Dynamic Decision Protocol For Adhoc Network

Mohamed Ben Othman, Member, IEEE,

Assistant Professor
Department of Computer Science, College of Science, Qassim University
Saudi Arabia
Email: mtothman@yahoo.ca

Abstract- **Several researches were conducted to Ad Hoc networks because of their facility of deployment, no need of an infrastructure and many other features. Thus, they still present many unsolved issues such as the hidden node problem, collisions, latency and overhead. Many protocols were proposed such as the Dynamic Source Routing (DSR) [1], Destination-Sequenced Distance-Vector (DSDV) [4], Ad Hoc On-Demand Distance Vector Routing (AODV)[2,3] and several others were derived. Our approach is using cluster head like the protocol Cluster Switch Group Routing CSGR[5]. One limitation of CSGR is that it is based on DSDV as the underlying route update method, which can cause problems. The other limitation is that it uses periodic route and cluster membership updates. The key feature of the protocol is that it uses a cluster head. The new idea in our protocol is that it combines several of such protocols. There is no need to know the cluster head between groups and then no need to update this information when a cluster head changes. Using the broadcast nature of adhoc networks all routing structures like neighbor list, head list and so on are updated without increasing the overhead in the system. In this paper we present the Dynamic Decision Protocol (DDP) and the simulation under which the measurements are conducted followed by the obtained results.**

INDEX TERMS — **Protocol, Adhoc, Routing, Cluster.**

I. INTRODUCTION

The main issue focused on in the Ad Hoc networks is a communication protocol that determine a path between the source node to the destination with a minimum overhead and keep the routing information updated to minimize the latency when sending data. Several protocols were proposed such as DSR[1] where the source node floods the network with the path request message and each node broadcast the message the same way until it reaches the destination. When receiving this request, the destination sends back a reply message containing the list of the nodes in the path. Depending on the path, the destination may need to send a request message to the source if the connection is unidirectional. Many optimizations were done on this protocol to minimize the overhead introduced by flooding the network such as caching where each node caches a new route it learns by any means. The protocol DSDV [4] is a distance vector protocol, which uses the modified Bellmann-Ford algorithm. This is a table-driven protocol, where the route is always available.

However, the protocol has some limitations as well. It maintains routing info among all the nodes, it uses periodic update messages. Mobile nodes maintain routes to all possible destinations and exchange routing info between each other. Hop counts are used as routing metrics, and in order to ensure that the routing information is up-to-date, sequence numbers are used. A given node keeps track of its own time and the sequence of events that happen. Thus the node assigns sequence numbers to distance vector updates, which updates contain information about the neighbors. CSGR - Cluster Switch Group Routing [5] is just like DSDV. Some of the key features of the protocol are that it uses a cluster head, code separation between the clusters and cluster-based channel access and routing. One limitation of this protocol is that it is based on DSDV as the underlying route update method, which can cause problems. The other limitation is that it uses periodic route and cluster membership updates. The protocol is based on the concept of clusters and cluster-heads. Routing is done via the cluster-heads and gateways. Data from a host is routed in such a way that it is sent to the nearest cluster-head, which then forwards that to a gateway node, which then sends it to the next cluster-head. The cluster member table is broadcasted periodically in order to have up-to-date information about the clusters. For this reason there is also need for cluster management. In the DDP protocol there is no need for this cluster member table and no need for nodes to know about far heads or clusters. The disadvantage of reactive protocols like DSR is the huge overhead caused by broadcasting the route request message. In the DDP protocol this overhead is minimized by using cluster technique and then the route request message is transmitted only by cluster-heads and gateways.

II. DDP PROTOCOL DESCRIPTION

The DDP protocol is based on cluster head to gateway communication technique. There are two sets of communication messages: the first set consists of setting up and maintaining a group and the second is used for setting a path and communicating data. The first set of messages is as follow:

1- **Hello** message: each node sends a hello messages when it wakes up and any time it looses contact with other nodes (does not hear from others within an interval of time). A hello message is sent a number of times (three times in our simulation). One of two states happens:
    a. A group head receives this hello message and replies with a **welcome**-to-group message and then this node will be part of this group.
    b. Or no reply is sent back and then after all hello messages are sent the node claims itself as a group head.
2- A group head should send periodically a **headmsg** message. This message is received by all neighbors of this node. A node that receives this type of message has one of two situations:
    a. Is a group head and in this case one of these two group heads must resign. The choice of which of them must resign can be done by several ways for example: the older one is kept, the one has the biggest ID or each sends a random number and the one has the biggest number is kept as a group head. A resignation message is sent then by the one is no longer a group head.
    b. Or is not a group head and then it has to add the head ID into its head list. This list contains all the head IDs of the groups to which this node belongs. A node that has more than one group head (belongs to more than one group) set itself as a gateway. Each node has a timer for each head of a group the node belongs to. When the timer expires the head ID is removed from its head list. When this head list is empty the node starts to send a hello messages. Every time a **headmsg** is received the timer corresponding to the sender-head is initialized again. The **headmsg** plays the role of a welcome message when it is received by either a head node or a node that does not belong to the group.
3- A **resign** message: sent by a group head that is no longer one. This message is received by all its neighbors. All nodes receiving this message remove this head ID from their head lists.
4- Any message received by a node is used to maintain the neighbor list. When the traffic is high this list is maintained using the normal traffic and when the traffic is low the group maintenance messages do not affect the bandwidth.
5- A **welcome** message: as it is said earlier, when a hello message is received by a non-head node it is ignored. In the case when the receiver is a head it sends back a welcome message. When the welcome message is received by a node one of two cases may happens:
    a. The receiver is a head and then the same scenario between two heads happens and one of them will resign.
    b. The receiver is not a head and then it adds the new head ID to its head list if it does not exist yet.

The second set of messages is used for real communication and is as follow:

6- A **rreq** is a route request message that is broadcasted from the source node every time a node would like to initiate a communication with a destination node. When a node receives this message if it is the destination it consumes it, if it is a head node or gateway it checks if the destination is a neighbor in this case the message is sent as a unicast message or the message is broadcasted again. The message is ignored if the node is neither the destination nor a head or a gateway. Each head or gateway adds its ID to the message before retransmission. To minimize the message duplication a gateway retransmits only messages received from a head node.
7- A **rrep** message: when a destination receives a **rreq** message it sends back a unicast rrep message as a unicast message using the path received in the rreq message.
8- **Data** messages are sent by the source node through the path in the first received rrep message. Generally this is the optimal path.

III. SIMULATION RESULTS AND DISCUSSION

Each node consists of a process that can run independently and has its own structure. Each node has two queues one for messages reception and one for transmission. The transmission over-the-air is represented by a concurrent process that moves messages between neighbor nodes. Each node maintains a neighbor list and a head list. The head list is null when the node is a head, contains one element when the node belongs to only one group and contains more than one element when the node is a gateway. The air process determines the neighbors of a node by calculating the distance between this node and each of the remaining nodes. If this distance is less than the scope of transmission called Radius **R** in our simulation then this node is a neighbor.

Different measurements are conducted by modifying the number of nodes, the geography and the scope of transmission (Radius). Using 70 nodes shown in figure 1 we measure the total number of request messages, reply messages and data messages sent within the network when path is set and data is sent from node S to note D the two network extremities.

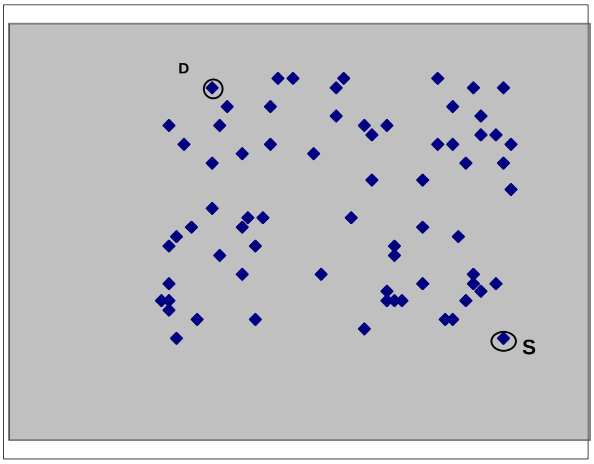

Figure 1: 70 nodes adhoc network

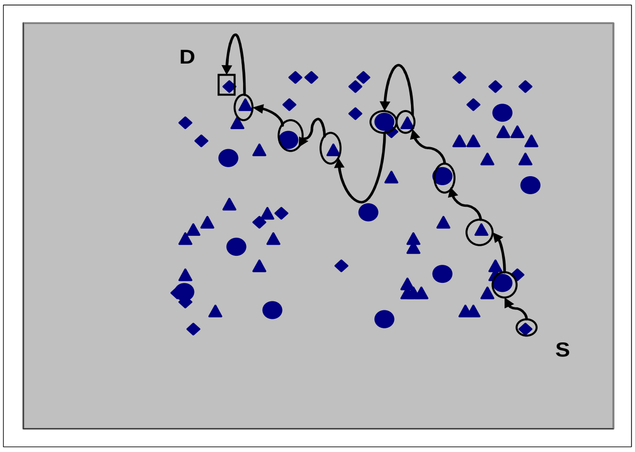

Figure 2: DDP (Radius = 8)

We used message sequence number for DSR and DDP protocols to minimize the number of transmitted messages. In the following figures we give the different paths from S to D with different Radius sizes. In cluster based network (DDP) cluster heads are drawn in circles, gateways in triangles and other nodes in squares.

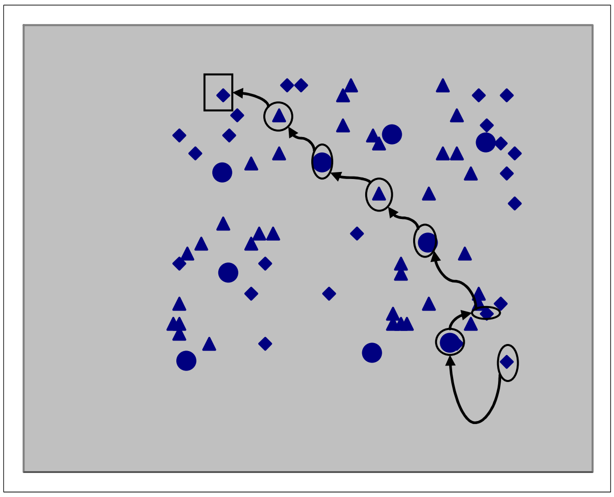

Figure 3: DDP (Radius = 10)

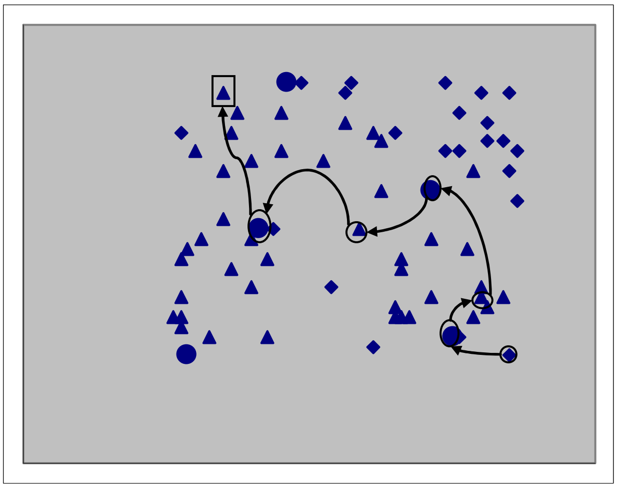

Figure 4: DDP (Radius = 15)

Top

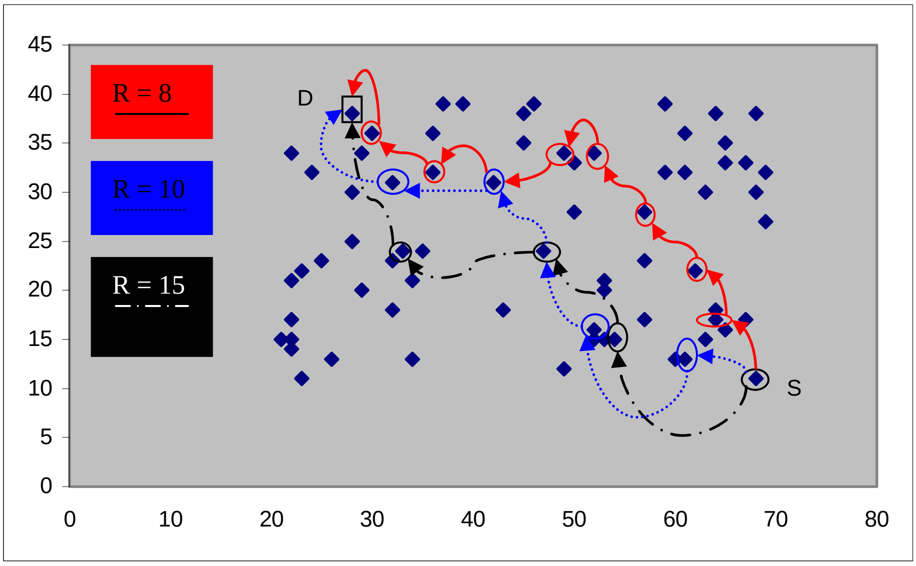

Figure 5: DSR (R= 8, 10 and 15)

In the following curves we can see that the number of request messages generated by sending one route request **rreq** message from source S to destination D is always less in DDP than in DSR. Also for the **rrep** messages the number in DDP is always less than in DSR. The difference in data message (10 data msgs are sent each time in a row) shown in figure 8 is caused by the fact that in DSR the path can be shorter than in DDP or any cluster based protocol. When the radius is small the path is more similar and then the data messages numbers are similar too. Further investigations will be done to build a cluster management protocol and have better performance.

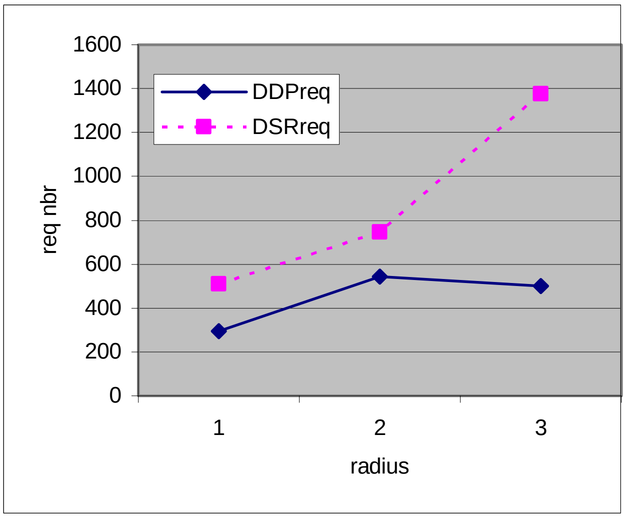

Figure 6: DSR and DDP rreq msg number

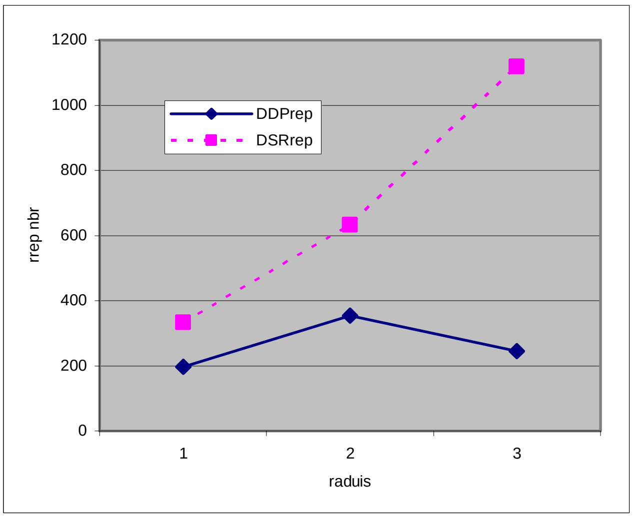

Figure 7: DSR and DDP rrep msg number

Top

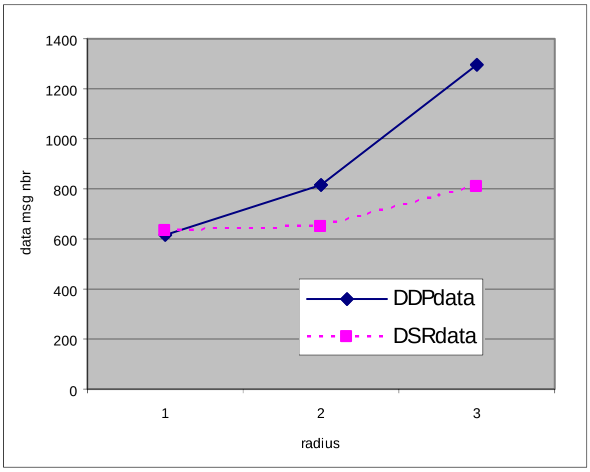

Figure 8: DSR and DDP data msg number

IV. CONCLUSION

With using a concurrent processes for the nodes of an Adhoc network and a concurrent process to determine the neighbor nodes of each node and for the transmission between these nodes we put an emulation environment of an Adhoc networks. We implement a new cluster based protocol DDP. The only information needed to be updated is within the cluster and no need for inter-cluster updates.

The results show that DDP is always better than DSR concerning path setting. Further optimization will be done to minimize the number of hops between the source and destination. Future work will consider also the impact of mobility on performance.

REFERENCES

[1] D. B. Johnson, and D. A. Maltz. "Dynamic Source Routing in Ad-Hoc Wireless Networks". *Mobile Computing*, edited by T. Imielinski and H. Korth, chapter 5, pp. 153-181, Kluwer, 1996.

[2] Elizabeth M. Royer and Charles E. Perkins. "Ad-hoc On-Demand Distance Vector Routing". *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, Feb. 1999, 90-100.

[3] S.J. Lee and M. Gerla, "Aodv-br: Backup routing in ad hoc network," in *IEEE WCNC 2000*. IEEE, 2000, pp. 1311–16.

[4] C. E. Perkins, and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance Vector Routing," ACM SIGCOMM Computer Commun. Rev., Oct. 1994, pp. 234-244.

[5] C. C. Chiang. Routing in clustered multi-hop, mobile wireless networks with fading channel. In *Proc. IEEE SICON*, pages 197–211, April 1997.

Acknowledgement: This work is partly funded by the Center of Scientific & Applied Research CSAR (QU KSA)