Research article Available online www.ijsrr.org ISSN: 2279–0543

International Journal of Scientific Research and Reviews

An Undependable Interehicular Routing Procedure For V e h i c u l a r ad Hoc Networks

Karthika N.
Department of Computer Technology,KG College of Arts and Science, Saravanampatti, Coimbatore-641035.
Email: karthikaraj.n@gmail.com Mobile: +91 8220261381

ABSTRACT

Vehicular accidental NET works (VANETs), associate degree rising technology, would enable vehicles on roads to create a self-organized network while not the help of a permanent infrastructure. As a requirement to communication in VANETs, associate degree efficient route between act nodes within the network should be established, and also the routing protocol should adapt to the rap- lazily dynamical topology of vehicles in motion. This is one amongst the goals of VANET routing protocols. During this paper [3], we have a tendency to gift associate degree efficient routing protocol for VANETs, known as the An Undependable Interehicular Routing Procedure. Watercourse utilizes associate degree a drift graph that represents the encircling street layout wherever the vertices of the graph square measure points at that streets curve or ran into, and also the graph edges represent the road segments between those vertices. In contrast to existing protocols, watercourse performs period of time, active traffic monitoring and uses these knowledge and alternative knowledge gathered through passive mechanisms to assign a dependableness rating to every street edge. The protocol then uses these dependableness ratings to pick out the foremost reliable route. Management messages square measure wont to establish a node's neighbours, verify the dependableness of street edges, and to share street edge dependableness info with alternative nodes.

KEYWORDS – Routing protocol, Traffic monitoring, Active, passive monitoring.

***Corresponding author**

KARTHIKA N.

Assistant Professor, Department of Computer Technology,

KG College of Arts and Science, Saravanampatti, Coimbatore-641035.

Email: karthikaraj.n@gmail.com

Mobile: +91 8220261381

INTRODUCTION

The vehicular ad hoc network (VANET) provides the facility for vehicles to instinctively and wirelessly net- work with other vehicles nearby for the purposes of providing travellers with new features and applications that have never been previously possible. Within this ever changing network, messages must be passed from vehicle to vehicle in order to reach their intended destination. To participate in such a network, a routing protocol must direct these message transfers in an efficient manner to make sure robust data communication. Discuss various design factors of VANET protocols, survey a number of VANET routing protocols, and presented an analysis of them.

As a special category of mobile ad hoc networks, VANETs have their own distinctive characteristics that distinguish them as a set of this larger category. Most nodes in an exceedingly VANET are mobile, however as a result of vehicles are usually unnatural to roadways, they need a definite controlled quality paltered that's subject to conveyance traffic rules. In urban areas, gaps between roads are typically occupied by buildings and different obstacles to radio communication, therefore routing messages on roads is usually necessary

MOTIVATION

An elementary facet of the success of any VANET is that the presence of a sufficient variety of network nodes to permit forwarding of messages within the network. Road characterise- tics like traffic signals and stop signs have an effect on the flow of traffic in urban areas, breaking any sufficiently dense streams of similar velocity vehicles. Traffic density, mea- sured in the quantity of vehicles per unit distance, has an oversized influence on road capability and vehicle rate. Messages in a VANET ar forwarded on streets thanks to the distinctive constraints of this type of network. However, thanks to numerous factors in a real-world state of affairs, there is no guarantee that network-participating vehicles ar gift on any specific street at a given time. an absence of net- worked vehicles might occur thanks to factors like date and time, building, detours, community events, traffic laws, and poor road conditions due to weather. Some of those factors have an effect on all streets in a very specific space, whereas alternative factors might cause solely a couple of chosen streets to be void of network nodes

The seminal VANET protocols like galvanic skin response and STAR [13]did not take traffic factors into consideration. A-STAR utilised static traffic info from bus schedules. The designers of A-STAR hypothesized that buses travel on ma- jor thoroughfares that square measure additional

probably to possess dense vehic- ular traffic. A-STAR was thus programmed to like these roads for forwarding. Alternative strategies of traffic mon- itoring thought of to be static approaches could embody caching ‘‘typical’’ traffic knowledge and doubtless supplement- ing that knowledge with updates regarding less-frequently regular traffic conditions. For example, nodes may store knowledge regarding typical traffic patterns like rush-hour commuter traffic on weekdays, and then they could additionally receive peril- ode updates regarding building or community events that disrupt these typical patterns.

While typical traffic patterns might persist for a signifi- cant quantity of your time, it is quite probable that temporary gaps in network coverage area unit common on most streets at frequent intervals. If distance between a node and it’s near- Eastern Standard Time neighbour is larger than the transmission ranges of each of them, it causes a network gap. These forms of gaps might occur overtimes due to traffic signals that stop conveyance traffic, for example. They may additionally be caused even once the road is full of vehicles if several of the vehicles area unit not network-equipped. These temporary gaps are {often|will be|is|may be} very unquiet as a result of they often happen in a non-deterministic manner. A typical network gap is de- picted in Fig. one wherever conveyance traffic on a street is moving off from one another, therefore partitioning the network.

Temporary gaps in network are common on most streets at frequent intervals. The use of static knowledge alone cannot adapt to dynamically ever-changing network gaps. A time period approach is needed, and a few protocols have tried this to variable degrees. STAR monitors the

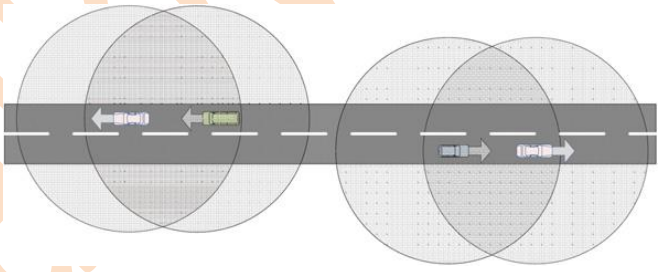

Figure1. Formation of A Network Gap.

Number of nodes it encounters in every of the cardinal and intercardinal directions relative to every node to aid in routing choices. every node in automobile adapts its beaconing interval to the amount of neighbouring nodes it has detected thus that beacons do not saturate network band-breadth in dense traffic conditions. SADV measures message delivery delays to estimate traffic densities.

Like the edges of a graph, road segments between inter- sections ar one-dimensional in terms of communication: messages will be sent either to vehicles sooner than the cur- rent node or to vehicles behind it. As such, the bulk of routing choices ar created at intersections, referred to as an- chord points. These choices are crucial: causing a message down a street that contains a network gap causes the mes- sage to either be born, buffered, or to go back. With these factors in mind, it becomes clear that the shortest path between a sender and receiver isn't forever the foremost thriving path since one disconnected street phase can cause a strictly shortest-path routing to fail. Instead, a VANET routing protocol should have a technique to work out that street edges ar presumably to lead to delivery of a packet to ensuing intersection.

These observations lead U.S.A. to a position-based VANET [5] routing protocol that utilizes period of time traffic data to come up with a route that travels on a reliable path (a path that is a smaller amount doubtless to contain network gaps), notwithstanding such a path isn't the shortest path during a geographic sense. The remainder of the paper is organized as follows: Section three introduces the essential plan behind our protocol. Section four presents the traffic observance element of the protocol. In Section five, we tend to gift however our rule calculates the dependableness of the perimeters in the road graph. In Section vi, we tend to gift our routing rule thoroughly. Section seven contains the per- formance analysis results and Section eight concludes the paper.

TRAFFIC MONITORING

Traffic observation in our protocol consists of each ac- tive and passive elements that operate in period of time. For active traffic observation, the first mechanism is that the probe message: a stream protocol packet that's periodically sent by every node in the network. Probes perform twin functions of traffic detection and traffic info distribution. Additionally, every node performs passive traffic observation by gathering knowledge from every packet that it receives. Probe and routing packets carry 2 different types of traffic information: the renowned edge list and weighted routes.

Active monitoring

In VANETs, beacon messages primarily function a mechanism for a node to advertise its existence to its neighbours. In a sense, this can be a sort of traffic awareness. Beacon-oriented traffic observance is used by a number of the routing protocols that have created restricted use of time period traffic observance, like STAR [6] and automobile . How- ever, a node will solely notice beacons emanating from nodes among its radio vary, and often, the reliable vary of a radio is also less than the gap between street intersections.

To determine if a message will be delivered on a particular street edge to consequent intersection, stream uses a pursuit message. a pursuit is best delineate as associate any cast message: it's sent to any node in a very cluster of nodes defined by a selected geographical area. Its content is comparable to a beacon message in that it doesn't carry a knowledge payload. However, probe messages square measure not one-hop broadcast messages.

Each node maintains a replica of the encompassing street layout in its street graph wherever every road phase is rep- resented by a position within the graph, incident on 2 vertices. a research message is distributed by a node that's situated close to a street vertex (within fifty m), and it is forwarded covetously to supposed next-hop recipients on the streets that square measure incident to that vertex. The destination node of a probe message isn't renowned to its sender; the probe traverses a street edge and is finally received by any node at intervals varying of the alternative street vertex. If there's a spot in the network coverage on the street edge, the probe is born. However, if the probe is delivered to its destination vertex, any nodes at that vertex become aware that the vertex is passable at that moment. Once a outward-bound probe is received, a comet probe is generated back to its original sender therefore that the sender can conjointly be aware of the property of the probed street edge.

Our protocol's probe messages act as implicit beacons for every forwarding node by as well as every forwarder's geographic position and address. They conjointly carry the ad- dress and geographic position of their original sender, and also the position of their destination vertex.

Passive monitoring

Each node conjointly monitors edge property by passively snooping into routing packets that square measure sent at intervals the net- work. Every message contains, either implicitly or expressly, dependability info concerning edges in the network. These monitored messages are also messages that square measure sent directly to a node as a next-hop or destination. However, every node conjointly faucets into the link layer of its network stack and listens for stream packets that square measure self-addressed to a different node. The learned dependability info is then shared at intervals the network during a distributed manner.

On this version, routing is aided via collecting and dispensing understanding concerning the connectivity of edges in the street graph. This is in part enabled thru passive tracking. Whenever a node close to a avenue vertex vx [7]gets a packet that has traversed an area that is incident on vx, this means that the traversed facet is presently connected. (by way of linked, we suggest that sufficient nodes are present along the edge to transmit a message alongside that area.) Similar to the probe

mechanism described in advance, our routing packets additionally contain facts that permits a node to determine the reliability of the rims traversed via a packet. therefore, whilst node nx near avenue vertex vx receives a probe or routing packet that has traversed an edge incident to vx, node nx resets the burden of that facet in its avenue graph to the minimal cost, which indicates that the traversed side is connected.

Passive monitoring also enables a node to learn about edges of the street graph that can be a ways away from the node. as depicted in fig. 2, suppose a node gets a rout- ing packet from node. the node is already aware about the reliability of edges near it because it sends and receives probe packets along the ones edges (marked with an ‘‘x’’ in the figure). in addition, every facet in the routing packet’s path (marked with a ‘‘y’’ inside the figure) might be rep- resented with an facet weight in the packet. subsequently, any edges incident on the direction will likely also have their relipotential captured due to the fact the nodes that ahead the packet from the supply to the destination may also add into the packet any reliability weights known to them also (marked with a ‘‘z’’ in the figure) in the acknowledged side listing. These features will be defined in addition on this segment.

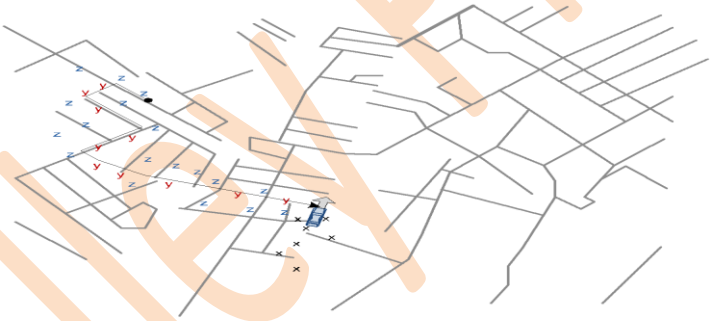

Figure2. Data Gained From Passive Monitor Of A Routing Packet.

Further to gathering traffic records from packets which might be immediately received by way of a node, every node also eavesdrops on the radio transmissions between other close by nodes. for example, probe and routing packets are forwarded to a septic recipient at each hop. With the aid of default, different nodes within radio range of sender discard the packet at the hyperlink layer of their protocol stack. However, statistics contained inside those probe and routing packets incorporates value for different nodes in the area besides their meant recipients. with the intention to perform passive traffic tracking, every node faucets into the hyperlink layer of its network stack. by means of eavesdropping at this level, any river probe and routing packets which can be no longer addressed to the present day node can be pushed up the protocol stack for processing.

Weighted routes

Each routing packet incorporates a listing of anchor points for the route, identified by their relocation. Any two consecutive route anchor factors in the list constitute an part in the street graph of the sender node and has an edge weight related to it. Whilst constructing the routing packet, the sender includes this aspect weight in the packet, in conjunction with a timestamp which represents time whilst that reliability cost changed into remaining up to date.

While a routing packet is acquired at a node, the node analyzes the path and procedures the reliability informa- tion associated with it. if the node is not the final recipient for this routing packet, it also updates the reliability infor- mation in the direction packet prior to forwarding it. the regulations in section five.2 govern the processing of incoming reli- ability facts and updating of outgoing reliability information.

Known edge list

Each node monitors beacon, probe, and routing mes- sages, each of which incorporates a regarded-area list (kel). the acknowledged-part listing identifies edges by means of their endpoint geolocations and communicates reliability facts about each facet (e.g. the ‘‘z’’-marked edges depicted in fig. 2) alongside the direction. upon sending a river packet, the sending node selects edges from its personal street graph to share with other nodes, and places them in the regarded- area list with their reliability values and the time whilst each reliability cost turned into closing updated. likewise, each time a river packet is received at a node, the node reads the regarded-part listing and methods any edge reliability values observed there. if the packet is a probe or routing packet that the node will forward on, the node selects edges to percentage from its street graph (which now consists of the statistics contained in the received kel) and updates the acknowledged- edge list in the packet earlier than sending it on.

EDGE RELIABILITY

A important element of our protocol is its potential to estimate the reliability of a particular road facet. river uses this reliability records as the primary component in figuring out a a hit routing path from a sender node to a receiver node. Vehicular nodes flow fast and regularly, so it is infeasible for each node to music the motion of all other nodes across a specific location to decide usable routes. Alternatively, we hypothesize that it is greater efficient to determine if a unique avenue part became reliable currently and proportion this data with other nodes.

Determining reliable paths

Every node within the river version assigns a weight to every recognized part in its avenue graph. to determine dependable paths, the protocol assigns those weights the use of both first- hand commentary and 2d-hand understanding. first-hand observations consist of the statistics that each node profits when it gets a packet or when it tries to ship a probe or routing message to any other node. 2d-hand observations include the passive tracking of regarded- side lists stored in beacons, probes, and routing packets, and the tracking of aspect weights contained inside rout- ing messages.

In shortest-route routing algorithms, every facet weight would be primarily based on the duration of the street segment repre- sented via the threshold. our protocol is no longer a shortest-course routing algorithm in this sense; its edges are weighted with their reliability rating. a small weight (the minimum weight is zero) shows extra reliability; a huge weight shows an unreliable part, and the most weight suggests an facet that is known to be not traversable. With those weights assigned to every aspect, our protocol uses Dijkstra's least weight route set of rules to calculate what it considers the maximum dependable routing path. The direction, together with every reliability score used inside the calculation, is writ- ten into the packet.

Once the use of reliability as a path metric, dis- tance (in phrases of the wide variety of edges in a direction) is still ta- ken under consideration. dijkstra's least weight route algorithm finds a route with least-weight based on the sum of the weights of edges on the route. if two paths px and py have equal weights on each facet but px has greater edges (is a longer course) than py, then py is selected because its total weight is much less. The shorter of the two paths is chosen.

Reliability distribution

While a node sends a beacon, probe, or routing packet that carries a regarded-edge listing, that node distributes its street graph reliability information inside the packet. for clarity here, we define an edge's reliability rating as shared whilst a node writes the brink's reliability score right into a p.c.- et's regarded-facet list for distribution. we define an facet's reliability score as declared when a node reads this rating from a regarded-aspect listing in a packet that it has received. in addition to the reliability rating, every node additionally tracks other values relative to each facet in its street graph, proven in desk 1; different essential facts factors calculated with re- spect to every edge in the road graph are proven in table 2. those values are used to make some of choices approximately edges, calculate the reliability of every edge, and to deter- mine while a declared cost have to be used or discarded.

In an attempt to conserve community bandwidth, a node does no longer without a doubt write all of its recognized-facet data into every packet it sends. Edges whose reliabilities are un-

regarded (and set to a default price) are now not shared. from the last edges, a node selects an area for sharing based totally on several standards: whether or not it has been up to date since the last time it changed into shared, how current the replace was, and whether or not the update originated from first-hand remark or a 2nd-hand declared value. the maximum selective element is whether or not the brink has been up to date for the reason that last time it turned into shared: facts about an part is shared simplest if this condition is real. past that, edges are ranked relative to one another for ‘‘shareability’’. an edge that was up to date more these days is preferred over an side that turned into up to date much less currently, so a relative shareability rank- ing is given to every area based totally on the time that has elapsed seeing that its last update.

Whilst a node gets declared records approximately the reliability of an part, it have to decide whether or not to take delivery of or reject the declared value based totally on the timestamp associated with the declared fee and the timestamp information the node buddies with its modern part rating. if a node has no reliability statistics for an edge from any supply (receiving a packet over the brink, marking the brink unreliable in the past, or from a prior declaration of the brink), then it accepts the declared value. if a node already has reliability information for the edge, then it compares the declared timestamp information with its personal ultimate up to date timestamp and accepts the declared rating if the declared timestamp is more current. after the declared va- lue is widely wide-spread, the node units the edge’s closing declared time- stamp to be the timestamp recorded within the packet (now not to the contemporary time whilst the fee is ordinary) and units the static reliability value for the edge to the declared value.

Reliability calculation

Network gaps often emerge and dissolve, so the river protocol discards notions of persistent, static traffic models in want of a extra dynamic model. the transmis- sion of a packet from sender to receiver occurs on a miles shorter time scale than traffic actions, so even a net- work gap that has most effective shaped for some seconds can cause many packets to be dropped or behind schedule. to make certain fewer packet delays, up to date data is most excellent. the freshness of the reliability statistics maintained through a node is crucial to don't forget. Older records is less in all likelihood to reflect truth than current data.

So that you can deliver choice to latest information, whilst first-hand located facts is available, our protocol calculates the reliability of an facet as the number of milli- seconds on the grounds that the brink become remaining recognised to be traversed by a packet. with this model, a

low reliability fee represents a currently-traversed aspect. edges with low values are pre- ferred over edges with excessive values while generating a course.

While a node gets a packet that has traversed some part e, the node sets the reliability price of e to zero (most dependable). as time elapses from that occasion, the reliability va- lue for the threshold decays in a linear fashion to a better (less dependable) value until any other packet traverses the edge. to boost up the decay of an edge that looks to be unreli in a position, a steady ready multiplier (10) is used within the cal- culation. when a node does no longer receive a reaction to a probe message sent alongside an edge, the waiting multiplier is used in the calculation to discourage the use of that side for routing. the waiting multiplier stays in impact for that part until the edge weight is up to date with new infor- mation. any other consistent in no way-acquired multiplier (2) is used in instances where no packet has ever been received alongside the edge.

Further to the dynamically calculated values, there are some static values utilized in river reliability scores. if no packet has been acquired alongside an area (and the node has not sent a probe alongside this edge to test it) for some time, a time out period called the reliability default (10 s) in the end expires. this price, also measured in millisecond, acts as a default price for any edge whose reliability is undetermined. if the reliability information approximately a specific facet is not up to date inside this period, its reliability re- verts to this default value. furthermore, if a node on some aspect attempts to forward a packet alongside that edge but can find no neighbor to whom the packet can be sent, the node immediately marks that side as unreliable via setting it to 1 (represented by the largest fee which can be saved in the statistics range). this unreliable rating is disbursed to other on hand nodes thru the recognised-area list of the packet.

GREEDY OPTIMIZATION

In the strictest sense, forwarding packets on AN an- chor route involves covetously forwarding toward every anchor purpose till the packet arrives at a node that is among some predefined vary of the anchor purpose, known as the vertex vary. However, throughout this method, complexi- ties arise because of the variations between the vertex vary and every node's radio vary and therefore the density of traffic.

Consider Fig. three wherever node metal is forwarding a packet to- ward the anchor purpose at the portrayed intersection. the next anchor purpose for this packet is on the road go up the direction on the far side node Nb. The vertex vary for the present anchor purpose is shown as a circle, and node Nb is that the nighest node to the anchor purpose however continues to be outside the vertex vary among that the anchor purpose is considered ''reached''.

According to greedy forwarding, node metal forwards the packet to the nearer node Nb. once Nb[N] receives the packet, there's still no node nearer to the anchor purpose than node Nb, and Nb continues to be outside the vertex vary. Since the anchor purpose has not however been reached, this can be technically a neighborhood most. Strict greedy routing would dictate that node Nb ought to drop the packet. However, since node Nb is on the road edge that leads to the next anchor purpose in this route, it is prema ture to drop the packet at now. Our protocol contains AN optimisation to handle this situation. once a node re- ceives a routing packet with multiple anchor points remaining in the route, it retrieves the current anchor purpose and therefore the consequent anchor purpose (or the final desti- nation if no additional anchor points exist) for the anchor route. If the node determines that it's situated between those 2 points, it increments the AP pointer in the packet. Thus, watercourse detects once a packet has passed AN anchor purpose, albeit the packet ne'er truly reached it.

A similar situation happens once node metal is nearer to the anchor purpose than node Nb, as in Fig. 4. Here, node metal is that the nighest node to the anchor purpose however continues to be outside the ‘‘reached’’ vary. In a typical greedy algorithmic program, node metal would drop the packet. However, since node Nb is with- in radio vary of node metal, and node Nb is within the direction of the next anchor purpose, dropping the packet may be a poor selection in this case. Our protocol can look for a neighbor nearest to the next anchor purpose specified the neighbor is found on the road edge between the present anchor purpose and therefore the consequent anchor purpose. rather than dropping the packet, node atomic number 11 finds node Nb and forwards to that node. Node Nb detects that the packet has passed the anchor purpose and increments the AP pointer befittingly.

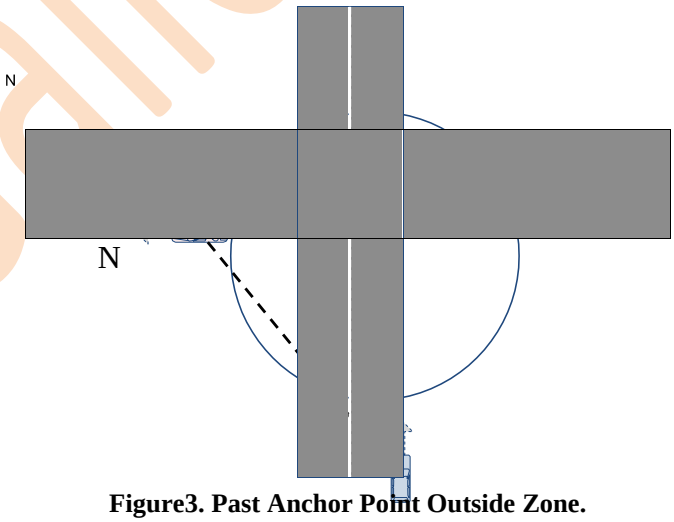

Figure3. Past Anchor Point Outside Zone.

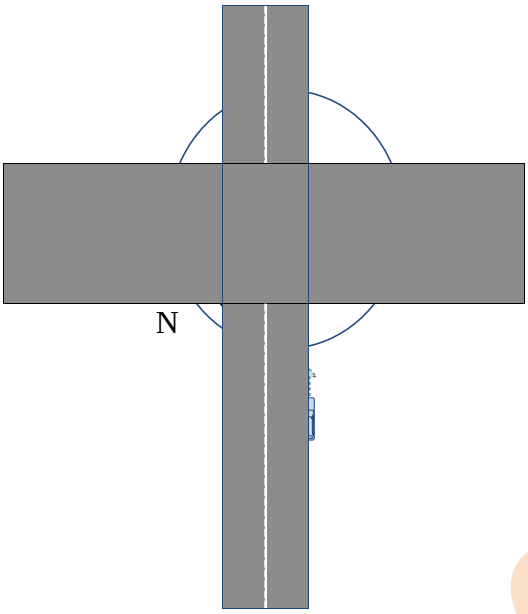

Figure4. Outside zone, no closer neighbour

CONCLUSION AND FUTURE WORK

On this paper, we have proposed ‘‘dependable inter-vehicular routing’’ , a routing protocol for vanets primarily based on expected network reliability. This takes benefit of real-time traffic monitoring using active and passive methods. The protocol is ready to correctly distribute relipotential records for the duration of the vanet the use of recognised facet lists and weighted routes.

In our simulation surroundings, observed that river provides the best throughput in most traffic densities whilst using its restoration strategy, however the recalculation strategy yields better throughput in low traffic density with much less overhead. It located that reliability distribution components perform quality in average to high density scenarios. Those components reason a significant increase in routing header length that may be efficiently negated with the aid of proscribing reliability distribution to beacon and probe packets. we also discovered that river’s optimized grasping forwarding method can significantly growth percent- et throughput with no acknowledged negative outcomes, and this strategy can be carried out to routing protocols that do no longer percentage river’s reliable-route routing approach. Ultimately, simulations confirmed that river plays nicely towards peer protocols – specifically in average to high-density traffic.

Extra improvements to this may additionally yield in addition benefits. Overall performance under low-density traffic was no longer a focal point during the protocol’s layout, so that is an region in which its performance will be stronger. Overall performance evaluation found out that routing header size could be substantially reduced without tons loss of throughput through elim- inating traffic distribution through routing packets. this have to be investigated in addition as

routing packets do disseminate data farther than other varieties of packets, and are looking for- ing a balance between range of distribution and community congestion seems smart. At the same time as in the current implementation, a probe message traverses best a unmarried fringe of the street graph, they may conceivably traverse multiple edges for the reason of retrieving statistics from (and distributing facts to) a greater area. word that messages must go back in a rather quick amount of time to their original sender earlier than that automobile actions too a ways faraway from its authentic function. to make sure this, a distance or time restrict will be imposed on the probe. also within the case of a multi-aspect probe, if a node this is forwarding that probe has no acquaintances inside the spec- ified path (local most) and the probe has already traversed at the least one side, the node ought to truely return the probe rather than losing it, and beneficial data might nonetheless be won from the probe on its go back experience. While vanets are an interesting location of research, they're not yet a practical truth. This presents a glimpse into the potential of reliability-based totally metrics for routing p.c.- ets inside a vanet and demonstrates convincing performance for high throughput within the vanet paradigm.

REFERENCES

1. S. Basagni, I. Chlamtac, V. Syrotiuk, B. Woodward, A Distance Routing Effect Algorithm for Mobility (DREAM), in: Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking, ACM, 1998.
2. J. Bernsen, A Reliability-Based Routing Protocol for Vehicular Ad-Hoc Networks, Master's thesis, University of Kentucky, 2011.
3. J. Bernsen, D." Manivannan, Unicast routing protocols for vehicular ad hoc networks:" a critical comparison and classification, Pervasive and Mobile Computing2009;5(1):118.http://linkinghub.elsevier.com/retrieve/pii/S1574119208000758.
4. S. Das, H. Pucha, Y. Hu, Performance comparison of scalable location services for geographic ad hoc routing, in: INFOCOM 2005. Proceedings of IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 2, IEEE, 2005.
5. E.W. Dijkstra, "A note on two problems in connexion with graphs, NumerischeMathematik"1959;1:269271.doi:http://dx.doi.org/10.1007/BF0138639.
6. Y. Ding, C. Wang, L. Xiao, " A static-node assisted adaptive routing protocol in vehicular networks", in: Proceedings of The Fourth ACM International Workshop on Vehicular Ad Hoc Networks (VANET '07), ACM, New York, NY, USA, 2007.
7. K. Fall, K. Varadhan, The ns Manual (formerly ns Notes and

Documentation), 2010 http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf.

8. B. Karp, H.T. Kung, GPSR: greedy perimeter stateless routing for wireless networks, in: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom '00), ACM, New York, NY, USA, 2000
9. M. Käsemann, H. Füßler, H. Hartenstein, M. Mauve, A Reactive Location Service for Mobile Ad Hoc Networks, Department of Computer Science, University of Mannheim, Tech. Rep. TR-02-014.
10. W. Kieß, H. Füßler, J. Widmer, M. Mauve, Hierarchical location service for mobile ad-hoc networks, ACM SIGMOBILE Mobile Computing and Communications Review 2004; 8(4): 47–58.
11. J. Li, J. Jannotti, D. De Couto, D. Karger, R. Morris, A scalable location service for geographic ad hoc routing, in: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, ACM, 2000.
12. C. Lochert, H. Hartenstein, J. Tian, H. Füßler, D. Hermann, M. Mauve, A routing strategy for vehicular ad hoc networks in city environments, in: Proceedings of the IEEE Intelligent Vehicles Symposium, 2003.
13. M. Mauve, A. Widmer, H. Hartenstein, A survey on position-based routing in mobile ad hoc networks, Network, IEEE 15, 2002; (6)
14. Banupriya.S and Sasirega.D, Mobile Adhoc Networks:Key Management., ISBN 978-93-83459-01-8.,National Conference on Innovative Trends in Information Technology (08/02/2014) .
15. Lavanya.A,"A Review of DDos attacks Protocols in the cloud environment", International journal of basic and applied research,2018; 8(6).