Fraser Cadger , Kevin Curran, Jose Santos, Sandra Moffett (2015) *Towards a location and mobility-aware routing protocol for improving multimedia streaming performance in MANETs.* Peer-to-Peer Networking and Applications, Vol. 8, No. 1, Jan 2015

Towards a location and mobility-aware routing protocol for improving multimedia streaming performance in MANETs

Fraser Cadger[1], Kevin Curran, Jose Santos, and Sandra Moffett

School of Computing and Intelligent Systems, Faculty of Computing and Engineering
University of Ulster, Londonderry, Northern Ireland, United Kingdom BT48 7PQ
Email: cadger-f@email.ulster.ac.uk

Abstract

The increasing availability and decreasing cost of mobile devices equipped with WiFi radios has led to increasing demand for multimedia applications in both professional and entertainment contexts. The streaming of multimedia however requires strict adherence to QoS levels in order to guarantee suitable quality for end users. MANETs lack the centralised control, coordination and infrastructure of wireless networks as well as presenting a further element of complexity in the form of device mobility. Such constraints make achieving suitable QoS a nontrivial challenge and much work has already been presented in this area. This paper proposes a bottom-up routing protocol which specifically takes into account mobility and other unique characteristics of MANETs in order to improve QoS for multimedia streaming. Geographic Predictive Routing (GPR) uses Artificial Neural Networks to accurately predict the future locations of neighbouring devices for making location and mobility-aware routing decisions. GPR is intended as the first step towards creating a fully QoS-aware networking framework for enhancing the performance of multimedia streaming in MANETs. Simulation results comparing GPR against well-established ad-hoc routing protocols such as AODV and DSR show that GPR is able to achieve an improved level of QoS in a variety of multimedia and mobility scenarios.

Keywords: Mobile Ad-Hoc Network, Quality of Service, Multimedia streaming, Geographic routing.

1. Introduction

The Internet is awash with streaming multimedia applications ranging from video-sharing websites such as YouTube to video and voice telephony applications such as Skype. As mobile devices such as smartphones and tablets become increasingly common, end-users are coming to expect such applications to be available on these devices. However, such applications require connectivity (typically to the Internet) which cannot always be guaranteed depending on the scenario. Ad-hoc networking allows for the creation of networks consisting only of wireless devices in which these devices act as the infrastructure as well as end-users. From a multimedia perspective, ad-hoc networks can potentially play a key role in facilitating the distribution of

---

[1] Corresponding author

Fraser Cadger is sponsored by a DEL Studentship from the University of Ulster and Northern Ireland Executive

multimedia content in instances where Internet connectivity is unsuitable (slow or intermittent) or unavailable.

Although ad-hoc networks present a number of exciting possible opportunities, both for multimedia streaming and other applications, they do suffer from some limitations. While Wireless LANs (WLANs) are often able to provide an almost identical degree of centralised control and coordination to that of wired LANs which helps them to surmount the challenges of the wireless medium, ad-hoc networks generally cannot rely on such features. When mobility is permissible this adds an additional set of problems that must be considered in the design and operation of ad-hoc networks. In infrastructure networks mobility is not considered as significant an issue as devices are able to seamlessly transition from one access point to another as the user moves throughout the physical domain. However, in MANETs there are no dedicated access points or base stations, and device mobility may lead to path breakage due to a node no longer being in a position to receive packets from other nodes on the path. This is naturally a particularly sensitive issue in multimedia networking, where the loss or delay of several packets can lead to a noticeable deterioration in end-user quality. Although approaches such as multipath routing aim to provide a degree of redundancy typically lacking in ad-hoc networks, they are themselves still vulnerable to path breakage caused by multiple node mobility and additionally require further overhead in order to build and maintain multiple links to each destination.

An alternative paradigm which displays a greater degree of resilience and adaptability to node movement is geographic routing. Geographic routing is highly localised, with nodes forwarding packets one hop at a time based on local information (i.e. physical proximity to destination). As such, geographic routing eliminates dependence on knowing an entire network topology which leads to a reduced update overhead as well as minimised exposure to the effects of mobility in distant nodes. Although geographic routing is typically more resilient to node mobility than conventional ad-hoc routing, it can itself be negatively impacted by mobility if neighbours have moved from their last known position. Location prediction algorithms have been explored in several geographic (or location-aware routing protocols) such as [1] and [2] in an attempt to anticipate future node mobility and counter its negative effects. More recently, the use of a neural network algorithm for predicting future device locations was implemented inside a geographic routing protocol and simulated in [3]. These experiments found that the NN (nearest-neighbour) algorithm was able to accurately predict future device locations in several different mobility scenarios and using different mobility models. The work of [3] is a continuation of [4] in which three machine learning algorithms (NN, decision tree and support vector regression) were analysed in Matlab for the purpose of analysing their ability to predict future MANET device coordinates. The NN algorithm was found to be the most accurate.

Although mobility is a significant factor in QoS, it is not the only one. Other factors such as network congestion can also have a critical impact on QoS. Therefore, while mobility may be considered as the most significant factor affecting QoS in MANETs, any successful MANET QoS protocol must not limit itself to only considering mobility. The best approach to managing QoS is to take a holistic view of the network and the environment which it operates and assess all possible factors affecting QoS and the degree of impact they will have. However, such a view would require either device-specific information or a meaningful way of abstracting such information. There are many challenges and obstacles to implementing such a protocol, not least deciding exactly what factors affect QoS. This paper does not propose such a protocol, but it does propose what can be seen as a step in this direction and used as the building blocks for such a protocol. Geographic Predictive Routing (GPR) is specifically designed for MANETs where human mobility is permissible. GPR uses an NN to predict where neighbouring devices will be at the time of transmission and then determine which neighbour is best-suited for selection as the next hop based on its location and other factors (such as levels of congestion).

GPR is intended as the first step in the development of a routing protocol for MANET streaming. Although GPR does not contain any explicit Quality of Service (QoS) mechanisms other than standard queuing procedures, it has been designed with the intention of optimising the key QoS parameters of reliability, delay, and delay variation (jitter). In simulations of various multimedia loads and differing scenarios of human mobility, GPR was found to outperform the likes of AODV [5], DSR [6], and DSDV [7] as well as another geographic routing protocol. Although AODV, DSR and DSDV are not geographic routing protocols, they were chosen for their popularity and because they represent the two main branches of ad-hoc routing; AODV and DSR are reactive (on-demand) routing protocols while DSR is a proactive (table-driven) protocol.

2. Previous Work

While there has been substantial research on streaming QoS in wired and wireless infrastructure networks many of the approaches used in these domains are inappropriate for MANETs. Although wireless infrastructure networks typically permit mobility this is handled through the use of handoff algorithms designed to transfer a device from one access point to another depending on their movement. Obviously such an approach is useless in MANETs as they do not contain access points. Similarly, traditional approaches to managing QoS in infrastructure networks such as overprovisioning or complicated marking and traffic marking schemes are not always appropriate for MANETs either because they require resources and processing power in excess of those found in MANET devices or simply because they do not take into account the unique nature of MANETs. While the issue of processing power is becoming less of a concern due to the increasing processing power and decreasing price of devices such as laptops, smartphones and tablets the lack of explicit support for MANETs is a crucial factor in the incompatibility of traditional QoS mechanisms with MANETs.

As such, most research into ad-hoc and MANET streaming has focused on developing new mechanisms (sometimes based on infrastructure techniques) for handling QoS. Several of these approaches have focused on the issue of path maintenance using either multipath [8] or single path approaches [9] [10]. Both single path and multipath approaches have their own weaknesses. Single path naturally lacks the redundancy of multipath, meaning that packets are often ‘stranded’ mid-stream if a path collapses. Multipath attempts to tackle this by introducing two (or more) paths and typically allowing packets to be swapped between paths should one path fail, however it does incur a significantly higher maintenance cost than single path routing. Multicast [11] and combinations of multicast and unicast [12] have also been explored as a means of delivering packets from a single source to several recipients. However, multicasting can carry a large overhead, and when involving large numbers of nodes is comparable to broadcasting. Even approaches which are explicitly developed for MANETs fail to tackle ad-hoc networking’s reliance on (and need for) maintaining full topology information in a network which is constantly changing (due to both node mobility and other dynamics).

In contrast, geographic routing frees nodes from the need to distribute, store, and maintain topology information. By relying only on information about 1-hop neighbours, geographic routing localises forwarding decisions and essentially eliminates the concept of traditional routes. This allows geographic routing a degree of resilience to node mobility and topology changes not found in traditional ad-hoc routing protocols. However, geographic routing can still be negatively affected by node mobility if they are not aware that a neighbour has moved from its last known position, or has moved out of its transmission range. In basic greedy geographic routing nodes are susceptible to the local maximum where they cannot find a neighbour closer to the destination than themselves and drop the packet to avoid it travelling backwards and potentially creating a loop. To counter this, face routing which is based on Compass II routing [13] acts as a recovery mechanism which uses planarised graph traversal to overcome the local maximum. However, face routing is less efficient than greedy routing and so many hybrid-schemes have been proposed such as Greedy Perimeter Stateless Routing (GPSR) [14] which use greedy routing as the default and then switch to face routing if the local maximum is encountered.

Face routing is itself susceptible to the problem of node mobility, as node movement can cause problems with the planarization of graphs if nodes have moved position, while the issue of sending a packet to a neighbour outside of a node’s transmission range is still an issue. To counter the impact of node mobility, location prediction has been used alongside geographic routing protocols in approaches such as [2], [15] and

[1]. Of these, the latter two focus on QoS by using location prediction to determine the ability of neighbouring nodes to fulfil QoS characteristics such as reliability and delay variation. Location prediction has also been used to great success in the area of wireless infrastructure networks where prediction algorithms from machine learning such as the Hidden Markov Model [16]. However, methods such as [16] rely heavily on infrastructure by performing location prediction on a discrete series of access points or cells and are therefore unsuitable for use in ad-hoc networks.

To date, the only application of machine learning to the problem of MANET location prediction is [3] which uses a NN running on top of GPSR to predict the coordinates of MANET devices with a high degree of accuracy (less than 1m error in most scenarios). There are some questions of the suitability of NNs for use in real-world scenarios where devices such as tablets smartphones typically have tight processing and memory concerns. However NN algorithms have been successfully implemented on Android mobile phones for purposes ranging from posture recognition [17] to robot control [18]. Additionally, NNs have been implemented on the iPhone for the purpose of pedometer-based activity monitoring [19] and on the iPad2 [20] for eye tracking. It is the intention of the authors to implement GPR on a testbed of six Android smartphones (five Samsung Galaxy Minis and one HTC Nexus One) to determine its performance in real-world scenarios and whether the NN location prediction algorithm is suitable for use on current hardware.

At present, there is a lack of MANET protocols which explicitly incorporate QoS mechanisms into the routing. Similarly, most MANET streaming protocols do not sufficiently address the weaknesses of conventional ad-hoc routing with regards to dependence on network-wide topologies. Geographic routing in combination with location prediction can be used as a means of predicting node mobility and countering it, while reducing the amount of information about the network that needs to be stored and maintained. This paper presents a geographic routing protocol using location predictions powered by a NN algorithm and other neighbour information to improve routing performance in MANET streaming scenarios compared to other ad-hoc routing protocols. GPR is intended as a step towards the development of a MANET routing protocol which uses location and QoS predictions to facilitate the streaming of multimedia at the best possible quality.

3. Design and Implementation

GPR focuses on four factors affecting QoS in geographic routing in MANET scenarios; mobility, congestion, neighbour range and neighbour information accuracy.

3.1 Mobility

Mobility is a significant factor in all networks where mobility is permissible. However its effects are typically more pronounced in ad-hoc networks, which lack the coordinated handover procedures found in infrastructure networks. The effects of mobility in MANETs can be broken down into three levels; individual, path, and device. Infrastructure networks typically span a well-defined physical area (a building, a city, a coverage cell, etc.), whereas MANETs as a result of their ad-hoc nature do not always have a clearly delimited physical network area. As a result, network coverage in MANETS is defined in terms of proximity to other participating devices instead of proximity to known access points. Therefore, mobility in MANETS is potentially more likely to lead to complete disconnection from the network.

At the path level, mobility can have the effect of disrupting a path or route if at least one device on the route moves out of position. If the movement is enough to remove that device from the path then other devices can potentially waste energy and other resources sending packets to a device which cannot receive them. Once the path failure has been discovered, the protocol must attempt to switch routing to a backup route or create a new one. While this particular issue is more of a concern for traditional end-to-end protocols, it can also adversely affect geographic routing. Even though geographic routing does not create an explicit end-to-end route, it does have an implicit path and while forwarding is carried out on a hop-by-hop basis, if a node is unaware that a neighbour is outside its transmission range it can still waste energy and other resources forwarding to that neighbour. Perhaps more significantly, mobility can lead to network-wide problems and possible network partition.

Although the effects of mobility cannot always be prevented, with the use of an accurate location prediction algorithm it is at least possible to anticipate them. Earlier, the use of location prediction in geographic routing was discussed. The use of location prediction is logical as it allows protocols to make geographic routing decisions based on where a node will actually be at the time of routing, and not where it was according to the previous updates. From a purely geographic routing perspective, this helps prevent the making of sub-optimal routing decisions. For instance, if node A has two neighbours (B and C) and according to their last updates, B is located closer to the destination than C, then B will be chosen as the next hop. However, since the last updates were received, B could have moved and may now be further away from the destination than C. Furthermore, B could actually be moving in a direction opposite to the direction and if given the packet could in turn send the packet to a device located further away from the destination than node A. Location prediction can help avoid this sort of scenario, as it would allow node A to determine that B was further from the destination than C and select node C as the next hop instead of node B.

Location prediction can be used to solve other issues caused by mobility that are not confined to geographic routing. The problem of path disruption can also be solved through the use of location prediction. If a node is able to predict the future location of its neighbours, then it can use this information to determine if a path break will be caused by future neighbour mobility, and if so at what time it will occur. In a conventional end-to-end protocol the node could then take action in the form of switching to a backup route or beginning the process of creating a new one. For geographic routing, knowing when a neighbour will move out of its transmission range will allow it to avoid sending messages to a neighbour when it becomes unavailable. Therefore, location prediction can be seen as useful in avoiding the effects of two of the three categories of mobility-induced error. It should be noted that the other category (network partition) is unavoidable from a network-perspective as it would involve altering mobility; either by having the node about to cause partition remain in place, or having other nodes move to that location.

GPR uses a NN algorithm in order to accurately predict future neighbour locations using two previous coordinates and their times. A NN-based approach was chosen based on the results obtained from [4] in which a NN algorithm outperformed two other machine learning algorithms and [3] in which a NN algorithm was implemented inside a geographic routing protocol and simulated in several scenarios of varying mobility. In GPR, location predictions are used as follows; when a forwarding decision is being made, for each neighbour their two previous coordinates and corresponding timestamps are passed along with the current time to the NN algorithm. Two coordinates are used here because previous works such as [4] and [1] also used this approach, and it was found that increasing the number of coordinates (to three or four) did not significantly impact on prediction accuracy. As it is possible that MANET devices will store this information in memory, it is desirable to reduce the amount of information needed to make predictions as much as possible.

The NN algorithm then provides the predicted coordinates which are initially evaluated using the geographic routing criteria and then compared to the node's own transmission range. If the coordinates are outside of the node's transmission range then that neighbour will not be selected as the next hop. Regarding the geographic routing criteria, in greedy geographic routing this would simply be a case of determining the node nearest to the destination. However, GPR uses a more complex criteria which will be discussed in the following sections.

3.2 Congestion

Network congestion is one of the most commonly studied problems in almost all areas of networking. It can be simply defined as the point where traffic exceeds resources, and its effects are delay, delay variation, and possibly packet loss. Most of this attention has come from the field of infrastructure networks; however the solutions applied in this domain such as over-provisioning or complex traffic policies are typically unsuitable for ad-hoc networks. The difficulty in dealing with MANET congestion is that unlike infrastructure networks which are carefully designed, coordinated, and implemented to meet specific goals and support specific applications ad-hoc networks are more spontaneous and less coordinated. However, there is still a need to deal with issues such as congestion, particularly in multimedia streaming where there is both a potentially large volume of traffic as well as strict QoS requirements.

As GPR is a routing protocol, it attempts to control congestion through routing mechanisms, although it does make use of information from the MAC layer and queues. GPR's congestion control mechanism is an extension of the basic congestion control (CC) algorithm found in the implementation of GPSR provided by [21]. The exact origins of the CC algorithm are unknown, as it does not appear in the original GPSR code provided by [14] and there is no reference to the code in either [22] or [23]. The basic CC variant works as follows; like the standard neighbour selection algorithm in GPSR nodes first calculate the distance between themselves and the destination, and determine the distance between each neighbour and the destination. A neighbour is selected as the next hop if its distance to the destination is less than the node's own distance to the destination and the value of 't' for that node (calculated using the CC equation) is less than the previous best value (which is by default set to 1). The CC algorithm is shown in (1).

t = alpha * tab[i]-> load/ 100 + (1 – alpha) * (new_dist / distance) **(1)**

In (1) tab[i]->load corresponds to the current neighbour's MAC-layer load and new_dist is the distance between the neighbour and the destination. Distance is the distance between the node and the destination. Alpha is a 'control' variable set in the tcl configuration file which determines how 'aggressively' congestion control is applied. The equation works as follows, on the left hand side, alpha is used to multiply the neighbour's MAC load and the product of this is then divided by 100. MAC load represents the percentage of time during which a node's MAC layer is utilised, and it is divided by 100 here to give the decimal.

This simple weighted equation allows nodes to create a trade-off between the need to minimise congestion (as represented by load) and geographical distance. When performing pure geographic routing other considerations such as congestion are not explicitly taken into account and this can therefore lead to issues such as high delay or packets being dropped, if nodes are only selected based on their physical location. Similarly, using congestion alone is unsuitable because it provides no other knowledge of the neighbour's state. Combining congestion-aware and geographic routing therefore allows nodes to utilise the simple (and often effective) method of geographic routing while providing some consideration of how neighbours are affected by congestion. This essentially allows for a simple form of load-balancing, in which neighbours who are overly congested be passed-by in favour of less congested neighbours.

3.3 Neighbour Range

A common problem with greedy geographic routing is the local maximum in which a device which is unable to find a neighbour closer to the destination than itself must drop the packet. The reasoning behind this is that if a packet was to travel physically backwards then a routing loop could occur. Although there have been several attempts at solving the local maximum problem such as face routing, it should be noted that they are significantly more computationally intensive than basic greedy geographic routing. In addition to the extreme of the local maximum problem, another issue with greedy geographic routing is that as nodes are only aware of their neighbour's locations they can often make decisions in which the neighbour closest to the destination is not the best next-hop. A less dramatic example would be where a node makes a forwarding decision that turns out to be suboptimal as although the chosen neighbour is closer to the destination than the current node, that neighbour's neighbours are less optimally placed than another neighbour's neighbours resulting in an increased hop count and therefore delay. Therefore, while it is desirable to overcome these limitations it is also desirable to do so without compromising the locality of geographic routing. Doing so therefore requires some degree of information about a neighbour's neighbours, but only as little as is possible to make improved decisions.

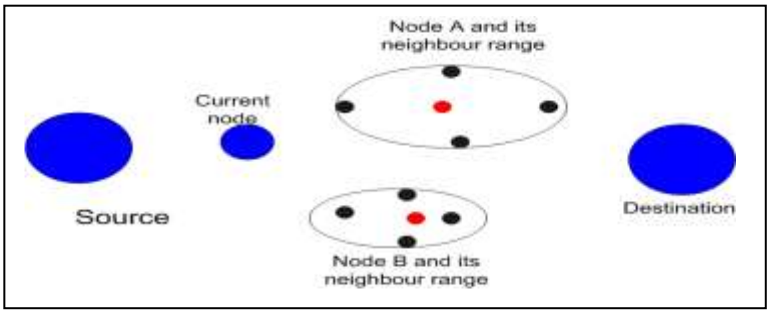

Figure 1: Scenario in which the neighbour range metric is used to select a neighbour with a better neighbour range

GPR presents a suitable compromise between these two competing demands through the use of a metric known as average neighbour range. Total neighbour range determines the range of a node's neighbours across all four compass points. This is done by calculating the neighbours located the furthest for each compass point, then calculating the distance between that neighbour and the node, and finally calculating the average of all these distances. This allows nodes to determine the average range of their neighbour positions which they then share with other nodes through a total neighbour range field in beacon messages. Therefore when a node goes to make a forwarding decision they are able to factor in the average range of a neighbour's neighbours. This allows nodes to determine with some degree of accuracy how likely a neighbour is to be able to find a suitable next hop itself. Figure 1 depicts the operation of the neighbour range metric; in this scenario nodes A and B are both located in similar positions, however node A clearly has a better neighbour range than node B. Using the neighbour range metric the current node is able to select node A over node B and thus decrease the overall distance to the destination.

While having a good average neighbour range does not necessarily mean that a neighbour will find a desirable next hop, it does help minimise the chance of selecting a bad next hop. Similarly, the total neighbour range metric can also be seen as significantly reducing the likelihood of selecting a neighbour which will encounter the local maximum problem as nodes that have a large neighbour range are more likely to find a suitable next hop than those with a smaller range. However, although the total neighbour range is not perfect, its inclusion has been observed to have yielded positive results in terms of both delay and reliability which confirms its significance. Figure 2 depicts the algorithm for calculating average neighbour range.

```
1. FOR i < size of neighbour table
    a. IF this is the first run
        i. Set north_difference = neighbour_y_coordinate - our_y_coordinate
        ii. Repeat for East, South, and West
    b. END IF
    c. ELSE
        i. IF neighbour_y_coordinate > north_difference
        ii. Set north_difference = neighbour_y_coordinate
        iii. Repeat for East, South, and West
    d. END ELSE
2. END FOR
3. Set total_difference = north_difference + east_difference + south_difference + west_difference
4. Return total_difference
```

Figure 2. Pseudocode for calculating total neighbour range

3.3 Neighbour information accuracy

Geographic routing protocols, like other types of ad-hoc routing protocols, rely heavily on information obtained from other nodes. This means that the information obtained from other devices could be unreliable or inaccurate – either intentionally or unintentionally. The former would most likely be as a result of malicious intent; either hoping to disrupt the network's operation or gain advantage. While the latter could be caused by a variety of factors relating to both the device's awareness of its own state or the device performing routing possessing out-of-date information about some or all of its neighbours.

In this paper we focus on the possibility of devices having out-dated information about its neighbours. The reasoning behind this is that it is easier to simulate, as it is possible that update messages will be frequently dropped or delayed. However, simulating inaccurate information as a result of devices possessing inaccurate information about themselves is more difficult. For instance, although it is known that commercial GPS units offer an accuracy of 2m, in ns-2 devices obtain their coordinates directly from the simulator and so they are always correct at the time of reading. Similarly, the simulation of intentional misinformation is a topic in itself, and while a worthy area of research, it is not the focus of this paper, although it could provide an important area of future research.

In GPR nodes obtain neighbour information from regular beacon messages. Depending on how long a beacon's lifetime is (this is a configurable option), the update may have been stored for a considerable period of time. Although the NN prediction algorithm can be seen as countering this, it is still susceptible to problems caused by out of date locations. As the NN prediction algorithm makes its predictions based on two previous locations and their timestamps, if the node was following a particular pattern of mobility for a period of time and then changed dramatically (such as stopping, changing direction, or altering speed significantly) the NN might not be able to predict this change in mobility and would instead base its predictions on the previous pattern of mobility. It is also worth noting that a neighbour could have moved away from a node's range or have died, and the neighbour table entry for it is irrelevant and its presence runs the risk of a node sending packets to a neighbour that cannot receive them. Therefore, a measure that indicates the 'freshness' of a neighbour (or at least its updates) is required. Such a measure should be independent of beaconing timers as it is intended as a relative measure for comparing neighbours with each other. Freshness is therefore simply the difference between subtracting the timestamp of a neighbour's last received beacon from the current time as shown in (2). Freshness is then incorporated into (1) as shown in (3). Note that the choice of 6.5 as the divisor for freshness was decided through experimentation with a range of values. Again, it is likely that different values will perform better in different scenarios, and 6.5 is not suggested as a universal value.

freshness = current time – tab[i]->ts **(2)**

t = alpha * tab[i]-> load/ 100 + (1 – alpha) * (new_dist / distance) + (freshness / 6.5) **(3)**

3.4 Implementation

GPR is based on the code of GPSR with several modifications similar to [3] to allow for location prediction and the other factors discussed previously, to be used in the routing process. For location predictions, a two-layer NN with 15 hidden neurons (on one hidden layer) using the iRPROPR training algorithm [24] is implemented on top of the GPSR protocol as described in [3]. By default, GPSR broadcasts beacon messages that allow nodes to discover new neighbours and ensure existing neighbours are still alive or within transmission range. These beacon messages also contain node locations and they are the means by which nodes learn their neighbours' positions. In [3] GPSR is modified to store the previous coordinates of a neighbour and these coordinates together with their timestamps and the current time are used as the inputs for the NN algorithm which predicts the node's current x and y coordinates. Two previous coordinates are used because they are the minimum necessary to predict mobility, while experiments with three previous coordinates did not find any significant improvement in prediction accuracy. Experiments with more than three coordinates could take place in the future, however as GPR will be implemented on smartphones it is desirable to use as little memory as possible. Although the NN configuration and training is exactly the same as in [3], however the application and use of the NN algorithm is slightly different. Whereas [3] uses the NN algorithm to predict coordinates for the basic greedy routing algorithm, GPR instead uses a modified version of the CC greedy routing algorithm which is contained in the ent_find_shortest_cc() method of GPSR.

The first step in altering the existing CC algorithm to produce GPR was to use location predictions instead of previous coordinates in calculating new_dist. This is done in the same manner as the modifications in [3] and passes two previous coordinates and their timestamps for each neighbour as well as the current time to the NN algorithm which then returns predicted coordinates. In cases where two previous coordinates are unavailable (for instance, at the start of routing or when a node has not received two beacons from a neighbour) or there is a problem predicting coordinates then a node's most recent coordinates are used. The latter is simply a contingency intended for real-world deployment where it is possible that the location system could experience failure (for instance, if GPS was being used and a node temporarily moved inside or there was a system error).

The next modification to the CC algorithm involved an alteration of (1) in order to incorporate a measure of how reliable a node’s updates area; this was achieved by first calculating freshness as shown in (2) and then implementing the modified CC algorithm shown in (3). The final step in implementing GPR was to add the neighbour range calculation. This was firstly done by creating a new method called neighborRange() which is called each time a node is sending a beacon. This method calculates the average range covered by a node’s neighbours as shown in the pseudocode of Figure 2. The neighbour range calculations are then incorporated into the ent_find_shortest_cc() method in the form of an if statement which comes after (3) and ensures that only nodes with a better neighbour range than the average are selected.

4. GPR Evaluation

GPR was simulated and evaluated against four other geographic routing protocols; AODV, DSR, DSDV, and the unmodified version of GPSR. All simulations were conducted using ns-2.34 in a simulation area of 1500m x 300m for a period of ten minutes. Topologies of 10, 30, and 50 nodes were simulated. An alpha value of 0.01 was used, while beaconing was set to 13.6s – the high beacon interval made possible through the use of location predictions, both of these values were decided through simulation with different values. The Reference Point Group Mobility model (RPGM) [25] (with a maximum speed of 2.5m/s and a maximum pause time of 20s) was used for simulating human mobility. The following QoS scenarios 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 360-480p 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, due to the difficulties of simulating with real video in ns-2. This traffic is intended to model general network-level characteristics of streaming traffic. As such it does not take into account application specific characteristics. For instance, live streaming applications often drop packets that exceed acceptable levels of delay, which would have the effect of keeping delay low but decreasing packet delivery.

Three metrics were used to assess the protocols' abilities to cope with streaming multimedia traffic in a MANET scenario; reliability (percentage of packets successfully delivered), delay, and delay variation (jitter). Delay and delay variation are measured in milliseconds. Tables 1-3 show the results from these simulations. Figures 3-5 provides a visualisation of these results for (GPSR is excluded from Figure 4 because its large spike at the 50 node scenario would skew the chart).

Protocol/No. of Nodes	10 Nodes	30 Nodes	50 Nodes
GPR	84.6%	98.7%	85%
AODV	84.7%	99.6%	83.2%
DSR	83.2%	99.9%	82.8%
DSDV	84.8%	98.5%	79%
GPSR	84.7%	99.9%	86.9%

Table 1: Packet delivery for each protocol in 10, 30, and 50 node scenarios.

Protocol/No. of Nodes	10 Nodes	30 Nodes	50 Nodes
GPR	1.8	2.1	49.8
AODV	1.2	3.3	59.3
DSR	1.8	2.3	163
DSDV	1.9	22	157
GPSR	1.9	22.7	1265

Table 2: Average end-to-end delay for each protocol in 10, 30, and 50 node scenarios.

Protocol/No. of Nodes	10 Nodes	30 Nodes	50 Nodes
GPR	5	95	225

AODV	6	21.4	207
DSR	1	40	600
DSDV	5	11.3	387
GPSR	5	11	474

Table 3: Average end-to-end delay variation for each protocol in 10, 30, and 50 node scenarios.

In the 10 node scenario, all protocols appear to perform well in terms of delay and delay variation, easily achieving average levels of delay well below the 150ms desirable threshold. Although AODV achieves the lowest level of delay in this scenario, there is really very little to separate the different protocols. While for delay variation DSR appears to come out on top with the other protocols lagging behind slightly. The good results for delay and delay variation are in contrast to the lower results for reliability. Again, there is very little difference in the results achieved by the different protocols with all results in this category being between 84-85%. The poorer performance by all terms of reliability is most likely due to the sparseness of the network, given that there are only 10 nodes all of which are in motion for some or all of the 10 minute duration. It is possible that levels of reliability under 90% are a factor in the low levels of delay as packets are dropped instead of being routed which leads to lower levels of queuing and congestion for those packets which can be routed. Conversely, the low levels of delay would seem to indicate that few packets are being dropped as a result of queuing and congestions.

In contrast to the results from the 10 node scenario, the 30 node scenario shows a significant increase in reliability and a slight increase in delay. All protocols achieve a packet delivery rate of at least 98% with GPSR and AODV being very close to 100% reception, although it is noticeable that in doing so GPSR incurs an average delay of 22.7ms. DSDV incurs a similar level of delay, but the other protocols all manage to maintain levels of delay around 2-3ms with GPR achieving the lowest level of delay.

The universal increase in reliability is most likely a direct result of the increased network size meaning that almost all destinations are reachable for the duration of the stream and there are therefore fewer dropped packets. Although conventional wisdom states that larger networks suffer from reduced levels of QoS, it is also true that if a small number of mobile nodes are grouped in a large area then it is possible that some nodes will be temporarily unreachable resulting in occasional packet drops.

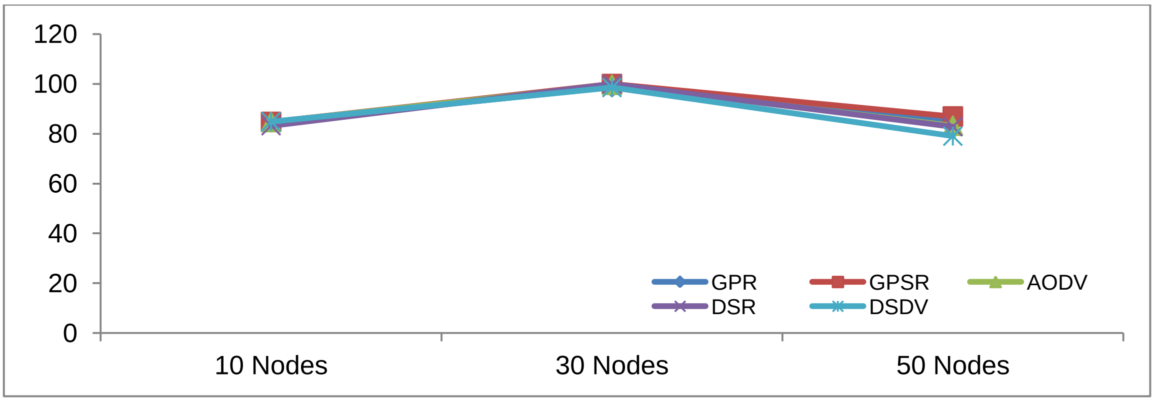

Figure 3: Number of successful packets delivered for selected protocols.

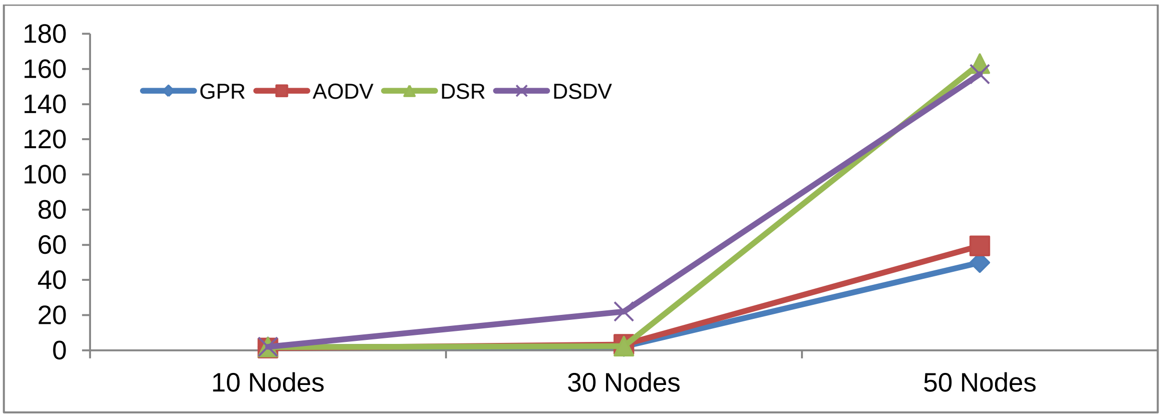

Figure 4: Average end-to-end delay of selected protocols.

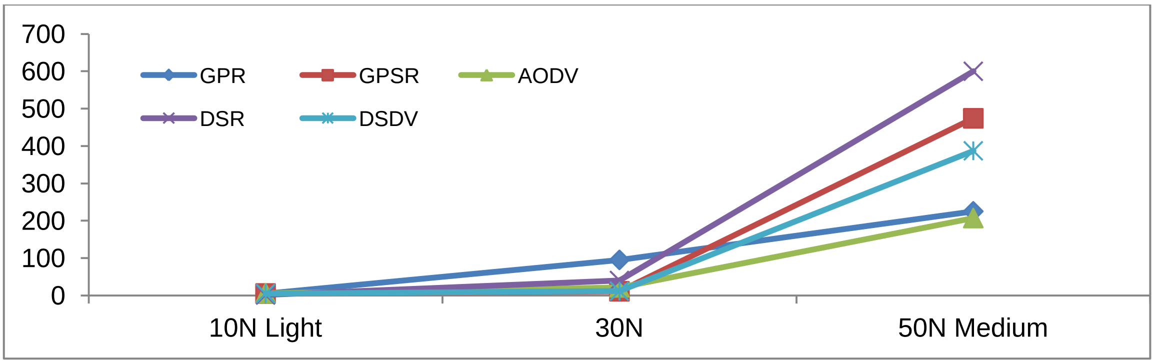

Figure 5: Average end-to-end delay variation of all protocols.

As there is only a very small increase in delay levels for most of the protocols (with the exception of GPSR and DSDV), the results from the 30 node scenario would seem to indicate that the increase in nodes has mostly been beneficial in terms of increasing packet delivery while preventing large volumes of congestion and packet drops despite an increase in traffic volume as evidenced by the low levels of delay. Although delay remains low, for all protocols, delay variation rises significantly – most notably for GPR. The increase in delay variation is probably due to the increased number of nodes leading to packets experiencing different levels of delay depending on the state of the network (both in terms of traffic and mobility) as well as the possibility for paths to change. This is possibly why the increase is most pronounced in GPR; as GPR does not have set paths it is likely that packets will take considerably different routes to the destination each time, therefore resulting in differing levels of delay and a higher than average level of delay variation.

While the increase in the number of nodes led to an overall improvement in QoS terms (significant reliability increase and very small delay increase) despite the increased traffic, the transition from the 30 node to 50 node scenario has led to an overall deterioration in QoS levels. This is expected as again both the load and the size of nodes are being increased. Although the increase from 10 to 30 nodes appears beneficial, it would seem that going from 30 nodes to 50 nodes brings the network to the point where congestion becomes an issue. What is perhaps most significant about this scenario is that three of the five protocols display a level of end-to-end delay greater than the desirable bound of 150ms, although of these three two are still within the acceptable limit of 300ms (while GPSR with over 1s of delay clearly performs poorly in this respect) [26]. Of the two protocols achieving less than 150ms delay, GPR is the best performer with 49.8s compared to AODV with 59.3ms. Although none of the protocols manage to achieve 90% reliability, it is worth noting that these constraints are largely based on metrics derived from the use of infrastructure networks rather than MANETs. Although noticeably poor quality audio and video will always be seen as detrimental by end-users, there is anecdotal evidence to suggest that they are willing to tolerate lower quality multimedia when using MANETs than when using infrastructure networks. Therefore, although achieving the highest level of QoS should always be the goal in developing a multimedia streaming system, it should also be recognised that it is not always possible to achieve the same levels of performance as can be achieved on infrastructure networks. Therefore, in the 10 and 50 node scenarios, the protocols that achieved less than 150ms delay and 84-85% packet delivery should not be considered failures. It is worth noting that the best performing protocol in terms of reliability for the 50 node scenario is the unmodified GPSR which achieves 86.9% packet delivery, but has an average delay of 12565ms. Again, when considering delay it is important to recognise that infrastructure levels of delay are not always attainable in MANETs, and that it could be argued that due to buffering for on-demand streaming delay is a less important metric than reliability. However, a delay in excess of 1s is clearly too high to achieve any realistic level of Quality of Experience (QoE).

There has again been an overall increase in delay variation, with 207ms (AODV) being the lowest recorded level. Although there is not a clearly predefined limit for jitter, there are suggestions of 100ms as a maximum. As with the thresholds for delay and reliability, this is based on infrastructure networks and it is possible that higher values could be tolerated for ad-hoc networks (particularly MANETs). However, levels of delay such as those exhibited by DSR, GPSR, and DSDV are by any standards clearly unacceptable for any form of multimedia streaming and video calling in particular. It is unclear whether the values of delay variation displayed by AODV and GPR (both in excess of 200ms but under 300ms) could be considered acceptable by end-users who are aware of the constraints of MANET streaming and are willing to accept a lower level of QoE compared with using conventional video calling and streaming applications over the Internet. It is perhaps not surprising that MANETs should suffer from large values of delay variation as mobility can lead to packets in the same stream taking different routes due to links being disrupted or modified as a result of node mobility. Therefore, even the non-geographic routing protocols can be affected by mobility-induced delay variation. As delay itself has increased significantly since the 30 node scenario, the further increase in delay variation for the 50 node scenario is unsurprising if also undesirable. It is interesting to note however, that GPR previously performed quite poorly in terms of delay variation for the 30 node scenario but performs relatively better in the 50 node scenario by coming in second place. Although its 225ms of delay variation is over the acceptable limit for infrastructure networks it avoids the ‘huge’ spikes of DSR, GPSR, and DSDV. Although GPR does not explicitly consider delay variation although by avoiding neighbours with heavy loads (where possible) GPR does help reduce the odds of encountering large levels of delay which themselves often account for large levels of delay variation. The low level of delay encountered by GPR in this scenario would seem to confirm this idea.

4.1 Analysis of GPR’s Performance

Regarding GPR itself, the 50 node scenario is perhaps most significant as it is the scenario in which GPR emerges as the clear best performer. By coming second in terms of reliability and delay variation and first in terms of delay, GPR can be said to be the overall best performer in this scenario. GPR is one of only two protocols that has an average delay within the bounds of the 150ms acceptable limit and compared to AODV which also manages that feat GPR has both a lower level of delay (9.5ms lower) and a higher level of reliability (85% compared to 83.2%), although AODV does have a better (18ms lower) level of delay variation. Similarly, only one other protocol has a level of reliability higher than GPR in this scenario (GPSR with 86.9%) but also has a level of delay well over the acceptable threshold of 150ms. The most significant aspect of GPR's performance is its comparatively low delay. This can be explained by considering the mechanisms GPR uses to reduce delay by taking the existing CC algorithm and modifying it to incorporate neighbour range, neighbour freshness, and node transmission range. While the original CC algorithm's use of load is undoubtedly responsible for ensuring a low level of delay, tests using an unmodified version of the CC algorithm did not achieve as low a level of delay or as a high level of reliability, therefore the modifications have clearly had a positive effect.

Although only neighbour range was explicitly designed with reducing delay in mind and it is not a part of the CC algorithm itself, the other modifications can also have a positive effect on delay. Neighbour freshness is an important metric in general as it acts as a means of reducing the chance that out-dated and inaccurate neighbour information is used; in terms of delay this can help nodes avoid selecting neighbours whose position, neighbour range, or levels of congestion may have changed for the worse since an update was last received. Similarly by reducing the impact of out-dated neighbour information, the freshness metric allows for the beacon period to be increased from 0.5s to 13.6s which reduces the amount of control traffic in the network thus lowering delay and increasing reliability. While the transmission range extension which checks to ensure a neighbour’s predicted location is within the node’s transmission range would not explicitly affect delay, it is important in ensuring the GPR’s level of reliability does not drop below 85% - in contrast to AODV, DSR and DSDV which all have less than 85% reliability as well as unacceptable levels of delay. While GPR’s level of delay variation is higher than the desirable threshold (based on infrastructure networks) it is still the second best, and in comparison to DSR, GPSR, and DSDV is at a much better level.

It should be noted that while a high level of delay is experienced in these simulations, in real-world applications (particularly live streaming) it is likely that packets exceeding a pre-determined latency level would be dropped (so as to prevent excessively late packets wasting resources), and this in turn would affect packet delivery rates. However the purpose of this simulation is to provide a general study of how the protocol handles traffic modelled on streaming protocols, and therefore incorporates features of both live and on-demand streaming traffic

6. Conclusion

This paper describes the design and simulation of GPR a geographic routing protocol which uses NN location predictions alongside an enhanced CC algorithm to improve routing performance in MANET streaming scenarios. The purpose of this work is to develop GPR as the first stage in the creation of a complete MANET streaming solution. This will eventually lead to the development of a protocol that combines location predictions with other forms of context-awareness to make intelligent QoS predictions. GPR can therefore be considered the location prediction ‘core’ of this future protocol. GPR is also a practical application of the work described in [3] where a NN for location prediction was implemented inside GPSR and its prediction accuracy observed. Unlike [3], GPR applies NN prediction to the CC algorithm instead of the basic greedy forwarding algorithm. Further modifications were made to the CC algorithm to allow it to incorporate other neighbour information into its decision making. This included comparing the node's transmission range to the distance between the node and its neighbour in order to determine whether the transmission range was large enough to allow successful reception. Another major change was the incorporation of a freshness metric into the CC algorithm which allows nodes to determine how up to date a neighbour's information is and help avoid the use of out of date neighbour information. While neighbour range metric allows node's to determine whether a neighbour is itself likely to find a suitable next hop.

GPR was then simulated against AODV, DSR, and DSDV as well as the unmodified GPSR running the standard greedy routing algorithm in scenarios of 10, 30 and 50 nodes using the RPGM. In the first two scenarios, all protocols achieved similar results and there was little to distinguish one protocol as standing out from the rest. However, in the 50 node scenario GPR stood out as the only node to achieve a good level of delay and a level of reliability close to 90% (85%). While GPR (and all of the other protocols) failed to achieve the 90% reliability level in the 50 node scenario, it was suggested that this threshold is possibly too high for MANET systems later on the development of dedicated MANET QoS thresholds was discussed. With this in mind, GPR's performance in the 50 node scenario was clearly the best overall and GPR can be considered a success in these experiments. Similarly, although all nodes displayed levels of delay variance over 100ms in the 50 node scenario, GPR still came second best despite being a geographic routing protocol that does not use fixed paths.

It is again worth nothing that neither GPR nor any of the other protocols are explicitly designed to be aware of and manage QoS. In light of this, these results can be considered as positive as they indicate the potential for a geographic routing protocol to outperform other ad-hoc routing protocols in MANET multimedia streaming scenarios. At present, work is taking place to implement GPR on a testbed of six Android smartphones with the purpose of evaluating GPR's performance in the real-world using actual MANET devices. It is hoped that GPR can therefore act as the first stage in the development of location and QoS-aware streaming solution for MANETs. On the whole, the results described in this paper can be considered a positive start.

References

[1] S.H. Shah, K. Nahrstedt, Predictive location-based QoS routing in mobile ad hoc networks, in: Communications, 2002. ICC 2002. IEEE International Conference On, 2002.

[2] I. Stojmenovic, M. Russell, B. Vukojevic, Depth first search and location based localized routing and QoS routing in wireless networks, in: Parallel Processing, 2000. Proceedings. 2000 International Conference On, 2000.

[3] F. Cadger, K. Curran, J. Santos, and S. Moffett "Accurate prediction of MANET device locations using Neural Networks". Under review. 2013.

[4] F. Cadger, K. Curran, J. Santos, S. Moffett, MANET Location Prediction Using Machine Learning Algorithms, in: Y. Koucheryavy, L. Mamatas, I. Matta, V. Tsaoussidis (Eds.), Wired/Wireless Internet Communication, Springer Berlin Heidelberg, 2012: pp. 174-185.

[5] C. Perkins, E. Belding-Royer, S. Das, Ad hoc On-Demand Distance Vector (AODV) Routing, (2003).

[6] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, in: Mobile Computing, Kluwer Academic Publishers, 1996: pp. 153-181.

[7] C.E. Perkins, P. Bhagwat, Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers, in: Proceedings of the Conference on Communications Architectures, Protocols and Applications, ACM, New York, NY, USA, 1994: pp. 234-244.

[8] E. Setton, X. Zhu, B. Girod, Congestion-optimized multi-path streaming of video over ad hoc wireless networks, in: Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference On, 2004: pp. 1619 - 1622 Vol.3.

[9] S.Y. Oh, M. Gerla, A. Tiwari, Robust MANET routing using adaptive path redundancy and coding, in: Communication Systems and Networks and Workshops, 2009. COMSNETS 2009. First International, 2009: pp. 1-10.

[10] T. Xu, Y. Cai, Streaming in MANET: Proactive Link Protection and Receiver-Oriented Adaptation, in: Performance, Computing, and Communications Conference, 2007. IPCCC 2007. IEEE Internationa, 2007: pp. 178-185.

[11] W. Wei, A. Zakhor, Multiple Tree Video Multicast Over Wireless Ad Hoc Networks, Circuits and Systems for Video Technology, IEEE Transactions On. 17 (2007) 2-15.

[12] W. Wei, A. Zakhor, Multipath unicast and multicast video communication over wireless ad hoc networks, in: Broadband Networks, 2004. BroadNets 2004. Proceedings. First International Conference On, 2004: pp. 496-505.

[13] E. Kranakis, S.O.C. Science, H. Singh, J. Urrutia, Compass Routing on Geometric Networks, in: In Proc. 11 Th Canadian Conference on Computational Geometry, 1999: pp. 51-54.

[14] B. Karp, H. Kung, GPSR: greedy perimeter stateless routing for wireless networks, in: ACM MOBICOM, Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, ACM, 2000: pp. 243–254.

[15] D. Son, A. Helmy, B. Krishnamachari, 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 (2004) 233-245.

[16] P.S. Prasad, P. Agrawal, Movement Prediction in Wireless Networks Using Mobility Traces, in: Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE, 2010: pp. 1-5.

[17] H.-H. Hsu, K.-C. Tsai, Z. Cheng, T. Huang, Posture Recognition with G-Sensors on Smart Phones, in: Network-Based Information Systems (NBiS), 2012 15th International Conference On, n.d. pp. 588-591.

[18] N. Oros, J.L. Krichmar, Neuromodulation, attention and localization using a novel Android #x2122; robotic platform, in: Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE International Conference On, n.d. pp. 1-6.

[19] M. Tomlein, P. Bielik, P. Krátky, S. Mitrík, M. Barla, M. Bieliková, Advanced Pedometer for Smartphone-based Activity Tracking, in: Healthinf, 2012: pp. 401-404.

[20] C. Holland, O. Komogortsev, Eye tracking on unmodified common tablets: challenges and solutions, in: Proceedings of the Symposium on Eye Tracking Research and Applications, ACM, New York, NY, USA, 2012: pp. 277-280.

[21] W. Kiess and T. Ogilvie, “HLS - Hierarchical Location Service for Mobile Ad-hoc Networks,” 2012. [Online]. Available: http://www.cn.uni-duesseldorf.de/alumni/kiess/software/hls-ns2-patch. [Accessed: 04/10/2012].
[22] W. Kiess, “Hierarchical Location Service for Mobile Ad-Hoc Networks,” University of Mannheim, 2003.
[23] W. Kieß, H. Füssler, J. Widmer, M. Mauve, Hierarchical location service for mobile ad-hoc networks, SIGMOBILE Mob. Comput. Commun. Rev. 8 (2004) 47-58.
[24] C. Igel, M. Husken, Improving the Rprop learning algorithm, in: Proceedings of the 2nd ICSC International Symposium on Neural Computation, 2000: pp. 115-121.
[25] X. Hong, M. Gerla, G. Pei, C.-C. Chiang, 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, ACM, New York, NY, USA, 1999: pp. 53-60.
[26] Cisco Systems, 2006. Understanding Delay in Packet Voice Networks. Available at: http://www.cisco.com/en/US/tech/tk652/tk698/technologies_white_paper09186a00800a8993.shtml [Accessed January 30, 2014].