be connected via contacts, still, we can observe the omnipresence of non marginal 2^+-hop links between pairs.

The presented sociostructures illustrate the illusion provided by the binary assertion. If we maintain contact-only knowledge in DTN, we miss the omnipresent power of nodes at 2^+ hops. For all datasets, there are almost always non-null 2^+ layers. These 2^+ layers represent powerful transmission opportunities as they only involve few relays that could reduce significantly end-to-end delays. These layers vary in importance but are almost always present. Considering only contacts provides a minor vision of what happens in the network. Observing a node's vicinity at a 2-hop distance may more than double the transmission opportunities as seen in the *Infocom05* dataset at 90,000 seconds. The binary assertion weakness highlights the importance of observing nodes beyond simple contact. In social settings, there may be a concentration of people around us (when commuting or at work), yet we limit our vision to contacts only while there is so much more at hand.

3.2.3 Pairwise behavior variability

We observe that a large portion of nodes display a significant fraction of time with end-to-end transmission capacities endorsed by contact and κ-contact. Nodes bearing end-to-end transmissions beyond contact for more than 10 minutes are as follows: 78.3% for *Infocom05*, 99.4% for *Rollernet*, 57.1% for *Sigcomm09*, 57.0% for *Unimi*, 100% for *RT*, and 73.7% for *Shopping*. If we increase the threshold to 20 minutes, the proportion of pair of nodes with extended end-to-end properties do not really change for the following datasets: 78.3% for *Infocom05*, 98.2% for *Rollernet*, 53.4% for *Unimi*, and 100% for *RT*. The values for *Sigcomm09* and *Shopping* decrease to respectively 44.8% and 57.7% but still remain quite high.

In Fig. 3.6, for all pairs of nodes, we plot on the x-axis the fraction of time they spend in contact and on the y-axis the fraction of time they spend in 2^+-contact. We visualize a wide variety of meeting patterns with many pairs having long 2^+-contacts. Previous studies used contact observations to derive DTN strategies. This involves a first granularity for pairwise behavior characterization. Beyond contact, nodes embed

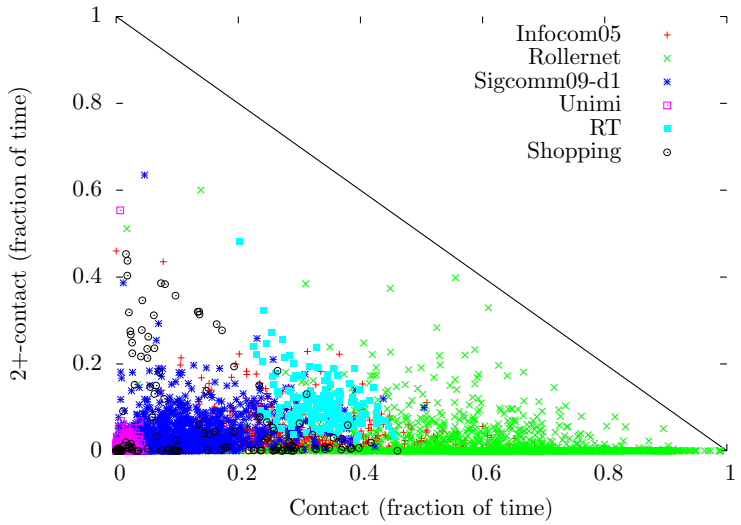


Figure 3.6: Pairwise behavior according to the fraction of contacts and 2^+-contacts. Each dot represents a pair of nodes. On the x-axis, we have the fraction of time they dwell in contact. On the y-axis, the fraction of time they observe κ-contacts. Note that, for the same density of contact, we can obtain a wide difference in 2^+-contact percentage.

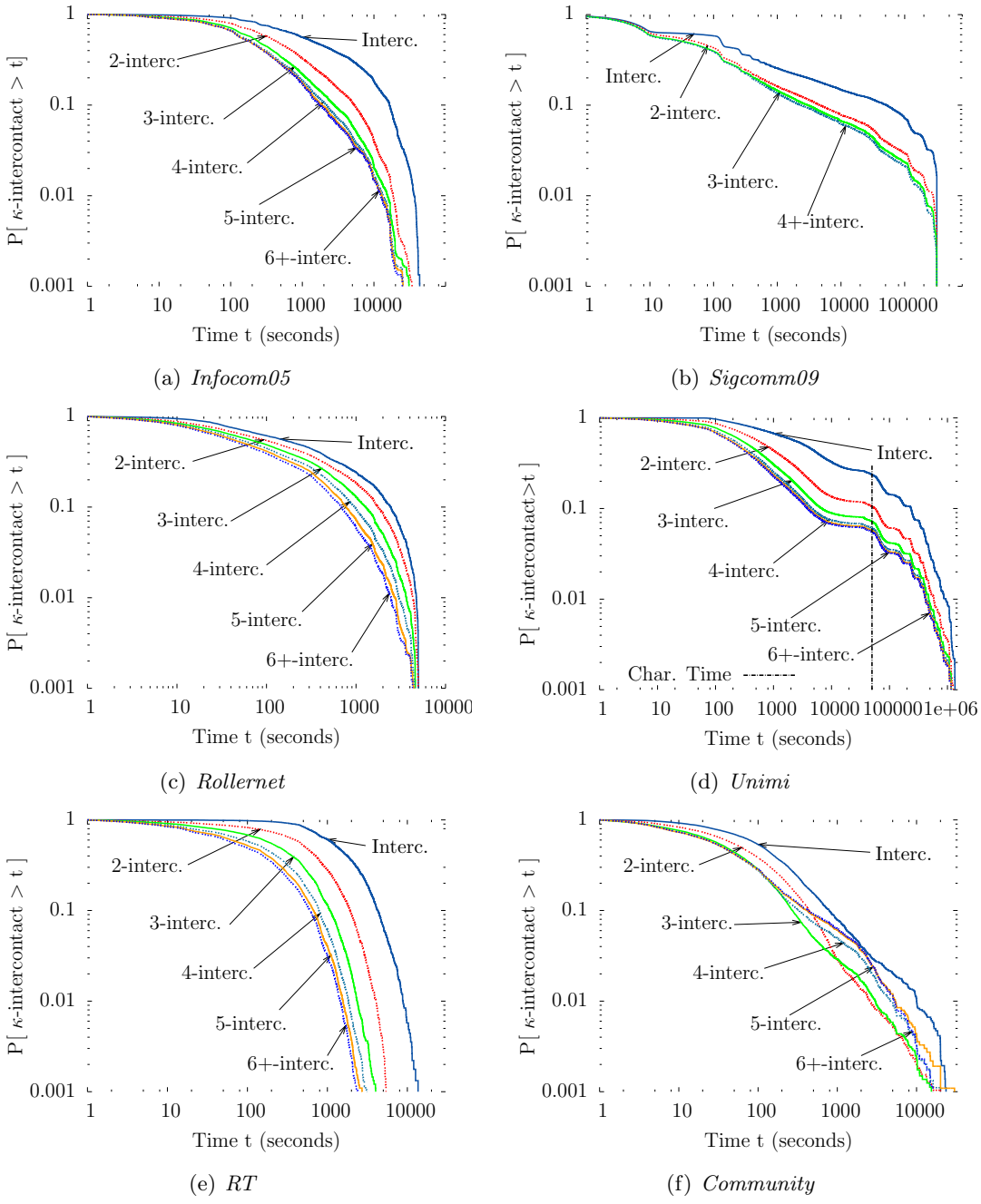
significant ad hoc properties and they are not alike. DTN characterization needs a finer granularity for its intercontact patterns; otherwise it misses promising ad hoc properties.

3.3 Temporal κ-vicinity Characterization

3.3.1 κ-intercontact distributions

Quantifying intercontact durations distributions helps us understand how long a node will have to wait before its next encounter. In Fig. 3.7, we present the aggregated average complementary cumulative density function (CCDF) κ-intercontact durations for every pair of nodes ($\kappa \in \mathbb{N}^*$). These CCDFs indicate the probability of a κ-intercontact lasting longer than t seconds. For the sake of clarity, we do not display *Stanford* and *Shopping*'s CCDFs. They display similar behaviors unless specified.

Binary intercontact. The binary intercontact is the 1-intercontact, meaning the dual of the usual contact notion. Karagiannis et al. observed that for most of the datasets they analyzed, aggregate intercontact bear distributions with a power law up to a knee point followed by an exponential decay [16]. We also find that for all our datasets, their binary intercontact CCDFs, denoted "Interc." in Fig. 3.7, follow a straight line up to a knee point when both x-axis and y-axis are on logarithmic scale. This implies power laws for each binary intercontact distribution until the observed knee point also known as the characteristic time. The phenomenon is particularly clear for *Unimi*. We observe a knee



(a) *Infocom05* (b) *Sigcomm09*

(c) *Rollernet* (d) *Unimi*

(e) *RT* (f) *Community*

Figure 3.7: κ-intercontact distributions. Except for *Community*, nodes display a lower probability of obtaining κ-intercontact intervals lasting longer than t seconds for high κ. On average, κ-vicinity reduces κ-intercontact times. Distributions follow power laws up to a characteristic time and then display exponential decay. All κ-intercontact distributions knee point concord. *Community* has inconsistent κ-vicinity patterns for $\kappa \geq 3$. Interc. stands for Binary intercontact (logscale on both axes).

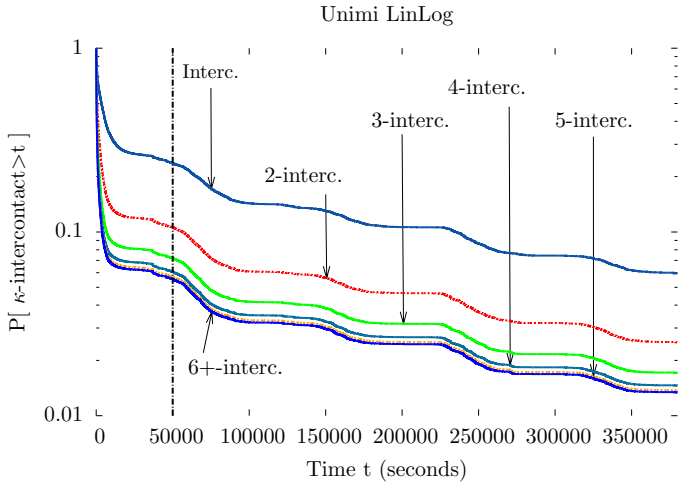


Figure 3.8: The *Unimi* κ-intercontact distributions with a linear x-axis.

point for binary intercontact at around 50,000 seconds. After plotting distributions with a linear scale on the x-axis and maintaining the log scale on the y-axis, we also observe that distributions can be approximated by a straight line beyond the knee point (see Fig. 3.8). This hints exponential decays for distribution tails. Observations on binary intercontact match results of previous studies.

κ-intercontact distributions. Concerning κ distributions for $\kappa \geq 2$, we find their general appearance to be quite similar to their respective binary intercontact distributions except for *Community*. All κ-intercontacts' lines have a partial shift after a given point with a sharper slope for each curve. For larger values of κ, we notice a bigger bottom left shift for each distribution. κ-vicinity logically reduces κ-intercontact times. The wider the node's vicinity knowledge, the later this node detects a node's departure from its vicinity and the quicker it detects its comeback. This results in shorter κ-intercontact durations. We see that, for $\kappa \geq 6$, CCDFs aggregate. For *Sigcomm09* this phenomenon even occurs for $\kappa \geq 4$.

An interesting remark is how κ-intercontact distributions exhibit the same properties as simple intercontact. κ-intercontact curves follow power laws until a specific point (the characteristic time) and then carry exponential decay. In Fig. 3.7(d), beyond 50,000 seconds, *Unimi*'s 2-intercontact curve is a vertical shift of the binary intercontact CCDF. The same occurs for further κ-intercontacts. However, an important information is that the knee point found for binary intercontact corresponds to changing points for κ-intercontact distributions. In *Unimi*, κ-intercontact curves ($\kappa \geq 2$) quickly decrease after the characteristic time found at 50,000 seconds.

Table 3.1 displays average intercontact durations and Table 3.2 the number of inter-

Table 3.1: κ-intercontact average duration in seconds.

	κ					
Dataset	1	2	3	4	5	6+
Infocom05	4,931	1,752	1,111	917	850	823
Sigcomm09	19,780	8,340	6,608	5,981	5,689	5,552
Rollernet	738	555	412	328	274	242
Unimi	66,435	28,688	19,529	16,585	15,535	15,110
RT	1,874	772	416	291	238	213
Stanford	1,359	1,145	1,093	1,081	1,076	1,074
Shopping	2,608	1,205	885	828	824	824
Community	525	232	193	262	317	296

Table 3.2: κ-intercontact number of intervals ($\times$ 1,000).

	κ					
Dataset	1	2	3	4	5	6+
Infocom05	3.7	11.0	15.4	16.8	17.2	17.1
Sigcomm09	45.8	107.7	135.0	148.7	156.1	159.9
Rollernet	2.5	7.5	12.4	15.8	17.6	18.4
Unimi	21.7	57.1	86.0	102.5	109.4	112.3
RT	2.3	5.0	8.3	10.5	11.9	12.6
Stanford	31.5	37.4	39.1	39.6	39.8	39.8
Shopping	52.6	109.1	145.0	153.6	154.1	154.0
Community	3.6	11.6	8.0	4.5	3.7	3.5

contact intervals for each dataset. Except for *Community*, the larger the κ, the lower the average κ-intercontact length. This enforces our rational expectations of κ-vicinity reducing κ-intercontact duration with longer κ. We notice decreasing cumulated κ-intercontact times for each κ. We also observe a logarithmic growth in the number of κ-intercontact intervals.

Remarks. The *Community* dataset stands out because of its non-monotonic average κ-intercontact duration and evolution of the number of intervals. When the average length grows, the number of κ-intercontact intervals decreases. This still results in a decreasing cumulated κ-intercontact duration for each κ. It enforces our first thoughts in the benefits of κ-vicinity for κ-intercontact times. Under the assumption that nodes in the vicinity dwell within low delay reach, κ-intercontact duration decreases with larger κ, strengthening our belief that κ-vicinity is beneficial to DTN protocols. The fact that characteristic time in all intercontact distributions corresponds is also an important finding. It could help protocols like WAIT also know as Direct Transmis-

sion [86], PROPHET [34], or Spray-and-wait [32] maintain their actual intercontact-based approach and extend them to their vicinity to benefit from shorter κ-intercontact times.

3.3.2 κ-contact distributions

Contact opportunities are the main feature for opportunistic mobile networks. Analyzing its distribution gives us insights on how protocols can benefit from these contact opportunities. Instead of considering contact between neighbors at a 1-hop distance only, we analyzed the potential of transmission to nearby nodes within the κ-vicinity. These paths enable low delay transmissions and a better neighborhood reach for a network node.

κ-contact distributions. In Fig. 3.9, we display aggregated CCDFs of contact alongside κ-contact duration for every pair of nodes in each experiment. These CCDFs indicate the probability of a κ-contact lasting longer than t seconds.

For *Infocom05* and *Rollernet*, in Fig. 3.9(a) and 3.9(c), their CCDFs maintain comparable aspects. We observe a small upper right shift for larger values of κ. As the κ-contact notion increases the node's vicinity scope, any nearby node may stay within the considered node's coverage longer than with a shorter vision. The longer the κ, the higher the probability of having longer κ-contact durations. Besides scanning granularity, smaller values of κ results in curves with a sharper slope than curves of longer κ-contacts.

Like *Infocom05* and *Rollernet*, for $\kappa \geq 3$, *Community*'s κ-contact CCDFs, in Fig. 3.9(f), bear the same overall outlook with a sharper slope for smaller κ. For 1- and 2-contact CCDFs, we hint an interesting phenomenon. We find two junctions around 400 seconds and another at 1,050 seconds. Opposed to our previous expectations, we have a better chance of getting contact of duration $D \in [400; 1,050]$ seconds than 2-contact slots of the same duration.

For *Unimi*, *RT*, and *Sigcomm09*, in Fig. 3.9(d), 3.9(e), and 3.9(b), their 1-contacts bear different behaviors than κ-contacts when $\kappa \geq 2$. As hinted in the *Community* dataset, for some values of κ, the probability of obtaining contacts longer than t seconds is higher than the probability for the same duration in other datasets. In Fig. 3.9(d), this phenomenon clearly appears for *Unimi*. In *RT*, $t = [0; 500] \cup [1,050; 10,000]$ seconds. In *Unimi*, the assertion is valid for the whole distribution. For $\kappa \geq 3$, CCDFs aggregate into an unique one. 2-contact distribution is a mixed behavior between 1-contact and larger values of κ.

Node-centered density dictates behavior. Due to the social nature of *Community*'s functioning, specific nodes tend to remain together and bring a high density around popular nodes. *Rollernet* is a dense sport setting and *Infocom05* has selective meeting points in the conference. They all exhibit an important node-centered den-

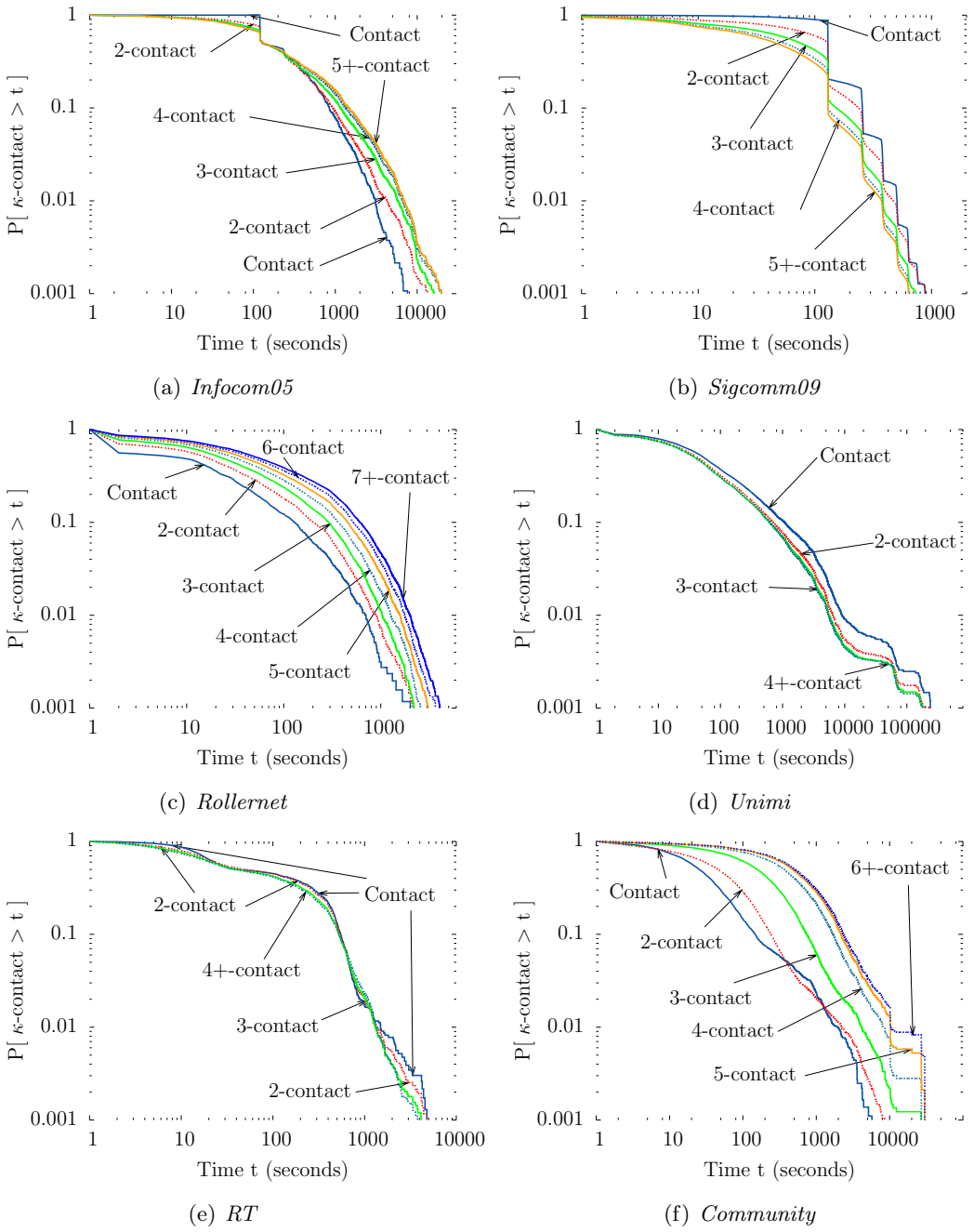
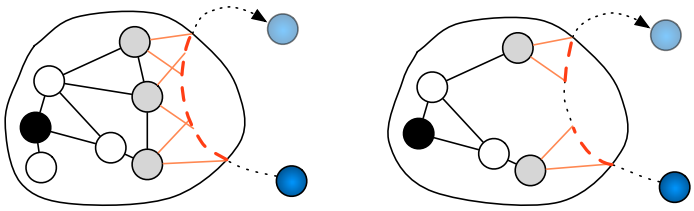


Figure 3.9: κ-contact distributions. There are two major patterns: (i) *dense distributions* where CCDFs having larger κ suffer a top right shift and a smoother slope than smaller κ, and (ii) *sparse distributions*, where all κ-contact distributions for $\kappa \geq 4$ aggregate and present a slight bottom left shift compared to the contact distribution (logscale on both axes).



(a) Dense (b) Sparse

Figure 3.10: Density related behavior for κ-contact. Density modifies the coverage zone of a node's κ-vicinity. For dense settings, we have a long continuous κ-contact interval. For sparse situations, we obtain two smaller κ-contacts for the same walk.

sity, whereas other datasets bear sparser density around each node. The ego density parameter may explain the difference between the κ-contact behaviors.

Fig. 3.10 illustrates a situation detailing the unexpected behavior of κ-contact distributions in sparse settings. Sparser densities limits geographical κ-vicinity coverage and induces smaller κ-contact intervals. Dense settings ignite distributions like *Infocom05*, *Rollernet*, *Shopping*, *Stanford*, *Sigcomm09*, and *Community* and will be henceforth mentioned as *dense distributions*. Low density settings like *RT* and *Unimi* enable the second type of distributions mentioned as *sparse distributions*.

κ-contact durations & number of intervals. In Table 3.3, we displayed the average duration of κ-contact intervals and in Table 3.4 the number of κ-contact slots for each of our five experiments. Two main behaviors arise. On the one hand, for *Infocom05*, *Rollernet*, and *Community*, we find an impressive continuous growth of average slots duration for every κ. On the other hand, all other dataset show the opposite evolution concerning average κ-contact duration. An increase in κ brings increased average κ-contact lengths.

For most datasets, we also find a logarithmic growth of the number of κ-contact intervals. Consequently, the number of intervals balances their length shortening. This testifies the growth in cumulated κ-contact durations in all datasets. Despite results observed in the previous section for *Sigcomm09*, *Unimi*, *RT*, and *Community*, we find that larger values of κ increase the overall κ-contact duration and modify its distribution. The main difference lies in the fact that *Infocom05*, *Rollernet*, and *Community* experience longer κ-contacts for large κ than other datasets, which have more shorter κ-contacts. In any case, both types have longer cumulated κ-contact times and it grows with κ.

Remarks. We have seen how κ-contact distributions predominantly exhibit two behaviors: sparse or dense distributions. Dense distributions follow our logical expectations. These distributions have sharper slope for lower κ and therefore a stronger demarcation among them than the next variety. Sparse distributions show κ-contact distributions

Table 3.3: κ-contact average duration in seconds.

	κ					
Dataset	1	2	3	4	5	6+
Infocom05	371	406	492	561	597	630
Sigcomm09	153	115	88	75	70	67
Rollernet	47	73	98	126	156	184
Unimi	1,325	901	821	797	791	801
RT	202	200	185	182	182	182
Community	96	139	359	752	1,001	1,123
Stanford	6	4	4	4	4	4
Shopping	101	88	74	67	65	65

Table 3.4: κ-contact number of intervals ($\times$ 1,000).

	κ					
Dataset	1	2	3	4	5	6+
Infocom05	3.7	11.1	15.4	16.9	17.3	17.2
Sigcomm09	43.0	137.2	247.2	323.1	366.1	388.4
Rollernet	5.2	15.4	25.2	32.7	36.3	38.1
Unimi	10.9	28.6	43.0	51.3	54.7	56.2
RT	2.3	5.1	8.4	10.6	12.0	12.8
Community	3.7	11.6	8.1	4.6	3.8	3.6
Stanford	11.6	23.1	28.9	30.8	31.3	31.5
Shopping	52.3	125.7	191.5	227.6	238.1	240.1

with comparable behaviors and no major demarcations. They quickly aggregate into a unique distribution above $\kappa \geq 4$. For these distributions, contrary to our primary beliefs, the probability of getting κ-contacts longer than t seconds is higher for shorter values of κ and contact durations.

However, for all measurements, the number of κ-contact intervals increases with every κ and springs a longer cumulated κ-contact time. Dense distributions obtain longer κ-contact intervals whereas sparse distribution has more short κ-contact intervals. Knowing which distribution fits, either sparse or dense, to a given situation modifies the way a protocol should consider its κ-vicinity. Adapting a routing technique to dense or sparse κ-contact distributions accordingly may help nodes leverage their κ-vicinity more efficiently than what is currently done.

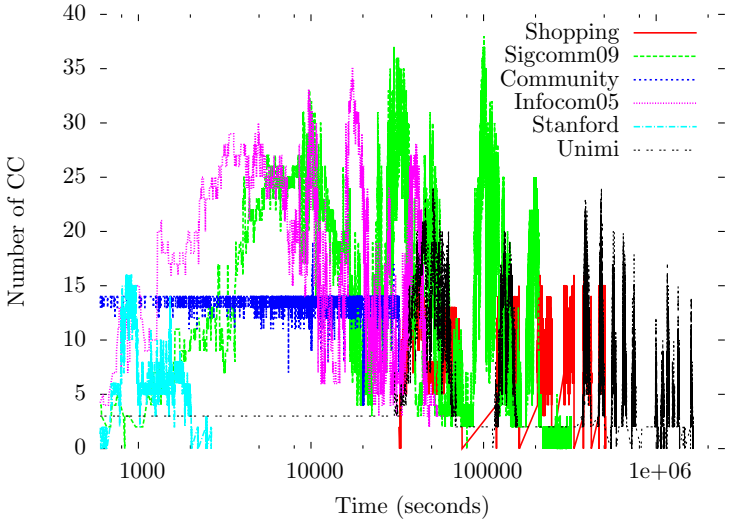


Figure 3.11: The presence of connected component (CC) of size higher than 2 according to time. As long as nodes find themselves in their "working" environment, they display numerous simultaneous CC. The conference datasets *Infocom05* and *Sigcomm09* due to their specific nature bear the highest number of concurrent CC (up to 38). The same pattern applies to other datasets but with a lower number of CC values. Still, they are most of time higher than 10. *Community* displays a peculiar number of CC evolution. Note the logscale on the x-axis.

3.4 Inner Topological Characterization

After analyzing the temporal properties of κ-vicinities in DTNs, we now focus on their inner topological characterization. To uncover these topological properties, we begin by observing all networks' connected components. Connected components embody κ-vicinities groundwork as, where there are connected components, there are potential κ-vicinities. Then we proceed with an investigation of ego densities. Finally, we provide a rule of thumb to derive the number of neighbors in the κ-vicinity based on the number of concurrent contacts.

3.4.1 The seat of κ-vicinities: connected components

Network connected components are the locus of κ-vicinity. Studying a network's connected components and their presence is a good indicator of the κ-vicinity potential in DTNs. Each node is its own connected component. Note that we use the connected components definition as used by Cormen et al. [87]. Since interesting κ-vicinity need at least contacts to be effective, we will be studying connected components of size higher than 2.

Presence. In Fig. 3.11, we represent the number of connected components (CC) of

Table 3.5: Average size, diameter, and gravity $\mathcal{G}$ of dataset's largest connected component.

Dataset	Size (s)	Diameter (d)	Gravity ($\mathcal{G} = s/d$)
Infocom05	20.2	5.0	4.0
Sigcomm09	16.3	4.5	3.6
Rollernet	12.9	6.0	2.1
Shopping	11.0	4.0	2.7
Unimi	9.1	4.1	2.2
Stanford	6.7	3.2	2.1
RT	12.4	5.9	2.1
Community	13.8	3.6	3.8

size longer than 2 for each of our datasets. We first observe a wide range in terms of number of connected components for all our datasets. For each scenario, there is a clear evolution through time. The number of CC may be close to zero at night when people are at home or outside the measurement environment. On the opposite, when people find themselves at work, school or at their conference venue, they bring a minimum density. This density ignites coexisting percolation phenomenon and results in the presence of numerous CC. During these periods, we often observe more than 10 simultaneous different CC [23].

For both conference datasets (*Infocom05* and *Sigcomm09*), we get the largest number of simultaneous CC. The conference setting where many people stay in different rooms to listen to talks may explain this large value. People stay together at different places, therefore, we have many small groups instead of a big one. For the sake of clarity, we did not display the *Rollernet* dataset here, though *Rollernet* displays between 5 and 35 concurrent CC during its course. *Shopping* and *Unimi* have a conference-like behavior, but at a lower level, they have periods of large number of CC but are still inferior to the one observed in conference datasets. The *Stanford* dataset has smaller concurrent CC values, but this stems from the nature of the logging devices used in the experiment. *Community* has a quasi constant number of CC which may be due to the synthetic nature of the dataset. The *Community* mobility model generates group-based meeting patterns.

There are many simultaneous CC in the observed datasets. Their numbers show how widespread the potential κ-vicinities are. But, since we observed the number of CC of size ≥ 2, we should study the size of these CC.

Size and diameter. In opportunistic networks, there are no connectivity graph diameters as such. The diameter notion only applies on a highly connected graph. However, the notion of diameter is still meaningful for each network connected component. In Table 3.5 , we present the average size of the largest CC, its average diameter for each

dataset, and their corresponding "gravity", given by:

$$\mathcal{G} = \frac{CC^{+}_{size}}{CC^{+}_{diameter}}, \tag{3.1}$$

where CC^{+}_{size} represents the largest connected component average size and $CC^{+}_{diameter}$ is the largest connected component average diameter. We define gravity $\mathcal{G}$ to be a simple indicator of how concentrated the largest CC is. The larger its value, the more the CC size tends to be bigger than its diameter, indicating a strong concentration in these CC. On the opposite, low gravity values show a weaker concentration of the components. Datasets with gravity values ≥ 3, like *Infocom05*, *Sigcomm09*, and *Community* display largest CCs that are more condensed than in all the other datasets. The distribution of nodes within these CC is tighter than in other situations. Therefore, their proximity in terms of connectivity seems stronger. They form components with stronger connectivity and allow resulting κ-vicinities to be more redundant link wise. Remaining datasets still bear a gravity ≥ 2, allowing numerous yet more loosely connected κ-vicinities. CC's gravity represent the link quality of anchored κ-vicinities.

3.4.2 κ-vicinity ego density $\mathcal{D}^{i}_{\kappa}$

As shown in Section 3.3, using a node's vicinity helps reduce κ-intercontacts and increase κ-contact times. With the κ-vicinity, we can measure the potential of such nearby companions in terms of opportunistic communications. Yet, obtaining information about a node's vicinity comes with a cost. To reduce the tradeoff between additional vicinity information and gathering costs, we can wonder how far a node must probe its vicinity to obtain the maximum information with the lowest probing cost. We may ponder this cost by limiting the scope of κ-vicinity to a given κ. To maximize the κ-vicinity utilization, we need to capture the nearest nodes and events as possible. The more neighbors and events we observe, the better we can use them to perform opportunistic operations.

To gather more neighbors, we must increase κ. The first aspect to analyze is each κ-vicinity ego density. Let $\mathcal{D}^{i}_{\kappa}$ be the density of nodes around i, obtained as

$$\mathcal{D}^{i}_{\kappa} = \frac{card(\mathcal{V}^{i}_{\kappa})}{\tau_t}, \tag{3.2}$$

where $card(\mathcal{V}^{i}_{\kappa})$ is the number of nodes in i's κ-vicinity and τ_t is the sum of all moments where $card(\mathcal{V}^{i}_{\kappa})$ was not null. κ-vicinity internal composition influences a node's κ-vicinity behavior. For a given probability p of having nodes at $\kappa + 1$ distance when there is a node at κ hops from i, the more κ-contacts a node has, the more chances it has of getting $\{\kappa+1\}$-contacts. In Table 3.6, we present the average $\mathcal{D}_{\kappa}$ according to κ. For all datasets except *RT* and *Rollernet*, above a certain threshold $\kappa_t = \{3, 4\}$, their $\mathcal{D}_{\kappa}$ does not increase anymore and is limited by the network diameter. More dynamic

Table 3.6: Average number of neighbors $\mathcal{D}_\kappa$ in a node's κ-vicinity.

	κ							
Dataset	1	2	**3**	**4**	5	6	7	8+
Infocom05	2.4	6.7	9.5	10.7	11.1	11.3	11.4	11.4
Sigcomm09	1.0	2.6	3.7	4.2	4.4	4.4	4.5	4.5
Rollernet	1.2	2.4	3.4	4.0	4.3	4.5	4.6	4.7
Shopping	1.6	3.3	4.3	4.6	4.7	4.7	4.7	4.7
Unimi	0.6	1.3	1.8	2.1	2.3	2.3	2.3	2.3
Stanford	0.5	0.7	0.8	0.8	0.8	0.8	0.8	0.8
RT	2.2	4.3	6.3	7.6	8.4	8.8	9.0	9.0
Community	2.0	4.1	4.7	4.8	4.8	4.8	4.8	4.8

or inconsistent patterns – *RT* and *Rollernet* – display logarithmic increase in $\mathcal{D}^i_\kappa$. For all cases, we verify card($\mathcal{V}^i_\kappa$) growth with κ indicating the presence of nearby nodes useable as relays for κ-contact.

For any datasets, observing only contacts shows limited $\mathcal{D}_\kappa$. For instance, the average $\mathcal{D}_1$ for *Infocom05* and *Sigcomm09* is respectively 2.4 and 1.0 neighbors. Whereas observing the κ-vicinity up to a few hops – $\kappa = \{3, 4\}$ – increases $\mathcal{D}_\kappa$ by at least doubling it. *Infocom05* and *Rollernet*'s average $\mathcal{D}_4$ is 10.7 and 4.2. By observing their 4-vicinity, we increased their number of number by a 4^+ factor. For $\kappa > 4$, the increase rate is less striking or even null. Nevertheless, longer κ-contacts in terms of path length may not be interesting because of potential path inconsistency due to all relays movements. Monitoring κ-vicinity up to a $\kappa = \{3, 4\}$ threshold brings most of the ego density a node can use.

Not-in-contact neighbors. An interesting situation occurs when pairs of nodes do not come into contact but belong to each other's κ-vicinity. Usual protocols miss this knowledge by overlooking the potential of nearby nodes. To analyze the impact of such situations, we studied the closest distance between nodes for all pair of nodes.

For *Unimi*, *Infocom05*, and *Sigcomm09* we find that respectively 92%, 91%, and 80% of pair of nodes do come in contact. This can be explained by the nature of the datasets where people are coworkers and have to meet to exchange ideas. However, we find that even there, some nodes do not find themselves closer than two hops distance respectively for 6%, 7%, and 12% of them. Other datasets deprived of the specific aim of meeting each other like *Rollernet*, *Community*, and *Shopping* show that contact only represent 31%, 42%, and 61% of the shortest distances. There, respectively more than 51%, 46%, and 35% of nodes stay at the closest between 2 and 4-hop distances. In the *Stanford* dataset, around 70 % of pair of nodes come into contact, but the remaining 30% of other nodes come at most at a 2-hop distance, most of them getting between two and three hops. In *RT*, all pairs of nodes come into contact.

By observing the $\{3,4\}$-vicinity, we manage to monitor additional situations of non-contact between nodes. As a result, we catch most pairwise κ-contacts occurring in a node's vicinity with only a threshold $\kappa = \{3,4\}$.

3.4.3 A rule of thumb for $card(\mathcal{V}_\kappa^i)$

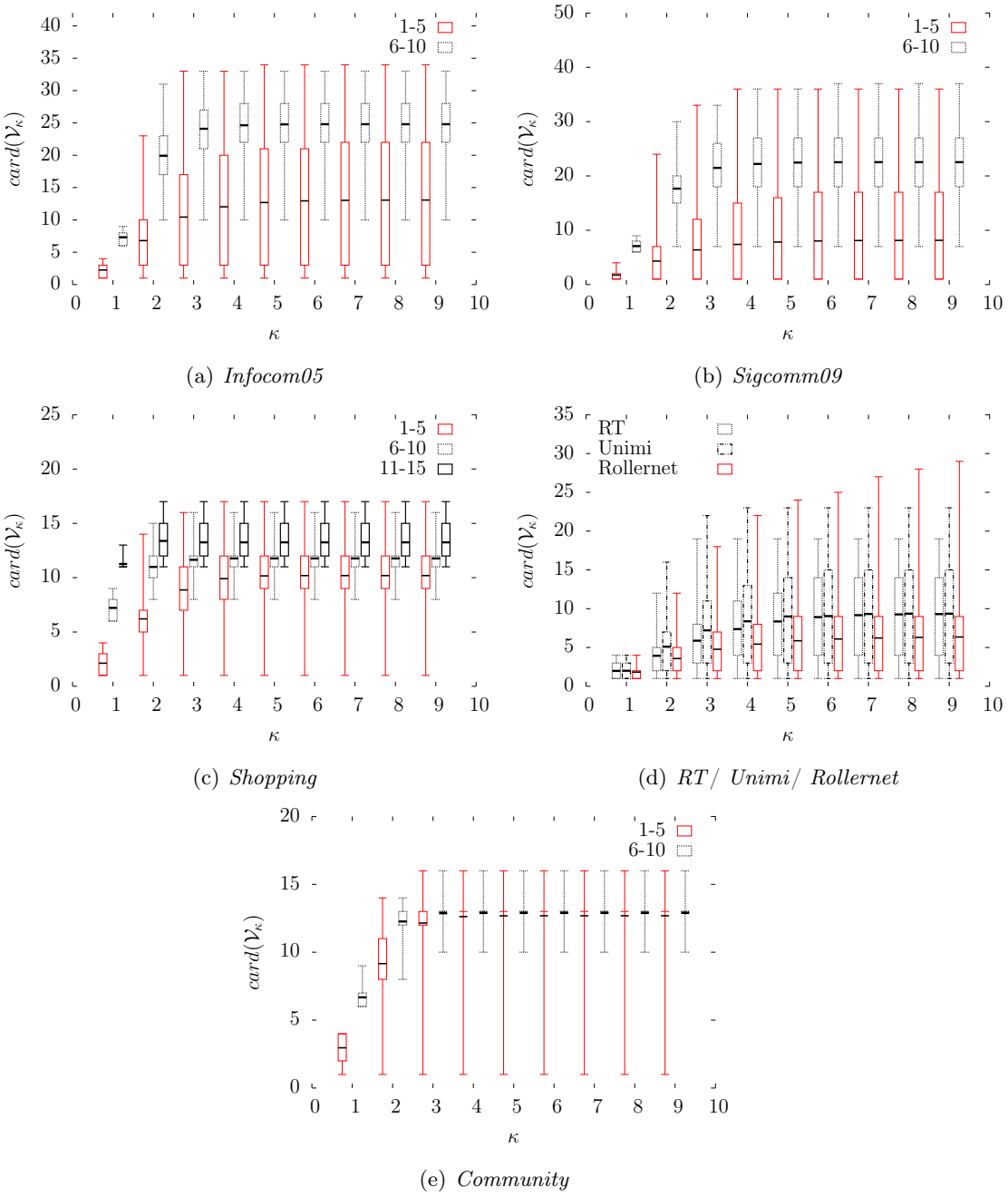
The strength of κ-vicinity lays in its size and extent. For a given node i, the most straightforward information relies on its current number of contacts ($\mathcal{C}^i$). To facilitate the deployment of our proposal without the costly neighborhood probing, we propose an heuristic based on $\mathcal{C}^i$ to derive node i's current κ-vicinity size – $card(\mathcal{V}_\kappa^i)$. We want to investigate the relationship between the number of nodes in contact and the current $card(\mathcal{V}_\kappa^i)$. To ease data understanding and their representation, we group the number of nodes in contacts by bins of 5 consecutive values. Fig. 3.12 presents this bin-based vision of the size of κ-vicinities. The x-axis indicates the considered κ. The y-axis represents $card(\mathcal{V}_\kappa^i)$. We observe for each 5-bins the distributions of κ-vicinity sizes. For its corresponding bin, each candlestick displays from bottom to top: the smallest vicinity size, the first quartile, the median size, the third quartile, and the longest value.

In the *Infocom05* figure, we observe that when nodes have between [1:5] nodes in contact, they have a median 3-vicinity size of 8 nodes. The minimum size observed was 1 and the highest 3-vicinity size was 33. The first and third quartiles are respectively 3 and 17. This means that for a node having 1 to 5 nodes in contact, 50% of the corresponding 3-vicinities have sizes between 3 and 17. Considering the second bin value of [6:10] contacts, the interquartile difference becomes more interesting. If a node has between 6 and 10 nodes in contact, its median 3-vicinity size is 24, the first and third quartile are 21 and 27. As a result, when a node has 6 to 10 nodes in contact, it can quite safely bet to have at least a 27 nodes in its 3-vicinity and all the more neighbors to use. For κ-vicinities with $\kappa \geq 3$, for the second bin [6:10], the vicinity size the interquartile value is restricted the interval 25 ± 3. The same observation for bin [6:10] holds for *Sigcomm09*.

The following datasets bear a "low" gravity (see Section 3.4.1). Their loosely connected vicinities may explain the small values we observed here. *Shopping* is the only dataset to have nodes with more than 10 simultaneous contacts and, therefore, 3 different contact bins. We notice a logarithmic growth of its median $card(\mathcal{V}_\kappa^i)$. As *RT*, *Unimi*, and *Rollernet* have only [1:5] concurrent contacts, we displayed them in a single figure. Their average size grows logarithmically with κ. Low gravity values indicate that the most secure connection occurs in contacts, therefore most of the neighbors are located near the considered node.

The *Community* dataset displays a different behavior. Its synthetic nature provides κ-vicinities for $\kappa > 2$ of constant size whatever the number of current nodes in contact.

As a rule of thumb, for dataset with high gravity like *Infocom05* and *Sigcomm09*,



(a) *Infocom05*

(b) *Sigcomm09*

(c) *Shopping*

(d) *RT / Unimi / Rollernet*

(e) *Community*

Figure 3.12: Contact-based $card(\mathcal{V}_\kappa)$ according to κ for each bin $[n{:}m]$ (bin size = 5). The x-axis represents the considered κ value for the κ-vicinity. The y-axis indicates the κ-vicinity size. For each bin, the candlebar displays from bottom to top, the lowest vicinity size, the first quartile, the median size, the third quartile, and the highest size value.

their $card(\mathcal{V}_\kappa^i)$ for $\kappa \geq 3$ is on average at least the double of their ceiling bin value (especially for bin [6:10]). For low gravity datasets, most events occur in contacts, so their $card(\mathcal{V}_\kappa^i)$ for $\kappa \geq 3$ is at least $\mathcal{C}^i$ or the ceiling bin value.

3.5 The Strength of Vicinity Annexation

In this section, we observe an example on how using a node's vicinity can improve opportunistic forwarding performances. Vicinity monitoring is an expensive process in disruption-tolerant networks. To lower its costs, we investigate the optimal κ threshold. First, we research the optimal κ for our datasets, then we observe the performances of a simple opportunistic forwarding approach with and without vicinity usage and finally we quantify the overhead implications of monitoring vicinity.

3.5.1 Threshold optimization

For each node, we analyze the average number of neighbors in their κ-vicinity. In the previous section, Table 3.6 shows this value for the whole dataset duration. We understand that above a certain threshold κ_t, a node's κ-vicinity does not expend much (except for the *RT* dataset, which has a random movement pattern and a high density). In *Community*, *Sigcomm09*, *Shopping*, *Stanford*, *Community* or *Infocom05*, a node's κ-vicinity does not grow significantly anymore above $\kappa_t = 4$. The same phenomenon appears with *Unimi* but with lower figures. The *Unimi* dataset is longer (two weeks) than other datasets. As we chose to analyze the average number of neighbors for the whole experiment duration, *Unimi*'s length lowered the expected average node number.

In Table 3.7, we focused on instants where nodes had at least one close neighbor. For each dataset, we analyzed all nodes' inner κ-vicinity distribution. Whenever a node had at least one neighbor, we observed the average number of neighbors located at a κ-hop distance. In *Infocom05*, we see that in average within a non void κ-vicinity, a node could find 3.0 nodes in contact, 4.4 at 2 hops, 3.0 at 3 hops, 1.4 at 4 hops etc. For *Community*, *Stanford*, and *Unimi*, the number of nodes at $\kappa > 2$ falls below 1. For *RT*, *Rollernet*, *Sigcomm09*, and *Infocom05*, the fall occurs after $\kappa = 4$. Above the threshold $\kappa_t = 4$, there will rarely be nodes at higher distances. Moreover, these distributions are linked to each datasets average diameter. All datasets average distributions are concentrated on shorter distance with $\kappa \leq 4$. For instance, *Community* does not have components larger than 4-hop distance. *Unimi* has components of at most 6-hop length. But within *Unimi*'s components, most neighbors appear in contact or at a 2-hop distance. The κ_t represents a high enough threshold so as to capture most of a node's surroundings. As a result, one would conclude that setting up a 4-vicinity monitoring for each node is optimal.

Table 3.7: Neighbors κ-distribution in a node's κ-vicinity.

	κ						
Dataset	1	2	3	4	5	6	7+
Infocom05	3.0	4.4	3.0	**1.4**	0.7	0.2	0.1
Sigcomm09-d1	3.5	5.7	3.7	**1.7**	0.7	0.3	0.1
Rollernet	2.0	2.5	2.1	**1.5**	0.9	0.6	0.3
Shopping	3.3	3.7	2.1	**0.8**	0.2	0.0	0.0
Unimi	1.5	**1.0**	0.7	0.4	0.2	0.1	0.0
Stanford	1.4	**0.6**	0.2	0.0	0.0	0.0	0.0
RT	2.3	2.3	2.0	**1.4**	0.8	0.4	0.2
Community	2.4	**2.3**	0.7	0.1	0.0	0.0	0.0

3.5.2 Loss and delays

To observe the performances of an opportunistic forwarding protocol, we chose to observe how vicinity annexation fares in a simple opportunistic protocol called the WAIT protocol also known as Direct Transmission [86]. In the WAIT protocol, the source stores the message until it meets the destination. The main criticism on this approach, although its minimal communication cost, is that the source may wait for a quite long time before being able to deliver the message or, worst, to completely fail delivering it. As we will see next, we observe reduction of delivery delays of up to 80% in average by extending of only one hop the vicinity knowledge; in some cases, delays can be reduced by several hours. This means that a simple variation of the WAIT protocol can be now applied in contexts that could not be considered previously. As the performance indicator, we observe the time a node has to wait before finding the destination with and without vicinity annexation.

For each mobility trace and each pair of nodes, we randomly generated 10 messages at different time instants. We chose a 10 message value as it is sparse enough to reflect the impact of neighborhood monitoring on waiting delays and there are still enough messages to make non marginal observations concerning waiting times. The most symptomatic situation arises when a pair of nodes never come into contact, but once and a while they belong to the same connected component. In this situation, the WAIT protocol drops the message whereas the neighborhood-aware variant can manage to forward it correctly. As scarce as this situation may sound, it happens for 10% of pair of nodes in *Infocom05*, 12% of *Unimi* nodes, 27.3% of *Shopping* nodes, 33.4% of *Sigcomm09*, 53% in *Community*, and around 55% of *Rollernet* nodes. If these nodes try to send a message using the WAIT protocol, they will simply fail. These fractions of nodes have *infinite waiting delays* when WAIT is in use. Otherwise, with the neighborhood-friendly version, they manage to deliver messages with bounded waiting

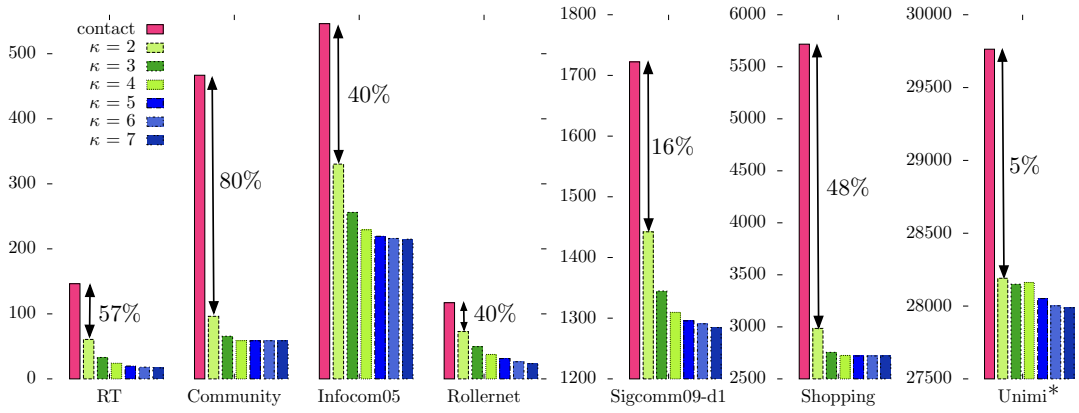


Figure 3.13: Averaged waiting times according to the threshold κ. For all traces, there is a clear improvement between the first and second bar (contact only vs. 2-vicinity). Being aware of a node's κ-vicinity can lead to divide waiting times by 4 like in *Community*. The higher the κ, the better the waiting delays, yet, above $\kappa > 4$, gains become negligible. Note that, for the *Sigcomm09**, *Shopping**, and *Unimi** datasets we focused on its top values. For *Unimi*, the average waiting time in contact is 18,232 seconds while in the 2-vicinity, it is 17,792 seconds. These high values come from the dataset length (two weeks).

times.

For these nodes with *bounded waiting delays*, we analyze to which extent neighborhood knowledge helps lower their waiting times. In Fig. 3.13, we show the averaged pairwise waiting times for each dataset. Each bar represents the average waiting delay we obtain with κ-vicinity probing. For every dataset, between the first and second bars, we notice significant reduction in the waiting times: 16% for *Sigcomm09*, 40% in *Infocom05* and *Rollernet*, 48% for *Shopping*, 57% in *RT*, and around 80% in *Community*. The *Unimi* dataset stands out because of its time scale. The experiment lasting two weeks, the random message generation process may choose values during weekends or nights. Even though, the relative difference between the first and second bar is more limited, the time reduction is still present between the first two bars.

For all datasets we observe that, although we keep reducing the waiting delays, the gains for $\kappa > 4$ are much smaller. This corroborates our first feeling that localized knowledge should be enough and suggests that we can, in practice, keep κ small.

3.5.3 Overheads

Supporting vicinity knowledge monitoring does not come for free. Any node needs to probe its vicinity and create a flow of messages around.

Impact of neighborhood knowledge overhead. There are many strategies for connected component gathering, from link state-like solutions to flooding techniques. For our study, we chose to compare two strategies:

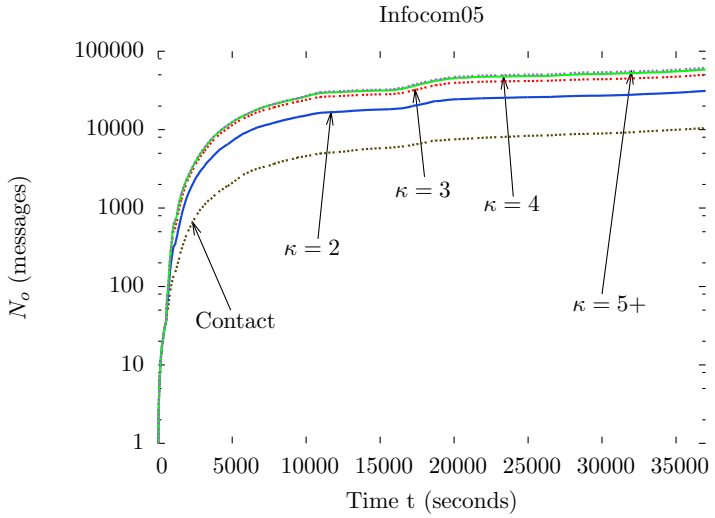


Figure 3.14: Neighborhood Knowledge Overhead (N_o) in terms of message sent by the discovery technique `Reg` for a node in the *Infocom05* dataset. On average, probing κ-vicinity with $\kappa > 4$ costs as much as probing the 4-vicinity. This version of neighborhood probing is very expensive. Note the logscale on the y-axis.

- Nodes keep monitoring their κ-vicinity at regular time intervals (called `Reg` hereafter).
- Nodes monitor their κ-vicinity when they have a message to send and stop when it expires (called `OnD` for "On Demand").

With `Reg` probing every 30 seconds, we quantified the volume of generated messages for different values of κ. Monitoring only contacts induces fewer overheads than any deeper neighborhood monitoring. For $\kappa = \{2, 3\}$, we have larger volumes of N_o. Beyond $\kappa = 4$, there are no noticeable differences for N_o. Overall behaviors are quite alike and depend on the surrounding density.

In Fig. 3.15, we plot N_o of the same source node as before. This time, we use the `OnD` method for neighborhood analysis. The reason we have noticeable jumps in all curves is, when the destination comes into the source's κ-vicinity, this latter stops monitoring its surroundings. Contact monitoring drops all but one message and is only plotted for the reader's information. As a result, the `OnD` technique appears more efficient than the naive `Reg`. In Fig. 3.15, we see how N_o evolves with time. With a simple probing technique `OnD`, we manage to constrain message overheads and deliver more messages than with the WAIT protocol. Also, an interesting result is how, for the same number of delivered messages (7), probing the 3-vicinity and beyond gives better results than probing only the 2-vicinity in terms of N_o. The faster the source finds the destination, the shorter the waiting delay and the lower the N_o.

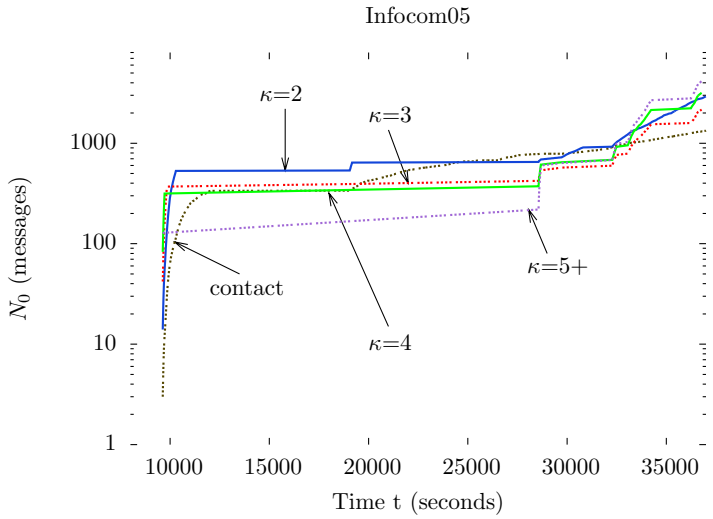


Figure 3.15: Neighborhood Knowledge Overhead (N_o) using `OnD` for a pair of node in *Infocom05*. Contact monitoring drops $9/10^{th}$ of messages and keeps monitoring its contacts without being able to deliver any messages. For 7 delivered messages, sensing 3-vicinity (or beyond) ends up cheaper than observing 2-vicinity. $\kappa \geq 3$ leads to shorter waiting delays and shorter probing periods than with 2-vicinity. Note the logscale on the y-axis.

Impact of data overhead. N_o seems to be the most expensive in terms of messages sent; yet, we also have to consider D_o (i.e., the number of messages over an end-to-end path). D_o adds an insignificant number of messages to N_o. It is important to underline that having a large D_o (i.e., a long path between the sender and the destination) can lead to undelivered messages. This is why one would prefer smaller κ.

3.6 Recommandations

Opportunistic Protocols. Until now, most opportunistic techniques used the obvious contacts – binary intercontact patterns. This straightforward vision may be sufficient for some approaches but it ignores nearby transmission opportunities by denying the existence of nodes beyond contacts. True opportunistic networking lives via human-driven movements, these patterns are not random and generate opportunistic encounters as well as topological connectivity. We need to leverage a city's popular places and hubs on the connectivity plane. By gathering surrounding information, the κ-vicinity knowledge can help us do so. Using a node's κ-vicinity improves both contact opportunities and intercontact durations. Moreover, in Section 3.4, we explain how observing a node's κ-vicinity with $\kappa = \{3, 4\}$ is enough to be aware of most pairwise activity in the vicinity beyond contacts and to benefit from ego densities. We also provide a rule of thumb to derive the number of nodes in a κ-vicinity depending on the current number of contacts.

Mobility Models. Musolesi et al. based their mobility model on social network theory [47]. Their model takes into account colocating patterns by mean of social attractiveness. Their intent, with HCMM proposed by Boldrini et al [88], is one of the most sensible we have seen in terms of binding synthetic models and social patterns. Still, they limit their approach to contact patterns which results in unexpected κ-contact and κ-intercontact distributions (see Fig. 3.7 and 3.9). In Fig. 3.11, *Community*'s number of connected components does not evolve, it remains constant through time. In Fig. 3.12, the *Community* dataset displays a very specific κ-vicinity size repartition. Due to the lack of available extensive realistic mobility traces, researchers must rely on synthetic mobility models. Musolesi et al. managed to create an interesting social mobility model, however, while this model respect most social patterns, incidental parameters like κ-vicinity behaviors and sizes as well as connected components characteristics may be offbeat.

3.7 Conclusion

In this chapter, we highlighted the "binary assertion" issue in opportunistic networks. The binary assertion issue consists in considering that all nodes that are not in contact are in pure intercontact. This binary contact-intercontact vision hampers our network vision. In Section 3.2, we solve the binary assertion problem by defining the notion of vicinity for disruption-tolerant networks (with κ-vicinity, κ-contact, and κ-intercontact). We start by acknowledging the hidden communication possibilities of intercontacts using sociostructures and we observe how variable are nodes behavior. Then in Section 3.3, we studied the temporal characterization of vicinities by observing κ-contact and κ-intercontact distributions. The resulting κ-contact and κ-intercontact distributions possess almost similar behavior as previous contact and intercontact distributions. Next, we focused on the composition of vicinities in Section 3.4 and provided a rule of thumb to derive κ-vicinity cardinality and gave hints on the κ value to optimize the give and take between additional surrounding knowledge and monitoring costs. Finally, we described the advantages of using vicinities in the WAIT protocol where we can reduce waiting delays up to 80% by considering the 2-vicinity instead of contacts only in Section 3.5. In the following chapter, we investigate the existing movements within the κ-vicinity. Can we find specific patterns? And if so, are we able to predict or mimic them?

Chapter 4

Digging into the Vicinity Dynamics of Mobile Opportunistic Networks

Contents

The utilization of opportunistic networks imposes many challenges. For the networking community, DTN had interesting, never seen before, features. As a consequence, many studies focused on characterizing network structures as well as node specific properties like degree or betweenness centralities [14, 89, 17]. Even if some properties have been unveiled by now, the relationship between a given node and its *vicinity* (closeby neighbors, including nodes without direct contacts) has not yet been thoroughly studied. This vicinity notion may be a key ingredient in optimizing opportunistic network forwarding. Why should protocol designer only consider contacts when we have other available vectors at hand?

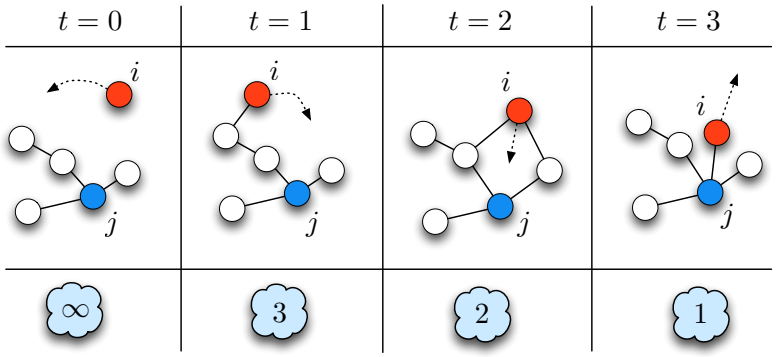


Figure 4.1: An example of asynchronous vicinity motion knowledge. At $t = 0$, node j is outside i's vicinity but coming closer. At $t = 1$, j pops into i's vicinity at a 3-hop distance. At $t = 2$, j moved closer to i at a 2-hop distance and even arrives in contact at $t = 3$.

4.1 Why Vicinity Dynamics?

As shown in Chapter 3, in real setups, nodes in direct contact represent only a small part of all opportunistic communications in DTN [84]. To leverage such unused connectivity, we propose to understand how neighbors move within a node's vicinity. In Fig. 4.1, we illustrate the evolution of a small network. At $t = 0$, nodes i and j have no path to each another – they are in intercontact. At time $t = 1$, nodes i and j are not in contact (1-hop distance) but are linked via a 3-hop path. At $t = 2$, i and j are at a 2-hop path and they finally come in contact at $t = 3$. The usual contact/intercontact vision would consider the time steps $t = \{0, 1, 2\}$ as the same, i.e., that i and j are in intercontact. Instead, when using the vicinity notion, such an "extended" view of communication opportunities is taken into account. Opportunistic networks can benefit from contacts that were not used before. To the best of our knowledge, no previous work has investigated how the vicinity structure of a node evolves through time in disruption-tolerant networks.

In this chapter, we introduce our Vicinity package for vicinity behavior analysis and generation in DTN. The Vicinity package takes as input a specific connectivity trace and extracts its vicinity behavior characteristics with the asynchronous vicinity motion framework. These vicinity characteristics can be used to train and evaluate the efficiency of protocols for opportunistic networks. We also use these vicinity characteristics as a model to regenerate different asynchronous vicinity motion with the same statistical properties as in the original one with the TiGeR generator. Asynchronous vicinity motion analyzes movements in a node's vicinity and TiGeR is in charge of the vicinity behavior generation process. In this chapter, we make the following contributions:

- **The asynchronous vicinity motion framework to analyze and understand vicinity behavior.** We provide a generating workflow to obtain meaningful vicinity analyses and manage to capture the statistical evolution of the

distance between nodes. Note that the synchronous vicinity motion will be defined in Chapter 5.

- **We identify two main network behaviors and three movement types with their application in opportunistic networks.** The first network type displays extended chains which represent a rich neighborhood with long instantaneous paths between nodes within the κ-vicinity. Whereas the other network type exhibits short chains with paths constrained to few hops. For extended chains, we noticed how three types of movements dominate all motions. *Birth*, *death* and *sequential* moves may represent 87% of all observed moves for a given dataset. We also explain how birth, death, and sequential movements evolution can be used in mobile networking.

- **Asynchronous vicinity motion take-away.** Following our results, we issue several directions concerning the use of vicinity in opportunistic network. These directions can help any opportunistic protocol designer leverage node's vicinity without the required monitoring cost.

- **We build TiGeR, a generic pairwise vicinity behavior generator for opportunistic networks.** Using the identified vicinity patterns – *Birth*, *death* and *sequential* – and the corresponding vicinity transitions probabilities, we created a process to generate representative pairwise timelines with two options. TiGeR allows pairwise vicinity behavior generation for various timescales. These timelines may bootstrap vicinity knowledge in other opportunistic protocols.

4.2 Vicinity Package Introduction

The Vicinity package contains most of our contributions in this dissertation. Two of its main parts are: the asynchronous vicinity motion framework who allows vicinity patterns analyses and TiGeR the pairwise vicinity behavior (timeline) generator. These two entities have been designed to function together as the asynchronous vicinity motion framework provides information such as state transitional probabilities and κ-contact durations distributions that are required by the TiGeR module. However, one can also use asynchronous vicinity motion and TiGeR on their own as long as one provides the required inputs. We next provide a few implementation details for both modules.

- **Asynchronous vicinity motion.** In this module, we need to recreate the provided network connectivity and compute all shortest distances for all pairs of nodes. To this end, we simulate network connectivity with the Python library NetworkX [90] and make the required arrangements using Python ≥ 2.7.

  -Requires: contact trace, number of nodes in the dataset.

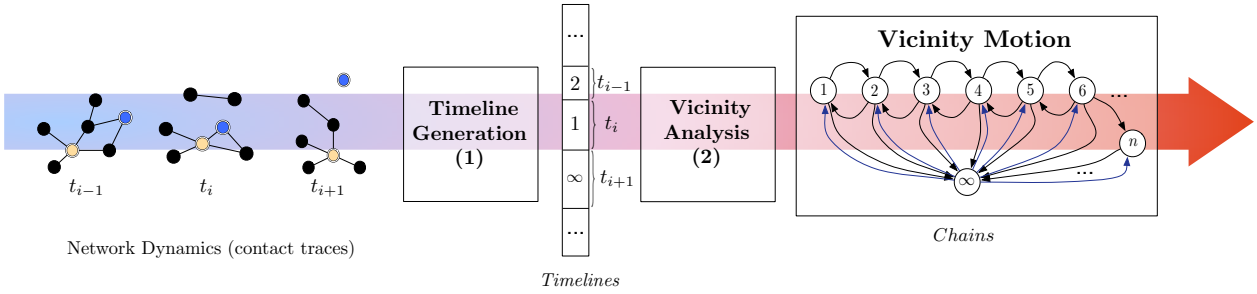


Figure 4.2: Vicinity motion generation workflow. We begin by reading Network Dynamics under the form of contact traces describing network connectivity through time. We process them using (1) the timeline generation module. This stage produces *timelines*. Step (2) aka Vicinity Analysis examines these sequences to compute transitional probabilities and corresponding vicinity motion *chains*.

-Provides: vicinity transitional probabilities, interval durations distribution.

- **TiGeR.** The timeline generator module processes extracted dataset characteristics (vicinity transitional probabilities and interval durations distribution) into synthetic vicinity behaviors. We use Python $\geq$ 2.7.

  -Requires: transitional probabilities, interval durations distribution, timeline required durations.

  -Provides: synthetic timelines.

The Vicinity package contains other tools useful for DTN trace processing as well as material developed in the next chapter. Details will be available on the dedicated webpage: http://vicinity.lip6.fr.

4.3 The Asynchronous Vicinity Motion Framework

The asynchronous vicinity motion (AVM) framework analyses node-centered vicinities (the κ-vicinity) for a given network. We want to answer the following question:

when the distance n between nodes i and j change,
what is the probability that their distance becomes m, with $m \neq n$?

In the remaining of our work, n is both the mutual shortest distance for a pair of nodes and the vicinity chain state while κ is the max hop distance in a κ-vicinity ($n \in (\{1, ..., \kappa\} \cup \{\infty\})$). We name the period between two changes in a pair's shortest distance a *step*. Note that AVM's step duration depends on the network dynamics, therefore, steps do not have a constant duration. AVM analyzes all changes in an asynchronous way. In opposition, we will study synchronous vicinity motion in Chapter 5. To answer the aforementioned question, we follow a two-stage methodology:

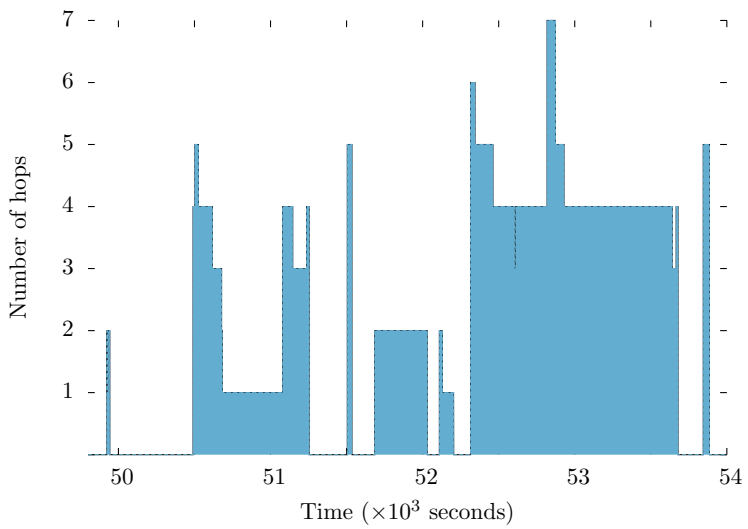


Figure 4.3: A pairwise timeline from the *Unimi* dataset. From 50,000 seconds to 50,500 seconds, the two nodes did not have a path to one another. Then, they briefly were at a 5-hop distance before coming closer at a 4-hop and then a 3-hop distance and so on.

1. **Timeline generation.** Knowing the shortest distances at different steps, we extract a vicinity *timeline*, which is the progression of shortest distance between any two nodes through time (see Fig. 4.3 for an example). By using these timelines we are able to perform probabilistic analyses.

2. **Vicinity analysis.** Timelines provide the necessary information to characterize the transition probabilities between given distances.

We recapitulate the whole workflow in Fig. 4.2.

4.3.1 Timeline generation

Our method takes contact traces as inputs. We first organize the trace as a chronological sequence of instantaneous events. Events can either be a link appearing or vanishing between a pair of nodes (i, j) at time t. We symbolize this type of event as $e =< t, i, j, \texttt{UP/DOWN} >$. `UP` indicates the appearance of a link between i and j and `DOWN` its disappearance.

For a given pair of nodes (i, j), a timeline consists in the sequence of their mutual shortest distance through to time (see Fig. 4.3). Formally speaking, we represent timelines as a sequence of tuples $< n, i, j, t_{begin}, t_{end} >$. This means that between t_{begin} and t_{end}, nodes i and j are at a n-hop distance.

The algorithm to generate timelines is detailed in Algorithm 1. It requires as input the contact trace ($\mathcal{C}$ in the algorithm) and the number of nodes in the network ($\mathcal{N}$ in the algorithm). All timelines are initialized with a starting tuple $< \infty, 0 >$ indicating

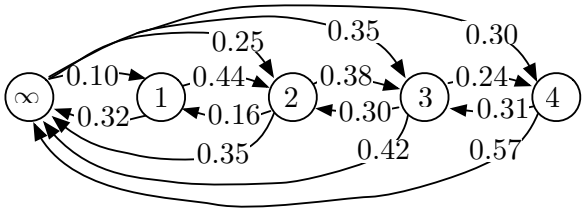


Figure 4.4: Infocom05 average asynchronous vicinity motion for a pair (i, j) and $\kappa = 4$. For the sake of clarity, we display only a few transitions. The probability of a node appearing in contact $\{\infty \to 1\}$ is 10% or when nodes are at a 3-hop distance, the probability for them to be next at distance 2 is 30%.

that they are in κ-intercontact '∞' at time 0. Following tuples indicate a change in this state and the time at which it occurs. As long as there are events in the trace, we read them and update the adjacency matrix before computing pairwise shortest distances. Then, we update the corresponding pairwise timelines accordingly. Finally, we format and print gathered data into timelines.

4.3.2 Vicinity analysis

To illustrate AVM, we use a chain process for *each* pair of nodes. For a given node i, let $X^s_{i,j}$ describe the distance between nodes i and j at step s. The vicinity analysis step (2) takes timelines as input and provides the corresponding transitional probabilities for vicinity chains. We describe the two main component type of our chain process here:

States. The chain states depends on the κ we choose, i.e., the size of the vicinity we wish to monitor. The number of states is $\kappa+1$; the first state, denoted '∞', corresponds to the case where the two nodes are in κ-intercontact. The state $\{1\}$ represents a contact and the remaining states $\{2, \ldots, \kappa\}$ correspond to a situation of κ-contact where the exact distance between the nodes is the corresponding state.

Remember that we consider each pairwise movement as a step unit. In this chapter, we do not consider specific time frame durations to avoid dataset dependence and focus on effective network movement.

Transitional probabilities. To understand asynchronous vicinity motion, we concentrate on the chain conditional probabilities between states, i.e., the probability of two nodes being at a distance of m at step s knowing that they were at a distance n in the previous step $s-1$: $\mathbb{P}(X^s_{i,j} = m \mid X^{s-1}_{i,j} = n)$, $m \neq n$. This informs us exactly on the movements within the vicinity.

As an example, we show in Fig. 4.4 the average transitional probabilities of asynchronous vicinity motion for *Infocom05*. For the sake of clarity, we omit certain transitions. As we can see, when nodes i and j are in κ-intercontact '∞', the probability that they meet directly is 10% while it is 35% for a 3-hop distance. This appearance behavior varies from one dataset to another and highlights the utility of the Vicinity package to easily gather such data.

Algorithm 1: Timeline (TL) generation.

Requires: $\mathcal{C}$, $\mathcal{N}$// `contact trace, number of nodes`
Ensure: $\mathcal{N} \times (\mathcal{N}-1)$ timelines (TL)
Local: $\{adj\}$ // `adjacency matrix of size` $\mathcal{N}^2$
initialization; // `all timelines initialized with the tuple <`∞`, 0>`
while *size of* $\mathcal{C} \neq 0$ **do**
  t_{cur}, i, j, `event` = pop first tuple of $\mathcal{C}$;
  if *event* = `UP` **then**
    $adj_{i,j} = 1$;
    $adj_{j,i} = 1$;
  end
  if *event* = `DOWN` **then** $adj_{i,j} = 0$;
  $adj_{j,i} = 0$; **for** $i \leftarrow 1$ **to** $\mathcal{N}$ **do**
    for $j \leftarrow 1$ **to** $\mathcal{N}$ **do**
      if $i \neq j$ **then**
        d_{cur} = shortest path(i, j);
        if *length of* $TL_{(i,j)} = 1$ **then**
          append (d_{cur}, t_{cur}) to $TL_{(i,j)}$;
        end
        else
          (d_{last}, t_{last}) = get last tuple of $TL_{(i,j)}$;
          if $d_{last} \neq d_{cur}$ **then**
            append (d_{cur}, t_{cur}) to $TL_{(i,j)}$;
          end
        end
      end
    end
  end
end
for $i \leftarrow 1$ **to** $\mathcal{N}$ **do**
  for $j \leftarrow 1$ **to** $\mathcal{N}$ **do**
    if $i \neq j$ **then**
      format and print $TL_{(i,j)}$;
    end
  end
end

4.4 Asynchronous Vicinity Motion: Analyses and Patterns

4.4.1 Short and extended chains

From our analysis, we observe two types of vicinity chains. Extended ones that can travel up far to states like 10 or 12 or shorter ones with movements up to 1 or 2 hops.

Short chains. In short chains we retrieve the previous assumption that nodes are either in contact or in κ-intercontact; the difference here is that they can drift to a 2-hop distance. We noticed such setting for two of our datasets: *Sassy* and *Stanford*. The observed chain consists in states $\{\infty, 1, 2\}$. As a result, such settings bear no or very low κ-contact advantages. Most of the times, when one detects a node, its next move will almost always be to vanish from the vicinity. Opportunistic protocols must also take these patterns into account when necessary.

Extended chains. Datasets like *Infocom05*, *RT*, *Rollernet*, and *Unimi* display extended vicinity chains (see Fig. 4.4 for an example). Extended chains bear more potential traveling states. Some even going to 12 and longer distances. Extended chains have the characteristic to allow high states and, therefore, the possibility of wider end-to-end transmissions via recurrent κ-contacts. Extended chains may also exhibit a wide range of intra chain movements. We will next see how three types of movements dominate tendencies. With only a few movement patterns, we will show it is possible to oversee most of a node's upcoming movements.

4.4.2 Max-min distance division

Before diving into vicinity chains properties as such, we need to understand more general characteristics of the network in order to adopt an appropriate κ value in our analyses. Here, we do not attempt to tune a specific value for each dataset, but only to estimate a reasonable generic value that will allow us deriving unbiased conclusions. In practice, in order to optimize the traditional vicinity monitoring costs and find a good balance between overhead and vicinity knowledge, a network designer might want to compute a specific κ that fits well the scenario.

To better understand how pairs of nodes evolve, we first perform an exhaustive analysis in which we select a value for κ that is large enough to cover the nodes' connected components. For each pair of nodes, we compute the max-min distance in the dataset, which is their maximum shortest distance value while still belonging to the κ-vicinity. Fig. 4.5 shows the repartition of nodes according to their respective max-min distance in five of the datasets. Results are shown in aggregated form – the bar for "d ={1,2,3}" means that the maximum shortest distance for a pair of nodes is between 1 and 3, and so on. For *Infocom05*, we observe that the largest proportion of pairs (around 78%) have at most a 7 to 9-hop distance between them, while only 16% of them have at most a 4 to 6-hop distance, 8% of them have at most 10 to 12 hop distance, and almost

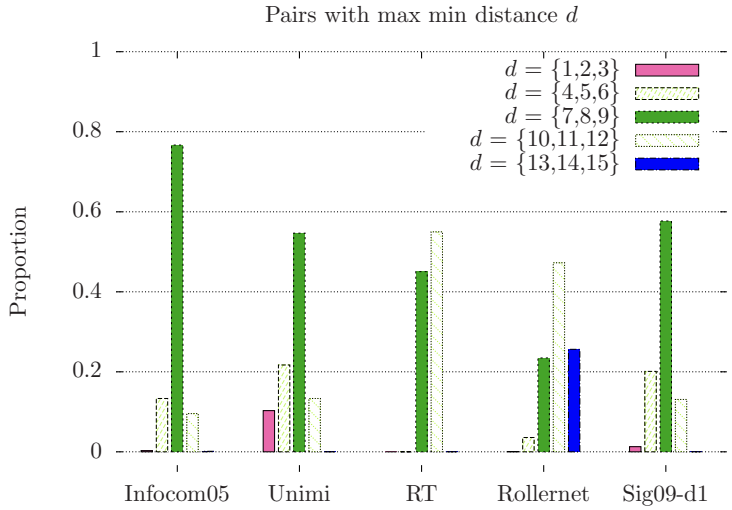


Figure 4.5: Repartition of pair of nodes with max-min distance d. d represents the maximum shortest distance observed for a pair of nodes in their corresponding experiments. For the sake of clarity, we summed proportions in sets of three consecutive distances.

none of them have at most a path of 1 to 3-hop distance. *Unimi* and *Sigcomm09* present the highest proportion of nodes with maximum distance 7- to 9-hop paths. *RT* and *Rollernet* have most of their max-min between 10- and 12-hop paths.

We confirm here, that even considering high level pairwise asynchronous vicinity motion, discriminating behavior is a must. For the remainder of our paper, we will focus on vicinity chains up to state $\{7, 8, 9\}$ as they represent the most part of situations observed.

4.4.3 Vicinity chains distributions

Average time spent in each state. Table 4.1 presents the average duration in seconds spent in state κ for each state. For *Sigcomm09*, *RT* and *Unimi*, we observe a gradual decrease of durations. On the other hand, *Rollernet* has an increasing tendency while *Infocom05* has a mixed behavior. The specific status of *Rollernet* as a dynamic sport event may explain the increasing values. Short distances have a very low life span because of the fickle and dynamic connectivity in the setting. The crowd absorbs longer distances (note that we do not discriminate path changes if they are of the same length).

Stationary distributions. In Table 4.2, we show the stationary distributions for the different datasets when $\kappa = 7$. In *Infocom05*, when we come across the setting, there is a 25.3% chance that the node we are looking for does not belong to our κ-vicinity, 5.5% chance of the node being in contact, 15.4% at a 2-hop distance, 20% at a 3-hop, and so on. Note that by observing its 4-vicinity, when we sum the birth probability from

Table 4.1: Average time spent in each state in seconds.

Dataset	State							
	∞	1	2	3	4	5	6	7
Infocom05	2,029	399	296	224	175	131	154	212
Sig09-d1	2,149	149	83	42	26	18	14	11
Rollernet	167	51	74	86	102	117	127	184
Sassy	157k	2,315	53,871	1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
Stanford	2,972	1	1	0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
Unimi	28,621	520	161	83	57	44	35	28
RT	172	202	88	60	41	30	24	17

Table 4.2: Stationary distributions in percentage.

Dataset	State							
	∞	1	2	3	4	5	6	7
Infocom05	25.3	5.5	15.4	20.0	16.0	9.7	5.1	2.2
Sig09-d1	26.2	4.8	14.9	21.4	16.6	9.8	4.4	1.9
Rollernet	28.2	2.3	7.7	11.5	12.5	11.5	9.5	7.3
Sassy	49.2	34.8	15.5	0.5	0.0	0.0	0.0	0.0
Stanford	45.0	48.0	6.9	0.3	0.0	0.0	0.0	0.0
Unimi	35.0	9.0	14.0	15.0	12.0	8.0	4.0	2.0
RT	29.1	5.0	10.6	14.1	14.3	11.5	7.7	4.5

states 1 to 4 and normalize the result by the probability of effectively having nodes, we have a 77% chance of spotting a node we are actually looking for. Such a posteriori knowledge is useful to evaluate the probability of finding a node quickly upon arrival or even to quantify the probing frontier in order to keep low maintenance costs.

4.4.4 Vicinity patterns

Datasets bearing extended chains offer more possibilities of next hop transitions. In the analyzed datasets, we observe three main types of transitions, namely *birth, death*, and *sequential movements*.

4.4.4.1 Birth in the κ-vicinity

We qualify of birth the phenomenon of appearance in the κ-vicinity after a period of κ-intercontact. The main interest of such knowledge is for a node or a protocol to know at which distance another node may appear. Imagine in the *Infocom05* dataset that node i wants to send a message to node j, which is currently outside i's κ-vicinity, without

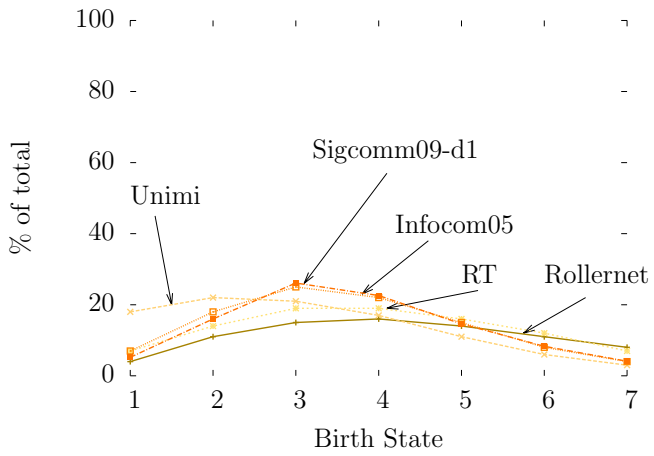


Figure 4.6: Birth rates.

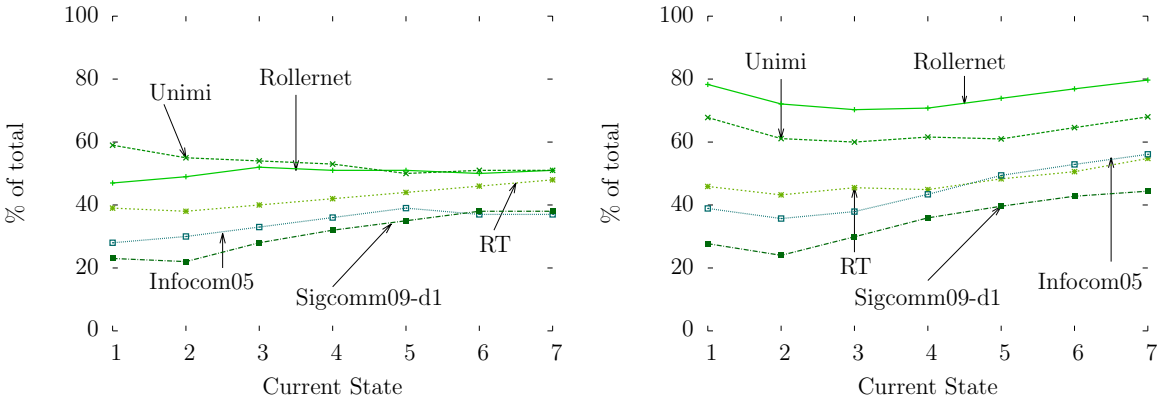
relying on fully opportunistic forwarding. Given the computed stationary values from Table 4.2, we now know that j will appear with a probability of 20% at a 3-hop distance.

In Fig 4.6, we present the values concerning the birth motion for our datasets. On the x-axis, we represented the actual incoming state (the distance at which a node appears). On the y-axis, we present the actual birth transitional probability for each distance. For all datasets, the highest birth probability belongs to the set {1, 2, 3, 4}. The cumulated transitional probabilities up to 4 represent from 50% to 70% depending on the dataset. For a random dataset, if we had chosen to extend these probing limits only to a state 4, we would detect from 50% to 70% of nodes vicinity appearance. Hence, probing the 4-vicinity is enough to get most of the arrivals patterns in a node's surrounding. This is a strong confirmation for our previous finding on the optimal limit of κ-vicinity probing [85].

4.4.4.2 Death in the κ-vicinity

In opposition to the notion of birth for arrival patterns, we call death the phenomenon of nodes vanishing from the κ-vicinity. We analyze the datasets in two different aspects: the proportion of deaths with regard to the full chain (absolute) and compared to *natural* movements only. We define as natural movements, all movements except transitions between non-consecutive states except toward ∞. By analyzing the part of death transitions in natural moves, we try to understand how representative is their part in all easily predictable transitions.

In Fig. 4.7(a), we show the evolution of death probabilities for the different states of the chain. Most datasets have steady absolute death rates, their maximum variation being 12%. However, *Sigcomm09* death rate evolves with higher states. The absolute



(a) Proportion of deaths with regard to the full chain.

(b) Proportion of deaths with regard to natural movements.

Figure 4.7: Death (absolute and relative to natural movements) rates.

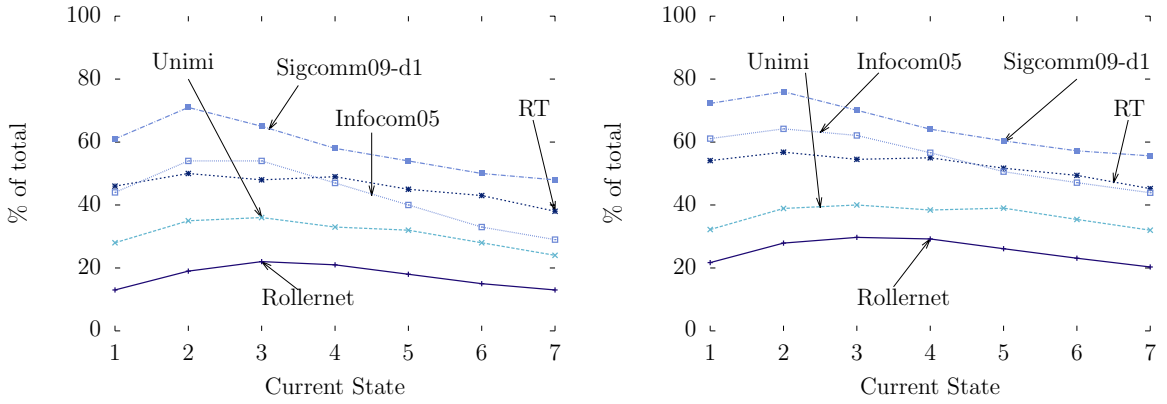
death rate steadily grows with observed states. In Fig. 4.7(b), we show the results in the case of natural movements. We observe an interesting phenomenon: almost all datasets bear the same relative death rate evolution. Relatively to natural movements, the proportion of death moves has a similar overall pattern (soft decrease followed by a soft increase), the main difference being the starting values on the ordinate axis for each dataset. Being able to foresee death movements, i.e., a node being in κ-intercontact can indicate when to begin a fully opportunistic routing technique. As long as nodes are in the vicinity, we can use end-to-end paths towards them. However, when we suspect that nodes will next be out of the κ-vicinity, it may be time to trigger a different routing approach.

Birth and death events represent a big share of the movements alongside sequential movements as presented next.

4.4.4.3 Sequential movements

We define as sequential movements for two nodes the process of drifting closer or further from each other using adjacent states of the chain: when nodes (i, j) are at a 4-hop distance, they sequentially move closer if they are at a 3-hop distance during their next step, they sequentially drift away if they are next at a 5-hop distance. Sequential movements are part of natural movements in general. Natural movements include death and sequential moves.

Absolute sequential probabilities. Our first observation is that a non-negligible part of vicinity movements stems from sequential behaviors. For *Unimi* and *Infocom05*, as long as nodes stay in the κ-vicinity, sequential movements represent between 50%



(a) Proportion of sequential movements with regard to the full chain.

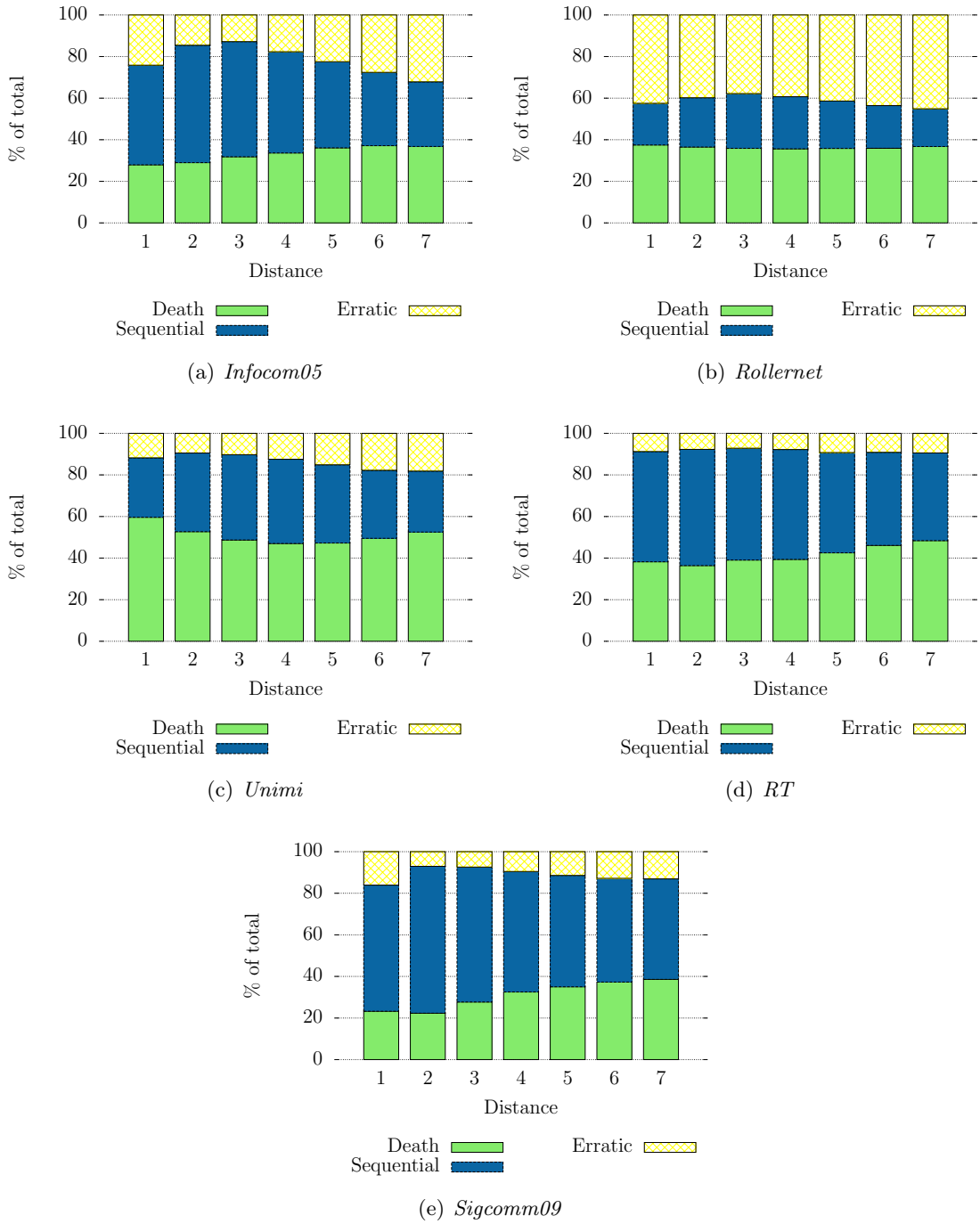
(b) Proportion of sequential movements with regard to natural movements.

Figure 4.8: Sequential (absolute and relative to nature movements) rates.

and 80% of movements. We call erratic or random movements, all movements that are not birth nor death nor sequential moves. They represent a minor share of asynchronous vicinity motions and can be overlooked as predicting their destination is tougher and brings only marginal knowledge gains. Sequential movements still rule. In Fig. 4.9, we display the proportion of death, sequential, and erratic movements (from bottom to top) among all vicinity moves for our long chained datasets. *Infocom05* is representative of other datasets. In Fig. 4.9(a), erratic movements grow with the distance between the nodes, while death processes remain stationary around 30%. Sequential movements are strong within the 4-vicinity. The effect of sequential patterns is less influential at further distance because of the environment perturbation.

The further two nodes are, the higher the proportion of erratic movements. Wider vicinity bears fickler connectivity at the borders and more random hopping. Also, the presence of erratic movements may also be related to datasets beaconing frequency (see Section 2.2). For the same dataset, the longer the beaconing intervals, the more we may miss movements in the dataset. This may result in a higher percentage of erratic movements.

Relative sequential probabilities. We go deeper in the analysis of sequential movements and divide the notion into two sub-notions: *incremental (inc)* and *decremental (dec)* movements. While *inc* movements consist in movements where the distance increases to the immediately higher state, *dec* are the opposite where distance decreases to the immediately lower one. Similarly to death rates, we investigate *inc* and *dec* moves using different scales: absolute, relative to natural movements, and relatively to the overall proportion of sequential moves. Concerning *dec* moves, datasets show-



(a) *Infocom05*

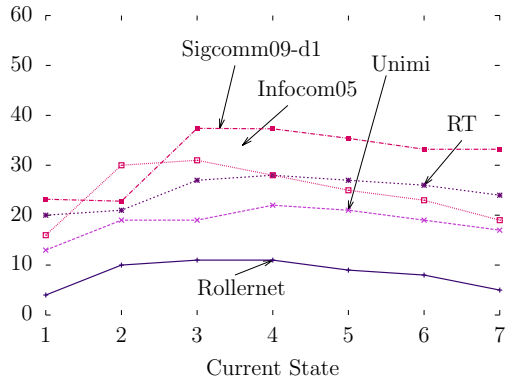
(b) *Rollernet*

(c) *Unimi*

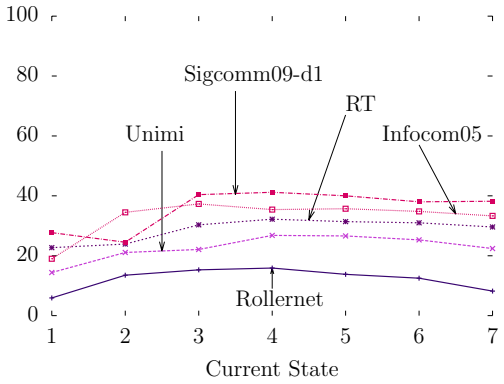
(d) *RT*

(e) *Sigcomm09*

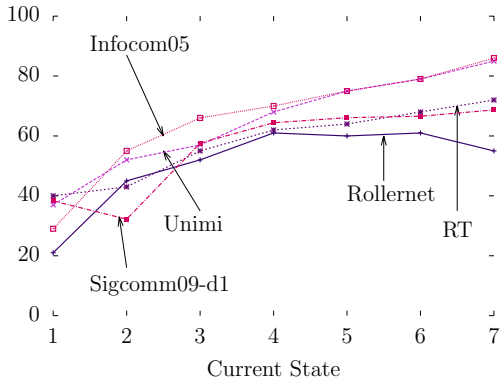
Figure 4.9: The asynchronous vicinity motion movements repartition (death, sequential and erratic) for all datasets for each distance. Death and sequential movements represent the greater part of all possible outgoing movements. Erratic movements represent a marginal part of the observed behavior. However the erratic movements presence seem to slowly grow with higher distances.



(a) *dec*: absolute values



(b) *dec*: relative to natural moves

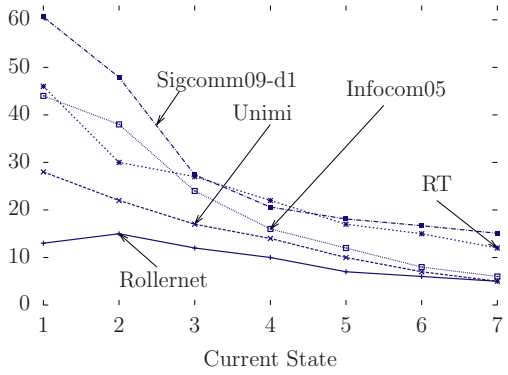


(c) *dec*: relative to sequential moves

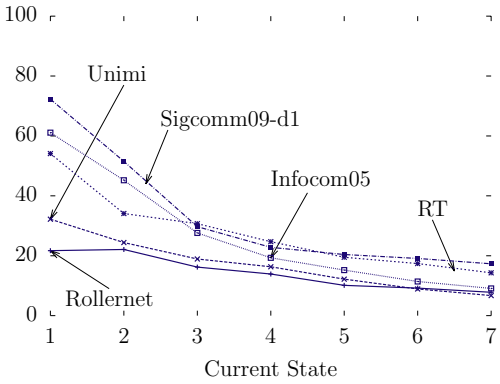
Figure 4.10: Average sequential *dec* movements in details. *Dec* movements are the opposite tendency, nodes incrementally moving closer to each other. On the x-axis, we present the current state nodes are in. On the y-axis, we represent the percentage of vicinity moves they represent.

ing extended chains (*Infocom05*, *Rollernet*, *Unimi*, and *RT*) display a slow increase until distance 2, followed by a smooth and soft decrease (see Fig. 4.8(a)). The *Sigcomm09* dataset displays the same pattern but drifted to state 3. *Sigcomm09* decreases at state 2 before increasing at state 3 and then slowly decays. The anomaly concerning *Sigcomm09* is present for all *dec* moves. The proportion of sequential moves is easy to predict. The same deduction can be made concerning the proportion of sequential movements with regard to natural movements in general (see Fig. 4.8(b)).

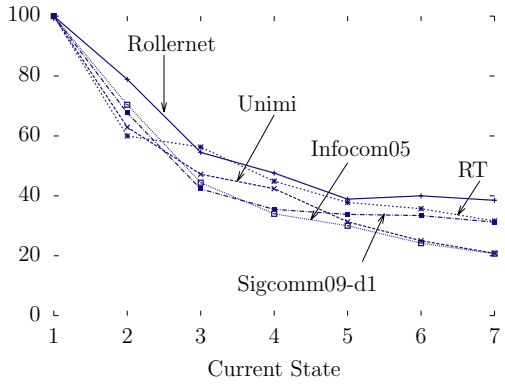
Observing a finer granularity for sequential movements helps us understand which patterns are more prominent: do nodes have a higher probability of drifting away (*inc*) or moving closer (*dec*)? In Fig. 4.11, we present *dec* and *inc* proportions. Concerning *dec* and *inc* absolute values (Fig. 4.10(a) and 4.11(a)), we observe that the moving



(a) *inc*: absolute values



(b) *inc*: relative to natural moves



(c) *inc*: relative to sequential moves

Figure 4.11: Average sequential *inc* movements in details. *Inc* movements indicates movements incrementally moving further. On the x-axis, we present the current state nodes are in. On the y-axis, we display the percentage of vicinity moves they represent.

closer pattern has a stationary distribution for all observed datasets. *Dec* does not vary much around their initial value, whereas the drifting away pattern (*inc*) quickly decreases with higher states. This can be explained by the fickle connectivity at longer distances. Movements result in death rather than drifting away moves.

With regard to natural and sequential movements, the proportion of *dec* and *inc* moves among natural movements is different in terms of values but bears quite the same evolution as previous absolute observations. *Dec* maintains quite stationary values while *inc* display quick decrease in Fig. 4.10(b) and 4.11(b). A closer analysis of *inc* and *dec* moves, in Fig. 4.10(c) and 4.11(c), compared to sequential movements shows clearer patterns. For all of our datasets, the proportion of *dec* almost linearly increase while the *inc* part linearly decrease. As *inc* and *dec* represent a full partition of sequential

movements, the observation seems logical. The increase of dec_{seq} moves stems from the decline of *inc* in the natural case, rendering up a higher proportion of *dec* in the sequential case.

4.4.5 Asynchronous vicinity motion take-aways

From our observations, we summarize some best practices for opportunistic protocol designer to use concerning vicinity movements in mobile networks.

Birth. The phenomenon of appearance in vicinity is interesting to notice. Birth indicates when pairwise end-to-end connectivity becomes useable. So to the question, how far one should monitor its surrounding to detect these births? The empirical answer based on our observations would be 3 or 4 hops away. In Fig. 4.6, we notice that the cumulated birth probability up to state 4 ranges from 0.46 to 0.72. For the *Unimi* dataset, we get a 78% chance of detecting an early arrival in the vicinity. In *Unimi*, the cumulated probability from states 2 to 4 is 60%. So we increase the arrival detection by a factor 3 compared with when monitoring contacts only. For the *Sigcomm09*, this increase factor is 12.4 compared with contacts only. As extended vicinity vision increases monitoring costs, we wish to maintain a quite low distance monitoring.

Death. We notice that their rate remains quite steady with a slight growth for higher states (see Fig. 4.7(a)). This means that we only need to probe death rates from contacts to get a hint on what occurs further hops away. When there is low network stability, death rates tend to be quite high (for instance with *Rollernet*). In these cases, one must expect the next vicinity movement to break the end-to-end connectivity.

Sequential. The notion of sequential movements includes moves from a given state to the immediately higher or lower state. These movements increase from state 1 to $\{2, 3\}$ and tend to slowly decrease further away (Fig. 4.8(a)). In more details, we considered *inc* movements (those going from a given state to the immediately higher one) and *dec* moves (those going to the immediately lower state). *inc* movements quickly decrease with higher states for all the observed datasets (Fig. 4.11(a)). When a pairwise shortest distance is greater than 3, there are very little chances for a node to drift away. They are more likely to "die" or come closer. On the opposite, *dec* movements increase for states between 1 and 3 and remain quite steady for states higher than 3. This shows there is a tendency to attraction between nodes. But for higher states, death process and erratic movements often break this tendency.

These observations apply to the datasets we analyzed. Nevertheless, with the asynchronous vicinity motion framework tool we implemented, any researcher can get these analyses on a pairwise or network-wide scale as long as he provides network connectivity.

4.5 TiGeR: Synthetic Timeline Generator

The direct application of asynchronous vicinity motion analyses is the possibility of generating synthetic timelines. Timelines embody the pairwise vicinity behavior. TiGeR (TImeline GEneratoR) relies on the asynchronous vicinity motion module outputs (extracted timelines and transitional probabilities).

4.5.1 Motivation

The use of timelines to bootstrap nodes' vicinity knowledge into opportunistic networks is original. Before, protocols like BUBBLE Rap used history of nodes contact periods in order to predict future encounters [11]. Now, instead of focusing on contacts only, we extend this knowledge to the node's κ-vicinity. Vicinity provides more network knowledge and therefore multiplying the possibilities of encountering another node [85]. We can extract pairwise contact patterns from the generated timelines. The timeline generation can also be useful to test opportunistic protocols because it generates timelines that can be used to predict the next distance appearance for a given node, for instance.

4.5.2 Generation processes

To generate pairwise vicinity behavior, TiGeR relies on transitional probabilities, a given κ value and κ-contact durations distributions. Based on the asynchronous vicinity motion transitional probabilities, we generate a sequence of pairwise shortest distance matching the AVM probabilities. The first step is always to generate a distance sequence matching the given AVM transitional probabilities. The challenge is then to match these timelines distances into correct interval durations as well as plausible distance intervals number. For each of the max distance observed, we perform the following steps:

4.5.2.1 Hop sequence generation

This step generates an AVM transition compliant hop sequence (a list of distances whose AVM transition will be similar to the provided transitions). We take a max distance D and process the provided AVM transitions as follows:

- **Beginning state.** We need to bootstrap the generated timeline with a first starting distance (a starting state for the provided AVM). We choose to get a random starting distance denoted d_0 among all the existing states $\{\infty, 1, ..., D\}$. For example, let us begin with $d_0 =$ '∞'.
- **Run the AVM chain.** We run the corresponding AVM chain from the starting state $d_0 =$ '∞'. We choose the highest outgoing probability from ∞ and decrement the taken transitional rates by a certain value Δ. In TiGeR, we set Δ to be the

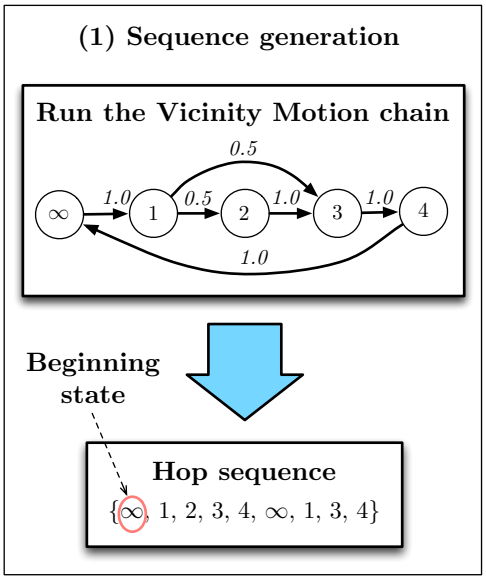


Figure 4.12: TiGeR's hop sequence generation example. From the given asynchronous vicinity motion transitional probabilities, TiGeR produces a possible hop sequence s. s has transitional probabilities similar to the initial asynchronous vicinity motion transitional probabilities.

greatest common factor among all transitional rates. When we find ourselves to be in a sink node (all output transitional rates are null), we randomly choose another output state. We stop the distance generation when all the transitional rates are depleted.

We then repeat the processes for all the max distance values in [1:D]. Considering the max distance distribution, we can generate several synthetic timelines. The only precaution to take is to normalize the corresponding AVM transitional probabilities before running the chain.

We detail an example from Fig. 4.12. For a max-min distance equals to 4, me assume the following transitional rates: { $(\infty \rightarrow 1 = 1.0)$, $(1 \rightarrow 2 = 1.0)$, $(2 \rightarrow 3 = 0.5)$, $(2 \rightarrow 4 = 0.5)$, $(3 \rightarrow 4 = 1.0)$, $(4 \rightarrow \infty = 1.0)$ } all other transitional probabilities are considered null here. We start with d_0 = '∞'. We determine $\Delta = 0.5$ (because it is the highest common factor among {1.0, 0.5}). From the AVM in Fig. 4.12, we take the transition '$\infty \rightarrow 1$'. The resulting AVM is the same as before except that the '$\infty \rightarrow 1$' transition value is now 1.0-0.5 = 0.5. We normalize this value by the total outgoing probabilities and '$\infty \rightarrow 1$' becomes 1.0. We are now in state 1 and can decide to go either to state 2 or 3 because they have the same outgoing probability 0.5. We randomly choose state 2 and decrease the '$1 \rightarrow 2$' to 0.0 and normalizing '$1 \rightarrow 3$' to 1.0. Then from state 2 we got to 3 and so on, until all transitional probabilities are ≤ 0.0. In our case, the matching hop sequence s would be s ={∞, 1, 2, 3, 4, ∞, 1, 3, 4}.

Now that we have s, we need to match these states/distances sequence with accurate intervals durations.

4.5.2.2 Time matching

Using hop sequence s, we match each of its distances with a plausible interval duration. Depending on the user need, TiGeR provides two modes. First Mode (I) mimics timelines with life-like interval durations while the second Mode (II) outputs timelines with more AVM compliant transitions. This step requires the user to give $\mathcal{L}$ the timeline length he wants to get. We call the timelines generated with Mode I, MI-timelines, and those by Mode II, MII-timelines. We next detail both functioning.

- **Mode I reflects plausible intervals duration.** The first available option means to reflect the κ-interval durations. For each distance from s, we use the κ-contact duration distributions from the AVM module. Let us say that $s = n$, then we use a Gaussian distribution based on the n-contact duration distribution (average duration, first and third quartile) to extract a plausible interval value. Then, we record and sum the obtained durations until the total intervals duration exceeds $\mathcal{L}$. MI-timelines may lack some step from s but they respect the required duration $\mathcal{L}$ and plausible interval durations.

- **Mode II focuses on transitional probabilities.** In the second available option, we focus on respecting AVM transitional probabilities. We keep the same process as in Mode I without limiting the time matching to the $\mathcal{L}$ duration. We keep on generating the κ-intervals durations to plausible ones for the entire sequence step s. Then, by the end of the sequence we use a fitting factor $\mathcal{F}$. L is s's sequence total duration and $\mathcal{F}$ the required length

$$\mathcal{F} = \frac{\mathcal{L}}{L}.$$

  If $\mathcal{F} < 1$, we find that the generated sequence duration is higher than the required duration and we multiply all the generated sequence by $\mathcal{F}$. Else if $\mathcal{F} > 1$, it means that the required duration is higher than the sequence matched duration then we need to either repeat step sequence or stretch the durations by multiplying them by a factor $\mathcal{F}$.

We present a recap of the Time Matching stage in Fig. 4.13.

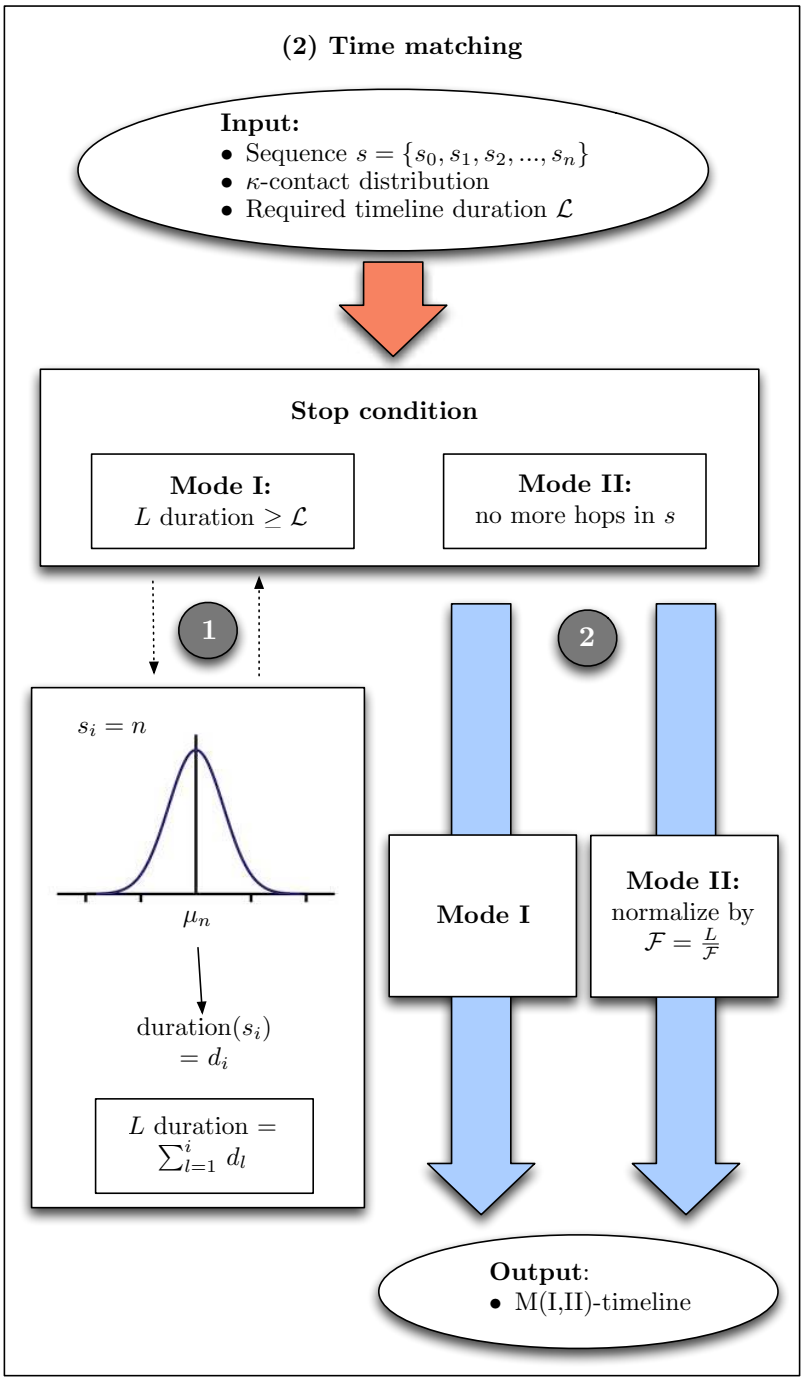


Figure 4.13: TiGeR's time matching process.

4.5.3 Evaluation

To evaluate the correctness of our approach, we compare M(I,II)-timelines to the original timelines characteristics. We analyze the accuracy of the transitional rates of the M(I,II)-timelines and the precision of κ-intervals durations.

4.5.3.1 Methodology

Pairs of nodes have specific behaviors and their max-min distance varies from one pair to another. For each of these max-min distances and each of the datasets, we generated corresponding M(I,II)-timelines. Then, we obtain timelines for max-min distances 1 to 7. For instance, MI-7-timeline replicates the pairwise vicinity behavior of a 7 max-min distance in Mode I. We generated MI- MII-timelines of around 50,000 seconds.

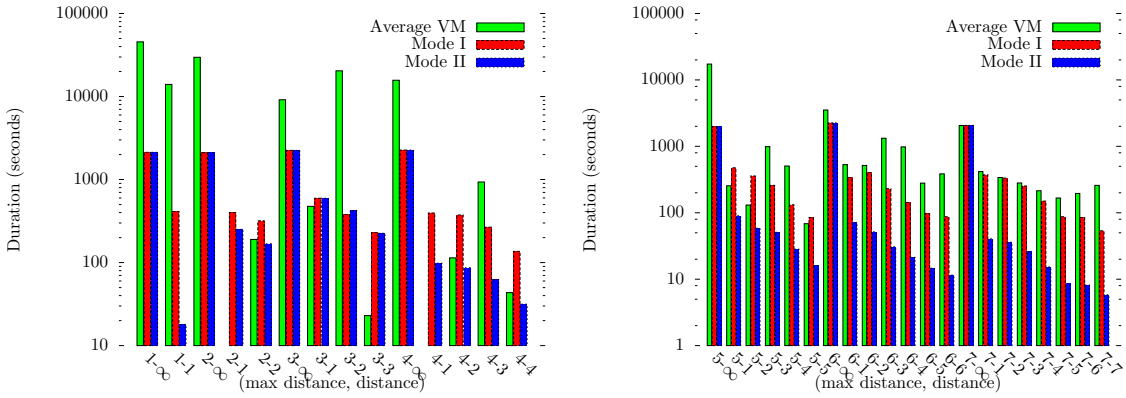
We compare pairs of nodes with max-min distance D behavior to MI-D-timelines and MII-D-timelines. For each timelines, we analyze the average κ-contact durations and compare them to chosen original timelines. We also compare synthetic M(I,II)-timelines AVM transitions to original timeline transitions. We perform these analyses with averaged datasets values and specific pairwise timelines.

4.5.3.2 κ-interval duration distributions

For all datasets and each max-min distances in [1:7], we computed the average κ-interval duration values for MI- and MII-timelines.

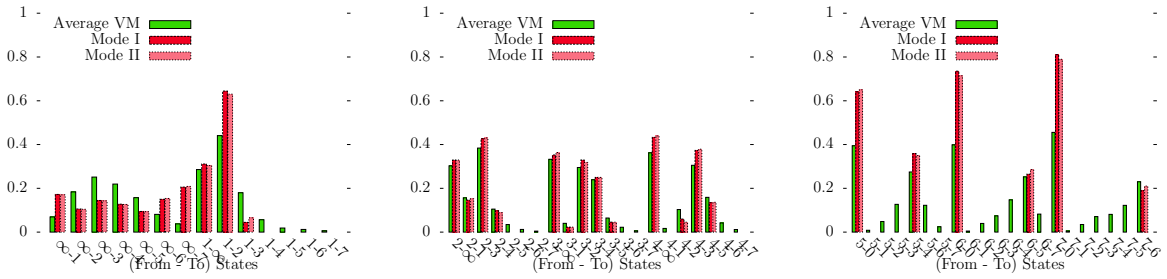
In Fig. 4.14, we present the averaged κ-contact duration for the *Infocom05* dataset. First bar displays the value for *Infocom05* average AVM, the second one, indicates the value for MI-timelines and the third one shows the value for MII-timelines. On the x-axis, we note the "max-min distance value D – the κ distance". Meaning D is the required max-min distance ($D \in$[1:7]) and κ is the value we consider for κ-duration ($\kappa \leq D$). In Fig. 4.14(a), we see the average durations for $D \in$ [1:4]. In Fig. 4.14(b), we display the average duration for $D \in$ [5:7]. Note that the y-axis is on logscale.

As expected when we designed TiGeR, Mode I focuses on reflecting plausible κ-contact durations. Therefore, most MI-timelines observed average durations line up with the average AVM durations compared to MII-timelines. For lesser $D \in$ [1:3], we notice the highest differences between average AVM and generated timelines. This difference can go up to a factor of 52 for "3-2" ($D = 3$ and $\kappa = 2$). For higher $D \in$ [4:7], this difference decreases with on average a 2.7 factor difference. Differences range between 0.30 and 8.7. The differences for lesser D may be explained by TiGeR's generating parameters. With MI-timelines of maximum duration of 50,000 seconds, we do not get the same duration as with original datasets where some nodes barely see each other. MI-timelines tend to coincide with the average AVM duration behavior. MII-timelines also coincide at a lesser degree as it is mainly MI-timelines divided by



(a) Max-min distance 1 to 4 (b) Max-min distance 5 to 7

Figure 4.14: κ-interval average durations for the averaged timelines and generated M(I,II)-timelines. Note the logscale on the y-axis.



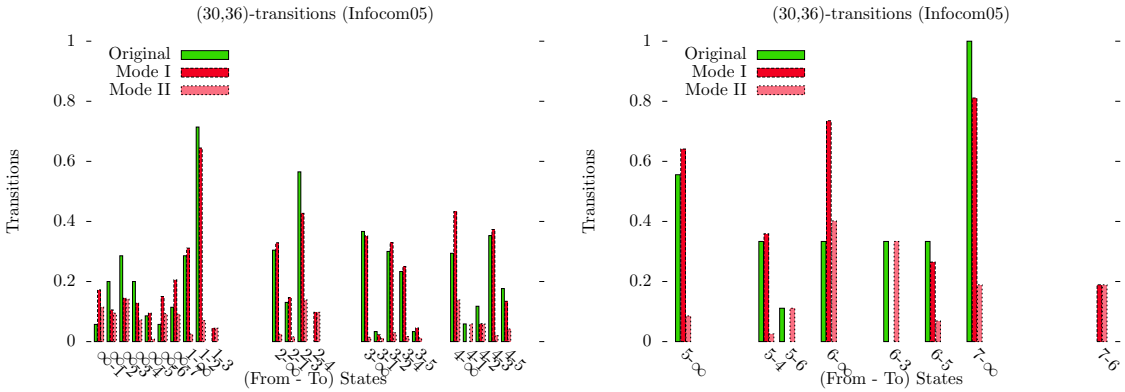
(a) Birth - Contact (b) States 2 to 4 (c) States 5 to 7

Figure 4.15: AVM transitions for the averaged *Infocom05* timelines and the M(I,II)-timelines.

the fitting factor $\mathcal{F}$.

4.5.3.3 Transitional rates

Another aspect we must look after while replicating pairwise vicinity behavior are the transitional probabilities from one state to another. In Fig. 4.15, we analyze the VM transitions for the average *Infocom05* behavior, MI- and MII-timelines. On the x-axis, we have the "from-to" states. For instance, 1-2 indicates the transition from a 1-hop distance (contact) to a 2-hop distance. On the y-axis, we display the transition probability value. In Fig. 4.15(a), the 1-2 transition value is 0.45 for the average AVM and around 0.63 for MI- and MII-timelines. Transitions generated by Mode I or II slightly differ from one another. These transitions tend to fit the general overlook of the average AVM timeline. For states below 4, transitions tend to fit the average



(a) States 0 to 4

(b) States 5 to 7

Figure 4.16: Transition values for the (30,36)-timeline, and M(I,II)-timelines in the *Infocom05* dataset.

timelines. For states above 5, TiGeR generated fewer non-null transitions. Nevertheless, we must understand that the average timeline represents an aggregated behavior and that movements in the {5,7}-hop vicinity zone are quite reduced in reality.

To envision this phenomenon in the {5,7}-hop vicinity zone and see how M(I,II)-timelines fit with specific original timelines, we observe nodes (30, 36) of the *Infocom05* dataset. In Fig. 4.16, we represent the AVM transitions for the pair (30,36), MI-, and MII-timelines. For states below 4 in Fig. 4.16(a), considering both transitions values and their presence, M(I,II)-timelines and the initial (30,36) timeline have very similar behavior. In Fig. 4.16(b), for states higher than 5, we notice that the (30,36)-timeline does not have many transitions in the {5,7}-hop vicinity zone. There are only $7 \rightarrow \infty$, $6 \rightarrow (\infty, 2, 5)$, and $5 \rightarrow (\infty, 3, 4)$ movements. For these movements, we notice that MI- and MII-timelines have close values. In this case, we clearly observe how M(I,II)-timelines do fit the original (30,36)-timeline. This shows how TiGeR generates realistic timelines.

4.6 Observations

In addition to the aforementioned observations from Section 4.4.5, we issue additional guidelines concerning vicinity usage in DTNs.

Practical utilization. Our study showed how pairwise moves are most of times divided into two main patterns: death and sequential. These two types represent up to 90% of observed movements. When using the vicinity, protocol designers can bet on the next mutual movement to be either death or ± 1 hop from the current state. This prediction on node behavior can be used to tune opportunistic protocols leveraging

end-to-end paths as in [6]. This knowledge can indicate when it is more advised to use pure opportunistic techniques.

κ recommendation. In Section 4.4, we see how node are usually at a 9-hop max-min distance when in κ-contact. So all vicinity moves occur below these 9 hops. However, considering the neighborhood probing costs, we imagine how probing further costs more. In Section 4.4.3, we see that probing the 4-vicinity allows nodes to detect up to 90% of arrivals. Table 4.2 shows that most movements are confined to states below 4 and only a marginal part of movements result in states above 4. We also notice how the average time spent in state above 4 mostly decreases (see Table 4.1) except for *Rollernet*. This $\kappa = 4$ threshold is an interesting give and take for vicinity probing. As a future work, applying dynamic κ threshold may be an alternative.

4.7 Conclusion

In this chapter, we introduce part of our Vicinity package for disruption-tolerant and opportunistic networks. First, the asynchronous vicinity motion framework allows vicinity understanding in opportunistic mobile networks via the notion of κ-vicinity, timelines, and asynchronous vicinity motion chains. Based on the asynchronous vicinity motion module's output, we identified three main vicinity movement types: birth, death, and sequential movements. All of them have their own patterns impacting mobile networking. For instance, we realize that by monitoring a node's vicinity up to 3 or 4 hops away, we can improve the vicinity arrival detection by up to 8 times. By observing sequential movements, we hint an attraction phenomenon when it comes to distances above 3 hops. For these distances, nodes are more likely to come closer or die instead of moving further away. Then, TiGeR enables synthetic pairwise vicinity behavior generation under the form of timelines. Based on the asynchronous vicinity motion framework outputs like state transitions, κ-contact distributions, we are able to generate timelines reflecting the behavior of a network node. TiGeR has two generating modes depending on what is expected: plausible intervals durations (Mode I) or life-like transitions (Mode II). The use of such synthetic timelines can help bootstrap vicinity knowledge in opportunistic protocols as well as testing them with synthetic vicinity properties. By providing mixed inputs (from various datasets type) to TiGeR, we can generate original vicinity types. When providing this tool, we offer the possibility to researchers to perform vicinity analysis on any dataset as long as he has dataset connectivity.

Chapter 5

Predicting Vicinity Dynamics

Contents

The contact–intercontact knowledge in opportunistic networks indicates past transmission opportunities. A lot of studies showed how a node's contact history may be enough to roughly determine future encounters. Beyond this simple knowledge, we observe how convenient it would be to be able to predict pairwise encounters for DTN. This would allow a finer tuning of opportunistic protocols and the possibility to discriminate between different protocol types. For instance, in Chapter 3 we mentioned the power of vicinity annexation. In a similar use, knowing whether two nodes are likely to be close again in the future could allow opportunistic protocols to either delay or force a transmission. In this chapter, we raise the question of the predictability of nodes vicinity behavior in the κ-vicinity. By using the inherent information of the vicinity motion model and its transient stochastic knowledge, we expose an heuristic to predict pairwise vicinity distances at future steps. In Chapter 4, we presented the asynchronous vicinity motion who focused on network transitions independently of any specific time frame. This approach avoided time dependence and had low additional probing costs. However, when it comes to prediction capacities, we feel it may lack

a necessary time notion. AVM's step definition has a non constant duration which may hamper prediction results. For this chapter, we will also be using a Synchronous Vicinity Motion (SVM) bearing a stricter time notion and possibly providing better prediction performances than AVM.

In summary, our contributions are the following:

- **A Markovian heuristic for vicinity distance prediction.** Using the vicinity motion model, we provide a heuristic based on vicinity motion transition matrixes. Our heuristic predicts pairwise distance for future steps based on the current pairwise distance.
- **A new vicinity motion variant integrating the time parameter: synchronous vicinity motion (SVM).** The asynchronous vicinity motion works by using steps (whenever there is a pairwise distance change) as a unit. This removes the temporal parameter of our observations because a step does not have a constant duration. With the synchronous vicinity motion, we put the time dimension back into the game and consider a "step" to be of a fixed duration. This fixed duration brings us a stronger prediction scheme.
- **An evaluation of our heuristic accuracy.** We find that our heuristic allows the user to guess the correct pairwise distance with a success ratio of up to 99% using SVM values. This heuristic performs well enough to allow a node to use either fully opportunistic networking or another alternative networking paradigm.

In the next sections we present our Markovian heuristic based on vicinity motion and detail the time-aware vicinity motion variant named synchronous vicinity motion (SVM). Then we observe how our heuristic fares using average vicinity motion knowledge. We find that SVM-heuristics perform better than AVM-heuristics and even achieves a prediction accuracy of up to 99%. Nevertheless, SVM needs to periodically probe its environment and has quite high functioning costs. AVM-heuristic still achieves up to 40% of correct guesses but needs less information than SVM.

5.1 Problem Statement

In the DTN research field, knowing which nodes belong to our vicinity and which ones do not is already a helpful point of view. With the κ-vicinity, instead of considering only nodes in contact, we observe neighbors beyond 1-hop. These "close yet not in contact" nodes could be message destinations, or information carriers.

Let us imagine a regular scenario of a daily commuting from home to work. Jean leaves home in the morning takes his favorite commuting mean and heads toward his workplace. At noon, he takes his meal along his coworkers, then goes back to work. At

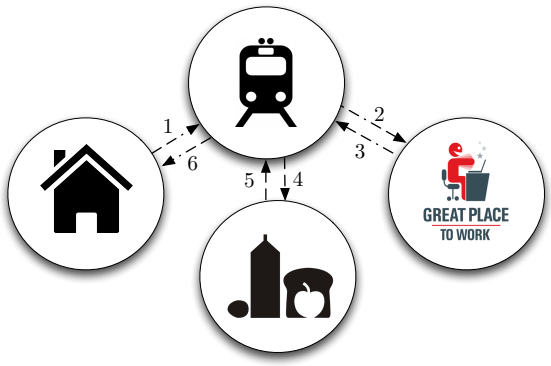


Figure 5.1: An example of a workday routine. Jean leaves his house in the morning and takes any public transportation mean (1). He heads to work (2) and stays there with his colleagues all day. At the end of the day, he leaves to get some groceries (3) and (4) and finally heads home for a well deserved rest (6). During his journey, Jean meets a lot of people and visits some key places like the train station, his workplace or his home. In these places, he may meet the same person on and off again. They are part of his vicinity.

the end of the day, he leaves his office, returns home and eventually gets groceries on his way back. During his whole daily journey, Jean meets a lot of people, whether he acknowledges them or not.

Everywhere Jean moves, people surround him, at home, in the bus, in the streets, at work. Currently, opportunistic networks only gather information about nodes directly around Jean (the 1-hop knowledge). A first way of using Jean's mobility is to observe his current vicinity beyond 1-hop contacts, the more people around him, the more there are potential message carriers. In a previous study, we showed how this simple vicinity observation can help improve performance [85]. During his daily trip, Jean maintains certain regularity. This regularity occurs at various levels. Every morning and every night, he is at home with his family and neighbors. During his commuting, he may travel with the same people whether he realizes it or not, the *familiar stranger* phenomenon [3]. Each workday, Jean interacts with his coworkers. This regularity in meeting patterns can be quite interesting to forward information with smaller costs. Using this potential regularity, we may be able to predict another node's future presence into our κ-vicinity. Without future insights into node's presence, choosing one technique or another is quite random. Possessing the presence forecasting ability allows us to get the best of both the end-to-end and the opportunistic communication world.

In our analyses, we forecast pairwise distances between nodes. By answering this question, we may provide a clear solution for choosing between delaying or forcing transmissions for instance.

5.2 Vicinity Motion-based Markovian Heuristic

We extract knowledge from vicinity motion modeling to insert in our heuristic. However the asynchronous vicinity motion version previously presented does not include a strict time dimension. With our heuristic, since we plan on predicting the pairwise distances for the following steps, we need to have a stricter step duration control. To obtain stronger predictions, we use SVM – a time-aware version of vicinity motion.

5.2.1 Synchronous vicinity motion (SVM)

Vicinity motion helps us understand the logic of movements within a node's vicinity. In this chapter, we will be collecting transition knowledge from AVM (presented in the previous chapter) and also from the synchronous vicinity motion (SVM) to use in our heuristic. SVM is defined below:

- **Synchronous vicinity motion.** SVM probes pairwise shortest distances every τ time units (a step has now a fixed duration of τ seconds). The synchronicity of analyzing the network every τ seconds may result in higher probing costs than AVM who follows network dynamics. The choice of the τ parameter is very important. On the one hand, choosing a small τ results in few changes as the vicinity is likely to evolve. On the other hand, the choice of a high τ may skip important events. A step lasts τ seconds.

Note that for AVM a step consists in a period without change in the considered pairwise distance. For SVM a step consists in an interval of τ seconds. In the following paragraphs, we make a short recap on the general vicinity motion functioning.

Vicinity motion quick summary. We model vicinity motion through Markov processes for *each* pair of nodes. For a given node i, let $X_{i,j}^s$ be the random variable representing the distance between nodes i and j at step s. Both vicinity motions share the same Markov states. However, their transitional probabilities differ completely.

- **States.** The states of the Markov chain depends on the κ we choose, i.e., the size of the vicinity we wish to monitor. There are $\kappa + 1$ states; the first state, denoted '∞', corresponds to the case where the two nodes are in κ-intercontact, $\{1\}$ represents a contact and the remaining states $\{2, \ldots, \mathsf{M}\}$ correspond to a situation of κ-contact.

  We assume that X satisfies the Markov property and $X_{i,j}^s$ is independent of $X_{i,j}^{s-1}$.

- **Transitional probabilities.** To understand vicinity motion, we focus on the markov chain transitional rates between states, i.e., the probability of two nodes being at a distance of m at step s knowing that they were at a distance n in the

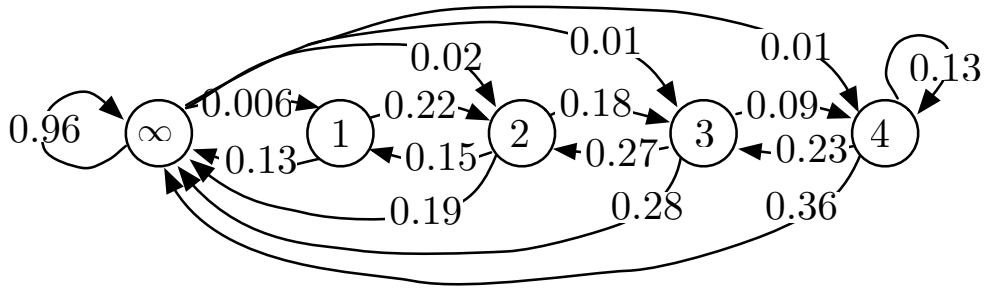


Figure 5.2: *Infocom05* average synchronous vicinity motion for a pair (i, j) and $\kappa = 4$, time slot $\tau = 200$ seconds. For the sake of clarity, we only displayed a few existing transitional probabilities.

previous step: $\mathbb{P}(X^s_{i,j} = m \mid X^{s-1}_{i,j} = n)$. AVM has a 0 probability of remaining in the same state while this probability is usually non null in SVM models.

As an example, we show in Fig. 5.2 the average transitional probabilities of SVM for *Infocom05*. For the sake of clarity, we omit certain transitions. As we can see, when nodes i and j are in κ-intercontact (∞), the probability that they meet directly is 0.6% while it is 2% for distance 2.

5.2.2 Heuristic

The Markov chain model in itself offers a future state prediction model. When we have the average transition probabilities from one state to another of the corresponding SVM, we follow the model evolution to obtain the probability of arriving at any state in the future slotting step. Using transition matrices T and the initial position vector, i.e., at what distance the two nodes are at the beginning, we can infer future steps movement probabilities. Not only can we do it for the future interval/step but also for several steps later.

The following heuristic allows us to predict the state i.e., the distance, between two nodes n steps later, based on the current situation. For a given pair of nodes, we apply the position vector to the transition matrix and deduce the probability of being in any state at the future step. We remind the reader that for AVM a step consists in a period without change in the considered pairwise distance. For SVM a step consists in an interval of τ seconds. This technique provides the probability for the given nodes of being at state $S \in \{\infty, 1, 2, .., \kappa\}$ at the n^{th} future step. The calculus follows:

$$\overrightarrow{\mathbb{P}^n_{Next}} = \overrightarrow{V_{pres}} \times T^n, \tag{5.1}$$

$\overrightarrow{V_{pres}}$ is the presence vector indicating the state where two nodes currently are. For example, given $\kappa = 5$, for the SVM model, the vector [0, 0, 1, 0, 0, 0] indicates that two

nodes currently are at a 2-hop distance (state 2). The presence vector [1, 0, 0, 0, 0, 0] shows that the two nodes are in κ-intercontact (state ∞). T^n is the corresponding SVM transition matrix of size $(\kappa+1)\times(\kappa+1)$ to the power $n, n \in \mathbb{N}^*$. $\overrightarrow{\mathbb{P}^n_{Next}}$ is the probability vector of being at each state $\{\infty, 1, 2, 3, 4, 5\}$ in the following n^{th} step.

Using the resulting $\overrightarrow{\mathbb{P}^n_{Next}}$ vector, we extract the highest probability state to derive the most plausible state prediction. However, given the nature of opportunistic networking, the connectivity graph is far from fully connected and most of times for the datasets we evaluate, a given pair of nodes is in κ-intercontact (∞). To better detect κ-contact events, we choose to also consider the second highest probability state as a potential prediction. The proposed heuristic outputs two states: the first highest probability S_f and the second one S_s.

5.2.3 Implementation

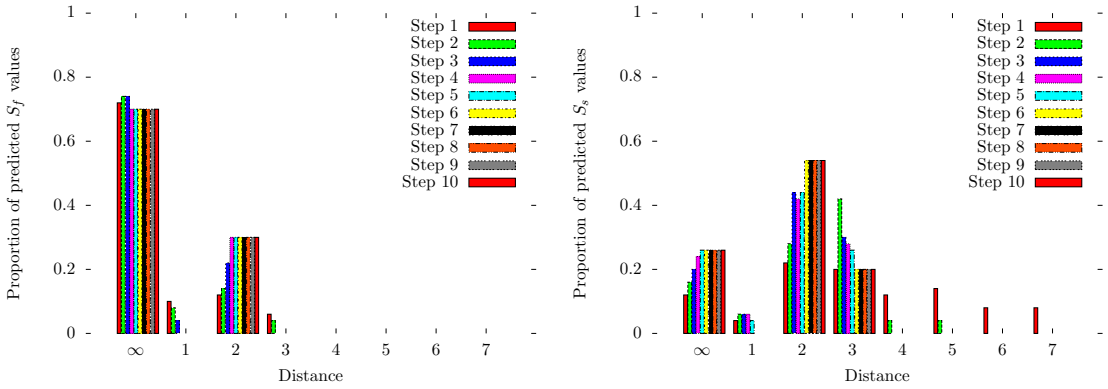
We supply the implementation of our heuristic in the Vicinity package introduced in the previous chapter. It relies on the transitional probabilities provided by the vicinity motion module and is implemented using Python $\geq$ 2.7 and the NumPy library.

5.3 Methodology

To evaluate the performance of our heuristic, we used the vicinity motion transitional probabilities values and pre-computed the prediction values S_f and S_s for each dataset, any initial position $\leq \kappa$, and n following steps ($n \in \{1, .., 10\} \cup \infty$). This means that for any pairwise initial distance between 1 and κ, our heuristic predicts two potential distance (S_f, S_s) for the n^{th} future step. Then for each dataset and all of their pairwise timelines, we observe the hop sequence. For each hop value, we observe the distance value of the different n^{th} following step and compare them to the corresponding (S_f, S_s) values. In this study, we choose to observe results for the n^{th} future steps with $n \in \{1, .., 10\}$. To evaluate the performance of our heuristic we will use two approaches:

- **Exact distance (`ED`).** If any of the two values (S_f, S_s) match, we consider the prediction to be accurate in an "exact" way. This shows how our heuristic is able to handle an exact distance prediction.

- **Upper bound distance (`UbD`).** If the real hop distance is below or equal to any of the two values (S_f, S_s), we deem the prediction to be accurate in an "upper" bound distance way. This shows how our heuristic handles vicinity presence prediction. The value $\max(S_f, S_s)$ gives an upper bound on the pairwise distance. Therefore, the prediction is accurate when the real future observed distance is below $\max(S_f, S_s)$ or equal to either S_f or S_s.



(a) AVM-full S_f prediction distribution.

(b) AVM-full S_s prediction distribution.

Figure 5.3: AVM-full prediction probability distribution.

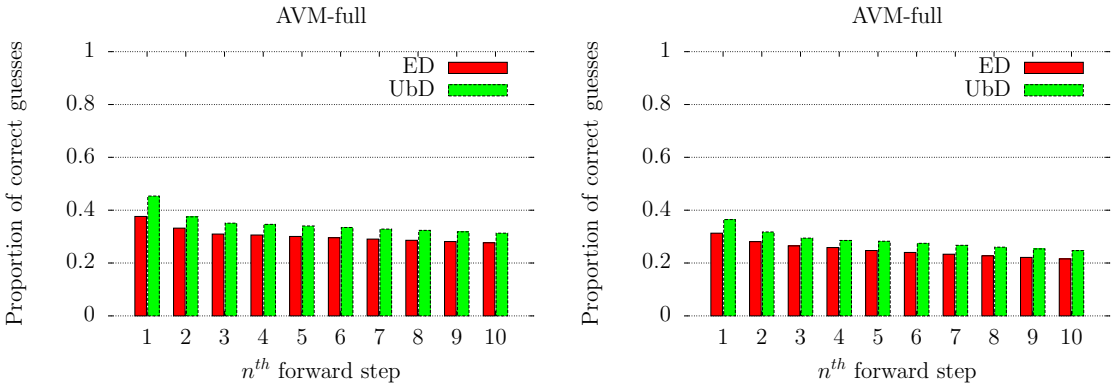
In the following sections, we evaluate the percentage of correct predictions in both `ED` and `UbD` modes. We begin by using the vicinity motion transitional rates computed over the full dataset duration. Then, we divide our datasets in half and use the first part as a *training set* and the second as a *test set*. We compute the averaged vicinity motion rates on the training set and observe its prediction capabilities on the test set.

5.4 Complete Knowledge Heuristic Evaluation

In this section, we compute vicinity motion transitional probabilities over full datasets.

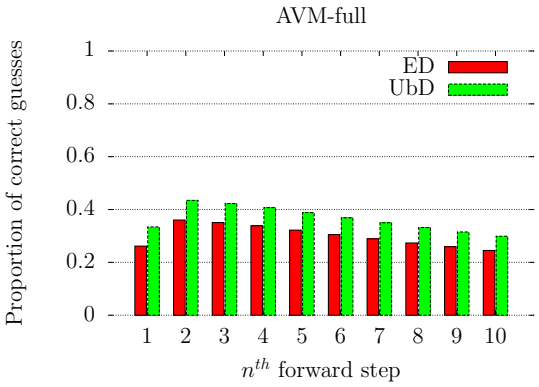
5.4.1 AVM-full

Prediction values distribution. In Fig. 5.3, we present the prediction probability distributions for S_f and S_s. On the x-axis, we present the expected distance. On the y-axis we have the probability of our heuristic predicting each distance for each of the future step. The AVM-full heuristic predicts mostly S_f values in the {0, 1, 2, 3} distance set, see Fig. 5.3(a). There is a more important part of distance ∞ prediction as the datasets we use in this study are often disconnected. The S_f predicted value is the one with the highest probability and has a clearly limited distance span. In Fig. 5.3(b), concerning S_s values, we observe a wider range of predicted values: from ∞ to {1,..,7}. In the case of the second highest probability output, there is a tendency to predict distance 2 or 3 more likely than any other distance. In both cases, the output is more likely to be ∞ or low distance values like 2 or 3. Note that the predicted value is hardly ever 1 indicating contact.

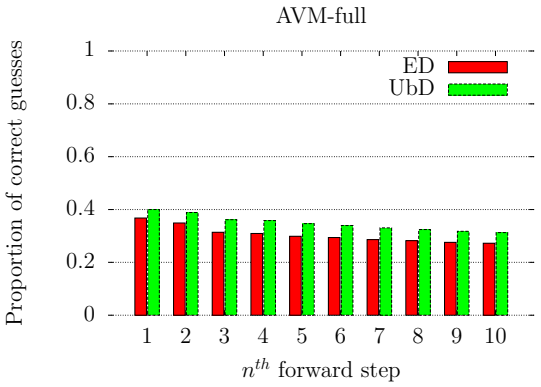


(a) *Infocom05*

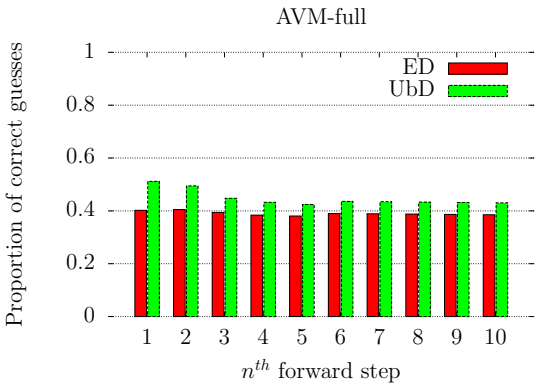
(b) *Sigcomm09*



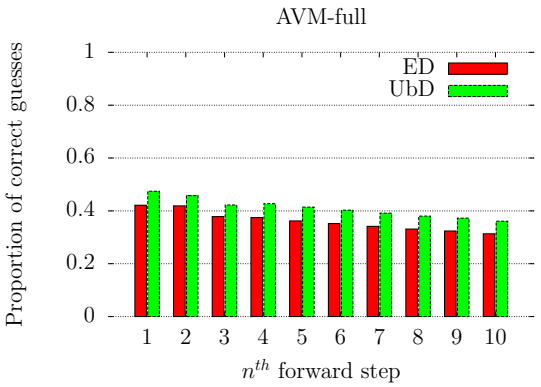
(c) *Rollernet*



(d) *Unimi*

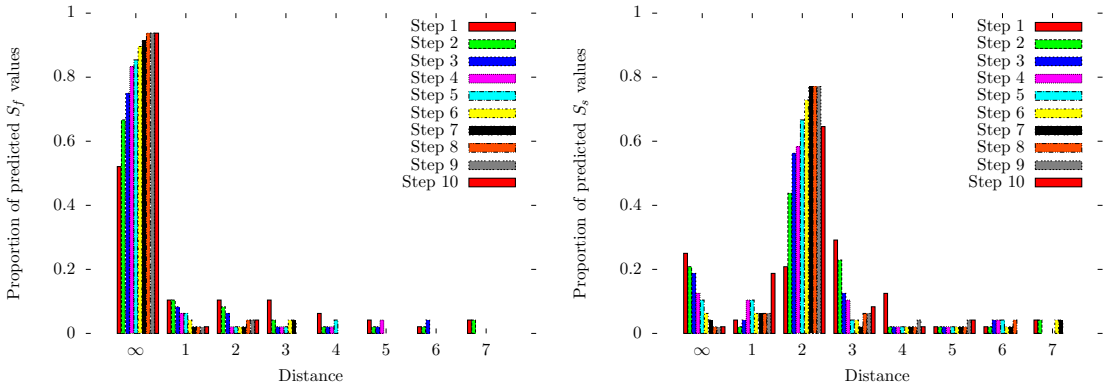


(e) *RT*



(f) *Shopping*

Figure 5.4: AVM-full heuristic performances.



(a) SVM-full S_f prediction distribution.

(b) SVM-full S_s prediction distribution.

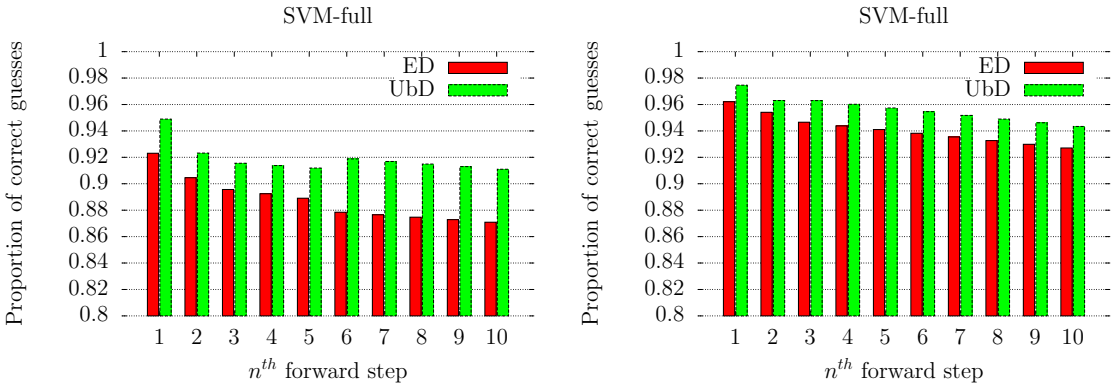
Figure 5.5: SVM-full prediction probability distribution.

The `ED` and `UbD` visions. In Fig. 5.4, we plot the proportion of accurate heuristic prediction for the following datasets: *Infocom05*, *Sigcomm09*, *Rollernet*, *Unimi*, *RT*, and *Shopping*. On the x-axis, we present the value of the n^{th} step. The y-axis indicates the proportion of accurate guesses our heuristic makes. We test our two evaluation parameters `ED` and `UbD`. For all our datasets, the `ED` metric gives performances between 24% and 42% of correct predictions. Most datasets (except *Rollernet*) have their prediction accuracy decrease gradually with higher step values n. But the results decreases of at most 12% between prediction for the next step ($n = 1$) or the 10^{th} next step ($n = 10$). *Rollernet* has a different progression curve with a lower value for the immediate next step ($n = 1$) than for the other steps. However, the sequence of remaining n values have the same evolution as for the other datasets. For `UbD` the prediction accuracy follows the same evolution as `ED`. The only difference being its higher results. On average it is 5% more efficient than `ED` prediction but it is less precise in terms of distance prediction.

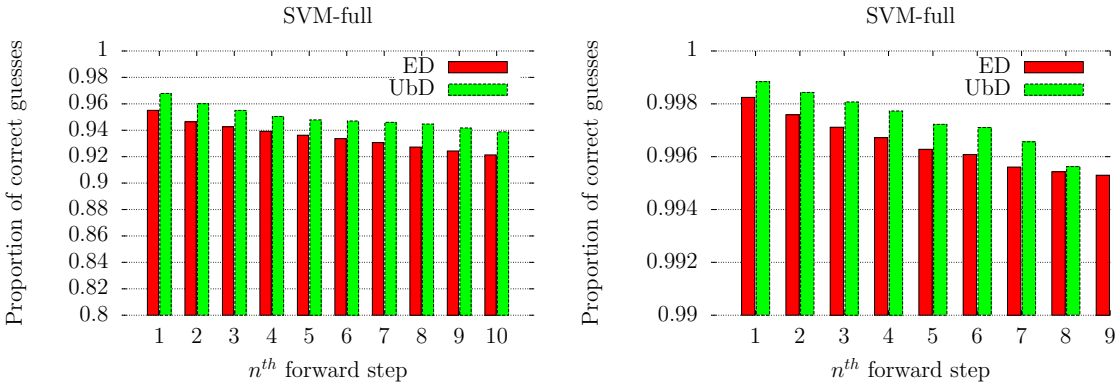
The exact distance prediction is tougher to get right than the upper distance bound. This feels natural as guessing a range is probabilistically easier than guessing an exact value. If we predict a large enough output value for the upper bound distance value, we may encompass the real observed value.

5.4.2 SVM-full

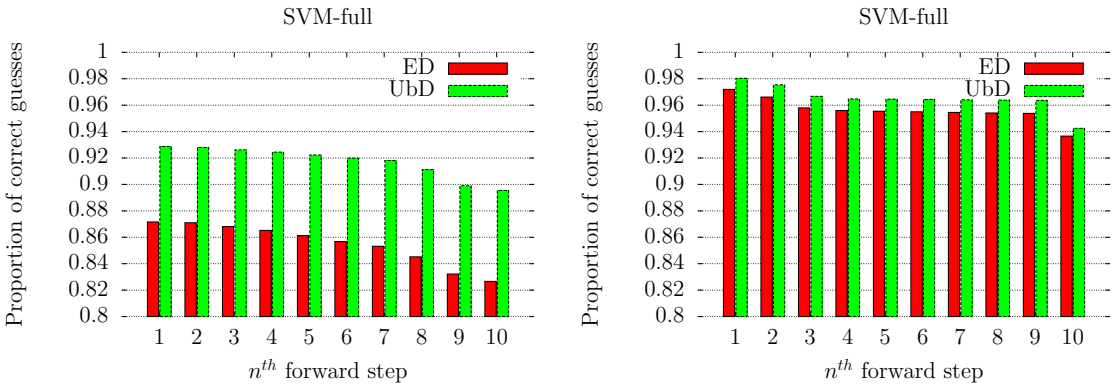
For the SVM-based heuristic, there is a crucial question before starting our analyses: which τ should we choose? τ being the interval duration between two network samples [20]. In the literature of opportunistic networks, most analyses using the datasets we use decided to analyses the network depending on their experimental sampling frequency (see Table 2.1 for more details on the sampling intervals). We decided to use the



(a) *Infocom05* (b) *Sigcomm09*



(c) *Rollernet* (d) *Unimi*



(e) *RT* (f) *Shopping*

Figure 5.6: SVM-full heuristic performances.

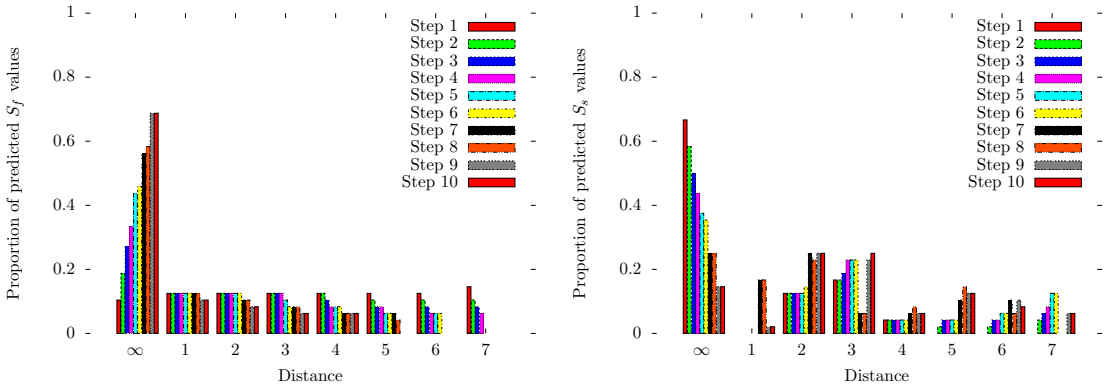
same approach here.[1] For all our datasets except *RT* and *Rollernet*, we used $\tau = 120$ seconds. *RT* and *Rollernet* have been slotted with a $\tau = 15$ seconds. Since the experiments that gathered this data used this sampling frequencies, there has not been any network change for durations lower than these. If we had chosen higher τ values, we might have skipped some interesting events.

Prediction values distribution. Fig. 5.5 shows the prediction probability distributions for S_f and S_s for the heuristic based on the SVM knowledge. The first observation is that the range of predicted values is larger for SVM than for AVM. In Fig. 5.5(a), we see that S_f predicted values go up to distance 7. While more than half of the predicted values indicate that the n^{th} next step distance will be ∞ independently of step value, we have a higher chance of predicting 1-contact than with the AVM heuristic. As for the AVM-based predictions, we find that whatever the n step value, the S_s predicted values have a very high chance to be 2 (Fig. 5.5(b)). We find that the tendencies observed with AVM predictions are enforced in SVM ones. This may be a simple effect of the slotting operation. If a κ-contact or κ-intercontact lasts longer than τ seconds we will observe two subsequent intervals with the same distance. This artificially increases the observed values and their incoming transitional probabilities in the SVM. However, we must not forget that SVM values share an idea of the time spent in each distance (each step is of length τ seconds) while AVM does not.

The `ED` and `UbD` visions. In Fig. 5.6, we present the percentage of correct guesses of the AVM-based heuristic. Note that the y-axes are represented on the range [0.8:1.0] instead of [0.0:1.0] else the difference between the metrics `ED` and `UbD` would be hard to notice. For *Unimi* it is even [0.99:1.0] as both approaches perform very closely. Concerning the evolution of both approaches, like with the AVM-full heuristic, the `ED` approach seem to decrease in efficiency with higher n values. However, the lowest correct prediction percentage is still of more than 82% of correct guesses. For the `UbD` technique, it follows the same trend with a better performance of around 2% on average.

The percentage of correct guesses with the SVM-heuristic is very high. This comes from the stability induced by the time slotting every τ time units and also stems from the fact that we often guess a ∞ distance between two nodes because datasets are not very connected. Considering the vicinity motion knowledge over the full datasets duration brings quite good performance levels in predicting the future distances between a pair of nodes. But we must remember that the part of guessing ∞ distance between two nodes represents an important part of our heuristic correct guesses. Still knowing that two nodes are not likely to be close by in a future step is very interesting in opportunistic

[1] However, to obtain a more optimal τ value, it would be useful to apply Latapy et al. approach with topological measures (e.g. degree centrality, betweenness centrality, Katz index, Adamic-Adar) we find important in our datasets [91, 92, 93]. For instance, applying Kolmogorov-Smirnov tests to observe if the two sampled measures originate from the same distribution could be an interesting indicator of optimal τ values [94].



(a) AVM-half S_f prediction distribution.

(b) AVM-half S_s prediction distribution.

Figure 5.7: AVM-half prediction probability distribution.

networks as such intelligence may allow a protocol to decide whether it should force or delay transmission or even completely switch to another networking paradigm. So far, we have used the knowledge of the full experiment before testing our heuristic. This allows our vicinity motion models to deeply capture vicinity behavior over time. In the next section, we wonder how our heuristic would perform if we used a training set and a test set.

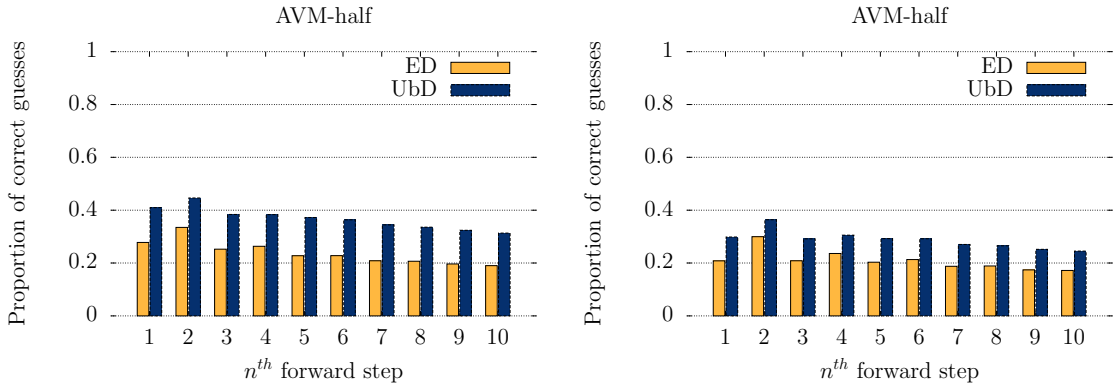
5.5 Partial Knowledge Heuristic Evaluation

We decided to divide our datasets in two equal parts and use the first half as a training set for vicinity motion knowledge while the second part would be our test set to observe heuristic performances.

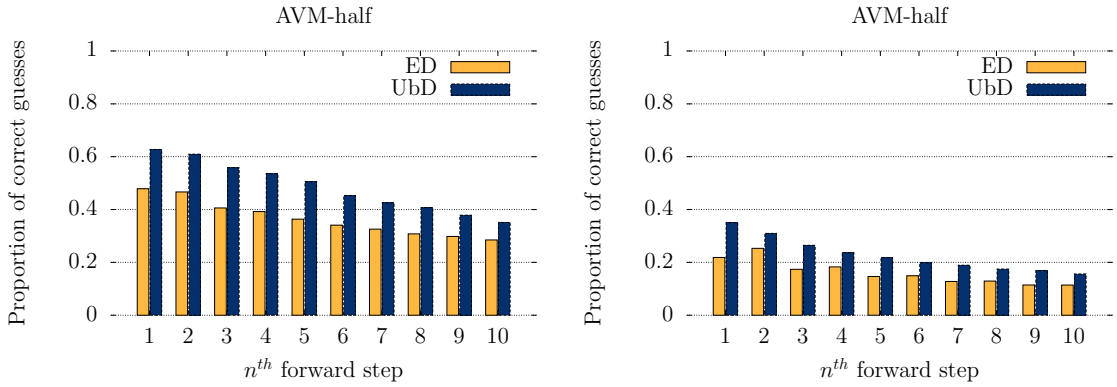
5.5.1 AVM-half

Prediction values distribution. The non-∞ predicted values here are uniform for the S_f values (see Fig. 5.7(a)). We still observe a higher tendency to generate infinite S_f values. Values of $S_f > 4$ tend to disappear for higher step values and be replaced by ∞. For S_s, we observe a large portion of ∞ and a few of $\{2, 3\}$ predicted distances and other distances up to 7 with a smaller presence.

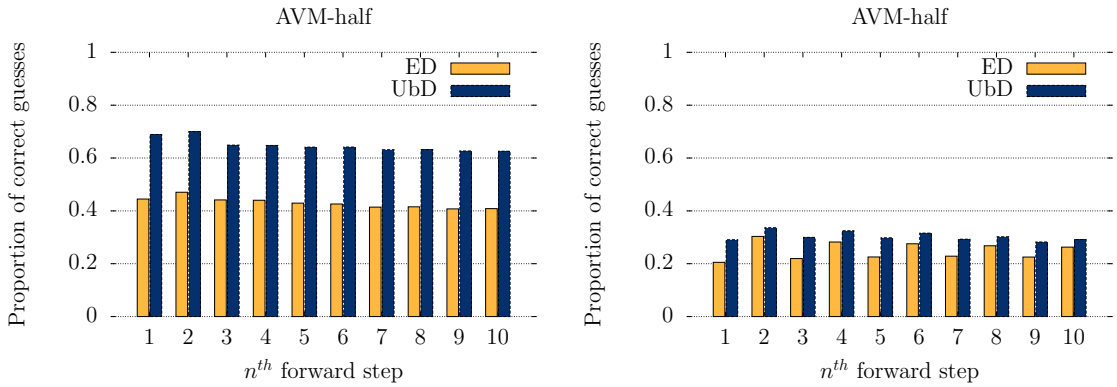
The `ED` and `UbD` visions. In Fig. 5.8, we show the performances of our heuristic when it relies on the AVM values when computed on the first half of each dataset. We use the remaining half to test the prediction capacities of our heuristic. We remind the reader that the AVM model gives more importance to pairwise distance evolution than to the temporal dimension. Compared to AVM-full based predictions, AVM-half has



(a) *Infocom05* (b) *Sigcomm09*

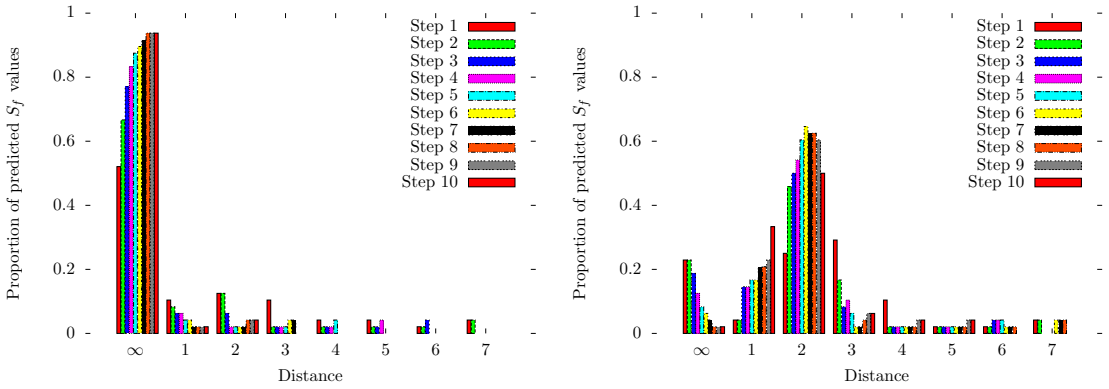


(c) *Rollernet* (d) *Unimi*



(e) *RT* (f) *Shopping*

Figure 5.8: AVM-half heuristic performances.



(a) SVM-half S_f prediction distribution.

(b) SVM-half S_s prediction distribution.

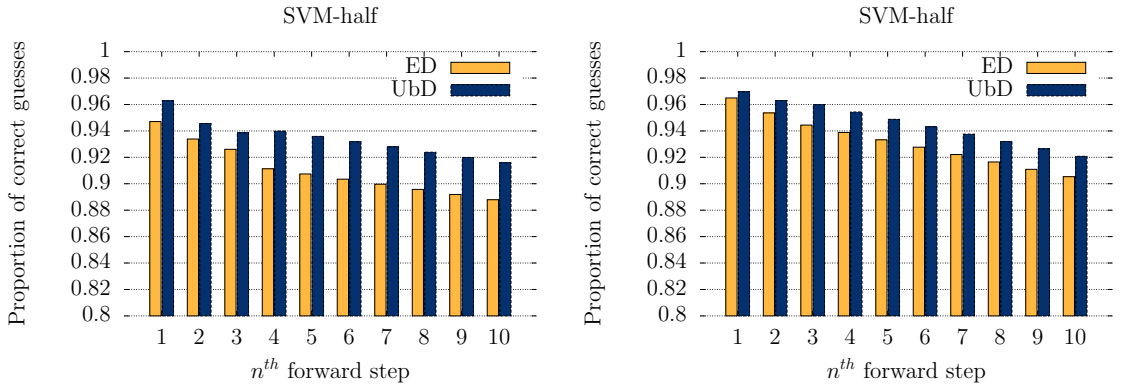
Figure 5.9: SVM-half prediction probability distribution.

less powerful performance for almost all datasets. Except for the *RT* experiment but this may come from the synthetic nature of *RT* and its random mobility. A borderline effect of the random mobility is that there are no exterior events affecting its mobility. This technique has at worst 11% of correct predictions in *Unimi* and 47% at best. For most of our datasets, we see that the larger the n value (the further in time the prediction), the lower the prediction accuracy. Except for *Shopping*, who has an up and down outlook. This can be explained by the experimental setting of *Shopping* and how the participants like vendors or shop owners have a very regular schedule when they stay in their shops but have a high variability during their free time. The `UbD` approach follows the same patterns as `ED` and it still has better performance ranging from 15% to 70%.

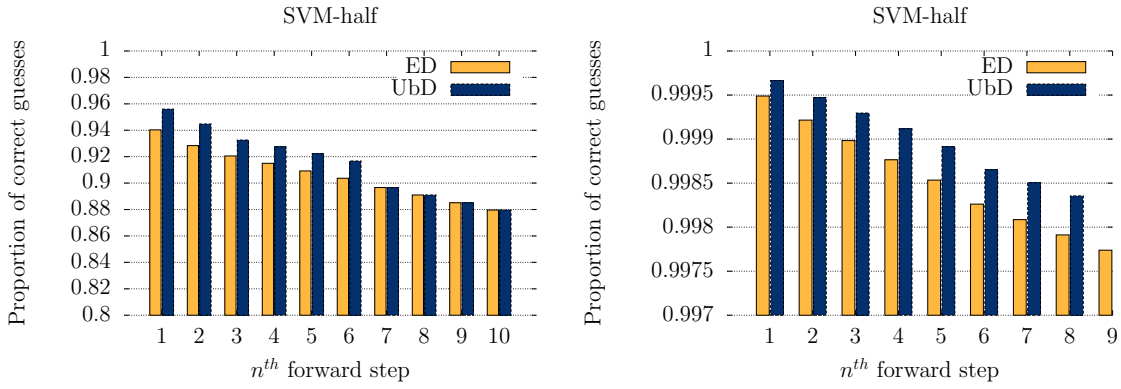
Concerning AVM-half performance, we observe that they are less efficient than AVM-full analyses. This shows that our division may not be optimal between the training set for AVM learning and the test set for heuristic testing. Our observations also show that our heuristic has correct guesses for 20% of the cases in average. This may look like a low value however other well known studies only achieve a 16% accuracy on the prediction and their prediction only apply to the next interval [59].

5.5.2 SVM-half

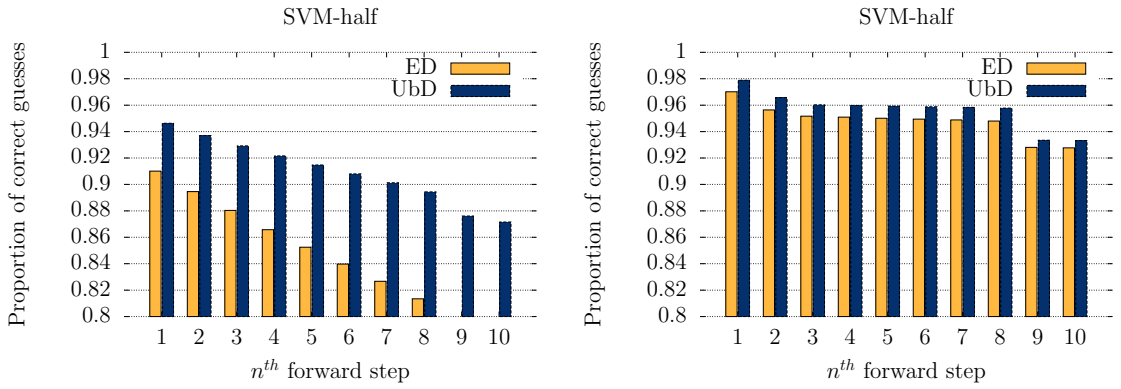
Prediction values distribution. In Fig. 5.9, the predicted values for S_f and S_s are very similar to SVM-full prediction distributions. By limiting the learning phase to half of the dataset duration, we obtain a similar knowledge than when we use the full duration. By slotting the dataset, we induce a distance stability over time which reduces distance changes (which is emphasized in the AVM model).



(a) *Infocom05* (b) *Sigcomm09*



(c) *Rollernet* (d) *Unimi*



(e) *RT* (f) *Shopping*

Figure 5.10: SVM-half heuristic performances.

The ED and UbD visions. In Fig. 5.10, we present the percentage of correct guesses for the SVM-half scenario. Interestingly, we find that SVM-half performs in average better than SVM-full for both ED and UbD. Similarly to previous analyses, larger values of n values bring lower prediction accuracy while UbD still performs better than ED as expected.

Given the performances of our SVM-based heuristics (SVM-full and SVM-half), we find that there is a clear potential for SVM-based forecasting in opportunistic networks. This approach could spark a distance prediction algorithm. Still, we must ponder the costs of vicinity probing that increases with vicinity probing frequency.

5.6 Conclusion

In this chapter, we raise the question of the predictability of vicinity behavior. We forecast vicinity pairwise distances using the inner vicinity motion knowledge since transitional probabilities capture vicinity movements patterns. If we were able to predict future distances between nodes, we could decide whether we should maintain an opportunistic routing or rely on other access networks or offloading techniques for instance. We use asynchronous and synchronous vicinity motions to bootstrap knowledge in our heuristic. AVM is the vicinity motion previously described which focuses on distance transitions within the κ-vicinity while SVM analyses states and transitions every τ seconds. SVM adds the temporal dimension lacking in AVM. Using the vicinity motion generated knowledge, we propose an heuristic predicting two distances for the next n^{th} future interval ($n \in \mathbb{N}^*$). Then, for the chosen datasets, we analyzed the accuracy of our approach using their vicinity movements. We find that SVM-heuristics performs better than AVM-based methods with their proportion of correct guesses reaching an interesting 99%. Yet, we have to mitigate this result by stating that our approach often predicts the lack of contact (∞) between nodes. Our scenario and most of DTN scenarios often bear poor pairwise connectivity. AVM-heuristics require less data to perform their predictions than SVM and may reach up to 40% of correct guesses which is one chance over 3 to obtain right predictions. Therefore, we find there is clear potential for vicinity motion based prediction in opportunistic networks.

Chapter 6

Conclusion & Perspectives

In our modern society, citizens tend to have more and more connected devices [95]. Through these devices, they require to always be connected to the current trends or news and they want to be able to communicate with other persons whether they are commuting, at work, at home, or even on holidays. This need of a "super-connectivity" shows its limits with the resulting telecommunication companies infrastructure overload [96]. On the lookout for alternative means to provide data to users, we find an attractive solution with the DTN paradigm. It is mostly user-based, does not need an overall infrastructure, takes advantage of user mobility as a transmission catalyst, and manages to deliver an interesting amount of information in the network. DTN has different characteristics from other networks like Wi-Fi, 2G, 3G or wired networks, therefore, their characteristics need to be thoroughly understood before we are able to use them. Many studies have shown the clear potential of DTN as a self-dependent network model. In this dissertation, we propose the utilization of the vicinity in DTN to improve its characterization, understanding, and functioning.

6.1 Summary of Contributions in this Thesis

Contribution 1: Uncovering Vicinity Properties of Intercontacts in DTNs. In this chapter, we questioned the binary assertion in opportunistic network. This usual binary contact – intercontact characterization in disruption-tolerant networks leads to a sub-optimal utilization of network transmission capacities. To address this issue and allow a better connectivity resource usage in DTNs, we enunciate a formal vicinity definition for opportunistic networks, namely the "κ-vicinity". We also defined the related concepts of "κ-contact" and "κ-intercontact" and analyze their overall temporal distributions. We show how their behavior depends on the ego density (sparse or dense). Getting vicinity knowledge is costly since we have to probe the network to obtain further connectivity information. Yet doing it 10 hops around us costs more than doing it only for contacts or at a 2-hop distance. In this contribution, we also analyze the connected components in the datasets. It showed how a κ value of $\{3, 4\}$ could be enough to leverage vicinity connectivity and maintain monitoring costs reasonable. Finally, we also show an example of vicinity utilization with the WAIT opportunistic protocol. According to our metric, we could improve performance by up to 80%.

Contribution 2: Digging into the Vicinity Dynamics of Mobile Opportunistic Networks. For this contribution, we developed a model capturing inner κ-vicinity movements called asynchronous vicinity motion. The asynchronous vicinity motion is a Markov chain using pairwise distance values as state values. Those states are linked to each other via their transitional probabilities to sequentially move from one distance to another. In this chapter, we identified three main movements patterns called *birth*, *death* and *sequential* movements. We show how sometimes by considering only death and sequential movements, we have more than 80% of vicinity movements. We also show how most κ-vicinity arrivals aka births occur at distance 3 or 4 and not in contact. Based on this observations and asynchronous vicinity motion transitional probabilities, we create TiGeR, a pairwise vicinity behavior generator who generate model vicinity behavior aka timelines. We can use these timelines to understand synthetic κ-vicinity functioning and test opportunistic protocols relying on κ-contacts. After investigating inner κ-vicinity behavior, we leverage vicinity motion capacities by using its transitional probabilities in a prediction scheme.

Contribution 3: Predicting Vicinity Dynamics. In this final chapter, we wonder how well our vicinity motion model captures vicinity dynamics. In order to observe this, we developed an heuristic based on vicinity motion transitional probabilities which predicts two possible output states in which a pair of node will find itself after n steps. Our heuristic answers the following questions: if two nodes are currently at a 3-hop distance, what are the two possible distances they will next be? What are the two distances they will be in n steps from now? Asynchronous vicinity motion lacks the time dimension in its collecting process, as a result its steps do not have a constant duration. Since our heuristic predicts pairwise distances for future steps, we feel the need to have a model with constant step durations for stronger predictions. Therefore, we propose a time-aware vicinity motion variant called synchronous vicinity motion. This model probes its surroundings every τ seconds and has steps of constant duration τ time units. We gather transitional data from asynchronous and synchronous approaches (AVM and SVM) and test the prediction capacities or our heuristic over our datasets. We use knowledge from full experimental durations as well as a training set – test set approach where we used the first part to gather vicinity motion data and the second part to test our heuristic. AVM-based heuristics provide around 30% of correct guesses, which is better than most prediction intents observed until now. In any case, we find that the SVM approach performs better with performance peaks at 99%. Yet we have to mitigate this result because most of the times, the predicted values in SVM are ∞. Our dataset being human driven, pairwise distances in SVM are often ∞. While some may consider this to be useless information, we find that it is quite helpful when it comes to choose whether you maintain a full opportunistic forwarding or rely on another type of networking like Wi-Fi or 3G to transmit data. AVM seems to be worse than SVM, however gathering knowledge for SVM is way costlier than for AVM. Both vicinity

motion models have their own advantages and drawbacks that have to be acknowledged in order to put the prediction scheme in action.

Contribution 4: The Vicinity Package implementation. The latest contribution of this thesis is transient to the three previous ones. We provide an implementation of our analyses in a Python package called "Vicinity". It is available at the following address: http://vicinity.lip6.fr. The Vicinity module has vicinity motion analyses, the TiGeR generator and an implementation of both AVM- and SVM-heuristics. It takes as inputs vicinity dynamics in the form of contact traces so it can be applied to any dataset with such knowledge.

6.2 General Remarks

In this dissertation, we investigated the impact of the vicinity in DTNs and opportunistic networks. In the current DTN vision, we found there was a grey area concerning end-to-end communication possibilities beyond direct contact. These transmission possibilities exist but they remain mostly unused in opportunistic networks. By defining κ-vicinity, we highlight their existence and how omnipresent they are. We know that probing vicinity is costly yet this notion can bring interesting additional information in DTN like κ-contact, vicinity movements or extended prediction schemes. All this information can be used in favor of a better DTN utilization. To reduce the costs of κ-vicinity probing, we found that κ values of $\{3,4\}$ may be enough to limit monitoring costs and obtain most vicinity events. We must underline that the observations we made in our work apply to the datasets we observed which are of different sizes and are held in different settings. However, we cannot tell how it would behave in bigger datasets with different densities and behaviors. We believe our observations would extend there yet; it still has to be observed whenever such datasets become available to our community. Applying our analyses should be easy since we implemented the appropriate tools in the Vicinity package. We hope our contributions as well as the provided implementation will be useful to our community for κ-vicinity integration in DTN.

6.3 Perspectives on Research Directions

In this section, we point out several research directions we deem interesting for future investigation concerning opportunistic networks.

Vicinity usage in routing protocols and vehicular networks. After showing the advantages of using κ-vicinity in the datasets we observed. We really feel that using the vicinity knowledge in routing protocol could bring sensible gains. This could take the form of a vicinity based protocol where routing would take into account knowledge belonging to the κ-vicinity or just an integration of vicinity knowledge in existing routing

approaches. For instance, Spray-and-wait could benefit from vicinity knowledge for a better next hop node choice. Vehicular networks embody another part of opportunistic networks that are also human driven but they display different mobility characteristics. Applying Vicinity analyses in their case would be interesting to see if our observations also apply or not for instance.

Opportunistic networks interoperability. DTN on their own have true potential, still they cannot cope with some delay constraints since it is against their main characteristic. To both keep the advantages of disruption-tolerant approaches and keep up with delay constraints, we find that the offloading schemes relying on 3G-infrastructure of Wi-Fi are interesting and more realistically applicable in real life. Applying vicinity knowledge in offloading scheme could be an interesting indicator on who to forward messages to.

Predictions in DTNs. Being able to predict future encounters in opportunistic networks is clearly a very interesting feature. Whoever knows what will happen next can finely tune its approach. There have been many attempts at doing so in a lot of different networks from citation networks, to mobile phone calls networks and even opportunistic networks. Because of opportunistic network high dynamicity, predictions in DTNs are very hard to perform. An interesting point of view would be to use the regularity of human schedule (day/night, work days/week ends, commuting periods) to extract specific prediction. Until now, most available datasets do not bring enough length or sizes to correctly perform such predictions. But in the future, performing these analyses would be very interesting and verifying if the heuristic we provide would work at this scale would be of high attraction.

Appendix A

List of Publications

Under review

- T. Phe-Neau, M. E. M. Campista, M. Dias de Amorim, and V. Conan, "Analyzing and Generating Vicinity Traces," submitted to *Elsevier Ad Hoc Networks*.
- T. Phe-Neau, M. Dias de Amorim, and V. Conan, "Uncovering Vicinity Properties in Disruption-Tolerant Networks," submitted to *Elsevier Computer Networks*.
- A. Tatar, T. Phe-Neau, M. Dias de Amorim, V. Conan, and S. Fdida, "Beyond Contact Predictions in Mobile Opportunistic Networks," submitted to the *Annual Conference on Wireless On-demand Network Systems and Services (IFIP/IEEE WONS)*.

Published

- T. Phe-Neau, M. E. M. Campista, M. Dias de Amorim, and V. Conan, "Padrões de Mobilidade de Vizinhança em Redes de Contato Intermitente (extended)," In *Revista Brasileira de Redes de Computadores e Sistemas Distribuídos (RB-RESD)*, accepted.
- T. Phe-Neau, M. E. M. Campista, M. Dias de Amorim, and V. Conan, "Examining Vicinity Dynamics in Opportunistic Networks", poster presented at *ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (ACM MSWiM)*, Barcelona, Spain, November 2013.
- T. Phe-Neau, M. E. M. Campista, M. Dias de Amorim, and V. Conan, "Padrões de Mobilidade de Vizinhança em Redes de Contato Intermitente," In *Brazilian Symposium on Computer Networks (SBRC)*, Brasilia, DF, Brasil, May 2013. **Best paper candidate.**
- T. Phe-Neau, M. Dias de Amorim and V. Conan. "The Strength of Vicinity Annexation in Opportunistic Networking," In *IEEE INFOCOM International Workshop on Network Science for Communication Networks (IEEE NetSciCom)*, Torino, Italy, April 2013.

- T. Phe-Neau, M. Dias de Amorim and V. Conan. "Caractérisation en diptyque de l'intercontact pour les réseaux à connectivité intermittente," In *Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (Algotel)*, La Grande Motte, France, May 2012.

- T. Phe-Neau, M. Dias de Amorim and V. Conan. "Vicinity-based DTN Characterization", In *ACM International Workshop on Mobile Opportunistic Networks (ACM MobiOpp)*, Zurich, Switzerland, March 2012.

- T. Phe-Neau, M. Dias de Amorim and V. Conan. "Fine-Grained Intercontact Characterization in Disruption-Tolerant Networks", In *IEEE Symposium on Computers and Communications (IEEE ISCC)*, Kerkyra, Greece, June 2011.

- T. Phe-Neau, M. Dias de Amorim and V. Conan. "Fine-Grained Intercontact Characterization in Intermittently-Connected Mobile Networks", poster presented at *École d'été RESCOM 2011*, La Palmyre, France, June 2011.

List of Figures

List of Tables

Bibliography

[1] V. Cerf, "The Internet is for Everyone," RFC 3271 (Informational), Internet Engineering Task Force, April 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3271.txt (Cited on page 1.)

[2] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," in *ACM Special Interest Group on Data Communication Conference (SIGCOMM)*, Karlsruhe, Germany, Aug. 2003. (Cited on page 2.)

[3] E. Goodman and E. Paulos, "The Familiar Stranger: Anxiety, Comfort, and Play in Public Places," in *ACM SIGCHI Conference on Human Factors in Computing Systems*, Vienna, Austria, Apr. 2004. (Cited on pages 3, 13 and 73.)

[4] N. Sarafijanovic-Djukic, M. Pidrkowski, and M. Grossglauser, "Island Hopping: Efficient Mobility-Assisted Forwarding in Partitioned Networks," in *IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, Reston, VA, USA, Sep. 2006. (Cited on pages 6 and 11.)

[5] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477–486, Aug. 2002. (Cited on page 6.)

[6] S. Heimlicher, M. Karaliopoulos, H. Levy, and T. Spyropoulos, "On Leveraging Partial Paths in Partially-Connected Networks," in *IEEE International Conference on Computer Communications (INFOCOM)*, Rio de Janeiro, Brazil, Aug. 2009. (Cited on pages 6, 11 and 69.)

[7] W. Gao and G. Cao, "On exploiting transient contact patterns for data forwarding in delay tolerant networks," in *IEEE International Conference on Network Protocols (ICNP)*, Washington, DC, USA, 2010, pp. 193–202. (Cited on pages 6 and 11.)

[8] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke University, Tech. Rep., 2000. (Cited on pages 9, 11 and 17.)

[9] A. Chaintreau, A. Mtibaa, L. Massoulie, and C. Diot, "The Diameter of Opportunistic Mobile Networks," in *ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, New York, NY, USA, Dec. 2007. (Cited on pages 9, 10 and 11.)

[10] E. Yoneki and D. Greenfield, "Inferring Significance of Meeting Groups in Human Contact Networks," in *European Conference on Complex Systems*, Lisbon, Portugal, Sep. 2010. (Cited on pages 9 and 17.)

[11] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE Rap: Social-Based Forwarding in Delay-Tolerant Networks," in *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, New York, NY, USA, Nov. 2008. (Cited on pages 9, 12, 17 and 62.)

[12] ——, "Bubble rap: Social-based forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 10, pp. 1576–1589, 2011. (Cited on page 9.)

[13] T. Hossmann, T. Spyropoulos, and F. Legendre, "Putting Contacts into Context: Mobility Modeling beyond Inter-Contact Times," in *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, May 2011. (Cited on page 9.)

[14] V. Conan, J. Leguay, and T. Friedman, "Characterizing Pairwise Inter-contact Patterns in Delay Tolerant Networks," in *International Conference on Autonomic Computing and Communication Systems*, Rome, Italy, Oct. 2007. (Cited on pages 10 and 45.)

[15] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 606–620, 2007. (Cited on pages 10 and 15.)

[16] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnović, "Power Law and Exponential Decay of Inter Contact Times between Mobile Devices," in *ACM International Conference on Mobile Computing and Networking (MobiCom)*, Montréal, Québec, Canada, Sep. 2007. (Cited on pages 10 and 25.)

[17] A. Passarella and M. Conti, "Characterising aggregate inter-contact times in heterogeneous opportunistic networks," in *IFIP Networking*, Valencia, Spain, May 2011. (Cited on pages 10 and 45.)

[18] A. Passarella, M. Conti, C. Boldrini, and R. Dunbar, "Modelling Inter-contact Times in Social Pervasive Networks," in *ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, Miami, Florida, USA, Oct. 2011. (Cited on page 10.)

[19] R. Calegari, M. Musolesi, F. Raimondi, and C. Mascolo, "CTG: A Connectivity Trace Generator for Testing the Performance of Opportunistic Mobile Systems," in *ACM SIGSOFT Symposium on the Foundations of Software Engineering*, Dubrovnik, Croatia, Sep. 2007. (Cited on pages 10 and 13.)

[20] V. Ramiro, E. Lochin, P. Senac, and R. Thierry, "On the limits of DTN monitoring," in *IEEE International Symposium on a World of Wireless Mobile and*

Multimedia Networks (WOWMOM), Madrid, Spain, Jun. 2013, p. 6. (Cited on pages 10 and 79.)

[21] V. Borrel, M. H. Ammar, and E. W. Zegura, "Understanding the Wireless and Mobile Network Space: A Routing-centered Classification," in *ACM MobiCom workshop on Challenged Networks (CHANTS)*, Montreal, Canada, Sep. 2007. (Cited on page 10.)

[22] J. Whitbeck and V. Conan, "Hymad: Hybrid dtn-manet routing for dense and highly dynamic wireless networks," *Computer Communications*, vol. 33, August 2010. (Cited on page 10.)

[23] S. Heimlicher and K. Salamatian, "Globs in the Primordial Soup – The Emergence of Connected Crowds in Mobile Wireless Networks," in *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Chicago, Illinois, USA, Sep. 2010. (Cited on pages 10, 11 and 34.)

[24] S. Scellato, C. Mascolo, M. Musolesi, and V. Latora, "On Nonstationarity of Human Contact Traces," in *IEEE ICDCS workshop on Simplifying Complex Networks for Practitioners*, Genoa, Italy, Jun. 2010. (Cited on page 10.)

[25] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro, "Time-varying graphs and dynamic networks," in *International Conference on Ad Hoc Networks and Wireless*, 2011, pp. 346–359. (Cited on page 11.)

[26] J. Whitbeck, M. Dias de Amorim, V. Conan, and J.-L. Guillaume, "Temporal Reachability Graphs," in *ACM International Conference on Mobile Computing and Networking (MobiCom)*, Istanbul, Turkey, Aug. 2012. (Cited on page 11.)

[27] A. Panisson, A. Barrat, C. Cattuto, W. Van Den Broeck, G. Ruffo, and R. Schifanella, "On the dynamics of human proximity for data diffusion in ad-hoc networks," *Elsevier Ad Hoc Networks*, vol. 10, no. 8, pp. 1532–1543, Nov. 2012. (Cited on page 11.)

[28] J. Ott, D. Kutscher, and C. Dwertmann, "Integrating DTN and MANET routing," in *ACM MobiCom workshop on Challenged Networks (CHANTS)*, Pisa, Italy, Sep. 2006. (Cited on page 11.)

[29] C. M. John Tang, Mirco Musolesi and V. Latora, "Characterising temporal distance and reachability in mobile and online social networks," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 1, pp. 118–124, January 2010. (Cited on page 11.)

[30] M. E. M. Campista, L. H. M. K. Costa, and O. C. M. B. Duarte, "A routing protocol suitable for backhaul access in wireless mesh networks," *Elsevier Computer Networks*, vol. 56, no. 2, pp. 703–718, Feb. 2012. (Cited on page 11.)

[31] W. Gao, G. Cao, T. La Porta, and J. Han, "On exploiting transient social contact patterns for data forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 1, pp. 151–165, Jan. 2013. (Cited on page 11.)

[32] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks," in *ACM SIGCOMM workshop on Delay-Tolerant Networking*, Philadelphia, Pennsylvania, USA, Aug. 2005. (Cited on pages 11 and 29.)

[33] ——, "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility," in *IEEE International Conference on Pervasive Computing and Communications*, Washington, DC, USA, 2007, pp. 79–85. (Cited on page 11.)

[34] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, July 2003. (Cited on pages 11 and 29.)

[35] S. Nelson, M. Bakht, and R. Kravets, "Encounter-based routing in dtns," in *IEEE International Conference on Computer Communications (INFOCOM)*, Rio de Janeiro, Brazil, 2009, pp. 846–854. (Cited on page 12.)

[36] G. Bigwood, D. Rehunathan, M. Bateman, T. Henderson, and S. Bhatti, "Exploiting Self-Reported Social Networks for Routing in Ubiquitous Computing Environments," in *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Avignon, France, Oct. 2008. (Cited on page 12.)

[37] A. Mtibaa, M. May, C. Diot, and M. Ammar, "PeopleRank: Social Opportunistic Forwarding," in *IEEE International Conference on Computer Communications (INFOCOM)*, San Diego, CA, USA, Aug. 2010. (Cited on page 12.)

[38] S. Gaito, E. Pagani, and G. P. Rossi, "Opportunistic Forwarding in Workplaces," in *ACM SIGCOMM workshop on Online Social Networks*, Barcelona, Spain, Aug. 2009. (Cited on pages 12 and 13.)

[39] ——, "Strangers help friends to communicate in opportunistic networks," *Elsevier Computer Networks*, vol. 55, no. 2, pp. 374–385, Feb. 2011. (Cited on page 12.)

[40] Q. Li, S. Zhu, and G. Cao, "Routing in Socially Selfish Delay Tolerant Networks," in *IEEE International Conference on Computer Communications (INFOCOM)*, San Diego, CA, USA, Aug. 2010. (Cited on page 12.)

[41] Y. Zhu, B. Xu, X. Shi, and Y. Wang, "A survey of social-based routing in delay tolerant networks: Positive and negative social effects," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 387–401, 2013. (Cited on page 12.)

[42] M. Vojnovic and A. Proutiere, "Hop limited flooding over dynamic networks." in *IEEE International Conference on Computer Communications (INFOCOM)*, Shanghai, China, 2011, pp. 685–693. (Cited on page 12.)

[43] R. Diana, E. Lochin, L. Franck, C. Baudoin, E. Dubois, and P. Gelard, "A DTN routing scheme for LEO satellites topology," in *IEEE Vehicular Technology Conference*, Québec, Canada, Sep. 2012. (Cited on page 12.)

[44] P. Raveneau, E. Chaput, R. Dhaou, E. P. Dubois, P. Gélard, and A.-L. Beylot, "Hybridisation of WSN and satellite," in *AIAA International Communications Satellite Systems Conference (ICSSC)*, Dec. 2011, pp. 1–9. (Cited on page 12.)

[45] S. Pal Chaudhuri, J.-Y. Le Boudec, and M. Vojnovic, "Perfect Simulations for Random Trip Mobility Models," in *IEEE International Conference on Computer Communications (INFOCOM)*, Miami, Florida, USA, Aug. 2005. (Cited on pages 12 and 16.)

[46] M. Musolesi and C. Mascolo, "A Community Based Mobility Model for Ad Hoc Network Research," in *REALMAN*, Florence, Italy, May 2006. (Cited on page 12.)

[47] ——, "Designing mobility models based on social network theory," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, pp. 59–70, July 2007. (Cited on pages 12, 16 and 44.)

[48] A. Munjal, T. Camp, and W. C. Navidi, "Smooth: a simple way to model human mobility," in *ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, Miami, Florida, USA, 2011, pp. 351–360. (Cited on page 12.)

[49] C. Boldrini, M. Conti, and A. Passarella, *User-centric Mobility Models for Opportunistic Networking*, ser. Lecture Notes in Computer Science. Springer, 2008, pp. 255–267. (Cited on page 12.)

[50] J. Whitbeck, M. Dias de Amorim, V. Conan, M. H. Ammar, and E. W. Zegura, "From encounters to plausible mobility," *Pervasive and Mobile Computing*, vol. 7, no. 3, pp. 206–222, Apr. 2011. (Cited on page 12.)

[51] F. Ekman, A. Keränen, J. Karvo, and J. Ott, "Working Day Movement Model," in *ACM SIGMOBILE workshop on Mobility Models*, Hong Kong, China, May 2008. (Cited on page 13.)

[52] R. Stanica, E. Chaput, and A.-L. Beylot, "Simulation of Vehicular Ad-Hoc Networks: Challenges, Review of Tools and Recommendations," *Elsevier Computer Networks*, vol. 55, no. 14, pp. 3179–3188, Oct. 2011. (Cited on page 13.)

[53] R. Albert, H. Jeong, and A. L. Barabasi, "The diameter of the world wide web," *Nature*, vol. 401, pp. 130–131, 1999. (Cited on page 13.)

[54] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins, "Microscopic evolution of social networks," in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2008, pp. 462–470. (Cited on page 13.)

[55] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: a social network perspective," in *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2009, pp. 299–308. (Cited on page 13.)

[56] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2011, pp. 1082–1090. (Cited on page 13.)

[57] M. Boc, A. Fladenmuller, and M. Dias de Amorim, "Otiy: Locators Tracking Nodes," in *ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, New York, NY, USA, Dec. 2007. (Cited on page 13.)

[58] S. Scellato, A. Noulas, and C. Mascolo, "Exploiting place features in link prediction on location-based social networks," in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2011, pp. 1046–1054. (Cited on page 13.)

[59] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *J. Am. Soc. Inf. Sci. Technol.*, vol. 58, no. 7, pp. 1019–1031, May 2007. (Cited on pages 13 and 84.)

[60] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi, "Human mobility, social ties, and link prediction," in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2011, pp. 1100–1108. (Cited on page 13.)

[61] M.-H. Zayani, V. Gauthier, I. Slama, and D. Zeghlache, "Tracking topology dynamicity for link prediction in intermittently connected wireless networks," in *IEEE International Wireless Communications and Mobile Computing Conference (IWCMC)*, Limassol, Cyprus, 2012, p. 7. (Cited on page 13.)

[62] M.-H. Zayani, V. Gauthier, and D. Zeghlache, "Improving link prediction in intermittently connected wireless networks by considering link and proximity stabilities," in *IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WOWMOM)*, San Francisco, CA, USA, 2012, p. 9. (Cited on page 13.)

[63] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010. (Cited on page 14.)

[64] X. Lu, E. Wetter, N. Bharti, A. J. Tatem, and L. Bengtsson, "Approaching the Limit of Predictability in Human Mobility," *Nature*, vol. 3, Oct. 2013. (Cited on page 14.)

[65] T. Hossmann, F. Legendre, and T. Spyropoulos, "From Contacts to Graphs: Pitfalls in Using Complex Network Analysis for DTN Routing," in *IEEE International Workshop on Network Science For Communication Networks (NetSciCom)*, Rio de Janeiro, Brazil, Apr. 2009. (Cited on page 14.)

[66] P. O. Vaz de Melo, A. C. Viana, M. Fiore, K. Jaffrès-Runser, F. Le Mouel, and A. A. Loureiro, "Recast: Telling apart social and random relationships in dynamic networks," in *ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2013. (Cited on page 14.)

[67] P. Nikolopoulos, T. Papadimitriou, P. Pantazopoulos, M. Karaliopoulos, and I. Stavrakakis, "How Much *off-center* Are Centrality Metrics for Routing in Opportunistic Networks," in *ACM MobiCom workshop on Challenged Networks (CHANTS)*, Las Vegas, Nevada, USA, Sep. 2011. (Cited on page 14.)

[68] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD data set cambridge/haggle (v. 2009-05-29)," Downloaded from http://crawdad.cs.dartmouth.edu/cambridge/haggle, May 2009. (Cited on page 15.)

[69] A.-K. Pietilänen and C. Diot, "Dissemination in opportunistic social networks: the role of temporal communities," in *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Hilton Head, South Carolina, USA, 2012, pp. 165–174. (Cited on page 15.)

[70] A.-K. Pietilainen and C. Diot, "CRAWDAD data set thlab/sigcomm2009 (v. 2012-07-15)," Downloaded from http://crawdad.cs.dartmouth.edu/thlab/sigcomm2009, Jul. 2012. (Cited on page 15.)

[71] P.-U. Tournoux, J. Leguay, F. Benbadis, J. Whitbeck, V. Conan, and M. D. de Amorim, "Density-aware routing in highly dynamic DTNs: The rollernet case," *IEEE Transactions on Mobile Computing*, vol. 10, pp. 1755–1768, 2011. (Cited on pages 15 and 22.)

[72] J. Leguay and F. Benbadis, "CRAWDAD data set upmc/rollernet (v. 2009-02-02)," Feb. 2009. (Cited on page 15.)

[73] S. Gaito, E. Pagani, and G. P. Rossi, "Fine-Grained Tracking of Human Mobility in Dense Scenarios," in *IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, Rome, Italy, Jun. 2009. (Cited on page 15.)

[74] P. Meroni, S. Gaito, E. Pagani, and G. P. Rossi, "CRAWDAD data set unimi/pmtr (v. 2008-12-01)," Downloaded from http://crawdad.cs.dartmouth.edu/unimi/pmtr, Dec. 2008. (Cited on page 15.)

[75] A. Galati and C. Greenhalgh, "Human mobility in shopping mall environments," in *ACM International Workshop on Mobile Opportunistic Networks (MobiOpp)*, Pisa, Italy, 2010, pp. 1–7. (Cited on page 15.)

[76] ——, "CRAWDAD data set nottingham/mall (v. 2013-02-05)," Downloaded from http://crawdad.cs.dartmouth.edu/nottingham/mall, Feb. 2013. (Cited on page 15.)

[77] M. Salathé, M. Kazandjieva, J. W. Lee, P. Levis, M. W. Feldman, and J. H. Jones, "A high-resolution human contact network for infectious disease transmission," *PNAS*, vol. 107, no. 50, pp. pp. 22 020–22 025, 2010. (Cited on page 16.)

[78] M. Salathe, M. Kazandjieva, J. W. Lee, P. Levis, M. W. Feldman, and J. H. Jones, "Salathe Group data set stanfordhigh," Downloaded from http://www.salathegroup.com/guide/school_2010.html, Dec. 2010. (Cited on page 16.)

[79] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," in *IEEE International Conference on Computer Communications (INFOCOM)*, vol. 2, 2003, pp. 1312–1321. (Cited on page 16.)

[80] S. Pal Chaudhuri, J.-Y. Le Boudec, and M. Vojnovic, "Mobility Model for random trip," Downloaded from http://ica1www.epfl.ch/RandomTrip/, Nov. 2005. (Cited on page 16.)

[81] M. Musolesi and C. Mascolo, "Mobility Model for social network founded mobility models for ad hoc network research," Downloaded from http://www.cl.cam.ac.uk/research/srg/netos/mobilitymodels/index.html, May 2006. (Cited on page 16.)

[82] T. Phe-Neau, M. Dias de Amorim, and V. Conan, "Vicinity-based DTN Characterization," in *ACM International Workshop on Mobile Opportunistic Networks (MobiOpp)*, Zurich, Switzerland, Mar. 2012. (Cited on page 20.)

[83] ——, "Caractérisation en diptyque de l'intercontact pour les réseaux à connectivité intermittente," in *Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (Algotel 2012)*, May 2012. (Cited on page 20.)

[84] ——, "Fine-Grained Intercontact Characterization in Disruption-Tolerant Networks," in *IEEE Symposium on Computers and Communication (ISCC)*, Kerkyra, Greece, Jun. 2011. (Cited on pages 20, 22 and 46.)

[85] ——, "The Strength of Vicinity Annexation in Opportunistic Networking," in *IEEE International Workshop on Network Science For Communication Networks (NetSciCom)*, Torino, Italy, Apr. 2013. (Cited on pages 21, 55, 62 and 73.)

[86] Y. Wang and H. Wu, "Delay/fault-tolerant mobile sensor network (dft-msn): A new paradigm for pervasive information gathering," *IEEE Transactions on Mobile Computing*, vol. 6, no. 9, pp. 1021–1034, Sep. 2007. (Cited on pages 29 and 40.)

[87] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*, 3rd ed. The MIT Press, 2009. (Cited on page 33.)

[88] C. Boldrini and A. Passarella, "HCMM: Modelling spatial and temporal properties of human mobility driven by users' social relationships," *Computer Communications*, vol. 33, no. 9, pp. 1056 – 1074, 2010. (Cited on page 44.)

[89] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, June 2008. (Cited on page 45.)

[90] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, Pasadena, CA USA, Aug. 2008, pp. 11–15. (Cited on page 47.)

[91] M. Latapy and C. Magnien, "Complex network measurements: Estimating the relevance of observed properties," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2008, pp. 1660–1668. (Cited on page 81.)

[92] L. A. Adamic and E. Adar, "Friends and Neighbors on the Web," *Social Networks*, vol. 25, no. 3, pp. 211–230, 2003. (Cited on page 81.)

[93] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39– 43, 1953. (Cited on page 81.)

[94] F. J. Massey, "The Kolmogorov-Smirnov test for goodness of fit," *Journal of the American Statistical Association*, vol. 46, no. 253, pp. 68–78, 1951. (Cited on page 81.)

[95] www.emarketer.com, "Smartphone adoption tips past 50% in major markets worldwide," 2013, `http://bit.ly/1axgpyk`. (Cited on page 87.)

[96] Financial Times by P. Taylor, "Data overload threatens mobile networks," 2012, `http://on.ft.com/1aWUNLb`. (Cited on page 87.)

Properties and Impact of Vicinity in Mobile Opportunistic Networks

Abstract: The market of mobile devices like smartphones, tablets, or laptops, has exponentially grown over the last few years. These devices have the necessary CPU and memory capacities to create, send, and forward information on the go. When people carry such equipments along their daily commuting, they become mobile information vector. They are able to carry, send, or receive information whenever they meet each other. The networking paradigm using such information vectors is known as disruption-tolerant networks (DTN) or opportunistic networks.

We begin by identifying and investigating the *binary assertion* issue in opportunistic networks. We notice how most DTNs mainly analyze nodes that are in contact. This vision implies that all nodes that are not in contact are in intercontact. Nevertheless, when two nodes are not in contact, this does not mean that they are topologically far away from one another. We propose a formal definition of vicinities in DTNs and study the new resulting "contact–intercontact" temporal characterization.

Then, we examine the internal organization of vicinities using the Asynchronous Vicinity Motion (AVM) framework. We highlight movement types such as birth, death, and sequential moves. We analyze a number of their characteristics and extract vicinity usage directions for mobile networks. Based on the vicinity motion outputs and extracted directions, we build the TiGeR (a synthetic TImeline GEneratoR) that simulates how pairs of nodes interact within their vicinities. Both module will be available in the Vicinity package that we provide on our dedicated website: http://vicinity.lip6.fr.

Finally, we inquire about the possibilities of vicinity movement prediction. We expose a vicinity motion-based heuristic for pairwise shortest distance forecasting. For this part, we also define a synchronous vicinity motion model (SVM) which is time-aware and analyzes datasets every τ seconds instead of following network dynamics like AVM. We find that our heuristics perform quite well with performances up to 99% for the synchronous vicinity motion-based scheme and around 40% for the asynchronous one.

Keywords: disruption-tolerant networks, opportunistic networks, DTN, vicinity, k-contact, k-intercontact, contact, intercontact.

Propriétés et impact du voisinage dans les réseaux mobiles opportunistes

Résumé: Notre décennie a connu une augmentation spectaculaire du taux d'équipement en smartphones, ordinateurs portables et tablettes multimedia. En 2013, quasi 50% des ménages sont pourvus de smartphones. L'information circule avec nous et peut être diffusée à partir de nous. Afin de profiter de ces nouveaux vecteurs de diffusions, la communauté scientifique définit de nouveaux types de réseaux de communications laissant de plus grands degrés de libertés aux communications. Ainsi les réseaux opportunistes ou réseaux tolérants aux interruptions (DTN) permettent d'utiliser ces nouveaux vecteurs de transmissions. Avant de pouvoir profiter de toutes les capacités des DTN, nous devons d'abord nous pencher sur la compréhension de ce nouveau paradigme. De nombreuses propriétés des réseaux DTN sont maintenant reconnues, cependant les relations entre un noeud du réseau et son voisinage proche ne semblent pas encore avoir été passée au crible. Dans la plupart des études que nous avons pu observer, la présence de noeuds voisins proches mais pas directement lié par le contact est souvent ignorée. Dans cette thèse, nous montrons à quel point considérer les noeuds à proximité nous aide à améliorer les performances des DTNs.

En identifiant et analysant le paradoxe binaire dans les DTN, nous montrons que les caractérisations actuelles basées sur la notion binaire de contact ou intercontact ne sont pas suffisantes pour bénéficier de toutes les possibilités de transmission dans les DTN. Nous proposons une définition formelle du voisinage pour les DTNs nommée le "κ-vicinity". Nous étudions les caractérisations temporelles du κ-vicinity dans différents jeux de données. Ensuite, nous nous sommes concentré sur l'étude de l'organisation interne du κ-vicinity. Nous avons crée le Vicinity Motion qui est un analyseur permettant d'obtenir automatiquement un modèle markovien du κ-vicinity à partir de n'importe quelle trace de contact. Nous avons pu extraire trois mouvements principaux dans les κ-vicinity: la naissance, la mort et les mouvements séquentiels. Grâce aux valeurs du Vicinity Motion, nous avons pu créer un générateur synthétique de mouvements de proximité nommé TiGeR. Enfin, nous posons la question de la prévisibilité des distances entre deux noeuds du κ-vicinity. En utilisant le savoir emmagasiné dans le Vicinity Motion, nous mettons au point une heuristique permettant de prédire les futures distances entre deux noeuds. La particularité de notre heuristique est qu'elle fournit deux distances possibles pour les n prochains intervalles considérés.

Mots-clés: Réseaux opportunistes, réseaux tolérant aux interruptions, voisinage, contact, intercontact.