Sub-Sahara African Academic
Research Publications
June, 2021, Editions

JPAS
Journal of Pure and Applied Science
Vol..20, No.9, ISSN 2318-9315

A RELIABLE CONNECTIVITY AWARE ROUTING PROTOCOL WITH VARYING VELOCITY FOR VANET

AHMADU MAIDORAWA1*
1Department of Computer Science, Federal Polytechnic, Bauchi, P. M. B 0231Bauchi, Bauchi State, Nigeria

ABSTRACT
This paper proposed a Reliable Connectivity Aware Routing (RCAR) protocol with varying velocity for Vehicular Ad hoc Network (VANET). The protocol uses a control broadcast to reduce the number of overhead packets needed in a route discovery process. It is also equipped with an alternative backup route that is used whenever a primary path to destination failed, which highly reduces the frequent launching and re-launching of the route discovery process that waste useful bandwidth and unnecessarily prolonging the average packet delay. NS2 simulation results show that the performance of RCAR protocol outperformed the original connectivity aware routing (CAR) protocol by reducing the average packet delay by 24%, control overheads by 13.4% and increased the packet delivery ratio by 10.4%.

Keywords: *alternative path, primary path, protocol, routing, VANET.*

INTRODUCTION
Vehicular Ad Hoc Network (VANET) is a self organized and self maintained communication network that is formed by moving vehicles to provide safety, entertainment and information to its users. To tap these benefits, efficient and reliable routing protocols are needed. Designing routing protocol in VANET is challenging due to high vehicles' mobility that result in frequent network disconnections. In order to tackle this challenge, numerous routing protocols have been proposed in the literature. The most promising among these protocols are real time connectivity aware protocols. These protocols have the

ability to reactively discover connected paths to destination by flooding the networks with a route request message and computing dynamically vehicles' density from an on the fly data collected to enable paths selection. Even though flooding will introduce bandwidth wastage, it is necessary since location service is not assumed. Once a path is selected, it is used for subsequent message transfer between source and destination nodes.

A vehicle equipped with wireless receiver and transceiver can participate freely in the VANET network. Each vehicle is considered as a node and its role can be described as either a host (source or destination) or a router for any given communication in the network. An end to end communication through a large distance is only possible through multi-hop of nodes due to limited wireless communication range of each vehicle. Sometimes permanent nodes are used as roadside units that serve as drop points for messages in sparsely populated roads (Mershad et al, 2012) or as a gateway to the global network, the Internet (Mohandas et al, 2008).

There are numerous VANET applications. The central ideas of these applications are to provide safety, information, and infotainment to its users. Safety applications help prevent accidents by providing accident avoidance warnings (Bernsen, 2009).

The earlier attempts to get the VANET routing protocols was to adopt the routing protocols developed for MANET such as Dynamic Source Routing, DSR (Johnson, 1996), Ad Hoc On-demand Distance Vector, AODV (Perkins, 1999), and Greedy Perimeter Stateless Routing, GPSR (Karp and Kung, 2000). These MANET protocols were found to be quite unsuitable for VANET routing because they are adversely affected by the high nodes mobility and signal obstructions like buildings (Naumov et al 2006).

A seminal VANET routing algorithm, Geographic Source Routing (GSR) was proposed (Lochert et al, 2003). It uses a static street map of urban city to route packet from source to destination by forwarding packets along the street. Sequences of intersections to be traversed in order to reach a destination are stored in the packet header. The approach can help overcome the problem of signal obstructions that are frequent in VANET but it has little chance of delivering a packet to its destination because it has not considered current vehicle density. This means that, if there is sufficient vehicles availability on the selected route path, the packet can be delivered to its destination otherwise not.

Anchor-based Street and Traffic Aware Routing, A-STAR (Seet et al, 2004) utilizes bus route information as a strategy to find routes with a high probability of delivering. Similar to GSR, A-STAR uses a static street map to route packet around signal obstacles. One can easily see that A-STAR is a bit better than GSR as it tries to estimate traffic density of a street based on bus route information, but it is not optimal as using dynamic approach that can explores latest traffic condition information. Furthermore, it is probable that a packet can be received by a node that has no neighbors nearer than itself.

Spatial and Traffic Aware, STAR (Giudici and Pagani, 2005) was among the first routing protocol that considered using dynamic traffic density in selecting a routing path. The STAR tries to overcome the problem of sending packet along the street where vehicles may be currently unavailable by exploiting real topology information gathered by network nodes. Two major drawbacks of STAR algorithm are network overheads and its recovery strategy. The algorithm's reliance on traditional beacons may introduce scalability and wasted bandwidth problems since there no heuristic for adapting the beacon to conditions such as high node density. Secondly, in case of routing failure, the node at which the failure occurs re-computes the path to destination as there is no backup route to destination in the packet header. This will likely increase the delay time.

Besides the VANET protocols discussed in the previous paragraphs, there are many other routing protocols proposed in the literature. In order to explain and highlight some of their weaknesses, consider Figure 1. A vehicle node S wants to send message to a destination vehicle node D. Road intersections, also called Anchor Points (APs for short) are represented with letters such as a, b, c, etc. A connected edge say bf means there are sufficient vehicles' availability to multi hop message between the two anchor point b and f.

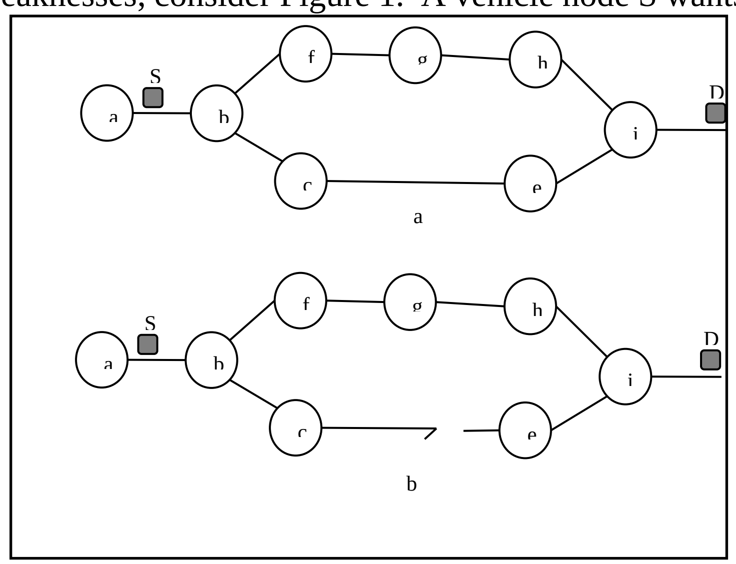


Figure 1 Connected path(s) from source to destination

Connectivity Aware Routing, CAR (Naumov and Gross, 2007) is a routing protocol that finds connected path to destination dynamically. Unlike most of routing protocols that rely on *location service* to find the position of destination, CAR locates the position of destination by the use of control broadcast called PGB mechanism that minimizes network overheads. Suppose that S wants to send a message to D (see Figure 1a). CAR algorithm will initiate a destination location and path discovery by broadcasting a propagating REQ packet. Each node that receives the REQ packet updates connectivity information (number of hops, average number of neighbors, and minimum number of neighbors) on the packet. It also adds an anchor point to path's field in the REQ header if the angle between its velocity vector and that of a sender is above a threshold value. This process is repeated until the REQ reaches the destination D. The node D chooses the path (between [b, c, e, i] and [b, f, g, h, i]) that provides better connectivity and lower delay. Assumed that the path [b, c, e, i] is the better route, D writes this path in reverse order to a response RES packet's header. Eventually RES reply is sent to S using unicast transmission. The node S and D will continue to use this path [b, c, e, i] to exchange messages.

Suppose after some time, the link between c and g is down (see Figure 1b) as a result of non-vehicle availability. The path maintenance of CAR requires the node that detects the problem to take a remedy. The remedy is to keep and carry the packet for a threshold time as disconnections may likely be a temporary one. After the threshold time elapsed, the node that detected the problem initiates a route discovery to D. If it cannot find route to D, it drops the packet and send error ERR message to S. On receiving the ERR message; S starts a new path discovery.

This approach will introduce additional control overhead that causes bandwidth wastage and increase packet delivery delay as there is no useful data transmission during the path discovery process. The problem would have been reduced if S has an alternative backup route to D.

Yang et al (2010) proposed Adaptive Connectivity Aware Routing (ACAR) protocol that is based on statistical and real time density data gathered from an on the fly density collection process. In order to avoid flooding the network to discover a connected path to destination D, statistical density data is used to initially select a route to destination. As packets move from source S to destination D, an on the fly density information is collected. At the destination

D, statistical density is compared with the real density collected. If there is significant discrepancy (say 30% difference or more) between the statistical and real density, then the destination D notifies the source S to select another route else the same route will be used for subsequent data packet transmission.

Now suppose that S wants to send a message to D (see Figure 1a). S uses location service to obtain the location of D. It then obtains route to D based on statistical density from a Geographic Information System (GIS). Suppose also that the statistical density information of the path [b, c, e, i] is better than that of the path [b, f, g, h, i]. ACAR will chooses [b, c, e, i] to send a message to D. At D real traffic density collected will be compared with statistical density. If there is no significant discrepancy between the two traffic densities, ACAR will continue to use the path [b, c, e, i] for subsequent transmission.

Now let us look at what will happen after a situation in Figure 1b happened. This situation can occur as a result of change in vehicle availability. To remedy the situation, ACAR suggests that the vehicle that detects this problem to keep and carry the packets until there exists available next hop. This remedy will work fine for short time disconnection. If the disconnection is of a long or permanent time, it will result in a long packet delivery delay or even packet lost. Since there is no feedback mechanism to S when a packet is enqueued or dropped, S will continue to send packets through the broken route path which will result in losing all the packets or prolonging the packet delivery time.

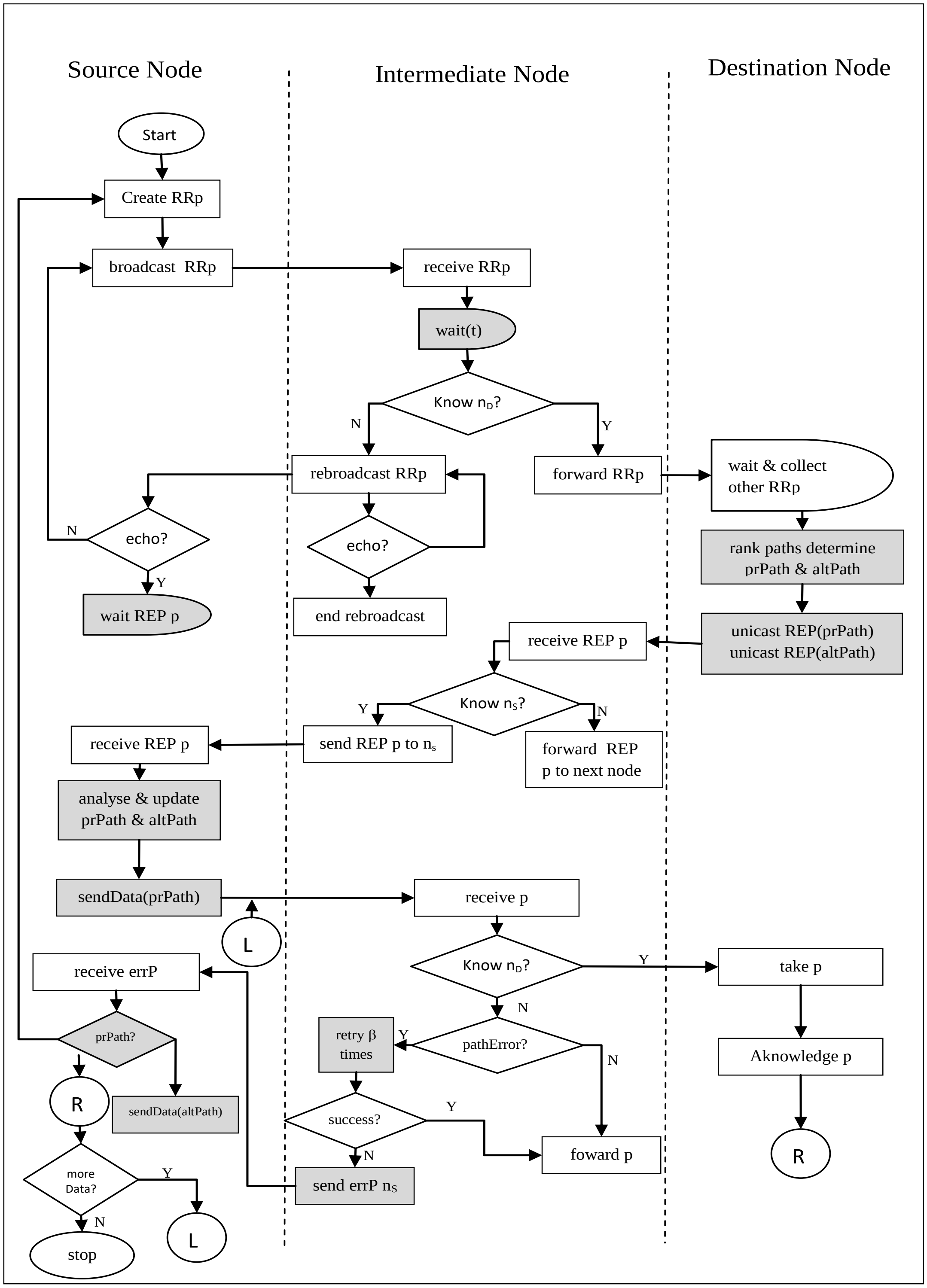
Anchor Geographical based routing Protocol, AGP was proposed by Yan et al (2011). Similar to the CAR (Naumov and Gross, 2007) protocol, S gets connected paths to destination D by reactive broadcast. Unlike CAR that identifies anchor points (APs) by the magnitude of angle between two velocity vectors, AGP extract APs from the digital map layout. At the destination D, the best path (between [b, c, e, i] and [b, f, g, h, i]) that provides better connectivity and lower delay is chosen. Assumed that the path [b, c, e, i] is the better route, D writes this path in the reversed order to a response, RES, packet's header. Eventually RES reply is unicast to S. The node S and D will continue to use this path [b, c, e, i] to exchange messages by unicast transmission. Suppose after some time, situation in Figure 1b occurs. The remedy assumed by AGP is just the traditional carry and forward strategy. This will prolong the delivery time or most likely cause the packet to be lost. As there is no feedback mechanism to S when disconnection is detected in AGP, S will continue sending packets through the path [b, c, e, i] which all will be lost. This will highly decrease the packet delivery ratio and increases the packet lost ratio.

Road Based using Vehicular Traffic – Reactive, RBVT-R (Nzounta et al, 2009) protocol is very similar to CAR (Naumov and Gross, 2007) in terms of how it discovers routes from S to D and how messages are sent between S and D. When a route error occurs (as in Figure 1b), the node that detects the problem unicast a route error packet to S. Upon receiving the route error packet, S puts packets for the respective failed route on hold for a threshold timeout. After a timeout, it tries to send the packets via the broken route hoping that the disconnection was temporary. S launch new route discovery after few consecutive route errors. The consequence of this remedial measure is that re-attempting to use the broken link will likely increase delivery delay and launching new discovery will cause additional control overheads that cause bandwidth wastage and increase the packet delivery delay. The problem will have been reduced if S has a backup route to D.

Therefore, this article enhances a Connectivity Aware Routing, CAR (Naumov and Gross, 2007) protocol to provide better performance. The resulting new protocol called *an Reliable Connectivity Aware (RCAR)* protocol, reduces the network control overhead by using a *control broadcast* and also reduces the need for re-launching of a new path discovery process whenever a connected path breaks by utilizing an *alternative backup route*.

MATERIALS AND METHODS

The proposed RCAR protocol consists of two major changes to the CAR protocol. The first change was in the route discovery process while the second was in the route maintenance handling. The route discovery was modified to use *control broadcast* during the route request and the route reply, to include an *alternative backup path* between the source and the destination nodes in addition to the primary path. The route maintenance mechanism was modified so that whenever a route failure occurred, the source node uses the alternative backup route for transmission of data and when the alternative route itself failed then the new path discovery is launched. Figure 2 depicts a diagrammatic representation of RCAR protocol. The colored shaped denote the areas of research contribution while the un-colored shapes are from the original CAR protocol.



Legend: Research contribution From original CAR protocol

Abbreviations:
nS and nD: Identification address of source and destination respectively.
prPath, altPath : primary and alternative paths respectively.
p, RRp, REP: packet, Route request packet and Route reply packet respectively.
ni, nj: current node and previous node respectively

Control Broadcast Waiting Time

For brevity brevity of explanation , consider Figure 3 where a vehicle node C at the center broadcast a route request packet. All the neighboring vehicles within its radio communication range will receive the broadcast packet. Without a *control broadcast* all vehicles that received the packets rebroadcast them further. With a control broadcast, only vehicles A and E rebroadcast the request packet further.

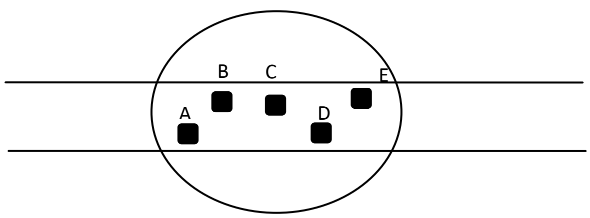


Figure 3: Illustration of Control Broadcast

The waiting time wait(t) of a neighboring node that receive route request broadcast is derived from the inverse proportion as:

$$wait(t) \propto \frac{1}{d} \quad (1)$$

where *d* is the distance between the sender and receiver. Thus:

$$wait(t) = \beta\frac{1}{d} \quad (2)$$

where β is the constant of proportionality. Now the concern is to determine the value of β. To get β, it is sufficient to do some analysis on radio waves. 1megahertz of radio signal has a wavelength of 299.8 meters (Wikipedia, 2012). Hence 1 GHz will have a wavelength of 0.2998 meters. In Naumov et al (2007) radio frequency was modeled at 2.4GHz. Using this frequency of 2.4GHz, the corresponding wavelength is 0.124917 meters, which equivalent to say 2.4MHz has a wavelength of 124.917 meters. This means that every $\frac{2.4}{10^6}$ second, the signal covers 124.917 meters. Using *R* meters radio communication range, β is set to the time a signal will take to cover distance *R* and it is calculated as follows:

$$\beta = \frac{2.4}{10^6} \times \frac{R}{124.917} = \frac{1.9213R}{10^8} \quad (3)$$

Substituting Equation 3 in Equation 2, the waiting time becomes

$$wait(t) = \frac{1.9213R}{d \times 10^8} \quad (4)$$

Thus, Equation 4 provides the waiting time used in the *control broadcast*. As mention before *d* is the distance between the broadcast sender and receiver and *R* is the signal radio communication range.

Route Discovery Process

To find a route to a destination node, RCAR uses *control broadcast* in addition to Preferred Group Broadcast (PGB). The PGB eliminates redundant retransmission of route request packet so that whenever a particular node broadcasts route request packet, it starts listening, if it is echoed further then it stops the rebroadcast otherwise it repeats the packet broadcast after waiting for a set time interval (Naumov, 2006).

In the other hand, the *control broadcast* helps to reduce the bandwidth wastage incurred as a result of route discovery. Any intermediate node that is not the destination that receives a route request packet, it is forced to wait for a time that inversely proportional to the distance between it and the immediate node that forwarded the route request packet. This will allow the farthest node that receives the route request packet to broadcast it further before the closer nodes. This is because the farthest node will have a shorter waiting time due to the inverse proportionality than the closer ones. Any node that receives an echo of the route request packet it hold within its waiting time, it stops the re-broadcasting since a further node has already rebroadcasted the packet.

Any intermediate node that receives a route request packet, it adds its *id* into the receive path discovery table to avoid routing loop. It also updates the routing information fields in the packet header. The number of hops is incremented by one. The average number of neighbors is updated as follows.

$$avrNeighbor = \frac{avrNeighbor \times (hopCount - 1)}{hopCount} \quad (5)$$

If the number of neighbors of the current nodes is less than the minimum number of neighbors, the *minimum neighbor* field in the packet header is set to the number of neighbors of the current node.

An anchor point (intersection) is identified when the angle between two velocity vectors is greater than 18o as in Naumov et al (2007). A new anchor point is added to a broadcast packet anytime the request packet passes a new intersection.

When the request packet finally reaches the destination, the destination node will have a complete sequence of anchor points that need to be traversed to reach back the source. After receiving the first route request, the destination waits a while to collect other routes that follow different paths. These paths are sorted or ranked in decreasing *average neighbor* field and then by decreasing *minimum neighbor* field. The top most path in the rank is chosen as the primary path while the second top most path in the rank is made the alternative backup path. The destination prepares two reply packets (one for the primary path and one for the alternative path) and send them to the source using unicast transmission.

When the source node receives a route reply from a destination node for the first time, it stores it as primary path and begin sending data packets using the path. When it receives the second route reply, it compares it to the primary path. If it is better than the primary path, the new path is made the primary path and the old primary path becomes the alternative path otherwise the new path is made to be the alternative backup path.

Data Packet Transmission

Once a source has connected path or paths to a destination, it starts sending data packets to the destination using unicast transmission through the primary path as default. A path consists of a sequence of intersections called anchor points that need to be traversed to reach the destination. Packets are forwarded greedily to the furthest node towards to the next intersection (anchor point) instead to the furthest node towards the destination. Each intermediate forwarding node checks if the distance between it and the next anchor point is less than half the node’s coverage range as in Naumov et al (2007). If so, then this anchor point is marked and the next one is set as the target. This process continues until the destination is reached.

Route Error Recovery Process

Route path disconnection is frequent in VANET and this is usually called route error. It is normally occurs when an intermediate node receives a packet but no any next node closer to the target intersection in which to forward the packet to. RCAR protocol is equipped with a route error recovery process to help overcome route error.

When a route error occurs, RCAR protocol enters route error recovery process. In this process, the node that detects the error starts buffering data packets and probes for the next available node. The probe is performed for a specific number of times. If the probe is successful, the buffered packets are then forwarded to the next available node in the direction of the target intersection. If the probe for the next available node was unsuccessful, the buffered packets are dropped and a route error message is sent to the source. When a source node receives an error packet, it checks whether the broken route path is the primary path. If so, it uses the alternative backup path to send the data packets to the destination. New path discovery will only be launched when both the primary path as well as the alternative backup path failed.

Probing broken route for a specific number of times helps to overcome intermittent disconnections which last only for a very short while. Sending error message to the source notified the source that there was a link failure in the route path. The error message serves two purposes. One, the source stops sending packets through the broken route path and secondly, it makes the source to utilize the alternative backup path for subsequent packets transfer.

Neighborhood Table Management

In this article, a neighborhood table management process of CAR protocol (Naumov, 2007) was used in the RCAR protocol.

Simulation Setup

In this research experiment, the NS-2.33 simulation tool was used as the simulation platform (Issariyakul and Hossain, 2009). The area of simulation was 1500m x 1500m. The radio transmission range and MAC type were 250m and 802.11 respectively. Scaling models used in this experiment were varying vehicle densities. The detail of this setup is displayed in Table 1.

Table 1: Simulation Setup

Parameter	Value
Simulator	NS-2.33
Simulation area	1500m x 1500m
Simulation time	300s
Transmission range	250m
MAC type	802.11
Radio propagation model	Two ray ground
Routing protocols	RCAR, CAR
Queue size	50 packets
Packet generation interval	0.25s
Packet size	512B
Varying vehicle velocity:	
No. of vehicles	100
No. of TCP connections	10
Vehicle velocity	20 – 60 km/h

The simulation parameters and their corresponding values are chosen based on previous researches to provide a good basis for the evaluation of RCAR protocol. NS2 was selected as in Naumov et al (2007), Yan et al (2011), Yang et al (2010) etc. Varying vehicle density as in Naumov et al (2007) , Yan et al (2011). Simulation area as in Nzounta et al (2009) and Yang et al (2010). MAC type, radio propagation and signal range as in Yang et al (2010) and Yan et al (2011). Simulation duration time as in Naumov et al (2007) and Yang et al (2010).

Evaluation Metrics

The metrics for performance comparisons are as follows:

i. Packet Delivery Ratio: The fraction of successfully delivered to the destination relative to the total packets sent.
ii. Packet Lost Ratio: The fraction of lost packets during transmission relative to the total packets sent.
iii. Average Delay: The average duration that takes a packet to travel from source to destination.

iv. Control Overhead: The absolute number of routing packets necessary to find connected paths and manage neighborhood tables.

These metrics were chosen for performance evaluation of RCAR because they are indicators used by previous researchers in evaluating the performance of routing protocols as can be found in Naumov et al (2007), Yan et al (2011), Yang et al (2010).

RESULTS AND DISCUSSION

This evaluation was done by varying vehicle velocity from 20km/h to 60km/h using interval 10km/h. The vehicle density was kept constant at 100 vehicles throughout the simulation. The metrics used are packet delivery delay ratio, packet lost ratio, average packet delay and routing overhead.

Packet Delivery Ratio

As the vehicle velocity increases, the packet delivery ratio slightly decreases for both RCAR and CAR protocol as shown in Figure 4. RCAR protocol outperformed the CAR protocol by increasing the packet delivery by 10.4%. This improvement was possibly as a result of RCAR protocol being equipped with an alternative route to use whenever primary path failed. In the contrary, CAR suffered from waiting for path discovery whenever a route error occurred which resulted in reducing the packet delivery.

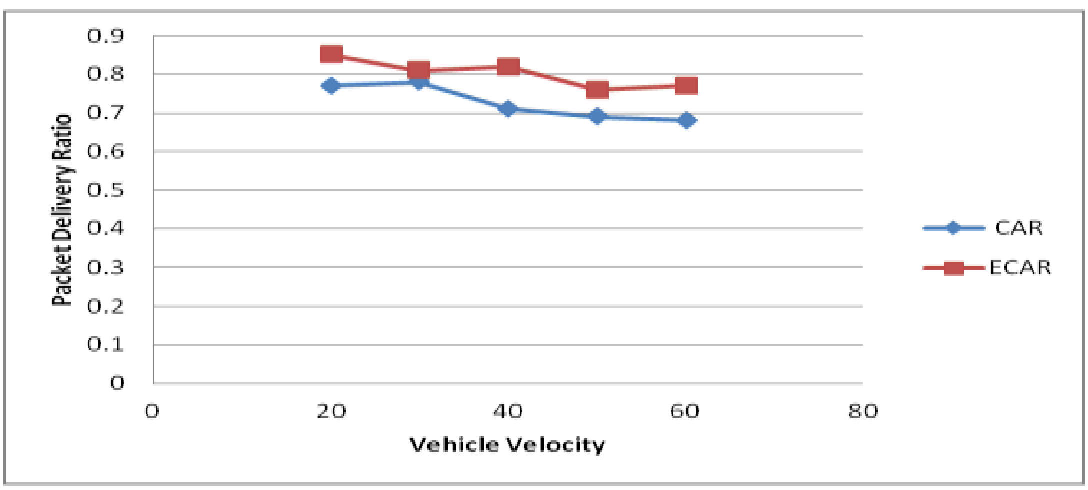


Figure 4: Packet Delivery Ratio - Varying Velocity

Packet Lost Ratio

RCAR protocol successfully reduced the packet lost ratio as shown in Figure 5. RCAR protocol brings the packet lost ratio down compared to CAR protocol. This advantage is achieved by utilizing the alternative path by the RCAR whenever the primary route path fails. The CAR reliance on re-launching new path discovery by either the intermediate nodes or the source node increases the packet lost ratio.

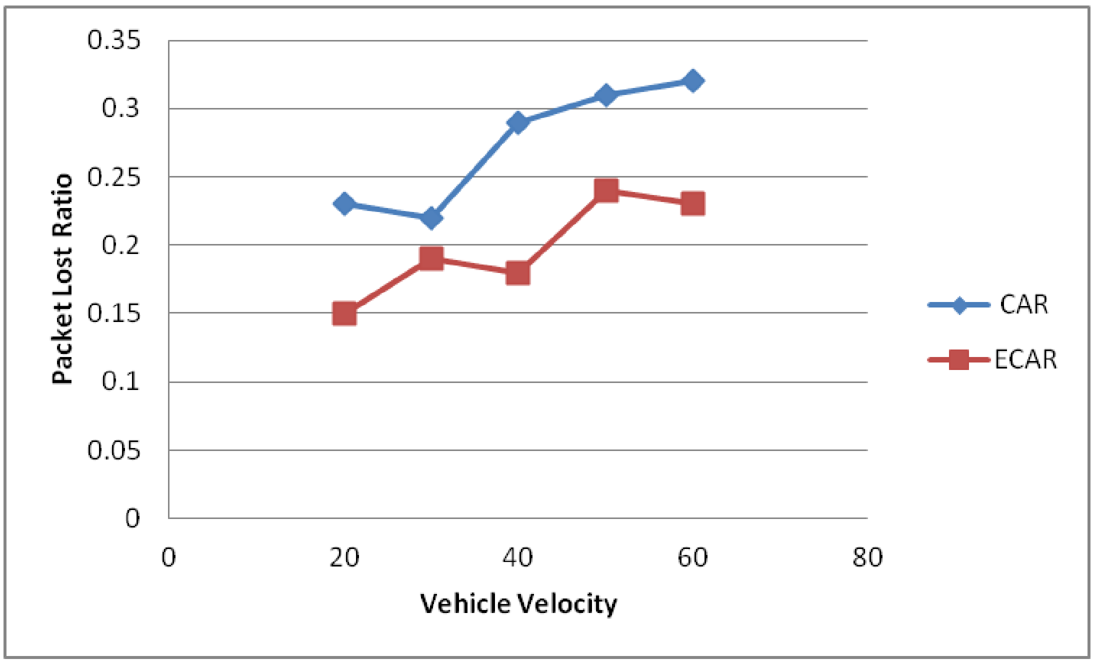


Figure 5: Packet Lost Ratio – Varying Velocity

Average Packet Delay

The RCAR protocol has reduced the average packet delay of CAR protocol by 24% as shown in Figure 6. The main reason for the higher average packet delay of CAR protocol is because of its usual launching and re-launching of route request by intermediate nodes or by source nodes whenever a route failure occurs. Every re-launching of route request will consume a substantial amount of time as all packets are forced to wait in queue which eventually increases the delivery delay. On the other hand, RCAR protocol was able to keep the average packet delay lower as it uses an alternative path to to recover, in most cases, from route failure.

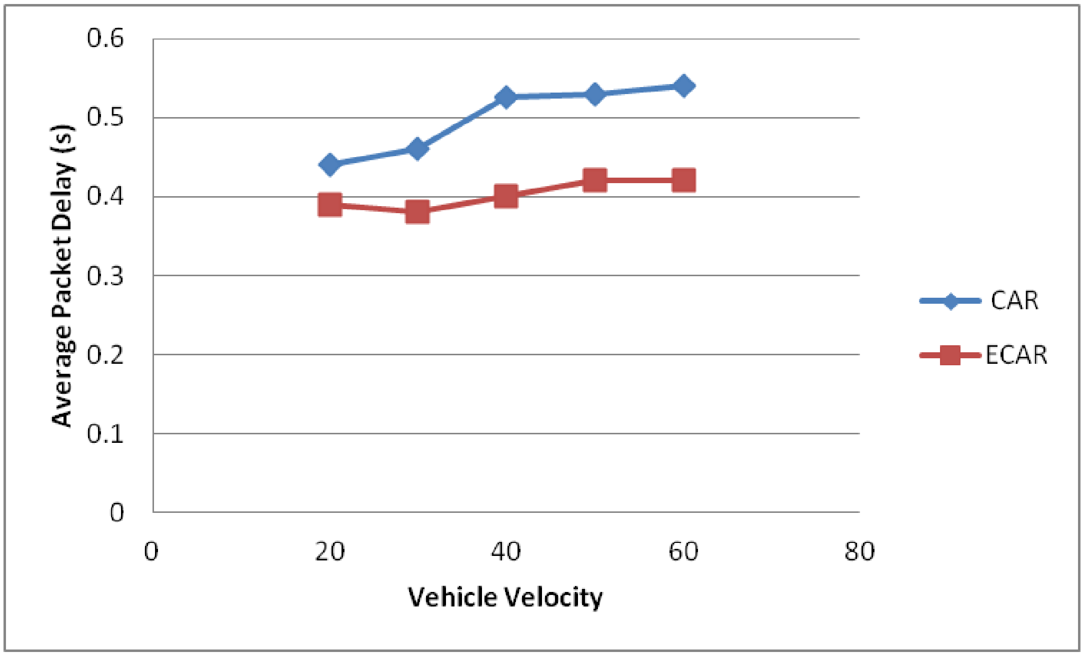


Figure 6: Packet Delivery Delay – Varying Velocity

Routing Overheads

Figure 7 displayed the results of routing overhead. It can be seen from the Figure that both the two protocols shows increase routing overhead with the increase of vehicle velocity. The routing overhead of RCAR protocol was 13.4% less than that of CAR protocol. The reduction of control routing overhead packets was achieved in new proposed RCAR protocol due the *control broadcast* embedded in the route path discovery process. The new protocol requires fewer control overhead since not all the nodes that receives a broadcast packets forward it further, it is only the farthest nodes that do the rebroadcast. Secondly, the first route path discovery is necessary to get connected paths to the destination and subsequent re-launching of a route discovery by the intermediate node or by the source depends on route failure. Since RCAR is equipped with an alternative route to destination to utilize whenever the primary route fails, the number of required route discovery process is highly reduced which automatically yields reduction in the routing control overhead. In the other hand, the original CAR protocol only uses PGB in its broadcast, only the sender of the broadcast packet is restricted but all other receivers are allowed to re-broadcast the route request further creating more control overhead. Since this will be needed in every re-launching of path request during long route failure, more and more control overheads are introduced into the network.

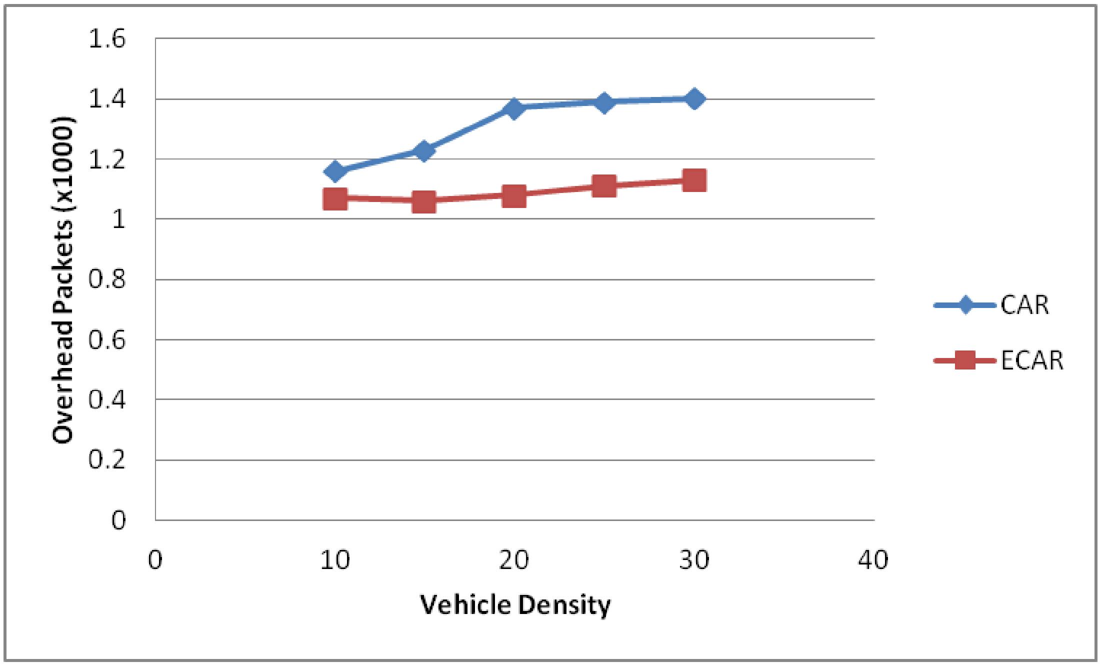


Figure 7 : Routing Overhead – Varying Velocity

RECOMMENDATION

In general, feature researches should target security and privacy issues in order to make the VANET technology deployment acceptable to users. This is because most of current researches on VANET routing protocols assume that vehicles freely communicate with each other without giving due consideration for security and privacy issues. If security issues are not addressed squarely, some people can exploit this vulnerability and cause harm to its users.

In particular, future research should consider extending the use of the alternative backup route to anchor points instead of restricting to only endpoint nodes (source and destination). Beaconing can further be enhanced by using clustering techniques.

CONCLUSION

There are two major contributions of this research that greatly help to enhance the performance Connectivity Aware Routing protocol. First, in the path discovery process, a *control broadcast* was introduced. The second contribution was introduction of an alternative backup path to be used in addition to the primary path. The alternative backup path is used by the endpoint nodes (source or destination) whenever the primary path fails. In particular, future research should consider extending the use of the alternative backup route to anchor points instead of restricting to only endpoint. In general, feature researches should target security and privacy issues in order to make the VANET technology deployment acceptable to users. This is because most of current researches on VANET routing protocols assume that vehicles freely communicate with each other without giving due consideration for security and privacy issues. If security issues are not addressed squarely, some people can exploit this vulnerability and cause harm to its users.

REFERENCES

Bernsen, J. and D. Manivannan, 2009. Unicast routing protocols for vehicular ad hoc networks: a critical comparison and classification, Pervasive and Mobile Computing 5(1): 1–18.

Giudici, F. and E. Pagani, 2005. Spatial and traffic-aware routing (STAR) for vehicular systems, in: Proceedings of High Performance Computing and Communications, Springer, Berlin, Heidelberg, vol 726, pp 77-86.

Issariyaul, T. and E. Hossain, 2009. Introduction to network simulator NS2, Sprnger science + business media, pp 1 – 400.

Johnson, D. B., Maltz D. A. and J. Broch, 1996. Dynamic source routing in ad hoc wireless networks. In: Imielinski T, Korth H (eds) Mobile computing, vol 353. pp 1-25.

Karp, B. and H.T. Kung, 2000. GPSR: Greedy perimeter stateless routing for wireless networks, in: proceedings of the 6th Annual International Conference on Mobile Computing and Networking, MobiCom 2000, ACM, New York, USA, pp 243-254.

Lochert, C., H. Hartenstein, J. Tian, H. Füÿler, D. Hermann and M. Mauve, 2003. A routing strategy for vehicular ad hoc networks in city environments, *in: Proceedings of the IEEE Intelligent Vehicles Symposium*

Mershad, K., H. Artail and M. Gerla, 2012. ROAMER: Roadside Units as message routers in VANETs, Ad Hoc Network Elsevier Vol 10 pp 479 – 496.

Mohandas, B.K., A. Nayak, K. Naik and N. Goe, 2008. ABSRP – A Service Discovery Approach for Vehicular Ad-Hoc Networks, in: Proceedings of the 3rd IEEE Asia-Pacific Services Computing Conference Pisa, Italy.

Naumov, V., R. Baumann and T. Gross, 2006. An Evaluation of Inter-Vehicle Ad Hoc Networks Based on Realistic Vehicular Traces, MobiHoc, pp 108 -119.

Naumov, V. and T. Gross, 2007. Connectivity-aware routing (CAR) in vehicular ad-hoc networks, in: Proceedings of the 26th IEEE International Conference on Computer Communications, INFOCOM 2007, IEEE, Piscataway, pp 1919–1927.

Nzouonta, J., N. Rajgure and G. Wang, 2009. Vanet routing on city roads using real-time vehicular traffic information, IEEE transaction on vehicular technology, 58(7):3609-3625.

Perkins, C and E. Royer, 1999. Ad-hoc on-demand distance vector routing. In: Proceedings, WMCSA 1999, 2nd IEEE workshop on mobile computing systems and applications, pp 90–100.

Seet, B. C., G. Liu, B. S. Lee, C. H. Foh, K. J. Wong, and K. K. Lee, 2004. A-STAR: A mobile ad hoc routing strategy for metropolis vehicular communications, in: Proceedings of the Third International IFIP-TC6 Networking Conference, Networking Technologies, Services, and Protocols, vol. 3042, pp. 989-999.

Wikipedia, 2013. Radio waves- speed, wavelength and frequency. Retrievable at https://en.wikipedia.org/wiki/Radio_waves.

Yan, S., J. Xiao-ye and C. Shan-zhi, 2011. AGP: an anchor-geography based routing protocol with mobility prediction for VANET in city scenarios, the Journal of China Universities of Posts and Telecommunications, pp 112-117

Yang, Q., A. Lim, S. Li, J. Fang and P. Agrawal, 2010. ACAR: Adaptive connectivity aware routing for vehicular ad hoc networks in city scenarios, Mobile Netw Application, pp 36-59.