A Survey of Routing Protocols for Vehicular Ad-Hoc Networks

ANDREW B. ROSEN
Georgia State University

With the growing amount of integration of Internet capable devices with other technology, non-tradition network scenarios become more common and need a correspondingly unique solution. This paper presents a summary and analysis of a number of routing protocols in Vehicular Ad-Hoc Networks (VANETs), a type of challenged network with a high demand for new applications.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing Protocols*; F.2.2 [**Theory of Computation**]: Nonnumerical Algorithms and Problems —*Routing and Layout*

General Terms: Algorithms, Reliability

Additional Key Words and Phrases: algorithms, epidemic, MANETs, protocols, routing, VANETs, vehicular networks

Contents

1. INTRODUCTION

The Internet of today has evolved into a nearly ubiquitous construct. In heavily urban areas, it is easy to make the assumption that a fast and stable connection to the Internet is easily available at all time. However, reality shows that this simply isn't the case. There are many environments where there exists no network infrastructure and too few resources to leverage. There are many situations where there are severe impediments to a stable connection. Distance, topology, and movement of nodes within the network are common challenges in what are known as *challenged networks* [Fall 2003] [Burleigh et al. 2003].

With challenged networks, designers of both the algorithms and the devices are subjected to a variety of challenges due to the environment:

Disconnection. One of the base assumptions of a standard TCP/IP network is the existence of some end-to-end communication. Within the framework of challenged networks, disconnection and partitioning can occur quite often. While disconnection can often be attributed to faults or energy saving techniques, we will primarily concern ourselves with disconnection due to the movement of the nodes in the network.

Regardless of the circumstance, working within a challenged network requires users, designers, and developers to recognize that a disconnect between two nodes does not immediately translate to a failure in the network. On the contrary, in many cases (like trying to communicate with an orbiting satellite), this may very well be a norm.

Latency. Some networks have to deal with great distances, communicating with other nodes over the void between stars. Other network need to deal with physical obstacles, such as vast mountain ranges. Sometimes the challenge is the sparsity of the network, spread out over a great expanse. As different as these challenges seem, they all force the network to contend with latency. In addition, challenged networks often have to contend with very low rates of data transmission; this further exacerbates the issue.

As a result, challenged networks must avoid using any protocol for communication that requires a great deal of overhead, such as TCP. Round-trips need to be minimized. Any sort of bandwidth control will need to be handled on a hop-to-hop basis, as, once again, we cannot rely on end-to-end connectivity.

Reliability is also a grave concern, as getting sending a message with an error will have repercussions that can tie up network resources with resending and correcting. This forces the network to provide mechanisms for redundancy.

Excessive Queues. A factor caused by the latency and disconnection is that messages will need to queue for a long time. This, in turn, requires nodes to have queues that can accommodate these messages. Rather than discarding messages due to disconnection, challenged networks need to hold messages within their queues for extremely long periods of time. Messages may be held for such a long period that new paths may need to be chosen based on changes in the network.

[Fall 2003] gives a extremely thorough examination of these factors and other considerations.

Of particular interest is a subset of challenged networks that arise from vehicular networks [Füßler et al.] - *vehicular ad-hoc networks* (VANETs). VANETs present a problem with a unique background as well a a plethora of routing algorithms and architectures used to overcome the challenges involved in highly mobile, partitioned networks. Both are explored here.

2. VANETS

VANETs are an offshoot mobile ad-hoc network research, a type of system with wide reaching applications. As such, it is worth noting that MANETs have several distinctive attributes and requirements[Gil-Castineira et al. 2008]:

Mobile. By definition, nodes within the network will be mobile. Any implemented protocol and algorithm will need to be aware of this fact.

Ad-Hoc. MANETs are built to be independent of any infrastructure [1]. This means that there are no nodes that have specific duties in terms of running applications or routing the traffic - nodes will need to perform both roles. In addition, nodes will need to be aware of their surroundings in some form, if only to be able to detect their neighbors.

Multihopping. Mobility and/or deployment will make some nodes only reachable by multihopping.

Energy Constraints. Nodes are usually small sensors or other devices. This means that the device has a correspondingly small source for power. They are not normally deployed in such a manner as to be able to receive power from an external source.

Scalbility. Applications might grow in scale, nodes may enter and leave a network due to mobility, causing the size of the network to change. A MANET has be able to able to scale and, to borrow a term from peer-to-peer networking, has to be able to tolerate *churn*.

Security. As all connections wireless and the devices are limited in resources, nodes are particularly vulnerable to attacks.

While VANETs are a type of MANET and share most of the same qualities, they have their own unique aspects[Pereira et al. 2011]:

Vehicles. The majority of the nodes, if not all, represent vehicles. These range from aircraft, locomotives, automobiles, and watercraft. The vast majority of research has been focused on cars.

[1] Which is not to say that MANETs can not or are not built to leverage any available infrastructure; MANETs just don't rely on one.

Extremely Mobile. Vehicles can move at extremely fast speeds, changing the topology of the network very rapidly. This movement is not random, but predictable, as discussed below.

Variable Topology. Due to the aforementioned high speeds, the topology of the network, ie which nodes are connected or can connect, will not remain stable. For example, two vehicles driving $25\frac{m}{sec}$ in opposite directions with a maximum transmission range of $1000m$ will only be able to maintain a connection for 40 second.

Scale. Every vehicle is considered a node, leading to extremely large networks on highways and urban areas.

Partitioned Networks. Due to the size of the network, most nodes will not be able to see other nodes. When the density of the network is low enough, the network can become disconnected.

No Energy Constraints. VANETs, by definition, able to draw power from their vehicle. Any vehicle with enough power to move at high speeds has more than enough to also power the applications [2].

Few Storage Constraints. Most vehicles hosting an applications will have physical space adequate to house sufficient storage. However, the storage must be rugged enough to deal with the motion of the vehicle.

2.1 Notes on Mobility

The vast majority of difficulties that are experienced in VANETs are due to extreme mobility of the network. The mobility is the cause of the variable topology, the area the network covers, the partitioning of the network, and the short connection times. As such, it is essential to understand the behavior of nodes in regards to their movement.

We have two different models for mobility between MANETs and VANETs. The first is *unpredictable* mobility. As the name suggests, this covers nodes who move in paths that cannot be easily predicted with a high degree of accuracy. Connections formed within these networks are opportunistic and arise from the mobility of the nodes in the network entering within communicating range of one another.

The other paradigm is *predicable* mobility. Predictability, however, occurs with different degrees of accuracy. For example, with a vehicular networks, cars are moving with varying velocities, yet they travel along completely predetermined routes [3]. Planes and other aircraft travel along pre-established paths determined by a flight plan. Ships and other maritime vessels act similarly.

Turning back to cars again, if the route and nodes being examined are primarily along an interstate, then the paths of nodes are one dimensional, with nodes entering and leaving the networks at the exits. A long, winding path in rural areas could be simplified in a similar manner. If the roads being mapped are in an urban area, then a two-dimensional map would be constructed due based on the intersections, traffic lights, and side roads. Even in this more complex scenarios, we are able

[2]Which is not to say that there's no need to be energy efficient; it's just not a top priority. This doesn't also mean there's unlimited power either.
[3]With the exception of off road driving.

to make valid assumptions about the paths the vehicles will take. If the nodes in question are only a small subset of all possible vehicles, such as taxis or buses, then the paths become even more trivial.

This especially impacts interplanetary communications, where satellites travel at extremely high velocities, yet their positions are predictable by their very definition. By being in orbit, satellites travel along a set path that can be modeled by a system of physics equations. This degree of predictability when applied to mobility is highly beneficial, as it allows us to know where nodes will be at any given time. Nodes can be programmed to save power until needed, calculated by the movement of other nodes.

In summary, it is the motion that is both the source of the primary challenges that VANETs face, but also an aspect that can be exploited.

3. ROUTING IN VANETS

As inferred from the above section, routing protocols for VANETs have been primarily influenced by the mobility of the nodes within the network. The number of protocols that have been developed are vast, but while there are many different approaches to routing, there are only two broad factors to consider for VANET routing protocols.

The Importance of Position. How important is the physical location of the node to the protocol? Algorithms that rely on the physical position of nodes will use maps of the surrounding area, combining the physical location of the node with the context of its surroundings to make intelligent routing decisions. Protocols that do not emphasize the precise knowledge of the physical location of nodes or their surrounds normally assume there is no reliable way to gain this knowledge.

Connectivity. Is end-to-end connectivity considered the norm of the network? A lack of connectivity is the norm in sparse networks, such as rural areas, or in highly partitioned networks [Wisitpongphan et al. 2007]. If end-to-end connectivity does not normally exist in the network, these protocols have to be *delay tolerant*. This approach to architectures and protocols is a common technique for overcoming the difficulties faced in challenged networks [**?**].

The two factors are normally intertwined: position-based protocols normally assume that end-to-end connectivity exist in some form through the network, given intelligent forwarding decisions are made. Disconnection a common obstacle to overcome, but it is not the defining difficulty of the network. On the other hand, protocols that do not rely on position typically assume that end-to-end connectivity is rare.

3.1 Trivial Cases

Before getting to the actual protocols, two trivial cases are presented below for completeness [Pereira et al. 2011].

3.1.1 *Direct Routing.* In Direct Routing, a sending node that wants to send a message carries it until it comes into contact with the destination. This can be accomplished in two ways. The first way is for the source to simply carry the message, relying on the random motion of the nodes to eventually bring it into

contact with the destination. Unfortunately, this has no guarantee of success, nor is it efficient. The other method is to have the source attempt to move to the destination, rather than randomly drifting. This has its own difficulties. For one, if multiple nodes are trying to send messages to each other, then we could end up with them chasing to each other in an infinite loop. This method also requires that a node move specifically to deliver the message, regardless of any other duty it may have.

3.1.2 *First Contact.* In this protocol, the source passes the its message to the first neighbor it comes into contact with and so on and so forth until the message arrives at it's intended destination. Effectively, the message performs a random walk to the destination. Once again, this protocol does not have any assurance of completion. However, as the protocol adheres to a store-carry-forward principle, where each node hands of responsibility of the message to the first neighbor, it is a trivial example of a delay tolerant protocol.

3.2 General Ad-hoc Routing

With the trivial examples examined, we now turn to two highly-used routing protocols for wireless networks in general, Dynamic Source Routing and Ad-hoc Distance Vector Routing [Gongjun et al. 2010]. Because they were developed for general MANETs with random mobility, both experience difficulty in the area of vehicular networks.

3.2.1 *DSR.* The first actual routing method to examine is Dynamic Source Routing (DSR) [Johnson and Maltz 1996]. DSR was designed as a general routing protocol specifically for wireless MANETs. As such, it has seen a lot of coverage in the field and is often used to as a starting point to compare new protocols against [Lochert et al. 2003]. DSR sends packets by constructing the end-to-end route at the source. This route is send as part of the packet's header, listed by addresses of the nodes. However, this route is determined dynamically by the sender as part of communicating.

Whenever a node needs to send a message, it checks if it already knows the route. If it doesn't, it broadcasts *route discovery* messages in an attempt to find one. When a node receives a *route discovery* message that it hasn't seen recently, it will return the path to the destination if available, or it will rebroadcast the message using itself as the host. This will allow the request to work its way through the network. If an error is generated due to a node being unreachable, then an error message is generated which propagates through the network, modifying routes concerning that particular node.

Once of the greatest benefits of DSR is the simplicity of the algorithm; it is extremely easy to implement. DSR avoids having to flood the network with periodic update messages, since routes are constructed as needed. This keeps the overhead down and conserves the almost invariably limited resources of the network. However, the reactive nature of this routing protocol means the nodes have to contend with latency from establishing new routes and from the maintenance overhead caused by stale routes. Furthermore, the more mobile the network, the longer the paths will be. Longer paths means that the overhead saved by not flooding the network with requests is overshadowed by the consumption of network resources in

the creation of routes.

In summary, DSR is well suited for less mobile ad-hoc networks that need to conserve bandwidth, but not for any challenged network involving a high degree of mobility.

3.2.2 *AODV.* Another popular general routing protocol often adapted for VANETS, Ad-hoc Distance Vector Routing (AODV) [Perkins and Royer 1999] has many similarities to DSR. It is a reactive protocol, only creating routes when required. However, the differences readily become apparent.

Rather than using source routing like DSR, AODV implements routing tables. When a node desires to communicate with another node, it broadcasts route requests messages through the network. Once the destination (or a node that has a recent enough entry to the destination) is contacted, it unicasts a message backwards through the newly discovered links. These links are given sequence numbers that correspond with the messages, which allows the protocol to avoid loops and "count to infinity" problems that normally affect distance vector algorithms.

Like DSR, AODV creates routes only when required, but it does implement beaconing to maintain links, which adds to the overhead. In addtion, mobility severely limits how efficient the protocol is. AODV was shown to be unable to effectively perform in VANET dues to numerous lost packets and being unable to establish links quickly. [Li and Wang 2007] This can make AOSV unsuited for challenged networks.

It should be noted that in a comparison of some routing protocols for a maritime network over WiMAX, AODV was found to perform reasonably well when there was high connectivity [Lin et al. 2010].

3.3 Greedy Position-Based Routing

The flaws of the above protocols prompted the development of new approaches for routing, ones that would be able to cope with the mobility inherent in vehicular networks. Because of the availability of GPS technology, protocols were developed that could easily take advantage of information provided by their locations and maps of their surroundings. This brought its own set of challenges and inherent assumptions[Lochert et al. 2003].

—All nodes in the network require a means of knowing their locations (i.e. they need to be equipped with GPS device).

—All nodes keep track of their neighbors. In other words, nodes will use beaconing to keep track of other vehicles in range. This is typically denoted by sending "hello" messages periodically.

—A node that wants to send a message to some destination will require the position of the destination. Each protocol has different ways of obtaining this[4].

By far the most popular and successful position-based routing protocols were those that implemented a greedy selection mechanism for forwarding messages. The

[4]This problem, when mapped to satellites and other extraterrestrial communications, becomes a bit easier. Orbiting objects have the advantage of being in an orbit, which allows us to find their location (present or future) by using a system of equations

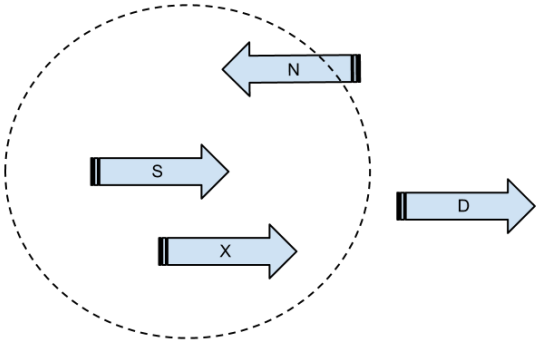

Fig. 1. Node S is trying to transmit a message to D. By the rules of GPSR and GSR and other greedy selections, S would select N to forward the its message to, despite the fact that N is traveling away from D.

specifics of how each protocol is differs between the protocols and each discussed in more detail below.

3.3.1 *GPSR.* Greedy Perimeter Stateless Routing (GPSR) [Karp and Kung 2000] uses a "greedy" approach to find a path from the source to the destination. Upon obtaining the physical location of the destination[5], the node will forward it to the neighbor closest to the destination. That neighbor will in turn forward the message to his neighbor closest to the destination. This continues until the destination receives the message. Once a connection is successfully established, each the two endpoints append their location to each packet to keep each other informed of their positions.

Due to the nature of greedy forwarding, the protocol need a way to deal with the problem of a local maximum. In other words, the node has to have another option if it is the closest to the destination among the set of it and its neighbors. The *perimeter* part of the name comes from how GPSR deals with this situation. When some node is the local maximum, it will use the right hand rule to select a neighbor in an attempt to "circle around" the gap separating the node from the destination.

GPSR works fairly well under certain scenarios, such as freeways and highways. However, it suffers from a number of issues in urban areas, such as buildings and other structures blocking visibility of nodes. [Lochert et al. 2003] discusses many of GPSRs flaws in deatil. Of relevence to our analysis is that GPSR also fails to take into account the velocity[6] of vehicles within the network, as shown in Figure 1. Once velocity is taken into account, it becomes readily apparent that the best decision is node X, not node N. This mobility also can cause routing loops, which is not acceptable.

[5][Karp and Kung 2000] assumes that some given server is available to all nodes in the network. This location server would be able to provide the physical location all nodes in the network. It would only need to be communicated with once at the start of each connection.
[6]Recall that velocity is a vector: it comprises both magnitude and direction.

3.3.2 *GSR.* Geographic Source Routing (GSR) [Lochert et al. 2003] was developed to address the faults that traditional ad-hoc routing methods such as the above DSR and AODV experience, as well as avoid some of the deficiencies of GPSR had within urban environments. Cities present a much different challenge than a highway setting due to three different factors:

—flow of traffic and congestion (multiple stops, lights vs no stops or lights).

—Primarily two dimensions of movement.

—Buildings provide obstruction where some would not normally exist.

GSR works by combining the positions of the nodes with street maps of the area to make intelligent decisions. GSR first obtains the locations of other nodes via a reactive location service (RLS). RLS involves flooding the network with queries for the position of a node[7]. Once obtained, the node then routes its message, but creates its route not based on hops, but on streets the message needs to travel along. Specifically, the node calculates the junctions[8] the message will encounter along the shortest path, as determined by Dijkstra's algorithm. Forwarding a message between two junctions is done in a greedy much the same as GPSR.

Tests showed that GSR outperformed DSR and AODV in terms of delivery rate and bandwidth usage [Lochert et al. 2003]. The protocol performs well in both the urban environment and on the highway. GSR still has a number of flaws. For one, GSR also does not account for the vehicles velocity, succumbing to the same mistake as GPSR. Another point against GSR is the shortest path is not always the best; GSR chooses the junctions it will send messages along without taking into account the node density [Jerbi et al. 2006]. Despite these flaws, GSR is still a strong algorithm.

3.4 Position Based, Traffic Aware Routing

The above protocols indicated quite strongly that while being able to use the positions of the nodes within the network, it is not enough. As shown in the analysis of GSR, position-based touting protocols also need to be aware of traffic in the network to make intelligent routing decisions.

3.4.1 *A-STAR.* [9]

Not to be confused with the A* search algorithm, Anchor-based Street and Traffic Aware Routing, or A-STAR for short, is an another algorithm designed for urban deployments. [Seet et al. 2004]. It greatly resembles GSR - both build their route by selecting the junctions (called *anchors* by in their paper) the message has to traverse and forward messages to nodes closest to the nearest junction. Where A-STAR differs is how it selects the junctions and how it recovers from a local maximum.

[7]Several techniques are applied to avoid overtaxing the network or causing a broadcast storm.

[8]A junction is simply where two roads happen to join or create an intersection. If we were to convert our system of roads into a more familiar representation of vertices and edges, our junctions would be the vertices.

[9]A-STAR could have been listed under the Greedy protocols, but it is novel enough to be set apart from them.

A-STAR proposes that vehicles will travel along certain paths more than others and the more popular paths will have denser traffic as a result. The authors reason that the streets served by buses correspond to streets with a high level of traffic. The streets with more bus routes are given a preferential weight creating a route. This creates a statistically-rated map. If the technology is available, a dynamically-rated map could instead be created by using the current density of traffic. The result is a route with the highest level of connectivity, rather than the shortest.

The protocol is also novel in how it deals with a local maximum. Rather than having a separate mode of operation for recovery, the node at the local maximum simply computes a new route starting from itself. The path the message was supposed to travel along is noted to be temporarily out of service in order to prevent other messages from heading in that direction. This new information is spread by piggybacking it with the message.

When compared against GSR, simulations showed A-STAR was able to deliver up to 40% more packets than GSR and had a much better end-to-end delay. The results also showed that the dynamically-rated version was able to perform better than the statistically rated version.

3.4.2 *GyTAR.* Much like all the other position-based algorithms presented thus far, the Improved Greedy Traffic Aware Routing Protocol (GyTAR) [Jerbi et al. 2006] consists of two components: junction selection and forwarding a message. Where the other algorithms create the whole series of junctions to traverse, the junctions are chosen dynamically.

First, the node needs to determine the optimal junction the packet should be sent to next. The vehicle currently responsible for the packet will want to forward that packet on towards the junction that has the best combination of being closest to destination and having the highest traffic density.

Once the desired junction has been selected, the node looks for a suitable neighbor to forward the packet to. It does this by looking at the list of all its neighbors and calculating their *predicted* position based on their velocity. The neighbor calculated to be closest to the next junction receives the packets. This is an improvement over the previous algorithms as it takes into account both the motion of the cars, the geography, and the traffic density.

To deal with a local optimum, GyTAR implements a "carry and forward" mechanism When the node can't pass off the packets to a better-placed vehicle, it will hold onto the data until that situation changes. In doing so the authors, knowingly or unknowingly, touched on one of the basic principles of routing in Delay Tolerant Networks. By storing the message and carrying it, the node has taken implicit responsibility for that data. This leaves GyTAR as a good stating point for creating a Delay Tolerant routing protocol based off of position.

Simulation results showed that not only did GyTAR deliver more maessages than GSR, it also incurred a much lower overhead.

3.4.3 *HTAR.* While the protocols have become more adept at accounting for physical traffic, none of them have yet accounted for network traffic. This is the basis for HTAR [Lee et al. 2011], which stands for Hybrid Traffic-Aware Routing Protocol. Like GSR, it focuses on finding the shortest path from source to sink.

To aid in this, HTAR uses a special node called the Junc-Tracker, which a specific node selected at each junction. The Junc-Tracker collects the physical and network traffic information of the roads, calculates weights associated with each node, and disseminates this information to neighbors and other Junc-Trackers.

A Junc-Tracker is selected based on how long the node will remain at the junction ($TTLJ$), which itself is calculated by

$$TTLJ = \frac{distance}{velocity}$$

. $TTLJ$ is transmitted as part of the period "hello" messages. When the Junc-Tracker leaves the junction, a successor is selected based on $TTLJ$. If traffic light information is available, more accurate predictions can be made.

Routing is done differently based on whether a node is between junctions or at some junction. When a node is between junctions, it will forward messages to the node that will bring it closest to the next target junction (taking into account velocity). If the node cannot forward the packet, it carries the packet until an opportunity to do so arises.

If the node is at a junction, it will checks if this is the first time the packet has been to this particular destination. If yes, the packet is given a new routing path and the next hop is chosen based on this. Otherwise, the next hop is selected from the preexisting route. If there is no suitable next hop is found, the node computes a new route until there is.

Simulations against GyTAR and GSR showed HTAR outperforming both in terms of packet delivery ratio and throughput. Despite the increased overhead of disseminating the traffic information throughout the network, HTAR has a built in safeguard of routing traffic around areas of high network congestion.

3.4.4 *FFRDV.* Fastest-Ferry Routing in DTN-enabled VANET [Yu and Ko 2009] is a routing scenario for messages with a limited number of destinations. The protocol works by first dividing the roads into logical blocks. Block sizes are chosen to reflect one-hop limits between the vehicles. The sender passes the message to the fastest moving vehicle it can reach to ferry the message to the destination. At each block, the ferry chooses the node with the best suited velocity to become the new ferry.

The advantages of this algorithm is that it performs reasonably well in terms of delivery and delay. It also is delay-tolerant and the selection scheme minimizes the number of hops that need to be made, reducing the total overhead of the network. It also treats velocity, rather than position, as the primary variable for greedy selection.

We believe this could be extended to the junction level. For example, say an ambulance is headed to a hospital; they would want to alert as many cars as possible to get them out of the way. The Ambulance alerts both the cars around it and selects the fastest ferry to carry the message ahead of it. Doing so, the ferry alerts other vehicles along the route to get out of the way.

3.5 Epidemic Protocols

We now turn to a class of approaches that do not use the exact physical position to route packets. Epidemic routing was originally designed for ad-hoc networks the were not connected or only partially connected. The nodes within the network have to rely on the random motion of their neighbors to deliver the message to its final destination. Essentially, it works like this: node that want to send messages "infect" the other nodes in the network with their messages. These nodes act as carriers, infecting other nodes along the way. When the destination node receives the message, it sends out a "cure" rather than being infected. The cure is then carried, acting to both acknowledge the delivery of the message and stop the spread of the "infection."

Such an approach has the potential for a large delay within the network, so the protocols much be tolerant of such delays and constant disruption [Harras and Almeroth 2006]. These protocols trade bandwidth, space, and short delivery times for a larger packet delivery ratio.

These protocols also face a unique challenge in dense networks. By not being position aware, Epidemic protocols rely more heavily on beaconing to keep track of their neighbors. Combined with the fact with most of them rely on flooding the network and using broadcast rather than unicast to send messages, these networks are more prone to causing a broadcast storm, where a node us unable to assess or respond all the messages it receives, causing it to drop any new message . Besides causing a broadcast storm, flooding also presents a security risk that needs to be addressed. [Hsiao et al. 2011] discusses the problem in greater detail, along with ways to address the security concern.

As solving a DTN routing problem optimally is NP-Hard [Balasubramanian et al. 2007], these algorithms are only a subset of the many solutions available.

3.5.1 *Epidemic Routing.* Epidemic Routing (Epidemic) [Vahdat et al. 2000] uses a fairly simple approach to routing. If some node wishes to send a message, it generates a uniques ID and then communicates with its neighbors, who store the message. Each neighbor will do the same in turn with their neighbors that do *not* have a copy. More specifically, whenever a pair of nodes encounter each other, the node with the smaller identifier communicates with the other, each obtaining a copy of a message the other has but they don't. Essentially, all nodes are "infected" with the message until the message reaches the intended destination, a which point the message is to be deleted, the nodes being "cured" of the infection. [Pereira et al. 2011]

Epidemic has a significant aspect: given unlimited resources, Epidemic can deliver 100% of its messages. Among four different protocols compared for maritime networks, Epidemic performed the best in terms of percentage of packets delivered, outperforming AODV with a 48% packet delivery ratio vs. 5% [Lin et al. 2010]. However, Epidemic will waste bandwidth and storage with the unfiltered proliferation of the messages.

3.5.2 *Spray and Wait.* Spray and Wait [Spyropoulos et al. 2005] has two phases. In the first, called *Spray*, the source message copies its message to each neighbor. It then does the same for each new neighbor until some hard maximum number of

copies L is reached. Each neighbor that receives a copy enters the *Wait* phase (as well as the source after leaving *Spray*). While in *Wait*, each node attempts to use Direct Routing to pass the message to the destination. It can be assumed that each node is "cured" in the same manner as Epidemic routing after the message reaches its destination.

The overall goal is get the delivery ratio of Epidemic's replication routing without the associated overhead cost. Simulations for maritime networks showed the performance in terms of packet delivery to be far superior to AODV due to the replication and the delay tolerance built into the protocol [Lin et al. 2010]. In some instances of web-browsing and FTP applications, Spray and Wait outperforms Epidemic [Dias et al. 2011] [Isento et al. 2011].

3.5.3 *PRoPHET.* Part of Epidemic's weakness is its generality. Epidemic was created to deal with networks using completely random motion. This is part of the reason why Epidemic spreads the message without restraint, causing huge amounts of bandwidth to be wasted. PRoPHET [Lindgren et al. 2004], short for Probabilistic Routing Protocol using History of Encounters and Transitivity, attempts to limit these unnecessary replications by exploiting the fact that motion is rarely entirely random.

Under PRoPHET, when two nodes meet, they share information about what messages they are carrying. Along with this information is a metric called *delivery predictability* for each destination, which estimates how likely the node will encounter that destination and deliver a message to it. Rather than exchanging all copies, a node will only give a replica of a message to its neighbor if that neighbor has a higher *delivery predictability* than it. The *delivery predictability* of some node A delivering a message to destination D is determined by three factors

—How often A encounters D.

—How long ago did A last see D.

—How often does A encounter nodes that frequently encounter D.

Tests showed that while PRoPHET and Epidemic experience similar delays, PRoPHET is much more efficient even in completely random scenarios. PRoPHET creates a much lower overhead and has a comparable delivery rate. PRoPHET is also better in networks where the queues grow to large sizes.

3.5.4 *RAPID.* Any attempt to minimize the overhead of a Epidemic-based protocol is based off the question "Should this node create a replica of the message?" PRoPHET answers this question by asking if the node has a higher probability of delivering the message. Resource Allocation Protocol for Intentional DTN (RAPID) [Balasubramanian et al. 2007] asks if the cost of creating a replica of the message outweighs the utility of creating a message. RAPID treats routing as a resource allocation problem in an attempt to efficiently rout a packet through a network.

A utility function is created based on some metric, such as average delay. When a node has the option of infecting another node, it will infect the node with the messages that will result in minimizing the average delay.

Experimental results show that RAPID greatly outperforms both Epidemic and Spray.

3.5.5 *PRoPHETv2.* After much testing, it was found that PRoPHET had problems dealing with certain situations. A deployment in the Swedish mountains found that the transitive property of PRoPHET failed when the frequency of encounters was not spread equally across the network, but were frequent enough to not decrease the *delivery predictability* due to age.

Another issue would arise with the *delivery predictability* if a cluster of nodes repeatedly exchanged information. This happened in another deployment where the algorithm interpreted disconnection and reconnection of poor wireless connections as new encounters with the node. PRoPHETv2 [Grasic et al. 2011] has a modified equation to calculate the transitive property that takes into account the expected interval between connections and reduces the amount the *delivery predictability* is modified by recently re-encountered nodes.

Tests showed that PRoPHETv2 outperformed PRoPHET and had a delivery ratio equal to that of Epidemic. However, PRoPHETv2 had a higher overhead than PRoPHET in one of the tests, and both had a much higher overhead than Spray and Wait.

3.5.6 *BBR.* While protocols such as PRoPHET and RAPID are excellent to reduce the overhead of the network, they are not effective in lower density, partitioned networks. Any metric is rendered worthless if the information to calculate it cannot be obtained. This type of sparse network is characteristic of a rural area. Border Based Routing (BBR) [Zhang and Wolff 2008] was designed for use in this type of situation.

Nodes within BBR use beaconing to discover their one-hop neighbors. When a node transmits a message, its neighbors decide whether or not they are *border nodes* for that event. In other words, the neighbors decide if they will be infected. This is done independently via a distributed algorithm and exchanged nearest neighbor lists. If the data indicates a node shares the least number of common neighbors with other nodes, it is ideal to be a border node.

Tests showed that BBR works well in a partially connected network and frequently partitioned networks.

4. FUTURE WORK

One on the big challenges now is figuring out how each type of protocol can leverage each other's advantages. Part of the difficulty in implementing delay tolerance within many protocols is the need to add an additional protocol, such as the Bundle Protocol [Scott and Burleigh 2007], to have that functionality. [Nordemann and Tonjes 2012] proposes a "store-carry-forward" mechanism that is transparent and autonomous. Rather than sending an ICMP message due to being unable to deliver the packets, the packets are sent to piggyback software. This software implements the "store-carry-forward" procedures without any changes to the underlying protocol. Applying context-awareness to delay tolerant networks has shown significant gains [Petz et al. 2012].

In fields other than automobiles, many vehicles make use of legacy and outdated technology, requiring the creation of intelligent middle-ware to successfully integrate modern vessels with aging satellites [Boreli et al. 2009].

Security is still a huge issue that needs to be researched in VANETs due to the

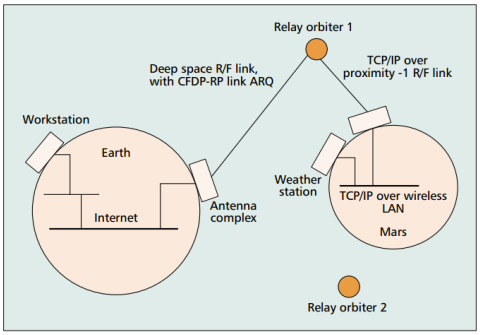

Fig. 2.

completely wireless nature of the connections. One issue is the authenticity of the message's source is verified, but not the nodes involved in routing[Pereira et al. 2011].

A. CHALLENGED NETWORKS AND TCP/IP

The TCP/IP model has become so ubiquitous within the Internet that it is all but impossible to separate the two. There are a number of unwritten assumptions when it comes to TCP [Fall 2003] [Burleigh et al. 2003] that do not apply to many networking scenarios.

Consider the following scenarios:

—A scientist wishes to communicate with a deployed station on Mars (robotic or manned makes no difference to this scenario). To do so, he needs to send his message to a large antenna relay to talk with some satellite. That satellite with have to communicate with other satellites, which may not be into range, before finally communicating with the station on Mars. To communicate back to Terra, the station will need to follow the same steps in reverse. This is shown by Figure 2.

—Multiple low-cost, solar powered consoles are deployed in isolated rural ares. These consoles have no permanent connection to the Internet, yet provide Internet based services such as email or market prices on crops. Requests generated by the users need to be held until they can be transmitted to one of the mobile nodes that act as carriers for messages, such as buses equipped to carry these messages (in addition to passengers). This example is the premise of Kiosknet [Guo et al. 2007].

The standard repertoire of protocols that are normally used to keep the Internet functioning are ill suited for the requirements demanded by these challenged networks. Focusing on the first example, a signal traveling at the speed of light[10] will take 10 to 20 minutes [Laboratory]. This is a huge delay to deal with using any protocol, but TCP requires a three-way handshake in order to validate. Under the

[10]299,792,458 meters/second. Messages over Ethernet travel at approximately 2/3 this rate.

best of circumstances, it will take 30 minutes before the scientist can start transmitting his data. This issue becomes even more grim when we take into account disconnects (it will take another three-way handshake - another 30 minutes).

TCP also only delivers data in the order of transmission. On other words, if some part of a transmission is lost, the data following the lost part will have to wait until the data has been retransmitted (another round time trip).

BGP is built to use TCP to determine paths, so BGP is also out of the picture. Even if BGP didn't rely heavily on TCP, the routing algorithms would draw incorrect conclusions from disconnects in the network, particularly scheduled and predictable disconnects.

This is not an argument to stop using TCP/IP. However, within the realm of delay tolerant networks and challenged networks, it is unable to provide the necessary services.

B. FORWARD ERROR CORRECTION

In disconnected networks that already deal with hefty delays, it is absolutely essential to minimize delays caused by transmissions errors. Theses can typically be compensated with replication and an error correction mechanism. [Altman and De Pellegrini 2011] gives an detailed mathematical analysis of the various trade offs involved.

[Sathiamoorthy et al.] presents how fountain codes can be adjusted for storage within VANETs. By distributing coded chunks of a file throughout a bandwidth-constrained network, a node searching for a particular file effectively sees each other node he encounters possessing a complete copy of the original file.

C. CLASSIFICATIONS

Table I. Routing Protocols

Protocol	Approach	Summary	Advantages	Weaknesses	Use
DSR [Johnson and Maltz 1996]	Ad-hoc, Dynamic, Source Routing	Dynamically creates end-to-end route for each message	Simple, low overhead in low-mobility networks	Long paths decrease efficacy, does not utilize position	General wireless MANETs Useful
AODV [Perkins and Royer 1999]	Ad-hoc, dynamic, distance vector	Updates routing tables in nodes for each desired end-to-end communication	avoids loops	route instability, weak with disconnected networks, long paths decrease efficacy	Proposed for MANETs, sometime implemented for VANETs

continued on the next page.

continued from the previous page.

Protocol	Approach	Summary	Advantages	Weaknesses	Use
GSR [Lochert et al. 2003]	Position Aware, Greedy	Uses street maps with node positions calculate the route along road junctions.	Better than DSR and AODV, works in urban environments.	Neglects velocity and will not always select the *best* path	Urban and highway VANETs.
GPSR [Karp and Kung 2000]	Position Aware, Greedy	Uses greedy forwarding with perimeter forwarding for recovery	Works very well in open areas. Can be efficient	Fails to take into account velocity, can cause loops, suffers in urban environs	non-urban VANETs
A-STAR [Seet et al. 2004]	Position Aware, Traffic Aware, Greedy	Like GSR, but uses bus routes to weight best junction path. Recovers by computing new path.	Much better performance than GSR, takes into account traffic.	Fails to take into account velocity	Urban VANET
GyTAR [Jerbi et al. 2006]	Position Aware, Traffic Aware, Greedy	Selects junctions dynamically at each hop. Greedy forwarding based on future position. Recovery via "carry and forward."	Factors in velocity. Less overhead, better performance than GSR.	Only gathers traffic information within one-hop	Urban VANET
HTAR [Lee et al. 2011]	Position Aware, Traffic Aware, Network Load Aware	Disseminates traffic info to all nodes. Routes chosen by physical and network traffic.	Aware of both network and physical congestion. Accounts for multiple attributes.	Increased overhead, not suited for DTN due to "verbosity" of communication	Urban VANET

continued on the next page.

continued from the previous page.

Protocol	Approach	Summary	Advantages	Weaknesses	Use
FFRDV[Yu and Ko 2009]	Position Aware, Greedy	Forwards to nodes headed to the destination the fastest.	Minimizes the number of hops.	Limited number of destinations.	Urban delay tolerant VANET for data collection or road alerts.
Epidemic [Vahdat et al. 2000]	Flooding, Epidemic	Each node gets sends a copy of the message(s) to any node it encounters that doesn't already have.	Extremely simple. Given unlimited resources, will delivery every messages.	Extremely resource heavy, wastes space, wastes bandwidth	General DTN protocol
Spray and Wait [Spyropoulos et al. 2005]	Epidemic	Source infects some maximum number of copies. Copies use Direct Routing to deliver message	Extremely simple. Lowest overhead.	Poor delivery ratio.	General DTN protocol
PRoPHET [Lindgren et al. 2004]	Epidemic	Like Epidemic, but only infects nodes that can better get to destination.	Same delivery, but better overhead than Epidemic in most scenarios.	Transitive property fails in some scenarios.	General DTN protocol
PRoPHETv2 [Grasic et al. 2011]	Epidemic	PRoPHET with an adjusted metric for nodes that encounter nodes that reach the destination	Same delivery, but better overhead than Epidemic.	Worse overhead than PRoPHET in some tests.	General DTN protocol
RAPID [Balasubramanian et al. 2007]	Epidemic	Views routing as a resource allocation problem. Like Epidemic, but only infects nodes if the benefit outweighs the cost.	Outperforms Epidemic and Spray and Wait	Complex	General DTN protocol

continued on the next page.

continued from the previous page.

Protocol	Approach	Summary	Advantages	Weaknesses	Use
BBR [Zhang and Wolff 2008]	Epidemic	Sends to nodes that share the least number of neighbors.	Good in sparse and rural areas with partitioned networks	Poor performance in dense networks.	Rural VANET DTN protocol

REFERENCES

Altman, E. and De Pellegrini, F. 2011. Forward correction and fountain codes in delay-tolerant networks. *IEEE/ACM Trans. Netw. 19,* 1 (Feb.), 1–13.

Balasubramanian, A., Levine, B., and Venkataramani, A. 2007. Dtn routing as a resource allocation problem. In *ACM SIGCOMM Computer Communication Review*. Vol. 37. ACM, 373–384.

Boreli, R., Ge, Y., Iyer, T., Dwertmann, C., and Pathmasuntharam, J. 2009. Intelligent middleware for high speed maritime mesh networks with satellite communications. In *Intelligent Transport Systems Telecommunications,(ITST), 2009 9th International Conference on*. IEEE, 370–375.

Burleigh, S., Hooke, A., Torgerson, L., Fall, K., Cerf, V., Durst, B., Scott, K., and Weiss, H. 2003. Delay-tolerant networking: an approach to interplanetary internet. *Communications Magazine, IEEE 41,* 6, 128–136.

Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., and Weiss, H. 2007. Rfc 4838, delay-tolerant networking architecture. *IRTF DTN Research Group*.

Dias, J., Isento, J., Pereira, M., Silva, B., Soares, V., Rodrigues, J., Pereira, P., Casaca, A., Cervelló-Pastor, C., and Gallego, J. 2011. Www vdtn-a web browsing application for vehicular delay-tolerant networks. In *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2011 IEEE 16th International Workshop on*. IEEE, 11–15.

Fall, K. 2003. A delay-tolerant network architecture for challenged internets. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 27–34.

Füssler, H., Transier, S., and Effelsberg, W. Vehicular ad-hoc networks: From vision to reality and back.

Gil-Castineira, F., Gonzalez-Castano, F., and Franck, L. 2008. Extending vehicular can fieldbuses with delay-tolerant networks. *Industrial Electronics, IEEE Transactions on 55,* 9, 3307–3314.

Gongjun, Y., Mitton, N., Li, X., et al. 2010. Reliable routing in vehicular ad hoc networks.

Grasic, S., Davies, E., Lindgren, A., and Doria, A. 2011. The evolution of a dtn routing protocol-prophetv2. In *Proceedings of the 6th ACM workshop on Challenged networks*. ACM, 27–30.

Guo, S., Falaki, M., Oliver, E., Ur Rahman, S., Seth, A., Zaharia, M., and Keshav, S. 2007. Very low-cost internet access using kiosknet. *ACM SIGCOMM Computer Communication Review 37,* 5, 95–100.

Harras, K. and Almeroth, K. 2006. Transport layer issues in delay tolerant mobile networks. *NETWORKING 2006. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*, 463–475.

Hsiao, H.-C., Studer, A., Chen, C., Perrig, A., Bai, F., Bellur, B., and Iyer, A. 2011. Flooding-resilient broadcast authentication for vanets. In *Proceedings of the 17th annual international conference on Mobile computing and networking*. MobiCom '11. ACM, New York, NY, USA, 193–204.

Isento, J., Dias, J., Neves, J., Soares, V., Rodrigues, J., Nogueira, A., and Salvador, P. 2011. Ftp vdtna file transfer application for vehicular delay-tolerant networks. In *EUROCON-International Conference on Computer as a Tool (EUROCON), 2011 IEEE*. IEEE, 1–4.

Jerbi, M., Meraihi, R., Senouci, S., and Ghamri-Doudane, Y. 2006. Gytar: improved greedy traffic aware routing protocol for vehicular ad hoc networks in city environments. In *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*. ACM, 88–89.

Johnson, D. and Maltz, D. 1996. Dynamic source routing in ad hoc wireless networks. *Mobile computing*, 153–181.

Karp, B. and Kung, H. 2000. Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM, 243–254.

Laboratory, N. J. P. Mars curiosity rover: Faq. http://www.jpl.nasa.gov/msl/curiosity/index.cfm?page=faq.

Lee, J.-W., Lo, C.-C., Tang, S.-P., Horng, M.-F., and Kuo, Y.-H. 2011. A hybrid traffic geographic routing with cooperative traffic information collection scheme in vanet. In *Advanced Communication Technology (ICACT), 2011 13th International Conference on*. 1496 –1501.

Li, F. and Wang, Y. 2007. Routing in vehicular ad hoc networks: A survey. *Vehicular Technology Magazine, IEEE 2,* 2 (june), 12 –22.

Lin, H., Ge, Y., Pang, A., and Pathmasuntharam, J. 2010. Performance study on delay tolerant networks in maritime communication environments. In *OCEANS 2010 IEEE-Sydney*. IEEE, 1–6.

Lindgren, A., Doria, A., and Schelen, O. 2004. Probabilistic routing in intermittently connected networks. *Service Assurance with Partial and Intermittent Resources*, 239–254.

Lochert, C., Hartenstein, H., Tian, J., Fussler, H., Hermann, D., and Mauve, M. 2003. A routing strategy for vehicular ad hoc networks in city environments. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*. Ieee, 156–161.

Nordemann, F. and Tonjes, R. 2012. Transparent and autonomous store-carry-forward communication in delay tolerant networks (dtns). In *Computing, Networking and Communications (ICNC), 2012 International Conference on*. IEEE, 761–765.

Pereira, P., Casaca, A., Rodrigues, J., Soares, V., Triay, J., and Cervello-Pastor, C. 2011. From delay-tolerant networks to vehicular delay-tolerant networks. *Communications Surveys Tutorials, IEEE PP,* 99, 1 –17.

Perkins, C. and Royer, E. 1999. Ad-hoc on-demand distance vector routing. In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA'99. Second IEEE Workshop on*. IEEE, 90–100.

Petz, A., Hennessy, A., Walker, B., Fok, C., and Julien, C. 2012. An architecture for context-aware adaptation of routing in delay-tolerant networks.

Sathiamoorthy, M., Dimakis, A., Krishnamachari, B., and Bai, F. Distributed storage codes reduce latency in vehicular networks.

Scott, K. and Burleigh, S. 2007. Bundle protocol specification.

Seet, B., Liu, G., Lee, B., Foh, C., Wong, K., and Lee, K. 2004. A-star: A mobile ad hoc routing strategy for metropolis vehicular communications. *NETWORKING 2004. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications*, 989–999.

Spyropoulos, T., Psounis, K., and Raghavendra, C. 2005. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. ACM, 252–259.

Vahdat, A., Becker, D., et al. 2000. Epidemic routing for partially connected ad hoc networks. Tech. rep., Technical Report CS-200006, Duke University.

Wisitpongphan, N., Bai, F., Mudalige, P., Sadekar, V., and Tonguz, O. 2007. Routing in sparse vehicular ad hoc wireless networks. *Selected Areas in Communications, IEEE Journal on 25,* 8, 1538–1556.

Yu, D. and Ko, Y. 2009. Ffrdv: fastest-ferry routing in dtn-enabled vehicular ad hoc networks. In *Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on*. Vol. 2. IEEE, 1410–1414.

Zhang, M. and Wolff, R. 2008. Routing protocols for vehicular ad hoc networks in rural areas. *Communications Magazine, IEEE 46,* 11, 126–131.