Imperial College London

Department of Electrical and Electronic Engineering

Final Year Project Interim Report 2017

Project Title: **Location-based Routing Algorithms in Mobile Ad Hoc Networks**

Student: **Sachin Leelasena**

CID: **00841878**

Course: **EEE4**

Project Supervisor: **Dr J.A. Barria**

Second Marker:

Contents

1 Project Specification

1.1 Motivation

Mobile Ad-Hoc Networks (MANETs) are increasingly used in situations where infrastructure networks are unsuitable due to damage caused by natural disasters or due to the mobility of the nodes in a network. For example, in the event of an earthquake, mobile cell towers may be damaged, reducing the ability of rescue teams to find survivors who require assistance. This can be overcome using an ad-hoc network to transmit data between different rescue teams and between survivors and rescue teams. Similarly, MANETs are used when the position or availability of nodes in a network are not fixed and location-based routing is required in order to transmit data within the network. MANETs are usually power limited due to the use of portable, battery powered devices so limiting power consumption is important.

The aim of this project is to identify the best algorithm to use in a MANET to relay local area information to people leaving a football stadium after a match. In this project, the 'best' algorithm will be determined by measuring performance metrics, specifically the packet transmission speed amongst a large number of nodes. A secondary objective function of minimising power consumption could also be evaluated if the packet transmission speed among the different algorithms tested was very similar. In this context, a local area event is something that is likely to affect the route taken by people as they make their way home. This can include underground station closures, road closures and fire alerts. In the case of these events, the slow transmission of information can lead to safety issues due to the density of the crowds in the area. The use of a MANET in this scenario could help people find the fastest route home and prevent potential injuries.

1.2 Objectives

To assess the success of this project, a number of objectives have been drawn up. The evaluation plan (section 4) will discuss how the success of each objective will be evaluated.

1. Simulate people leaving a football stadium and heading towards nearby public transport hubs through the use of a realistic mobility model.
2. Simulate a minimum of two routing algorithms in a MANET created between smartphone users in a local area.
3. Measure the speed of packet transmission in the MANET. Speed of transmission will be defined as the time difference between the first and last nodes in the network receiving a message.
4. Compare the implemented routing algorithms and identify the best one in terms of transmission speed.

1.3 Deliverables

The main deliverables for this project will be the MATLAB scripts used to simulate and and compare different algorithms along with the analysis used in different scenarios in order to determine the best algorithm to use. If a different software package is used to simulate the network, the relevant configuration files will be a deliverable. The final project report and presentation documenting the findings will also be a deliverable.

2 Background

2.1 Proactive Routing Protocols

Proactive Routing Protocols require each node to maintain routing information to every other node in the network. This routing information may be held in a number of tables which are updated periodically or if the network topology changes. Some commonly used proactive routing protocols will be described in this section.

2.1.1 Destination-sequenced distance vector (DSDV)

The destination-sequenced distance vector (DSDV) protocol produces a single path to the destination based on the distance vector shortest path routing algorithm and guarantees loop free routes. Each routing table entry is tagged with a sequence number so nodes can identify stale routes and thus avoid routing loops being formed [1]. A route with a greater sequence number will be picked first [2]. DSDV uses two types of update packet in order to reduce network overhead, known as "full dump" and "incremental" packets. The full dump packets carries all the available routing information whereas the incremental packets only carry information changed since the last full dump. Incremental packets are sent more frequently than full dump packets [3]. DSDV introduces large amounts of overhead to the network due to the periodic update messages so is unsuitable for a large network since a large amount of bandwith is used in the updating process [3].

2.1.2 Wireless routing protocol (WRP)

WRP also generates loop free routes but requires each node to maintain four routing tables: a distance table, a routing table, a link-cost table and a message retransmission list [4]. This introduces a significant memory overhead at each node in the network so does not scale well for large networks. WRP uses hello messages which are exchanged between neighbouring nodes whenever there is no recent packet transmission. This uses a large amount of power and bandwidth in order to keep all the nodes in the network active [3].

2.1.3 Cluster-head gateway switch routing (CGSR)

CGSR is a hierarchical routing protocol where nodes are grouped into clusters. Each cluster is maintained with a cluster-head which is a node that manages all the other nodes within the cluster. The cluster-head controls the transmission medium and all inter-cluster communications go through it.

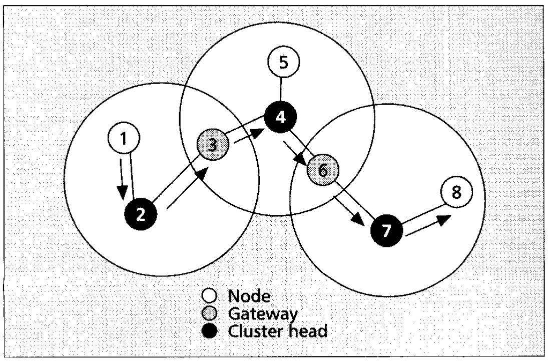

Figure 1: CGSR routing from node 1 to node 8. Figure obtained from [5]

In this routing protocol, each node only maintains routes to its cluster-head, resulting in lower routing overheads compared to flooding routing information through the entire network [3]. However, maintaining clusters incurs a significant overhead since each node needs to periodically broadcast and update its cluster member table [3].

2.1.4 Optimised link state routing (OLSR)

OLSR is a point-to-point protocol based on the link-state algorithm [6]. Each node stores topology information about the network by periodically exchanging link-state messages. OLSR minimises the size of each control message and the number of rebroadcasting nodes used per update by using multipoint replaying (MPR) strategy. During each update, each node selects a set of neighbouring nodes to retransmit its packets. This set is known as the multipoint relays of the node. Any node not in the set can read but not retransmit the packets. To select the MPRs, each node periodically broadcasts a list of one hop neighbours using hello messages [3]. From this list, each node selects a subset of one hop neighbours, which covers all of the original node's two hop neighbours. The MPR set is an arbitary subset of the neighbourhood of a node N, where every node in the two hop neighbourhood of N must have a bidirectional link toward the MPR of N. This allows routes to every destination to be available when transmission begins. A smaller MPR set leads to more optimal routing [6].

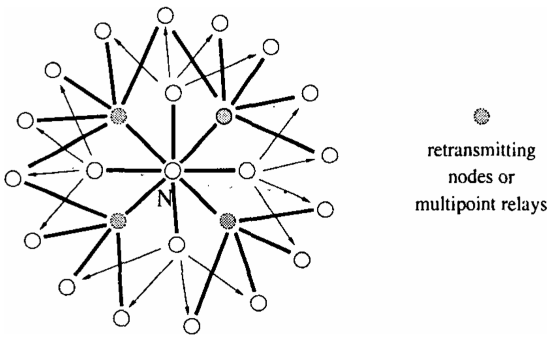

Figure 2: Multipoint Relays for a node N. Figure obtained from [6]

Proactive Routing Summary

Most flat proactive routing algorithms scale poorly due to the high bandwidth and power consumption required for the topology update process. OLSR may scale the best [3] by using multipoint relays to reduce the number of rebroadcasting nodes. This reduces the number of control packets in the network compared to flooding based protocols. CGSR can scale well due to the structure of the network which controls the amount of overhead in the network. This is done by only the cluster-head rebroadcasting control information. However, a highly dynamic network will introduce a lot of overhead to the network (cluster formation and maintenance), making CGSR unsuitable [3].

2.2 Reactive Routing Protocols

Reactive routing protocols aim to reduce overheads and network control traffic compared to proactive routing protocols by determining routes only when required.

In a source routed protocol, the packets contain the source and destination addresses with each intermediate node forwarding the packets according to the information in the header. This removes the need for the intermediate nodes to maintain routing information for each active route. The downside to this is that the protocol doesn't scale well. In a large network, the number of intermediate nodes is large so the probability of link failure can be high. Additionally, increasing the number of intermediate nodes in a route increases the the overhead of each packet as the header is larger.

In hop-by-hop or point-to-point routing, each packet contains the destination and next hop address. This requires the intermediate nodes to use their routing tables to forward the packets to the destination. An advantage of hop-by-hop routing is that the routes adapt to the dynamic nature of the network as the routing tables can be updated as nodes move. This allows more efficient routes to be used and fewer route recalculations are required. A disadvantage of hop-by-hop routing is the need for each node to store routing information for each active route and beaconing messages may be required to update routing tables based on the behaviour of neighbouring nodes [3].

2.2.1 Dynamic source routing (DSR)

In the DSR protocol, each packet carries the address of every hop in the route from source to destination. This limits the effectiveness of the protocol in large networks due to the large overhead required in each packet. However, DSR may outperform other protocols such as AODV in small to moderately sized networks [3]. Each node can store multiple routes in its route cache, allowing the source node to check for a valid route before initiating route discovery. DSR can therefore be very efficient in a network with low mobility as the routes will be valid longer. DSR does not require periodic beaconing (hello messages) so nodes can enter sleep mode to conserve power which also saves bandwidth [3].

2.2.2 Ad Hoc On-Demand Distance Vector (AODV)

AODV is based on the DSDV protocol described in section 2.1.1. The periodic updates and sequence numbering procedure from DSDV are used but the number of broadcasts are reduced by the routes being created on-demand. Route requests are forwarded to neighbours until the destination or an intermediate node with a route to the destination is reached [7]. AODV uses a similar route discovery procedure as DSR but the packets only carry the destination address instead of the route, giving a lower routing overhead than DSR. The route replies in AODV only carry the destination IP address and the sequence number instead of the address of every node in the route as in DSR. The AODV protocol is adaptable to highly dynamic networks but delays may be experienced during route construction and link failures which trigger route discovery [3]. AODV only works in a network with symmetric links, which can be hard to achieve in a mobile network, whereas DSR can be used with asymmetric links. DSR is also more resilient than AODV when it comes to link failure due to storing multiple routes in the route cache [7].

2.2.3 Light-weight mobile routing (LMR)

The LMR protocol uses flooding to discover its routes. Each node maintains multiple routes to each destination which increases the reliability by allowing nodes to select the next available route without initiating route discovery [3]. Additionally, each node only maintains routing information to their neighbours, avoiding the delays and storage requirements for maintaining complete routes. However, this introduces the possibility of invalid routes which leads to delays in determining a correct route [3].

2.2.4 Temporally ordered routing algorithm (TORA)

The TORA protocol is a highly adaptive loop-free distributed routing algorithm based on the concept of link reversal [5]. During route creation and maintenance, nodes use a ”height” metric to create a directed acyclic graph (DAG) rooted at the destination. Links are then assigned a direction (upstream or downsteam) based on the relative height of neighbouring nodes. In this protocol, nodes maintain routing information about adjacent nodes, meaning only a few control messages where the network topology has changed is required to update the routing. This makes TORA suitable for highly dynamic networks with dense nodes. TORA relies on timing as the ”height” metric is dependent on the logical time of a link failure and the protocol assumes all nodes have synchronised clocks, such as from the Global Positioning System [5].

2.2.5 Associativity-based routing (ABR)

ABR uses a query-reply technique for route discovery, but route selection in ABR is primarily based on stability. Each node maintains an associativity tick with its neighbours and links with higher associativity are selected for the route [3]. While this may not lead to the shortest route, the route tends to last longer so fewer route rediscoveries are required and more bandwidth is available for data transmission. However, periodic beaconing is required so nodes cannot sleep to reduce power consumption. Another disadvantage is the lack of multiple routes or a route cache so route discovery is required after link failure [3].

2.2.6 Signal Stability Adaptive (SSA)

SSA descends from ABR but routes are selected based on signal strength and location stability rather than an associativity tick. As with ABR, SSA may not pick the shortest route to the destination, but each route will last longer meaning fewer route rediscoveries are required. Unlike DSR and AODV, intermediate nodes in SSA cannot reply to route requests so route discovery can cause long delays. SSA also makes no attempt to repair routes where link failure occurs [3].

2.2.7 Relative distance micro-discovery ad hoc routing (RDMAR)

RDMR minimises routing overheads by calculating the distance between source and destination nodes and limiting each route request packet to a certain number of hops. This allows route discovery to be limited to a small region of the network. The same technique is used for route maintenance when links fail which conserves bandwidth and battery power [3]. RDMR does not require location aided technology such as a GPS to determine routing patterns, but the relative-distance discovery process can only occur if the two nodes have previously communicated. If no previous communication has occurred, the protocol will behave in the same manner as flooding algorithms where route discovery has a global effect [3].

2.2.8 Cluster-based routing protocol (CBRP)

In CBRP, nodes are organised in a hierarchy by grouping them into clusters. As with CGSR, each cluster has a cluster-head which manages data transmission within and between clusters. Since only cluster heads exchange routing information, the control overhead transmitted through the network is significantly lower than flooding methods. However, the hierarchical aspect of the protocol means there is an overhead associated with cluster formation and maintenance [3]. A disadvantage of CBRP is the possibility of temporary routing loops. This can occur due to nodes carrying inconsistent topology information due to long propagation delays. This protocol is therefore suitable for medium size networks with slow to moderate mobility and is likely to perform best when nodes within a cluster remain together [3].

Reactive Routing Summary

Most reactive routing protocols follow similar route discovery and maintenance procedures so will have roughly the same routing cost for a worst-case scenario. For example, RDMR has the same cost as a flooding algorithm if the source and destination nodes have not previously communicated, so is unlikely to be efficient in a highly dynamic network. This will be the case when nodes join the network, but they become more aware of their surroundings as they remain part of the network. DSR initiates a network wide flooding route discovery method when routes expire in the route cache but RDMR uses a more localised route discovery procedure to reduce overhead. Control overhead can also be reduced by picking routes based on stability as ABR (associativity tick) and SSR (signal strength) do. While this may not result in the shortest routes, the increased stability should reduce the frequency of route discovery as links are less likely to fail. This can lead to better performance than shortest path protocols such as DSR but may scale badly since each packet carries the full destination address and the probability of link failure increases with network size. CBRP reduces control overhead by splitting the network into clusters. During route discovery, cluster-heads exchange information rather than every intermediate node; this results in a significant reduction in control overhead compared to flooding algorithms. However, CBRP may have a large amount of processing overhead in a highly mobile network related to cluster formation and maintenance [3].

2.3 Hybrid Routing Protocols

Hybrid routing protocols are both proactive and reactive and attempt to combine the best features of each class of protocol. Hybrid protocols are designed to increase scalability by allowing nodes in a local area to form a backbone to reduce route discovery overheads. This can be achieved by proactively maintaining routes to nearby nodes and using route discovery to determine routes to far away nodes. Most hybrid protocols are zone-based where the network is split into different zones for each node [3]. Nodes may also be grouped into trees or clusters.

2.3.1 Zone routing protocol (ZRP)

In ZRP, each node has a routing zone which defines a range where network connectivity needs to be maintained proactively. For nodes within the routing zone, routes are immediately available. For nodes outside the routing zone, routes are determined reactively using any on-demand routing protocol. ZRP significantly reduces the amount of communication overhead compared to purely proactive protocols [3]. Delays are also reduced by allowing routes to be discovered faster. This speed increase occurs because in order to determine a route from within the routing zone to a node outside it, the routing only needs to travel to a node on the boundary of the destination node's routing zone. This boundary node proactively maintains routes to the destination node so can send a reply to the source with the required routing address in order to complete the route. If the routing zone is large, ZRP can behave as a purely proactive protocol while if the zone is too small, it behaves like a purely reactive protocol so the optimal zone size needs to be determined [3].

2.3.2 Zone-based hierarchical link state (ZHLS)

ZHLS uses a hierarchical structure made up of a node level topology and a zone level topology. The network is divided into non-overlapping zones where each node has a node and zone ID calculated using a GPS. No cluster-head is used to coordinate the data transmission so there is no processing overhead compared to the CGSR protocol. There is also a reduction in communication overheads compared to reactive routing protocols. When the source and destination nodes are in different zones, the source node broadcasts a zone-level location request to the other zones, resulting in significantly lower overhead compared to flooding used in reactive protocols. ZHLS is adaptable to dynamic topologies since only the node and zone IDs of the destination are required for routing, as long as the destination does not move to another zone. This may allow the protocol to scale well to large networks. A disadvantage of ZHLS is that each node must have a preprogrammed zone map so the protocol is not suitable for networks where the geographic boundary is dynamic [3].

2.3.3 Scalable location update routing protocol (SLURP)

SLURP is similar to ZHLS in that nodes are organised into non-overlapping zones. SLURP reduces the cost of maintaining routing information by removing global route discovery by assigning a home region for each node. The home region is determined using static mapping:

$$f(\text{NodeID}) \mapsto \text{regionID} \tag{1}$$

$$f(\text{NodeID}) = g(\text{NodeID}) \mod K \tag{2}$$

Equation 2 shows an example of a mapping function where g(NodeID) is a random number generator using the node ID as the seed and K is the total number of home regions [8]. Since each node has a constant node ID, the function will always calculate the same home region. Each node unicasts a location update message towards its home region which is then broadcasted to all nodes in the home region. To determine the location of any node, each node can unicast a location discovery packet to the home region of the node in question. The source can then send data towards the destination once the location has been discovered. When packets reach the region of the destination node, source routing using DSR gets the packet to the destination node [8]. A disadvantage of SLURP is that it also relies on a preprogrammed zone map like ZHLS.

2.3.4 Distributed spanning trees based routing protocol (DST)

DST groups nodes in a network into trees which contain root nodes that control the tree structure and internal (regular) nodes. Nodes have three states; router, merge and configure and will be in one of the states depending on the task being performed [7]. Route determination can be done by hybrid tree flooding (HTF) or distributed spanning tree shuttling (DST). In HTF, control packets are sent to neighbours and adjoining bridges in the tree whereas in DST, control packets are broadcasted from the source along the tree edges. When control packets reach leaf nodes, they are sent up the tree until it reaches a certain height (shuttling level). Once this level is reached, the control packets can be send down the tree or to adjoining bridges. The main disadvantage of DST is the reliance on the root node to configure the tree as this is a single point of failure [7].

Hybrid Routing Summary

Hybrid routing protocols can scale better than purely proactive or reactive protocols by attempting to minimise the number of rebroadcasting nodes through defining a structure to the network. The structure allows nodes to work together to optimise the route discovery process. Collaboration between nodes can help to maintain routing information for longer, reducing or even eliminating the need for flooding. Hybrid routing protocols also attempt to remove single points of failure and bottlenecks in the network. This can be done by allowing any number of nodes perform routing or data forwarding if a preferred link fails [3].

2.4 Location-based Routing Protocols

2.4.1 Distance routing effect algorithm for mobility (DREAM)

In the proactive DREAM protocol, each node knows its location coordinates using GPS. Nodes periodically exchange coordinates and store them in a location table. Exchanging location information consumes significantly less bandwidth than exchanging link state or distance vector information so it scales better than other protocols [3]. The routing overhead is also reduced by making the update message frequency proportional to mobility and the distance effect. This means static nodes don't need to send update messages.

2.4.2 Location-aided Routing (LAR)

The reactive LAR protocol attempts to reduce the routing overheads in flooding algorithms by using location information. The protocol assumes each node knows its location through GPS. The routing can then be determined by defining a boundary where the route request packets can travel to reach the destination [3]. Alternatively, the route request packets can store the GPS coordinates and only travel in the direction where the distance to the destination decreases. Both methods conserve bandwidth by limiting the control overhead transmitted in the network. In most cases, this will determine the shortest path from source to destination. A disadvantage of LAR is that each node requires a GPS and the protocol may behave similarly to DSR and AODV in highly mobile networks [3].

2.4.3 Geocast Adaptive Mesh Environment for Routing (GAMER)

Geocasting is a variation of multicasting, where the goal is to deliver packets to a group of nodes in a geographical area known as the geocast group [9]. Source routing, forwarding zones and meshes are used in the protocol. Source routing is used to route geocast packets where each packet carries the full route in its header. Forwarding zones are used to route geocast packets by flooding packets in the forwarding zone rather than the entire network, similar to DREAM. GAMER uses a mesh "to establish multiple paths between a source node and geocast regions" [9]. The GAMER protocol is able to dynamically adjust the density of the mesh; high mobility nodes create a dense mesh, low mobility nodes create a sparse mesh. The close proximity of geocast group members means it is less costly to establish redundant paths to a geocast region than to a multicast region. The mesh is created by flooding JOIN-REQUEST packets to the mobile nodes in the geocast region via a forwarding zone which dynamically changes in size which changes the density of the mesh [9]. GAMER has better transmission accuracy compared with non-adaptive, mesh-based geocast protocols [9].

2.4.4 Two-hop Routing

Static wireless sensor networks (WSN) can use 2-hop neighbour knowledge to minimise the number of forwarder nodes in a network to reduce energy consumption [10]. Each forwarder node selects a neighbour node which has the largest number of neighbours and is closest to the destination D to guarantee packet delivery. The forwarding mechanism can be improved using the average distance from two-hop neighbours to D and the probability related to the average [11].

In the TN-CMAD algorithm [11], the forwarder node computes the progress of its one-hop neighbours and the average distance from two-hop neighbours to D. The forwarder node then selects the neighbour node with the minimum two-hop distance from D as the next sender.

In the TN-CRDN algorithm [11], the forwarder node calculates a probability index which reflects the one-hop distance to D and the number of two-hop neighbours. The closest one-hop neighbour of the forwarder node will have the lowest probability p_{avg}. The one-hop neighbour with the most neighbours will have the highest probability p_n. The overall probability index of a neighbour v_x is determined as follows with the node with the highest probability index being selected as the next hop:

$$Pr(v_x) = (1 - p_{\text{avg}}(v_x))p_n(v_x) \tag{3}$$

These two-hop routing algorithms reduce the network overhead and have lower end-to-end delays, while TR-CDN also reduces the number of hops compared to GAMER [11].

2.4.5 Greedy Perimeter Stateless Routing (GPSR)

GPSR uses router position and the destination of a packet to make packet forwarding decisions [12]. GPSR is greedy as it selects the next hop as the node closest to the destination of the packet until the destination is reached. GPSR maintains routing information through the use of periodic beaconing which provides all nodes with positions of their neighbours. If no beacon is received for longer than a time-out period T, the node is assumed to have failed or left the network. An advantage of greedy forwarding is that it only relies on knowledge of immediate neighbours, which can be significantly less than the number of nodes in the entire network. However, only using the position of neighbouring nodes can require packets to temporarily move further from the destination.

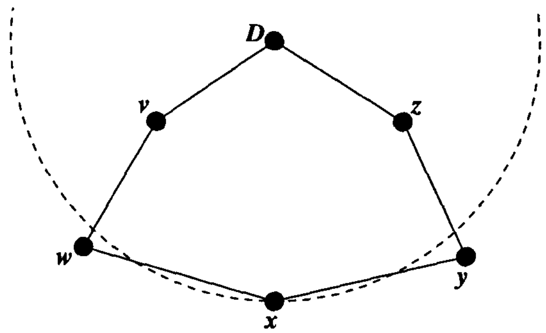

Figure 3: Greedy forwarding failure. x is a local maximum in geographic proximity to D; w and Y are farther from D. Figure obtained from [12]

In this scenario, node x sees a void where no neighbours exist and the destination D is on the other side of the void. In order to route to D, node x employs the right-hand rule to route around the void to a node closer to D than x.

2.5 Mobility Models

A number of different mobility models can be used to simulate the motion of nodes in a MANET. Mobility models typically describe the velocity and direction of nodes amongst other parameters. This section explores some mobility models that may be suitable to describe the motion of people.

2.5.1 Random Walk

In the random walk model, nodes move from their current position to their next position with a fixed velocity and random direction. Subsequent movements are independent of each other i.e. the model is memoryless [7]. At every time interval, the node takes one of eight possible directions, each of which have equal probability. While this is useful for modelling irregular movement, the scenario for this project is concerned with more regular movement of people towards points of interest, so a random walk mobility model is not suitable for a very accurate simulation.

2.5.2 Random Waypoint Model

The random waypoint model (RWP) is one of the most popular synthetic mobility models. Each node in the model moves towards the next waypoint with a constant velocity v, where $v \sim U(0, V_{max})$. The model also allows for 'thinking times' when nodes reach a waypoint. In the RWP process, consecutive legs are not independent as they share a common waypoint [13]. Additionally, the RWP process is 'time reversible' as a path along waypoints P_0, P_1, ..., P_n is equally likely as the time reversed path P_n, P_{n-1}, ..., P_0 [13]. Once a node reaches a waypoint, it travels in a new direction with probability P or remains in the current direction with probability 1 - P [7]. While this mobility model may be more appropriate than the random walk model for most cases, it is unlikely to be useful for this project as nodes need to move towards fixed points to simulate people walking towards points of interest.

2.5.3 Shortest Path Map Based Movement (SPMBM)

The shortest path map based movement model is a more advanced version of the map based movement (MBM) model. The MBM model places nodes randomly in the map area and moves them along path segments until the end of the road where they turn back, or until an intersection is reached. At an intersection, nodes randomly select a new direction but do not move backwards [14]. SPMBM also starts with nodes in random locations, but selects a certain destination for all nodes and finds the shortest path to the destination using Dijkstra's shortest path algorithm [14]. The map data can contain Points of Interest (POIs) which have a higher probability of being chosen as a node destination than regular places on the map. This mobility model is promising for use in this project as it will give a good approximation for the movement of people towards POIs such as underground stations and bus stops around a stadium. The ONE simulator supports the use of maps which can be used by the SPMBM model to simulate the movement of people.

2.6 Simulation Packages

A range of simulation packages have capabilities to model MANETs, some of which would be more suitable to use than MATLAB for this project.

2.6.1 Opportunistic Network Environment (ONE) simulator

The ONE simulator is designed to simulate delay tolerant networks (DTNs). The software package is written in Java and includes movement modeling, routing simulation, visualization and reporting [14]. The simulator has a number of movement models built in, including the Shortest Path Map Based Movement (SPMBM) model and also allows for mobility traces to be imported from external sources. The movement of nodes towards points of interest (POIs) is supported. The simulator has a GUI where node movement and routing behaviour can be observed, along with a reporting module which allows data about the simulation to be measured. This simulator would be useful for this project if it could be modified to simulate conventional MANET router types.

2.6.2 ns-3

ns-3 is a network simulator for internet systems written in C++ with optional Python bindings. The simulation package provides models for packet data networks and a simulation engine. A lot of documentation and tutorials for ns-3 appear to be available including resources for MANET simulation. However, installation issues have prevented testing of the software so far.

2.6.3 OMNeT++

OMNeT++ is a C++ simulation library and framework for building network simulators. A number of simulation frameworks already exist which cover MANET protocols and mobility, such as the INET framework. Open source frameworks specifically for MANETs also exist so this simulation package looks promising at this stage. However, the lack of familiarity with the package is a significant disadvantage compared to MATLAB.

2.7 Wireless Communication Technologies

The nodes in the MANET will need to transmit packets between themselves using existing wireless communication technologies as there is likely no benefit to designing a new protocol for this use case. Designing a new technology is also outside the scope of this project. The main aspects of the technologies to be used are the transmission range, transmission speed and power consumption to make the simulation as accurate as possible. This section will explore the wireless communication technologies available in current smartphones as these devices are the most suitable to build a MANET amongst a crowd of people leaving a stadium. Near Field Communication (NFC) was not considered due to the need for close proximity of devices to transfer information ($<$ 10 cm [15]).

2.7.1 Bluetooth

Bluetooth Low Energy (BLE or Bluetooth 4.x) is a widely used variant of the Bluetooth standard with a range of up to 50m [16]. The technology was developed by the Bluetooth Special Interest Group as a low-power solution so appears suitable for this project. Modern mobile phones have BLE connectivity which allows for the creation of a concentrated network with a large number of nodes. Smartphones typically contain Class 2 Bluetooth devices which have a maximum power consumption of 2.5 mW and a range of 10m [15]. Bluetooth 4.0 enables high-speed operation up to 24 Mbps [15]. While Bluetooth devices generally require pairing before data can be transferred, it is possible to transfer data through the advertising channels only, bypassing the pairing requirement [17]. Therefore, the simulation in this project will make the assumption that pairing is not required to transmit data in the MANET.

2.7.2 Wi-Fi Direct

Wi-Fi Direct builds on the IEEE 802.11 infrastructure mode and allows devices to negotiate which will perform the access point (AP) like functionalities [18]. This lets Wi-Fi Direct inherit Quality of Service (QoS), power saving and security mechanisms from the Wi-Fi infrastructure mode. Wi-Fi Direct also has the same range as conventional Wi-Fi connections of up to 200m [19], far greater than the 10m range for Class 2 Bluetooth devices. Wi-Fi Direct is supported by fewer mobile devices than Bluetooth as iOS devices have a proprietary version instead which combines Wi-Fi and Bluetooth connectivity [20]. Using Wi-Fi Direct could produce better performance of the MANET in terms of transmission speed as the network is more sparse, but the simulation would be less realistic; this suggests the use of Bluetooth is better.

2.8 Background Summary

From the background reading, the most appropriate wireless technology to simulate for this project is Bluetooth Low Energy due to the high adoption rate. The routing protocols to simulate have been identified as Distance routing effect algorithm for mobility (DREAM), Light-weight mobile routing (LMR) and Location-aided Routing (LAR). DREAM was identified due to the low or variable routing overhead which can allow message transmission to be fast. The GPS aspect for both DREAM and LAR is easy to achieve as smartphones usually have GPS and other location functionality. LMR was identified to allow the performance of a flooding based protocol to be tested. Simulating LAR will allow a performance comparison between reactive and proactive location-based routing algorithms.

At this stage, the best simulation package to use is MATLAB due to the familiarity with the package. Other packages will be tested further but not for too long to ensure the main part of the project gets underway as soon as possible. The MATLAB scripts accompanying the MSc thesis by L. Feng [7] have been analysed and the simulation seems relatively straightforward. Although the mobility model and algorithms for this project are different, a similar simulation framework can be used.

The most suitable mobility model identified from the background is the shortest path map based movement (SPMBM) model as it can model the movement of people leaving a stadium fairly accurately.

3 Implementation Plan

The implementation plan for the remainder of the project can be broken into a few subsections which are presented in this section. Table 1 displays a preliminary timeline for the remainder of the project.

3.1 Implementing mobility model

From the literature review, a good mobility model for people walking towards points of interest is the shortest path map based movement model. This model selects a certain destination point for each node in the system and uses Dijkstra's shortest path algorithm to find the shortest path to the destination. While the Opportunistic Network Environment (ONE) simulator already includes this mobility model, MATLAB does not so the algorithm will need to be implemented.

There may be an issue related to creating accurate map based movement in MATLAB so a fallback would be to create a small number of predetermined paths for the nodes. Each path would have a probability of being picked by nodes moving towards a point of interest. This should result in a good approximation for the mobility of people leaving a stadium after a match since most people have predetermined routes due to the different transport options available near the stadium. If sufficient data can be obtained, the probability of each route being picked can be made more realistic, although a basic estimate based on capacity and transport links will likely be enough for this simulation.

In either case, node speed will be an important consideration to create an accurate simulation. The range of speeds that nodes can take will be fixed and a random value will be generated for each node. Preferred walking speed is approximately 1.42 m/s [21] so the speeds can be distributed around this mean value. The speed of movement can be made to change as the node progresses along the route, but not too often in order to retain realism.

3.2 Implementing routing algorithms

Once the mobility model has been implemented, the routing can be built on top. The different algorithms to implement are DREAM, LMR and LAR as described in section 2.8.

The simulation will be sequential where each node moves before transmitting or receiving packets. Each node will keep track of the number of packets sent and the number of packets received in order to calculate the overall packet delivery rate. The simulation will have a fixed energy cost for packet transmission so the number of packets transmitted will be enough to calculate the energy consumption. Time permitting, the simulation could use more accurate ways of estimating the power consumption.

Once a local area event has been noticed, the closest nodes to it can transmit the location of the nearest alternative point of interest to the surrounding nodes. This will then lead to the nodes changing direction and moving towards their new destination. This would enable the simulation to not only measure the speed of transmission and power consumption, but to also see the effect of the MANET on the movement of the nodes.

Packet transmission will be designed to simulate a Bluetooth Low Energy system as the wide range of devices supporting the standard will allow for a network with a high density of nodes. The high density of nodes should reduce the link failure rate within the MANET, resulting in lower power consumption. Transmission via Bluetooth will be simulated using the range constraint of BLE along with its power consumption as detailed in section 2.7.

3.3 Comparison of routing algorithms

Once the routing algorithms have been implemented, their absolute and relative performance can be measured in order to find the most optimal protocol for the situation.

The primary objective will be to maximise the speed of transmission within the network. This will be measured by the time taken for all relevant nodes in the simulation to receive a message regarding the local area event being simulated e.g. station closure. To measure this, a timestamp for the first packet transmission will be stored. Each node will also have a status 'flag' which indicates whether a packet has been received. Once all nodes have received the message, another timestamp will be stored, allowing the total time to be calculated. Alternatively, this could be measured using the number of steps in the simulation as each step will be a certain period of time.

A secondary objective could be to minimise power consumption as each node is simulating a mobile phone with a limited battery capacity. This will only be required if the speed of the different algorithms tested are similar or if one is faster but has a significantly higher power consumption. This will be measured by tracking the number of packets transmitted by each node and using the assumption that each packet transmission has the same energy cost. The power consumption of the network can be measured at the same time as measuring the transmission speed, but may not be used if there are significant differences in the transmission speeds of the different protocols being tested.

3.4 Preliminary Timeline

Table 1: Preliminary timetable for project completion. Timings and tasks are subject to change.

Due Date	**Task**
06/02/17	Finalise the most suitable software package for simulation. Most suitable will be judged by ease of implementing the mobility model along with simulating the routing protocols.
	If no specialised network simulator package is determined to be suitable, MATLAB will be used as the testing to this point shows MATLAB as being the most suitable due to usability concerns.
27/02/17	Implement the mobility model and test to verify accuracy of implementation. This will also include setting some of the simulation parameters e.g. number of nodes, simulation area, node minimum and maximum velocity and simulation time.
	If the SPMBM model cannot be implemented, a model using predetermined routes will be the fallback. Each route will be determined by a shortest path algorithm but will lack the dynamic nature so will require groups of nodes to have similar starting positions.
20/03/17	Implement the first routing algorithm and test to ensure transmission time throughout the network along with power consumption can be measured. Further simulation parameters will need to be determined, including radio range and path loss parameters. Radio range will be 10m to simulate Class 2 BLE radios.
	This stage is crucial to the project so has the fallback is to use simpler algorithms for the simulation. A fallback for measuring power consumption is to measure packet delivery rate instead by tracking the total number of packets transmitted and received in the network.
22/05/17	Implement the remaining routing algorithms and collect timing and power consumption data for all the algorithms. The results need to be analysed in order to find the most appropriate algorithm for the use case. The data analysis will use average data sets from multiple simulations (minimum three) to avoid the use of anomalous data.
	A fallback at this stage would be to again use algorithms that are simpler to implement and test, or to reduce the number of algorithms to implement as a minimum of two are required.
12/06/17	Complete final project report detailing the simulations used, results obtained and conclusions drawn. MATLAB (or other simulation package) scripts should be finalised having been checked for errors to ensure the accuracy of the results.
	This stage has no fallback as the final project report is the key deliverable of the project along with the simulation scripts used.

4 Evaluation Plan

The success of the project will depend on whether the objectives outlined in section 1.2 have been met.

4.1 Mobility Model Implementation

The first objective is concerned with the implementation of a mobility model to simulate the movement of people leaving a football stadium and making their way to public transport hubs. The background research identified the Shortest Path Map Based Movement (SPMBM) model as a good estimate for the mobility of people in this scenario. For this objective to be fulfilled completely, the SPMBM model will have to be successfully implemented and simulated. This can be tested by monitoring the movement of a few nodes (1% to 5% of nodes in the network) and comparing with real world movement of people. The comparison will mainly look for abnormalities such as nodes moving too fast for a prolonged period of time or stuck in strange movement patterns such as loops.

If the SPMBM model implementation is not possible, the fallback of predetermined routes for groups of nodes in the network will partially fulfil this objective as a relatively accurate mobility model has been implemented. However, the predetermined paths must be realistic e.g. nodes cannot walk along inaccessible or closed paths.

4.2 Routing Algorithm Simulation

As a minimum of two routing algorithms need to be implemented for this project to be successful, this objective can be tested by counting the number of routing algorithms that have been implemented successfully. If only certain aspects of an algorithm have been simulated, the objective will only be partially complete, although it could be argued that a slightly different algorithm has been implemented.

This objective does not strictly depend on the implementation of the mobility model, but the two are linked as an accurate mobility model can lead to a more accurate implementation of the routing protocols.

4.3 Measuring Network Performance

The main performance metric for the simulation will be the speed at which packets propagate throughout the network. For this objective, packet or network transmission speed will be the time difference between the first and last nodes receiving messages about a local area incident. Storing timestamps for each transmission is trivial, but a system will be required to track which nodes in the simulation have received the message.

If the transmission speed for different algorithms is similar ($\pm 5\%$), the power consumption of nodes will need to be measured in order for this objective to be fully met.

The completeness of this objective can be tested by analysing the output data from the simulation and checking for timing and power consumption data if required.

4.4 Algorithm Comparison

The objective of comparing the performance of different routing algorithms is highly dependent on the number of algorithms implemented as a minimum of two are required for a comparison. Additionally, there must be some quantifiable performance measures for each of the algorithms implemented, so objective 3 is also a dependency.

This objective will be achieved if there is a conclusion drawn from the project which indicates the best algorithm with reasoning for the use case.

5 References

[1] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," *SIGCOMM Comput.Commun.Rev.*, vol. 24, pp. 234–244, oct 1994.

[2] A. Sachdev, K. Mehta, and L. Malik, "Design of protocol for cluster based routing in vanet using fire fly algorithm," in *Engineering and Technology (ICETECH), 2016 IEEE International Conference on*, pp. 490–495, IEEE, 22-27 May 2016.

[3] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks," *Ad Hoc Networks*, vol. 2, pp. 1–22, 1 2004.

[4] S. Murthy and J. J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," *Mob.Netw.Appl.*, vol. 1, pp. 183–197, oct 1996.

[5] E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE personal communications*, vol. 6, no. 2, pp. 46–55, 1999.

[6] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*, pp. 62–68, IEEE, 2001.

[7] L. Feng, "Break link prediction of location-based routing in the vanet," 2011.

[8] S.-C. M. Woo and S. Singh, "Scalable routing protocol for ad hoc networks," *Wireless Networks*, vol. 7, no. 5, pp. 513–529, 2001.

[9] T. Camp and Y. Liu, "An adaptive mesh-based protocol for geocast routing," *Journal of Parallel and Distributed computing*, vol. 63, no. 2, pp. 196–213, 2003.

[10] H.-S. K. Young-Chul Shim, "An efficient geocast algorithm using 2-hop neighbour knowledge in sensor networks," *Int. J Latest Trends Computing*, vol. 2, no. 4, pp. 521–526, 2011.

[11] P. Phoummavong, K. Utsu, C. O. Chow, and H. Ishii, "Location-aided route discovery mechanism based on two-hop neighbor information for ad hoc network," *The Journal of Supercomputing*, vol. 72, no. 3, pp. 1201–1214, 2016.

[12] B. Karp and H.-T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 243–254, ACM, 2000.

[13] E. Hyytiä and J. Virtamo, "Random waypoint mobility model in cellular networks," *Wireless Networks*, vol. 13, no. 2, pp. 177–188, 2007.

[14] A. Keranen, "Opportunistic network environment simulator," *Special Assignment report, Helsinki University of Technology, Department of Communications and Networking*, 2008.

[15] A. Al-Akkad, *Working Around Disruptions of Network Infrastructures: Mobile Ad-Hoc Systems for Resilient Communication in Disasters.* Springer, 2016.

[16] M. Siekkinen, M. Hiienkari, J. K. Nurminen, and J. Nieminen, "How low energy is bluetooth low energy? comparative measurements with zigbee/802.15. 4," in *Wireless Communications and Networking Conference Workshops (WCNCW), 2012 IEEE*, pp. 232–237, IEEE, 2012.

[17] G. Corbellini, S. Schmid, and S. Mangold, "Two-way communication protocol using bluetooth low energy advertisement frames," in *Proceedings of the 1st International Workshop on Experiences with the Design and Implementation of Smart Objects*, (New York, NY, USA), pp. 19–24, ACM, 2015.

[18] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, "Device-to-device communications with wi-fi direct: overview and experimentation," *IEEE wireless communications*, vol. 20, no. 3, pp. 96–104, 2013.

[19] W.-F. Alliance, "How far does a wi-fi direct connection travel?." Available From: `http://www.wi-fi.org/knowledge-center/faq/how-far-does-a-wi-fi-direct-connection-travel`.

[20] Apple, "Multipeerconnectivity." Available From: `https://developer.apple.com/reference/multipeerconnectivity`.

[21] R. C. Browning, E. A. Baker, J. A. Herron, and R. Kram, "Effects of obesity and sex on the energetic cost and preferred speed of walking," *Journal of applied physiology*, vol. 100, no. 2, pp. 390–398, 2006. http://jap.physiology.org/content/100/2/390.full.pdf.