A Survey of Geographical Routing in Wireless Ad-Hoc Networks

Fraser Cadger, Member, IEEE, Kevin Curran, Senior Member, IEEE, Jose Santos and Sandra Moffett

Abstract—Geographic routing offers a radical departure from previous topology-dependent routing paradigms through its use of physical location in the routing process. Geographic routing protocols eliminate dependence on topology storage and the associated costs, which also makes them more suitable to handling dynamic behavior frequently found in wireless ad-hoc networks. Geographic routing protocols have been designed for a variety of applications ranging from mobility prediction and management through to anonymous routing and from energy efficiency to QoS. Geographic routing is also part of the larger area of context-awareness due to its usage of location data to make routing decisions and thus represents an important step in the journey towards ubiquitous computing. The focus of this paper, within the area of geographic routing is on wireless ad-hoc networks and how location information can benefit routing. This paper aims to provide both a comprehensive and methodical survey of existing literature in the area of geographic routing from its inception as well as acting as an introduction to the subject.

Index Terms—Geographic routing, geometric routing, location-aware routing, wireless networks.

I. INTRODUCTION

Wireless ad-hoc networks (which shall be referred to in this paper as ad-hoc networks) are a field of networking in which networks are formed when required and typically for short durations. Ad-hoc networks are typically decentralised and do not feature dedicated devices with defined roles such as routers or switches. Instead all participating nodes act as both routers and end-users. As devices are limited by their radio range ad-hoc networks typically employ a strategy known as multi-hopping in which a source node will send a message to the destination by passing it to a series of intermediate node. This enables geographically disparate nodes to communicate wirelessly. Multi-hopping is typical of the distributed architecture of ad-hoc networks, and one of its biggest advantages. As ad-hoc networks use multi-hopping and do not rely on infrastructure they can be deployed anywhere two or more devices that share a suitable communications medium (WiFi, Bluetooth, UWB, etc.) are present. The field of ad-hoc networking itself contains several subfields such as Mobile Ad-Hoc Networks (MANETs) where all nodes are assumed to be mobile, Wireless Mesh Networks (WMNs) a combination of ad-hoc and infrastructure network, Wireless Sensor Networks (WSNs) ad-hoc networks made up of small sensor devices, and Vehicular Ad-Hoc Networks (VANETs).

Although ad-hoc networks have the potential for use in a wide range of application scenarios as diverse as battlefield communications and smart home environments, they also have some drawbacks. In addition to the general challenges of wireless communications such as interference, path loss, and fading that are also present in infrastructure wireless networks, the unique characteristics of ad-hoc networks lead to some unique challenges. While a lack of centralisation can be seen as an advantage, it can also be a disadvantage as there is no means of ensuring all devices are operating using the same standards (especially if they are all under the control of different entities). Similarly, ad-hoc networks are typically more dynamic, being formed to fulfil a particular goal and terminated when that goal has been achieved. In addition, most ad-hoc networks allow nodes to join and leave the network at will, so that the topology is constantly changing. Depending on the application, some or even all nodes may be battery powered which presents the possibility of nodes ‘dying’ during operation.

This means that conventional wired and wireless network protocols are not suitable for use in ad-hoc networks. This has led to both the adaptation of conventional routing protocols and (more commonly) the design of new ones. Generally ad-hoc routing protocols fall into one of two categories; proactive or reactive. Proactive protocols store and maintain topology information through a series of regular update (hello) messages sent between network nodes. Reactive protocols do not regularly share network information and instead send out route request messages to other nodes when they need to reach a particular destination (although they will typically store routes found during this process for later use). While proactive protocols guarantee that where a network is connected every node will have a route to a particular destination in advance, they also require the storage and transmission of frequent update messages which can cause problems in the wireless medium. On the other hand, reactive protocols do not require continual sharing of topology information, but cannot always guarantee a route will be available when required and requires the transmission of potentially expensive request messages each time a route cannot be found.

An emerging field of research in the area of wireless ad hoc and mesh networks is geographic routing in which packets are
routed to the destination based not on identity or logical address but the geographic position of the destination [1]. This is in contrast to both proactive and reactive routing protocols which are together known as topology routing in which all nodes are required to store details about the entire network or in the case of zonal routing, only certain zones [2]. Both proactive and reactive routing protocols suffer from the same disadvantages of being high maintenance (in terms of memory and communications) and relying on static end-to-end routes in a dynamic environment. This means that in addition to the potentially high communication and resource cost (whether from route updates or route requests) there is also the potential of such information being out of date when it is used.

These disadvantages are particularly relevant to networks where high mobility is possible as is the case with many ad hoc networks and some WMNs. While the potentially high resource costs are a significant drawback for WSNs, Geographic routing protocols help avoid these disadvantages by (in most cases) eliminating the need for nodes to share and store topology information (nodes only store information about neighbours directly accessible via radio) and by reducing the reliance on topology information. This decreases the costs associated with sharing information and helps reduce the possibility of out of date information being used in a routing decision. Another important feature of standard geographic routing is the lack of end-to-end links. Instead of attempting to construct and then follow a single static link from source to destination routing decisions are made on a hop-by-hop basis. This means that when a node receives a packet it will inspect its neighbour table and select the most appropriate neighbour as the next hop (based on some geographic criteria). Therefore, in addition to eliminating the need for expensive topology maintenance, geographic routing also avoids the use of similarly expensive route request messages. This also means that nodes do not have to worry about path or link breakage as the path in the conventional sense of the word does not exist, and packets sent from the same source to the same destination may always take different routes depending on the network state. By reducing reliance on topology and link information nodes are better suited to handling the dynamic conditions that are often present in all types of ad-hoc network.

Although geographic routing protocols do not store network topology, in order to function they must still be aware of where other nodes are physically located in the network. In addition with the information they share with one-hop neighbours nodes must also make sure that non-local nodes have a means of discovering their position should they wish to send a message to them. Most geographic routing protocols make use of some form of location service such as the Grid Location Service proposed by [3] which allows them to determine the location of their target in a distributed manner. Typically location services will consist of nodes designated as location servers (these may be mesh nodes in WMNs or sinks in WSNs) which are responsible for receiving and storing the positions of certain nodes (typically those within a fixed geographical region). When a source node wishes to find the location of a destination it will query these location servers. Location services allow routing to the destination using either a conventional forwarding mechanism such as flooding or a geographic one. Where flooding is used it is usually only partial flooding, restricted to a geographic area such as that used by DREAM [4], or a strategy more specific to geographic routing such as geocasting.

Greedy forwarding and face routing are two of the earliest geographic routing strategies and have together formed the basis of numerous subsequent approaches. Of the two approaches, greedy is the earliest and most basic; tracing its routes back to an approach known as Cartesian routing which was introduced for routing in large-scale internets [5]. Greedy forwarding is a conceptually simple form of geographic routing in which packets are forwarded to the neighbour located closest to the destination at each hop. Greedy forwarding is both simple to understand and implement as well as being efficient (having a worst-case complexity of O(d^3)) [6]. However, greedy forwarding does have one significant drawback; when a node is unable to find a neighbour closer to the destination than itself it must drop the packet [6]. This is done to prevent the existence of loops where the packet travels (physically) backwards; however it can have the adverse effect of leading to packets being dropped where a route to the source is possible.

Face routing is derived from Compass Routing II where faces on a planar graph are traversed using a technique known as the 'right hand rule' (sometimes left hand rule instead) in which the algorithm keeps track of all the times it crosses the line connecting the source to destination [7]. Once an entire face has been covered, the algorithm moves onto one of the intersections nearest the destination and explores that with the algorithm continuing to do so until it eventually reaches the destination [7]. The application of the Compass II algorithm to Unit Disk Graphs (UDG) as well as an algorithm for planarising UDGs was first presented in [8] and can be considered the first face routing algorithm. The UDG in which two nodes are connected if their disks overlap is a common abstraction for ad-hoc networks. Generally, the UDG is used as a base-model for a planarization algorithm; typically based on the Gabriel Graph. The Gabriel Graph is a planar sub-graph in which two points are connected if they are endpoints of a circle’s diameter [9].

The main advantage of face routing is that it guarantees delivery [6], however disadvantages include its possible inefficiency; Kuhn, et al. (2008) assert that face routing has the same efficiency as basic flooding) and its reliance on planar sub-graphs which calls into question its effectiveness in non-planar environments (i.e. networks where nodes aren’t situated on a flat surface) [6]. Variants of basic face routing include Path Vector Face Routing (PVR) that allows for a limited amount of local face information to be stored by nodes [10], Adaptive Face Routing (AFR) is an augmentation of basic face routing designed to tie the cost of finding a destination to a function of the optimal route [11]. Bounded Face Routing (BFR) in which faces searched are limited to an elliptical shape containing the optimal path [11]. If the
algorithm hits the ellipse then it will continue examining the face, but in the opposite direction until it hits the ellipse again and then moves onto the next face [11]. In turn, AFR and BFR can be applied together to allow AFR to restrict the area BFR searches [6].

The work carried out on BFR and AFR (as well as other modified face routing algorithms) has led to the development of GOAFR a hybrid algorithm that aims to use greedy routing as much as possible while remaining worst-case asymptotically optimal when performing face routing [12]. The work of [8] also proposes a hybrid greedy-face algorithm known as GFG that alternates between greedy and face mode. Other greedy-face hybrids include Greedy Perimeter Stateless Routing (GPSR) [13], which has been widely cited but appears to be a duplication of [8]. In addition to greedy and face routing, various other geographic protocols have been proposed that use different strategies to route packets using location information; for instance One of the earliest routing protocols to use location, Location-Aided Routing (LAR) developed by [14] uses location information to determine a ‘request zone’ that the destination is believed to located in and then forwards packets to nodes in this region. Although LAR itself can now be considered obsolete it has been the subject of modifications and enhancements such as in [15] in which a baseline (a line between the source and destination nodes) is used when drawing up the request zone leading to improved results. Clearly neither greedy forwarding nor face routing are involved in the design of these two protocols.

In addition to protocols such as LAR, several ‘niche’ protocols have been designed with specific goals in mind (as opposed to generally improving routing performance); for instance [16] presents a geographic routing protocol that features Quality of Service (QoS) predictions based on device mobility while [19] have developed an energy-efficient geographic routing protocol that guarantees delivery. These protocols demonstrate the variety of ways in which location information can be used in addition to making forwarding decisions based purely on physical location. The aforementioned QoS predictions of [16] calculates a parameter known as motion stability based on a node’s mobility pattern (i.e. whether it is moving at high speed, with varying direction, etc.) and in turn uses it to determine the level of jitter (variation in delay) routing through this node will incur (clearly nodes with greater motion stability are preferred over those with unstable motion patterns). Similarly, authors such as [17] and [18] explore ways of providing anonymous geographic routing through the use of measures such as temporary IDs and authentication to strike a balance between the need for privacy and the efficiency of geographic routing. Through their use of location information to perform routing it could be argued that geographic routing protocols represent an aspect of context-awareness; with the context in their case being location. Clearly there are various advantages to the use of geographic data in ad-hoc and wireless mesh networks, and this paper aims to explore these uses from a routing perspective.

The overall aim of this paper is to provide a comprehensive survey of existing literature in the area of geographic routing in wireless ad-hoc networks for the purpose of providing an introduction and overview to the field. A unique point of this paper is the approach that has been used for surveying and reviewing literature. The approach taken is that of ‘challenges and solutions’; the vast majority of the paper is taken up with a discussion of the various challenges facing geographic routing (some of which are general to ad-hoc routing) and how existing protocols are affected by these issues and in turn how some of those protocols address such issues. While the emphasis of this paper is not on classification; for a comprehensive classification of geographic routing protocols see the taxonomy devised by [20], the nature of the approach means that some classification will be evident when protocols that address similar issues are grouped together. With this in mind, it is important to acknowledge that although there will be some overlap between sections, most protocols are only discussed in detail in one section; for instance, although mobility prediction is used by protocols specialising in QoS and mobility management, discussion on mobility prediction will be confined to the relevant sections depending on the emphasis of the protocol. Similarly, as the emphasis of this paper is on ad-hoc networks protocols that are intended for use in VANETs are not included as these represent a separate series of constraints and challenges, and so a more specialised survey is required.

The paper is organised as follows; Section II provides an overview of geographical routing protocol design issues. Section III looks at geographical security routing issues while section IV examines location & mobility. Section V investigates the issues of power consumption. Section VI examines quality of service issues, section VII summarises the previous sections. Section VIII provides several proposals and suggestions for future research and section IX concludes.

II. GEOGRAPHICAL ROUTING PROTOCOL DESIGN ISSUES

Two popular categories of geographic routing protocols - greedy forwarding and face routing were presented earlier. It was mentioned that greedy forwarding was both the most simple and efficient of the two with a worst-case complexity of \(O(d^2)\) with \(d\) representing the distance between the source and destination [6]. In contrast, face routing requires \(O(n)\) messages to find a route from source to destination where \(n\) is the number of nodes in the network. This is a performance bound similar to that of flooding [6]. However it was also pointed out that where face routing guarantees delivery (where delivery is possible), greedy routing cannot make such a guarantee. Here in this section, we examine such issues and we begin by looking at greedy forwarding.

A. Greedy Forwarding

As mentioned earlier, greedy forwarding works by continually forwarding a packet closer to a destination. When the source wishes to send a packet to a specific destination it will first consult its neighbour table to determine which of its neighbours is closest to the destination and select that
neighbour as the next hop (assuming the destination is not one of its neighbours in which case the packet will be forwarded straight to the destination). Upon receiving the packet the next node will then perform the same operation; determine which of its neighbours is closest to the destination and select it as the next hop. This process continues until either the destination is reached or the aforementioned local maximum problem occurs. Figure 1 provides an illustration of a simple greedy forwarding scenario.

Figure 1: Greedy Forwarding

In this diagram the thick black circles represent nodes, the red lines represent the route of the packet and the blue circles represent the transmission range of the present node. It should be noted that despite the use of circles to represent transmission ranges of nodes participating in the routing process that this graph is not a UDG Our graph only shows the transmission ranges of nodes directly involved in the routing process whereas a UDG will show the transmission ranges of all nodes in the network.

The originating node is the one marked with S (for source) which wishes to send a packet to the destination (marked D) but is unable to do so directly. Following the principles of greedy routing S will determine which of its neighbours (the nodes within its transmission range) are closer to the destination than itself (if any) and send the packet to that node. The sending of the packet to the next node is indicated with the red line. Upon receiving this packet the next node will carry out the same process as the source and check its neighbour table for a suitable successor node. Eventually the penultimate hop is reached, where the receiving node is able to consult its neighbour table and see that the destination is one of its neighbours and then forward the packet directly to the destination.

The scenario depicted in Figure 1 is a somewhat idealized scenario in which every node participating in the routing process always has a neighbour closer to the destination than itself. In real networks this might not always be possible due to dynamics such as node movement, failure, or even aberrations in transmission. Therefore, it is entirely possible that a node will be unable to find a neighbour closer to the destination than itself. This means that a packet could be dropped even when a route from the source to the destination exists, but would require sending the packet ‘backwards’. This situation was earlier referred to as the local maximum and is a serious problem in greedy routing protocols. Nodes in this situation are referred to as concave nodes and when a concave node is unable to forward a packet it will by default drop it [21]. Although this behaviour may seem odd, the logic behind it is that if a packet cannot travel physically backwards then loops can be easily avoided. A related phenomenon is that of voids. Voids are areas of a network that are uncovered or unreachable [21]. This naturally means that greedy routing will fail when a void is encountered as it would be impossible for the current node to have any nodes nearer the destination than itself [22]. As concavity and voids present a serious risk to the performance of greedy forwarding, research has frequently focused on designing techniques for recovering from or avoiding concave nodes and voids.

Geographic Landmark Routing (GLR) proposes a solution to this problem of voids through the discovery of paths that bypass voids [22]. GLR attempts to do so through the storage of landmark nodes – nodes at which the recovery scheme is terminated and greedy routing resumed [22]. In [22] it is claimed that setting landmark nodes as intermediate targets allows GLR to avoid two problems common to face-routing based schemes; blind detouring and triangular routing. Blind detouring is described as the problem whereby the face routing algorithm inadvertently choses a longer than necessary traversal process due to its decision to ‘blindly’ traverse the face. Similarly, the triangular routing problem arises in instances where packets are continually rerouted to another node from a dead end to a better placed node, when it would be more efficient to route these packets to the other node without needing to send them to the dead end node.

GLR’s functionality can be split into two main areas; landmark discovery and landmark routing. Landmark discovery is initiated when a node wishes to send a packet to a particular destination and there is no landmark nodes stored for that node. Landmark discovery begins with the source node sending a forward landmark discovery (FLD) packet which may also contain data in the direction of the source. This packet is used to record the hop count from source to destination as well as the landmark nodes encountered at each void (if any are encountered). Upon receipt of the FLD packet, the destination copies the hop count and landmark nodes into a backward landmark discovery (BLD) packet and sends this to the source node, while the packet is being transmitted the hop count and number of landmark nodes are recorded and stored separately from the FLD information. The source node will then select either the FLD or BLD path for later use in subsequent transmissions based on the number of hops each path has. If the FLD has the least hops and no landmark nodes then the landmark nodes field for the destination is set to null, while if landmark nodes are present they are stored in that field. If the BLD packet is selected and landmark nodes are present then they are also stored, but interestingly if there are no landmark nodes then one node from the BLD hop count is selected as a ‘virtual landmark’ node.

The authors reason that this is necessary because it is impossible to retrace the destination to source route using only the destination location (even though it would seem the BLD packet records hops in between the destination and source). Although the authors claim that without the virtual node a longer path than the FLD path would be followed they do not
present any reasoning for this, and it is therefore unclear to the reader as to why the virtual landmark is necessary.

It is not stated explicitly by the authors how the initial routing of the FLD is done. Although a later reference is made to the beaconing functions of the GPSR protocol, it is not explicitly stated that GLR uses beacons to exchange neighbour information with other nodes. If this is not the case, then GLR would either need an alternate means of obtaining neighbour node information or would have to resort to flooding if it did not have such information. If the latter is true then landmark nodes seem unnecessary and the protocol would essentially be a flooding protocol. However, due to the mentions of greedy routing as the standard means of routing, it can be assumed that some form of beaconing or update mechanism is present although the authors have created some ambiguity and uncertainty by not mentioning this.

Once landmark discovery has been completed, landmark routing is fairly simple and one of two outcomes can occur. If the landmark node entry for a destination is set to null then ordinary greedy routing is performed as there are no voids on the path. If there is a list of landmark nodes then these landmark nodes are used as intermediate destinations by the source. An additional layer of complexity (and overhead) is added by the fact that GLR is recursive and that upon looking up the list of landmark nodes, landmark discovery must be performed for each one of these landmarks! The logic presented in [22] for this is that doing so avoids the possibility of voids existing between two landmarks. Although the simulation results presented by [22] show better performance by GLR compared to GPSR in terms of energy consumption and packet delivery ratio, the solution proposed by [22] appears on the whole to be overly complex, and with regards to its description in [22] some aspects such as the role of beaconing and landmark discovery are left so vague as to make independent implementation difficult.

The work of [23] offers an augmentation of basic greedy forwarding that uses the concept of the potential field. Potential field is a technique commonly used in robot navigation. Potential fields are used to move a robot from one point to another using the virtual repulsive force from deflected objects in combination with the virtual attractive force of the target to create a dynamic potential field. Therefore, the target exerts a virtual force that is positive and the obstacles exert a virtual force that is negative [31]. The sum of both the positive and negative virtual forces is calculated (referred to as the resultant force) and then used to determine the direction the robot will move in [31]. Potential field is applied to geographic routing by contrasting the roles of the packet with the robot and routing voids with obstacles and determining the potential field of neighbouring nodes to select the neighbour with the highest potential field [23]. In simulated experiments the protocol achieved a similar performance to GPSR in terms of path failure rate and outperformed it in path stretch [23].

Another alternative to face routing as a backup scheme is Greedy Distributed Spanning Tree Routing (GDSTR) [25] that proposes the use of spanning trees in place of planarization algorithms. As with most backup schemes, the protocol begins life in greedy mode before switching to a spanning tree algorithm when the local maximum is encountered. More specifically, GDSTR uses a form of spanning tree known as a hull tree in which parent nodes store information about all of their child nodes in a convex hull [25]. The purpose of this is to reduce the number of trees that must be traversed, as GDSTR is able to only traverse a subtree containing only nodes with convex hulls that includes destination points [25]. When GDSTR switches from greedy to hull tree-mode then the packet is either sent to the parent node (if the current node's descendants do not have the destination in their convex trees) or sent to an appropriate child node if at least one child node contains that point in its convex tree. The former continues occurring until either the packet reaches a node that has a child containing the destination in one of its children's convex trees or the packet reaches the route and is dropped as the destination is deemed unreachable.

If the latter occurs then the appropriate child node is selected using the following process. If the packet arrived at the current node from a parent or was previously in greedy mode then the first appropriate child node is selected. If the packet was received from a child node (but not the last child) then the next child node whose convex hull contains the destination is selected. Finally, if the packet arrived from the last child node then the packet is sent to the parent unless the current node is the root in which case the packet is sent to the first appropriate child node [25]. GDSTR was simulated and evaluated against several hybrid greedy-face protocols including GPSR and GOAFR and outperformed all of these for average hop stretch. However, GDSTR is untested on mobile networks (acknowledged by the authors) and so it is unsure how reliable the hull tree architecture will be in networks where node mobility can significantly change the network. Although [25] does acknowledge the possibility of some network dynamics (mostly node death) through the construction of several spanning trees, the addition of mobility could lead to problems such as routing on trees that have changed significantly since their construction as well as the need to frequently construct new trees or repair existing ones to cope with such mobility. Therefore, it is difficult to determine whether the hull tree approach used by GDSTR is a suitable alternative to planarization without an a side-by-side comparison of GDSTR and several planarization-based approaches in scenarios of varying mobility.

Another interesting alternative to graph planarization-based backup schemes is presented in [26] which uses an algorithm known as Rotational Sweep (RS) that offers improved and more localised performance over planarization approaches. RS is introduced in the context of contention-based (beaconless) geographic routing protocols; protocols where nodes broadcast Request to Send (RTS) messages and await a reply from a neighbouring node which they will select as the next hop [26]. [26] asserts that such previous contention-based geographic routing protocols are essentially greedy geographic routing protocols and are therefore still susceptible to the local maximum problem. To solve this, and still maintain the ability...
of contention-based routing to forward packets without the need for local or global information they proposed a contention-based alternative to face routing. As with most backup schemes, the RS algorithm is activated when the local maximum has been reached and performs a counter-clockwise sweep to determine the next hop as well as a boundary traversal path for escaping a void [26]. The RS algorithm presented in [26] is based on the work of [27] and [28] both of which use a sweeping algorithm to create a planar Gabriel Graph. This is done by rotating a curve from the previous node to the next node so as to determine the next node for void traversal. The RS algorithm used by [26] differs from these as it acknowledges that the creation of a Gabriel Graph in a contention-based framework is impossible and instead uses the selected neighbour as the next hop – thus eliminating the need for UDG planarization altogether.

A proof for the combined greedy-contention RS algorithm is given that shows it is capable of guaranteeing packet delivery in connected UDG networks [26]. Simulation results also show that it is capable of achieving a hop count equal to or lesser than that of Gabriel Graph planarization algorithms. Similarly an implementation of the RS algorithm for use in standard, beacon-based algorithms is also provided. This work is therefore highly interesting as it presents a clear alternative to face routing backup schemes that has several similarities in terms of performance and also explicitly takes the UDG into account. The proofs of correctness as well as simulation results show that it is viable for real applications, as does its operation with beacon-based protocols as well as contention-based protocols.

A potential drawback of many popular backup schemes is that they are generally initiated when a dead-end is reached, however, as they are only utilised when the dead-end has been reached, this can lead to instances where they must traverse several nodes again in order to reach a point where normal performance can resume [29]. NEAR (Node Elevation Ad Hoc Routing) utilises a combination of real and virtual coordinates to identify and mark concave node [29]. NEAR is novel in its approach of trying to actively avoid routing into voids, and by doing so tries to avoid the necessity of using a potentially expensive recovery scheme. NEAR uses virtual repositioning in which nodes detected as concave are elevated by adding 1 to its coordinate dimension thereby virtually repositioning the node. The node is said to be elevated as with its virtual coordinate it is now located above the ordinary plane on which other (non-concave) nodes are located. The purpose of node elevation by NEAR is in avoiding use of concave nodes. The repositioning algorithm is run at start up as well as when neighbouring nodes move or a connection fails (i.e. hello messages time out). Once the NEAR has succeeded in performing node repositioning a void bypass detection algorithm is started which proactively identifies routing obstacles and determines recovery paths around them using perimeter (face) routing [29]. For instances in which voids cannot be avoided, the recovery strategy of NEAR uses a variant of face routing known as perimeter routing. Face routing exists both as a means of recovering from local maximum situations in greedy forwarding and as a routing approach in its own right. Although face routing can theoretically guarantee delivery on a connected planar graph, it holds the disadvantage of being less efficient than greedy routing. There may be merit in therefore developing protocols that combine the relative strengths of greedy geographic forwarding with those of face routing whilst negating or minimizing the disadvantages of both. Before discussing these hybrid protocols however, it is important that we examine face routing itself in greater detail. The next section examines this class of geographic routing protocol.

B. Face Routing

In addition to being used as a recovery strategy in hybrid greedy-face routing protocols, face routing is often considered an alternative to greedy routing as it builds upon the UDG by employing a process known as planarization in order to construct a planar sub-graph suitable for face routing [30]. A planar subgraph is one that contains no intersecting edges and is composed of a sequence of polygonal regions separated by edges (faces) [30]. Figure 2 shows a very simple graph which contains three intersecting edges and Figure 3 shows its planar subgraph following the removal of two of these edges. It should be noted that the original graph is a very simple one and is not a UDG, the intention of Figure 2 is to provide a simple illustration of what a non-planar graph looks like while the purpose of Figure 3 is to provide a simple illustration of what Figure 2 would look like once planarised.

Figure 2: Non Planar graph

Figure 3: Planarised Graph

Once the planar sub-graph has been obtained face routing can be performed using what is known as the right-hand rule where a face is traversed and the points at which the line connecting the source and destination is crossed are stored. Once the face has been completely traversed the intersection point that is closest to the destination is chosen and the algorithm repeats itself [7]. Figure 4 illustrates the basic operation of face routing. In this diagram the four red points marked S and D represent the source and destination respectively with the plain black lines representing the faces derived from a planarisation algorithm. While the black points represent the points closest to the green line which are intersected by edges of the graph. The faces that are traversed on the path from source to destination are numbered sequentially 1 to 5.

The routing begins at the source and from there face 1 is traversed using the right-hand rule with the point closest to the destination which intersects the s-d line being the point at which the next face is traversed; the next face being face 2.
Face 2 is then traversed using the right hand rule before the point closest to the destination that intersects the s-d line is discovered and the process moves on to face 3. The process of traversing a face and then switching to the next face at the point nearest the s-d line continues until eventually the destination is reached.

![Face Routing](image4)

Theoretically, planar subgraphs are connected as planarization does not affect connectivity, but in reality specifics of wireless environment mean planar subgraphs may lack connectivity [21] possibly leading to network partitions such as when one group of nodes is unable to reach another group. The following have been identified as causes of failure in planarised subgraphs; unidirectional links, disconnected links (i.e. links that were previously connected but are now disconnected due to planarization), cross links, and collinear links [30]. These causes of failure result from either irregular communication range due to the presence of radio-opaque obstacles or incorrect position estimation by one or more nodes [30] which are general instances of how real networks differ from the idealized UDG. Thus if the actual network does not conform to the UDG model then it is possible that planarization will result in a disconnected subgraph. Therefore, it should be apparent at this point that conventional planarization by its very nature inherits all of the weaknesses of the basic UDG and in turn can create a flawed subgraph that will potentially contain any or all of the previous three causes of failure.

In order to understand the weaknesses of face routing it is important to first understand the underlying model on which planarization is performed; the UDG. The UDG is one of the most popular models for describing wireless ad hoc networks, largely due to its simplicity and ease of understanding [7]. A UDG depicts a series of circles (all of equal size; the Unit Disks) containing at their centre a vertex, when two circles overlap a line (edge) between their vertices is drawn to indicate the link [31]. When the UDG is used to model ad hoc networks, the vertices represent network nodes which are located on the Euclidean (two-dimensional) plane, the circles/disks are the nodes transmission ranges, and so the edges are used to determine when a link exists between to network nodes (i.e., when their transmission radii overlap, two nodes will be able to communicate with each other) [7].

Figure 5 is an example of a UDG modelling of an ad-hoc network. The thick black circles at the center of the thinner black circles are the vertices representing nodes with the thinner circles being the disks representing the nodes’ transmission radii. It should be noted that in this diagram some of the circles are of slightly uneven length, but for the purposes of illustration they should be considered to be the same length. The red lines between the nodes are the edges which represent the bidirectional link between two nodes that occur when the disks representing their transmission radii overlap. In this diagram there are two crossing links although real networks would be likely to contain more than two crossing links. In spite of, and in some instances because of, its popularity as a base model for ad hoc networks a number of practical problems exist with the UDG which will be described in this section.

![Unit Disk Graph](image5)

One such problem with the UDG in practice is that conventional ad hoc UD graphs assume the transmission radii between all nodes is equal, and thus do not take into account factors such as interference. This in turn brings into question the effectiveness of protocols that rely heavily on them [32]. Other factors that can lead to a reduction in a node’s transmission radius include radio-opaque obstacles and the presence of multiple paths [39]. Another way in which actual networks can deviate from the unit disk graph model is when one or more devices have an inaccurate position estimate [32]. In addition to not taking into account physical factors (either due to obstacles or radio interference) the UDG also fails to take into account heterogeneity (i.e. some nodes may have weaker radios than others, thus even in perfect conditions their transmission radius is lower than that of other devices) [7]. This is especially relevant in wireless mesh networks where mesh routers may be equipped with more powerful radios than mesh clients. Similarly, it is possible that some nodes might intentionally alter their own transmission power (and thus decrease their transmission range) to save power or avoid interference with other devices (an approach often used in WSNs and cognitive radio). For protocols based on the DG and which work on its assumptions always holding true, it is possible that their performance in real scenarios will be undesirable as the real conditions will not match those of the UDG [33], something which will be explored later in the discussion of planarization.

As the UDG is that most popular model for wireless ad hoc networks much research has been carried out into ways in which the UDG can be modified or augmented to provide more accurate modelling of realistic physical conditions. An important development has been the modification of the Unit Disk Graph model to assume transmission ranges of all nodes are not equal or constant [36]. In this alternative model, two
nodes are assumed to be able to communicate with each other if the distance between them is less than a defined minimum transmission range (defined in terms of the devices and environment). If the distance is more than the minimum transmission range but lower than the maximum then direct communication is possible but not guaranteed; and finally if the distance between the nodes exceeds the maximum transmission range direct communication is impossible. Although the actual changes made to the basic UDG model might seem simplistic the introduction of minimum and maximum transmission ranges (as opposed to a single, uniform transmission range for all nodes and environments) allows for the modeling of scenarios in which transmission range can fluctuate and thus enables the design of more realistic routing protocols while maintaining the simplicity of the popular Unit Disk Graph model. This approach allows for robust geographic routing in networks where the ratio between the minimum and maximum transmission range is less than \( \sqrt{2} \). The ratio of less than \( \sqrt{2} \) is justified by the authors claiming that this is the largest value at which it can be guaranteed that when there are two nodes in a disk and there is another node in that disk at least one of those nodes will be aware of the other node’s existence [36].

Node mobility or other causes of topology variation (i.e. node death) are not addressed. Node mobility is acknowledged as a factor in ad hoc networks, and includes the probability of nodes discovering neighbours at any stage of the 3-step scheme and possibly returning to a previous stage. It is unclear how the loss of an existing node (i.e. due to mobility or death) will affect the graph/network’s connectivity and whether this would trigger a return to previous stages and/or the need for a new graph. Additionally, there is an assumption that all links are bidirectional (i.e. if node a can hear node b then node b can hear a), and in instances where links are not bidirectional, the system fails due to problems with Gabriel Graph planarization [36]. Kuhn et al. (2008) analyses the performance of geographic routing algorithms using the quasi-Unit Disk Graph of [36]. It was found that a combined greedy geographic and geographic flooding algorithm is able to obtain a message-optimal performance in the worst-case scenario and message-efficient performance in the average case. Conventional geographic routing algorithms (such as AFR, GOAFR, and GOAFR+) are able to match their UDG performances if the ration between minimum and maximum transmission ranges is less than \( \frac{1}{2} \) [7].

In [37] the Unit Disk Graph model is adapted to take into account interference and noise using the Signal Interference plus Noise (SINR) radio model that allows a modified UDG to be applied to real wireless networks. The SINR model, as its name suggests, takes into account signal interference and noise as well as using a distance model for signal decay, which results in nodes being able to hear transmission from other nodes when the ratio of signal to noise is above a defined threshold [37]. A stated drawback of the SINR model is that it does not provide any means of developing a suitable decentralised algorithm [37] hence its combination with the UDG. Although there are similarities between the combined UDG-SINR model and the quasi-Unit Disk Graph model of [36] it is argued in [37] that the quasi-UDG model is relatively simplistic whereas scheduling using the SINR model has been shown to be an NP-complete problem by [38]. The work of [37] focuses on the development of a theoretical algorithm that emulated the UDG topology under the constraints of the SINR model, thus allowing UDG routing to be performed on a more realistic network model.

In addition, to the problems caused by underlying weaknesses in the Unit Disk Graph model, various errors resulting from the planarization process are possible, [30] identifies the possibility of incorrectly changing face during the traversal process, as a potential error resulting from incorrect planarization of real networks (incorrect in practice, but correct in theory). A stated drawback of the commonly used right hand method for face traversal is that by always using this method face traversal is in effect blind and can lead in significantly longer path being traversed than may always be necessary [22]. Na and Kim (2006) identify a similar problem known as triangular routing in which packets arriving at the dead-end (i.e. a node that has a local maximum) are backtracked upon switching to face routing and sent to a different node [22]. However this is obviously inefficient and a solution that could identify a more suitable node than the dead-end without the need for backtracking is obviously desirable. Similarly, if planarization has been incorrectly applied (due to problems described above) and the resulting subgraph is not actually planar then face traversal using the right-hand rule may fail entirely. An example of this is where a node is located inside a face and is connected to the graph by an edge that crosses the face [33]. In this instance, the right-hand rule would lead to traversing the entire face but fail as it would be unable to find an edge crossing the line between source and destination that is located closer to the destination than the point at which traversal of that face started [33].

The claim of incorrectly changing face made in [30] is challenged in [34]. In [34] a formal proof is given that shows the correctness of the face routing algorithms used by GFG and Compass II to demonstrate that it guarantees delivery on any planar graph. It is stated that the claim of incorrect faces changes being made by face routing protocols in [30] is incorrect and that such a claim arises from errors in GPRS’s implementation of face routing.

The next section reviews hybrid greedy-face protocols, a category of geographic routing protocols that was mentioned briefly earlier in this paper. The general aim of the protocols in this class is to combine the efficiency and simplicity of greedy forwarding with the delivery guarantee of face routing; however there is still significant diversity in the actual implementations of this general idea.

C. Hybrid Greedy-Face Routing

The previous section introduced and explained the principles and operations of face routing and its variants. In previous sections it was mentioned that in addition to existing as a routing strategy in itself face routing was often used as a backup strategy when the local maximum problem was
encountered. Typically hybrid greedy-face routing protocols will start in greedy forwarding mode and then switch to face routing when the current node is unable to find a neighbour closer to the destination than itself. As greedy forwarding is generally more efficient than face routing it is therefore desirable to perform as much routing in greedy mode as possible. However, there is also the issue of when exactly to switch back to greedy mode; switch too soon and another local maximum may be encountered (thus necessitating another costly switch back to face routing) but switching too late means more of the routing process is carried out in inefficient face mode. The protocols reviewed in this section are all examples of protocols that combine greedy and face routing but that differ in both their approaches to the issue of when to switch from face to greedy and other protocol design issues.

GOAFR (Greedy Other Adaptive Face Routing) which builds on OAFR is an example of a hybrid greedy-face routing algorithm [12]. GOAFR aims to use greedy routing until a local maximum is encountered, at which case OAAR face routing is used as a recovery strategy. Once OAAR has been initialised it will continue to run until either the destination is reached the first face is probed, or disconnection is reached. In all but the second case these will result in the termination of the protocol, however the second case will result in a switch back to greedy routing using the node identified as closest to the destination by OAAR [12]. GOAFR+ is a hybrid greedy-face algorithm which aims to be asymptotically optimal [30]. It is an augmentation of the GOAFR algorithm that contains several key differences. A difference between GOAFR and GOAFR+ is that when GOAFR+ uses face routing, it does not always traverse the full boundary of a face but instead uses two counters to keep track of how many nodes traversed were closer or further away from the destination than the node at which face routing started [30]. In instances where the number of nodes located further from the destination that have been visited is higher than the number nearer the destination that have been visited GOAFR+ will return to the previously visited node which is closest to the destination and from there resume greedy routing. Another difference between GOAFR and GOAFR+ is that GOAFR+ makes use of clustering which allows it to avoid the assumptions made by GOAFR that the distance between nodes cannot fall below a constant minimum bound. The use of clustering allows for the extension of GOAFR+’s asymptotic and worst case optimality to general (unbounded) Unit Disk Graphs when using linearly bounded cost functions [30]. For general (unbounded) Unit Disk Graphs a Clustered Backbone Graph is identified through which routing between ordinary nodes is performed by sending the packet to the current node’s associated dominator (node’s belonging to a dominated set which exists as a backbone embedded in the graph) which routes it along the backbone to another dominator which transfers it to the ordinary destination node [10]. Thus the Clustered Backbone Graph contains all ordinary nodes not in the routing backbone (i.e. non-dominating nodes) and is not bounded while the routing backbone consisting of dominating nodes is [10]. GOAFR+ performs optimally on average case graphs as well as achieving asymptotically optimal performances on worst case graphs [12].

The earliest and possibly most important example of hybrid greedy-face routing is the aforementioned GFG [8]. In [8] several protocols relating to geographic routing and geocasting are proposed with GFG described as beginning operation in greedy mode before switching to the Compass II (face) algorithm when the local maximum was encountered and then switching back to greedy mode when the local maximum had been resolved. Although Greedy Perimeter Stateless Routing (GPSR) [13] is one of the most widely cited geographic routing protocols it would appear that much of its work is in fact a duplication of GFG. The main differences between GPSR and GFG are documented on the website of Ivan Stojmenovic [35] and are as follows; GPSR’s use of the Relative Neighbourhood Graph (RNG), changing face before crossing, the use of a realistic MAC simulation, and simulation with mobile nodes. Another apparent difference between GFG and GPSR is the process for deciding when to terminate face/planar routing and switch back to greedy mode. The original description of GFG states that face routing will terminate when the packet arrives at a node that is located closer to the destination than the node where face routing commenced [8]. GPSR is described as terminating perimeter mode when the distance between the current node and the destination node is lower than that of the node where the packet entered perimeter mode and the distance².

While GFG and GPSR both use distance to terminate face mode, GOAFR+ [12] switches back when the number of nodes further to the destination than the node at which face routing started is greater than the number located further away. Both approaches have their relative strengths; GFG/GPSR’s aims to avoid areas in which several nodes may experience the local minimum problem (i.e. sparsely populated regions) while GOAFR+’s aims for optimality by switching from face routing in regions where a large number of sub-optimal nodes have been traversed. The weaknesses of both are that the approach used by GPSR could lead to situations in which face routing is employed in non-sparse regions of the network following a local maximum, while GOAFR+’s has the potential for switching to greedy forwarding only to encounter a local maximum. However it would appear that due to the slight differences in face mode termination between GFG and GPSR that GFG is not likely to suffer the possibility of using face routing in non-sparse regions as it would terminate face mode quicker than GPSR. GPSR’s greedy routing implementation has been modified to include ‘non-Euclidean’ distance metrics for measuring progress and selecting neighbours [40]. The metrics used by [40] were local shortest path and weighted distance gain that together aim to provide loop freedom, guaranteed delivery, efficient obstacle avoidance, and load balancing [40].

The approach used in [10] offers a radical alternative to hybrid greedy-face routing that differs from other approaches by opting r to allow nodes to store complete face information about every face they are adjacent to. Although storing more
information may run contrary to the localised aim of geographic routing, it is argued in [10] that storing this information significantly improves face routing performance. Path Vector Exchange (PVEX) works on a planar graph (either the Gabriel Graph or Relative Neighbourhood Graph) by using beacons to share path vector information for each face that it is adjacent to, the number of faces it is adjacent to is equal to its degree (number of edges to which it is adjacent to). Beacons are only sent to neighbours adjacent to one of the same faces at the sending node, and thus beacons received from nodes adjacent to other faces are ignored as are messages with a sequence number lower than the node’s current sequence number. The sequence number is used to identify cases in which a new node has entered the network or an existing node has left. In both cases the sequence number will be incremented (i.e. in the first case, upon receiving a beacon the new node will increase the sequence number and send a beacon itself, and in the second the node that discovers the loss of another node will send beacons with an increased sequence number to the other relevant nodes). The content of the beacon depends on whether a face has stabilised or not, this is dependent on the sequence number remaining constant for a defined period of intervals, when the face enters a steady state beacons will no longer contain path vector information, with only the sequence number being broadcast. When a node receives a beacon with an increased sequence number it will resume transmission of path vector information as the increased sequence number indicates that either a new node has joined or an existing node left.

Oblivious Path Vector Face Routing (OPVFR) is where all nodes have full face information for the entire network, and is thus able to determine which edge from its set of edges connect to adjacent faces that have the lowest Euclidean distance between itself and the destination [10]. To reach this node two forms of forwarding are used called direct and target node. In direct forwarding the packet is forwarded to any neighbour which has a distance between itself and the chosen node lower than the distance between the current node and the chosen node. Direct routing is only applied if the chosen node is on a face adjacent to the current node, and in all other cases target node forwarding in which the packet is forwarded to any neighbour adjacent to the same face as the chosen node where the distance between the target node and the chosen node is less than the distance between the current node and the destination – if this condition is not met the packet is deemed undeliverable and dropped. OPVFR is proven to guarantee delivery from source to destination on a planar graph where node in the graph knows all of its faces [10]. However, in some networks faces are so large that it is undesirable to have all nodes exchanging complete path vectors of certain faces, therefore a limiting factor is determined known as h, that allows all nodes to have knowledge of its path vectors up to depth h+1. However, it is impossible to guarantee delivery if nodes are limited to knowledge of nodes located h+1 away from them. However it does not mean that delivery in such instances is impossible [10].

Greedy PVFR (GPVFR) is a tripartite hybrid protocol that combines greedy, OPVFR, and perimeter routing (from GFG/GPSR). As with most greedy-face hybrids, routing takes place in greedy mode until a local maximum is encountered at which point OPVFR is entered using ‘virtual’ edges in instances where only partial information is available [10]. Target nodes chosen by OPVFR are recorded in the packet, and it is possible that a better node is identified before the target node is reached (as a result of only partial information being available) in which case this node is made the new target node. When OPVFR or greedy forwarding fails and OPVFR cannot be used as the current node cannot identify a suitable node adjacent to one of its faces, perimeter routing is used based on the right-hand rule method used by GPSR. Face traversal is terminated when a face closer to the destination than the current face is discovered, a packet is declared undeliverable, or a node closer to the destination than the node at which perimeter mode was entered is discovered. GPVF performs better than both GPSR and GOAFR+ in terms of path stretch and hop stretch, and needs only slightly more state information at each node than GOAFR+ [10]. However there may be costs incurred by switching between the three available ‘nodes’.

The protocols that have been encountered in previous sections whether greedy, face, or hybrid all have one thing in common; they are designed with the (usually) implicit assumption that the terrain on which they will be used is planar (flat). There are various examples where such an assumption will not hold true (i.e. where nodes may be located at different heights) and this has led researchers to begin focusing on the design of geographic routing protocols that are specifically able to handle such environments. The next section takes a detailed look at the protocols belonging to this category and the design issues surrounding them.

D. Geographic Routing in 3-Dimensional Networks

The majority of research in geometric routing has focused on two-dimensional (2D) ad hoc network; networks that are flat as opposed to three-dimensional (3D) environments such as hills [41] [42]. GPS however is able to provide three-dimensional coordinates (latitude, longitude, and altitude) [14]. As the 2D approaches assume the surface is flat they are not suitable for environments in which nodes may be positioned at different planes (i.e. a hilly environment or a high-rise building) [42]. Similarly, routing approaches based on the use of a planar sub-graph would be unable to guarantee delivery in 3D networks as concepts key to planar sub-graph routing such as face perimeters do not exist because the terrain is not planar [43]. It is important to note that while protocols that assume the network is two-dimensional may not work optimally (or at all) in a three-dimensional network, an approach that takes into account the possibility of nodes being position at different heights will still work in a network where all nodes are positioned at the same height. However, the performance of such a protocol may not always be as efficient in a 2D environment as a pure-2D approach. It is also important to acknowledge that although 3D algorithms by
their very nature take into account the fact that nodes may be positioned on a non-flat system, they do not necessarily take into account other characteristics such as interference and obstacles that may prevent nodes located a short distance from each other communicating successfully.

Regarding geographic routing in non-planar (three-dimensional) networks, several novel approaches have been proposed including a state/memoryless and localised geographic routing algorithm for three-dimensional networks that uses a randomised (rather than deterministic, as if often used in 2D solutions) recovery strategy [44]. The protocol proposed in [44] uses the Unit Ball Graph, described as a 3D equivalent of the Unit Disk Graph as the basis for their 3D geographic routing protocol. As mentioned earlier, [44] asserts that deterministic recovery algorithms are impossible for 3D networks and so their solution is randomized, and is in fact based on the Random Walk model [44] – a mathematical model often used in the simulation of node mobility in wireless networks.

The Unit Ball Graph model is also used by [45] who use a non-random approach to explore the possibility of a simple routing algorithm guaranteeing delivery in a 3D network – ultimately concluding that such a task is impossible. [42] attempt to use the principles of hybrid face-greedy routing protocols designed for two-dimensional networks in three-dimensional networks through the use of Partial Unit Delaunay Triangulations. Its operation is somewhat similar to that of 2D greed-face hybrids, as it begins operation in greedy mode before switching to a recovery mode based on splitting the network into triangular sections similar to the concept of faces in 2D planar solutions. These triangular sections are then traversed in order to move from one point to another. Another similarity is apparent from the use of the PUDT algorithm to remove intersecting triangles – an approach analogous to the removal of intersecting edges in planarization algorithms.

In contrast to attempts which seek alternative techniques for 3D geographic routing (although some of these techniques were similar to those used in 2D geographic routing; i.e. the Unit Ball Graph and Partial Unit Delaunay Triangulations) [46] set out to modify 2D face routing for use in 3D networks through Slab Routing [37]. Slab Routing works by dynamically creating space partitions known as slabs; nodes contained within a slab are projected onto a plane, and face routing is then applied to this plane [37]. A potential drawback of Slab Routing is the requirement for a dense network [37], which makes it inappropriate for environments in which either the entire or significant portions of the network are sparse (whether permanently or temporarily).

### III. GEOGRAPHICAL ROUTING SECURITY

Security is one of the biggest problems facing wireless ad hoc networks [17] where inherent weaknesses can act as an obstacle to widespread commercial use of ad hoc networks [47]. A particularly significant threat facing ad hoc networks is that of traffic analysis (in which network traffic is observed by a malicious node, which then uses information gathered to launch an attack) due to the inherent vulnerability of the wireless medium [17]. Figure 6 shows an illustration of how a malicious node (the laptop) could easily intercept unencrypted wireless communications between two nodes (the PDAs).

![Figure 6: Wireless packet interception](image)

In addition to stealing information through traffic analysis and interception, malicious nodes have been known to erroneously state that they are one hop away from a popular destination and thus receive all traffic sent to that node (could be used in conjunction with, or as an alternative to traffic analysis) and corrupting an in-transit-route request to artificially alter the flow of data [47]. These techniques allow nodes that are not able to directly intercept communications destined from a particular node to receive these packets by means of deception.

This is however a threat to which conventional, centrally administered Wireless LANs (WLANS) are also vulnerable. However, it is a lack of infrastructure and coordinated control that makes MANETs more vulnerable to attack than other wireless networks [48]. This is logical, as WLANS are likely to be administered in a manner similar to wired LANs with centralised security and authentication procedures. In contrast, ad hoc networks generally rely on individuals to provide their own security and as a result the level of security varies depending on the users and application of the network (for instance, a military or commercial network is likely to be more secure than a group of individuals running an ad hoc network on their devices default settings).

Location data is a specific security threat and geographic routing protocols typically put performance ahead of security [17]. Attackers can be classified into malicious (a node outside the network) and compromised user (a node inside the network) depending on their intent (i.e. a malicious node sets out to attack the network while a compromised node is merely an unknowing pawn for an attacker). The possible types of attack a node running geographic routing is vulnerable to (some of which are general to all ad hoc routing protocols) are message tampering, message dropping, message reply (wormhole), location table tampering and blackmail attack; such as falsely identifying a good node as bad such as could
be used in a network where some form of rating or trust system is in place [49]. Zhi and Choong (2005) frame the security weaknesses of ad hoc routing in terms of privacy, stating that geographic routing protocols present an additional level of complexity due to their obvious reliance on geographic information [50]. In terms of privacy, location updating (both to neighbours and location servers) helps malicious nodes track their targets and possibly create profiles of a node’s motion for use in future attacks [41]. [18] highlights a particularly alarming scenario in which a malicious user uses position data to physically track down another user, thus violating that user’s virtual and physical privacy. Both [50] and [18] highlight the impossibility of removing or reducing geographic information and instead argue that some form of anonymous routing in which position information is separated from a node’s identity is a viable compromise that protects the privacy of users while maintaining routing efficiency.

The protocols described in this section all address the issue of security when dealing with location data in various ways. Several of the protocols attempt to perform anonymous (or quasi-anonymous) routing in which some form of obfuscation is used to hide the identities of the nodes participating in the routing process. We will begin our review of secure geographic routing by examining these protocols in the next section.

A. Anonymous Geographic Routing

In [50] a comprehensive framework for anonymous geographic routing that includes an anonymous table, authentication service, and anonymous location service is presented. The basic premise of this approach is that although location data can be used maliciously, if the accuracy was to be decreased then routing would be no more (and possibly less) efficient than standard topology-based routing schemes. Therefore this system separates location data from a node’s identity. For instance, the anonymous table stores pseudonyms alongside position data rather than actual node IDs. Similarly, messages contain what is known as a trapdoor, which is a value that can only be decoded by the recipient, and informs the recipient that it is the intended target; the assumption that all nodes are able to obtain certificates from a trusted third party is made. In order to protect the sender’s anonymity, broadcasts are used for distributing messages at all stages. The anonymous neighbour table is periodically updated based on a node’s mobility. In order to resolve the issue of pseudonyms being mixed up between updates, nodes remember previous pseudonyms so they can determine if they are the recipient. It is not mentioned explicitly how pseudonyms are generated, and the only information given about them is that they are numerical. An authentication system based on ring signatures in which a ring is considered anonymous only if all nodes are indistinguishable from each other. The authors of [50] do not disclose however the security mechanisms which will affect the complexity of the algorithm. The overhead that is incurred is only briefly mentioned when the authors state that the higher the level of anonymity, the higher the overhead. Similarly, an assumption is made that all nodes are able to obtain authentication certificates, which may not always be practical.

An anonymous routing protocol is proposed in [17] that also includes an anonymous position management service. In [17] destination positions are encrypted using a key held by all legitimate members of the network, so that only member nodes can obtain destination locations. Position servers (ordinary nodes that index the locations of other nodes) are responsible for providing authentication services. Nodes wishing to obtain position information of other nodes must first authenticate themselves with the position server who will then provide the relevant information. By having ordinary nodes act as trusted position servers, the possibility of corrupting or faking position servers is raised. Similarly, in order to register with the position server the node only needs to provide a temporary ID (all IDs in this system are temporary) as well as its position in order to gain access to the public and private keys necessary for authentication. Once registered (most likely with fake positional data) a malicious user could then query the positional server for updates on a target using the keys it has obtained from the position server.

Also, no mention is made regarding the level of anonymity between a node and the position server (i.e. whether it keeps track of MAC or IP addresses), thereby raising the possibility of one malicious node registering itself with the position server in order to corrupt the server’s information (i.e. flush out real node data) or obtain multiple node location’s in quick succession (thus acquiring the ability to launch a targeted attack against several nodes). When a source node contacts the position server to obtain a destination, it is issued with an authentication and token value (in addition the position of the target node). The authentication value is used by the source to authenticate route replies from the destination, while the token is used by the destination to authenticate the source. Similarly, if the destination does not receive any data from the source then it will not engage in further communications, as it is assumed the source is not a legitimate entity. Forwarding is limited a calculated number of hops between the source and destination in order to reduce overhead. The downside of this is that due to dynamic behaviour the destination may have moved so that is no longer located within the maximum hop count estimated by the source, in which case communications fail.

In [18] the concept of an anonymity zone is used; an anonymity zone is an area in which a message is broadcast based on a fake destination address. The idea being that the destination’s real location cannot be obtained from packets, while at the same time the packet can still reach the destination. This is achieved by calculating the fake destination address so that the genuine destination node will be within the anonymity zone, which is a geographic area of set-size containing several nodes. When the packet arrives at the first node in this area that node broadcasts it to all other nodes, and the genuine destination is then able to decrypt the message. This means that any malicious nodes will not be able to
directly find the real destination and if the anonymity zone is large enough, they will have several nodes to guess from.

The ‘fake’ address used for routing is known as the pseudo destination (PD) and is generated so that there is reasonably high chance of the intended destination receiving it (i.e. it is reasonably close to the destination’s real location). Due to the use of anonymity zones, the PD only needs to be accurate enough for any node in the anonymity zone to receive the message and then broadcasts it so the correct destination can receive it. A potential drawback that is not addressed is that if the PD is that of another node, thereby leading malicious nodes to target that node instead; a ‘wrong node’ attack. Another possible issue is to ensure that the real destination actually receives the packet it must be able to hear the broadcast. This in turn means that the distance between the destination and the PD must not exceed a certain value, so that the destination can hear the broadcast from the node nearest the PD. In sparsely populated networks, it is then possible that the biggest possible anonymity zone may only contain a small number of nodes (i.e. four or less) which would allow a malicious node to concentrate on these four nodes and analyse their behaviour, thus the smaller the population of the anonymity zone, the lower the value of anonymity it can offer.

A variant of the basic approach uses geocasting to increase the size of the anonymity zone. The basic idea behind this variant is to use localised flooding and replace the PD with an area, in which the first node to receive the message will flood the message to other nodes in the area. In order to reduce the considerable overhead that flooding can cause, a variant in which the area is subdivided into a number of regions in which only one transmission is needed for all nodes in a region to receive the message is required. Nodes from another region that receives the message can then flood it in their own region. There is a problem with the first two approaches in which a certain area (i.e. the PD in the basic approach, or the geocast region in the second) could turn into a ‘hot spot’ leading to a large communication overhead in that area. To this end, a third variant is proposed in which a route featuring redundant hops is built in order to increase the anonymity zone to cover an entire path. To do accomplish this, a PD is selected in which the distance between the actual destination and the connection between the source and the PD is below a certain value. This means that the actual destination will intercept the message from a node on the path before it reaches the PD (hence redundant hops). This means that the anonymity zone includes not only the range from which the PD can or the source can broadcast to, but the entire path between the source and PD. Thus, even in sparsely populated networks the anonymity zone is significantly increased in comparison to the anonymity zones of the other approaches. Several weaknesses exist however (such as an intersection attack where an attacker is able to determine destination lies on the intersection between two anonymity zones, thus decreasing the number of potential nodes) and the issue of node mobility is not addressed. A node must remain in the anonymity zone, or that other nodes leaving the anonymity zone can affect the level of anonymity, however no mechanism exists for coping with either the destination or other nodes leaving. Possible modifications could allow for an adjustment to the anonymity zone side (possible by switching from the basic approach to geocasting) or even drawing up a new anonymity zone without starting a new session (i.e. by updating the destination with a new PD).

The work of [2] also uses the concept of pseudo identifiers to hide the real identity of nodes; however the pseudo identifiers used by them are calculated based on location and time. The argument is that no two nodes can be at the same position at the same point in time; this is logical enough, however it is possible that due to discrepancies in position accuracy and lack of clock synchronisation that two nodes believe they are at point X at time Y, thus giving them the same pseudo identifier. Unlike other geographic routing protocols, the one proposed by [2] does not involve the regular exchange of position information between neighbouring nodes, but instead features a route contention mechanism. This mechanism works by having the first sending node (i.e. the message’s source) send route requests to neighbouring nodes who then contend to become the next hop, the forwarding node then classifies these nodes and selects the best one (which is usually the node closest to the destination’s location or reference point) and that node will receive the message and repeat the process. The reference point mentioned is used as an alternative to disclosing the exact location and is calculated using large random values, so that malicious nodes are left with an unfeasibly wide area to search. Nodes that win the contention process are then provided with the actual destination of the node, although it is possible that as the contention process is based on the reference point rather than actual location that the successful node may not be the nearest to the actual destination. The use of the reference point is somewhat similar to the redundant hop variant of ZAP used by [18] in which a pseudo destination beyond the actual location of the destination is used in order to broaden the possible area in which the actual destination is located. The main difference between these two approaches is that the one used by [2] does not explicitly feature redundancy in the sense that the message will not intentionally ‘travel past’ the destination, but due to the contention process being based on the reference point rather than the actual destination it is possible that the route taken to the destination is sub-optimal. One of the interesting features of this protocol is that it has neighbouring nodes via for the right to forward a message rather than having the current node make a decision based on information it has about its neighbours. The obvious advantage of this is that it does not require the continual sending of update messages to inform nodes of their neighbour’s states, as the current node will discover such information during the contention process. Similarly, the contention process decreases the chance of a node having ‘stale’ information as it will be receiving state information directly from neighbouring nodes rather than relying on a table.

In [51] it is claimed that both source and destination anonymity are provided through the use of random geographic-based routing in hierarchical zones. The network is assumed to be rectangular and is divided into two zones
Initially, and in turn one of these zones is then partitioned further, this process continues until n number of zones exist [51]. When a source node wishes to send a packet to another node it will determine whether the destination area and the random forwarding node used to reach this area are in the same zone as itself, and if so will further partition the zone until itself and the other nodes are in different zones. The selection of the random forwarding node (following the selection of a random location) is based on GPRS’s greedy forwarding strategy (to select the neighbouring node nearest the random area), and is repeated until the message arrives in the destination zone, at which point it is broadcast to all nodes in the zone. Although the use of broadcasting is undoubtedly inefficient, it does have the advantage of aiding in anonymity preservation as any listening nodes will not be able to discern one particular target. Similarly, the random routing method may suggest inefficiency despite its obvious advantages in helping preserve anonymity (by using ‘random’ next hops any malicious nodes will be unable to determine a traffic pattern/profile from source to destination). Regarding encryption, route request are encrypted by the destination’s public key (presumably obtained from either the location server or some form of network-wide key distribution scheme) while data messages are encrypted using a symmetric key between destination and source (and vice versa). Location servers do not provide individual node’s locations and instead only provide requesting nodes with a zone’s location, with MD5 being used to hash a node’s MAC address, time, and position to provide its anonymous identifier.

In [52] the concept of an anonymity zone (referred to as a cloaking region) is combined with that of privacy profiles in which users are able to associate specific levels of privacy with specific locations (i.e. a hospital) so that cloaking regions can then be constructed based on a user’s privacy needs. This can be defined either in terms of number of users located in the region or how much of the region includes the specified location, however the work of [52] does not explicitly cover cloaking region construction, and instead accepts any valid cloaking regions as inputs. Cloaked node locations are obtained through the use of a directory service the provides requesting nodes with node ID and cloaked locations – as no exact locations are provided, once the packet is inside the cloaked region geocasting is necessary (geocasting have the disadvantage of incurring a high communication cost without guaranteeing delivery). Neighbour information is also shared using geocasting in which nodes advertise themselves to other nodes within their cloaking region as well as nodes located slightly outside the cloaking region but within transmission range. The use of geocasting at the beaconing stage makes tracking down the advertising node impossible as all nodes will relay the advertisement to all other nodes within the cloaking region (as well as node’s just outside) thus preventing malicious nodes from determining whether it has received the advertisement from the advertising node or another node relaying it. Beaconing is a function of time so that nodes moving faster send beacons more frequently. The authors identify three types of criteria for next hop selection during the routing process; advancement (how closer the packet will move to the destination), coverage (whether the next hop can be reached directly, for instance if the next hop’s cloaking region is partially contained within the current node’s transmission range then geocasting must be used), and cloaking region size (node’s located in a large cloaking region have a higher communication cost than those in smaller regions as geocasting must be used within cloaking regions). These three criteria are then considered so as to optimise QoS metrics such as packet delivery ration, end-to-end delay, and communication overhead. A linear combination function is then used to determine a merit value for each potential forwarding node which is used as the basis for deciding the next hop.

B. Integrity and Authentication

In addition to privacy threats, another security risk specific to geographic routing protocols is that of position falsification. Position falsification is where a malicious node gives a false location. This is done to ensure that it is the next hop in a geographic routing strategy (i.e. if greedy forwarding is used, then the packet will be forwarded to the nearest neighbour) to allow for packet interception, which can then be used to monitor packets, alter them, drop them, or create routing loops [53]. In addition to posing a security threat, falsified position data can also be used as a means of avoiding participation in the routing process (i.e. by giving a position that is significantly distant from the route that the likelihood of being selected is low) [53]. It was mentioned earlier that most geographic routing schemes make use of some form of location services in which nodes send their latest position to update servers who can then be queried by nodes looking for the destination of another node. Given the potential for nodes to use false position data maliciously, it would seem apparent that some form of authentication should be in place to stop nodes from providing location servers with intentionally inaccurate information about themselves (as discussed above) or another node (i.e. using their own location as the location of another node in order to intercept its messages).

A hierarchical, zonal-routing protocol based on GLS with attachments for security is proposed in [54]. For all nodes that are maintained within a node’s neighbour table, the node has a public and private key to enable encryption and decryption of messages sent between the two nodes in order to prevent eavesdropping. A node’s identity and public key are provided by trusted servers, although exactly how these are obtained is not described. Although it is mentioned that the servers are responsible for facilitating key provision, the possibility of using keys to validate position updates is not mentioned. Similarly, since it is not stated otherwise, it can be assumed that node’s identity remains permanent in contrast to the systems describe earlier in which a node only has a temporary identity. As nodes regularly broadcast hello messages containing their identity and public key, it is possible that a malicious node could obtain a certificate (as a certificate is required to view ID and public keys) from the relevant server in order to harvest such data.
GLS is also used as a base model by [49] Secure GLS (SGLS) adds features that allow for authenticated location update and query, as well as secure hello message exchange. Secure location update is accomplished through the inclusion of a digital signature in the link update message alongside TIK-enabled neighbour authentication. After receiving a location update, the location server will store the relevant digital signature alongside other relevant node information; this means that when another node wishes to discover the location of that node it can use the digital signature to ensure that the information provided by the location server is correct. Although this mechanism does not address the possibility of node’s providing false position data for themselves, it does prevent malicious nodes from pretending to be other node’s by sending location updates with their own location data in place of the location data of the node they are pretending to be. To combat the problem of malicious or selfish intermediate nodes, SGLS allows nodes to determine whether other nodes are suspicious or unable to deliver the message, and then specify preferred forwarding nodes and squares that receiving nodes will deliver the message to. Regarding the routing (or forwarding) aspect of SGF (Secure Geographic Forwarding), potential problems arise from the fact that it inherits several flaws common to geographic routing protocols; namely the assumption that all links are bidirectional (which does not always hold true in practice) and the use of greedy forwarding and directional flooding as its primary routing strategies – both of which have been shown to be inefficient. However, it is also important to note that the emphasis of the work in [49] is on providing a secure means of achieving geographic routing via an integration of routing, position update/lookup, and neighbour reputation scoring all of which aim to authenticate genuine nodes and isolate malicious or compromised nodes from all aspects of the routing process. In simulated tests, the location reputation system (both the first-hand and second-hand information variants were used) was found to only result in a significant packet overhead when compared to standard GLS when the number of attackers increased, while simulations with increased mobility did not result in a significantly increase level of packet overhead. The work of [55] also uses digital signatures to verify updating nodes identities on top of a location service similar to that of GLS.

In addition to anonymous routing and secure location services, another area in which geographic data has been used to enhance ad hoc security is that of key distribution. Key distribution is the process of distributing secure keys amongst nodes in a network to allow them to send encrypted messages to each other. In [56] a secure key distribution scheme is proposed that uses a node’s physical location as a means of saving energy when transmitting keys to multiple neighbouring nodes by using the Euclidean distance between nodes to create clusters of nodes with minimal spatial distance between each other. The clustering is achieved through the use of the K-means algorithm in instances where a node’s transmission power is proportional to the distance between itself and another node, and the K-medoids algorithm is used when the distance is not proportional to transmission power. The reliance on transmission power fails to take into account factors that could impair a node’s ability to communicate with another node such as interference of obstacles, thus it is possible a cluster could be built in which one or more members are unable to communicate with each other despite having adequate transmission power. Similarly, no mention is made of how the clusters will react to member mobility.

IV. MOBILITY AND LOCATION

Most geographic routing protocols make the assumption that all nodes are able to obtain their own location through a location system such as GPS. However GPS has a number of weaknesses, for example, it is not always accurate enough, does not work indoors, and is not available on all devices by default (therefore additional hardware or software may be required). Similarly, GPS can potentially lead to significantly increased power consumption [57] which is obviously undesirable in mobile devices, the majority of which are battery powered. With specific regards to geographic routing, the potential unreliability of GPS can have a negative impact on routing, particularly in the case of node selection [58]. In the second section location inaccuracy was mentioned as a potential cause for failure of the UDG model and planarization techniques, while location inaccuracy can also negatively affect greedy routing (for instance, a node believing it has reached a local maximum situation or conversely believing a neighbouring node is suitable for forwarding when it is not). Kwon and Shroff (2006) specifically state the possibility of location inaccuracy leading to routing loops where the packet travels backwards [58].

Although various authors indicate that their geographic routing algorithms are for use with either GPS or alternative systems, and the likes of [59] state that their GPS-free positioning systems could be used in conjunction with geographic routing protocols, very little research has been conducted in the area of GPS-free geographic routing outside of wireless sensor networks. Although it is possible that GPS alternatives such as localisation algorithms could be used in place of GPS, these constitute a highly experimental area, and have several flaws of their own. Negative effects of incorrect or out of date location information include incorrect neighbour selection (i.e. selecting a neighbour that is located further from the destination than another neighbour because the position information about one or more of the neighbours is inaccurate). As mentioned above, this is especially significant in greedy routing where it would be possible for a local maximum situation to be entered into incorrectly (i.e. when position information incorrectly indicates that no neighbour is located closer to the destination than the current node, but in fact another node is located closer but the position recorded for that node is incorrect) which could lead to a packet being dropped [60] or an inefficient recovery strategy launched. Loops in which the packets travel backwards and sub-optimal routes are also problems that can arise from inaccurate location data. In contrast, it can be argued that in a few instances location inaccuracy is actually beneficial; for
instance, in a local maximum scenario a node would either drop the packet or initiate an expensive recovery strategy, but if the positions of one of its neighbours was inaccurate to the extent it believed that neighbour was closer to the destination than itself, it would forward the packet to that node, and the local maximum could potentially be ‘skipped’ – albeit purely by accident [60].

A. Location Errors

The work of [58] takes into account the probability of location error when making a routing decision. Using an objective function, [58] estimate the probability of both transmission failure and backward progress based on calculations of error likelihood performed by each node and shared with their neighbours alongside their position. These error likelihood estimations are then used to decide the next hop, with the neighbour which maximises the objective function being selected. They only consider static transmission ranges, however an enhanced version could take into account transmission range as well as location error probability in determining which node to forward to or even attempt to adjust the radio to reduce the probability of transmission failure.

In [61] a novel solution is presented for instances in which location information is not available amongst for nodes; virtual coordinates. Virtual coordinates allow nodes that cannot perform localisation to participate in geographic routing through the use of virtual coordinates which are used in place of real coordinates. Other approaches that utilise virtual coordinates in place of real location data include the use of continued adjustment of virtual coordinates as routing progresses [61]; however this approach is intended for static systems.

B. Beaconing and Updates

Location errors are often combined with problems arising from unpredictable mobility patterns, thereby further decreasing the accuracy of location data and decreasing the chances of calculating a rate of error (i.e. a node’s location error will be inconsistent due to dynamic mobility, and thus harder to predict/mitigate) [60]. In ad hoc and mesh networks there are generally no fixed base stations and node mobility can appear random [16]. Although mesh networks might make use of fixed mesh routers, typically client mobility is permissible. This can have a significant effect on topology (i.e. need for updates when node moves) and resource utilisation (i.e. a bi-directional link may become uni-directional due to one node moving) [16]. Mobility can lead to increased overhead due to the number of number of control messages that must be exchanged before transmitting data (possibly limited to multi/geocast systems) [62]. As well as leading to link costs of a path changing during utilisation [63].

The GPSR protocol was simulated in a series of experiments to determine the effects of mobility induced errors and [60] found that for all mobility models used when node speed was increased, packet delivery rate decreased and when beacon intervals were increased packet delivery rate again decreased. Interestingly, a positive effect of mobility was decreased sub-optimal hop selection when mobility was increased; however this was outweighed by the decrease in overall packet delivery resulting from increased node mobility [60]. The existence of loops was identified and lost links (where a supposed neighbour is outside of a node’s range, i.e. due to being previously within range and then moving out of it) as the two primary problems resulting from increased node mobility.

Although it is often argued that geographic routing protocols (which are generally localised) help reduce the effect of mobility as they do not rely on topology, if a node moves significantly, it must provide an update of some sort (whether to nearby neighbours or the entire network) when it moves significantly so as to avoid position information becoming stale. Going further, [16] argue that it is not enough to merely know that a node has moved and that nodes should also provide information on their velocity so at to allow for prediction of future location before updates are received. A common approach to updating neighbours on a node’s position or general state is that of beacons; packets which are sent either at regular intervals or when a specific event occurs to neighbouring nodes, these packets are sent in addition to normal data packets and therefore incur an extra overhead in terms of processing and transmission [65]. By the time a beacon is generated and disseminated, it is possible that the information it contains is already out of date and thus is of little or no use to the nodes that receive it [57]. The likelihood of information becoming out of date, as well as the effect outdated information will have on the network increases when beacons are sent at fixed intervals [66].

Figure 7 illustrates a scenario in which a suboptimal geographic routing decision is made as a result of out of date position information. In this diagram the sending node, and two neighbours (Neighbour A and Neighbour B) are shown. Neighbour A is shown twice to indicate its movement between location updates. According to the most recent beacon messages Neighbour A is located closer to the destination than Neighbour B, however since the last update Neighbour A has moved and is now further away from the destination than Neighbour B. Therefore when the sending node makes the routing decision it will believe Neighbour A to be in a better position than Neighbour B and forward the message to Neighbour A.

In [66] two protocols are proposed that aim to avoid the reliance on stale information caused by inflexible beaconing schemes. The first approach is based on a hybrid geographic and topology-based routing scheme in which nodes cache routes to previous destinations for future forwarding. In instances where no such destination is stored the current node will send a request message to 1-hop neighbours and then determine which of these is closest to the destination [66]. This is similar to basic greedy geographic routing (a fact that is acknowledged by the authors) with the notable difference being the use of cache and route request mechanisms as opposed to the conventional neighbour table. The main advantage of [66] is that by introducing the ability to send request messages nodes are not limited to choosing from
nodes they believe to be neighbours which could potentially help avoid local maximum situations (i.e. if a new node moves within the range of the current node), while the use of the cache retains some advantages of neighbour tables (i.e. ability to consider neighbours used in previous forwarding decisions) with a reduced maintenance overhead.

The disadvantages are the overhead caused by sending route request messages when a suitable next hop is unavailable (this is somewhat mitigated through the use of a response timer that aims to prevent replying node messages colliding), as well as the need for some cache maintenance (albeit less than would be required with a neighbour table) without the benefits of a full neighbour table. Additionally, a recovery strategy is proposed in [66] based on an expanding ring search in which a new search is performed with the maximum number of hops increased to 2 (i.e. immediate neighbours of the current node’s neighbours), and the backup period again being adjusted (to take into account the increased number of messages being transmitted). Nodes receiving a reply will drop the reply if they have previously received a reply from the same node or have received a reply from a node in a better position than the current node; the aim of doing so is to reduce overhead. The process of increasing the maximum number of hops continues until a defined maximum is reached. Although increasing the number of hops increases the likelihood that a suitable intermediate node is found it also increases the overhead in terms of both communications and computation (it is unclear whether nodes from previous rounds will forward a request without considering their own position (i.e. they have already been ‘ruled out’) in which case it is possible that have since moved and are now suitable, or they will check their own position and risk redundant computation if they have not moved) despite attempts to limit the number of transmissions.

The second approach by [66] introduces what the authors describe as ‘reactive beaconing’ in which beaconing commences when a node overhears a message from its neighbour while reducing the number of neighbours considered to those within 1-hop of the current node (i.e. only immediate neighbours). Beacons are only sent by nodes hearing a request or data packet after a period of silence and the sending time is delayed by a random time less than a set maximum in order to reduce both the total number of beacons and the number of beacons being sent simultaneously. If a node is due to forward a packet and the length of time between the current time and the last request being sent is greater than or equal to a minimum beacon interval then the current node sends request message to its 1-hop neighbours to start beaconing and then delays forwarding until 3X the maximum sending time so as to allow time for its neighbours to reply. The advantage of this approach is that requests are restricted to 1-hop neighbours dramatically reducing the number of requests, while limited beaconing allows nodes to keep track of their neighbours with lower costs than conventional beaconing. Disadvantages include the limiting of requests to 1-hop neighbours (thus increasing the chance of a local maximum situation being encountered), as well as the cost associated with (limited) beaconing and the lack of a recovery strategy when no 1-hop neighbours are closer to the destination than the current node. The adaptive parameters of this approach are the beacon and timeout interval.

Although most geographic routing protocols do not require knowledge of the entire network topology, several still rely on partial information about neighbouring nodes (usually those within direct transmission range or a hop-count limit); however where nodes are mobile there is a risk that this information will be out of date when it is needed. While beaconing is often seen as a means of mitigating mobility-induced routing errors through the transmission of regular updates this introduces an overhead and can still lead to stale information being used (for instance in cases of high mobility). This has led several authors such as [67] to adopt approaches in which update frequency is proportional to a node’s mobility. For instance, there is no need for a node that remains static to constantly update its neighbours. However when a node is highly mobile but does not send an update until the update timer has expired it is possible that the node will have moved significantly from its previous location. This could have ramifications for other nodes who still believe it is at the same location (i.e. another node may calculate the node that has moved to be nearest to the destination, however, the node may have moved into a position where it is no longer the best candidate).

The scheme proposed by [67] features two main components; the first is a mobility prediction module that allows a node to predict when its last positional update will be out of date and send a new update message to neighbouring nodes. In addition to tracking their own mobility, nodes also track the mobility of other nodes (based on their last update of position, direction, and velocity) in order to determine when a node will move outside of their transmission range. This allows nodes to keep their neighbour table relatively ‘fresh’ by removing nodes that are expected to move outside of their transmission range. This approach [67] is in contrast to that of both ‘conventional’ approaches to updating in which periodic
updates are sent, or the approach used by systems such as GLS [3] in which a node reactively informs other nodes (in this instance location servers) that it has moved away from a particular region. While it is possible that a node could erroneously predict that another node will move outside its transmission range (i.e. if a node suddenly stops and turns back) the node in motion should continue to send update messages, which will presumably lead other nodes to add it to their tables again. Although proactive mobility updates may produce an increase in updates being sent by highly mobile nodes, they will not only significantly decrease the number being sent by static nodes, but also ensure that neighbour data is kept up to date as possible which should decrease the overhead in finding a route. Therefore, although the mobility prediction used by [67] may not be the most accurate or desirable, the basic principle that update frequency should depend on mobility is sound.

The approach used by [67] is not the only one to utilise mobility predictions, on the contrary, several other researchers have made use of some form of mobility prediction in order to counteract movement-induced dynamics. For instance, a system in which neighbour locations are predicted prior to transmission using the most recent beacon messages is proposed in [60]. This allows a sending node to not only determine the expected position of a neighbour it is about to send to, but to determine whether or not that neighbour is expected to be within transmission range. Using this information, a node can then decide to send to another neighbour if a particular neighbour is expected to be outside of its transmission range or no longer an optimal next hop. As with the system described by [67] it is possible that an estimation error could cause a node to incorrectly predict a neighbour as being outside its transmission range, or a suboptimal next hop. Therefore, it is important that updates are sent frequently by nodes in motion, so that any inaccurate predictions can be rectified quickly. However, no mention is made of the frequency at which beacon messages are sent before simulation, therefore it would be interesting to see how this approach works with an adaptive beaconing system such as that used by [67]. In addition to neighbour estimation, [60] also describe a destination location prediction method in which all nodes involved in the sending process check their lists to see if the destination is predicted to be within their transmission range, and if so attempt to forward the message to it directly. This would be particular useful in instances where the destination may have moved, so that instead of relying on the destination location in the packet (which may be out of date) a node would first check its neighbour table to see if it could send the message directly to the destination.

Whereas previous approaches have seen node mobility as a disadvantage and their proposed protocols have therefore treated it as such, it can be argued that mobility could be used as a positive factor in deciding which neighbour to forward a packet to [68]. The work of [68] is motivated by the desire to overcome the weaknesses of greedy forwarding without using planarization. To do so they have devised a scheme in which a node will first attempt to forward the packet using normal greedy mechanism (i.e. selecting the node located nearest the destination) and use what they describe as motion potential if this fails. Using forward potential, nodes that would normally be ignored using standard greedy forwarding (for instance nodes that are located further away from the destination than the current node) are considered if it is possible that they will move within transmission range of the destination. This calculation is based on both direction and velocity of movement, so that nodes moving in the opposite direction are automatically ruled out, and nodes that are moving towards the destination (determined by calculating if a node is travelling at the critical angle that will lead it to the destination) are prioritised based on speed (i.e. which one will move within transmission range of the destination first).

To prevent the possibility of a source loop in which the chosen node forwards the message to the source when it moves within the source’s range, a timer based on how long it is predicted the forwarding node will take to reach the source is utilised, so that the forwarding node cannot forward the packet until the timer has expired. If a non-source node receives the packet after forwarding it to another node (i.e. because it has moved ahead of the node it forwarded the packet to) this is not considered a loop by the authors who term it ‘reuse’ [68]. Although the theory and approach are both novel and interesting it would seem as though there are several issues; the first is a question of how reliable the prediction model is? For instance, could it not be possible that a node that is making forward progress and is given the task of carrying the packet would suddenly alter its route so as to head away from the direction of the source? Although this is not explicitly addressed in the original paper, it could be argued that since all forwarding nodes are able to forward the packets themselves (as in basic greedy routing) in such an instance a node may pass another node who is now nearer to the destination than itself and forward the packet to that node. Of course, this would not cover ‘extreme’ circumstances in which a node suddenly changes direction after receiving the packet and moves away from the network altogether, however, this could be averted by including a mechanism that allows a node to determine if it will move outside of the network altogether and pass the packet off before doing so.

[62] presents a route maintenance algorithm known as Dynamic Route Maintenance (DRM) which aims to use mobility predictions for both adjusting beacon intervals as well as determining how long information stored in a node’s routing table is valid for. As with previous approaches, the beacon interval is a function of a node’s velocity (more specifically it’s velocity over a period of time, to account for the fact that a node’s velocity may not be constant). The routing table management is performed along similar lines with each entry being assigned a link expiration time (LET) based on both node’s positions and velocities in order to reduce the chance of out of date or irrelevant (i.e. when a node moves out of the current node’s range) being stored. Both mathematical and simulated analyses of DRM’s performance against a static beaconing interval algorithm were provided with both analyses finding DRM to provide higher levels of
reliability (i.e. more packets delivered or less position errors), although an increased beacon overhead was noted for DRM when mobility increased, yet when low levels of mobility were simulated DRM had a lower beacon overhead than static beacon intervals. This suggests that an increase in beacon overhead in periods of increased mobility is a reasonable price to pay for higher reliability and lower beacon overhead in periods of low or no mobility.

The work of [69] uses mobility prediction methods on top of the LAR protocol to estimate link and route duration metrics which are then used to evaluate routes with a longer route duration being preferred to a shorter one. Link calculations are based on the velocity and direction of both nodes with the assumption that communications are available between two nodes whose distance is within transmission range. The total route duration is then taken to be the shortest link duration. In addition to using geographic information for route discovery and evaluation, [69] also uses it for proactively repairing broken links, through the use of a maintenance table which includes path information and replacement node IDs. When nodes on a path send messages towards the destination they record their own position and movement information and broadcast this to their neighbours for use in partial reconstruction. Partial reconstruction is activated when a node determines that a link will break (due to expiration of the link duration) and the information obtained from other nodes sending their information is used to determine a request region in which route request messages are sent. The resulting replies are then used to populate the maintenance table and if possible a suitable backup node is found, if such a node is not found then a route error message is propagated and route discovery starts from scratch. An obvious disadvantage of this approach is that it is only able to perform route maintenance on links that are expected to expire and is unable to handle more dynamic instances of link failure (i.e. due to battery depletion or even instances where a node moves in an unpredictable manner). Two other prediction schemes of interest are those used by [64] and [16], however these will be discussed in the section covering QoS, as although they both make use of mobility prediction mechanisms, the emphasis of these works is on QoS rather than mobility management.

Moving away from prediction-based systems, an interesting alternative to previous mobility management schemes that rely on adaptive beaconing is to forgo beaconing at all. Beaconless (contention-based) routing was mentioned earlier in the context of planarization alternatives [26]. However, given both the potentially high overhead of beaconing as well as the possibility of information obtained from beacons going out of date, it is worth considering the viability of an approach which avoids beacons altogether. It was mentioned earlier that routing takes place through the broadcast of an RTS message and based on this message nodes which receive it bid to be the next hop. Typically RTS messages will contain the destination and source location so that a timer value can be calculated. Upon receiving this message the node will initiate a timer using the calculated value in the message and when the timer expires a contention message will be sent to the source. The timer value that a node receives depends on the advance of that particular node (the difference in distance between the current node and target, and the candidate node and target) with nodes with a higher advance receiving a shorter timer value [26]. Upon receiving the CTS reply from a candidate, a node will then send the DATA packet to that node, and all other candidates that overhear the CTS will cancel their own timers. The concept of beaconless routing was first proposed in [70] and [71] which [26] observe exhibit strong similarities. Similarly, the work of [72] considers the problem of beaconless multicasting while [73] focuses on using beaconless geographic routing in WSNs that utilise energy harvesting.

Although beacons can be observed to have strong advantages attached to them, as this section has discussed they also have several drawbacks. Therefore, the basic premise of beaconless routing is somewhat interesting and novel. Clearly by avoiding beacons altogether, beaconless routing significantly reduces the overhead caused by periodic or mobility-based beacons, - with [26] claiming that beaconless routings broadcasting cost is lower than typical beaconing costs. Similarly, the possibility of information obtained from beacons going stale is also avoided, as nodes do not store any information about neighbouring nodes. However no mention is given in either [26] or [70] as to whether nodes are able to ‘learn’ about neighbouring nodes or must always follow the process of broadcasting RTS messages. If a particular node is sending packets to multiple destinations then it may receive CTS messages from several different nodes depending on where the destination is. In a typical static WSN it is likely that (exempting node death) a node which replies first to a CTS for a particular destination is always likely to be the best located for that destination. Therefore, it is perhaps worth considering the possibility of memorising the nodes that have responded to CTS messages for particular destinations, and storing them in a preferred nodes table. These nodes could then be contacted the next time the source wishes to send a packet to a particular destination, thus saving the cost of a broadcast message. However, as it is possible that one of these nodes may have died a timer could be put in place so that if no CTS from that node is received within a certain period then an RTS is broadcast. Similarly, it is not yet known whether beaconless routing is suitable for mobile networks or only works well in static networks. If beaconless routing performed well in mobile scenarios then it potentially has several advantages over beacon-based protocols; namely eradication of the problem of out of date information as well as reduced message costs. Therefore, although early results from WSN simulations are mostly positive it remains to be seen whether beaconless routing is a valid technique for real-world WSNs or MANETs.

Finally, [74] presents a protocol known as rope ladder routing which uses multiple paths to provide reliability in the event of a single, or multiple node or path failures. The basic idea behind rope ladder routing is indicated in its name; instead of having either a single path with backup nodes or
multiple disjoint paths, rope ladder routing creates a structure in which two paths (a primary and backup path) are created and at each hop a link between the two paths exists (i.e. the link between the two paths is the rung). The aim of rope ladder routing is to improve both reliability (through maintaining multiple paths to the source) and speed of path change (by having all nodes know not only the next hop on their path, but what node is their backup/primary). When the next hop is unavailable (either through node or link failure) the current node will send the message to the backup node (i.e. along the rung) who will route it down the backup path, however because rope ladder routing uses position information it is possible that the packet could be sent from the backup path to the primary path (i.e. if the failure on the primary path was limited to one or a small number of nodes). This also means that if a failure is encountered on the backup path, it may be possible to switch back to the primary path, thus further enhancing reliability.

Route discovery is conducted by nodes identifying the two neighbours nearest the destination (using Euclidean distance) and assigning them as primary and backup respectively. Each node knows the identity of the other (so as to allow path switching at the node level) and the backup node also knows the identity of the next hop on the primary route and finds a two-hop minimum route to it. This is repeated at each stage until the destination is reached on both sides. There may be a problem in which it is impossible to find a two-hop minimum path between a backup node and the primary node’s next hop. When this occurs a backtracking procedure is put in place the primary node attempts to find a new next hop to which the backup node will attempt to find a two-hop minimum link to, if a two-hop minimum link between the backup node and next primary node is still not found, then the algorithm will backtrack to the previous primary node and so on. The process could also be too strenuous (both in terms of time and resources) and allow for the creation of partial rope ladder structures (i.e. with missing ‘rungs’) once a certain backtracking threshold (in terms of number of attempts) is reached, thus allowing for a trade-off between reliability and time/resource consumption. Even with this backtracking threshold, it seems as though the system is potentially capable of consuming a large number of resources by attempting to find two paths to the source and then links between the two paths, which would surely result in a potentially large number of messages being exchanged (thus consuming bandwidth and energy) and significant computation time. However, the primary advantage of increasing reliability and decreasing the amount of time taken to switch from primary to backup link (which the authors argue will reduce the number of packets lost in the event of node failure) may be acceptable trade-offs.

So far this paper has looked at issues that are of importance to ad-hoc and mesh networks; from general protocol design issues to security threats and problems arising from node mobility. Just as node mobility is an inevitable part of most ad-hoc networks another common feature that can have significant influence on network operation is power consumption. As ad-hoc network devices are often battery-powered the potential exists for some or all nodes to experience failure, this has provided motivation for researchers in ad-hoc networking to pursue means of reducing energy consumption at the routing layer. The following section provides an overview of how geographic routing can be used to reduce power consumption.

V. POWER CONSUMPTION

Monitoring and reducing power consumption is an important issue in wireless networking (particularly the areas of MANETs and WSNs) as a large number of potential devices (laptops, mobile phones, PDAs, sensors, etc.) are battery-powered [75]. This means that protocols that do not take into account power consumption and energy constraints can cause rapid loss of power. This can in turn result in multiple nodes running out of power and thus reduce the overall network lifetime.

Both link channel and battery power resources are limited, therefore protocols that rely on flooding (in which all or a large number of nodes relay a message through the network) are costly and unsuitable for energy-constrained devices [65]. Limited resources and mobility/other dynamics lead to utilisation of multi-hopping (with the exception of some mesh network architectures) as a means of overcoming physical distance and reducing resource usage. However next hop selection is generally based on some criteria that does not take power consumption into account (i.e. in greedy geographic routing the next hop chosen is the one physically nearest to the destination) thus the route that ends up being taken may not be the most efficient in terms of power consumption. End-to-end energy consumption in multi-hop networks increases as a function of the number of relaying nodes and energy consumption of individual nodes (dictated by distance between hops) [79]. Often a device’s radio is one of the biggest consumers of power, and so a large number of power-saving approaches focus on the radio. One of the most common ways of saving energy is to reduce radio transmission power, however this has the undesirable side effect of limiting transmission range – see weaknesses of UDG model. A major problem with radios in terms of power consumption is that they can consume energy when listening or idle in addition to when sending or receiving [75]. Therefore the only way to guarantee reduced power consumption from a radio is to turn it off which is not always desirable.

There are some who advocate adjusting radio power levels in order to avoid undesirable effects such as path-loss and multi-channel fading. The work of [80] combines geographic routing with a scheme for adjusting radio power with the aim of reducing signal-to-noise ratio. Although they briefly mention the possibility of combining their protocol with power-aware metrics in order to conserve battery lifetime. Therefore, while the work may focus on decreasing radio noise, as the approach focuses on adjusting radio power levels it has potential to be adapted for the explicit purpose of reducing power consumption. As limited energy is a prominent issue in many WSNs, it is perhaps not surprising that most of the research into location-aware energy efficiency
has come from this field. One of the earliest attempts at combining position and power was presented by [1] who describe several protocols for either decreasing energy used per packet or maximising the total node lifetime. Although these protocols are localised, they exchange control messages which adds energy overhead, additionally the authors state testing was only conducted in static networks with high levels of connectivity. Thus they are potentially not applicable to dynamic, highly-mobile networks. This work was then used as the basis for future augmentations that would allow it to provide an energy-efficient solution that also guarantees delivery [19].

In [76] the work of [1] and other protocols is discussed in the context of designing WSN protocols with the aim of developing a framework for designing routing protocols based on the optimal ratio of cost (i.e. power or some other metric) to progress (i.e. physical distance). Building on the work of [1], the amount of power required to send a packet can be used as a cost metric, so that neighbours can be selected on the basis of which ones will minimise the power cost of sending a packet. It is stated in [76] that power-aware protocols often suffer from the problem of actually decreasing certain nodes power levels and that a better aim is to maximise the lifetime of a network. The reluctance [77] and power-reluctance measures [79] are given as examples of methods of maximising network lifetime. In [77] reluctance is a measure of a particular path’s desirability, i.e. paths that are more desirable have less reluctance attached to them. While in [78] power_reluctance is described by [76] as being parameter-free and performing well in experiments.

Another approach that considers distance and cost is Normalized Advance (NADV) which determines the best link through a combination of link cost and physical distance. NADV allows the possibility factoring in energy consumption as one type of link cost [81]. Before calculating the cost of a link, the distance known as (ADV) is first calculated by the current node’s destination from the candidate node’s destination. The ADV is then used as a tradeoff with the cost metric, to determine the neighbour which best satisfies the distance and cost constraints. Although greater consideration is given by the authors of [81] to packet error and delay estimation a metric for determining energy cost is given. Using this simple metric (which assumes the required transmission power is given), the total energy cost of sending to that node can be calculated. This is then divided by the ADV metric to give the combined distance/cost metric for that neighbour. The authors then simulate NADV alongside SP-Power and argue that although simulated results show very little difference between the two, NADV has the advantage of calculating power consumption of the link to the next neighbour whereas SP-Power estimates the path loss exponent, which they argue is a complex step in real systems.

Energy and location aware routing is also considered in [82] where two energy-aware geographic routing algorithms are introduced: optimal range forward (ORF) and optimal forward with energy balance (OFEB). In ORF the algorithm estimates the best location for the next hop and then selects the neighbour closest to this location. ORFEB is more explicitly energy aware and calculates which neighbour within the optimal range is best suited in terms of both energy consumption and energy distribution. It is claimed that ORFEB seeks to minimise not only per-hop energy costs but also equally distribute energy consumption amongst nodes. Simulation results show that ORFEB outperforms both ORF and conventional geographic routing in terms of reliability, throughput, network lifetime and per-packet energy consumption.

In [83] a location-aware multicast (not to be mistaken with geocast) protocol has been designed that aims to achieve energy efficiency through the use of balanced bi-directional multicast trees and zonal routing. A potential drawback of using a multicast tree is that (as with any tree structure) the root node is liable to perform more tasks than other nodes and therefore has the potential to expend more energy and die sooner than other nodes. To reduce the impact of root death, a backup route, responsible for sharing some duties with the primary route, as well introducing a proactive power monitoring function is created. Zones are defined in terms of hops, so that a particular node’s zone will contain all nodes that are within k hops (where k is a pre-defined number of hops). All nodes maintain and share status information with neighbouring nodes in their zone via unicast connections. It is argued by [83] that learning a local topology is less expensive than learning an entire network topology, and that most destinations are located close to the source (the latter being a potentially contentious claim!). The zone structure is then used to construct and maintain multicast trees through a series of messages first at unicast zone level then network broadcast level [83].

Location information is primarily used for backup root selection and multicast routing, where packets are forwarded in a general geographic direction. Therefore [83] combines both virtual tree and real physical locations for multicast routing. The power saving approach largely focuses on proactively detecting nodes who energy levels are about to fall below a certain threshold. These nodes then have their links removed from the tree. The authors claim that their approach is more energy efficient than mesh-based multicast protocols which waste energy through broadcast flooding [83]. They also claim that their approach is comparable (in terms of energy consumption) to per-source tree multicast and standard shared-tree multicast. However, these comparisons are formed on the basis of the author’s analysis, of which no supporting evidence or even an indication of methodology used is provided.

Alongside power/energy consumption another important issue in ad-hoc networking is QoS. Although it may not seem as important as power consumption, QoS ultimately governs the quality of the end-user’s experience and so much time and effort has been dedicated to its research. The next section explores ways in which geographic routing and QoS have been combined.
VI. QoS

Although Quality of Service (QoS) is by now a well-established field in conventional wired networking, in ad hoc networking it is an emerging field with many challenges yet to overcome. Perhaps the most accurate comparison would be to contrast ad hoc network QoS with that of the Internet, as both network models offer best effort rather than guaranteed levels of service [84]. While multimedia content is predicted to be a large source of traffic on wireless networks, most MANETs only support best-effort transmission [85]. This as well as other possible differences between wired and wireless QoS standards/metrics (i.e. connection time on path between source and destination) should be considered [65]. Resource management is a central part of many conventional QoS approaches, but in ad hoc networks must be conducted in a distributed fashion as there is no management infrastructure in ad hoc networks (although mesh routers could play this role in mesh networks [84]). In mesh networks, users are often assigned to the nearest gateway (node connected to the Internet), however, this can often lead to several gateways being congested and offering a poor quality of service while others are relatively underused [86].

Similarly, communications in WSNs are not typically end-to-end, with a large volume of traffic often being directed to a central sink node. As most QoS approaches (including those of ad hoc networks) focus on providing quality in terms of end-to-end links they are often inappropriate for direct application in WSNs. This can also be a problem for non-WSN geographic routing protocols which also do not construct conventional end-to-end links and are highly localised, generally only taking into account local rather than global information. The lack of localised a GPS-aware algorithm has also been cited as a major stumbling block in geographic QoS [65].

Typical QoS parameters include; bandwidth, delay, jitter (delay variance), and packet loss (potentially high due to wireless radios) [87]. Due to the uniqueness of ad hoc networks, several researchers have proposed additional/alternative metrics and characteristics for ad hoc QoS. For instance, [88] states that the following are desirable properties for a QoS-supporting ad hoc routing algorithm:

- Robustness and ability to degrade gracefully as mobility increases
- Local route computation (i.e. no need to store global topology)
- Chosen route should be able to maintain request bandwidth level for flow duration
- Broadcast should be avoided & route selection conducted with minimal network overhead
- Nodes should be able to utilise connections quickly

It is interesting to note, that of these requirements most geographic routing protocols would claim to address the first two, however most geographic routing protocols do not rely on dedicated routes and so it is unlikely they could fulfil the last three. As a result of the increasing profile of QoS in ad hoc networking, attention has turned to means of using geographic routing (and geographic information in general) as a means of solving QoS problems, although there have only been a small number of approaches (seven such approaches are reviewed in this section) that focus primarily on QoS.

The approach described by [64] claims to be the first ad hoc routing protocol that is both localised and combines GPS data with QoS requirements. Their approach uses location data to determine QoS metrics and then uses Depth First Search (DFS) to select appropriate nodes. The basic DFS routing algorithm has nodes mark themselves as either grey or white depending on whether they have received a message (grey nodes have received the message). This approach has been observed as somewhat similar to greedy forwarding, in the sense that each node attempts to forward the packet to a node closer to the destination than itself [20]. Similarly, the addition of memorising the node which the packet was received from, and returning it if no suitable neighbour is found can be seen as comparable with some backtracking recovery methods. As a node will only forward to another node if that node is nearer the destination than itself (unless it is returning the message to the node it received it from) the assumption that this approach is loop-free can be made.

To adapt the standard DFS algorithm for QoS, delay should be considered as proportional to hop count, and thus it is imperative to minimize hop count (as far as possible, given the constraints of ad-hoc networking) [64]. To reduce bandwidth usage, the bottleneck model in which nodes that cannot meet the bandwidth demands are removed from the graph is used. DFS is then performed on the resulting graph to find a QoS-optimal route. Memorization is extended to include both previous and next node on an established path, and any node that returns or rejects a packet is removed from the path. A new parameter known as connection time is also introduced. This is defined as the estimated lifetime of a connection between two nodes and is based on speed and direction of movement (obtained through updates sent to neighbours). Using these updates a node can then determine when a neighbour is likely to move outside of their transmission range and from this the connection time can be determined. Connection time can be used in a similar way to bottleneck bandwidth so that nodes that are unable to provide a suitable connection time are removed from the graph. If no such paths are available, an alternative method of splitting the data into fragments and determining separate connections with suitable connection times for each fragment can be employed. Note that the model used for determining connection time does not account for interference or obstacles; thus the connection could end earlier than expected i.e. if one of the nodes becomes blocked by an obstacle, but is still within the ‘ideal’ range of the other node. Similarly, it is possible that a node could move outside of this transmission range only for a very brief period of time, and then move back inside the other node’s connection range. In such an instance, the time spent outside the connection range may be negligible, but the connection might still be rejected if the estimated connection time is deemed insufficient. The analysis in [64] found that path length of DFS-QoS was at most 1.34 times longer than
that of shortest path, with DFS-QoS having the advantage of localisation of over shortest path. The authors also state path discovery costs are reduced while the flooding rate (which they take as a measure of overhead) is never above 5. Proving DFS-QoS is a practical and desirable routing protocol.

Also concerned with bandwidth is the work of (89) in which a combined distance, bandwidth, and power consumption metric for admission control in WSNs is proposed. The proposed QoS metric is a combination of two components; available bandwidth conditioned by progress and consumed energy ratio. The former of these is a combination of progress and estimated bandwidth. Progress is defined in terms of degree of divergence the proposed route will take from the (imaginary) line between the source and destination, and the physical gain (in terms of Euclidean distance) towards the destination that a chosen neighbour will bring [89]. The bandwidth conditioned by progress metric is then obtained by dividing the progress by the available bandwidth of a link (obtained from the MAC layer). The second part of the QoS metric is obtained by dividing the neighbouring node’s current level of energy by its initial value. The hybrid QoS metric then consists of a multiplication of both the progress conditioned by bandwidth and energy metrics by a priority factor, followed by their addition.

The QoS metric is used in performing admission control, when upon obtaining a route request packet the current node will calculate values of the QoS metric and forward the packet to its neighbour which has the largest total value [89]. When the route request reaches the source, the total path bandwidth is calculated and placed in a route reply packet to be sent to the original source. After the reply packet has been sent, the destination then generates a new route request packet in the direction of the source. Upon receiving the reply and request packets, the source averages the bandwidth values of both to determine whether the path is appropriate. Although the proposed protocol places a strong emphasis on bandwidth it does not appear to consider other factors such as reliability or delay. However, it could be argued that by trying to maximise bandwidth (while minimising energy consumption), delay and (to a lesser extent) reliability will be improved. Similarly, it can be observed that the energy metric is somewhat simplistic and does not take into account the actual costs of transmission between nodes, only the ratio of current to initial energy. In spite of these criticisms, simulated experiments found that the proposed protocol outperformed AODV in terms of average delay and reliability. These results are interesting, as although AODV has been implemented in WSNs before, it is still considered in terms of conventional ad-hoc routing. Therefore, two possible conclusions can be made; one that the protocols is potentially of benefit to non-WSN ad-hoc networks, and secondly that by choosing an non-WSN protocol for comparison the authors are not providing an accurate means of evaluating its suitability for WSNs.

The work of [16] was briefly mentioned earlier in the context of location prediction schemes for mobility management. In [16] location predictions are used for the purpose of determining the level of delay that a path will incur based on the mobility of its nodes. An important part of the prediction mechanism is the update scheme of [16] which contains information relevant to QoS such as resource information and a parameter designed by the authors known as motion stability. This is used to determine whether an update has been sent routinely or because the node has moved. As with the adaptive beaconing schemes discussed in the mobility section, [16] contains a mechanism where nodes monitor their own mobility and if they deviate (in terms of speed or direction) significantly from their previous pattern an update message is sent to neighbouring nodes. Knowing whether updates were sent routinely or in response to changes in mobility can then be used to determine whether the motion and speed of a node varies constantly or dynamically. This in turn affects the ability of another node to accurately predict the node’s location. Nodes moving in a constant direction at the same speed are comparatively easy to predict, while nodes moving at variable speeds and/or directions are harder to predict. The authors claim that this mechanism is useful for reducing jitter, as nodes that are seeking to avoid a large degree of jitter will not choose a node with an unpredictable pattern of motion. This is similar to the connection time metric used by [64]. Both take into account the node’s motion (in terms of direction and speed) in order to minimise the chance of including a potentially unreliable node in the route. However, the main differences are that the metric used in [64] actively eliminates nodes that are deemed ‘undesirable’ while the metric used in [16] merely serves as an indicator and an undesirable node may be picked if no better options are available. The prediction mechanism is also used to calculate delay using the assumption that the end-to-end delay between two nodes will be the same as the delay between the two nodes when the last update was sent. The location and delay prediction mechanisms are then made use of by the dedicated QoS routing module which maintains two tables; the update and routing table. The update table contains the information obtained from all update packets in addition to a proximity list that for all nodes in the table contains all known nodes located within 1.5x transmission range of that node. Metrics for each route are; connection duration (in terms of battery life), maximum delay, and maximum jitter (connected to the motion stability metric, as discussed earlier). Nodes that are unable to satisfy the combined connection demands are excluded from the routing process.

Route discovery and admission control are performed locally with each node that receives a request packet performing a depth-first search to determine which (if any) of its neighbours is most appropriate. This would seem to contradict the earlier claim that all nodes maintain a complete network topology [16], as there is little point in doing so if route discovery and admission are then handled locally. The only possible reason for doing so would be to use the proximity list in order to restrict the initial request process to nodes within the proximity of the destination. However, given the chances this information could ‘go stale’, surely a better approach would be to use the position estimates in order to determine the general area the destination might be in and then
send messages to nodes located there? The protocol could then cut overhead by only sending positional updates to neighbours or nodes with which a relationship has been established rather than the entire network, an approach favoured by most geographic routing protocols which seek to cut the update overhead? Although the source wouldn’t be able to calculate the motion stability or expected delay in advance, it could place its desired levels of QoS in packets and then enable neighbours of the destination (and neighbours of the neighbours) to determine locally whether they could provide a route to the destination that fulfilled these requirements.

In [90] a combination of location information and an approach known as ticket-based routing are used to manage QoS constraints in static ad hoc networks. This approach uses a modified version of ticket-based probing in which the source node evaluates each of its neighbours to determine the most appropriate in terms of bandwidth, delay, and path cost. With tickets being issued based on this evaluation. Node’s with a lower level of delay will receive more tickets than those with a higher level). Intermediate nodes will then distribute their allocated tickets to neighbours they deem best suited to meet the route’s QoS demands. If none of a node’s neighbours can satisfy the QoS demands then that node will not issue any tickets. The advantage of ticket-based probing is the ability to achieve a high success ratio with a relatively low overhead [90]. The downside of ticket-based probing is that it requires the compilation and maintenance of a state model for the network, and thus is proactive leading to significant maintenance costs. Thus it would seem that when [90] mention a low overhead they are referring purely to the routing process, as it would appear that maintaining the state model would result in a high overhead as nodes would be frequently updating each other. To overcome the disadvantages of ticket-based probing while still maintaining the advantages, [90] propose to combine a modified version of ticket-based probing with local information (referred to as Location-aided Ticket-based Routing (LTBR)) to allow for the use of ticketing without the need for creating and maintaining a state model. In LTBR tickets contain both QoS and previous node details (thus allowing the destination to reach the source by traversing the path backwards) and are generated by the source and sent to suitable neighbours, unlike conventional ticket routing intermediate nodes are allowed to send more than one ticket even if they only receive one themselves, the authors arguing that this approach allows for the exploration of more paths therefore resulting in a higher possibility of finding the appropriate QoS without incurring the high overhead of flooding [90]. The location-aware aspect of the protocol is encapsulated in the progress-over-cost metric used in the 1-ticket variant of LTBR in which the geographic progress between the current node, the destination, and the current node’s neighbour is evaluated against the QoS cost of that path, with the progress-over-cost method being used as a means of selecting which neighbour(s) to send the ticket(s) to. In order to avoid situations in which a node either fails to select or selects a sub-optimal neighbouring node when a better option is available but unknown to the current node (a problem somewhat similar to the local maximum problem in greedy routing) all nodes memorise the neighbour set of the previously traversed hop in order to determine whether a path involving one of these nodes has a better cost than a path involving the current node and one of its own neighbours. A 2-ticket version of LTBR is also proposed which is described the authors as a hybrid of single path routing and flooding in which nodes at all stages are allowed to send a maximum of two tickets, the aim of which is to improve the probability of finding a suitable path in sparse networks [90].

It is interesting to note, that unlike some traditional models of QoS (i.e. the DiffServ model) none of the previous approaches have provided any form of traffic differentiation, and have instead (with the exception of QoS-GRID) concentrated on reserving resources that meet QoS standards, and approach that has more in common with the IntServ model of QoS. An interesting approach to geographic QoS that differs from the above due to its use of a model that claims to be inspired by the DiffServ model is introduced in [85]. In the proposed protocol, traffic is divided into two classes; multimedia and non-multimedia. Unlike DiffServ, however, classification only really affects route discovery (multimedia traffic is allowed to use a more reliable form of flooding than ordinary traffic), and once routes are established all traffic is treated exactly the same. Similarly, the geographic element of this particular protocol is very small and is nothing more than a backup method in which nodes attempt to determine whether a secondary route is still available by determining the distance between itself and the next node should the primary route fail.

The downside of this approach is that the source node is required to store not only the complete primary route to the destination, but also an alternative, while nodes on both the primary and alternative route are themselves required to store alternative routes from themselves to the destination. While this approach does not itself claim to be a geographic routing protocol (rather, it seems to be a normal source-routing protocol augmented by positional data), increased use of geographic data could drastically reduce the amount of information being stored and computed. The use of basic traffic classification inspired by the DiffServ model is interesting, and potentially beneficial in providing priority to traffic which needs it, but a more localised approach is surely preferable? The original DiffServ model is itself comparatively localised, with individual routers classifying traffic based on some criteria and then treating it accordingly. Similarly, geographic data could also be used to eliminate the cost of route discovery and maintenance by using a pure geographic system in which packets are routed towards the destination (although not necessarily in greedy fashion) with each node that receives the packet using a classification scheme that gives priority to multimedia traffic. The approach in [85] is the only protocol to combine geographic data with some form of traffic differentiation, and one of the few ad-hoc routing protocols based on the DiffServ model. Therefore, although it may seem somewhat inefficient when compared with other protocols, it should nevertheless be considered...
interesting and a potential stepping stone for further exploration of the use of DiffServ in geographic QoS routing.

VII. FINDINGS

This paper has reviewed and discussed numerous papers in the area of geographic routing and the issues surrounding geographic routing in general. This section presents a brief summary of the findings from all previous sections of the report.

A. Geographic Routing Protocol Design Issues

It has been mentioned several times in this paper that two of the most prominent and popular approaches for designing geographic routing protocols are greedy forwarding and face routing, both of which have their own respective advantages and disadvantages. Greedy forwarding boasts efficiency and simplicity yet cannot guarantee delivery while face routing guarantees delivery in theory but is less efficient than greedy routing. This has resulted in several hybrid protocols that typically use greedy routing by default but switch to face routing when greedy routing fails, protocols such as GOAFR+ [12], PVCX [10], and GPSR [13] all fall into this category. While the idea behind these protocols is generally to combine the efficiency of greedy forwarding with the delivery guarantee of face routing potential drawbacks include the overhead incurred from switching to face routing and then back (most hybrid protocols will typically return to greedy forwarding once the local maximum has been recovered from), the reliance on reactive recovery techniques (i.e. waiting until the local maximum has been encountered rather than trying to avoid it), and the potential failure of these protocols to address the underlying issues with greedy forwarding and face routing. On the other hand, GOAFR+ boasts asymptotic and worst-case optimality while GPSR is frequently cited and used as the basis for developing other protocols.

Combining greedy forwarding and face routing is not the only way of resolving their problems; on the contrary several papers have focused on the development of improvements to both approaches independent of each other. In greedy forwarding the two (related) main problems are often considered concave nodes (nodes with no neighbours closer to the destination than themselves, thus a local maximum occurs when a packet is forwarded to them) and voids (areas of the network uncovered by nodes) and as a result research has tended to focus on resolving these issues. Approaches used include the memorization of ‘landmark nodes’ (the node where recovery mode is terminated) in GLR [22], spanning trees [25], a combination of real and virtual coordinates (as well as face routing) [29], and potential field (a technique used for avoiding obstacles in robot navigation) [23]. In addition to augmentations designed to avoid concavity or voids, a number of protocols have used basic greedy routing as the basis for other improvements (such as security or QoS), however, these will be discussed in the relevant sections, suggesting greedy forwarding is a popular approach to geographic routing. As stated previously, this popularity is most likely due to its simplicity and efficiency, despite its failure to guarantee delivery. It is interesting to note that of the protocols listed as augmenting greedy routing, only one of them used any form of face routing [29], despite the popularity of face routing both in hybrid greedy-face protocols and on its own it contains a number of drawbacks as well as advantages.

In addition to being less efficient than greedy forwarding, face routing often experiences problems in practice, relating to the use of planarization and the Unit Disk Graph model. These problems arise from the Unit Disk Graph’s assumption that all nodes have equal transmission radii, which can in turn lead to problems during the planarization process such as links being removed that cause graph disconnection. Thus, due to errors such as interference and inaccurate location estimation face routing algorithms may fail in practice despite their theoretical delivery guarantee. As a remedy, several alternatives to the Unit Disk Graph have been proposed that aim to provide a more accurate model of wireless networks including the quasi-Unit Disk Graph [36], and a combined Unit Disk Graph/Signal Interference plus Noise radio model (SINR) [37]. It is interesting to note that both of these approaches have used the Unit Disk Graph as their basis rather than attempting to create or adopt another model. This is perhaps a sign of the popularity of the UDG, largely due to its simplicity and theoretical effectiveness. However, given the failure of the UDG to take into account the possibility of differing transmission radii, network dynamics (such as node death or movement), and position errors its use both as a conceptual model and an underlying graph for planarization to be performed upon cannot be relied upon to produce protocols that guarantee delivery in the real world. Therefore, either modifications such as those described above or the development or adoption of new models (either through the creation of a dedicated model or through the use of an existing graph/model that more accurately resembles wireless networks) are necessary if protocols that rely on graph planarization are to be continually used and developed. In addition to improving the underlying model, research has been conducted on the planarization process itself, with the development of the Cross Link Detection Protocol (CLDP) [33] which probes links to determine if it is safe to remove crossing links and was deployed alongside GPSR in simulated and practical experiments.

Another important aspect in geographic routing protocol design is that of dimensions; or more specifically whether the network is two-dimensional (i.e. a flat surface where height difference is negligible) three-dimensional (a hierarchical area such as a hill or large building where there is significant variance in height as well as longitude and latitude). While most geographic routing protocols are by their nature two-dimensional and use two-dimensional coordinates) recently the need for protocols capable of operation in three-dimensional (3D) environments has arisen. This is due to the perceived inability of standard (2D) geographic routing protocols to adequately cope with the demands of 3D networks; for instance, the Unit Disk Graph assumes a two-dimensional Euclidean space, while face routing protocols
which rely on planarization techniques will naturally fail as the faces/perimeters created during planarization cannot be applied to three-dimensional surfaces [42]. This has in turn motivated research into the area of 3D geographic routing; an interesting development has been the use of the Unit Ball Graph (described as a 3D version of the Unit Disk Graph) as an underlying model for 3D routing [44], while the possibility of modifying 2D routing protocols by projecting certain nodes onto a plane (in effect a virtual plane) thus allowing face routing to be applied to a non-planar terrain [37]. Despite these advances, 3D geographic routing (and 3D routing in general) still remains an emerging research area, however with the emergence of cheap wireless sensor devices it is expected that research in this area will grow (as a result of their deployment in 3D environments) while conventional ad hoc and mesh networks could also benefit from greater research into 3D routing (in order to allow for their deployment in large buildings, large urban areas with varying terrain, remote communities, etc.). Therefore, it will be interesting to see if such research continues and whether it will focus on adapting existing techniques and models for 3D environments (as much of the previous work has done) or will focus on developing new models and approaches.

B. Geographical Routing Security

It is generally accepted that security is of importance to all but the most trivial of networks, however wireless networks and in particular wireless ad hoc networks exhibit an increased degree of susceptibility to malicious behaviour. This is largely due to the nature of the wireless medium which allows for packet-snooping techniques to be deployed. However ad hoc networks are particularly vulnerable to this as it is difficult or sometimes impossible to deploy sophisticated countermeasures such as cryptography and authentication which can be used in commercial wireless networks under centralized control and coordination.

Geographic routing protocols present a novel vulnerability to security and privacy in which malicious nodes can identify a node based on its physical location due to the use of such information for routing. This can have serious consequences as nodes that are able to track a node or its user’s movement can in turn determine their physical location at a given time or create a profile based on their activities/movements for use in both cyber and physical crime. As geographic routing protocols rely on positional data for routing it is impossible to remove this element entirely while attempts to reduce the accuracy of geographic information can lead to undesirable routing performance of even complete failure. This dilemma has led many researchers to design protocols based around anonymisation (often featuring pseudo-identifiers which expire after a certain time), in order to decouple the node’s identity from its physical location. The protocol introduced in [50] use pseudo-identifiers alongside authentication and cryptography to separate node’s identifiers from its location data as well as placing a ‘trapdoor’ that only the intended receiver can decode in each message so that it can determine that it is the intended receiver. [17] place emphasis on authentication servers which are trusted nodes who provide ordinary nodes with public and private keys in exchange for their temporary ID and position data, however the reliance on servers brings into question issues related to the reliability of the servers (i.e. if they are maliciously compromised or fail for some other reason) as well as performance issues (possible bottlenecks). [2] omit the exchange of position information entirely, instead favoring a contention mechanism in which nodes compete to be the next hop in a route based on a destination reference point (an area in which the destination node is located, but not its exact position thus leaving malicious nodes with a large area to search). Anonymity appears to be the most popular approach to security taken by geographic routing protocols with other approaches which focus on anonymisation including [15,] [51], and [52]. The latter two both using some form of anonymity zone/region; an area in which a target node lies and which is used instead of the node’s exact destination (similar to the reference point used by [17]).

In addition to papers focussing on anonymity, another significant area of security-related geographic routing research concerns the possibility of falsified position data. Falsified position data can be used by either malicious nodes (i.e. pretending to be the destination or a next hop in order to intercept a packet) or lazy nodes (nodes wishing to conserve their own resources might state a destination that is farther from the destination than other nodes in order to avoid participation in forwarding). Within this category, an interesting strand of research is that of secure location services; either through the modification/augmentation of an existing platform such as GLS or the creation of a new one entirely. An example of this type of work is that of [54] who modify GLS to facilitate encryption of messages between nodes with the location servers being responsible for key provision. However, it is not explicitly stated whether location updates are also encrypted therefore although the possibility of message interception/tampering is reduced/mitigated the possibility of false information being provided does not appear to have been addressed. GLS is also used as a base for modifications by [49] again focusing on the use of authentication to prevent message tampering or interception while ignoring the possibility of false information being uploaded. However it does prevent malicious nodes from pretending to be other nodes and uploading their own position data in place of their target’s real position data. Key distribution is another area relating to security that can benefit from the use of position data as [56] demonstrates through the use of position data to reduce energy consumption through the creation of clusters in which nodes are able to overhear messages sent to other nodes thus reducing the number of transmissions required.

C. Mobility and Location

Two of the most significant factors affecting geographic routing protocols are node mobility and location accuracy. While the former also affects non-geographic routing protocols, the latter is a problem only when location is of use...
to the routing protocol. Location inaccuracies are widespread in GPS and non-GPS (such as localization) systems and can have numerous undesirable effects such as nodes falsely entering the local maximum situation, routing loops, reachable destinations being declared unreachable, and the failure of face routing (location inaccuracy can cause network disconnection during planarization). Although GPS is generally considered the default location system for most geographic routing protocols, some authors have acknowledged the possibility of using alternative systems (such as RSSI-based localisation), and it is possible that most geographic routing protocols could function without GPS so long as another localisation system was able to provide them with coordinates. Although GPS does have limitations such as cost, energy consumption, inaccuracy, and cannot function indoors, it is important to recognise that alternative localisation systems still represent an emerging area of research and cannot always guarantee better or even equal accuracy compared with GPS. Therefore, it is important to acknowledge the possibility of location error regardless of the type of system being used. One such means of achieving this is to determine the probability of a given node’s position being inaccurate, an approach used by [58] who then use the probability of location inaccuracy to determine the likelihood of a chosen node experiencing transmission failure or sending the packet backwards. However, it is important to note that this approach assumes transmission range will remain constant (an assumption that does not always hold true in practice) and therefore is susceptible to failures arising from interference or obstacles. As an alternative to both GPS and alternative location systems, some researchers such as [61] proposes the use of virtual coordinates as a means of avoiding location errors. A virtual coordinate approach was also adopted by [29] as a means of avoiding concave nodes and voids, however [29] used a combination of real and virtual coordinates whereas the work of [61] uses virtual coordinates alone.

Although node mobility is a problem that affects all types of ad hoc routing protocol, it can have particular effects on geographic routing protocols such as failure to reach destination (if the destination node moves significantly from its previous position), planarization errors (a connected planar graph could become disconnected after a node moves out of the network), while out-of-date position information (caused by mobility) can lead to loops, suboptimal routing, and delivery failure. To counter the effects of mobility several approaches have been taken; the two most popular combine some element of adaptive beaconing with mobility prediction so that nodes moving at high speeds and with varying acceleration send updates more frequently than sedentary or low-mobility nodes. [67] uses mobility prediction to determine when a node’s previous update will be out-of-date and then sends an update message when this occurs, while also keeping track of neighbouring node’s mobility patterns in order to determine when their previous updates will become out-of-date and remove nodes who are outwith their transmission range. [60] also uses location prediction, in combination with their location error estimation to predict the position of their next hop before transmitting and if the next hop is not within its transmission range the node will choose another packet. Similarly, [60] also describes a mechanism where all nodes involved in the forwarding process determine whether the destination is within direct transmission range and attempt to forward the message directly if it is. Although mobility/node position prediction schemes are also used in determining ability to meet QoS requirements such as in the work of [64], their focus on QoS means that for the purposes of this paper, they will be dealt with in this context.

[66] uses adaptive beaconing without mobility prediction, instead favouring an approach based on ‘reactive beaconing’ in which beacons are only sent by nodes hearing a route request or data packet sent to a neighbour and who will then send beacon messages to its 1-hop neighbours. An alternative method involving caching previous routes and sending route requests messages when a route is not available in the cache is also described in [66]. A complete alternative to adaptive beaconing and mobility updates is presented in [74] with their rope ladder routing protocol which uses a logical rope ladder structure to construct multiple linked paths so that in the event of node failure an alternative path can be used, and then (if possible) the original path can be returned to at a future node.

D. Power Consumption

Despite the significance of reducing power consumption and conserving energy, only limited work in geographic routing has focused on this area, while little work has been done on using location data to reduce power consumption in non-geographic routing protocols. Adjusting transmission strength is widely recognized as a means of reducing power consumption as a node’s radio is frequently a significant consumer of power and although authors such as [80] acknowledge the possibility of adjusting radio strength to save energy, they typically use such adjustments to reduce interference and noise. In [1] several protocols aimed at reducing energy consumption and increasing network lifetime are discussed, although they make use of control messages (which adds an overhead in terms of power consumption and resource usage) and the protocols were only tested on static networks. Building on the work of [1] [19] then developed an energy-efficient geographic routing protocol that guaranteed delivery. [81] use location-aware multicast trees and zonal routing to create backup routes that perform some tasks of the primary route to reduce energy consumption and also include the feature for nodes to monitor their energy levels and remove themselves from the tree if their levels fall below a certain threshold. While [81] introduces an energy consumption metric that can be used for evaluating link costs.

E. QoS

The decreasing cost of mobile computer-like systems capable of communicating with each other has led researchers in industry, commerce, academia, and the military to consider the possibility of using ad-hoc networks for multimedia applications. An important aspect of multimedia networking is QoS which ultimately governs the quality of the received
multimedia; this in turn has led to recent research in ad-hoc networks focusing on adapting and designing new QoS metrics and technologies to cope with the requirements of multimedia transmission. Although there have only been a few geographic routing protocols that have explicitly considered QoS, such work has shown promise, most noticeably in the area of QoS predictions based on mobility and location. Such predictions are used for route selection purposes by [64] and [16] both of which use a variant of the ‘connection time’ metric – the calculated time for which a link between two nodes will remain active based on node mobility – to meet QoS demands. While [64] focus on ensuring adequate connection time and reducing delay (which they argue is a function of hop count and thus seek to reduce hop count) [16] use their motion stability parameter (similar to the connection time metric used by [64]) to reduce jitter (node’s needing low levels of jitter will prioritise nodes with high levels of motion stability). While [90] use a combination of geographic and ticket-based routing to manage QoS, however their approach is only suitable for static networks.

Although the above protocols can be broken into two broad categories (prediction-based and non-prediction-based) they all share several characteristics of the IntServ model for QoS (also known as hard QoS) in which source nodes determine their requirements and whether suitable nodes exist in the network, and if such nodes exist then a path is constructed to the destination. Although IntServ is a popular model for QoS in conventional networks it has several drawbacks related to its reservation mechanism and its inability to compromise (nodes that cannot meet the QoS requirements are automatically excluded from the routing process), the first of which seems to go contrary to the general goal of localisation in geographic routing. While none of the above protocols explicitly use the IntServ model (and thus are not vulnerable to all of its faults) they do appear to share several of its characteristics including a lack of compromise and some form of resource reservation; albeit in a more localised form rather than source-destination resource reservation. In conventional QoS, the main alternative to the IntServ model is DiffServ in which each node upon receiving a packet determines its classification and treats it accordingly. Packet classification can be based on factors such as QoS parameters specified in the packet, the type of content, or the node it originated from but classification is always the responsibility of the receiving node and thus it is at liberty to treat every packet as it sees fit. Although it would seem that a highly-localised scheme that allows compromise would be more suitable for QoS in geographic ad-hoc routing protocols, so far only [85] has used the DiffServ model in this domain. However, [85] only features minimal classification (multimedia and non-multimedia) and geographic routing is used as a backup rather than primary means of routing.

VIII. OPEN AREAS OF RESEARCH

So far this paper has focused on reviewing and comparing existing literature, however in doing so drawing attention to areas that had not been sufficiently covered or were still emerging was unavoidable. The purpose of this section is to expand on earlier comments and identify area of geographic routing that could benefit from further research.

A. Alternative Models

While the UDG has proved consistently popular as both a theoretical model to aid comprehension as well as a base for the design of geographic routing protocols it is far too high-level and abstract to realistically model ad-hoc network conditions. Its failure to take into account differing transmission radii which exist for a variety of reasons (different radios, obstacles, interference, etc.), node movement and death has led to protocols that depend on it (most noticeably face routing algorithms) resulting in sub-optimal performance in realistic conditions. Promising work such as that of [36] and [37] in developing variants of the UDG that allow for varying and unequal transmission radii as well as research into 3D models such as the Unit Ball Graph [44]. Similarly, frequent mention of the UDG’s weaknesses suggests a small (but growing) trend of exploration into alternative models. It is important to note that the UDG was originally conceived as a general mathematical graph and was not therefore designed to model ad-hoc networks, its simplicity and generality have seen it appropriated by various researchers in the area of geographic routing. Despite its faults the basic model is one of clarity and simplicity, and any alternative models should endeavor to retain these features while providing more realistic abstractions of actual ad-hoc networks. It was commented upon earlier, that while 2D routing protocols often fail in 3D environments, 3D routing protocols cannot always offer optimal performance in 2D environments. An interesting area for future research could be a hybrid protocol that is able to determine the environment and adapt accordingly.

B. Applications

The focus of this paper has been on the routing protocols themselves with only limited mention of their application. Although most of the protocols surveyed in this paper have been evaluated through theoretical analysis and simulation, some authors such as [33] have deployed their algorithms on real devices for testing. However, even if these protocols are able to operate on real devices and in the real world, it is important to consider what these protocols will actually be used for. Ad-hoc networks are often mentioned in the context of military and disaster recovery applications when limited or no infrastructure is available; for instance, [91] uses geocasting for streaming multimedia between emergency services personnel in disaster recovery scenarios.

A possible area of use for geographic routing protocols is that of Geographic Information Systems; technologies used to gather geographic and spatial data [92], as these systems are used to extract geographic data it is possible that they can in turn benefit from routing mechanisms that also use geographic and spatial information. Similarly, the information gathered from these systems could in turn be used for modeling
environments for simulation and design (as discussed in the last section) thus enhancing their accuracy.

Another emerging area of research that uses location information is that of location-aware services; part of the larger area of context-aware services, in which location information is used in the provision of services in areas as diverse as entertainment, healthcare, and shopping [93]. Context-awareness can itself be seen as an aspect of ubiquitous computing; a technological philosophy developed by Mark Weiser which focuses on making human-computer interaction more natural by integrating them with the physical environment and in turn making the devices aware of the environment they are in (the context-aware aspect) [94]. As with GIS’s, location-aware services are based on the use of geographic information, so the use of geographic routing protocols seems logical here. It could also be argued that as geographic routing protocols use location information for routing they are themselves a form of location-aware service as the routing they perform is essentially a service to the end users.

An area of research with potentially interesting implications for geographic routing is Content Centric Networking (CCN). In CCN the emphasis is placed on the content itself instead of the location at which it is located, and content is accordingly addressed by name rather than address [95]. Although CCN is built on top of IP it differs from the conventional TCP/IP stack through its use of content and not location as its main abstraction [95]. This may immediately seem to be at odds with geographic routing given geographic routing’s natural reliance on (physical) location, but there has been some work that looks at geographic routing in the context of CCNs. For instance, [96] uses the CCN approach in emergency situation MANETs.

With the end-user in mind, QoS is an important issue as it is ultimately the network-level mechanism for guaranteeing the information is transmitted in a manner suitable for the needs of the end-user application. With this in mind, we will turn our focus to the area of QoS and how further research into geographic routing can be of benefit.

C. QoS

While there has been some promising research into the area of geographic QoS routing it is still an emerging area and one that would benefit from greater attention. Of particular interest is the ability to use movement and location in predicting whether other nodes can fulfil the required QoS standards. Previous work such as that of [64] and [16] as well as some of the papers included in the mobility section could potentially be used as the basis for more advanced prediction models. For instance, machine learning has been for recognising behavioural patterns in p2p networks by [97] and it would be interesting to see if techniques such as machine learning and pattern recognition could be used to monitor mobility and location data and in turn make ‘educated’ guesses about a node’s ability to fulfil QoS constraints. Similarly, the extent to which geographic QoS routing algorithms can be classified as localised should also be considered. If localisation is a desirable feature of geographic routing protocols then QoS-based protocols should not be an exception. Clearly the DiffServ model with its emphasis on local classification and prioritisation is more compatible with this ideal than IntServ which relies on resource reservation, however the DiffServ model is intended for conventional and not ad-hoc networks and should not be taken as a template. Instead a DiffServ-inspired ad-hoc QoS model might be a better approach, or even borrowing some concepts from DiffServ rather than using/modifying the entire model. With the rise of multimedia WSNs (MWSNs) there has been some work on applying techniques from geographic routing to WSN QoS such as [98].

In addition to QoS another important factor that must be considered when addressing the need of the end-user is privacy and security. Although the routing layer cannot be expected to be the sole guardian responsible for security, it is also not reasonable to expect future routing protocols to ignore the issue of security when so many potential attacks are targeted at the routing layer. The next section addresses the issue of privacy and security both within a general ad-hoc routing context and from the more specific point of view of geographic routing.

D. Privacy and Security

The majority of the work in this area has focused on providing anonymous routing as seen in the likes of [18], [51] and [52]. An interesting approach is the use of temporary IDs which are often a function of position and time – the aim of which is to prevent two nodes having the same temporary ID as they both cannot be at the same position at the same time, however this basic approach can suffer from location inaccuracies which can lead to two nodes having the same coordinates even if they are not at the same position. More advanced methods of determining temporary IDs are essential if geographic routing protocols are to be used in the real world. Similarly, the concept of the anonymity zone (and its variants) is promising and future work should focus on the ability to provide varying degrees of anonymity/privacy to nodes based on their needs, while at the same time avoiding decreased routing accuracy.

One approach in particular that showed promise was the location-aware privacy profiles of [52]. These allowed the user to determine the level of location privacy related to a specific place or area, for instance to avoid other devices being able to determine they were in a hospital. This extension to the basic anonymity zone approach used by many secure geographic routing protocols leads to a greater degree of flexibility and ‘personalization’ rather than applying catch-all policies. Machine learning has already been mentioned in the context of QoS, and it is possible that such techniques can also be made use of in designing more intelligent and secure geographic routing protocols.

In addition to addressing the level of security or privacy, future research in this area should consider efficiency and power consumption as counter-constraints to privacy and security so as to avoid the development of protocols that are optimally secure but unfeasible for practical use. However, as
indicated earlier, while the routing layer cannot remain ignorant of issues such as security and power consumption neither should it be the only layer responsible for dealing with them. There is potentially much merit in developing approaches that transcend the boundaries of the traditional layers and combine the abilities of two or more layers for overall benefit. The next section proposes some ideas for how geographic routing can be expanded to take a more holistic, cross layer approach.

E. Cross Layering

Although the focus of this paper is on geographic routing protocols, it is often desirable that a cross layer approach is adapted in certain situations. Future research should focus on ways in which the routing layer can interact with other layers. For instance, a routing protocol that is able to use information from the MAC layer to determine its exact transmission range so as to reduce potential next-hop nodes to those that have a good probability of being reached. Similarly, energy-efficiency in geographic routing is an area that has not received a great deal of coverage, but that is of great importance when battery-powered devices are used. Although several approaches did raise the possibility of adjusting transmission power to save energy, there is still considerable scope for research in this area. Recent advances in cognitive radio can also be explored as a means of adapting transmission parameters both for saving energy and reducing interference such as in [99] where geographic routing is used to minimise the total number of transmissions.

IX. THE FUTURE OF GEOGRAPHIC ROUTING

This paper has largely focussed on surveying, describing, and analysing existing literature relevant to the field of geographic routing. However, as the purpose of this paper is to provide a comprehensive introduction to the field of geographic routing it is also important to discuss the direction that geographic routing research is going in. Much of the early research appeared to be dominated by the novelty of using location information for routing purposes, and as such protocol such as greedy forwarding, Compass II, LAR, etc. could be said to have existed before there was really such a field as geographic routing. As the field began to become established, emphasis was placed on finding a solution to the local maximum problem that plagued greedy forwarding; this largely took the form of face routing although various other alternatives have (and are still being) explored. From there the focus moved on to improving the performance of face routing and developing hybrid greedy-face protocols.

As geographic routing expanded as a field research became increasingly diverse with researchers applying the geographic routing paradigm to areas such as QoS, security, and energy reduction. Or in some cases they sought to bring these areas to geographic routing. At present it would appear that the area of QoS is an area that is undergoing continued research. This is in line with the general trend towards considerations of QoS in ad-hoc protocols. As mobile devices (whether sensors, smart phones, handheld computers, etc.) become cheaper and more powerful there is an increasing demand for multimedia content on these devices. As QoS is seen as a network-level means of guaranteeing end-user quality. Similarly, the recent growth of location-based services and applications, particularly on smartphone platforms also poses a potential opportunity for geographic routing.

If it is seen as desirable to incorporate location-awareness at the application layer then this could lead to more developers and researchers considering the use of location at lower layers. This may also be linked in with the area of ubiquitous/pervasive computing in which context (including location) is seen as a desirable attribute. As location has already been used to great success in geographic routing it is possible that other forms of context will also be included and geographic routing protocols could find themselves at the forefront of context-aware communications. Similarly, the rise of other technologies such as CCNs and the Internet of Things are also potential areas of application for geographic routing.

Therefore, it is likely that one of the major areas of development in geographic routing will be in supporting end-user applications, and more specifically supporting multimedia and QoS. In addition to the increasing drive towards multimedia another area of research in ad-hoc networking that has undergone significant development is security. It is now widely acknowledged that the need to provide guaranteed levels of security is crucial if MANETs are to be used in the real world. Although there has already been a great deal of research on secure geographic routing there is scope for much more. As devices become more powerful this should act as an enabler for implementing more secure systems such as complicated cryptography and authentication schemes. However, there must be a trade-off between security and functionality, and it is therefore necessary to strike a balance between achieving a high level of security and a high level of performance. Context-awareness was mentioned in the context of QoS and multimedia, but could also be of use to secure applications so that instead of focussing on securing geographic routing, geographic routing concepts such as location-awareness could be incorporated in the grander scheme of context-aware security.

Finally, it is important to recognise that a major driving force in the development of existing geographic routing protocols has been WSNs. Indeed, WSNs typically represent the main real-world application of geographic routing protocols. It is likely that the use of geographic routing in WSNs will continue as WSNs themselves become more widespread and often contain positioning hardware such as GPS or are able to perform relative localisation. Within WSNs energy consumption is still one of the major issues and is likely to remain so as long as WSN devices rely on batteries. Although there has been a lot of research in WSNs as to how to reduce energy consumption there have been very few geographic routing approaches that incorporate energy saving. However, there are clearly advantages to considering position, location and distance when trying to conserve so it is likely that energy consumption will play a large role in future geographic routing protocols for WSNs. Aside from energy.
consumption other areas of WSN research that are expanding are QoS and security, in a manner similar to conventional ad-hoc networking. Therefore it is likely that developments in these areas will be mirrored in ad-hoc networking and WSNs with WSNs placing more of an emphasis on energy consumption.

With regards to WSN applications, it is likely that as WSNs are continually deployed in unusual environments that this will have an impact on the development of protocols. WSNs are continually being deployed in 3D environments where position varies in terms of altitude/height as well as longitude and latitude. There has already been some work into the development of 3D geographic routing protocols and models and this is likely to continue as WSNs are continually deployed in 3D environments. These developments are also likely to be of use to conventional ad-hoc networking. Similarly, research into 3D models is also liable to fuel research into more accurate models for other purposes. The UDG model is still widely used and forms the basis of most face routing/planarization algorithms but it is too idealised to hold true in the real world. There has already been some research into modifying it to incorporate heterogeneous transmission radii or interference. However, although the model has strengths such as its simplicity and ease of understanding its inability to incorporate uni-directionality and other deviations from the norm found in real-world networks renders it somewhat out of date. Therefore it is likely that as research into geographic routing continues a new model will be devised to take the role of the UDG as both the base model for planarization and for modelling ad-hoc networks in general.

X. CONCLUSION

Since its inception in 1987 significant and diverse research has led to the establishment of geographic routing as a promising field of research with various potential applications ranging from telecare monitoring to multimedia streaming. One of the primary aims of this paper was to capture the diversity of such work with protocols being reviewed ranging from QoS prediction systems to anonymous routing and from enhanced face routing variants to 3D protocols based on the UBG. Although a significant number of protocols are based on the greedy and face routing approaches (including hybrids) alternative approaches are an active field of research. Whether this entails applying techniques from robot navigation or designing variants of the UDG, many researchers are focusing on developing protocols and models that more accurately resemble the real world.

Similarly, a number of ‘niche’ protocols concentrating on aspects such as QoS, security, mobility management, and energy efficiency have been developed some of which eschew the traditional reliance on greedy or face routing models. Secure and private geographic routing has proven to be a popular topic for geographic routing protocol, which is hardly surprising given the prominence of security-related research in most areas of networking. Within security, anonymous geographic routing appears to be a promising area of research with authors such as [17] and [18] developing frameworks that facilitate efficient geographic routing without compromising privacy and security. The work of [52] is particularly interesting due to its incorporation of privacy profiles which allow users to control the extent to which their positional data is associated with particular areas or landmarks. As well as providing security, another important aspect of geographic routing is location accuracy as inaccurate position information caused by location errors and unpredictable mobility can have disastrous effects on geographic routing protocols. Work such as the scheme for calculating the probability of location error used by [58] aims to mitigate the effects of location inaccuracy while mobility prediction schemes used by the likes of [68] adapt a node’s position updates depending on its speed and direction of movement. Mobility prediction is also a factor in some geographic QoS routing schemes such as those used by [64] and [16] which use a node’s mobility pattern as the basis for determining its ability to meet QoS constraints. Other geographic routing approaches that consider QoS include a hybrid location-aware ticketing approach developed by [90] and an IntServ-inspired multimedia routing protocol for streaming [85]. An area which has been somewhat neglected in comparison to other areas covered by this paper was that of energy/power-aware routing. Although authors such as [19] and [83] demonstrated the potential for location information to be used for saving energy, very little research has been performed in this area outside of WSNs.

To reflect the diversity of geographic routing protocol research the main review portion was structured logically in terms of issues relating to geographic routing rather than chronologically. This allowed the paper to address specific issues such as security and QoS in order to give an overview of how these issues affected geographic routing before examining how previous literature has addressed these issues (and sometimes identified new ones). Research patterns and trends were then identified, as were areas that could benefit from increased research. It is therefore hoped that this paper will be beneficial not only as a summary of existing work in the area or an aid in understanding the basics of geographic routing but also as a stimulus for future research.

REFERENCES

Fraser William Cadger (M’–10) was born in Glasgow, Scotland in 1989. He received his BSc in Networking and Systems Support with First Class Honours from Glasgow Caledonian University, Scotland in 2010. He is currently working towards his PhD at the University of Ulster where he is a member of the Ambient Intelligence team within the Intelligent Systems Research Centre (ISRC). The focus of his PhD is on combining geographic routing and p2p streaming to deliver high-quality multimedia over wireless mesh networks in disaster recovery scenarios. His research interests include wireless ad-hoc networking, context-aware communications, peer-to-peer routing, pervasive networking and applied machine learning.

Kevin Curran (M’–SM’) is a Reader in Computer Science at the University of Ulster and group leader for the Ambient Intelligence Research Group. His achievements include winning and managing European Framework projects and Technology Transfer Schemes. Dr Curran has made significant contributions to advancing the knowledge and understanding of computer networking and systems, evidenced by over 700 published works. He is perhaps most well-known for his work on location positioning within indoor environments, pervasive computing and internet security. He is a regular contributor to BBC radio & TV news in the UK and is the Editor in Chief of the International Journal of Ambient Computing.

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

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