An Improved Route Cache Method for Dynamic Source Routing Protocol

Ragb O. M. Saleh[1] and Abu Baker A. Aldeeb[1]

Abstract:

Mobile Ad-hoc Networks (MANETs) are described by connectivity through a collection of wireless mobile nodes along with rapid changing network topology. Wireless mobile nodes are free to move autonomous of each other which make routing processes more complicated. In order to assist communication in the network, a routing approach is required to discover and maintain routes between wireless mobile nodes. This article suggests a new approach for MANET routing protocols, specifically for Dynamic Source Routing (DSR) protocol. We named it IRC. Basically, IRC utilizes an Improved Route Cache method for efficient route caching. It aids mobile nodes to reorder their route caches as soon as a new route has discovered. Since IRC employed two different concepts; freshness and shortness of the source route to select an efficient source route for possible future use. As results, excessive experiments show that the improved DSR (which uses IRC method) outperformance the standard DSR protocol in several simulation scenarios, specifically: the packet delivery ratio, routing overhead, and dropped packet ratio with respect to the mobility. All the simulation scenarios have been carried out with GloMoSim simulator.

Key wards: *MANET, DSR, Source Routing, Shortest Route, Route Caching*.

[1] Faculty of science, Azzytuna University

1. Introduction:

Mobile Ad-hoc Networks (MANETs) (Barbeau et al, 2007) (Basagni et al, 2004) consist of wireless mobile nodes communicating without the aid of any centralized administration or established infrastructure. MANET is deployed in circumstances where no access point is available, and a network has to be built unplanned. The hosts in MANET move arbitrarily, self-organized and decentralized nodes. Furthermore, all hosts can be mobile and the topology of the network change continually. Foremost challenges in MANETs are routing of data packets with often free nodes movement. Thus, routing protocols are an important issue in MANET communications. One of mainly popular MANET routing protocols is reactive routing protocols (Kashif et al, 2017) (Abu-Salem et al, 2014) (Gurdeep et al, 2017), such as DSR (Dynamic Source Routing). These types of routing protocols construct routes only when a source node wants to send data packets to some destination node. The source confirms for route availability. If there is no route exists, it invokes the route discovery mechanism to discover a new route to the intended destination (Golsum et al, 2018). The route discovery process typically consists of the network-wide propagating of a route request message. When a route to the destination node has been established; this route maintained by route maintenance mechanism. In routing protocols, one of the majority concerns is how to get a recent cached route to the intended destination in presence of portable nodes (Barış et al, 2017) (Saleh et al, 2013).

In this article, we study the performance of two route caching strategies (improved DSR and standard DSR protocol), and compare their outcomes in same simulation environments. Since the route caching of standard DSR is based on “shortest route first”, whereas the improved route caching method is based on “freshness and shortness route”.

2. Related Work

This work (Barış et al, 2017), presents the analysis of the effects of the “Route Cache Timeout” parameter on the performance of DSR protocol. Since, “Route Cache Timeout” parameter of the DSR protocol is analyzed using the network simulator. This parameter is analyzed both in high mobility and low mobility ad hoc networks. High mobility and low mobility simulation environments were created in the simulator and the simulation scenarios were run with 150s, 90s, 60s, 30s and 10s Route Cache Timeout values. The PDR of the simulation results is examined and analyzed. PDR is a significant performance metric to calculate the performance and throughput of a MANET network. Simulation results demonstrated that the Route Cache Timeout parameter considerably changes the throughput of the DSR protocol in MANET networks. It's examined from simulation results that the outcome of the Route Cache Timeout parameter on the PDR performance of the DSR protocol is not same in high mobility and low mobility environments. It concluded that before using the DSR protocol in a MANET network, the mobility of nodes should be properly examined and the Route Cache Timeout parameter should be set to a suitable value according to the level of mobility.

In other work by (Abu-Salem et al, 2014), it made analysis about the route cache sizes in DSR protocol using the NS-2 simulator. In the simulation scenarios, they used ad hoc networks having relatively high mobility. From the simulation results, they observed that route cache size significantly affects the DSR protocol's performance. In their simulation scenarios, they observed that optimum performance is obtained when primary route cache size is between 5 and 10 routes and secondary route cache size is between 10 and 20 routes.

Since, work by Zaiba Ishrat et al. proposed a more effective method for route selection in DSR protocol (Ishrat et al, 2013). In DSR protocol, if there is more than one route from source to destination, the route having minimum hop count is preferred. Zaiba et al. proposed the "path ranking" technique. In this technique, each node in the network is ranked according to some criteria and the rank of each node is updated periodically. The rank of a route is the sum of all nodes' ranks on the route. When there is more than one route from source to destination, the route having the highest rank is chosen. For instance, when there is congestion at a node, its rank is lowered. By this way, the nodes having congestion is less preferred.

3. An Improved Route Caching Method for DSR protocol

In DSR protocol, nodes automatically discover and maintain the routes in the network by storing source routes, discovered dynamically on-demand. They maintain route caches that contain the source routes which the node is aware of (Gurdeep et al, 2017) (Gani et al., 2009). In the route cache, entries are continually updated when new routes are learned, and reorder the cached routes according to shortest path policy. In addition, the node caches the route when overhear it or forward a packet for future use (Saleh et al, 2013). It can store multiple routes per destination, and check its route cache for a suitable route before initiating a new route discovery process. Cached routes prevent the flooding early, also reducing the overhead and delay. Despite those advantages, route selection will not be an efficient method without an effective route caching strategy (Husieen et al., 2011). However, our concern is that routes which maintain in caches might be into disorder. For instance, if the route cache contains multiple routes per destination, and have the same number of hop counts (Gurdeep et al, 2017). These cached routes might be disorder despite equality of the number of hop counts and their destination. In this case, the source node uses the shortest path policy to select the source route to the destination from its route cache, regardless of the time of construction the route. The problem with this approach is that, while the source is still using the primary shortest route, the primary route might fail, and the source would remain unaware of that its cache contains a recent/fresh route, which has the same number of hop count, and to the same destination. To solve this problem, we present a new strategy for route caching process based on the recent-shortest route between a source and a destination in the network. The key of enhancement in the proposed approach is that the performance of DSR protocol can be achieved by caching a stable route for possible future use. It called IRC scheme. Basically, IRC tries to provide the recent-short source route to the intended destination based on construction time of the source route; where IRC allows mobile nodes to re-sorting their route caches as soon as a new route has learned. Since the re-sorting process will do according to the principle of recent-short route first. In addition, IRC presents some advantages; nodes can save its resources (i.e. bandwidth and power consumption) by reducing recall the route discovery process, which is costly. Other advantages can be achieved by using this approach is some performance objectives such as high delivery ratio, low overhead and fewer dropped packets. As the response to solve route caching problems in DSR protocol, we introduce a new route caching strategy that utilized two different concepts; freshness and shortness of the source route to select an efficient source route for possible future use.

- The Description of the proposed IRC strategy:

Basically, IRC is a method to help DSR protocol to organize its route cache, and provide an efficient source route from the source to the destination in the network. However, we assume that the source node has more than a source route (SR) in its route cache. In addition, we consider that SR has three operators:

i. The construction time of SR.
ii. Number of hop-count of SR.
iii. The source of SR (a destination/ an intermediate node).

Whereas, we classify SRs to three types:

i. SRD: (SRs reply by the destination node).
ii. SRMHC: (SRs have the minimum number of hop-count).
iii. SRRCT: (SR has the recent construct time).

According to previous assumptions, IRC extracts the Recent Source Route (RSR) to the destination as follows:

$$RSR = SR_{RCT}(SR_{MHC}(SR_D(SR_n)))$$

Where: SRn is set of SR per destination.

- Put the RSR on the top of cached source routes for future use.

Essentially, the principle of IRC strategy is to organize the route cache, and finding out the best cached source route which has: the recent construction time and minimum hop-count of the cached routes, which sent back to the source node by other nodes. However, IRC has three phases during its routing process, which will be described as follows:

Phase 1: If the source node wants to send data packets:

- If the node's cache has one or more routes; it selects the recent-short route in its route cache according to IRC strategy.
- If the node's cache has not a route to the destination; call the route discovery process, where the source node propagates RRQ packet, and wait for RRP packet.

Phase 2: If the source node receives a new RRP:

- It re-sorting its cached routes as soon as a new route has received according to IRC strategy.
- Stop the propagating of RRP.

Phase 3: If an intermediate node receives a new RRP:

- It re-sorting its cached routes as soon as a new route has received according to IRC strategy.
- Forward the RRP to the next intended node toward the source node.

4. Pseudo-Code of the IRC Strategy

In this subsection, we present a pseudo-code to describe the details of IRC method. Table (1) lists the common variables and functions that are used in the pseudo-code. In this manner, nodes call IRC as soon as receives a new source route during routing process in the network. IRC tries to provide the recent-short route for the intend destination using its routing information, such as the number of hop counts, and the construction time of the source route.

Table 1: Common Pseudo-Code Variables and Functions

Variable	Description
S	The source node.
D	The destination node.
SR	The source route which selected by S to send its data.
SR_D	Set of SRs have same hop-count per destination.
RC	Rote cache is provided to store SRs.
RSR	The recent-short route between two nodes.
T	A specific time for route discovery process.
Check_RC()	Function to find a specific route in the node's RC.
IRC()	Function to cache and select the recent-short route to D.
RRP	A route reply packet which replied to S.
RRP→SR	The SR which send back by D or an intermediate node to S.
RRP→D	The intended destination for RRP.
Crnt_node	The current node which received the RRP packet.
Send_RRQ()	Function uses to send RRQ packets for D.
Send_Data()	Function uses the selected SR to send data packets to D.
Insrt_RC()	Function to save the new SR and its construct time in RC.
SR_c	The construct time of SR.
SR_{new}	A new source route.
Forward()	Function to forward RRP to the next intended node.
Free()	Function to stop forwarding the RRP packets.
Reorder_RC()	Function to insert the new SR in the suitable order in RC according to the recent-short route first policy.
Crnt_node_adr	The current node address
Intnd_nod_adr	The address of the intended intermediate node of RRP.
n	Number of SR in RC.

The application of IRC will be illustrated via an abbreviated pseudo-code as follows.

```
Input: S: source node; D: destination node;
    SRD:Set of Source Routes per
        destination, RC=Set of SRD.
Output: The Recent-Short Route (RSR)
Initialization: T= n sec; RSR =∅;
______________________________
Begin
//when a source wants to send a data
Check_RC():
Begin
If (∄(SR∈RC & SR∈D)) Do
Run Route_Discovery(D):
 Begin //process at source node
  Send_RRQ(D);
  For t=1 to T Do
   // waiting T sec for RRPs to D.
   while (∃(RRP)) Do
    If (Crnt_node=( RRP→D))Do
    { //if the current node is D
     SRnew=RRP→SR;
     Free(RRP);//don't forward RRP
     Call  IRC (D,SRnew,SRt);// return RSR
    }
   else //if the current node isn't D
   {
     Insert_RC(RRP→SR);
     SRnew=RRP→SR;
     Call  IRC (D, SRnew,SRt);
     Forward(RRP);
   }
  EndWhile;
End; //end of route discovery.
//Using the selected RSR to send data
If (∃(RSR∈ RC & RSR∈D)) Do
  Send_Data(RSR);
 else
  Recall Route_Discovery(D);
End if ;
// Update the route cache and return the recent-short route
//Insert the new SR in the suitable order in the route cache, and reorder cached routes
  according to the recent-short route first.
  IRC ():
Begin
  Foreach(SRnew∈D & SRnew∈SRD) Do
  Insert_RC(R.RP→SR);
  Reorder_RC(SRnew ,SRt );
  Return (RSR);
End; // end IRC .
```

5. Simulation Environment

Basically, the proposed IRC route caching scheme of DSR protocol has been simulated by using GloMoSim simulator. In addition, we evaluate our proposed scheme by a comparison with the DSR protocol. For all the simulations carried, a total of five simulation runs have been carried out for each performance metric and the simulation results are discussed below. However, the simulation model and the performance parameters of the proposed scheme as shown in Table (2).

Table 2: Simulation Parameters

Parameter	Value
Routing-Protocol	DSR/ IRC
Simulation-Time	900s
Terrain-Dimensions	2200m x 600m
Number-of-Nodes	50 mobile nodes
Max speed	10m/s
Mobility Model	Random Way-point Model
Pause Time	0, 300,600 and 900s
Bandwidth (in bits per second)	2Mbps
Mac-Protocol	IEEE 802.11
Promiscuous-Mode	Yes
Network-Protocol	TCP - UDP
Data traffic - CBR	4 UDP packets a second
Packet Size	512 bytes

6. Simulation Results and Performance Analysis:

The purpose of below figures are to show the development given by IRC scheme concerning packet delivery ratio, routing overhead and dropped packet ratio with respect to the mobility (varying pause time).

a) Packet Delivery Ratio (IRC vs. DSR)

In this subsection, Figure (1) shows the Packet Delivery Ratio (PDR) with respect to the mobility of both schemes (IRC and DSR).Basically, the PDR of both schemes reduces when the mobile nodes move more rapidly because high mobility is more prone to link failures, which may force the protocol to select a broken route. In all mobility environments, IRC introduces more PDR compared to standard DSR. This is because IRC has well-organized out caching scheme, since it provides the recent-shortest cached route to the intended destination for future use. The other observation is that IRC scheme always provides the recent and shortest route of the alternative routes available in the case of broken link. On the other hand, DSR route caching does not support the freshness of source route, which may increase the possibility of link failure of the candidate cached route. This may lead to decrease the PDR of the DSR. However, IRC has a generally positive effect on the caching process of the routing protocol. As the transmitted data packets can typically travel earlier and with less possibility of drop packets through a recent cached route, which enhances the total PDR in the network. As a result, IRC improves the PDR over DSR in all mobility environments, since IRC improves the average of DPR by between 4% and 10% over typical DSR.

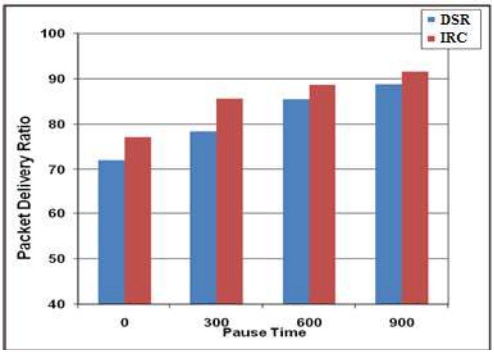

Figure (1): The PDR (IRC vs. DSR)

b) Routing Overhead (IRC vs. DSR)

For this experiment, Figure (2) shows the routing overhead (OR) of both schemes (DSR and IRC) with respect to the mobility. The OR for both schemes increases when the mobile nodes move more rapidly because high mobility is more prone to link failures. This may force the protocol to invoke new route discovery processes, which is the main cause of OR in the network. In all mobility environments, IRC scheme introduces less OR compared to standard DSR. The reduced OR due to the well-organized route caching strategy, which provides the recent-shortest route to the intended destination. In this manner, IRC's nodes may do not need to recall a new route discovery for more than a needed route. In constract, DSR route caching scheme provides just the shortest route regardless of the validity of the route. Thus IRC introduces less RO compared to DSR. However, IRC offers one or more recent cached route for future use that prevent recalling route discovery, in addition IRC is very helpful to provide sources and intermediate nodes by alternative available routes in case of like failures. As these actions of IRC scheme outcome less RO in the network. As a result the RO of the IRC is better than that of the DSR in all mobility environments, since IRC improves the average of RO by between 6% and 13% over typical DSR in the simulation experiments.

c) Dropped Packet Ratio (IRC vs. DSR)

For this experiment, Figure (3) shows the Dropped Packet Ratio (DPR) with respect to the mobility. Basically, the DPR increases when the mobile nodes move more rapidly because high mobility is more prone to link failures. Furthermore, when the node fails to select an available cached route to send its data. In almost all mobility environments, IRC scheme introduces less DPR compared to DSR scheme. The reduced DPR of IRC scheme is due to its well-organized route caching strategy. The other observation is that IRC's route caching scheme provides a recent-short route in case of data salvaging process, which lead to decrease the DPR in the network. On the other hand, DSR does not have an efficient route caching strategy; this may lead to caching more stale routes. Thus DSR's nodes may drop more data packets due those stale routes; in addition it may fails to salvage these data packets. This may increases the number of drop packets of DSR compared to IRC. However, the experimental results demonstrate that IRC drops much fewer data packets than DSR in low mobility networks and the difference decreases dramatically as the speed increases. As a result, the DPR of the IRC scheme is better than that of the DSR in all mobility cases. The observation is that IRC improves the DPR by between about 1% and 13% of data packets dropped by DSR in network environments.

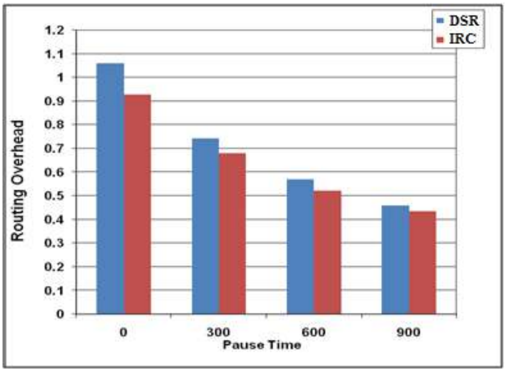

Figure (2): The RO (IRC vs. DSR)

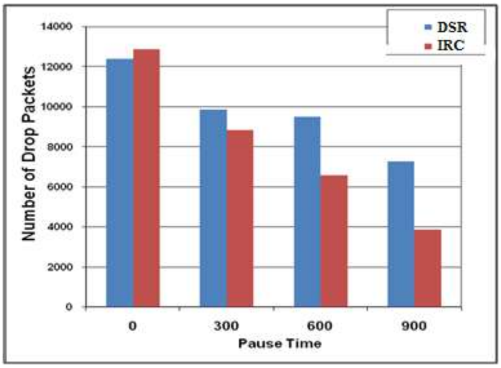

Figure (3): The DPR (DSR vs. IRC)

7. Conclusion

We present a new strategy for route caching of DSR protocol called IRC. It based on the freshness and shortest route as route caching metric. While standard DSR based on the shortest route metric. In addition, our contributions in this paper are the investigation about the impact of route caching metrics on the performance of the DSR protocol, and a comparison of the performance of IRC with standard DSR. This article evaluated the performance of IRC and DSR using GloMoSim simulator. Comparison was based on the packet delivery ratio, routing overhead and dropped packet ratio. We concluded that in most simulation scenarios IRC presents better performance as compared to standard DSR in terms of packet delivery ratio, routing overhead and drop packets ratio. Also we have seen that standard DSR protocol is superlative in term of drop packets ratio in individually high dynamic network scenario (only when Pause Time equal zero sec).

References

Abu-Salem A. O., Samara, G., and Alhmiedat, T. (2014). Performance analysis of dynamic source routing protocol. Journal of Emerging Trends in Computing and Information Sciences, 5, 97–100.

Barbeau M., E. Kranakis, Principles of Ad-hoc Networking, Wiley, 2007.

Barış O., Ibrahim A. Dogru, Muhammet A. Akcayol. "Analyzing the Route Cache Timeout Parameter of DSR Protocol in Mobile Ad Hoc Networks", International Journal of Computer and Communication Engineering, Volume 6, Number 1, January 2017.

Basagni S., m. Conti, and i. Stojmenovic, Mobile Ad Hoc Networking, Wiley IEEE Press, 2004.

Golsum Najafia, Sajjad J. Gudakahriz, "A Stable Routing Protocol based on DSR Protocol for Mobile Ad Hoc Networks", I.J. Wireless and Microwave Technologies, 2018, 3, 14-22.

Gurdeep K., Vinay Bhatia, Dushyant Gupta, "Comparative Study of the performance of existing protocols of MANET with simulation and justification of an improved Routing Protocol", International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE), Volume 6, Issue 6, June 2017.

Ishrat, Z., and Singh, P. (2013). An enhanced DSR protocol using path ranking technique. International Journal of Engineering Research and Applications, 3, 1252-1256.

Kashif H. Memon, Muhammad A. Q., Sufyan M., Mohsin S. and Ramesh K., "performance evaluation of routing protocols under security attacks in mobile ad hoc networks", IJCSNS International Journal of Computer Science and Network Security, VOL.17 No.12, December 2017.

Saleh R. O. M., Md Yazid Mohd Saman1 and M Nordin A Rahman, "Impact of Route Selection Metrics on the Performance of DSR Protocol for MANETs", IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 2, No 3, March 2013.