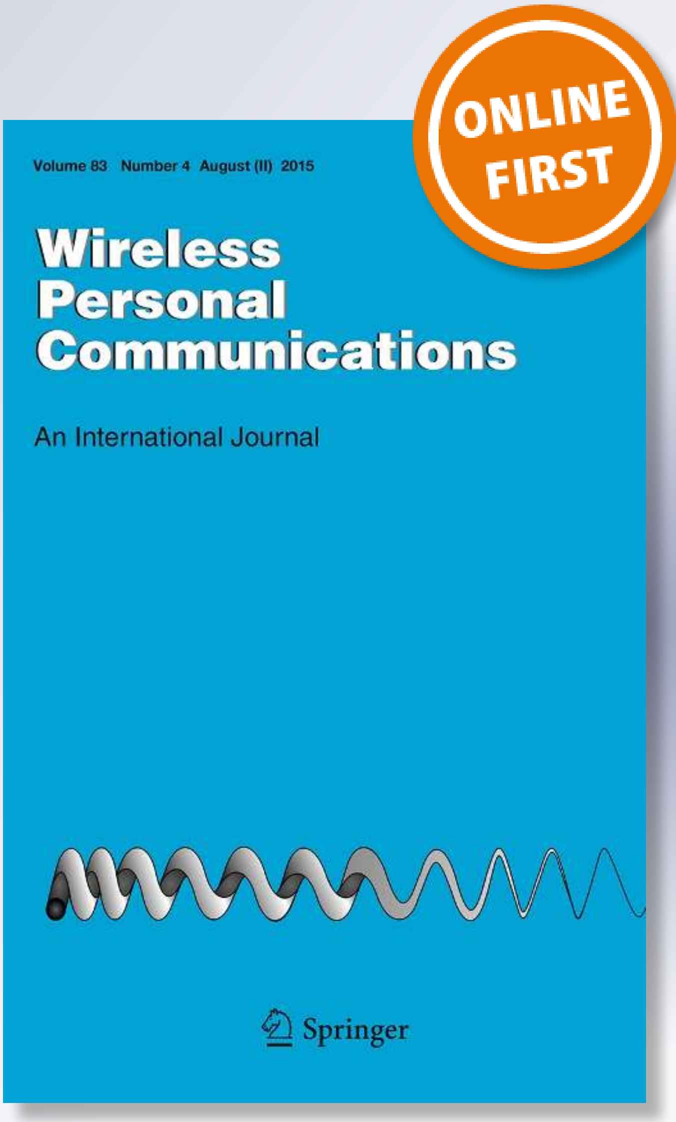
Location and Mobility-Aware Routing for Improving Multimedia Streaming Performance in MANETs

Fraser Cadger, Kevin Curran, Jose Santos & Sandra Moffet

Wireless Personal Communications
An International Journal

ISSN 0929-6212

Wireless Pers Commun
DOI 10.1007/s11277-015-3012-z



Springer

Wireless Pers Commun
DOI 10.1007/s11277-015-3012-z

Location and Mobility-Aware Routing for Improving Multimedia Streaming Performance in MANETs

Fraser Cadger[1] · Kevin Curran[1] · Jose Santos[1] · Sandra Moffet[1]

Abstract Device mobility is an issue that affects both Mobile ad hoc networks (MANETs) and opportunistic networks. While the former employs conventional routing techniques with some element of mobility management, opportunistic networking protocols often use mobility as a means of delivering messages in intermittently connected networks. If nodes are able to determine the future locations of other nodes with reasonable accuracy then they could plan ahead and take into account and even benefit from such mobility. In an ad hoc network, devices form a network amongst themselves and forward packets for each other without infrastructure. Ad hoc networks could be deployed in a disaster scenario to enable communications between responders and base camp to provide telemedicine services. However, most ad hoc routing protocols cannot meet the necessary standards for streaming multimedia because they do not attempt to manage quality of service (QoS). Node mobility adds an additional layer of complexity leading to potentially detrimental effects on QoS. Geographic routing protocols use physical locations to make routing decisions and are typically lightweight, distributed, and require only local network knowledge. They are thus less susceptible to the effects of mobility, but are not impervious. Location-prediction can be used to enhance geographic routing, and counter the negative effects of mobility, but this has received relatively little attention. Location prediction in combination with geographic routing has been explored in previous literature. Most of these location prediction schemes have made simplistic assumptions about mobility. However more advanced location prediction schemes using machine learning techniques have been used for wireless infrastructure networks. These approaches rely on the use of infrastructure and are therefore unsuitable for use in opportunistic networks or MANETs. To solve the problem of accurately predicting future location in non-

✉ Kevin Curran
kj.curran@ulster.ac.uk

Fraser Cadger
cadger-f@email.ulster.ac.uk

1 Intelligent Systems Research Centre, University of Ulster, Londonderry BT48 7JL, Northern Ireland

Published online: 08 August 2015

infrastructure networks, we investigate the prediction of continuous numerical coordinates using artificial neural networks. Simulation using three different mobility models representing human mobility has shown an average prediction error of <1 m in normal circumstances.

Keywords Geographical routing · Manets · Ad hoc networks · Routing · QoS

1 Introduction

If end-user applications are able to make use of a device's location and mobility data then it is logical to consider the possibility of using such information at the network-layer. Geographic routing covers a broad range of protocols that make use of such information varying extents. Geographic routing uses physical location in forwarding decisions. In its most basic form, greedy geographic forwarding, geographic routing forwards packets to neighbours based on their proximity to the destination. In addition to making use of physical locations, greedy routing is also lightweight as nodes do not store routing tables or topology. Instead nodes maintain a list of directly connected neighbours and perform forwarding on per-hop basis, selecting the neighbour closest to the destination and dropping the packet if no neighbour closer to the destination than the node itself can be found. This is done so as to avoid the possibility of routing loops where a packet travels backwards. Geographic (or location-aware) protocols can also utilise location information to optimise specific criteria where location is used to compute the connection time between two nodes, and where mobility serves as an indicator of delay and jitter. Instead of merely using existing information about neighbours' locations, they actively try to determine where their neighbours will be in the future, and thus what effect this will have on routing. Location-prediction is therefore a potentially powerful tool for geographic routing protocols, and can also be of benefit to other areas such as the MAC by reducing transmission power if all neighbours are located nearby and are expected to remain so. Despite this, there has been relatively little attention paid to location-prediction in the area of geographic routing and ad hoc networking in general. Considerably more attention has been paid in infrastructure wireless networks such as WLANs and cellular networks, where Machine Learning has been deployed to predict the future locations of neighbours and thus assist in hand-offs and capacity management. Examples of this include the application of a Hidden Markov Model (HMM) to predict future connectivity based on mobility or the use of Bayesian Networks to monitor mobility with regards to managing hand-offs. These approaches boast high accuracy and success rates. However from an ad hoc networking perspective they are unsuitable as they perform location-prediction in terms of the infrastructure itself, by predicting what Access Point (AP) or cell a node will connect to, and not the actual geographic location of a node. There is great potential therefore, for a geographic routing protocol that is able to accurately predict the future of locations of other devices in a MANET scenario. Such a protocol could also incorporate other context information about the user and environment, and use this to anticipate their future behaviour and how such behaviour would impact the network.

Mobile ad hoc network (MANET) mobility can lead to dynamic behaviour and problems such as sub-optimal routing or link breaks caused by nodes changing their position. Potentially intermittent connectivity poses a significant challenge in performing end-to-end routing. Location prediction for mobility management has been studied in geographic routing protocols such as [8, 7] and [27] and has also been used for quality of service (QoS); [26, 29], and [3]. Geographic routing is localized and does not build end-to-end paths instead performing forwarding on a hop-by-hop basis. It does however still rely on continuous connectivity to an extent; if a node is unable to find a suitable next hop using the forwarding condition then the packet will be dropped. Although this is common in conventional infrastructure as well as ad hoc routing, it is not always desirable. Opportunistic networking is a subfield of Delay Tolerant Networking (DTN). DTNs acknowledge the possibility that connections between network nodes may be intermittent; both in terms of end-to-end connection (which might never be possible using conventional routing) or even nodes being completely disconnected from all other devices for large periods of time. Existing ad hoc routing protocols generally rely on continuous connectivity for building paths, and are unable to cope with prolonged periods of isolation or intermittent connectivity.

Opportunistic networking is an attempt to provide communications in DTNs and other scenarios with intermittent and potentially unpredictable connectivity. Opportunistic networks make forwarding decision on a hop-by-hop basis using local information. Where opportunistic networking differs from geographic routing is its use of the store-and-forward paradigm; if a node does not have any suitable neighbours or is completely disconnected from the network then it will store the message until it finds a suitable next hop. Mobility is therefore a significant factor in most opportunistic networking protocols, being both the cause of and solution to intermittent connectivity. If devices are able to accurately predict where either they or their current neighbours would be at a particular time then they could make forwarding decisions with the aim of increasing reception potential or saving energy. In [14] a geographic routing protocol with some similarities to opportunistic networking allows nodes to favour neighbours who are moving towards the destination over static nodes.

Wireless infrastructure networks such as Wireless Local Area Networks (WLANs) and cellular networks have employed techniques from the field of machine learning such as artificial neural networks (NN) [6] and HMMs [23] for location prediction, which boast high accuracy and success. However, all of these works are reliant on either WLAN or cellular infrastructure as they formulate location prediction as a discrete classification problem in which the aim is to classify a user's location in terms of proximity to a wireless AP or cell and which makes them unsuitable for use in ad hoc or opportunistic networks. An approach which is able to predict locations in a manner which is not dependent on the existence of infrastructure and requires knowledge of the area is therefore highly desirable. This led to the consideration of continuous regression techniques instead of discrete classification for location in prediction in MANETs. In [4] three location prediction algorithms were used to predict future device locations using MANET mobility traces with a NN performing best. The difference between [4] and previous works in infrastructure networks such as [6] and [23] is that [4] is able to take previous coordinates and use these to predict the future coordinates for that device, without the need for any infrastructure or area-specific knowledge which makes it suitable for opportunistic networks.

Although the NN algorithm used in [4] performed well, these tests were performed in Matlab using mobility traces obtained from ns-2 simulations. Instead of receiving coordinates from neighbouring nodes and predicting their future location, the algorithm was simply provided with a complete series of previous locations. Similarly, the experiment did not actually simulate a network itself. Therefore, although the NN algorithm achieved a high level of prediction accuracy (average Mean Square Error (MSE) of 0.102) it was not evaluated under network conditions. This paper presents an implementation of a NN for predicting neighbour locations inside a geographic routing protocol in ns-2. Simulation of mobile networks of varying sizes using three different mobility scenarios of human movement has shown that a NN trained using data from only one of these mobility models is able to predict neighbouring device coordinates often obtaining an average of <1 m error. The differences between the work described in this paper and [4] are; in this paper the NN algorithm is running within a routing protocol, predictions are made based on location updates sent by neighbouring nodes (and are thus susceptible to missing/out of date information), and actual network/mobility behaviour is simulated. While this experiment (and [4]) uses Cartesian coordinates, the general approach could be extended to GPS or any other form of numerical coordinates. Thus the approach described in this paper is suitable for use with either conventional location systems such as GPS or Galileo or localization based on alternative means, and can therefore be used in both indoor and outdoor environments depending on the location system.

2 Previous Work

Ad hoc networks are typically decentralised and do not feature dedicated devices with defined roles such as routers or switches. Instead all participating nodes act as both routers and end-users. As devices are limited by their radio range ad hoc networks typically employ a strategy known as multi-hopping in which a source node will send a message to the destination through a series of intermediate node. This enables geographically disparate nodes to communicate wirelessly. Multi-hopping is typical of the distributed architecture of ad hoc networks, and one of its biggest advantages. As ad hoc networks use multi-hopping and do not rely on infrastructure they can be deployed anywhere two or more devices that share a suitable communications medium (WiFi, Bluetooth, UWB, etc.) are present. Although ad hoc networks have the potential for use in a wide range of application scenarios as diverse as battlefield communications and smart home environments, they also have some drawbacks. In addition to the general challenges of wireless communications such as interference, path loss, and fading that are also present in infrastructure wireless networks, the unique characteristics of ad hoc networks lead to some unique challenges. While a lack of centralisation can be considered an advantage, it can also act as a disadvantage as there is no means of ensuring all devices are operating using the same standards. Ad hoc networks are also more dynamic than infrastructure networks, being formed to fulfil a particular goal and terminated when that goal has been achieved. In addition, most ad hoc networks allow nodes to join and leave the network at will, potentially leading to frequent changes to the topology. Depending on the application, some or even all nodes may be battery powered which presents the possibility of nodes ‘dying’ during operation. Conventional wired and wireless network protocols are therefore not suitable for use in ad hoc networks. This has led to both the adaptation of conventional routing protocols and (more commonly) the design of new ones. Most ad hoc routing protocols can be divided into one of two categories; proactive or reactive. Proactive protocols store and maintain topology information through a series of regular update (hello) messages sent between network nodes. Reactive protocols do not regularly share network information and instead send out route request messages to other nodes when they need to reach a particular destination (although they will typically store routes found during this process for later use). While proactive protocols guarantee that where a network is connected, every node will have a route to a particular destination in advance. They also require the storage and transmission of frequent update messages which can cause problems in the wireless medium. On the other hand, reactive protocols do not require continual sharing of topology information, but cannot always guarantee a route will be available when required and requires the transmission of potentially expensive request messages each time a route cannot be found.

Research into opportunistic networking and MANETs have common roots; both are concerned with networks in which there is limited or no infrastructure and mobility is a factor. Where opportunistic networks and MANETs differ is in their approach to routing. Although MANET protocols attempt to offer solutions to the lack of infrastructure and effects of mobility, they still largely rely on the traditional networking concept of end-to-end communications in which a packet neatly proceeds along a link from source to destination in a timely manner. A few approaches such as the aforementioned geographic routing do not have any concept of a path at all and simply perform forwarding on a hop-by-hop basis. However, such approaches still have the common limitation of needing to make a decision immediately; once a packet is under consideration the protocol will attempt to find a suitable next-hop, and if it is unable to do so they packet will be dropped altogether. While some QoS-intensive applications rely on maintaining a low level of delay, other applications can be considered delay-tolerant and in these instances receiving a packet at *some point* is more important than receiving it within a particular time frame.

These applications therefore require an alternative paradigm in which packets that cannot be forwarded immediately do not necessarily have to be dropped. Opportunistic routing, therefore aims to approach the issue of routing in infrastructureless mobile networks from the perspective of using mobility and intermittent contact to forward packets in instances where delay is permissible (DTNs). The term opportunistic network (or oppnet) was first introduced in 16] and discussed in [17] originally in the context of disaster recovery networks. However, the idea of opportunistic networking can be characterised as being part of the wider area of pervasive/ubiquitous computing and are now being explored outside of the area of disaster recovery. A survey of various approaches to opportunistic networking is provided in [22] who identify three main categories of opportunistic routing protocols; context-oblivious, mobility-based, and social context-based. Context-oblivious protocols are described as being dependant on flooding techniques and includes protocols that utilise both blind (network-wide) and limited flooding (flooding limited based on some criteria) [22]. The latter category includes protocols such as epidemic routing [30] as well as Spray and Wait [28] and network coding schemes [31]. Mobility-based protocols, use mobility patterns and information to make decisions on whether to forward or store a message. Opportunistic protocols that can be classified as mobility-based include PRoPHET [18] which calculates a device's delivery probability based on its previous contact with other nodes and only forwards packets to nodes with a higher delivery probability and CAR [19] which uses Kalman filters to predict context information for message placement. Finally, social context-based protocols attempt to extend mobility-based protocols by taking into account the social context responsible for mobility and contact between devices [22]. Examples of social context-based protocols include HiBOP [2] and Propicman/SpatioTemp [20, 21].

Research on location prediction in wireless networks can be split into two broad categories; research involving MANETs, and research involving wireless infrastructure networks. The majority of location prediction research in MANETs has involved some form of geographic (or location-aware routing). The earliest work on combining geographic routing with some form of location prediction was performed by [8] who devised a connection time metric based on node mobility. A similar metric is that of the motion stability metric proposed in [26] which is defined as the degree of mobility variation by a neighbour, with the authors believing a large degree of motion instability is a factor in high levels of jitter. Unlike [29], the method devised in [26] actually predicts coordinates through the use of a simple prediction scheme based on two previous coordinates. A similar approach to predicting coordinates is used in [29]. However [29] claims that it does not make assumptions about direction (whereas [26] assumes linearity) and works regardless of direction of movement. Despite the success of the various MANET location prediction schemes most of them have been somewhat basic. This was observed in [3] where it was speculated that more advanced location prediction schemes (which in turn would yield more accurate results) would lead to further improvements in routing performance. More accurate location prediction schemes are also desirable as they will allow nodes to have greater awareness of the state (and future state) of both themselves and their neighbouring nodes which is particularly important for opportunistic networks.

In the other area of location prediction, wireless infrastructure networks have sought the use of more advanced location predictions schemes in the form of machine learning algorithms such as artificial NNs [6] and HMMs [23] have. However these approaches are unsuitable for use without infrastructure as they formulate the problem of location prediction in terms of predicting location as proximity to and usage to network infrastructure. For instance, [23] trains a HMM algorithm using traces obtained from real WLANs which show the sequence of APs nodes connect to. Once trained, the algorithm can then predict the next AP a node will connect to based on its previous connections. Both [6] and [23] are examples of classification algorithms. It is however still possible to use classification approaches for non-infrastructure networks so long as a discrete list of locations is available. However such approaches would be inflexible and imprecise.

As classification-based methods do not appear particularly well-suited to opportunistic network location prediction it is perhaps worth considering the possibility of using regression to predict future device locations. The use of regression-based machine learning algorithms to predict neighbour locations in MANETs was initially proposed in [4]. In [4] the authors suggested that instead of performing discrete classification, previous (non-machine learning) methods of location prediction in which continuous coordinates are the output should be modified to use machine learning algorithms. This approach was implemented in the form of mobility traces from six ns-2 scenarios of varying mobility which were used as the training and testing data to evaluate three machine learning algorithms (decision trees, NN and support vector regression) in Matlab. The results indicated that the NN was the overall best performer in terms of accuracy.

This paper follows on from the work of [4] by implementing a NN algorithm inside a geographic routing protocol in ns-2. Unlike [4] in which all testing was done using inputs obtained from previous simulations, the experiments detailed in this paper document a NN algorithm performing predictions from inside a routing protocol during mobile network simulations. Although the work described in this paper still relies on the use of simulation and mobility models, it is important to acknowledge that designing and implementing a machine learning algorithm for location prediction on mobile devices is a time consuming endeavour. Thus, these experiments should serve as a reasonable indication of whether the NN algorithm is able to perform in real mobility scenarios and based on this, whether or not to proceed with a real-world implementation of the NN algorithm.

3 Predicting Neighbour Locations Using Neural Networks

This paper is primarily motivated by two factors; to determine whether NNs can produce results similar to [4] when applied to a geographic routing protocol in a mobile network scenario, and to improve the accuracy of location prediction schemes for use with geographic routing protocols in such scenarios. A potential weakness of the NN algorithm is its 'black box' nature, which means that it is difficult for humans to observe exactly how a NN operates. The NN is fed a series of inputs and then outputs the predicted values without a clear means for the human to determine how the output was obtained. This makes performance analysis somewhat difference, as although the results can be observed it is not obvious how these results were achieved. This makes the design, configuration, and adjustment of NNs somewhat difficult and there is no universal optimum configuration. The design of NN typically varies depending on the task and data, as well as the computational restrictions.

The main adjustable parameters in NN are the number of layers, the number of neurons, and the algorithm used to train the network. All NNs contain at least two layers; the input and output layers. Most NNs also contain at least one hidden layer. The number of hidden layers is a matter of much debate with some believing that more layers increases prediction accuracy, while others arguing that one layer is sufficient for most applications. There is no consensus as to optimum number of layers, and often the decision depends on the processing power available as more layers increases computational cost. Similarly, the number of hidden neurons is also subject to some debate. One belief [25] is that the number of hidden neurons can only be determined through experimentation with different configurations. If the number is too low then there is a risk of predictions being inaccurate and the NN being unable to generalise. If too many however, it can still lead to poor generalisation as a result of overfitting and high variance [25]. Unlike the number of hidden neurons, the number of input layers is typically determined by the number of variables to be used as inputs, while the number of output neurons is determined by the number of desired outputs. In [4] seven inputs were used to represent the current time, two most recent coordinates and timestamp, and the previous coordinates and timestamp, while there were two outputs (the predicted x and y coordinates). The choice of training algorithm on the other hand, is often limited by the availability of algorithms for the particular software used to implement the NN, although hardware resources are still a factor. The NN configuration used in [4] is as follows; one hidden layer, 10 hidden neurons (sigmoid) and the training algorithm was Levenburg-Marquadt.

4 Neural Network Design and Implementation

The NN algorithm was designed and tested using FANN [14] which is an open-source C library for creating multilayer perceptron NNs. FANN was chosen for its lightweight architecture and desirable performance levels, as well as the fact it is written in C and provides a C++ wrapper as ns-2 routing protocols are written in C. Following experimentation with various architectures, the following architecture was chosen; 7 input neurons, 15 hidden neurons, 1 hidden layer, and 2 output neurons the hidden transfer function was sigmoid-symmetric while the output transfer function was linear. The iRPROPR [10] based on the RPROP algorithm [24] was used for training. After testing, the NN algorithm was implemented in a routing protocol in ns-2. The protocol used was the

Greedy Perimeter Stateless Routing algorithm (GPSR) [12] using code from [13]. GPSR is a hybrid geographic routing algorithm which combines greedy and face routing. However, for the purpose of this work the NN implementation was simulated in greedy-only mode. GPSR like most other geographic routing protocols uses the beaconing system for discovering (and keeping alive) links with other devices and disseminating location information. Prior to implementation of the NN algorithm GPSR was modified as in [3] to store the two most recent (instead of just the current) coordinates and their timestamps. Location prediction is then performed when GPSR is determining which of its neighbours is closest to the destination; instead of using the coordinates from the last update, the NN implementation passes the two previous locations and their time to the NN prediction method which inputs these parameters along with the current time into the NN and then receives the predicted x and y coordinates as outputs. Figure 1 shows a high-level pseudocode overview of the original GPSR greedy routing algorithm while Fig. 2 shows the pseudocode for the modified GPSR including the NN implementation.

5 Experiment Configuration

5.1 Prediction Accuracy Setup

Testing was performed by running simulations of the modified GPSR in ns-2 to determine prediction accuracy. A total of 27 simulations were run using three different mobility models and three different numbers of nodes as well as varying the beacon interval. The mobility models used were Random Waypoint Model (RWP) [11], Reference Point Group Mobility (RPGM) [9] and an implementation of the Gauss–Markov model (GM) [15] based on [5]. The reasons for selecting these three mobility models were as follows; RWM and RPGM were the mobility models used in [14] so in order to compare the results from the ns-2 simulation of the NN algorithm with the results from [4] it was necessary to use these two mobility models. RWM is generally considered a simplistic (and possibly inaccurate) representation of mobility as it is purely random and does not take into account either history or surrounding nodes. On the other hand, RPGM is based on group mobility (but where individual mobility is also permissible) and is therefore considered more realistic. In addition GM was chosen because it allows varying degrees of randomness and memory-based decisions depending on the value of the alpha parameter [15]. Both RPGM and GM trace files for ns-2 were generated using the BonnMotion tool [1]. While RWP traces were generated using the setdest tool that comes with ns-2. For RPGM the

```
Select neighbour closest to destination:

1.  Set shortest = distance between us and destination
2.  FOR counter < neighbour table size
        a.  IF current node's distance to destination is less than shortest
                i.  Select current node as next hop
               ii.  Set shortest = current node's distance to destination
        b.  END IF
3.  END FOR
4.  Return next hop or NULL
```

Fig. 1 Original GPSR greedy routing algorithm

Select neighbour closest to destination:

- Set shortest = distance between us and destination
- FOR counter < neighbour table size
    - IF two previous coordinates are available
        - Pass previous coordinates to NN
    - IF coordinates have been predicted
        - Set predicted coordinates = true
    - ENDIF
    - END IF
    - IF prediction coordinates = true
        - IF current node's predicted distance to destination is less than shortest
            - Select current node as next hop
        - END IF
    - END IF
    - ELSE
        - IF current node's distance to destination is less than shortest
            - Select current node as next hop
        - END IF
    - END ELSE
- END FOR
- Return next hop or Null

Fig. 2 Modified greedy routing algorithm using location-predictions

probability of a node joining a group was set to 0.75, while for GM the update frequency was set to 1, the angle standard deviation 0.5, and the speed standard deviation 0.5 (the use of 0.5 for both parameters means that the model is balanced in the middle between memoryless random and memory-based mobility).

Traces for all mobility models were produced for 10, 50, and 100 node scenarios and all scenarios ran for 600 s on an area of size 1500 m × 400 m (with the exception of the GM which ran on 1400 m × 300 m due to a known bug in which traces are produced containing destinations outside of the specified area causing the simulation to crash). Similarly, all scenarios had a maximum velocity of 2.5 m/s and a maximum pause time of 20 s. The maximum velocity of 2.5 m/s is the same as the one used in [4] where it was decided to use this value as it was deemed the most appropriate maximum speed for ordinary human movement. The traffic pattern used for the 50 and 100 node scenarios was as follows; constant bit rate (CBR) User Datagram Protocol (UDP) traffic containing 30 streams with a packet send rate of 0.5. For the 10 node scenario contained 6 streams of CBR traffic with a send rate of 0.5. While it is important to recognize that opportunistic networks might not feature constant streams of traffic, the lack of established opportunistic networking applications makes the modelling of traffic somewhat difficult. Therefore the decision to use CBR traffic was taken as it would allow for the possibility of congestion (at some nodes) and the resulting loss of location updates.

An important difference in the experiments detailed in this paper and those described in [4] is the variation of beacon intervals. Where [4] uses a beacon interval of 0.5 s for all scenarios, our experiments use three different beacon periods; 0.5, 5 and 50 s. The purpose of this is to determine how the NN algorithm is able to react when faced with less frequent updates (particularly in the case of the 50 s where there is a large duration between updates), as this is a potential issue for real-world opportunistic networks and MANETs where beacons may be delayed either unintentionally (as a result of buffering) or intentionally (to reduce traffic in a congested network or save energy). Prediction accuracy was estimated as follows; a comparison of this prediction with the neighbour's actual coordinates was made using the God utility found in ns-2 which allows routing protocols to directly access the simulator and retrieve parameters.

5.2 Reliability, Delay and Delay Variation Experimental Setup

The configurations described in this section refer to the evaluation of GQPR using ns-2. A similar configuration and similar scenarios to the ones described in the Implementation section were used, and they are described in Table 1. Note that the CMUPriQueue queue was used for simulations of DSR due to a bug in DSR. For GQPR a beacon period of 10 s a congestion control alpha value of 0.001 was used. While ten seconds may seem like a high value this is based on GQPR's ability to predict future locations and thus reduce the number of beacons required, while the 0.001 alpha value was arrived at after experimenting with other values. Simulations of 10, 30, and 50 nodes using the RWM, RPGM, and GM models were performed giving a total of 9 unique scenarios. All simulations use a maximum velocity of 2.5 m/s and a maximum pause time of 20 s. The RPGM scenario used a join probability of 0.75 meaning nodes have a 75 % chance of joining a group, while the GM scenarios used an update frequency of 1, angle standard deviation 0.5, and a speed standard deviation of 0.5 to allow a suitable mix of random and non-random mobility. These mobility models were chosen because in addition to being used for earlier experiments they reflected different and diverse aspects of mobility modelling. RWM is purely random, whereas RPGM incorporates both individual and group mobility, while GM exhibits varying degrees of random and non-random mobility.

For traffic the following configurations were used. For the 10 node scenario 1 video call and 1 video stream, for the 30 node scenario 2 video calls and 4 video streams and for the 50 node scenario 3 video calls and 4 video streams. Each video call consisted of two nodes sending CBR packets of size 512 bytes and with a send rate of 58 packets per second. Video streams also use 512 byte packets but have only one node sending and use a higher send rate of 128 packets per second and is intended to reflect the streaming of 360–480 p traffic. These scenarios are intended to realistically model video calling/VoIP and on-demand video streaming. It was decided to use traffic characteristics based on these applications instead of the applications themselves, as simulating real VoIP and video streaming traffic in large topologies would take a great deal of time. To evaluate the performance of GQPR the three standard QoS metrics of reliability, delay and delay

Table 1 ns-2 configuration parameters

Parameter	Value
Duration	500 s
Grid size	1500 m × 300 m
Channel	Channel/wireless channel
Propagation model	Propagation/tworayground
Network interface	Phy/wirelessphy
MAC	Mac/802_11
Queue	Queue/droptail/priqueue
Queue length	512
Antenna	Antenna/omniantenna
Data rate	10 Mb

variation were used. Reliability is the rate of data packets successfully delivered, while delay is the duration between a packet being created and received, and delay variation is the standard variation of packet delays at a node. While there are no fixed QoS parameters and the amount of visible disruption a user will be willing to tolerate varies on the individual, there are some good practices with regards to QoS. For instance, Cisco recommends delay not exceed 150 ms and delay variation no more than 30 ms with no recommendations for packet loss. It is important to recognise that these metrics are intended for streaming over infrastructure networks or the Internet, which will have greater resources available than that of an emergency MANET. However, if the video being streamed by our framework is not of adequate quality from the user's perspective, it will be of little use. Therefore, the users may have reduced perceptions of the quality available minimum standards must be adhered to. Thus delay should typically be below 300 ms and packet loss below 10 % and preferably 3 %. Note that due to DSDV continually freezing on the 50 node RWM scenario there are no statistics for its performance here.

6 Results

6.1 Prediction Accuracy

Tables 2, 3, 4 show the results of the simulations for each of the three mobility models. Note, that here error is not MSE and is the actual difference in meters between the predicted and actual coordinates. When evaluating these results, it is important to note that there is no obvious 'state of the art' to compare them with. As [4] was the first approach to propose the use of machine learning algorithms for predicting continuous coordinates, while there is no data on the accuracy of geographic routing location prediction algorithms. Therefore, establishing a threshold for prediction accuracy is somewhat difficult.

Results show that the NN algorithm is able to predict future coordinates very accurately and with minimal error except for the 50 s beaconing scenarios. Although there is no commonly accepted threshold for location prediction error, many systems in the area of localization strive for an error of <2–3 m therefore anything below this should be considered highly desirable and anything below 1 m exceptional. In a total of 18 scenarios out of 27 an absolute error of <1 m was observed. Furthermore an error of >3 m was only observed in 5 scenarios and all of these instances occurred when beacon duration was set as 50 s. For the 0.5 and 5 s scenarios only one of these 18 scenarios contained an error >1 m (1.08 m recorded in RPGM 50 nodes, 5 s) and none >2 m. This indicates a high degree of accuracy in predicting future locations when provided with frequent (0.5 or 5 s) location updates, although performance does appear to decrease when faced with infrequent (50 s updates).

With regards to the 50 s beaconing, it is not surprising that the results found in these scenarios contain the highest level of error (11.82 m is the highest error of any scenario) given the large duration between updates. At a maximum velocity of 2.5 m/s it is possible that a

Table 2 Absolute error for all RWM scenarios

Number of nodes	Error (0.5 s)	Error (5 s)	Error (50 s)
10	0.123	0.53	7.55
50	0.1025	0.484	4.24
100	0.148	0.835	2.45

Table 3 Absolute error for all RPGM scenarios

Number of nodes	Error (0.5 s)	Error (5 s)	Error (50 s)
10	0.155	0.475	1.58
50	0.206	1.08	11.82
100	0.1385	0.727	0.145

Table 4 Absolute error for all GM scenarios

Number of nodes	Error (0.5 s)	Error (5 s)	Error (50 s)
10	0.182	0.775	3.345
50	0.135	0.397	1.59
100	0.188	0.894	3.97

person can move up to 125 m from their previously recorded location. Similarly, when the two previous updates are received 50 s apart it is also more difficult to develop a pattern in these updates than if they are received more regularly as it is possible that a person's mobility will have changed significantly between previous updates. However, the 50 s scenarios are included as exceptional scenarios to demonstrate how the NN algorithm performs when faced with conditions not well-suited to location prediction. For the 50 s scenarios only 4 out of 9 scenarios contained an error >2 m, however large errors such as 11.82 and 7.55 indicate that the algorithm can struggle when faced with irregular updates. It is also possible that the poorer results are also due to the training set used and that a training set based on less frequent updates would perform better in such a scenario. In general though, the NN algorithm performs reasonably well when faced with infrequent updates.

Performance with regards to the different mobility models appears to be fairly balanced. Although the NN algorithm was trained using only the 50 node RPGM dataset from [4] which itself used beacon periods of 0.5 s, it is still able to accurately predict (often with <1 m of error) future device locations in scenarios which use different mobility models, number of nodes, and beacon periods. For instance, the lowest prediction error (0.1025 m) is found in the 50 node RWM scenario with a beacon interval of 0.5 s, while the highest prediction error (11.82 m) is from the 100 node RPGM scenario with a beacon period of 50 s. Considering number of nodes there again appears to be no obvious 'perfect' scenario with the highest and second highest level of error occurring at 50 node and 10 node scenarios respectively.

6.2 Reliability, Delay and Delay Variation Results

6.2.1 Reliability

Table 5 contains the reliability results for the RWM simulations. In the 10 node scenario, only AODV is able to attain a standard close to the requirements for streaming QoS.

Table 5 Reliability for RWM scenarios

Protocol	10 nodes	30 nodes	50 nodes
GQPR	77.6	83	73
AODV	90.3	75.2	71.2
DSR	43.8	80.3	–
DSDV	48.8	73.5	67

Although GQPR comes second, 77.6 % packet delivery would generally be considered unsuitable. Both DSR and DSDV perform extremely poorly in this scenario. All protocols except AODV show a marked improvement in the 30 node scenario, and GQPR comes close to reaching a level suitable for multimedia streaming, but falls short by 7 p.p. GQPR again outperforms AODV (and DSDV) in the 50 node scenario, but again the result obtained here is unsuitable for streaming QoS.

Regarding the overall performance, it is interesting to note that all protocols (except AODV) experience an increase in packet delivery between 10 and 30 nodes, but then a decrease at 50 nodes. The mobility created by the RWM is most likely a factor in the poor performances seen here. As the RWM is purely random, it is to be expected that routing will be disrupted by the constant and unpredictable mobility. Although the NN location-prediction algorithm used by GQPR was often able to accurately predict future locations in RWM that does not necessarily mean that it will always be able to utilise this information to improve routing. Mechanisms such as motion stability are also of little use if all motion (including that of the sending node itself) is purely random, as a neighbour that may have previously been relatively stable could suddenly make a 'random' and unforeseen change. While the RWM is not intended as an accurate model of human (or any other kind of) mobility, the results are still useful as they enable GQPR to be observed in differing contexts.

6.2.2 Delay

The results for delay in all RWM scenarios are presented in Table 6. In the 10 node scenario GQPR, DSDV and DSR all perform well while AODV incurs an unacceptable 2 s of delay. These results should however be considered in the context of the reliability results, and as both DSR and DSDV had <50 % packet delivery it is hardly surprising that they experienced low levels of delay. GQPR's performance can be seen as a positive, but with reliability only 77.6 % it comes at a price. GQPR again achieves the lowest level of delay for the 30 node scenario, with all other protocols exceed the informal limit of 300 ms. The performance by GQPR is particularly notable in comparison with AODV as GQPR achieves a slightly higher level of reliability, and a significantly higher lower level of delay. This suggests that GQPR is able to handle trade-offs between reliability and delay well when conditions are favourable. However the strong performance by GQPR in the 30 node scenario, must be considered alongside the 10 and 50 node scenarios where GQPR achieves very low levels of delay, but relatively poor reliability. This may be as a result of GQPR prioritising reduced delay over packet delivery and making routing decisions that lead to routable packets being dropped. When evaluating the results for the 50 node it is necessary to take into account that all other protocols performed poorly in this scenario as well, and that GQPR was the best performer in terms of both reliability and delay.

Table 6 Delay for RWM scenarios

Protocol	10 nodes (ms)	30 nodes (ms)	50 nodes (ms)
GQPR	4	5.9	8.6
AODV	2060	2200	1798
DSR	12.4	1910	–
DSDV	9.6	310	556

6.2.3 Delay Variation

Delay variation results are presented in Table 7. Comparing the results of delay variation with delay shows that while DSR and DSDV achieve good levels of delay, they experience a high level of delay variation; GQPR has only a minor variation and is the only protocol within the 10–50 ms window of acceptable jitter. In contrast, given that DSR and DSDV both had extremely low packet delivery levels, the large levels of delay variation experienced are likely a consequence of this. GQPR experiences a slight increase in the 50 node scenario, but still outperforms the other protocols. Although GQPR does not predict delay variation and does not explicitly try to manage it, GQPR achieves acceptable levels of jitter when the other protocols fail to do so. This is particularly interesting given the randomness of the RWM scenario is likely to create a continuously changing environment, that could be a potential source of a high jitter. This may be the reason that the other protocols struggle in this area, as a low level of delay is not a guarantee of low jitter. This would be a logical explanation for DSR and DSDV experiencing low delay but high delay variation in the 10 node scenario, given that the low delay was likely a result of frequent packet drops.

6.3 Comparison with Other Results

As previously outlined the only paper to discuss the use of NNs for predicting coordinates is [4]. Tables 8 and 9 show a comparison between the errors in MSE between the results obtained from these experiments and the previous results obtained in [4]. It is important to note that only the results from the 0.5 beacon interval are considered as [4] only experimented with a beacon interval of 0.5 s and that the results from the GM simulations are excluded as [4] did not use the GM model.

The purpose of this paper is to ascertain whether the approach (but not necessarily the particular implementation) proposed in [4] is viable for use in an actual opportunistic networking routing protocol. Before performing these experiments it was expected that the NN algorithm deployed inside the GPSR routing protocol would not achieve as accurate a performance as the Matlab analysis in [4]. This was because the Matlab analysis was somewhat idealized and was not being performed 'live' where changes were actually occurring. Another aspect in favour of [4] was that there was no chance of missing or delayed update packets which is always a possibility in simulated and real deployments. Furthermore, the Matlab analysis used training and testing data derived from the same overall dataset. Even more importantly, testing was performed using a training set from the same scenario (for instance, training and testing for the 10 node RWM scenario was performed using training and testing sets from 10 node RWM scenarios) whereas in our simulations only the 50 node RPGM training set was used for all scenarios even when the number of nodes and/or mobility model differed. Therefore it initially appears surprising that the results from our simulations are in all scenarios better than the results from the

Table 7 Delay variation for RWM scenarios

Protocol	10 nodes (ms)	30 nodes (ms)	50 nodes (ms)
GQPR	2.9	20	22
AODV	2269	2940	2711
DSR	110	4144	–
DSDV	580	934	914

Table 8 Comparison of MSE between results from [4] and our simulations for RWM scenarios

Number of nodes	MSE from [4]	MSE from ns-2 simulation
10	73	0.054
50	3.31	0.0475
100	0.172	0.076

Table 9 Comparison of MSE between results from [4] and our simulations for RPGM scenarios

Number of nodes	MSE from [4]	MSE from ns-2 simulation
10	0.357	0.143
50	0.539	0.1463
100	0.5049	0.145

Matlab analysis of [4]. However, it is important to remember that in [4] the configuration for the NN is based on the default configuration in Matlab. On the other hand, the NN algorithm discussed in this paper was developed by training and testing different architectures and configurations using the FANN application with the best performing configuration being chosen. Although this seems like the logical explanation for this NN evaluation outperforming [4], it is important to remember the purpose of these experiments was not to devise a NN implementation that outperformed [4] but instead to prove that such an NN implementation was viable for use inside an opportunistic networking protocol. The fact that the specific configuration used in this paper outperformed [4] is further proof that the general concept of using a NN algorithm for predicting continuous coordinates is a viable and desirable approach for performing location prediction in opportunistic networks or MANETs. Overall, all these results indicate that a 2-layer NN algorithm with 15 hidden neurons is able to accurately predict future device locations in various scenarios of human mobility where number of nodes, mobility model, and beacon interval are all varied. That the same NN architecture trained on a dataset from one of these scenarios (50 nodes RPGM with 0.5 s beacon interval) was able to perform such accurate predictions in a wide variety of different scenarios confirms that the NN prediction approach proposed in [4] is suitable for use in mobile network location prediction.

In addition, GQPR's performance with regards reliability, delay and delay variation can be considered on the whole as positive. Although GQPR did not always meet the 90 % packet delivery criteria, the only mobility model where it completely failed to achieve this was RWM. In RPGM GQPR was able to attain at least 90 % delivery in all scenarios, and only failed to do so in one GM scenario (where it achieved 88 % packet delivery). In contrast, AODV achieved >90 % reliability in one scenario of RWM, but failed to do so for one scenario in RPGM and GM. Given the random nature of RWM it is not surprising that GQPR performed poorly, as did all of the other protocols except AODV in the 10 node scenario. Real-life human mobility is seldom purely random, and while the RWM should not be discounted as a mobility model, it is also not representative of the way humans are liable to move in a disaster-recovery scenario. Even in a dynamic environment potentially containing various obstacles and hazards, humans are still likely to move in an organised fashion. Thus the RPGM and GM models should be seen as more representative of human mobility. While the GM contains some elements of random behaviour, it also includes memory-based mobility, therefore allow it to model for the possibility of random behaviour that can exist in human mobility. Except for the 30 node scenario, GQPR performs well in the GM simulations. As the RPGM is based on group (as well as individual) mobility, the results provided by it are interesting as GQPR not only consistently achieves its best packet delivery rates, but also comes close to 100 % delivery in the 10 and 50 node scenarios. The results from the RPGM scenarios are particularly positive when considered alongside the delay results, with GQPR achieving its lowest level of delay in the 10 node scenario, and never rising above 5 ms of delay. While high packet reception may be expected to lead to higher levels of delay, GQPR is able to achieve delivery rates close to 100 % and very small levels of delay. The mobility model may be a factor here, with GQPR being able to better predict neighbour locations and use this information for QoS predictions, and general geographic routing. That GQPR is able to achieve this balance also suggests that the trade-off it makes between the competing demands of reliability and delay are made successfully so as to allow the right balance that does not sacrifice low delay for high reception, or vice versa. As the other protocols all obtain high levels of delay and do not perform as well as GQPR in reliability for the RPGM scenarios, this further strengthens the case for GQPR's routing logic.

7 Conclusion

Ad hoc networks provide the possibility of forming a network consisting only of the end-user devices with no infrastructure. Such a network can be created spontaneously and managed in a distributed manner. Almost any device equipped with a WiFi radio can take part in an ad hoc network if it has the correct software. Ad hoc networks have however largely been confined to novel research problems and most real-world deployments are of a military nature. In order to run a successful telemedicine service stringent QoS demands must be met by the network so as to achieve a suitable level of video/audio quality. Existing ad hoc routing protocols typically prioritise packet delivery over QoS management and may therefore be unsuitable for handling QoS-sensitive traffic. The work of this thesis has explored the possibility of designing and developing a framework that is able to provide streaming multimedia over ad hoc networks. GQP2PS aims to leverage location and mobility information, along with other context information to make QoS predictions that will allow for a suitable streaming quality to be achieved.

The main contributions of this paper are to demonstrate the suitability of an artificial NN implemented inside an ad hoc routing protocol for predicting human mobility in opportunistic network scenarios in addition to further demonstrating the effects of implementing a mobility model and its effects on reliability, delay and delay variation.. This work builds on that of [4] where three machine learning algorithms were evaluated for their ability to accurately predict future device locations based on mobility traces obtained from mobile network simulations. In [4] the NN was overall the best performer in terms of prediction accuracy, and this work therefore focuses on further evaluating its potential for use in mobile network location prediction. This paper illustrates the process of implementing a two-layer feedforward NN algorithm created with the FANN library inside the GPSR geographic routing protocol in ns-2. Results showing the accuracy of the NN algorithm's predictions in actual simulations are then presented. The average prediction error from all scenarios is only 1.88 m. However it is important to recognize that this figure contains the errors from the 50 s scenario which are significantly higher than the other scenarios' errors, and if the results from the 50 s scenarios are excluded the average is <1 m of error. Similarly, a comparison of the results from the RPGM and RWM simulations with the corresponding results from [4] showed that the NN algorithm simulated in ns-2 actually performed better than the NN algorithm analyzed in Matlab.

Even when the duration, number of nodes, and mobility model was changed the NN algorithm was still able to predict neighbours' locations with a very high degree of accuracy. The paper also discussed possible future applications of the results most of which were related to developing an opportunistic networking routing protocol based on NN location prediction. It is the intention of the authors to pursue the development of such a protocol and also to implement the NN prediction algorithm on Android smartphones in order to assess its performance on realistic mobile hardware. Overall, these results are very promising in particular the consistently high prediction accuracy of the NN algorithm in a variety of differing mobile scenarios. These results are applicable to a wide variety of areas where mobility and location are important factors.

We also presented a simulated evaluation of GQPR and an analysis of its result. GQPR alongside three other ad hoc routing protocols (AODV, DSR, and DSDV) was simulated using three mobility models (RWM, RPGM, GM) and three network configurations (10, 30, and 50 nodes). Traffic based on characteristics typical of video streaming and video calling was simulated and the protocols were evaluated in terms of reliability, delay, and delay variation. GQPR was the best performer overall, often obtaining a packet delivery rate over 90 % and sometimes close to 100 % (except in the RWM scenarios), and continually achieved the lowest levels of delay and delay variation.

References

1. Aschenbruck, N., Ernst, R., Gerhards-Padilla, E., & Schwamborn, M. (2010). BonnMotion: A mobility scenario generation and analysis tool. In *Proceedings of the 3rd international ICST conference on simulation tools and techniques*, pp. 51:1—51:10.
2. Boldrini, C., Conti, M., Jacopini, J., & Passarella, A. (2007). HiBOp: A history based routing protocol for opportunistic networks. In *World of wireless, mobile and multimedia networks, 2007. WoWMoM 2007. IEEE international symposium on a*, pp. 1–12.
3. Cadger, F., Curran, K., Santos, J., & Moffett, S. (2011). An Analysis of the Effects of Intelligent Location Prediction Algorithms on Greedy Geographic Routing in Mobile Ad-Hoc Networks. In *Proceedings of the 22nd Irish Conference on artificial intelligence and cognitive science*, Derry.
4. Cadger, F., Curran, K., Santos, J., & Moffett, S. (2012). MANET location prediction using machine learning algorithms. In Y. Koucheryavy, L. Mamatas, I. Matta, & V. Tsaoussidis (Eds.), *Wired/wireless internet communication* (pp. 174–185). Berlin Heidelberg: Springer.
5. Camp, T., Boleng, J., & Davies, V. (2002). A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing, 2*(5), 483–502.
6. Capka, J. & Boutaba, R. (2004). Mobility prediction in wireless networks using neural networks. *Management of multimedia networks and services*, pp. 320–333.
7. Chou, C., Ssu, K., & Jiau, H. (2008). Dynamic route maintenance for geographic forwarding in mobile ad hoc networks. *Computer Networks, 52*(2), 418–431.
8. Chen, Q., Kanhere, S., Hassan, M., & Lan, K.-C. (2006). Adaptive position update in geographic routing. In *2006 IEEE international conference on communications, vol. 0, no. c*, pp. 4046–4051.
9. Hong, X., Gerla, M., Pei, G. & Chiang, C.-C. (1999). A group mobility model for ad hoc wireless networks. In *Proceedings of the 2nd ACM international workshop on modeling, analysis and simulation of wireless and mobile systems*, pp. 53–60.
10. Igel, C., & Husken, M. (2000). Improving the RPROP learning algorithm. In *Proceedings of the second international symposium on neural computation (NCC 2000).*

11. Johnson, D. B., & Maltz, D. A. (1996). Dynamic source routing in ad hoc wireless networks. In *Mobile computing*, pp. 153–181.
12. Karp, B., & Kung, H. (2000). GPSR: greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 243–254.
13. Kiess, W. (2012). *HLS patch for ns-2.29 and ns-2.33* [Online]. http://www.cn.uni-duesseldorf.de/alumni/kiess/software/hls-ns2-patch
14. Li, J., & Shatz, S. M.(2008). Toward using node mobility to enhance Greedy-forwarding in geographic routing for mobile ad hoc networks. In *The international workshop on mobile device and urban sensing (MODUS 2008)*, (pp. 1–8), St. Louis, MO.
15. Liang, B., & Haas, Z. J. (1999). Predictive distance-based mobility management for PCS networks. In *INFOCOM '99. Eighteenth Annual joint conference of the IEEE computer and communications societies. Proceedings. IEEE*, vol. 3, pp. 1377–1384.
16. Lilien, L., Gupta, A., Member, S., & Yang, Z. (2006). Opportunistic networks for emergency preparedness and response. In *Performance, Computing, and Communications Conference Proceedings, IPCCC 2007*, (p. 588–593) New York, USA
17. Lilien, L., Kamal, Z. H., Bhuse, V., & Gupta, A. (2006). Opportunistic networks: the concept and research challenges in privacy and security. In *International workshop on research challenges in security and privacy for mobile and wireless networks (WSPWN 2006)*, (p. 134–147) Miami, Florida.
18. Lindgren, A., Doria, A., & Schelén, O. (2003). Probabilistic routing in intermittently connected networks. *SIGMOBILE Mobile Computing and Communications Review, 7*, 19–20.
19. Musolesi, M., & Mascolo, C. (2006). A community based mobility model for ad hoc network research. In *Proceedings of the 2nd international workshop on multi-hop ad hoc networks: From theory to reality* (pp. 31–38) ACM, New York, NY, USA.
20. Nguyen, H. A., Giordano, S., & Puiatti, A. (2007). Probabilistic routing protocol for intermittently connected mobile ad hoc networks (PROPICMAN), *IEEE AOC* 2007, Helsinki.
21. Nguyen, H. A., & Giordano, S. (2008). Spatiotemporal routing algorithm in opportunistic networks. In *World of wireless, mobile and multimedia networks, 2008. WoWMoM 2008. 2008 international symposium on a*, pp. 1–6.
22. Nguyen, H. A., & Giordano, S. (2009). Routing in opportunistic networks. *International Journal of Ambient Computing and Intelligence (IJACI), 1*, 19–38.
23. Prasad, P. S., & Agrawal, P. (2010). Movement prediction in wireless networks using mobility traces. In *Consumer communications and networking conference (CCNC), 7th IEEE*, pp. 1–5.
24. Riedmiller, M., & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Neural Networks, 1993., IEEE international conference on*, vol. 1, pp. 586–591.
25. Sarle, W.S., ed. (1997), *Neural network FAQ, part 1 of 7: Introduction*, periodic posting to the Usenet newsgroup comp.ai.neural-nets, URL: ftp://ftp.sas.com/pub/neural/FAQ.html. Last Accessed 10 February 2013.
26. Shah, & Nahrstedt, K. (2002). Predictive location-based QoS routing in mobile ad hoc networks. In *Proceedings of IEEE international conference on communications. ICC*, no. 1, pp. 1022–1027.
27. Son, D., Helmy, A., & Krishnamachari, B. (2004). The effect of mobility-induced location errors on geographic routing in mobile ad hoc and sensor networks: Analysis and improvement using mobility prediction. *IEEE Transactions on Mobile Computing, 3*(3), 233–245.
28. Spyropoulos, T., Psounis, K., Raghavendra, C.S. (2005). Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on delay-tolerant Networking* (pp. 252–259) ACM, New York, NY, USA.
29. Stojmenovic, I., Russell, M., & Vukojevic, B. (2002). Depth first search and location based localized routing and QoS routing in wireless networks. In *Parallel processing, 2000. Proceedings. 2000 international conference on*, pp. 173–180.
30. Vahdat, A., & Becker, D. (2000). *Epidemic routing for partially connected ad hoc networks*.
31. Wang, Y., Jain, S., Martonosi, M., & Fall. K. (2005). Erasure-coding based routing for opportunistic networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on delay-tolerant networking*, (pp. 229–236) ACM, New York, NY, USA.

Fraser Cadger B.Sc. (Hons), Ph.D. is currently employed as a System Test Engineer at Hitachi Data Systems. Fraser's Ph.D. project focussed on telemedicine in disaster recovery scenarios, and led to the development of routing and streaming protocols suitable for ad-hoc networks. In addition, Fraser also taught part-time as a technical instructor in subjects such as networking, operating systems, and object oriented programming.

Kevin Curran B.Sc. (Hons), Ph.D., SMIEEE is a Reader in Computer Science and group leader for the Ambient Intelligence Research Group at the University of Ulster. Dr Curran has made significant contributions to advancing the knowledge of computing evidenced by over 800 publications. He is perhaps most well-known for his work on location positioning within indoor environments, pervasive computing and Internet security. His expertise has been acknowledged by invitations to present his work at international conferences, overseas universities and research laboratories. He is a regular contributor to BBC radio and TV news in the UK and quoted in trade and consumer IT magazines. He was the recipient of an Engineering and Technology Board Visiting Lectureship for Exceptional Engineers and is an IEEE Technical Expert for Internet/Security matters since 2008. He is a member of the EPSRC Peer Review College.

Jose A. Santos was born in Maracay, Venezuela in 1973. He received his Electronic Engineering Degree from the Electronics Department, Universidad Simon Bolivar, Caracas, Venezuela in 1998, and his Ph.D. in Electronic Engineering from the School of Electrical and Mechanical Engineering, University of Ulster, Northern Ireland, UK in 2003. He has been a Lecturer in Computer Science at the School of Computing and Intelligent Systems of the University of Ulster since 2002, he is also a member of the Intelligent Systems Research Centre at the University of Ulster where he is part of the Ambient Intelligence Research Group. His current research interests lie in the fields of Ambient Intelligence and Mobile Computing, Wireless Communication Systems, Computational Intelligence, Biomedical Engineering, Sensor Technology and Robotics.

Sandra Moffett holds a B.A. (Hons) Business Studies, a MSc Computers in Education, a Post Graduate Certificate in Education and a Ph.D. in Knowledge Management. Sandra is currently employed as a Lecturer of Computer Science with the University of Ulster's School of Computing and Intelligent Systems, Magee Campus. As a research active member of staff she is also a member of the Ulster Business School Research Institute and the Intelligent Systems Research Centre (Ambient Intelligence Group). Her expertise on Knowledge Management contributes to her being one of the UK leading authors in this field with over 50 publications in peer reviewed Journals and International Conferences. Sandra is founder of the MeCTIP model and associated 'Benchmarking KM' tool, and has received both internal and external funding to support her research investigating KM application within UK organisations.