ARTICLE IN PRESS

Ad Hoc Networks xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Neighbour discovery in opportunistic networks

M. Orlinski [a,*], N. Filer [b]

[a] Fraunhofer IAIS, Schloss Birlinghoven, 53757 Sankt Augustin, Germany
[b] School of Computer Science, University of Manchester, Manchester M13 9PL, UK

ARTICLE INFO

Article history:
Received 21 February 2014
Received in revised form 17 July 2014
Accepted 28 July 2014
Available online xxxx

Keywords:
Opportunistic networks
Neighbour discovery
Human mobility

ABSTRACT

Continuous or frequent scanning for opportunistic encounters would quickly drain the battery on existing personal mobile wireless devices. Furthermore, there is a great deal of uncertainty about when encounters between devices carried by humans will take place.

This paper will discuss some of the drawbacks of using current short range neighbour discovery technology in opportunistic networks. Finally, we proposes a new neighbour discovery algorithm called PISTONSv2 which enables mobile devices to dynamically alter the rate at which they search for others, thus creating a fully decentralised and autonomous network whilst saving energy.

1. Introduction

The goal of opportunistic networks such as PeopleNet [1], is to facilitate peer-to-peer communication using the mobility of participants and the occasional connections between them. Opportunistic networks created using personal mobile wireless devices are also sometimes referred to as Pocket Switched Networks (PSNs) because they are created using devices that are often carried in participants' pockets [2].

Rapid and unpredictable topological changes which occur as a consequence of dynamic human mobility patterns and short communication range [3] mean that opportunistic networks may not be suitable for all real-time communication. However, infrastructureless communication between personal mobile devices with fully distributed coordination is expected to facilitate a number of new applications. For example:

1. Continued, albeit limited, communication during unplanned outages of cellular communication networks [4].
2. Basic communication for developing and remote regions where it would otherwise be too expensive or difficult to do otherwise [5].
3. Provide some relief for overloaded cellular networks [6].
4. Sharing of resources between devices [7].

This paper addresses the problem of detecting encounters between mobile devices (called neighbour discovery) in opportunistic networks. Neighbour discovery is critically important because not only is data delivery latency within the network constrained by transmission range, bit rate, and the movement patterns of devices, but also by the rate at which devices scan for others [8].

However, current personal mobile devices cannot be in a constant state of neighbour discovery as this would lower the battery life to the point where they need to be fully recharged at least twice a day [9]. In addition, caching of known neighbours implies an element of exploitable stability within the network which is not the case in opportunistic networks. It is therefore critically important that neighbour discovery procedures in opportunistic networks be as sensitive to change as possible as well as energy efficient.

* Corresponding author.
E-mail address: dr.m.orlinski@ieee.org (M. Orlinski).

http://dx.doi.org/10.1016/j.adhoc.2014.07.024
1570-8705/© 2014 Published by Elsevier B.V.

Please cite this article in press as: M. Orlinski, N. Filer, Neighbour discovery in opportunistic networks, Ad Hoc Netw. (2014), http://dx.doi.org/10.1016/j.adhoc.2014.07.024

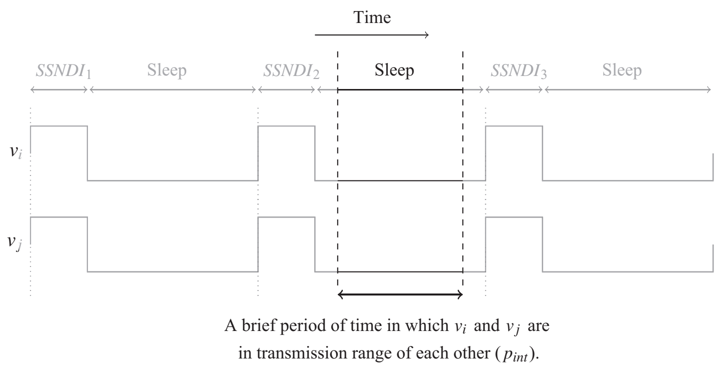


Fig. 1. An inter-probe time which is too long may lead to some short encounters being missed. In this example the devices v_i and v_j briefly come into transmission range of each other during the period p_{int}. There is no Synchronised Symmetric Neighbour Discovery Interval (SSNDI) during p_{int}, thus v_i and v_j do not discover each other.

2. Related work

The different types of signals that are sent between wireless devices during neighbour discovery can be categorised as either Neighbour Discovery Requests (NDREQs) and Neighbour Discovery Replies (NDREPs) [10], or independent neighbour discovery beacons [11,12].

The neighbour discovery procedure of Bluetooth (which is present in most smartphones sold today) uses NDREQs and NDREPs. Bluetooth's neighbour discovery procedure is also *asymmetric* in that Bluetooth devices in transmission range need to be in different but compatible states at the same time in order to detect each other [13].

Asymmetrical states are problematic for opportunistic networks because the assumption that two devices in transmission range are configured as one being the sender and the other the receiver at a particular time is not realistic [14]. Furthermore, Bluetooth devices also need to be in transmission range for long periods of time in order to discover each other and exchange data. Bluetooth 4.0: Basic Rate devices in the Inquiry state can sometimes wait up to 10.24 s (or more in error-prone environments) for a NDREP because of the time needed to perform the 1024 frequency hops outlined in the Bluetooth specification [13]. This means that discovery between stationary devices is often much more reliable than neighbour discovery between mobile devices [15].

In *symmetric* neighbour discovery there is a single state in which devices are required to support transmission, reception, and the processing of NDREQs and NDREPs or beacons. For example, the IEEE 802.15.4 standard outlines the procedure for symmetric state neighbour discovery between Full-Function Devices (FFDs) in ZigBee [16] using periodically broadcast beacons. Symmetric state neighbour discovery is more commonly used in wireless networks where it is difficult to ensure that devices are in complementary neighbour discovery states when in transmission range of one another. For example, when detecting the encounters between Zebras in Kenya [11] or in mountain rescue scenarios [17].

2.1. Autonomous neighbour discovery

So that mobile devices can detect each other in an unsupervised manner and still save energy, new methods are required that keep radios powered off for most of the time but which can still guarantee new encounters between devices will be discovered.

2.1.1. Synchronised symmetric neighbour discovery intervals

One way that personal mobile wireless devices can save energy yet still discover one another autonomously is to initiate symmetric neighbour discovery intervals *simultaneously* on every device. For example, GPS-aided time calibration and a regular operation schedule are used in Impala [11] and CenWits [17] to synchronise symmetric neighbour discovery intervals.

Synchronised symmetric neighbour discovery intervals must overlap whilst mobile devices are in transmission range so they can discover one another. If devices are in transmission range but their synchronised symmetric neighbour discovery intervals do not overlap, then encounters may be missed entirely as Fig. 1 shows, or encounters may be part detected as Fig. 2 shows. During the design of the CenWits system it was calculated that two hill walkers have 102 s to discover the presence of each other if they have a maximum transmission range of 70 m and are moving past each other on the same path. As a result, CenWits devices were configured to start a new symmetric neighbour discovery interval every 90 s.

2.1.2. Asynchronous symmetric neighbour discovery intervals

Section 6.1.2 will show that symmetric neighbour discovery intervals need not be synchronised on every device in order for the opportunistic network to function. This has the benefit of creating an opportunistic network without the need to synchronise time between devices, and allows for different participants to have different encounter patterns.

Two approaches that allow devices to choose the length of time between symmetric neighbour discovery intervals are STAR [18] and DWARF [10]. For brevity, we will now refer to the time between symmetric neighbour discovery intervals as chosen by a device as their *inter-probe time*.

STAR was produced after observing that optimal inter-probe times will vary over time [19]. However, whilst STAR

Please cite this article in press as: M. Orlinski, N. Filer, Neighbour discovery in opportunistic networks, Ad Hoc Netw. (2014), http://dx.doi.org/10.1016/j.adhoc.2014.07.024

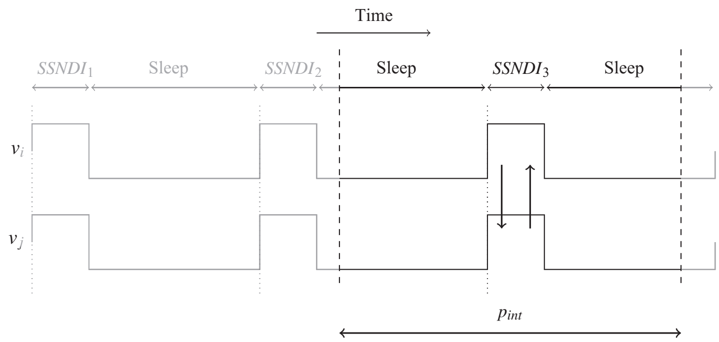


Fig. 2. In this example the devices v_i and v_j are in transmission range of each other during a different period from Fig. 1. This time the two devices can exchange beacons within the third pictured SSNDI.

is a very low energy protocol, it can miss a large fraction of short encounters due to the long inter-probe times it produces. Single encounter duration in opportunistic networks has a heavy tailed distribution [20], with the vast majority of encounters being brief. Therefore, failing to detect the brief encounters can have a drastic negative effect on network connectivity [21]. STAR also requires that devices collect detailed encounter statistics in order to dynamically alter inter-probe times.

Unlike STAR, DWARF requires no knowledge of previous encounters, but DWARF needs to continuously scan for fluctuations in wireless signals in order to initiate symmetric neighbour discovery intervals. However, continuously probing and listening for wireless signals can waste energy when there are no new devices in transmission range.

2.2. Calculating movement speed

In this paper we will assume that mobile devices can estimate their current mode of transportation and/or movement speed. Mobile phones can detect users current mode of transportation by combining readings from accelerometers and gyroscopes [22]. The time and distance between GPS readings can also be used to give an approximation of movement speed. However, in the absence of accurate GPS readings, knowing the users mode of transportation is sometimes enough to set suitable upper bounds for their movement speed (e.g. values up to 1.4 m/s for walking and 30 m/s for driving as are used in Section 4).

3. Inter-probe time calculation

A number of strategies can be adopted in order to choose suitable inter-probe times for opportunistic networks. For efficiency, the inter-probe time giving the smallest number of neighbour discovery intervals per new discovery can be used. It has been shown previously that there can be many efficient inter-probe times when measured in this way [10]. However, it is not clear how inter-probe time choices will affect other measures of connectivity in opportunistic networks, such as the number and length of encounters.

3.1. Troël's method

Troël stated that a mobile device which comes within a given distance should be detected as a neighbour [23]. Troël then proposed Eq. (1) where f_{opt} is the optimal inter-probe time which depends on the relative movement speed of the devices B divided by a threshold distance. The threshold distance is calculated using the value a in the range $0 < a \leqslant 1$ which is multiplied by a maximum uniform transmission radius R.

Troël's equation uses a as a means of specifying how close devices should be to each other in relation to a maximum uniform transmission radius before being classified as neighbours. Inter-probe times are increased with lower values for a as it is assumed that devices which are within the threshold distance spend more time within transmission range than those outside of the threshold distance.

$$f_{opt} = \frac{2B}{aR} \quad (1)$$

One possible issue with Troël's equation is that it assumes the mobility of neighbours tends toward the centre of the transmission radius, which may not always be the case in opportunistic networks due to the many paths along which humans can travel. Troël's equation also produces longer inter-probe times when transmission radius decreases. In opportunistic networks, and other networks where devices can move past each other on the outskirts of transmission range, shorter transmission range means that there may be less time for devices to communicate. Therefore shorter inter-probe times are needed when transmission range is decreased or movement speeds increase.

3.2. A new Inter-Probe time Calculation (IPC)

This paper proposes that we use a different equation for calculating inter-probe times in opportunistic networks based on the intuition that devices carried by participants can quickly pass each other on the outskirts of transmission range. Please note, no formal justification is currently being offered for the method due to the heterogeneity of human movement, the difficulty in defining a shape of

Please cite this article in press as: M. Orlinski, N. Filer, Neighbour discovery in opportunistic networks, Ad Hoc Netw. (2014), http://dx.doi.org/10.1016/j.adhoc.2014.07.024

the transmission area for all mobile devices, and the huge variety of environments for which opportunistic networks are suitable.

The first half of the new Inter-Probe time Calculation (IPC) shown in Eq. (2) needs the movement speed at which the device is travelling measured in metres per second (B), or an approximation based on the mode of transport (see Section 2.2). Eq. (2) then calculates a sampling rate (f_s) using the estimated transmission radius of the device in metres (R).

$$f_s = \frac{2B}{R} \quad (2)$$

$$\tau = \frac{1}{\alpha f_s} \quad (3)$$

The second half of the IPC seen in Eq. (3) produces the inter-probe time referred to as τ. Eq. (3) also introduces the variable α which should be within the range $0 < \alpha \leqslant 1$, and can be used to specify the required level of connectivity and energy saving for the network or individual device.

Together, Eqs. (2) and (3) form the IPC, and unlike Troël's method the IPC has been designed to produce shorter inter-probe times when transmission range decreases. This is more suited to opportunistic networks where mobile devices with short transmission range who spend very little time within transmission range of each other can compensate by sending beacons or NDREQs more often.

Another important point to note here is that the length of symmetric neighbour discovery intervals must be less than or equal to the inter-probe time when using the IPC, and faster movement speeds may mean shorter neighbour discovery intervals. For example, when a movement speed of 30 m/s is used with an α of 1, this leaves a maximum of 0.166 s with which to perform a neighbour discovery interval. Another important issue with the IPC is how to choose suitable values for α, which will now be discussed in more detail.

3.2.1. Choosing α

To show how α changes the inter-probe times produced by the IPC, consider devices moving at a leisurely 1.4 m/s with a transmission range of 10 m. The resulting τ using an α of 1 will be 3.57 s, meaning there will be a new neighbour discovery scan every 3.57 s which could consume a large amount of energy on current mobile devices. However, if detecting every device in range is not a priority then α can be lowered in order to save energy. For example, an α of 0.01 using the same values for movement speed and transmission range will produce a very long inter-probe time of 357.14 s.

Further examples of inter-probe times calculated by the IPC for cases where the maximum speed of devices is set at 1.4 m/s and 30 m/s are provided in Fig. 3. The effects of different α values on other network measurements are presented in Section 5.

4. Experimental environment

The simulated opportunistic networks which are studied in this paper are created using the Working Day Movement (WDM) [24] and Random Walk Movement (RWM) [25] models. Synthetic movement models allow experimentation with higher degree of fidelity and participation than is possible when using existing Reality Mining data-sets; which are often collected using inter-probe times measured in minutes rather than seconds.

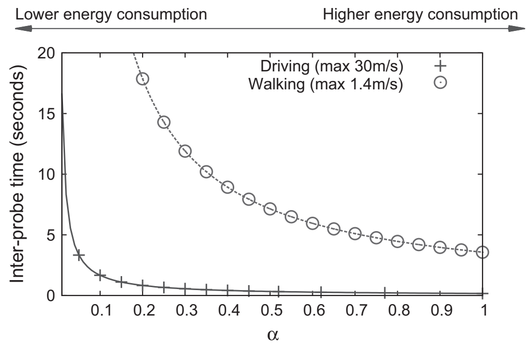


Fig. 3. Inter-probe times produced by the IPC for slow and fast moving devices with a transmission range of 10 m and α values between 0.01 and 1.

The inter-encounter times of the RWM model are distinct from those of the WDM model in that they are light-tailed. Simulated devices moving to the RWM model also move independently and are identically distributed in bounded regions [26]. WDM is used for more realistic pedestrian movement as devices following this model spend long periods being stationary in "home locations" [27] and are not evenly distributed geographically.

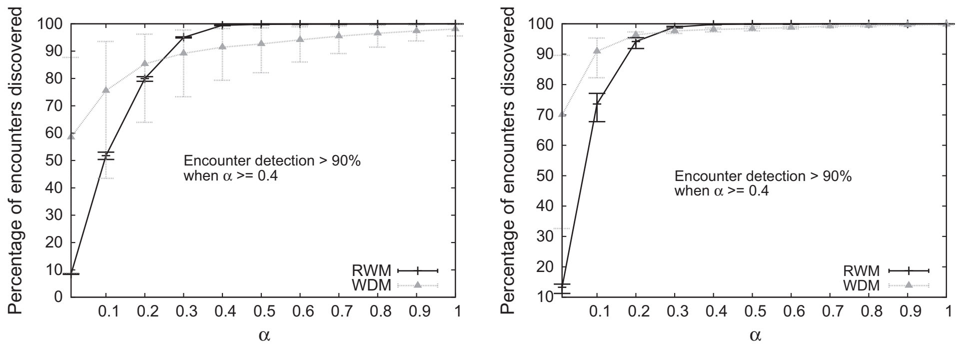
Both the RWM and WDM movement models will be used in conjunction with the simulations settings in Table 1. These settings describe simulated "maps" which limit simulated devices to paths found in real life environments, and the ONE simulator [28] provides a means of altering the inter-probe times of the simulated devices.

The Helsinki map is bundled with the ONE simulator, and University is available as a third party download [21]. Each of these maps tests a different realistic urban environment, and each has been confirmed to work correctly with the RWM and WDM models [24,21].

Each map and movement model combination is repeated using different movement speeds for participants. The speeds range from 0.8–1.4 m/s to mimic typical human walking speeds, or 7–30 m/s to mimic fast traffic speeds. It is not intended to suggest that mobile devices can be driven around all paths in a university, the different movements speeds are simply used to test the effects of the inter-probe time calculation.

Table 1
Simulation settings showing the number of participants used in the different neighbour discovery experiments.

Path layout	University	Helsinki
Number of simulated devices	235	200
Area size (m)	380 × 265	7200 × 6800
Transmission range (m)	10	
Days	2	



(a) Slow movement.

(b) Fast movement.

Fig. 4. Percentage of encounters detected by the IPC using α values between 0.01 and 1, with different movement models and speeds.

It is also important to point out the lost or partly understood neighbour discovery beacons are not considered in this paper. Furthermore, the physical mechanism by which nearby devices are discovered during symmetric neighbour discovery intervals is not discussed further. It is assumed that NDREQs are sent and listened for during symmetric neighbour discovery intervals, and that the length of the interval is sufficient for nearby device to receive, process, and reply to NDREQs.[1]

5. Neighbour discovery results for the IPC

Experiments in this section were conducted using inter-probe times chosen by the IPC proposed in Section 3. The movement speed in the IPC was manually set to the maximum speed at which devices could move in each experiment (1.4 m/s in slow experiments and 30 m/s in fast experiments). Furthermore, symmetric neighbour discovery intervals are synchronised as the devices share the same clock and their IPC result are always the same. Unsynchronised results will be given in the next section.

5.1. Number of encounters detected

Fig. 4 shows the effect α has on the number of encounters captured when using the WDM and RWM models. The figures show the median, minimum, and maximum results taken from the Helsinki and University maps that were described in Section 4. Encounter percentage is measured against the theoretical best case where devices know as soon as another comes into transmission range.

It is stated in [28] that simulated devices moving to the WDM model exhibit more continuous co-location than when using the RWM model, and as a result simulated opportunistic networks created using WDM are more resilient to longer inter-probe times. Yet Fig. 4 shows that this claim is not entirely straightforward. Whilst α values below 0.275 (which give long inter-probe times) result in a higher percentage of encounters detected in the WDM experiments than in the RWM experiments, α values above 0.275 result in a lower percentage of encounters detected in the WDM simulations than in the RWM simulations. This is partly to do with the different number of encounters when using the RWM and WDM models in bounded regions. For example, RWM in the University scenario creates 320,000 possible encounters per day, whereas only 4000 are created when using WDM. Therefore, a missed encounter in WDM experiments is more significant than in the RWM model experiments.

Fig. 4 also shows that the percentage of encounters is largely unaffected when using α values between 0.3 and 1, especially in the fast moving simulations. This information can be used to suggest α values greater than 0.3 as suitable default values for mobile devices using the IPC.

5.2. Encounter duration

The total cumulative encounter duration in the network is also dependent on inter-probe times in both of the movement models tested. Figs. 5 and 6 show that the mean encounter duration increases with decreases to α, whilst total encounter duration in the network decreases when α is decreased. This indicates that fewer short encounters are being detected when inter-probe times are increased.

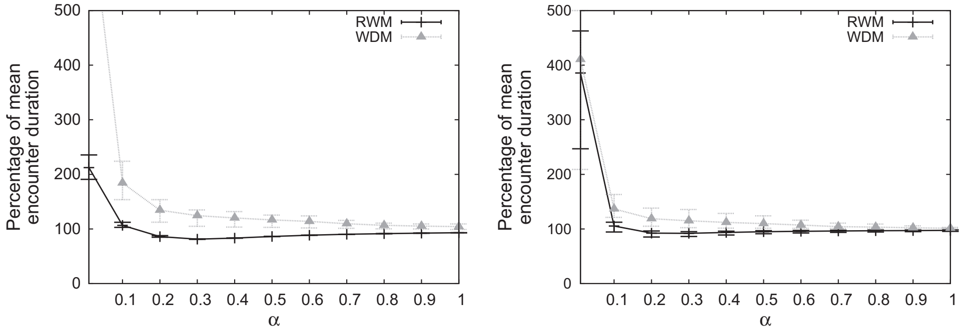
5.3. Energy consumption

Energy consumption for the IPC was measured by calculating how many synchronised symmetric neighbour discovery intervals took place for different values of α. The calculation was performed using 200 devices that run for a total of 2 days and do not exchange any data. A more detailed study including measurements of the energy consumed by NDREQs, NDREPs, and data packets shall be offered later in Section 6.2.

Fig. 7 shows that over 3 times more synchronised symmetric neighbour discovery intervals are performed when α is 1 than when α is 0.3, which could result in about 3 times more energy being consumed. However, the previous sections on the number and cumulative duration of detected encounters showed there was sometimes very little difference in performance when α is greater than 0.3.

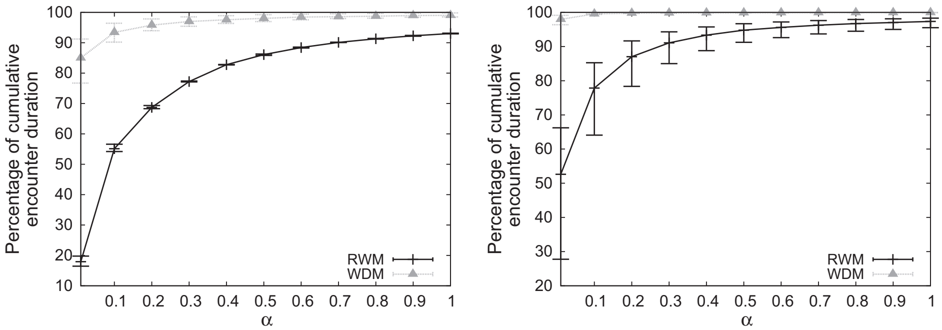
The next section will describe a simple protocol with which devices can alter their inter-probe times in order

[1] Each neighbour discovery interval is 0.01 s long, which is the shortest amount of time possible using our version of The ONE simulator.



(a) Slow movement. (b) Fast movement.

Fig. 5. Mean encounter duration using different α values compared with the best case neighbour discovery.



(a) Slow movement. (b) Fast movement.

Fig. 6. Cumulative encounter duration percentage for the IPC using different α values compared with the best case neighbour discovery.

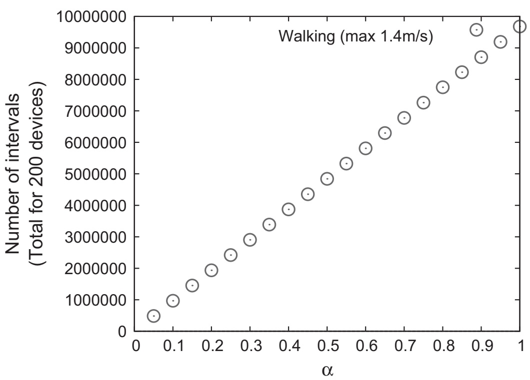


Fig. 7. The total number of synchronised symmetric neighbour discovery intervals for 200 devices over 2 days for α values between 0.01 and 1.

to save energy. Using the IPC, the new protocol has the ability to increase inter-probe times as devices move more slowly or are stationary.

6. PISTONSv2

This section presents a distributed inter-probe time selection protocol for mobile devices in opportunistic networks called PISTONSv2 which is summarised in Algorithm 1. Unlike the IPC in the previous section, this new approach alters the inter-probe times on each device based on their current movement speed. This results in a slightly lower encounter detection rate than was seen in the previous IPC experiments because neighbour discovery intervals are no longer synchronised. However, we now have a more realistic opportunistic network without the need for time synchronisation.

Algorithm 1. PISTONSv2 algorithm.

if CurrentTime >= (LastChecked + InterProbeTime) **then**
  LastChecked = CurrentTime
  Broadcast neighbour discover beacon and listen for responses.
  $f_s = \frac{2B}{R}$
  InterProbeTime $= \frac{\frac{1}{f_s}}{\alpha}$
end if

Like the IPC experiments before, the previous version of this protocol (referred to as PISTONSv1 [21]) required that the maximum likely speed of devices be fixed on each device. Version 1 then altered α based on the results of neighbour discovery intervals. However, with PISTONSv2 we wanted to use the actual movement speed of devices to allow for more realistic heterogeneous environments. Therefore, in this section α will always be set to 1 in order to show the best possible performance of the method. It also is important to note that when the current speed of devices is 0 we set the speed in the IPC to 0.01 m/s in order to prevent errors caused by dividing by zero.

6.1. PISTONSv2 results

This section will present a comparison of the neighbour discovery intervals produced by PISTONSv2 and STAR. We will also show that PISTONSv2 can adapt to changing network conditions by producing longer inter-probe times when movement speed decreases, whilst enabling higher data delivery probabilities than STAR. For these experiments we have used the Helsinki map and increased the number of participants to 400 and the experiment duration to 4 days.

6.1.1. Mean inter-probe times

The inter-probe times produced by PISTONSv2 and STAR in the WDM simulations are presented in Fig. 8. The figures show that the inter-probe times of both protocols follow the behavioural patterns of participants in different ways. The mean inter-probe times of PISTONSv2 tend to be lower than STAR, with STAR adopting the low default inter-probe time (set to 3.7 s for both protocols) on occasion.

The times when STAR is adopting the default inter-probe time are when the arrival rate of new encounters is highest (the 24th hour, 48th hour, etc.). However, on these occasions the mean inter-probe time of PISTONSv2 increases. This is because participants tend to be stationary as soon as they reach their final destination.

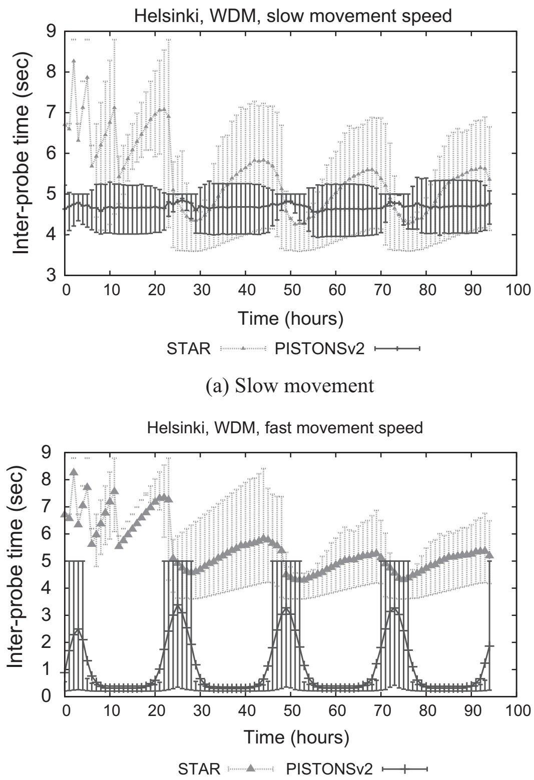
It is also interesting to note that the mean inter-probe time of STAR does not tend to change when movement speed increases in Fig. 8b. This is because the inter-probe times produced by STAR are mostly dependent on the arrival rate of new encounters, and the number of possible encounters are similar in the slow and fast moving scenarios as the next section will show.

6.1.2. Number of missed encounters

Fig. 9a and b shows that the possible number of unique encounters in both the slow and fast moving scenarios are similar. This is because the movement patterns are the same between experiments – only the speed that devices move from point A to B has been altered.

Fig. 9a shows that both STAR and PISTONSv2 miss a large fraction of encounters in the slow moving experiment. This suggests that further improvements to the PISTONSv2 algorithm are necessary in order to detect nearby moving devices when the local device is stationary. We will offer some possible solutions to this problem later in our summary section.

Fig. 9b also shows that the low inter-probe times of PISTONSv2 in the fast moving experiments get closer to detecting the maximum number of unique encounters than STAR. However, the figures again show that PISTONSv2 still misses a large fraction of possible encounters because it increases inter-probe times when participants are stationary.



(a) Slow movement

(b) Fast movement

Fig. 8. Mean, 1st quartile, and 3rd quartile inter-probe times for different movement speeds using STAR and PISTONSv2.

6.1.3. Opportunistic data delivery

This subsection will assess how the different neighbour discovery algorithms effect the data delivery performance of different opportunistic data delivery algorithms. The data delivery simulations were carried out using a pseudo-random data generation pattern that created a new data packet once every 5 min. Every data packet has a time to live of 1 day and a random source and destination device. It is important to remember that the data delivery algorithms can only transmit data after a bi-directional data connection has been established between devices by the neighbour discovery processes.

The opportunistic data delivery algorithms Epidemic [29] and DRAFT [30] were used as one (Epidemic) simply floods the network with duplicate packets at every opportunity, and the other (DRAFT) attempts to make more conservative decisions to cut the number of duplicate data packets at the expense of data delivery probability. The Epidemic protocol needs no additional parameters to function because it simply floods the network. The important variables given to the DRAFT protocol which will allow these results to be reproduced were: familiar threshold of 1000 s, degrade value of 0.7, and frame size of 300 s.

The results in Fig. 10 show that the choice of inter-probe times can have a drastic effect on the data delivery performance of the network. The figures also confirm that

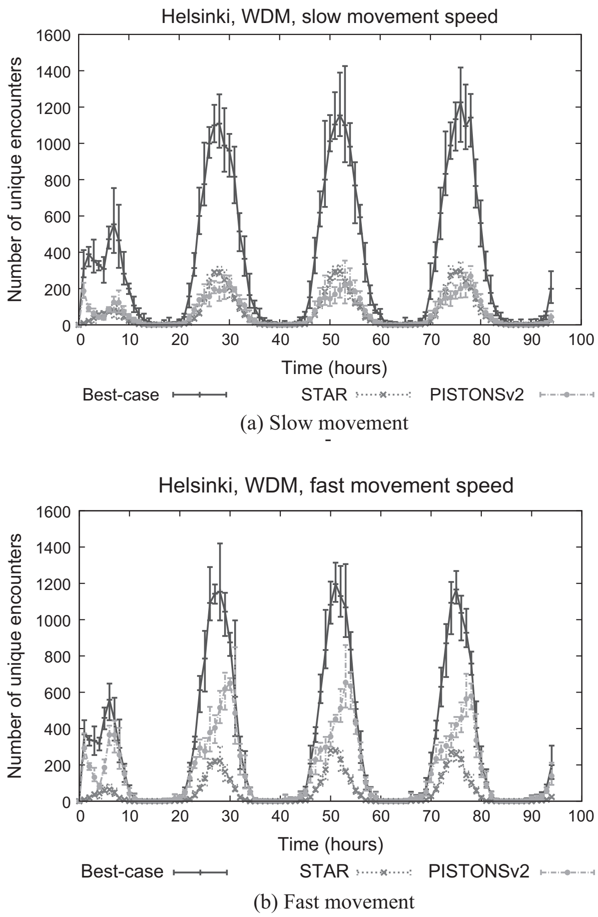


Fig. 9. Mean, min, and max number of unique encounters for different movement speeds using STAR and PISTONSv2.

PISTONSv2 establishes data connections between mobile devices more reliably than STAR. This is most evident in the fast moving scenarios where PISTONSv2 achieves 76% and 55% of the maximum data delivery performance of Epidemic and DRAFT respectively. In contrast, a fast opportunistic network using STAR only achieves 8% and 31% of the network's maximum capability.

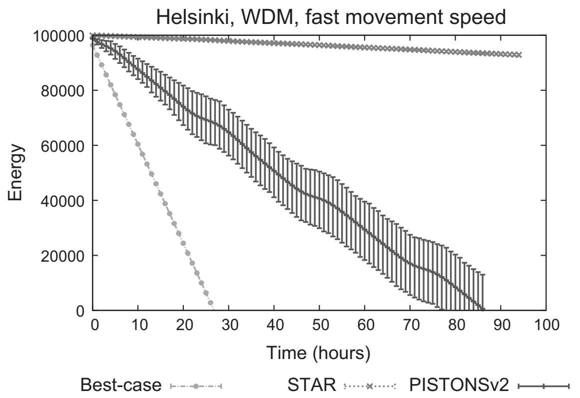


Fig. 11. Remaining energy levels over time for devices running PISTONSv2 and STAR, and using Epidemic data delivery in the Helsinki WDM model experiments.

6.2. Energy consumption

In this section we will see how much energy is used by STAR and PISTONSv2. This is a slightly more detailed study than in Section 5.3 where we only counted intervals. In the following experiment a single unit of energy is depleted when broadcasting a NDREQ, when responding with a NDREP, and when transmitting a data packet. The devices in the best case experiments (first introduced in Section 5.1) broadcasts a neighbour discovery packet and reply to any received neighbour discovery packets every 0.01 s. Every device is given a finite amount of energy and is instructed to send the same data packets as in the previous experiments using the Epidemic protocol.

Fig. 11 shows that STAR barely uses any of the available energy because devices are not sending many NDREQs, NDREPs, or data packets. In contrast, devices running the best case algorithm run out of energy after a single day, but devices using PISTONSv2 can last for 3 days, which is a satisfactory result that balances both data delivery performance and battery life.

There is an obvious trade-off between energy usage and neighbour discovery performance. In the first 3 days PISTONSv2 uses 300 times more energy than STAR, but

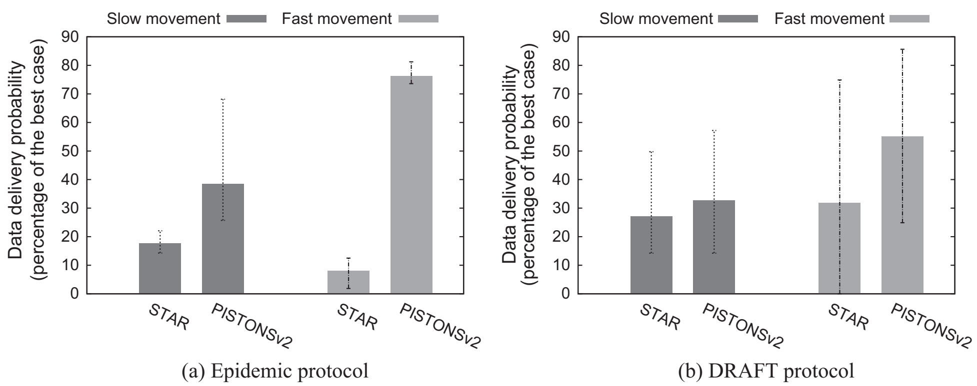


Fig. 10. The mean, minimum, and maximum data delivery performance of PISTONSv2 and STAR in the Helsinki WDM experiments, compared to the best case where devices know as soon as others are in transmission range.

Please cite this article in press as: M. Orlinski, N. Filer, Neighbour discovery in opportunistic networks, Ad Hoc Netw. (2014), http://dx.doi.org/10.1016/j.adhoc.2014.07.024

PISTONSv2 will discover 3 times more encounters and deliver 9.5 times more data. Furthermore, because devices moving to the WDM are stationary for long periods, PISTONSv2 can save energy compared to having all of the devices running the IPC with fixed variables and GPS for time synchronisation.

7. Summary

Autonomous detection of encounters between mobile devices in an opportunistic network is a challenge which has had little attention compared to data routing and mobile social applications. However, this paper has shown that the neighbour discovery process can have as much of an effect on opportunistic data delivery as the routing algorithms used in the network.

In this paper we have also shown that movement speed is an important factor in opportunistic networks. However, knowing the movement speed of one device is not enough to efficiently detect all possible encounters because other nearby devices may be moving at higher speeds.

Section 5 shows that many encounters crucial to performance of future opportunistic networks will be missed if the time between symmetric neighbour discovery scans is longer than the time given by the IPC. It also showed us that a synchronised inter-probe time of 3.57 s in a network made entirely of pedestrians would be sufficient to discover over 99% of encounters due to the limited speed and movement patterns of participants.

However, such a short time between symmetric neighbour discovery intervals may not be needed between bursts of encounters. In Section 6 we proposed a novel neighbour discovery protocol called PISTONSv2 with which mobile devices can increase their own inter-probe time during periods of inactivity whilst still detecting many encounters.

8. Future work

There is still some work left to do to improve PISTONSv2. For example, it may be desirable to decrease inter-probe times for a short period of time after the device becomes stationary in order to detect other moving devices. It may also be possible to adapt our algorithm based on the current requirements of the user, device, and network. For example, if at certain times inter-probe times are very low without subsequent data being exchanged, then it may be desirable to increase inter-probe times during those times to preserve battery life (e.g. by decrease α).

Further improvements to the ratio between encounters detected and the number of symmetric neighbour discovery intervals may be made possible by accurately predicting when bursts of new encounters will occur. For this purpose it may be possible to use existing techniques which attempt to predict the whereabouts of participants using a record of previous locations visited or by exploiting the mobility data of friends [31,32]. However, these techniques often only predict social encounters, when what is necessary are methods which can predict regular encounters with strangers as well. It may also be desirable to detect departures from regular encounter patterns [33].

We believe that a combination of approaches is needed for reliable and efficient neighbour discovery. We cannot simply rely on the movement speed of the local device when calculating inter-probe times, but it should be possible to improve methods such as STAR by taking into account movement speed as well.

Acknowledgements

This work was carried out during the tenure of an ERCIM “Alain Bensoussan” Fellowship Programme. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under Grant agreement 246016.

References

[1] M. Motani, V. Srinivasan, Peoplenet: engineering a wireless virtual social network, in: Proceedings of ACM MobiCom, 2005, pp. 243–257.
[2] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, J. Scott, Pocket switched networks: real-world mobility and its consequences for opportunistic forwarding, Tech. Rep. University of Cambridge, Computer Lab. UCAM-CL-TR-617, 2005.
[3] E.J. Birrane III, S.C. Burleigh, V. Cerf, Defining tolerance: impacts of delay and disruption when managing challenged networks, Tech. Rep., Jet Propulsion Laboratory, National Aeronautics and Space Administration, Pasadena, CA, 2011.
[4] T. Hossmann, F. Legendre, P. Carta, P. Gunningberg, C. Rohner, Twitter in disaster mode: opportunistic communication and distribution of sensor data in emergencies, in: Proceedings of the 3rd Extreme Conference on Communication: The Amazon Expedition, ExtremeCom '11, ACM, 2011.
[5] S. Grasic, E. Davies, A. Lindgren, A. Doria, The evolution of a DTN routing protocol – PRoPHETv2, in: ACM Workshop on Challenged Networks, 2011, pp. 27–30.
[6] B. Han, P. Hui, V.S.A. Kumar, M.V. Marathe, J. Shao, A. Srinivasan, Mobile data offloading through opportunistic communications and social participation, IEEE Trans. Mob. Comput. 11 (5) (2012) 821–834.
[7] F. Vázquez-Gallego, J. Alonso-Zarate, L. Alonso, M. Dohler, Analysis of energy efficient distributed neighbour discovery mechanisms for machine-to-machine networks, Ad Hoc Networks.
[8] N. Nguyen, T. Dinh, S. Tokala, M. Thai, Overlapping communities in dynamic networks: their detection and mobile applications, in: Proceedings of ACM MobiCom, ACM, New York, NY, USA, 2011, pp. 85–96.
[9] G.P. Perrucci, F.H.P. Fitzek, J. Widmer, Survey on energy consumption entities on the smartphone platform, in: 73rd Vehicular Technology Conference, IEEE, 2011, pp. 1–6.
[10] H. Izumikawa, M. Pitkanen, J. Ott, A. Timm-Giel, C. Bormann, Energy-efficient adaptive interface activation for delay/disruption tolerant networks, in: ICACT, vol. 1, 2010, pp. 645–650.
[11] T. Liu, C.M. Sadler, P. Zhang, M. Martonosi, Implementing software on resource-constrained mobile sensors: experiences with Impala and ZebraNet, in: Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services, 2004, pp. 256–269.
[12] A. Kandhalu, K. Lakshmanan, R. Rajkumar, U-connect: a low-latency energy-efficient asynchronous neighbor discovery protocol, in: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, ACM, New York, NY, USA, 2010, pp. 350–361.
[13] B. SIG, Bluetooth specification version 4.0 – master table of contents & compliance requirements, Tech. Rep., 2010.
[14] D. Yang, J. Shin, J. Kim, C. Kim, Asynchronous probing scheme for the optimal energy-efficient neighbor discovery in opportunistic networking, in: IEEE Pervasive Computing and Communications Workshop, Washington, DC, 2009.
[15] A.K. Pietilainen, E. Oliver, J. Lebrun, G. Varghese, J. Crowcroft, C. Diot, Experiments in mobile social networking, Tech. Rep., 2008.
[16] A. Zare, H. Taheri, M. Moghaddam, Using limited flooding in on-demand distance vector junior for reduction power consumption in

ZigBee networks, in: V. Das, Y. Chaba (Eds.), Mobile Communication and Power Engineering, Communications in Computer and Information Science, vol. 296, Springer, Berlin, Heidelberg, 2013, pp. 103–108.
[17] J.H. Huang, S. Amjad, S. Mishra, CenWits: a sensor-based loosely coupled search and rescue system using witnesses, in: Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems, 2005, pp. 180–191.
[18] W. Wang, M. Motani, V. Srinivasan, Opportunistic energy-efficient contact probing in delay-tolerant applications, Trans. Netw. 17 (5) (2009) 1592–1605.
[19] B. Choi, X. Shen, Adaptive asynchronous sleep scheduling protocols for delay tolerant networks, Mob. Comput. 10 (9) (2011) 1283–1296.
[20] Y. Wang, B. Krishnamachari, T. Valente, Findings from an empirical study of fine-grained human social contacts, in: WONS, 2009, pp. 153–160.
[21] M. Orlinski, N. Filer, Movement speed based inter-probe times for neighbour discovery in mobile ad-hoc networks, Ad Hoc Networks, vol. 111, Springer, Paris, France, 2012.
[22] L. Bedogni, M. Di Felice, L. Bononi, By train or by car? Detecting the user's motion type through smartphone sensors data, in: 2012 IFIP Wireless Days, 2012, pp. 1–6.
[23] A. Troël, Prise en compte de la mobilité dans les interactions de proximité entre terminaux á profils hétérogénes, Ph.D. Thesis, Université de Rennes I, 2004.
[24] F. Ekman, A. Keranen, J. Karvo, J. Ott, Working day movement model, in: ACM SIGMOBILE Workshop on Mobility Models, Hong Kong, China, 2008, pp. 33–40.
[25] A. Keranen, T. Karkkainen, J. Ott, Simulating mobility and DTNs with the ONE (invited paper), J. Commun. 5 (2) (2010) 92–105.
[26] G. Sharma, R. Mazumdar, B. Shroff, Delay and capacity trade-offs in mobile ad hoc networks: a global perspective, Trans. Netw. 15 (5) (2007) 981–992.
[27] T. Henderson, D. Kotz, I. Abyzov, The changing usage of a mature campus-wide wireless network, Comput. Netw. 52 (14) (2008) 2690–2712.
[28] A. Keranen, J. Ott, T. Karkkainen, The ONE simulator for DTN protocol evaluation, in: International Conference on Simulation Tools and Techniques, ICST, Rome, Italy, 2009.
[29] X. Zhang, G. Neglia, J. Kurose, D. Towsley, Performance modeling of epidemic routing, Comput. Netw. 51 (10) (2006) 827–839.
[30] M. Orlinski, N. Filer, The rise and fall of spatio-temporal clusters in mobile ad hoc networks, Ad Hoc Netw. 11 (5) (2013) 1641–1654.
[31] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, A. Campbell, Nextplace: a spatio-temporal prediction framework for pervasive systems, Pervasive Comput. (2011) 152–169.
[32] M. De Domenico, A. Lima, M. Musolesi, Interdependence and predictability of human mobility and social interactions, in: Proceedings of the Nokia Mobile Data Challenge Workshop, Newcastle, United Kingdom, 2012.
[33] J. McInerney, S. Stein, A. Rogers, N. Jennings, Breaking the habit: measuring and predicting departures from routine in individual human mobility, Pervasive Mob. Comput. (2013) 808–822.

Mathew Orlinski graduated with a Ph.D. from the University of Manchester in 2013. He also holds a B.Eng. (2007) in Internet Engineering from Aberystwyth University and an M.Sc. (2008) in Computer Security from the University of Birmingham. His research interests include pattern recognition in human movement and encounters, and data dissemination techniques for opportunistic networks.

Nick Filer graduated with a BSc in Computer Science from Queen Mary College London University in 1980. He obtained a PhD from the University of Manchester in 1988 for work applying Artificial Intelligence to CAD systems for electronics design. He has been a Faculty member in Computer Science at Manchester University since 1984 and currently teaches computer networks at under/post graduate levels. He has research and 60+ publications in ECAD, AI, Software Engineering and more recently wireless networking. His current interests are sensor and ad hoc mesh networks, intelligent network stacks, storage and information dissemination in wireless social networks.

Please cite this article in press as: M. Orlinski, N. Filer, Neighbour discovery in opportunistic networks, Ad Hoc Netw. (2014), http://dx.doi.org/10.1016/j.adhoc.2014.07.024