RS-CBAODV: An Enhanced Reactive Routing Algorithm for VANET to Reduce Connection Breakage using Remote Storage Concepts

N. Arul Kumar
Asst. Prof., Department of Computer Science, Bishop Heber College, Trichy - 620 017.
E-mail: arulkumar@live.com

Dr. E. George Dharma Prakash Raj
Asst. Prof., Department of Computer Science and Engineering, Bharathidasan University, Trichy - 620 023.
E-mail: georgeprakashraj@yahoo.com

Abstract—Vehicular Ad hoc Network (VANET) is formed to share information on a temporary basis between the vehicular nodes in a wireless medium. The routing information is used to discover the vehicles and the route has to be maintained to transfer the data. It may lead to link failure or breakage in the communication, if there is high network mobility and constrained topology arrangement. In case of failure, it may cause loss of data and delay in the network. So, to avoid breakage, the routing information is stored in traditional protocols like CBAODV and CS-CBAODV. Instead of storing routing information in client side, the idea of storing data in the remote side is taken into consideration to backup permanently. This remote server manages the data in client server which is to be delivered to vehicular node. After analyzing related protocols and simulators, a new reactive based routing protocol is designed in this research work and it is named as RS-CBAODV is used to handle routing information between vehicular node, Client side node (RST i.e. Road Side Terminals) and also in Remote side storage node. To analysis the performance of the proposed protocol, MOVE and NS2 simulator is used to compare both CS-CBAODV and RS-CBAODV protocols.

Index Terms—CBAODV, CS-CBAODV, Reactive Routing, Remote Route Backup, VANET and Connection Breakage.

I. INTRODUCTION

For this research work in Vehicular Ad hoc Networks (VANETs), the various researchers suggests that the link breakage can be solved by using both client and server side backup. To use this idea, the Ad hoc On demand Distance Vector (AODV) protocol is taken into consideration because it minimizes the number of broadcasts by generating the routes on-demand that uses the list of all the routes. Here, the Route Request (RREQ) is used by source node to broadcast the data to the nearby vehicular nodes [1]. This RREQ is also used when a vehicular node does not have a proper route to reach destination during travel. At this time, a RREQ is broadcasted throughout the network. Moreover, RREQ uses the sequence numbers to avoid congestion in the network. To test the link is active, periodic HELLO messages are broadcasted to the vehicular nodes. These HELLO messages will not be forwarded to other nodes. The Time To Live (TTL) value is sent with TTL = 1. The vehicular node uses this HELLO message to refresh its corresponding lifetime value of the neighboring information in the routing information. A vehicular node can send Route Error (RERR) message while there is a link breakage.

In literature, the Connection Breakage avoidance using Ad hoc on demand Distance Vector (CBAODV) protocol is designed with the help of AODV reactive protocol which delivers better routing information by reducing Connection Breakage (CB) i.e. Link Breakage in VANET [2]. This CBAODV acts as a hotspot between the vehicular nodes to bridge the connection for maintaining the routing information. Additionally, a Client Storage based mechanism to reduce Link Breakage or Connection Breakage Ad hoc on Demand Vector (CS-CBAODV) protocol is developed to enhance reactive routing mechanism for VANET to reduce connection breakage. This CS-CBAODV uses the backup option to store routing information in the Road Side Terminals (RSTs) as a client side.

RS-CBAODV (Remote Storage - Connection Breakage Ad hoc on Demand Distance Vector) is developed to enhance reactive routing algorithm for VANET to reduce connection breakage using remote storage concepts to have multiple route backup on client side and also to monitor the entire network.

II. RELATED WORK

RST (as a static node in highways) is designed to avoid link breakage in the network by using internet connectivity with storage option. Utilizing the routing information in RST is easier than using route discovery

process periodically. This research work is done by using the concepts like link breakage, link quality predictability, infrastructure based VANETs, reactive routing mechanism and fast path discovery.

Gabteni et. al. presented a PHY layer based link state indicator for Ad hoc networks to predict the link failure [3]. This approach is related to OFDM based IEEE 802.11 standard. The strength of the proposed indicator is analyzed with reference to PHY and NET packet reception ratio of the link which is monitored during data transmission. Moreover, the quality of the proposal is compared with signal to noise ratio based predictive link state estimator by using NS-3 simulator.

Iqbal et. al. designed an infrastructure based Ad hoc on-demand distance vector routing protocol for vehicular Ad hoc networks [4]. Here, the fixed infrastructure terminals are used to find the paths available in the networks for reducing link breakage. To support for this work, directional antennas are used to achieve line of sight connection.

GS Raj et. al. proposed an efficient reactive location based ad hoc routing protocol for vehicular networks [5]. This protocol moves to reactive routing as location information degrades. It increases the network scalability and also it reduces routing overhead in the network. The link failure is reduced in presence of high location errors.

MU Kim et. al. created a new fast path recovery scheme for V2V communications using AODV protocol [6]. Metrics are added to the existing AODV hello messages to maintain the links in the network. Whenever the link failure happens in the network, this modified protocol determines the direction of destination node and broadcasts the message to its nearby node with Time To Value = 1. By using the direction flag in the received broadcast message, the data packets are forwarded to its destination vehicular node.

III. PROBLEM DEFINITION

In VANET, each vehicle can move from one location to another location by random in nature. The sender may not be aware of link quality from the beginning to the end of data transmission. So, the link failure is the unavoidable and unpredictable.

In Fig.1, S is the Source Node which tries to communicate with Destination Node D. Node D moves out of the network and it creates connection breakage in the network. This creates a connection breakage between the node E, F and D in the network and it generates the RERR messages to the source node via the RST in the network. In this situation, Source Node S takes more time and resources to retransmit the route request RREQ. Requests generated by Source node is not maintained in any node in the network. The concept of Request Manager is not available to receive the data. While receiving the data, the routing information is not delivered to appropriate users. Data could be merged with the Data Manager to avoid the data mismatch.

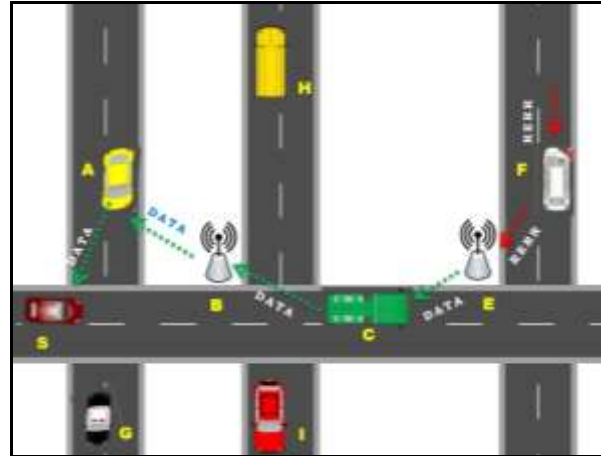The loss of connection breakage between the vehicular node in the network leads to:

Fig.1. Connection Breakage in VANET

- Increases Routing Overhead
- Increases End to End delay
- Packet loss
- Reduces Packet Delivery Ratio
- Throughput value decreases

Connection breakage issue can't be solved by using RST alone. Because, RST may fail to support for the routing information and it may lose the data during disaster or in failure situation happens [7]. Client storage is not able to maintain the routing information even if no longer valid and it can't be utilized as a backup option to path discovery. Client storage may also face the early loss of routing information. The link failure in client side may be happen in client side due hello messages, MAC layer feedback and also for passive acknowledgements.

Client storage routing information in the roadways could be damaged due to network failure [8]. It is not be able to make off-site backups for updating during routing information changes. This RST with client storage option does not much support for on-the-go data access. To overcome these issues, there must be a remote storage to monitor and to control the VANET. The maintenance of routing information between the vehicular node and the RST is a big issue in huge networks like VANETs.

IV. PROPOSED MODEL

The proposed remote storage concept i.e. RS-CBAODV is used to avoid and to solve connection breakage issues in VANET. Further, it enhances the performance of the routing by storing the route requests and responses for vehicular nodes. The main objective of this proposed protocol is to,

- Propose a new reactive routing mechanisms for reducing connection breakage
- Implement Request Manager and Data Manger for both client and server side
- Simulate and to analyze the performance of the proposed algorithm using MOVE and NS-2 simulator

- To incorporate storage related aspects on proposed protocols

The RS-CBAODV contribution is used to store routing information in remote storage concept with the help of client side support by using CS-CBAODV. The Remote storage is also used to backup the routing information whenever there is chance for replaying the client nodes like RST in the network [9]. This RST communicates with both client side backup and also with the server side backup center. The routing information is stored in multiple client storage places. The routing information in the form of new updates will be taken to the remote server after storing the routing information in the RST as a client storage. So, RST interconnects with Remote Storage to get updated routing information in VANET which is given in Fig. 2.
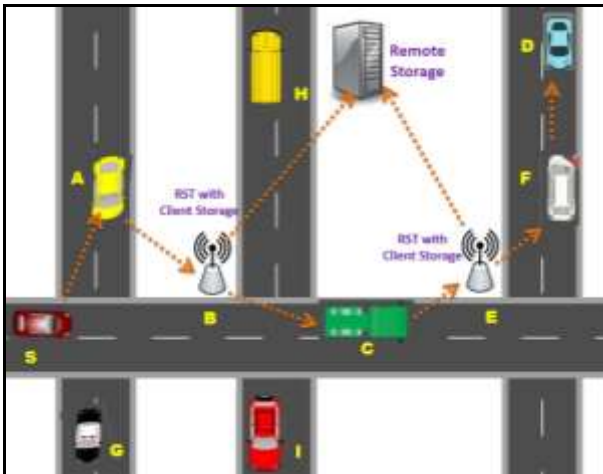

Fig.2. RST communicates with Remote Storage

The Client Storage maintains the query cache in terms of delivering the data locally to the concerned vehicular node. It updates the queries to the neighboring vehicular nodes. The client storage does the cache replacement technique to replace the old queries. Finally, the old routing information is deleted based on the TTL values.

The Remote Storage uses the method of storing routing information as a query cache directory in the remote place in the vehicular network. The RST in the client side receives the acknowledgement from the source node and updates the routing information to the node in the remote place. The remote storage server also does the cache replacement and deletion. The remote storage node is responsible for delivering the routing information to the client storage based on the requests generated [10].

In the proposed mechanisms, the breakage in the network is reduced by making use of RST to store and retrieve frequently accessed routes in the static nodes like RST. Further, the routes are stored in both client and server storage by including vehicular cloud concepts. It makes smooth connection between the vehicular nodes by utilizing the routes which is stored in more than one places [11].

A. Vehicular Client Storage

Vehicular routing information is stored in the client side by using the RST. The Client Storage gets the updated routing information from the remote server storage and then it delivers to Source Node. The following is the algorithm used in RS-CBAODV to store routing information of a vehicle in the client storage

> *Step 1: Maintains Query Data in Client Storage*
>   *Storage Server VDATi*
>   *if Vreq = yes then*
>   *Generate List VDLi*
> *Step 2: Updates the queries to the Client Storage*
>   *if LS = Vreq then*
>   *VDir = Vdir = index I*
> *Step 3: Client Server storage data replacement*
>   *// receives VCSdata*
>   *if VCS = old route data then*
>   *VDATi = New VDAT*
> *Step 4: Client storage server data is deleted*
>   *if VDATdel = yes then*
>   *VDATi = VDAT – 1*
>   *// update VQdir*

B. Vehicular Remote Storage

The Vehicular Remote Storage node is responsible to deliver real time data to all the vehicular users. The following is the algorithm used in RS-CBAODV to store routing information of a vehicle in the remote storage.

> *Step 1: Query Request Directory in Remote Storage Server*
>   *if VDreq = yes then*
>   *sends VDAsg to QM*
>   *Sends VDAT*
>   *VDATdir = VDATdir + index i*
>   *waits for Ack I*
>   *Ack i = Ack i + index i*
> *Step 2: Updates the request to the Remote Storage server*
>   *if VDATres = yes then*
>   *VDATdir = VDATdir + 1*
>   *send Ack*
> *Step 3: Remote storage server data replacement*
>   *if new VDAT = yes then*
>   *VDAT = new VDAT*
>   *// sends VsourceAdd*
> *Step 4: Remote Storage Server data is deleted*
>   *// sends VDATdel*
>   *VDAT = VDAT – 1*

The schematic diagram of RS-CBAODV protocol is given in Fig. 3. RS-CBAODV makes vehicular node to generate route requests to RST with Client Side Storage. Client Side Storage acts as a Request Manager to deliver the response with the help of Data Manager. Data Manger gets ready to deliver the routing information if it is available in its storage. To receive the updated result, RST forwards the requests to node in the remote place. This remote node acts as a remote server to control and to monitor the entire network. The following is the complete

information about the proposed RS-CBAODV algorithm uses both vehicular client storage and vehicular remote storage to store routing information in the RS-CBAODV routing protocol. The proposed RS-CBAODV routing protocol uses RST to store routing information of the network and gets routing information updates from the remote server.
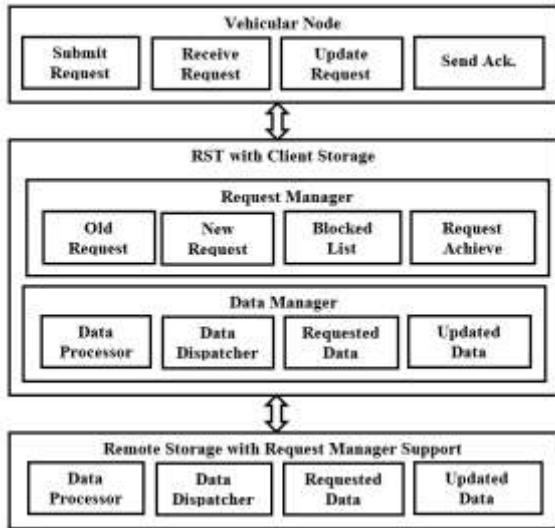

Fig.3. The schematic diagram of RS-CBAODV protocol

The flow diagram of the proposed RS-CBAODV protocol is given in Fig. 4. Here, Source Node generates the route requests to the RST. RST check whether the Source node is listed in the Blocked List. If Source Node is in Blocked List, RST confirms and blocks the Source Node. If Source Node is not in the Blocked List, RST checks whether the request is old or the new request. For Old request, RST confirms and delivers the data using Data Manager. Then RST receives the Acknowledgement (Ack). To deliver the new update, RST gets update from the Remote Server Storage and Stores the routing information in its routing table before it is delivered to the vehicular user.
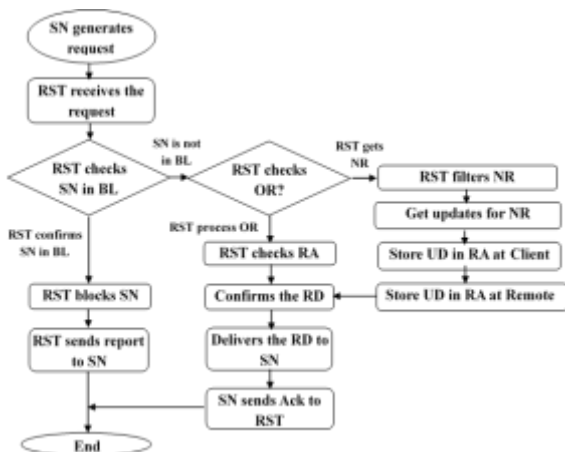

Fig.4. Flow diagram of proposed RS-CBAODV protocol

The Complete Algorithm of proposed RS-CBAODV model

Step 1: Vehicular Node generates VRReq to RST
Step 2: RST receives the Vehicular Route Request VRReq
Step 3: RST sends VRReq to Request Manager (RM)
Step 4: RM classifies request as Old Request (OR) and New Request (NR)
Step 5: RM maintains the frequent requests in Request Archive (RA)
Step 6: RM filters the requests and maintains NR

```
        RST receives the request {
        if SN is in blocked list (BL) {
        Stage 1: RST blocks the SN
        Stage 2: RST sends report to SN
        Stage 3: Vehicles receives denied message
} else if SN is not in BL {
RST check for Old Request (OR) {
        Stage 1: RST check the RA
        Stage 2: RST confirms the Requested Data (RD)
        Stage 3: RST delivers the RD to SN
        Stage 4: SN sends Acknowledge (ack) to RST
} else if request is New Request (NR){
        Stage 1: RST filters the New Request (NR)
        Stage 2: RST gets update for the NR
        Stage 3: RST stores Updated Data (UD) in RA
at client
        Stage 4: RST stores UD in RA at remote side
storage
        Stage 5: RST stores and delivers Requested Data
(RD);
        Stage 6: Vehicle sends ACK to RST;
} else {
        continue searching RST for data;
}       }}
```

Step 7: RM sends the query to Data Manager (DM)
Step 8: DM send the requested data to RM and then to RST
Step 9: RST confirms the data delivery by receiving Acknowledgement (Ack) from vehicle.

The primary purpose of using remote storage is to keep information readily available for the long term access. Whenever a vehicular node requests the routing information, it first checks the client storage to see of the required routing information can be accessed more quickly, rather requesting from the remote server. The biggest challenge of using remote storage is to keep the data stored in the client and remote storage in sync, i.e. the routing information should be same one [12].

A. Data Expiration Model

The routing information stored in remote storage uses the following types of data expiry model

Time Based Data Expiry Model: This model is considered to update the routing information independently [13]. The best option to use the data in sync is make the data expire by fixing a certain period of time. If the routing information is expired with time means, it will be removed from the client server. An

updated routing information will be retrieved from the remote system and then it will be used in the client storage machine.

Query Based Data Expiry Model: This model is used to delete or update the data by using the queries [14]. By just sending a query to the client storage, the routing information can be removed that was updated. This type is used to make updated information as possible after making the update in the remote system. It is not used to detect changes in the remote storage. The following is the complete algorithm of the proposed RS-CBAODV routing protocol which uses RST to store routing information of the network and gets routing information updates from the remote server.

B. Data Replication Models

Based on this proposal, the RSTs with client storage is connected to a remote storage to backup the routing information permanently in the network. A vehicular node could connects to any RST in the network but all the RST is connected to a single remote storage. This method of copying routing information in one node to another node helps all the users to access the same level of information and it is called as Data Replication. Based on this technique, the routing information is distributed to the entire network in which each node can access the relevant routing information to their tasks without disturbing with the request of other nodes. This idea of data replication is used to eliminate data ambiguity or inconsistency for storing vehicular data among the different nodes.

By using Photocopy based data replication model, the entire routing information in one RST is copied to the remote storage server. Association based data replication model, the routing information from two or more RSTs is combined and stored in a remote storage server. However the Task based replication model is used to deliver the complete routing information to user and then the vehicular user gets the periodic updates whenever there is a change in the network.

This proposed model is used for changes, additions, modifications and deletions of routing information performed on any RST is indicated to the remote server. So, this is automatically reflected in the routing information stored in all the other RSTs. Thus, each and every vehicular node gets the route to deliver the data that is reliable with the information accessed by the entire network.

V. PERFORMANCE ANALYSIS

The mobility pattern in VANET completely differs in terms of the boundaries of the roads and buildings which are the constraints in the network. Travel direction and speed of the vehicular nodes cannot be predicted and they travel more random in nature. It creates the critical situation by making breakage in the communication. Moreover, the connection breakage may occur at any time, because the vehicular nodes has high node mobility and fast topology changes.

The NS-2 simulator is used with MOVE tool to setup and to stimulate the realistic vehicular ad hoc network [15]. The vehicular nodes are placed randomly and the RST with client storage option is used as a fixed node in the network. These RSTs are controlled by the remote server to backup the routing information. The simulation parameter of the proposed routing algorithm for NS2 simulator is given in Table 1.

Assumption: RS-CBAODV doesn't uses the cache replacement algorithms to update the content of the routing information in both client and server side because it uses the concepts of TTL, no. of hops, sequence numbers and HELLO messages only [16]. The RST and remote server is considered as unlimited storage capacity. The remote server was implemented to update the contents to all the RST to support client and server concepts. Here, the client storage and server storage is always reachable by the RST and vehicular nodes. Moreover, this network is simulated with random queries that could request by the vehicular nodes and the RST.

Table 1. Simulation Parameter

| Simulation Parameter	Value
Tool	NS2 – 2.34 and MOVE
Data Packet	512 bytes
MAC Type	Mac / 802.11
Interface queue type	Queue / Drop Tail
Link Layer Type	LL
Channel Type	Channel / Wireless
Movement	Random
Network Interface Type	Phy / Wireless Phy
Interface Queue Type	Queue / Drop Tail / PriQueue
Routing Protocol	CBAODV and CS-AODV
Antenna Model	Omni Antenna
Radio Propagation	Two Way Ground
Number of nodes	10, 20, 30, 40, 50
Pause Time	10, 30, 90, 120, 150
Network Size (sq.m)	200, 400, 600, 800, 1000
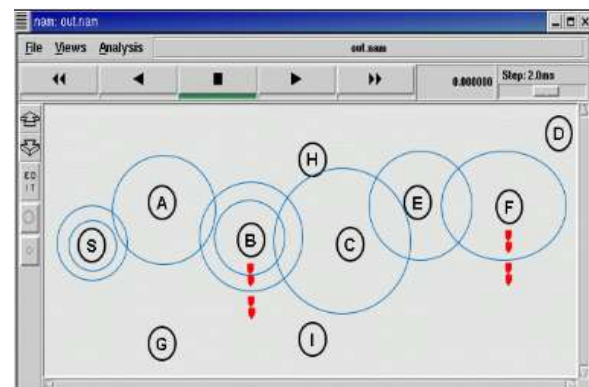
Fig.5. NS2 Simulation

The NS2 simulation environment using Remote Server Storage is given Fig. 5. In this simulation setup, node B and E are considered as RST. S is the Source Node and D is the Destination Node. Initially, the A, C, D and F are

the considered as vehicular nodes which are setup ready to participate in the network. In this simulation H is made as a remote server to monitor requests generated by the RST and vehicular nodes. The output of the proposed RS-CBAODV is compared with the CS-CBAODV protocol based on varying number of nodes, varying number of pause time, varying the size of the simulation environment. The performance analysis of the proposed RS-CBAODV protocol is done by using various Quality of Service (QoS) metrics to prove the quality of the proposal. The QoS metrics considered Packet Delivery Ratio, Average End-End Delay, Throughput and Control Overhead.

A. Varying Number of Nodes

The proposed routing protocol is supplied to the NS-2 simulator to generate the multiple trace files and NAM files according to the various parameters. The simulation results of the two protocols by varying the number of nodes in the network is given in the Table 2. The number of nodes considered in the simulation are 10, 20, 30, 40, 50.

Table 2. Varying number of nodes

Nodes	Packet Delivery Ratio		End to End Delay		Throughput		Control Overhead	
	CS-CBAODV	RS-CBAODV	CS-CBAODV	RS-CBAODV	CS-CBAODV	RS-CBAODV	CS-CBAODV	RS-CBAODV
10	94.58	98.14	0.12	0.08	61897	84562	59	48
20	86.39	89.64	0.19	0.13	58529	74935	52	48
30	69.79	73.91	0.32	0.28	48694	62491	47	42
40	63.19	68.54	0.52	0.49	51936	58327	29	24
50	62.11	64.23	0.64	0.58	49965	51963	31	28

The performance of both RS-CBAODV and CS-CBAODV is given in graphical representation in Fig. 6, 7, 8 and 9. Initially, the RS-CBAODV gives better performance in terms of all the QoS parameters. RS-CBAODV delivers more data packets and faces less delay when compared to CS-CBAODV routing protocol. RS-CBAODV generates more throughput value and undergoes a less control overhead matched with CS-CBAODV.
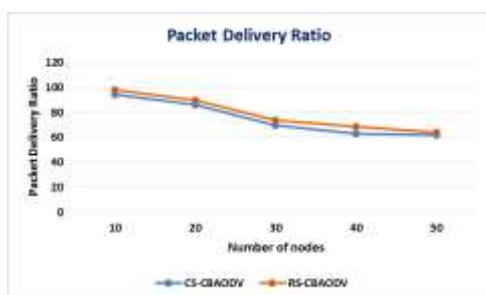

Fig.6. Comparison of Packet Delivery Ratio by varying number of nodes

In Fig.6. The Packet Delivery Ratio (PDR) is compared with varying number of nodes in the network. When Node is 10, the RS-CBAODV generates 98.14 PDR value but CS-CBAODV generates only 94.58 ratio.

RS-CBAODV performs better even the node is increased to 20, 30, 40 and 50. RST as an intermediate node to reduce connection breakage and to deliver more data packets in the network.

Fig.7. Comparison of End to End Delay by varying number of nodes

In Fig. 7, CS-CBAODV is compared with RS-CBAODV to analyze End to End Delay by varying number of nodes in the network. When 30 nodes are participating in the network, RS-CBAODV takes 0.28 seconds which is 0.04 lesser than CS-CBAODV. RS-CBAODV requires less delay to deliver data packets than CS-CBAODV even when the nodes are increased to 40 and 50. The delay is reduced by using remote storage assisted proposal. RST delivers updated data, without taking more delay to consult with remote storage.
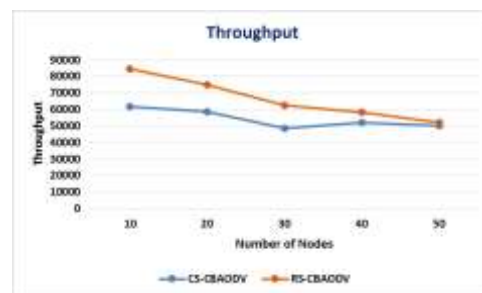

Fig.8. Comparison of Throughput by varying number of nodes

In Fig. 8, the throughput is compared by varying number of nodes for both CS-CBAODV and RS-CBAODV algorithm. When number of node is 10, throughput value generated by RS-CBAODV is 84562 which is 22665 greater than CS-CBAODV due to support of RST. The throughput value for RS-CBAODV is high when the node value is increased to 20, 30, 40 and 50.
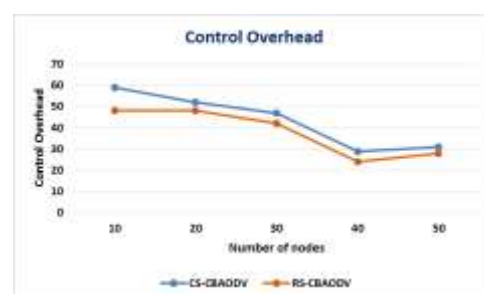

Fig.9. Comparison of Control Overhead by varying number of nodes

In Fig.9, the Control Overhead is compared by varying number of nodes for both CS-CBAODV and RS-

CBAODV. When node is 10, CS-CBAODV value is 59, but RS-CBAODV value is 48. The RS-CBAODV requires less value even the node value is increased to 20, 30, 40 and 50. CS-CBAODV requires more resources to find the routing information but RS-CBAODV requires less resources because it utilizes the routing information RST.

B. Varying Number of Pause Time

Table 3. Varying number of Pause Time

Pause Time	Packet Delivery Ratio		End to End Delay		Throughput		Control Overhead	
	CS-CBAODV	RS-CBAODV	CS-CBAODV	RS-CBAODV	CS-CBAODV	RS-CBAODV	CS-CBAODV	RS-CBAODV
10	96.98	94.44	0.32	0.27	86598	89456	57	46
30	91.65	93.74	0.35	0.23	79951	86548	42	38
90	82.42	91.19	0.42	0.37	72321	82154	38	29
120	72.54	81.65	0.59	0.43	62741	72159	31	28
150	63.61	78.12	0.62	0.57	62357	73861	28	21

The NS2 simulation is done to compare RS-CBAODV with CS-CBAODV by varying the pause time in the network like 10ms, 30ms, 90ms, 120ms, 150ms is given in Table 3. The graph in Fig. 10, 11, 12 and 13 shows the improved result in RS-CBAODV when compared to CS-CBAODV.
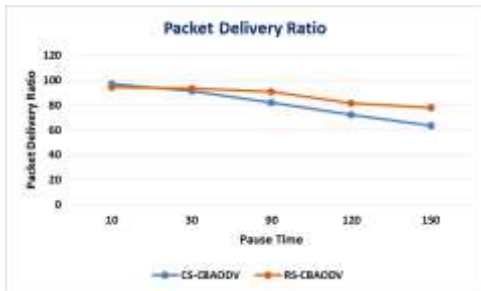

Fig.10. Comparison of Packet Delivery Ratio by varying number of pause time

The Comparison of Packet Delivery Ratio of CS-CBAODV Vs. RS-CBAODV is performed by varying number of pause time in the simulation. When the pause time is 10 and 30 both CS-CBAODV and RS-CBAODV performs similar variance in results. RS-CBAODV performs more PDR when the nodes are 150 due to the presence of RST in the network.
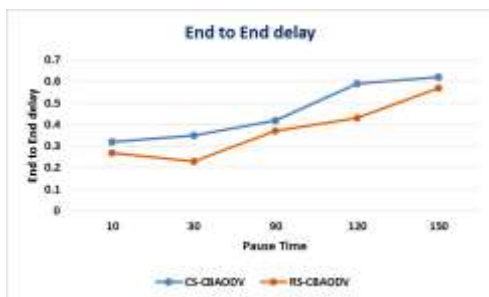

Fig.11. Comparison of End to End delay by varying number of pause time

The End to End delay by varying number of pause time is analyzed on both CS-CBAODV and RS-CBAODV in Fig. 11. The End to End delay value is .05 varies in performance when the pause time 90ms. The delay values increases in 30ms and 120ms due to network mobility.
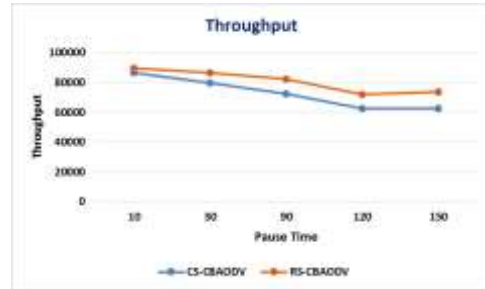

Fig.12. Comparison of Throughput by varying number of pause time

The Throughput of both CS-CBAODV is compared with RS-CBAODV by varying number of pause time which is given in Fig. 12. When the pause time is 10 ms the CS-CBAODV produces 86598 and RS-CBAODV produces 89456. There is a change in throughput when the pause time varies. The reason is due to retransmission of entire routing information and delay in route maintenance to all the vehicular nodes
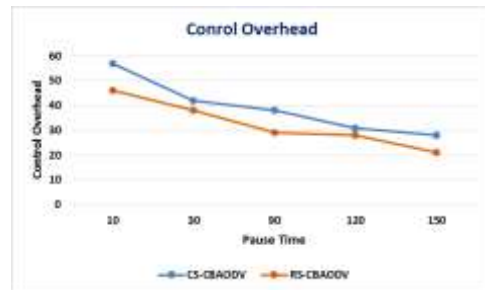

Fig.13. Comparison of Control Overhead by varying number of pause time

The performance analysis of Control Overhead for both CS-CBAODV and RS-CBAODV is performed by varying number of pause time is given in Fig. 13. Initially when the pause time is 10ms, the RS-CBAODV requires 47 whereas CS-CBAODV requires the 57. The reason is RS-CBAODV requires only a few number of bytes but CS-CBAODV requires extra bytes of information to broadcast routing information.

C. Varying the network size

Table 4. Varying the network size

Network Size	Packet Delivery Ratio		End to End Delay		Throughput		Control Overhead	
	CS-CBAODV	RS-CBAODV	CS-CBAODV	RS-CBAODV	CS-CBAODV	RS-CBAODV	CS-CBAODV	RS-CBAODV
200	88.66	92.45	0.42	0.36	86459	92145	54	43
400	82.45	91.45	0.48	0.34	85469	89456	59	42
600	75.45	79.25	0.57	0.41	62458	72458	48	35
800	72.45	83.45	0.64	0.54	65489	75489	37	31
1000	56.45	67.95	0.62	0.53	58796	62458	31	27

The NS2 simulation is done to compare RS-CBAODV with CS-CBAODV by varying the network size in the network like 200x200, 400x400, 600x600, 800x800, 1000x1000 and the results are given in Table 4. The graph in Fig. 14, 15, 16 and 17 shows the improved result in RS-CBAODV when compared to CS-CBAODV.
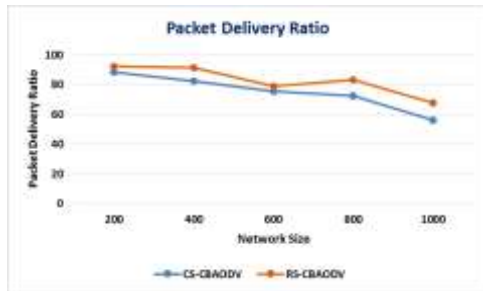

Fig.14. Comparison of Packet Delivery Ratio by varying network size

The Comparison of Packet Delivery Ratio by varying network size is given in Fig. 14 for both CS-CBAODV and RS-CBAODV. When network size is 200x200, CS-CBAODV produces PDR 88.66 but RS-CBAODV produces PDR 92.45. At 1000x1000 size, the PDR of RS-CBAODV reduces to 67.95 due to increase in network. In that case, the vehicular nodes move randomly to any corner of the network and it produces very less PDR.
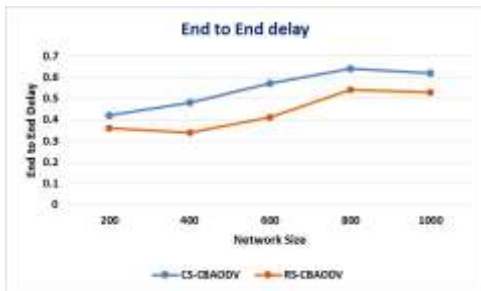

Fig.15. Comparison of End to End Delay by varying network size

The Comparison of End to End delay by varying network size is presented in Fig.15 for both CS-CBAODV and RS-CBAODV. CS-CBAODV has increase in delay in all the network sizes due to network sizes and data corruption occurred based on broken links. RS-CBAODV increases delay in 200x200, 600x600 and 800x800 but RS-CBAODV has decrease in delay rate in 400x400 and 1000x1000 sizes with the support of discovering node and finding node movement.
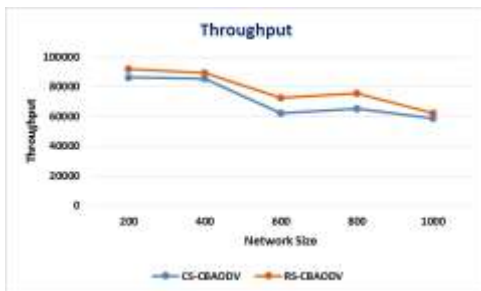

Fig.16. Comparison of Throughput by varying network size

The Throughput analysis by varying network size is done for both CS-CBAODV and RS-CBAODV is given in Fig. 16. RS-CBAODV produces throughput of 92145 when network size is 200x200 but CS-CBAODV produces throughput of 86459. The throughput value of CS-CBAODV decrease in 400x400, 600x600 and in 1000x1000 because of missing of routing information. But RS-CBAODV has higher throughput value even when the network size increases because it utilizes the support of routing information stored in RST which is placed in between the vehicular nodes. RS-CBAODV has good throughput value even when the density of the network reduces.
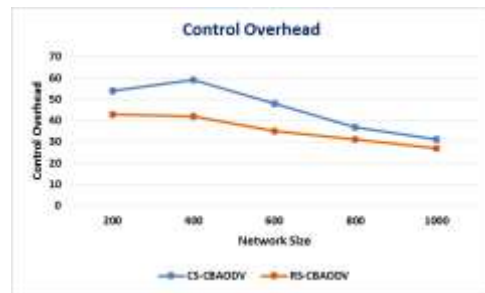

Fig.17. Comparison of Control Overhead by varying network size

The comparison of Control Overhead by varying network size is displayed in Fig.17 for both CS-CBAODV and RS-CBAODV. When network size is 200x200, the congestion occurs in the network and produces throughput value of 43 for RS-CBAODV which has difference of 11 when compared to CS-CBAODV. The RS-CBAODV has decrease in throughput value for 400x400, 600x600, 800x800 and in 1000x1000 because it requires a less number of processing time and resources than CS-CBAODV.

It is observed that the RS-CBAODV serves better than CS-CBAODV in delivering the required routing information to the multiple clients. The multiple route requests are managed and served by the remote storage server. This solution employs the concept of infrastructure to infrastructure communication between the client storage and remote storage. Finally, the availability of the routing information in this model can be further be increased by integrating vehicular cloud concepts [17].

VI. Conclusion

In this paper, the RS-CBAODV algorithm is used to make the offline backup of routing information using remote storage service to help data in the client side. Based on the performance analysis, the RS-CBAODV serves better than the CB-CBAODV protocol. Here, multiple route requests can be served by using remote storage server. The infrastructure to infrastructure based communication is used send routing information between the vehicles. In CS-CBAODV mechanism, the routing information is stored permanently in the client side to reduce the connection breakage between the vehicles. To

carry out this, RSTs are used as a static vehicular node with storage option.

In RS-CBAODV mechanism, RST will supports only to the nearby vehicles to deliver the stored client side routing information. But, when the complete network is taken to Remote storage means it will have the complete control to deliver the routing information to all the RSTs. To store data in remote side, the remote storage concept is used to deliver the routing information to all the RSTs. Then, the RST will act as client storage to deliver the routing information to the nearby vehicles. In this mechanism, the transmission of routing information using remote storage increases the data delivery and reduces the data loss when it is compared with CS-CBAODV protocol. This proposed algorithm in suggested to use in VANET to provide traffic related information to the public in real time scenario.

In future, the availability of the routing information can be increased by using vehicular cloud concepts in remote server. Here, speed, direction and distance can be predicted to incorporate intelligent transport system concepts using knowledge grid with vehicular cloud.

REFERENCES

[1] Saha, Soumen, Utpal Roy, and Devadutta Sinha. "Application of RREQ Packet in Modified AODV (m-AODV) in the Contest of VANET." In Computational Intelligence in Data Mining-Volume 1, pp. 489-502. Springer India, 2015.

[2] Arulkumar, N., and E. George Dharma Prakash Raj. "CBAODV: An Enhanced Reactive Routing Algorithm to Reduce Connection Breakage in VANET." In Artificial Intelligence and Evolutionary Algorithms in Engineering Systems, pp. 533-539. Springer India, 2015.

[3] Gabteni, Hanene, Benoit Hilt, Frederic Drouhin, Jonathan Ledy, Michel Basset, and Pascal Lorenz, "A Novel Predictive Link State Indicator for Ad-Hoc Networks", Global Communications Conference (GLOBECOM), 2014, IEEE, pp: 149 – 154.

[4] Iqbal, Faisal, Mohammad Inayatullah Khan Babar, Mohammad Haseeb Zafar, and Megat Farez Zuhairi. "I-AODV: Infrastructure based Ad Hoc on-demand distance vector routing protocol for vehicular Ad Hoc Networks." In Smart Instrumentation, Measurement and Applications (ICSIMA), 2013 IEEE International Conference on, pp. 1-5. IEEE, 2013.

[5] Raj, G. S. "An efficient reactive location based ad hoc routing protocol for VANETs." In Advanced Computing (ICoAC), 2013 Fifth International Conference on, pp. 468-471. IEEE, 2013.

[6] Kim, Min-Uk, and Kyoungro Yoon. "Fast Path Recovery Scheme for V2V Communications Using AODV." In Information Science and Applications (ICISA), 2013 International Conference on, pp. 1-2. IEEE, 2013.

[7] Ghosh, Arindam, Vishnu Vardhan, Glenford Mapp, Orhan Gemikonakli, and Jonathan Loo. "Providing ubiquitous communication using road-side units in VANET systems: Unveiling the challenges." In ITS Telecommunications (ITST), 2013 13th International Conference on, pp. 74-79. IEEE, 2013.

[8] Gerla, Mario, Chuchu Wu, Giovanni Pau, and Xiaoqing Zhu. "Content distribution in VANETs." Vehicular Communications 1, no. 1 (ELSEVIER, 2014): 3-12.

[9] Barcelos, Vladimir P., Thiago C. Amarante, Carlos Daniel Drury, and Luiz HA Correia. "Vehicle monitoring system using IEEE 802.11 p device and Android application." In Computers and Communication (ISCC), 2014 IEEE Symposium on, pp. 1-7. IEEE, 2014.

[10] Farooq, Mohd Umar, Mohammad Pasha, and Khaleel Ur Rahman Khan. "A data dissemination model for Cloud enabled VANETs using In-Vehicular resources." In Computing for Sustainable Global Development (INDIACom), 2014 International Conference on, pp. 458-462. IEEE, 2014.

[11] Vondra, Michal, Soufiene Djahel, and John Murphy. "VANETs signal quality-based route selection in smart cities." In Wireless Days (WD), 2014 IFIP, pp. 1-8. IEEE, 2014.

[12] Abdessamed, Derder, and Moussaoui Samira. "Target Tracking in VANETs Using V2I and V2V Communication." In Advanced Networking Distributed Systems and Applications (INDS), 2014 International Conference on, pp. 19-24. IEEE, 2014.

[13] Zheng, Jingqing, Yi Wu, Zhexin Xu, and Xiao Lin. "A reliable routing protocol based on QoS for VANET." In Advanced Infocomm Technology (ICAIT), 2014 IEEE 7th International Conference on, pp. 21-28. IEEE, 2014.

[14] Chaqfeh, Moumena, Abderrahmane Lakas, and Imad Jawhar. "A survey on data dissemination in vehicular ad hoc networks." Vehicular Communications 1, no. 4 (2014): 214-225.

[15] Mathews, Sam. "An effective strategy for pseudonym generation & changing scheme with privacy preservation for vanet." In Electronics and Communication Systems (ICECS), 2014 International Conference on, pp. 1-6. IEEE, 2014.

[16] Yang, Yang, Qian Liu, Zhipeng Gao, Xuesong Qiu, Lanlan Rui, and Xin Li. "A data dissemination mechanism for motorway environment in VANETs." EURASIP Journal on Wireless Communications and Networking 2015, no. 1 (2015): 1-11.

[17] Dressler, Falko, Philipp Handle, and Christoph Sommer. "Towards a vehicular cloud-using parked vehicles as a temporary network and storage infrastructure." In Proceedings of the 2014 ACM international workshop on Wireless and mobile technologies for smart cities, pp. 11-18. ACM, 2014.

Authors Profiles

N. Arulkumar is currently working as an Assistant Professor of Computer Science, Bishop Heber College, Trichy. He has completed his Master of Computer Science in 2008 under University of Madras. He has also completed his M.Phil. Computer Science in 2010 and M.Tech. Information Technology in 2012 under Bharathidasan University. At present, he is pursing his Ph.D. in Computer Science under Bharathidasan University. He has published several papers in both Conferences and Journals at National and International level.

Dr. E. George Dharma Prakash Raj completed his Masters Degree in Computer Science and Masters of Philosophy in Computer Science in the years 1990 and 1998. He has also completed his Doctorate in Computer Science in the year 2008. He has

around twenty-four years of Academic experience and sixteen years of Research experience in the field of Computer Science. Currently he is working as a Faculty in the School of Computer Science and Engineering at Bharathidasan University, Trichy, India. He has published several papers in International Journals and Conferences related to Computer Science and has been an Editorial Board Member, Reviewer and International Programme Committee Member in many International Journals and Conferences. He has convened many National and International Conferences related to Computer Science.