A Dynamic Core based Multicast Routing Protocol with Shortest Core-Passive Link (**DCMP-SCPL**) algorithm in Mobile Ad-hoc Networks

Moukhtar A. Ali and Ayman EL-SAYED
Computer Eng. and science Department,
Faculty of Electronic Eng.
Minufiya University
moukhtar_ahmed@yahoo.com, ayman.elsayed@free.fr

Ibrahim Z. MORSI
Electrical Engineering Department,
Faculty of Eng.
Minufiya University
izmorsi@hotmail.com

Abstract. In mobile ad hoc network, there are many multicast protocols that apply mesh topology and on-demand concepts such as On-Demand Multicast Routing Protocol (ODMRP). ODMRP uses periodic broadcasting transmission of control packets (e.g. Join Requests control packets) to build and maintain mesh topology. But, it suffers from higher control overhead as the network size and the number of sources increases. DCMP's designers solve the scalability problem of ODMRP by reducing control overhead. DCMP reduces control overhead by applying a new style of centralization (core) approach. It allows to each passive source to multicast its data packets only via its associated individual core source after constructing passive-core link. But, DCMP suffers from ease breakage of each passive-core link as a result of applying high mobility speeds. In this paper, we modify DCMP protocol by implementing a new algorithm called "SCPL" inline DCMP. Main target of SCPL is to make passive-core link more stable and hence to allow any source to enter passive status for a long time if it wants. Each passive source is prevented from broadcasting Join Request control packets which reduces control overhead and then improves scalability.

Keywords: ODMRP, DCMP, DCMP-SCPL**,** Multicast, Ad Hoc Wireless Networks (MANETs)

I. INTRODUCTION

Mobile ad hoc network (MANET) [1] represents a system of mobile nodes that connected by wireless links. These nodes can freely and dynamically self-organize into arbitrary and temporary network topologies, allowing people and devices to internetwork in areas without any preexisting communication infrastructure. Due to limited radio propagation range of mobile devices, communication between a couple of mobile nodes out of the transmission range of each other requires that the intermediate nodes between them to act as relays (i.e. routers). Hence, the route between two nodes may consist of hops through other nodes (i.e. multihop routing).

In typical ad hoc networks (such as classrooms, conventions, and battle fields), nodes work in groups to carry out a given task. Group communication (multicasting) plays an important role in ad hoc networks. Multicasting is the transmission of packets to a group of zero or more nodes identified by a single destination address. In wired networks, there are many multicast protocols which are usually constructing multicast tree topologies. They are surveyed in [2]. Multicast tree topology works effectively in wired networks because it prevents endless loops and to deliver data packets effectively through minimal multicast routes (shortest routes) among network nodes which characterized by their fixed locations

In ad hoc networks, topology is dynamic and unpredictable due to mobility of nodes and wireless links. Applying tree topology in ad hoc networks [3] results in two basic drawbacks: (1) ease of tree structure fragile which leads to problem of low delivery ratio and (2) tree reconstruction delay which leads to high control traffic overhead problem. To deliver multicast packets robustly, recent researches take the advantage of multicast mesh topology rather than tree topology [3]. Mesh topology refers to the possibility to provide multiple routes during data packet delivery. It means that mesh topology provides alternative paths on link failure due to continued change of network topology. Mesh-based protocols exhibit robustness. But, this robustness is at the expense of high control overhead and hence, they have a scalability problem. The problem of high control overhead becomes worst when mobility of nodes increases. Many ad hoc multicast protocols are surveyed and classified in [4].

In this paper, we modify Dynamic Core-based Multicast routing Protocol (DCMP) [5] to be more efficient and scalable especially against high mobility speeds. We implement a new algorithm called Shortest Core-Passive Link (SCPL) which built inline DCMP. DCMP operation relies basically on On-Demand Multicast Routing Protocol (ODMRP) [6].

The rest of paper is organized as follows. In section II, we introduce brief description of the protocols (ODMRP and DCMP). Motivation for implementing SCPL is described in section III. Description of SCPL is found in section IV. Simulation results are discussed in section V. Finally, we conclude this paper in section VI.

II. Related Multicast Protocols Overview

II.1 *On-Demand Multicast Routing Protocol (ODMRP)*

ODMRP is a mesh based multicast protocols. Its operation relies on the concept of forwarding group. It means that a subset of nodes is responsible about creating mesh topology and forwarding multicast packets between sources and receivers. In ODMRP, group membership and information of multicast routes are established and updated only by the sources on demand. To initialize mesh topology , a source needed to send data, will broadcast a Join Request (*JoinReq*) control packets to all nodes in the network whether they are members of group or not. Only receivers or intermediate nodes will deal with any *JoinReq*.

When an intermediate node receives *JoinReq*, information about source ID and packet sequence number must be added or updated in message cache of that intermediate node to detect any potential duplicate of the same *JoinReq*. Also upstream node ID (i.e. last node that send *JoinReq*) must be added or updated in another data structure called "*RouteTable*" to help in backing the *JoinReply* control packet (*JoinReply*) to the original source via the shortest reverse path. Then, intermediate node rebroadcasts *JoinReq*.

When a multicast receiver receives *JoinReq*, it creates and broadcasts a *JoinReply* to its neighbors. *JoinReply* has information about original source ID that *JoinReply* should reach to it and next node ID (the last node that sends *JoinReq* to that receiver). When an intermediate node receives *JoinReply*, it checks to see if the next node ID of one of the entries of *JoinReply* matches its own ID. If it does, the node sets a forward flag "*FG-Flag*" and become a member of forwarding group and the node realizes that it is on the path to the source. Then it broadcasts its *JoinReply* after updating the information of its next node ID which extracted from its *RoutingTable* and matches with the ID of original source. Each node in forwarding group acts in the same manner to propagate the *JoinReply* until it reaches the original multicast source via the selected path (the shortest route). Figure 1 shows an example of mesh initialization and propagation of *JoinReply* control packets.

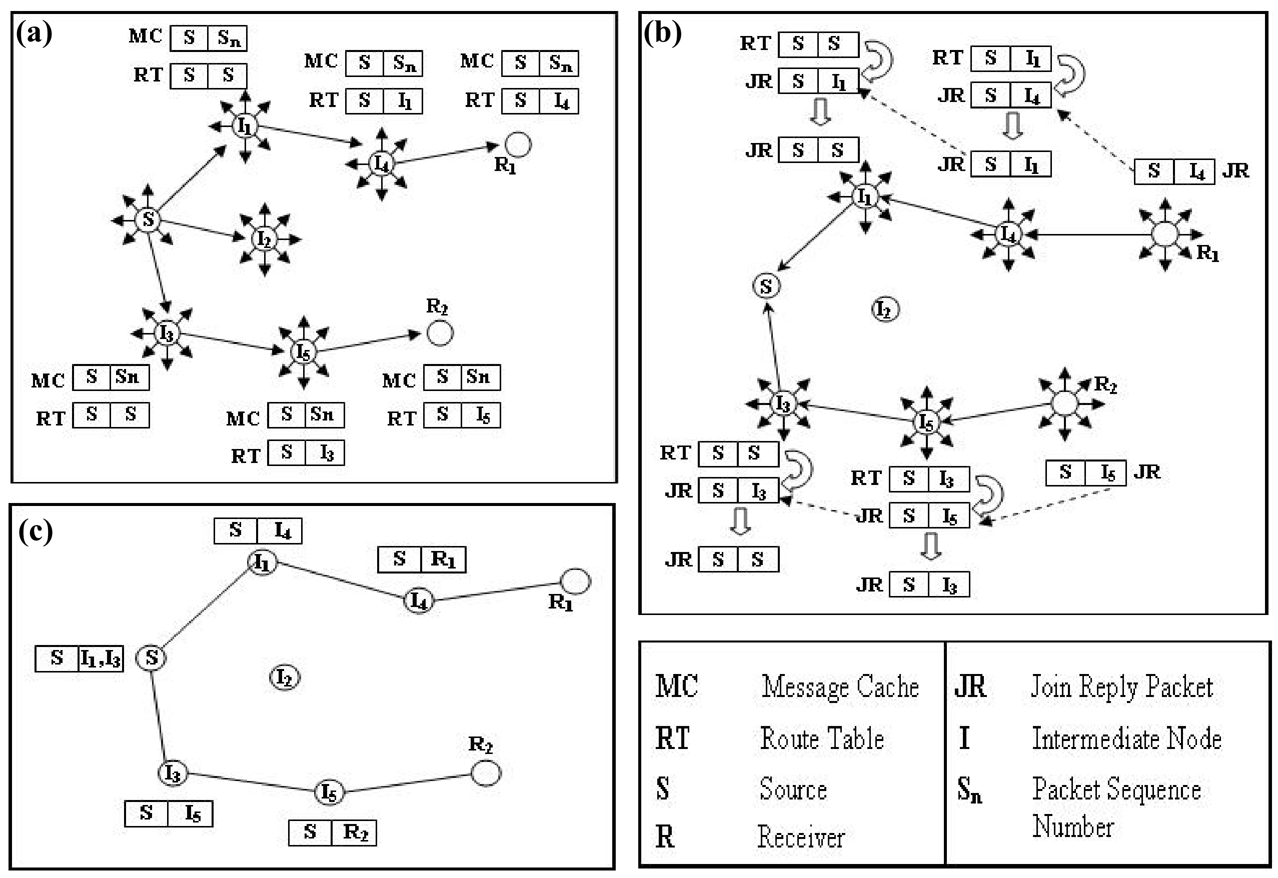

Figure 1 An example of ODMRP: **(a)** Mesh initialization by source. **(b)** *JoinReply* control packets propagation. **(c)** Final multicast information that each node requires to send and forward data packets

Figure 1.c shows that the process of constructing the routes and forwarding group between sources and receivers is completed. After that, sources multicast data packets via selected routes and forwarding groups. Sources responsible of maintaining and refreshing information of forwarding

group and routes by periodic sending of *JoinReq*. Nodes in forwarding group will forward data packets to receivers.

II.2 *Dynamic Core based Multicast routing Protocol (DCMP)*

DCMP is a modification of ODMRP which improves the efficiency of ODMRP by reducing control overhead which leads to improve scalability. In ODMRP, only receivers or intermediate nodes can response up on receiving a *JoinReq*. But, ODMRP has a broadcast nature (i.e. *JoinReq* will be sent to all network nodes in which sources are members). DCMP allows some sources (sources in ODMRP which receives *JoinReq*s and do nothing) to send its data packets only via a selected other sources (proxies).

DCMP classifies sources into three types: Passive sources, Active sources, and Active Core sources (core source for abbreviation). Active sources are similar to sources in ODMRP which floods *JoinReq*s periodically. Active core sources plays two tasks: (1) to help in constructing a mesh topology by periodic flooding of *JoinReq*s as Active sources, (2) to forward data packets received from one or more Passive sources (i.e. acts as proxy) over its mesh. DCMP prevents each passive source from sending *JoinReq*s or to be core for other sources. It allows each passive source to associate with only one core source.

DCMP's operation consists of two main parts: mesh creation and passive-core source links creation. Initially, each source floods *JoinReq*s similarly as in ODMRP with one exception that each *JoinReq* contains an additional flag called "*CoreAcceptance* flag". *CoreAcceptance* flag is used to determine if source, that send *JoinReq*, can act as a core whether it is a core for more than one passive source and number of its passive sources not exceeds a predefined *MaxPassSize* parameter, or it is an active source and it wants to be a core for another source. When an intermediate node or a receiver receives a non duplicate *JoinReq*, they act in similar manner as in ODMRP to build mesh topology. Passive-core link creation begins when an active source receives *JoinReq*. An active source changes its status to passive if all the following conditions are satisfied:

- *CoreAcceptance* flag is set.
- Hop distance traveled by *JoinReq* must be less than or equal to a predefined number called "*MaxHop Parameter*".
- ID of source, that received *JoinReq* (source will be in *ToBePassive* status), is less than ID of source, that sent *JoinReq* (source will be in *ToBeCore* status).

If all the above conditions are satisfied, the source, that want to be passive, sends a Passive Request (*PassReq*) control packet to its "*ToBeCore*" source with setting a new flag called "*CoreReqField* flag". It prevents itself from sending *JoinReq*s, becoming core for other sources, or sending *PassReq*s to other active or core sources by setting a "*Lock*" flag. Intermediate node stores ID of downstream node in a new data structure called "*ConfirmRouteFind* Table" up on receiving *PassReq*. downstream node ID is determined from information of *RouteTable* which built during propagation of *JoinReq*s. *ConfirmRouteFind* table is used in returning *Confirm* control packet back to "*ToBePassive*" source. When "*ToBeCore*" source receives *PassReq*, it checks its "*Passive supported*" entry (counts of the number of passive sources being currently supported by core source). If it is lesser than predefined *MaxPassSize*, it sends *Confirm* packet declaring that it accepts request of "*ToBePassive*" source. It then converts its status to Active Core. It adds information about its new passive in a new data structure called "*PassSourceAddr* Table" and increments count of its *PassiveSupported* entry.

When an intermediate node receives a *Confirm* packet, it sets its *FG flag* and becomes a forwarding node. It then forwards the *Confirm* packet according to its relevant entry in the *ConfirmRouteFind* table and then it deletes this entry from the table. When "*ToBePassive*" source receives the *Confirm* packet, it changes its status from Active to Passive.

DCMP maintains its mesh topology as ODMRP does. Each Passive-Core link is maintained by periodic sending of *JoinReq*s from core source to its passive sources. Each passive source ignores value of *CoreAcceptance* flag when it receives *JoinReq* from its associative core. Core source then expects a *PassReq* (its *CoreReqField* is set) from its passive sources which check previous three conditions directly after receiving *JoinReq*s. If three conditions are not met, passive source will send a *PassReq* with its *CoreReqField* reset.

When core source cannot receive *PassReq* (due to collisions or link breakage) or receives it with *CoreReqField* reset, it will delete any information about that passive source from its *PassSourceAddr* table and decrements *PassiveSupported* counter. When *PassiveSupported* entry of any core source reaches to zero, core source will convert its status to Active source. Each passive source is converted back to active status due to the following reasons: (1) when it receives a *JoinReq* that travels through hop distance more than *MaxHop Parameter*. (2) When it waits for *Confirm* packet for time more than its *Confirm-Wait-Timer*. Passive source returns to Active status by resetting *Lock* flag.

III *Motivation*

In DCMP, passive-core link can be considered as a unicast link because it established between two specific nodes' addresses (i.e. point-to-point link). High mobility of ad hoc network nodes has a great negative effect on that unicast link causing ease of passive-core link breakage and causing that passive-core sources become in a distance more than *MaxHop* Parameter. The previous two problems cause that each passive source and core source lose their functions and returns quickly to its Active status and floods *JoinReq*s as in ODMRP and hence DCMP may be lose its function wholly or partially and then return back to the operation of ODMRP.

There is another problem in DCMP. If hop distance of a passive-core link is, for example, 3 hops and if a *JoinReq* from another core or active source travels through 2 or 1 (i.e. lesser than the first link's hop distance) and reaches to that passive source. According to DCMP's function, passive source is not allowed to deal with any other sources except its associative core source. But logically, when using smaller hop distance between passive-core sources, it reduces control packets that exchanged between them and reduces the probability of ease breakage of their links. For example, if hop distance of a passive-core link is 3 hops, it means that two intermediate nodes are located on their path. Any movement (especially at high speeds) of one of four nodes (passive source, core source, or each one of two intermediate nodes) causes link breakage. But, the situation become different when number of hops be 2 or 1 hop because it reduces number of nodes participated in that new link. Our paper's motivation is to solve the previous problems by implementing a new algorithm called Shortest Core-Passive Link (SCPL) built inline DCMP.

IV. *Shortest Core-Passive Link (SCPL) algorithm*

SCPL focuses in solving problem of core-passive link which mentioned previously in section III. When SCPL algorithm works, it will take in its consideration that there is a link between passive source and its associative core source. It permits passive source to accept any *JoinReq* control packets that come from other sources (e.g. another core source or active source).

Each passive source keeps information of its core sources in a new table called "*CoreSourceAddr* Table". It accepts new *JoinReqs* from other sources according to the following two conditions: the first condition is that number of core sources, that this passive related to them, must be less than a novel predefined parameter called "*MaxCoreSize*". The second condition is that hop distance of link between passive and new core not exceeds *MaxHop* parameter. If two conditions are satisfied, passive source adds core source in its order according to hop distance that its *JoinReq* travels.

In SCPL, passive source is permitted to send its data packets via only one core source which characterized with a shortest hop distance comparing with other core sources that founded after it in *CoreSourceAddr* table (i.e. selected core source always resides at top of *CoreSourceAddr* table). Passive source maintains its core sources with periodic sending of *PassReqs* and periodic receiving of *Confirm* packets. When it fails maintaining its link with a specific core source, it deletes its entry from *CoreSourceAddr* table.

V. *Performance Evaluation*

A. Simulation Environment and Methodology

To study performance of modified protocol DCMP using our new algorithm SCPL and also performance of ODMRP protocol, we implement DCMP and DCMP-SCPL in GlomoSim simulator [7] developed at the University of California, Los Angeles using PARSEC [8]. Our simulation models a network of 50 mobile nodes placed randomly with in 1000 m x 1000m area. Radio propagation range of each node was 250 meters and channel capacity was 2 Mbits/Sec. Nodes move according to the "Random-Way-Point" model [9] which is characterized by a pause time taken as 10 seconds. The IEEE 802.11 Distributed Coordination Function (DCF) [10] is used as the MAC protocol. The Free-Space propagation model [11] is used at the radio layer. In the radio model, we assumed that the radio type was Radio-Capture. Constant Bit Rate (CBR) model is used for data flow and each data packet size is taken as 512 bytes. Each multicast source sends data packets at rate of 2 Pkts/Sec and floods *JoinReq* control packets at intervals of 3 seconds. The multicast sources are selected randomly from all 50 nodes and most of them act as receivers at the same time. Receivers, which are selected also randomly, join one multicast group at the beginning of simulation and never leave the group during the simulation time. Each group size is 20 members. We have used four multicast groups with random selection of sources and receivers. Multiple runs (for each multicast group) with different random seed numbers are conducted for each scenario. Each run executes for 300 seconds of simulation time. Collected data is averaged over those runs and over four multicast groups. We use the same parameter for every protocol unless otherwise specified. We have used the following metrics in comparing protocols performance. These metrics were suggested by the IETF working group for routing/multicasting protocol evaluation [12].

- **Control Overhead Ratio**: it is the ratio of control packets transmitted to data packets that actually delivered to the destinations.
- **Data Delivery Ratio**: it is the ratio of data packets count that actually delivered to the destinations and data packets count from individual transmission of data by each node over the whole network. These data packets may be transmitted from a source node or via intermediate nodes.
- **Total Packets Transmitted per Data Packets Delivered**: it measures channel access efficiency. It is essentials because most link layer protocols of mobile ad hoc wireless networks are typically contention-based.

B. Simulation Results

B.1. Mobility Speeds

Scenario:

Each node moved constantly with the predefined speed. Also each node selects its direction in random way and when nodes reached the simulation terrain boundary, they bounced back and continued to move. Speed of each node is varied from 0 m/s to 20 m/s. Multicast members are 20 nodes. Five nodes from twenty act as sources and receivers at the same time and remain 15 nodes act only as receivers.

Results and Analysis:

Figure 2 shows control overhead ratio under different mobility speeds. ODMRP shows that the previous ratio goes up as mobility increases. It is a predictable behavior due to periodic transmission of control packets (e.g. *JoinReqs* and *JoinReplys*) and broadcast nature of ODMRP. At no mobility speeds, DCMP shows that control overhead become smaller than in ODMRP by (**23%**). In DCMP, passive sources are not permitted to flood *JoinReqs* which reduce control overhead. DCMP-SCPL increases control overhead slightly compared to that in DCMP. But, it is still less than ODMRP. It is a predictable situation because each passive source is permitted to associate with more than one core source which leads to more sending of *PassReq* control packets and more receiving of *Confirm* packets.

As mobility speed increases (especially beyond 5 m/s), control overhead increases sharply in DCMP. It reaches nearly to ODMRP's control overhead. At mobility speed 20 m/s, we can observe that DCMP lost its function due to high mobility speeds. There are two reasons that cause degradation of DCMP function: (1) ease of breakage of unicast link between any passive and its core source, (2) increasing of distance between them to be more than *MaxHop* parameter due to nodes' movement with high speeds. The previous reasons cause DCMP to operate nearly as ODMRP. Behavior of DCMP-SCPL is different from DCMP. In DCMP-SCPL, control overhead increases slightly against change of mobility speed. At high speeds (20 m/s), DCMP-SCPL reduces control overhead by **(21%)** from ODMRP and by **(18%)** from DCMP. DCMP-SCPL is characterized by its ability to operate effectively under high speeds as shown in Figure 2. The reason beyond effective operation of DCMP-SCPL is its passive sources' ability to construct and maintain multi-unicast links with more than one core source. When it loses its link with any core source, it can choose other one from its *CoreSourceAddr* table to be its new proxy which forwards data packets.

The variation of data delivery ratio as a function of mobility is shown in Figure 3. ODMRP's data delivery ratio decreases slightly with increase in mobility speeds. But, it achieves high data delivery ratio compared to DCMP and DCMP-SCPL. ODMRP, including its broadcasting nature, allows to large number of intermediate nodes to be forwarding nodes which help in delivering data packets. DCMP-SCPL's results are closed to ODMRP's results whether in low or high speeds because it saves link between passive-core sources. At low speeds, DCMP's data delivery ratio is reduced compared to ODMRP and DCMP-SCPL due to the fact that unicast passive-core link reduces number of forwarding nodes. As mobility increases, DCMP's data delivery ratio increases until be closed to ODMRP at high speeds. When DCMP fails to do its functions, it acts nearly as ODMRP.

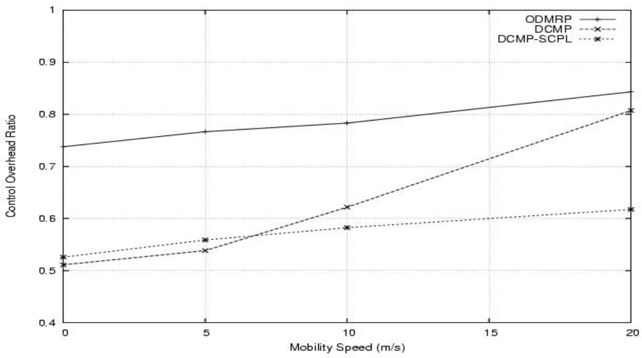

Figure 2. Control Overhead Ratio as a function of mobility speed

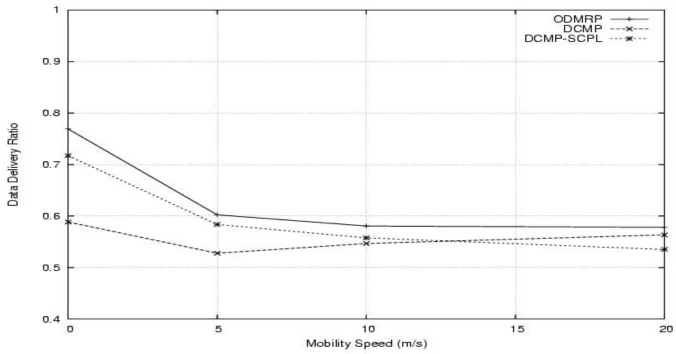

Figure 3. Data Delivery Ratio as a function of mobility speed

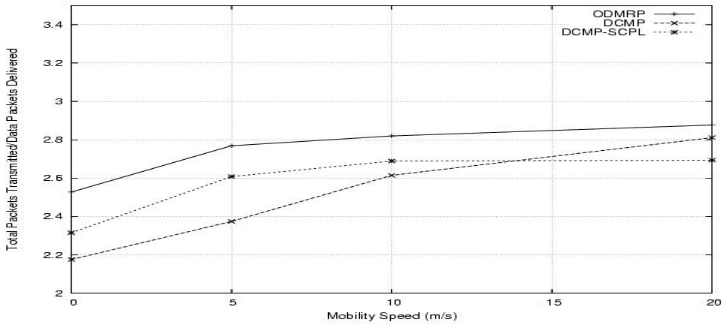

Figure 4. Total packets transmitted per data packets delivered as a function of mobility speed

Figure 4 shows the average of total packets transmitted per data packets delivered. Since most ad hoc network medium access control packets are contention based, having less packets transmitted per data packets delivered is very important. At low speed, DCMP has an interesting result because each passive source limits its control packets transmission. This is different from ODMRP and DCMP-SCPL. ODMRP's result stems from the fact that it transmits more control packets. At high speeds, DCMP-SCPL becomes more effective than DCMP. This result stems from stable unicast link between passive and any core from its core sources. Negative effect of high speeds on that link causes DCMP to operate nearly as ODMRP.

B.2. Number of Senders

Scenario:

In this experiment, we varied number of source ranges as in the set {5, 10, 15, and 20} to investigate the scalability issue. Member group size is limited to 20 nodes. All nodes move with high speed at 20 m/s. We kept traffic load at 2 Pkts/Sec.

Results and Analysis:

Control overhead results are shown in Figure 5. Each protocol shows that its control overhead is increases as number of senders is increases. At small number of senders, results of ODMRP and DCMP are nearly closed. The previous results show how high speeds have a negative effect on function of DCMP. At large number of senders, DCMP reduces control overhead. When number of senders increases, they become near to each others and then each source can become passive source. DCMP's control overhead still high compared to DCMP-SCPL whether at small or large number of senders because it still suffers from ease unicast link breakage between each passive-core source pairs due to high speeds.

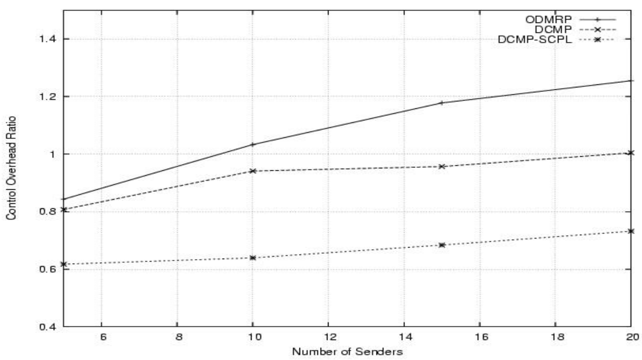

Figure 5. Control Overhead Ratio as a function of number of senders

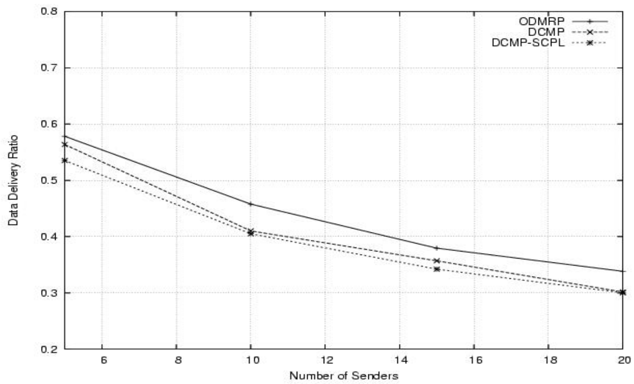

Figure 6. Data Delivery Ratio as a function of number of senders

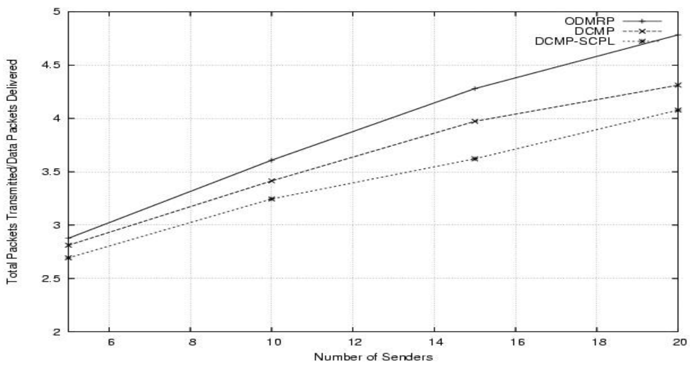

Figure 7. Total packets transmitted per data packets delivered as a function of number of senders

DCMP achieves high data ratio which is closed to ODMRP as shown in Figure 6. Comparing the protocols in Figure 5 and 6, we find that DCMP-SCPL gives interesting results because of reducing control overhead and achieving considerable data delivery ratio which is closed to ODMRP. When results of Figure 7 are taken also in consideration of three protocols' comparison, we can sum up that DCMP-SCPL has an efficient and scalable behavior more than ODMRP and DCMP at high speeds and also at large number of senders.

B.3. Multicast Group Size

Scenario:

We varied the number of multicast members to investigate also the scalability behavior of each protocol. We fixed number of senders at 5 and mobility speeds at 20 m/s. Also, traffic load was kept at 2 Pkts/Sec. Multicast group size was varied from 10 to 40 members as in the following set {10, 20, 30, 40}.

Results and Analysis:

Data delivery ratio is illustrated at Figure 9. When number of group members increases, the number of forwarding group nodes increases accordingly. Also, group members can be in locations near to sources which reduce hop distance between sources and receivers. Small hop distance and large forwarding group help in high and fast delivery of data packets. The previous description explains increasing in data delivery ratio as a function of group members. DCMP-SCPL improves its result of data delivery ratio to be near and closed to ODMRP 's results as group members increased which come up on low cost of control overhead as shown in Figure 8.

.

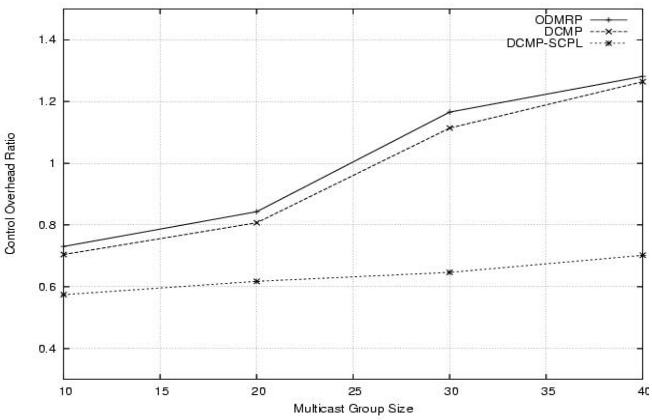

Figure 8. Control Overhead Ratio as a function of multicast group size

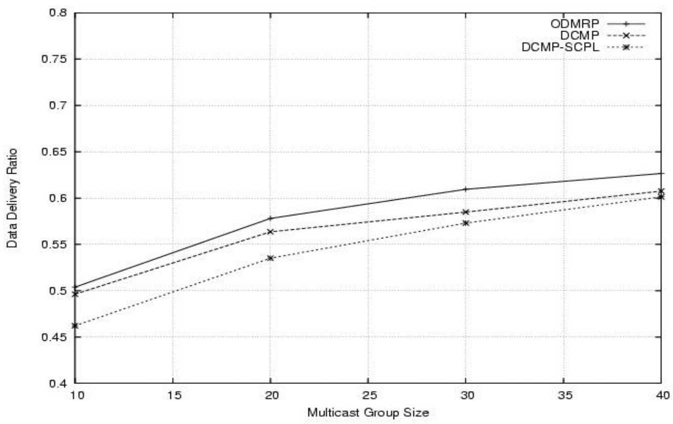

Figure 9. Data Delivery Ratio as a function of multicast group size

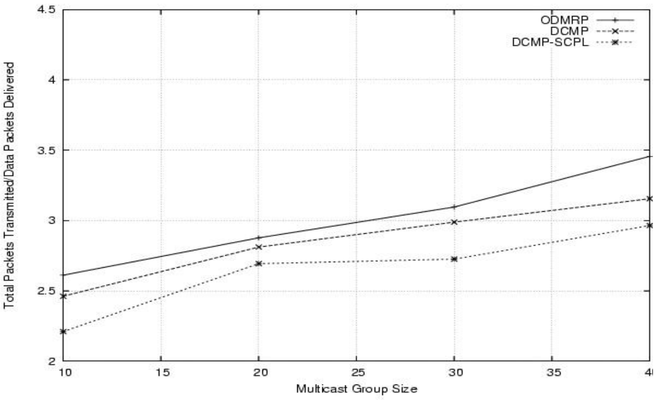

Figure 10. Total packets transmitted per data packets delivered as a function of number of senders

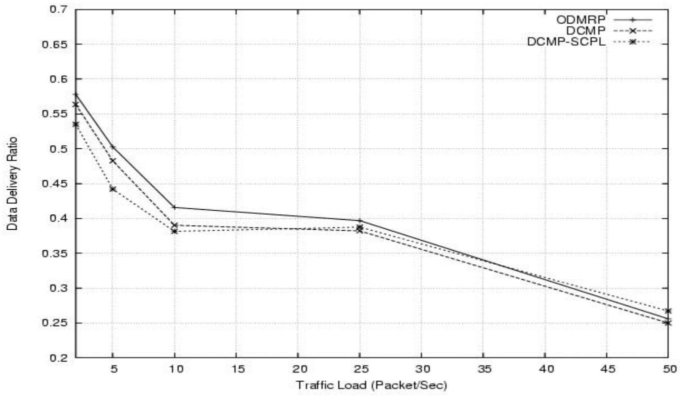

Figure 11. Data Delivery Ratio as a function of network traffic load

In Figure 10, ratio of total packets transmitted to data packets delivered confirms the following two results: the first is that DCMP confirms the fact that it loses its operation and work nearly as ODMRP. The second result indicates that DCMP-SCPL confirms its scalability behavior

B.4. Network Traffic Load

Scenario:

To study the impact of data traffic load on protocols that simulated in this paper, we varied the load on the network between 2 Pkts/Sec and 50 Pkts/Sec. number of senders are kept at 5 and the multicast group size was 20 nodes. In this experiment, nodes moved with mobility speed kept at 20 m/s.

Results and Analysis:

Data delivery ratio for various traffic loads are shown in Figure 11. Number of data packets that transmitted at each second (traffic load) affect on packets drop because it increases number of control packets that are required and help in delivering data packets. Using high speeds beside heavy traffic loads increases possible occurrence of buffer overflow, collisions, and congestion. The previous problem cause data delivery ratio drops rapidly for each protocol as shown in Figure 11. We can observe that results for three protocols are nearly closed whether at light or heavy loads. At light loads, rich connectivity of ODMRP overcomes the negative effect of high speeds and cause ODMRP delivers more data packets. High data delivery ratio of DCMP stems from its failure to do its function at high speeds as explained previously in section V.B.1. Also, DCMP-SCPL delivers data packets robustly, but its data delivery ratio is less than ODMRP and DCMP by a small percentage **(3.6%)** because its connectivity cannot reach to ODMRP's connectivity. The reason behind data delivery ratio of DCMP-SCPL is its ability to reduce control overhead at high speeds.

Reduced control overhead of DCMP-SCPL comes up because it keeps link between any passive and its core source for a long time.

At heavy loads, results of ODMRP and DCMP become nearly closed especially at 50 Pkts/Sec. But, DCMP-SCPL achieves a slight increasing in data delivery ratio over ODMRP. Heavy loads in ODMRP causes producing extremely large control overhead in which rich connectivity of ODMRP cannot reduce it. Large number of control packets reduces number of data packets that can be delivered to destinations.

VI. CONCLUSION

In this paper, we modify mesh based and on-demand multicast protocol, DCMP, by adding SCPL algorithm. DCMP-SCPL achieves considerable results at high speeds. It overcame the DCMP's problem of losing its function at high speeds and returning to its default ODMRP operation. DCMP lose its function due to ease link breakage between each passive source and core source pairs. We implements DCMP-SCPL using GlomoSim and the simulation results show that there is **(18%)** reduction in control overhead rather than DCMP at high speeds. Control overhead reduction comes up at the cost of a small reduction in data delivery ratio by **(3.6%)** for light network loads. At heavy loads, DCMP-SCPL improved data delivery ratio slightly than ODMRP and DCMP. To sum up, adding SCPL algorithm to DCMP protocol improved scalability which can be mainly attributed to the reduced control overhead especially at high speeds.

REFERENCES

[1] M. Ilyas, "The hand book of ad hoc wireless networks," *CRC Press LLC.* , 2003.

[2] A. Elsayed, V. Roca, and Laurent Mathy, "A survey of proposals for an alternative group communication service," IEEE Network, pp. 2–7, January 2003.

[3] S. J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, "A performance comparison study of ad hoc wireless multicast protocols," *Proceedings of IEEE INFOCOM*, pp. 565–574, March 2000.

[4] M. A. Ali, A. EL-SAYED, and I. Z. MORSI "A Survey of Multicast Routing Protocols for Ad-Hoc Wireless Networks," Minufiya Journal of Electronic Engineering Research, vol. 17, no. 2, pp. 185-198, July 2007.

[5] S. K. Das, B. S. Manoj, and C. Siva Ram Murthy, "A dynamic core-based multicast routing protocol for ad hoc wireless networks," *Proceeding of ACM MOBIHOC*, pp. 24– 35, June 2002.

[6] M. Gerla, S.J. Lee, and W. Su, "On-demand multicast routing protocol (odmrp) for ad hoc networks," *Internet Draft, <draftietf- manet-odmrp-02.txt>*, 2000.

[7] UCLA Parallel Computing laboratory and wireless Adaptive Mobility Laboratory, " GlomoSim ; A Scalable Simulation Environment for wireless and wired network systems, " http://pcl.cs.ucla. edu / projects/domains/glomosim.html

[8] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, and H. Y. Song, “PARSEC ; A parallel Simulation Environment for Complex Systems,” *IEEE Computer*, vol. 31, no. 10, pp. 77-85, October 1998.
[9] D. B. Jonson, and D. A. Maltz, “Dynamic Source routing in ad hoc wireless networks,” in *Mobile Computing*, eds. T. I mielinske and H. Korth, Chapter 5, pp. 153-181, 1996.
[10] IEEE Computer Society LAN MAN Standards Committee, wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification, IEEE standard 802.11- 1997, The Institute of Electrical and Electronics Engineering, New York, 1997.
[11] T. B. Rappaport, “Wireless communications principles and practice,” Prentice Hall PTR, *NewJersey*, 1996.
[12] IETF Mobile Ad Hoc Networks (MANET) Working Group Charter ,” http://www.ietf.org/html.charters/manetcharter.html.

الملخص العربى :

بروتوكول النقل المتعدد ذو المتحكم الديناميكى و المستخدم لتقنية أقصر وصلة بين كل من core source و passive source فى الشبكات اللاسلكية ذات العقد المتحركة

يطبق بعض بروتوكولات النقل المتعدد مفهوم بناء شكل الوصلات العشوائى (Mesh Topology) عند الطلب (On-Demand Concept) مثل بروتوكول ODMRP الذى يقوم بالإرسال المتكرر و المنتظم لرسائل التحكم و ذلك لبناء و الحفاظ على الشكل العشوائى للوصلات. يعانى بروتوكول ODMRP من عدد رسائل التحكم الضخم و خاصة عندما يزيد عدد أعضاء الشبكة أو عدد المرسلين و يؤدى هذا إلى ظهور مشكلة Scalability. ولحل هذه المشكلة تم إصدار بروتوكول DCMP و الذى يهدف إلى تقليل عدد رسائل التحكم فى بروتوكول ODMRP و ذلك بمنع مرسل ما (يسمى Passive source) من إرسال رسائل التحكم بصورة منتظمة و متكررة و السماح لهذا المرسل Passive بإرسال حزم بياناته إلى المستقبلين فقط من خلال الوصلات العشوائية التى قام ببنائها مرسل آخر(يسمى Core source). يسمى المفهوم السابق بتقنية التحكم المركزى (Coring). يسمح DCMP لكل مرسل passive ببناء وصلة واحدة فقط (Unicast Link) مع مرسل core واحد فقط. تعانى الوصلة الأحادية بين كل من مرسل passive و مرسل core الخاص به من سهولة الكسر نتيجة لحركة أعضاء الشبكات اللاسلكية ذات العقد المتحركة بحرية و أحيانا بسرعات عالية. الكسر السهل لهذا النوع من الوصلات الأحادية يفقد DCMP وظيفته و يعيده مرة أخرى للعمل بنفس وظيفة ODMRP مما يعيد مشكلة Scalability للظهور. و كانت مشكلة DCMP السابقة هى الدافع الرئيسى لتقديم هذا البحث فقمنا بتطويره و ذلك بإضافة تقنية "أقصر وصلة بين كل مرسل passive و مرسل core داخل بروتوكول DCMP (Shortest Core-Passive Link (SCPL) algorithm)". يسمح SCPL لكل مرسل passive ببناء و الحفاظ على أكثر من وصلة مع أكثر من مرسل core ثم يتم إختيار مرسل core الذى يبعد أقصر مسافة عن هذا المرسل passive ثم يرسل له حزم بياناته و اذا حدث كسر لهذه الوصلة يتم إختيار core آخر من الجدول الذى يقوم ببناؤه هذا المرسل passive و يحتفظ فيه بالمعلومات الخاصة بـ core sources الخاصة بهذا المرسل passive. إستطاع SCPL حل مشكلة Scalability و ذلك بتقليل رسائل التحكم خاصة فى السرعات العالية مع الإحتفاظ بنسبة توصيل لحزم البيانات تقارب مثيلتها فى بروتوكول ODMRP.