HAL
open science

The Strength of Vicinity Annexation in Opportunistic Networking

Tiphaine Phe-Neau, Marcelo Dias de Amorim, Vania Conan

To cite this version:

Tiphaine Phe-Neau, Marcelo Dias de Amorim, Vania Conan. The Strength of Vicinity Annexation in Opportunistic Networking. Fifth International Workshop on Network Science for Communication Networks (NetSciCom 2013), Apr 2013, Torino, Italy. pp.217 - 222, 10.1109/INFCOMW.2013.6562898 . hal-00790012

HAL Id: hal-00790012

https://hal.science/hal-00790012

Submitted on 19 Feb 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The Strength of Vicinity Annexation in Opportunistic Networking

Tiphaine Phe-Neau[†], Marcelo Dias de Amorim[†], and Vania Conan[‡]

[†]CNRS and UPMC Sorbonne Universités [‡]Thales Communications
{tiphaine.phe-neau, marcelo.amorim}@lip6.fr vania.conan@fr.thalesgroup.com

***Abstract*—Most disruption-tolerant networking protocols available have focused on mere contact and intercontact characteristics to make forwarding decisions. We propose to relax such a simplistic approach and include multi-hop opportunities by annexing a node's vicinity to its network vision. We investigate how the vicinity of a node evolves through time and whether such information is useful when routing data. By analyzing a modified version of the pure WAIT forwarding strategy, we observe a clear tradeoff between routing performance and cost for monitoring the neighborhood. By observing a vicinity-aware WAIT strategy, we emphasize how the pure WAIT misses interesting end-to-end transmission opportunities through nearby nodes. Our analyses also suggest that limiting a node's neighborhood view to four hops is enough to improve forwarding efficiency while keeping control overhead low.**

***Index Terms*—Opportunistic networks, disruption-tolerant networks, contact, intercontact, vicinity.**

I. Introduction

As our urban society lives on, the more technologically nomadic its citizens get. During their daily commuting, people carry electronic devices like smartphones, portable game stations, or laptops. Previsions show how there will likely be more than 1 billion smartphones in 2016 [1]. Common devices embed wireless interfaces and important storage abilities traveling with their owners. The penetration of such technology on our daily life leads to new and increased mobile usages as well as new potential networking paradigm like disruption-tolerant networking (DTN) also known as opportunistic networking [2]. Devices like smartphones are ideal actors of DTN, they leverage people's mobility to carry information toward new places or new persons without requiring any help from an infrastructure, therefore avoiding the payment of a subscription to any provider.

Opportunistic networks rely on user mobility to store and forward information. Unlike usual wired or MANET networks, nodes in opportunistic networks inherently lack global network knowledge, as they are only aware of what they learned via encounters (aka *contacts*) [2]. Routing in DTN is thus challenging by nature. There are two solutions that bound the performance of routing strategies in such networks. On the one hand, the most efficient solution in terms of communication costs consists in waiting until the source meets the destination to transfer the message (at the cost of longer delays) [3]. On the other hand, full epidemic forwarding (flooding) yields the shortest delay but generates the costliest traffic overhead [4]. In between, other solutions such as PRoPHET or Spray-and-Wait employ a wiser strategy [5], [6]. Nodes choose their next hops based on probabilistic likeliness of meeting the destination or through distributed flooding.

All the solutions listed above share a common characteristic – they consider that whenever nodes are not in contact they are in *intercontact*. As we will see later on, this leads to suboptimal results as nodes are likely to miss transfer opportunities when the destination of a message is nearby but not in direct contact. In this paper, *we provide elements toward the adoption of a node's close vicinity as a more appropriate mean to help deliver messages in DTN*. The idea is to leverage short-length, multi-hop paths whenever possible to achieve immediate message delivery while keeping signaling overhead low. The motivation behind our work is that nodes that show interest to communicate are likely to occupy similar geographic areas, even if not within direct communication range [7]. We provide the following contributions:

- We propose and evaluate the interest of extending vicinity knowledge beyond one hop. To this end, we analyze both real-world and synthetic mobility traces.
- We show the perks of using short-length multi-hop paths in the WAIT protocol and validate the strategy of annexing a node's vicinity as a routing asset.
- We investigate if there is any empirical tradeoff in terms of vicinity knowledge that allows better waiting times while constraining monitoring costs.

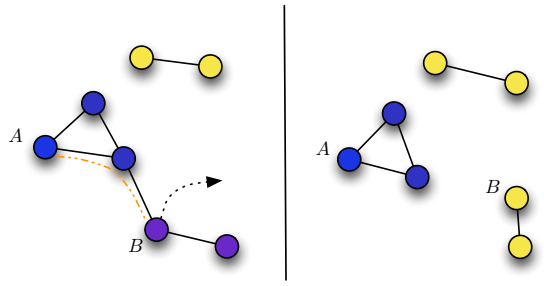
In the remainder of this paper, we first clarify the problem in Section II. Then, we define the notion of vicinity in DTN and enunciate the metrics used for our evaluation in Section III. We analyze the gains of vicinity annexation and its optimal setting in Section IV. In Section V, we list how our work relates to previous analyses and finally, we conclude the paper in Section VI.

II. Contact-based vs. Deeper Vicinity Vision

Currently, DTN protocols deduce their transmission opportunities using contact observations. We consider the traditional definition that a contact happens between two nodes whenever they are within each other's wireless range.

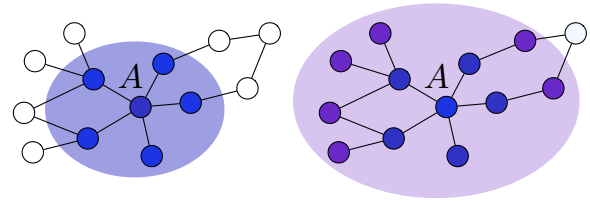
A. The WAIT protocol

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



(a) t_0 (b) t_1

Fig. 1. A motivating example where A wants to send a message to B. Fig. 1(a): at t_0, nodes A and B are at a 2-hop distance but not in contact. Fig. 1(b): at t_1, B moves away without ever coming in contact with A. A missed the opportunity to send its message while B was nearby. In this situation, vicinity annexation could help A. If A had known B was so close, it could have used the existing end-to-end path.



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

Fig. 2. Node A's κ-vicinity illustration, here, $\kappa = \{1, 2\}$. In Fig. 2(a), we represented node A's 1-vicinity. When $\kappa = 1$, A only knows nodes in contact with him. In Fig. 2(b), we have node A's 2-vicinity ($\kappa = 2$). A knows all nodes within a 2-hop distance and so on. A has an end-to-end path, of at most length κ, to any member of its κ-vicinity.

message or, worst, to completely fail delivering it. As we will see in Section IV, 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. The WAIT protocol is also known as Direct Transmission.

B. The pros and cons of monitoring contacts only

In a node-centric scheme, gathering contact information comes naturally. Any device can sense its surrounding and feel whenever other nodes are around through appropriate probing tactics. The application of contact knowledge in the WAIT protocol is straightforward: the source delivers a content to the destination when they get in contact with each other. In practice, a node may be collocated with many other nodes but it is not always or ever in direct contact with some of them. An example is illustrated in Fig. 1. At a given time t_0, A is at a 2-hop distance from B. Then at time t_1, B decides to leave and breaks any existing contemporaneous path from A to B. Let us consider that A knew the network topology at t_0. If A had known the path to destination B at t_0, it could have considered sending the message to B using multiple hops (in this example, only two), instead of waiting until meeting B (or trying some other non-deterministic strategy), which might take forever.

PRoPHET uses delivery predictabilities (preds) to evaluate the utility of using one node as a relay toward the destination. The protocol updates these preds only when the node comes in contact with other nodes and ages them as long as they move away (even if nearby). PRoPHET also uses a transitivity property to update its preds but mellows its impact via a β factor that can be quite low. Moreover, it does not give any freshness insights on this "transitive proximity". Concerning Spray-and-Wait, the distributed flooding approach sends L copies of a message to L different nodes, in somehow a "first in contact first served" fashion. As in the case of PRoPHET, Spray-and-Wait only uses knowledge about nodes that get in direct contact, without considering vicinity information beyond one hop that could be helpful.

C. Why not taking a look around?

DTN protocols rely on contacts as they are easy to gather, while extended neighborhood knowledge is more costly. Due to the DTN nature, offering nodes a consistent and full knowledge of the network topology is unrealistic. An alternative would be to make nodes have information on the connected component they are in. This would allow each node knowing with whom it has a contemporaneous path. However, as nodes do not know a priori the size of the connected component, it becomes difficult to limit the control overhead. Still, by limiting the scope of a node vision (to nodes up to a few hops, say two or three), we also limit the signaling overhead. Localized vicinity knowledge can be an important asset for DTN nodes. However, the main challenge in providing extended vicinity knowledge to nodes is how to find a good balance between efficiency and probing costs. In fact, we can wonder how far a node should probe its vicinity for surrounding knowledge. The more information about the network, the better decisions we can make but the higher the costs induced.

III. Waiting less with vicinity knowledge

We evaluate how the vicinity knowledge can help improve the responsiveness of a routing scheme for DTN. In this paper, we consider the particular case of the WAIT protocol because of its specific properties: upper bound in terms of delay and lower bound in terms of signaling overhead. We leave the evaluation with other protocols for future work.

A. The κ-vicinity

We define how nodes see their vicinity with the κ-vicinity. We assume that nodes monitor the presence of other nodes up to a distance of κ hops [7]:

Definition 1. ***κ-vicinity.*** *The κ-vicinity of a node A, noted κ_A, is the set of nodes with a contemporaneous end-to-end path of at most κ hops to A. All other nodes are considered outside κ_A.*

In Fig. 2, we show an example of node A's κ-vicinity for $\kappa = 1$ and $\kappa = 2$. The traditional definition of a contact

corresponds to the case of $\kappa = 1$. Now, whenever node A wants to send a message to node B, A scans its neighborhood up to κ hops. If B arrives within A's κ-vicinity, A can send the message to B via a multi-hop contemporaneous path, avoiding unnecessary waiting periods.

B. The WAIT protocol with extended knowledge

To observe the impact of vicinity awareness, we investigate the effects of neighborhood knowledge in the WAIT protocol by focusing on the waiting parameter – the time a node waits before being able to send a message straight to the destination. The secret, which is application-dependent, is how to balance delay and signaling cost. Recall that in the case of the WAIT protocol, the waiting time only stops when the source meets the destination (no intermediate relays), which is the worst case in terms of delay. The waiting time has a particular meaning as it is related to the user experience – any user notices the duration before its message gets delivered (if it gets delivered at all) and judges a service accordingly. Instead of analyzing delivery percentage, we chose to focus on waiting delays which is a direct representation of neighborhood observation raw advantages.

C. Costs

To take into account the costs of multi-hop messaging and neighborhood monitoring, we identified two main sources of overhead. We use the message as the unit of comparison.

Data Overhead (D_o): represents the total cost to deliver a message. Clearly, any protocol with extended neighborhood knowledge is costlier than its simple version. Whenever the source switches to multi-hop transmission mode, the message follows a contemporaneous end-to-end path to the destination and has to sustain several store and forward processes. The "extra" cost of such a communication, in terms of additional messages sent, is the number n of hops between A and B minus one:

$$D_o = n - 1. \tag{1}$$

Neighborhood Knowledge Overhead (N_o): represents the signaling overhead to gather information about the neighborhood. Node A broadcasts a discovery message (`DM`) to its contacts with a TTL set to κ. All nodes receiving the `DM` rebroadcast this message with a TTL set to $\kappa - 1$, and so on. We assume that each transmission is acknowledged (see Fig. 3 for a detailed example). This leads to a cost of:

$$N_o = ||\kappa_A|| + ||(\kappa - 1)_A|| + 1, \tag{2}$$

where $||\cdot||$ stands for cardinality. N_o does not depend on the path length that `DM`s have to cross. With little aggregation, N_o only depends on the number of neighbors in a node's connected component.

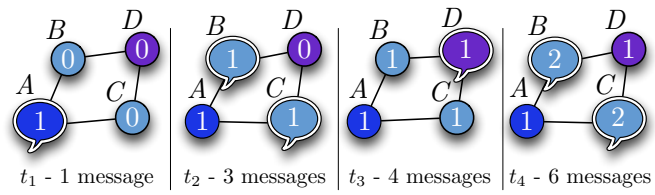


Fig. 3. Neighborhood knowledge discovery technique. At t_1, A ignites the discovery by broadcasting a message with a TTL set to 2. Its contacts, B and C receive the message. At t_2, they broadcast a message with a TTL set to $2 - 1 = 1$. At t_3, D received discovery messages with a TTL of 1, then broadcasts its reply. At t_4, B and C aggregate all replies they received and send their knowledge to A. In the end, we obtain 6 sent messages.

IV. Performance analysis

To evaluate the performance gains enabled by neighborhood awareness, we simulate the WAIT protocol for different values of κ (recall that $\kappa = 1$ corresponds to the basic WAIT protocol) using various mobility data.

A. Datasets

We performed our study with a wide range of datasets both synthetic and real-life based. Real-life experiments all involved devices (iMotes, T-Motes) carried by participants. These Motes logged the presence of other devices within a 10-meter range unless specified. Researchers derive contact intervals from these presence logs. We chose the following scenari captured during various experiments. As all these experiments are available to the community, we also provide the downloading links.

- **Infocom05** measurement was held during the Infocom 2005 conference [8], [9]. Researchers gave iMotes to 41 attendees. We study a 12-hour interval with the highest activity. Each iMote probes its environment every 120 seconds. *Infocom05* represents conference environment.
- **Rollernet** had 62 participants during a 3-hour dominical rollerblading tour in Paris [10], [11]. Motes probed their surrounding every 30 seconds. This dataset illustrates a dynamic sport event.
- **Unimi** involved 48 people among the students, faculty members, and staff from the University of Milano for two weeks in 2008 [12], [13]. Devices probed their surroundings every second. *Unimi* provides a longer academic and working situation.

We also generated scenari from two well-known mobility models:

- **Random Trip** is an improved Random-Waypoint mobility model [14], [15]. We generated the patterns of 20 nodes on a surface of 50 x 60 m^2 with speed between 0 and 7 m/s.
- **Community** is a mobility model reflecting human tendencies to aggregate in specific locations [16], [17]. We created 50 nodes with a 10m wireless range on a 1,500 x 2,500 m^2 plane during 9 hours.

TABLE I
AVERAGE NUMBER OF NEIGHBORS IN A NODE'S κ-VICINITY (WHOLE DATASET DURATION).

	κ							
Dataset	1	2	3	4	5	6	7	8+
Community	2.0	4.0	4.6	**4.7**	4.7	4.7	4.7	4.7
RandomTrip	2.0	3.2	4.7	**5.7**	6.3	6.7	6.9	7.1
Infocom05	1.5	3.8	5.3	**6.0**	6.4	6.4	6.4	6.4
Rollernet	1.4	3.2	4.7	**5.7**	6.3	6.7	6.9	7.0
Unimi	0.3	0.7	0.9	**1.1**	1.1	1.2	1.2	1.2

TABLE II
NEIGHBORS κ-DISTRIBUTION IN A NODE'S κ-VICINITY.

	κ							
Dataset	1	2	3	4	5	6	7	8+
Community	2.4	**2.3**	0.7	0.1	0.0	0.0	0.0	0.0
RandomTrip	2.3	2.3	2.0	**1.4**	0.8	0.4	0.2	0.1
Infocom05	3.0	4.4	3.0	**1.4**	0.7	0.2	0.1	0.0
Rollernet	2.0	2.5	2.1	**1.5**	0.9	0.6	0.3	0.2
Unimi	1.5	**1.0**	0.7	0.4	0.2	0.1	0.0	0.0

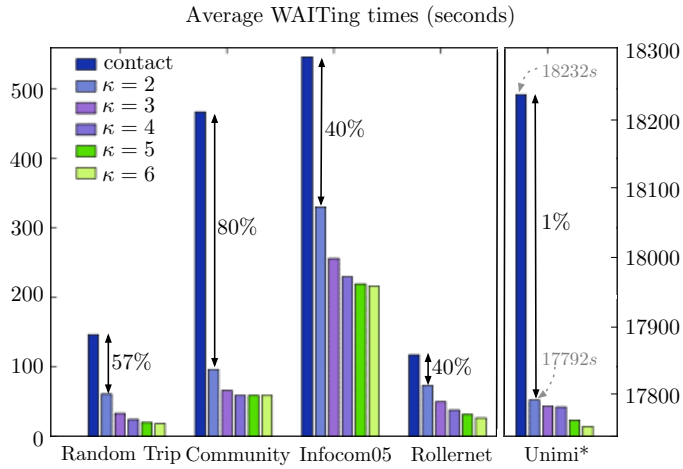


Fig. 4. Averaged waiting times according to the threshold κ. For all traces, there is a clear improvement between the first and second bar (contact only vs. 2-vicinity). Being aware of a node's κ-vicinity can lead to divide waiting times by 4 like in *Community*. The higher the κ, the better the waiting delays, yet, above $\kappa > 4$, gains become negligible. Note that, for the *Unimi** dataset we focused on its top values. 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).

B. Threshold Optimization

Neighborhood monitoring is an expensive process in opportunistic networks. To lower its costs, we investigate the optimal κ threshold. First, we consider the amount of people needed in the vicinity then we observe the difference between a static and a dynamic κ setting.

For each node, we analyze the average number of neighbors in their κ-vicinity. Table I 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 *RandomTrip* dataset, which has a random movement pattern and a high density). In *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 number of nodes.

In the next table, 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 (see Table II). In *Infocom05*, we see that on 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 hop, 1.4 at 4 hops etc. For *Community* and *Unimi*, the number of nodes at $\kappa > 2$ falls below 1. For *RandomTrip*, *Rollernet* 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.

C. Routing Effects

1) Loss & Delays: For each mobility trace and each pair of nodes, we randomly generated 10 messages at different time instants. We chose to generate sparse messages for waiting times to better reflect the impact of neighborhood monitoring. 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, 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 times.

For these nodes with *bounded waiting delays*, we analyze to which extent neighborhood knowledge helps lower their waiting times. In Fig. 4, 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: 40% in *Infocom05* and *Rollernet*, 57% in *RandomTrip*, and around 80% in *Community*. The *Unimi* dataset stands out because of its time scale. The

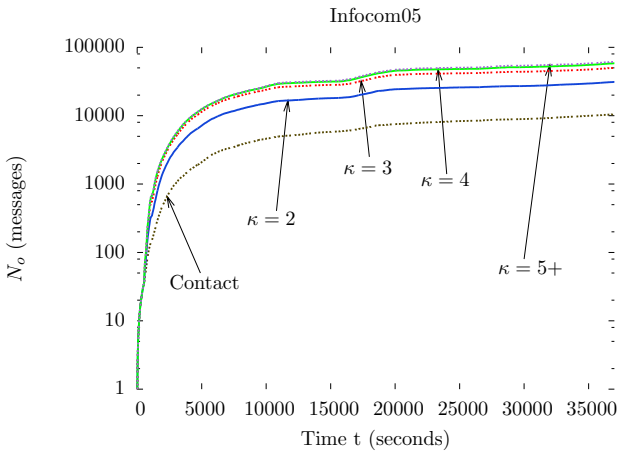


Fig. 5. Neighborhood Knowledge Overhead (N_o) in terms of message sent by the discovery technique `Reg` for a node in the *Infocom05* dataset. On average, probing κ-vicinity with $\kappa > 4$ costs as much as probing the 4-vicinity. This version of neighborhood probing is very expensive. Note the logscale on the y-axis.

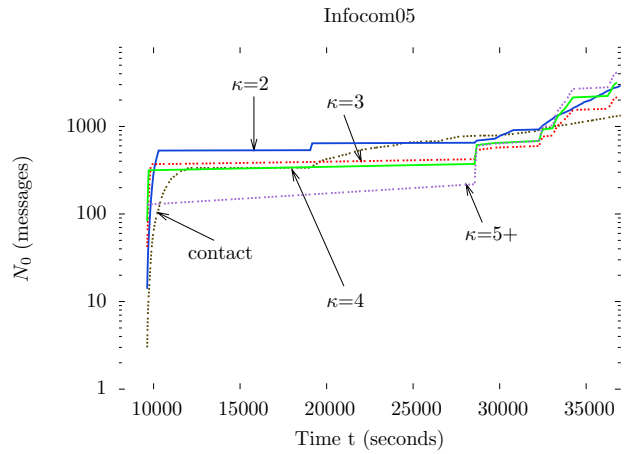


Fig. 6. Neighborhood Knowledge Overhead (N_o) using `OnD` for a pair of node in *Infocom05*. Contact monitoring drops $9/10^{th}$ of messages and keeps monitoring its contacts without being able to deliver any messages. For 7 delivered messages, sensing 3-vicinity (or beyond) ends up cheaper than observing 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.

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.

2) 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 naive behaviors:

- 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. 6, 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. 6, 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.

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.

V. Related and Future Work

Concerning DTNs, 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 [18]. 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 [19]. Later, Heimlicher and Salamatian laid their study over the groundwork that mobile wireless networks tend to have connected crowds [20]. 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 [21]. Tang *et al.* focused on the nature of these spatio-temporal paths to better understand how to use them [22]. Whitbeck *et al.* proposed an interesting way to capture a node's reachable vicinity through a new graph type [23].

In our work, the κ-vicinity reflects the existing topology and uses it to lower transmission delays. The κ-vicinity leverages immediate neighborhood structure further than techniques mentioned in the first paragraph and considers reachability in a node's connected component. We do no wait for potential contacts appearing later, we use existing links beyond mere contacts. As an additional example of use, in their latest paper Diana *et al.* applied a similar vicinity notion to satellite communications [24]. By leveraging neighboring stations, their proposal allowed valuable routing performance gains.

In Section IV-B, the recommended threshold of $\kappa = 4$ is static. Deploying a strategy with a dynamic κ threshold may be another alternative. As seen in Section IV-C, adapting your vicinity vision according to your needs allows considerable overhead gains. Determining the accurate policy for vicinity sensing may be linked to applications needs as well as performance requirements. As a future work, we would like to investigate the relationship between vicinity probing policies and local densities to provide

VI. Conclusion

In this paper, we examine the impact of vicinity awareness on the waiting time in a variation of the pure WAIT protocol. Most DTN techniques only focus on sensing direct contacts and do not inquire about their nearby neighbors. This strategy is the most straightforward and rational. However, we cannot deny the sociological nature of DTNs. People do not wander randomly in a city. They gather around specific persons or locations. To our opinion, ignoring a node's immediate neighborhood results in a loss of useful information.

Our findings show that neighborhood probing significantly improves performances of the WAIT protocol in terms of waiting delays. When delays used to be infinite, they are now bounded. When delays were high, they are now lowered by a factor up to four in our scenarios. Yet, neighborhood monitoring ignites messaging overhead. But, by limiting a node's vicinity knowledge to a threshold κ of four is an optimal setting. We constrain costs and still enhance performance. According to our observations, we have now enough motivation to consider potential performance gains that vicinity knowledge could bring to more elaborated DTN schemes.

Acknowledgment

This work is partially supported by the European Commission in the framework of the FP7 Mobile Opportunistic Traffic Offloading (MOTO) under grant agreement number 317959.

References

[1] Forrester Research, "Mobile is The New Face of engagement," 2012, http://www.forrester.com/go?objectid=RES60544.

[2] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," in *ACM Sigcomm*, Karlsruhe, Germany, Aug. 2003.

[3] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: the single-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 63–76, Feb. 2008.

[4] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke University, Tech. Rep., 2000.

[5] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, July 2003.

[6] 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.

[7] T. Phe-Neau, M. Dias de Amorim, and V. Conan, "Fine-Grained Intercontact Characterization in Disruption-Tolerant Networks," in *IEEE Symposium on Computers and Communication*, Kerkyra, Greece, Jun. 2011.

[8] 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.

[9] 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.

[10] 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.

[11] J. Leguay and F. Benbadis, "CRAWDAD data set upmc/rollernet (v. 2009-02-02)," Downloaded from http://crawdad.cs.dartmouth.edu/upmc/rollernet, Feb. 2009.

[12] 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.

[13] 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.

[14] S. PalChaudhuri, J.-Y. Le Boudec, and M. Vojnovic, "Perfect Simulations for Random Trip Mobility Models," in *IEEE Infocom*, Miami, Florida, USA, Aug. 2005.

[15] S. Pal Chaudhuri, J.-Y. Le Boudec, and M. Vojnovic, "The random trip mobility model," http://icawww1.epfl.ch/RandomTrip/, Apr. 2005.

[16] M. Musolesi and C. Mascolo, "Designing mobility models based on social network theory," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, pp. 59–70, July 2007.

[17] ——, "Community based mobility model," http://www.cl.cam.ac.uk/research/srg/netos/mobilitymodels/, May 2006.

[18] J. Ott, D. Kutscher, and C. Dwertmann, "Integrating DTN and MANET routing," in *ACM MobiCom workshop on Challenged Networks*, Pisa, Italy, Sep. 2006.

[19] 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.

[20] S. Heimlicher and K. Salamatian, "Globs in the Primordial Soup – The Emergence of Connected Crowds in Mobile Wireless Networks," in *ACM Mobihoc*, Chicago, Illinois, USA, Sep. 2010.

[21] A. Chaintreau, A. Mtibaa, L. Massoulie, and C. Diot, "The Diameter of Opportunistic Mobile Networks," in *ACM Conext*, New York, NY, USA, Dec. 2007.

[22] J. Tang, M. Musolesi, C. Mascolo, 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.

[23] J. Whitbeck, M. Dias de Amorim, V. Conan, and J.-L. Guillaume, "Temporal Reachability Graphs," in *ACM Mobicom*, Istanbul, Turkey, Aug. 2012.

[24] 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.