

Enhanced Route Re-Construction Method for Associativity Based Routing Protocol for Mobile Ad hoc NETWORKS (MANET)

¹Fawaz A. M. Masoud, ²Said Abu Shaar, ³Ayman Murad and ⁴Ghassan Kanaan

¹Computer Information Systems Department, The University of Jordan
King Abdullah II School for Information Technology, Jordan

²Amman University, Jordan

³Arab Academy for finance & Banking Sciences, Jordan

⁴Amman Al-Ahliyya University, Jordan

Abstract: A mobile Ad-hoc NETWORK (MANET) is wireless network composed of mobile nodes that are dynamically and randomly located in such a manner that the interconnections between nodes are capable of changing on a continual basis. In order to facilitate communication within the network, a routing protocol is used to discover routes between nodes. The primary goal of such an ad-hoc network routing protocol is correct and efficient route establishment between a pair of nodes so that messages may be delivered in a timely manner. Route construction and maintenance should be done with a minimum of overhead and bandwidth consumption. The ABR is a source-initiated protocol and is working on the assumption of stable route from the source to the destination node. Maintenance for the route when the destination node moves will be performed in backtracking scheme starting from the immediate upstream node from the destination. If this process results in backtracking more than halfway to the source, it will discontinue and a new route request will be initiated from the source. In the case if the Source Node moves, then the Source Node will invoke a route reconstruction because the ABR is source-initiated protocol. This study presents an enhanced method for the route reconstruction in case the source, the intermediate, or the destination node changes its location by giving more active role to the moving node in maintaining the established route.

Key words: ABR, EABR, ad-hoc mobile network, associativity

INTRODUCTION

Currently, there are two main kinds of wireless networks^[1]. The first is known as the infrastructure network with fixed and wired gateways. The bridges for these networks are known as the base stations. Mobile units will connect with the nearest base station within their communication radius. As the mobile unit travels out of the range of one base station to another, it will continue communicating seamlessly.

The second type is the infrastructure less network without fixed routers, known as a Mobile Adhoc NETWORK (MANET). All nodes in these networks are capable of moving and can be connected dynamically in an arbitrary manner. Nodes of these networks will function as routers, which discover and maintain routes to other nodes.

Applications such as conferences, meetings, lectures, crowd control, search and rescue, disaster recovery and automated battlefields typically do not have central administration or infrastructure available. In these situations, ad hoc networks, or packet radio networks consisting of hosts equipped with portable radios must be deployed spontaneously without any wired base stations.

Routes in MANET are multihop because of the limited propagation range of wireless radios. Since nodes in the network move freely and randomly, routes often get disconnected. Routing protocols are thus responsible for maintaining and reconstructing the routes in a timely manner as well as establishing the durable routes. In addition, routing protocols are required to perform all the above tasks without generating excessive control message overhead. Control packets must be utilized efficiently to deliver data packets and be generated only when necessary. Reducing the control overhead can make the routing protocol efficient in bandwidth and energy consumption.

Moreover, wireless channel bandwidth is limited. The scarce bandwidth decreases even further due to the effects of multiple access, signal interference and channel fading. Network hosts of ad hoc networks operate on constrained battery power, which will eventually be exhausted. Ad hoc networks are also more prone to security threats. All these limitations and constraints make multihop network research more challenging.

MANET has several outstanding characteristics^[2] such as; Dynamic topologies, bandwidth-constrained,

variable capacity links, Energy-constrained operation and Limited physical security.

The existing routing protocols on the MANET can be classified into two categories^[1]:

Table driven (proactive protocols): These protocols need to maintain consistent, up-to-date routing information tables from every node in the network. When the topology changes, the nodes propagate the messages throughout the network in order to keep the consistency of the routing information. Algorithms like DSDV^[3] (Destination Sequenced Distance Vector Routing Protocol), Cluster-head Gateway Switch Routing (CGSR)^[4] and Wireless Routing Protocol (WRP)^[5] are all considered as examples for this categories.

Source-initiated on-demand (reactive protocols): These protocols create routes only when desired by the source node. When a node requires a route to a destination, it initiates a route request process. Once the route has been established, it is maintained by the route maintenance procedure unless the route is inaccessible. Algorithms like Ad-hoc On-Demand Distance Vector (AODV) routing protocol^[6], Dynamic Source Routing (DSR)^[7], Temporally-Ordered Routing Algorithm (TORA)^[8] and the Associativity Based Routing Protocol (ABR)^[9-11] are all considered as examples for this category.

Motivation for the enhancements: Definitely, there are two scenarios for the Source or the destination nodes to settle after movements:

- * Close to one or more Intermediate Node(s) who are participating in the established route;
- * Faraway from the Intermediate Node(s), which are participating in the established route.
- * The ABR Behavior When the Source Moves

If the Source Node moves, then a route reconstruction will be invoked. If the Destination Node moves, the route maintenance will be performed in backtracking scheme starting from the immediate upstream node from the destination, as follows: the immediate upstream node erases its route and determines if the Destination Node is still reachable by a Localized Query (LQ) process. If the Destination Node receives the LQ packet, it replies with the best partial route; otherwise, the initiating node times out and the process backtracks to the next upstream node. If this process results in backtracking more than halfway to the source, the LQ process is discontinued and a new Broadcast Query (BQ) process is initiated at the source node.

In the second scenario, eventually a route re-construction will be invoked by the source after the overhead of route re-construction is done.

The EABR protocol description: The Enhanced Associativity-Based Routing protocol consists of two phases, namely:

- * Route Discovery Phase
- * Route Re-Construction (RRC) Phase

Initially, when a source node requests a route, the route discovery phase is invoked. When the link of established route changes due to SRC, DEST, INs or subnet-bridging MH's migration, the RRC phase is invoked. These two phases will be discussed below.

The EABR route discovery phase: The route discovery phase allows approximation of the data throughput associated with the selected route to be computed. This is achieved through the knowledge of associativity ticks of neighbors in the route and the relaying load of nodes supporting the route. The route discovery phase consists of a Broadcast Query (BQ) and an await-reply (REPLY) cycle, which is described below.

BQ-reply cycle: Initially, all nodes except those of DEST'S neighbors have no routes to the DEST. When a node requests a route to the DEST, it broadcasts a BQ message, which is propagated throughout the ad-hoc mobile network in search of MHs, which have a route to the DEST. Here, a sequence number is used to uniquely identify each BQ packet and no BQ packet will be broadcasted more than once. Once the BQ query has been broadcasted by the SRC, all INs that receive the query will discard that packet if it has previously processed the packet; otherwise, the IN will check if it is the DEST. If is not the DEST, the IN appends its MH address/identifier at the IN IDs field of the query packet and broadcasts it to its neighbors (if it has any). The associativity ticks with its neighbors will also be appended, along with its relaying load, link propagation delay and the hop count.

The next succeeding IN will then erase its upstream node's neighbors' associativity ticks entries and keeps only those concerned with itself and its upstream node. In addition, because of the association ticks symmetry between nodes, the associativity ticks received from the upstream node can be checked for validity. In this manner, the query packet reaching the DEST will only contain the intermediate MHs' addresses (hence recording the path taken) and their associativity ticks (hence recording the stability state of the INs supporting the route) and the route-relaying load, together with information on route propagation delays and hop count. The process is illustrated in Fig. 1. The resulting BQ packet is variable in length and its format is shown below in Fig. 2. The resulting BQ packet is variable in length and its format is shown below in Fig. 2.

The DEST will at a suitable time; after receiving the first BQ Packet, know all the possible routes and their qualities. If a route consists of MHs having

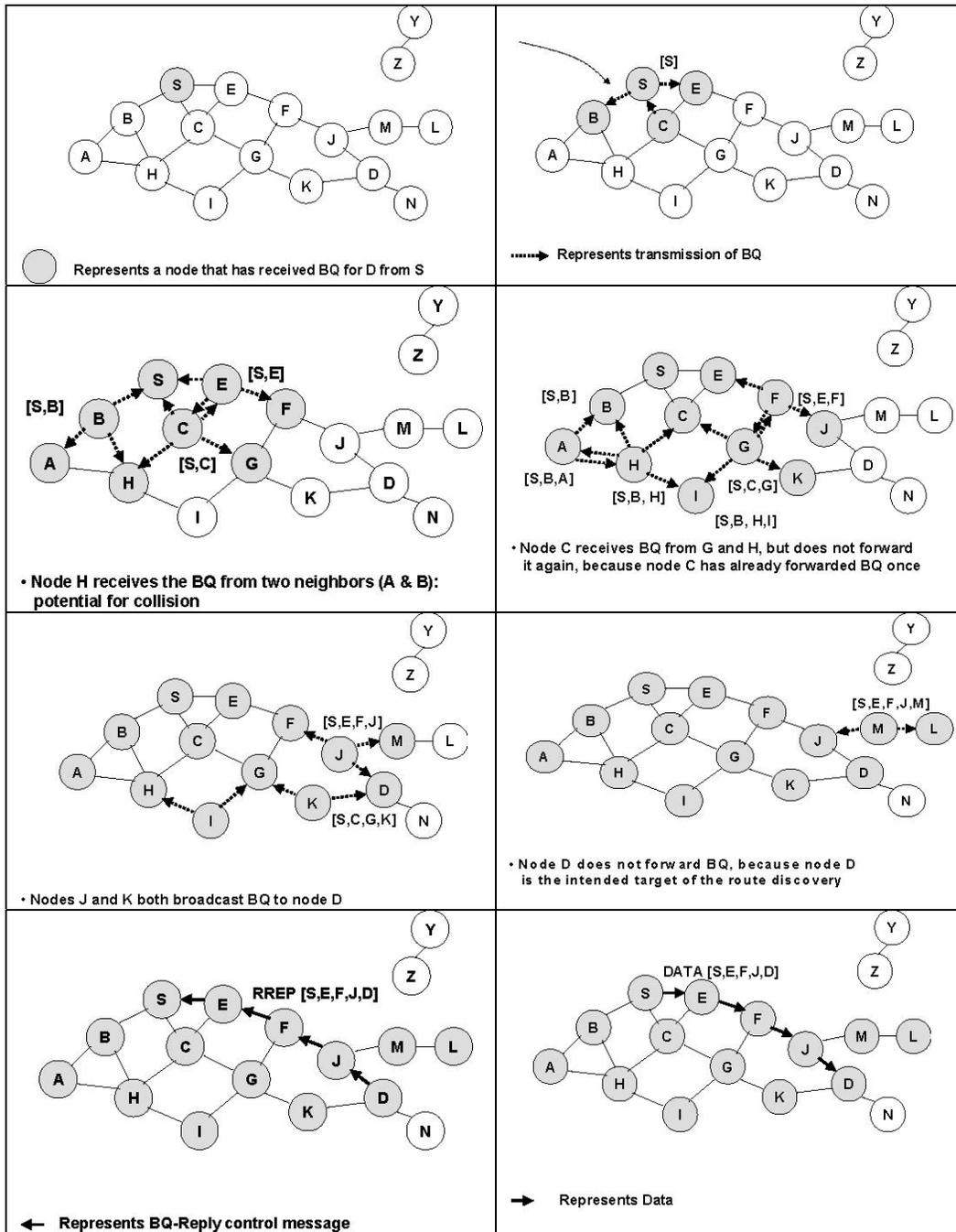


Fig. 1: Finding the route from source to destination in EABR

TYPE	SRC ID	DEST ID	LIVE	IN IDs	METRICS	SEQ NO.	CRC
------	--------	---------	------	--------	---------	---------	-----

Fig. 2: BQ control packet in the EABR

TYPE	SRC ID	DEST ID	INs IDs	SERIAL NO.	SEQ ID	ROUTE QUALITIES	CRC
------	--------	---------	---------	------------	--------	-----------------	-----

Fig. 3: Reply control packet in the EABR

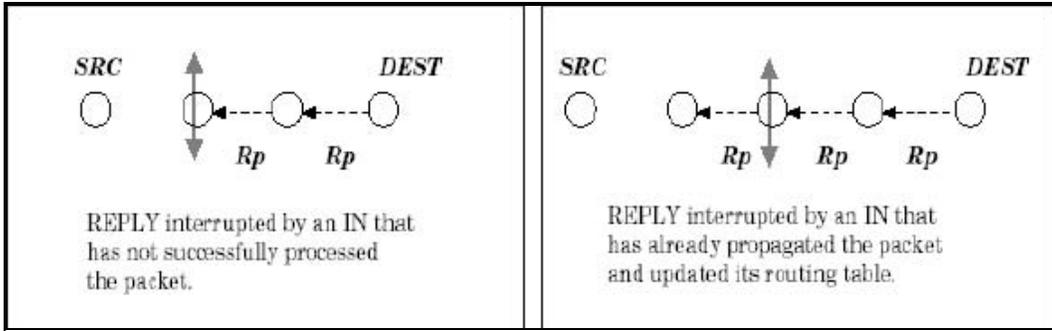


Fig. 4: Reply interruption caused by unexpected INs' movements

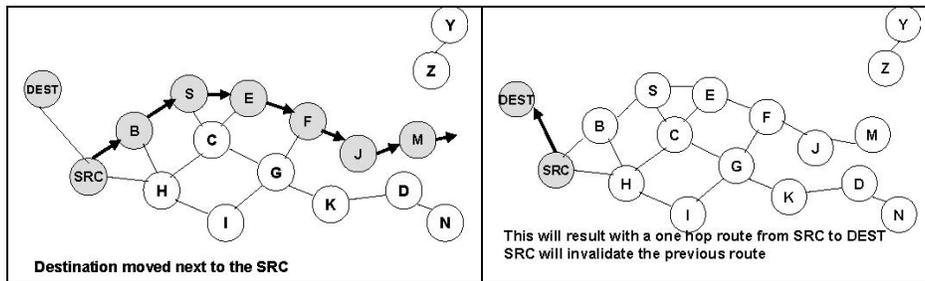


Fig. 5: Route maintenance when DEST moves to SRC coverage in the EABR

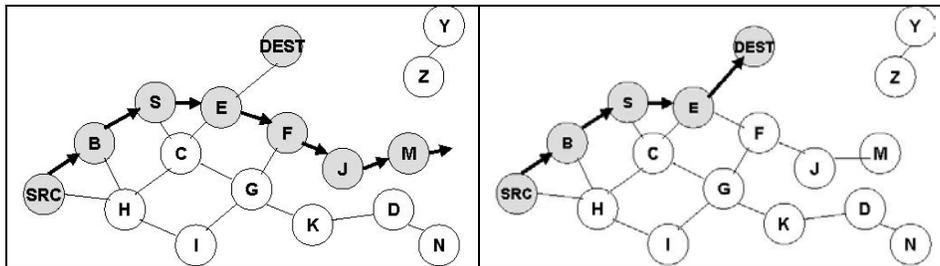


Fig. 6: Route maintenance when DEST moves to IN coverage in the EABR

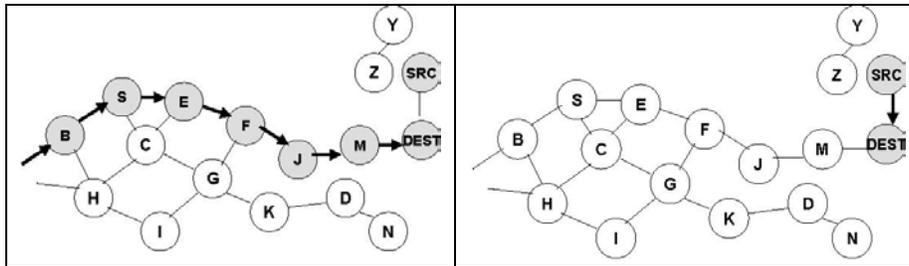


Fig. 7: Route maintenance when SRC moves to DEST coverage in the EABR

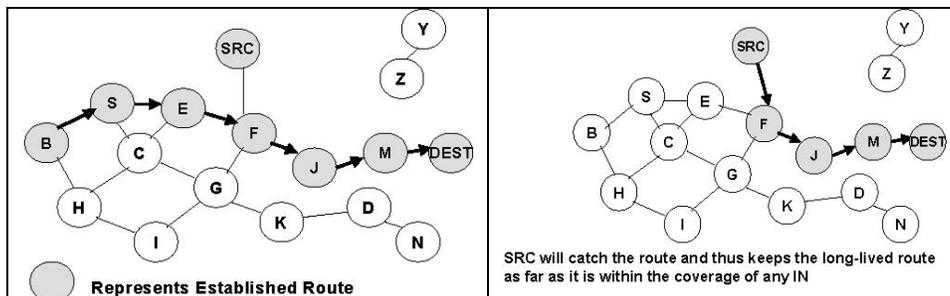


Fig. 8: Route maintenance when SRC moves to IN coverage in the EABR

associativity ticks (therefore indicating spatial and connection stability), then that route will be chosen by the DEST despite other shorter hop routes. However, if the overall degrees of association stability of two or more routes are the same, then the route with the minimum hops will be chosen. If multiple routes have the same minimum-hop count, then one of the routes is arbitrarily selected. The DEST then selects the best route (based on the selection criteria mentioned earlier) and send a REPLY packet back to the SRC, via the route selected. This causes the INs in the route to mark their routes to DEST as valid and this means that all other possible routes will not relay packets destined for from arriving at the DEST.

While the BQ query packet propagates to the DEST, each node relaying the BQ packet will know its hop count from the SRC. Similarly, when the REPLY packet propagates back to the SRC, the INs can also compute their distances to the DEST. Initially the DEST will set the SERIAL NO field to Zero, then each IN relaying the REPLY packet will compute its serial number in the route by incrementing the SERIAL NO field by one. The REPLY packet is variable in length and has the format shown in Fig. 3.

Case when the source never receives reply: There may be some odd instances when the SRC never receives DEST's REPLY because of some unexpected INs' movement. In such situation, the SRC will eventually BQ_TIMEOUT and sends another BQ query. Since the downstream neighbors of the migrating IN realizes the associativity change, it will send an RN [STEP=1] (Route Notification) packet in the downstream direction, deleting all the downstream nodes' invalid routing table entries. Another situation occurs when a selected IN moves while the REPLY propagation is still in progress. The upstream neighbors of the migrating node will perform an LQ[H] (localized query) process to discover a new partial route, while the downstream neighbors sends an RN[1] packet towards the DEST, thereby erasing all invalid downstream nodes' routing entries. Hence, while the RRC is in progress, the REPLY packet continues to propagate towards the SRC. This is illustrated in Fig. 4.

The EABR route re-construction phase: The route reconstruction RRC phase will be invoked in three situations; specifically when the SRC, DEST or the IN moves. The EABR will try to locate an alternate valid route without the need to broadcast a query unless it was the last resort.

There are two scenarios for the SRC, DEST or the INs to settle after movements:

- * Within the radius of one or more INs who are participating in the established route,
- * Faraway from the INs who is participating in the established route.

Route re-construction when the destination moves:

As all INs are aware of the route in question from their own routing table, the DEST, after settling, will broadcast an LQ packet [Here I am] searching for one of its immediate upstream nodes. If any INs listens to this message, then each listening IN will reply to the DEST by selecting the best partial route based on association stability criteria. Then the DEST will select the best route based on the criteria described above. The DEST then sends a REPLY packet back to the source along this path. Nodes propagating the REPLY mark their routes as valid. All other routes remain inactive and the possibility of duplicate packets arriving at the destination is avoided. This will generate either the same previous path or a shortcut path, which is better than the maintained one, which will introduce optimization while re-construction.

The route reconstruction phase is accomplished by a LQ and await-reply (BQ-REPLY) cycle. The destination node will broadcast an LQ message in search of intermediate mobiles that are participating in the current route to the source. The following scenarios are anticipated:

If the SRC receives the LQ, then the DEST is within the Source' radio coverage range, packet duplicates will result at the DEST since the DEST now receives packets from the SRC directly and also from the original SRC-DEST rout. Hence, to avoid cell duplicates and non-optimal routes, the SRC, on discovering that the DEST is within range and is in stable state, will send an RN[1] packet downstream to erase existing route and will re-establish a new single hop route with the DEST, as described in Fig. 5.

Some INs will receive the LQ, consequently the IN will reply to the DEST by selecting the best partial route based on association stability criteria. Each IN will update its routing table to reflect a shortcut to the Destination, as described in Fig. 6.

If no intermediate node replies to the Destination, then a route discovery is initiated from the DEST.

Route re-construction when the source moves:

As all INs are aware of the route in question from their own routing table, the SRC, after settling, will broadcast an LQ packet Here I am searching for one of its immediate downstream nodes. If any IN listens to this message, then each listening intermediate node will reply to the Source Node by selecting the best partial route based on association stability criteria. Then the SRC will select the best route based on the above-described criteria and this will generate either the same previous path or a shortcut path, which is better than the maintained one. Which will do optimization while re-construction.

The route reconstruction phase is accomplished by an LQ and await-reply (BQ-REPLY) cycle. The SRC will broadcast an LQ message in search of INs that are participating in the current route to the Destination

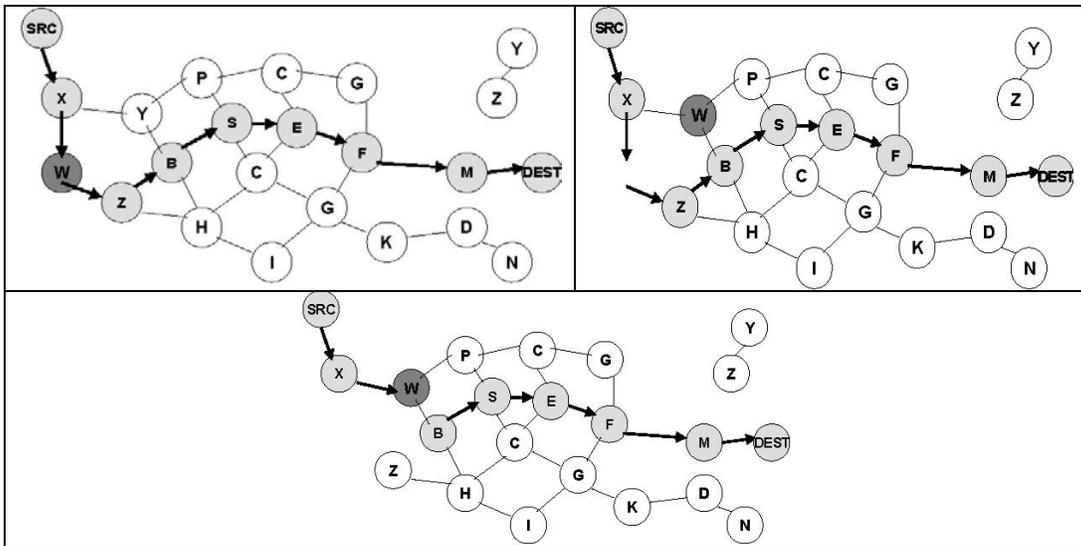


Fig. 9: Route maintenance when IN moves within the IN coverage the EABR

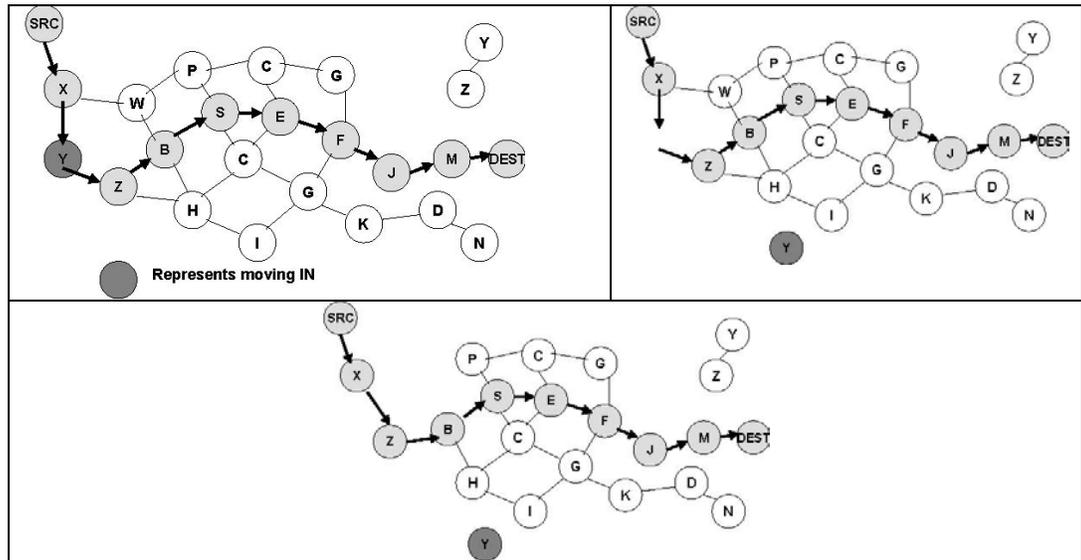


Fig. 10: Route maintenance when IN moves in lower arm the EABR

Node. The following scenarios are anticipated:

If the DEST receives the LQ, then the destination is within the SRC radio coverage range, packet duplicates will result at the DEST since the DEST now receives packets from the SRC directly and also from the original SRC-DEST route. Hence, to avoid cell duplicates and non-optimal routes, the SRC, on discovering that the DEST is within range and is in stable state, will send an RN[1] packet downstream to erase existing route and will re-establish a new single hop route with the DEST, as shown in Fig. 7.

Some Intermediate Nodes will receive the LQ; consequently, the IN will reply to the SRC by selecting the best partial route based on association stability above-described criteria. Each IN will update its routing table to reflect a shortcut to the Source, as shown in Fig. 8.

If no intermediate node replies to the Source Node, then a route discovery is initiated from the Source.

Route re-construction when the intermediate node moves: The IN, after settling, will broadcast an LQ packet Here I am searching for one of its immediate downstream and upstream nodes. The LQ packet will include the SRC ID & the DEST ID and its Serial No in the route under reconstruction. The listening INs will check first if they are participating in the route in question or not. If not then the packet will be discarded. If yes then the IN will send a reply packet, which contains its ID and its SN. After receiving all the REPLYs, the broadcasting IN will sort in ascending order the received Serial NOs and store them in a list. Based on the biggest and smallest SNs of the list, the following cases may appear:

Case one: The broadcasting IN' SN is greater than the smallest and less than the biggest SN of the list. In this case, the IN has successfully found a shortcut path, as shown in Fig. 9.

The RT entries for the three INs will be updated as follows:

The RT of the broadcasting IN:

1. The Incoming IN = the ID of the IN who has the maximum SN in the list.
2. The Outgoing IN = the ID of the IN who has the minimum SN in the list.
3. The SN = the SN of IN who has the minimum SN in the list + 1.

The RT of the IN who has the minimum SN in the list:

1. The Incoming IN = the ID of the broadcasting IN.
2. The Outgoing IN = Outgoing IN. (No Change)
3. The SN = the SN. (No Change)

The RT of the IN who has the maximum SN in the list:

1. The Incoming IN = Incoming IN. (No Change)
2. The Outgoing IN = the ID of the broadcasting IN
3. The SN = the new SN of the broadcasting IN + 1.

The IN with maximum and minimum SN will invalidate the unused parts of the route, using the RN [1] packets as follows:

1. The IN with the minimum SN will invalidate the entire upstream partial route from its first upstream IN until the IN with SN = the SN of the broadcasting IN-1.
2. The IN with the maximum SN will invalidate all the downstream partial route from its first downstream IN until the IN with SN = the SN of the broadcasting IN + 1

Case two: The broadcasting IN' SN is greater than any SN within the list. In this case the route from the IN' downstream node to the DEST is still valid. In addition, the moving IN is closer to the SRC. The IN's immediate downstream node (i.e. the pivoting node) removes its incoming node entry and its immediate upstream neighbor propagates an RN[1] packet towards the SRC, thereby deleting all the subsequent upstream nodes' invalid routing entries. A new partial route to the SRC needs to be found.

An LQ[H] process is then invoked by the pivoting node to locate alternate partial routes. The SRC may receive multiple LQs, hence it needs to select the best partial route and return a REPLY to the pivoting node. This causes all INs between the SRC and the pivoting node to update their RTs. On receiving the REPLY, the pivoting node updates its RT entries and appends the next hop (incoming) node ID into the data packet. This ensures that only one partial route is selected, as shown in Fig. 10.

Case three: The broadcasting IN' SN is smaller than any SN within the list. In this case the route from the IN' upstream node to the SRC is still valid. Moreover, the moving IN is closer to the DEST. The IN's immediate upstream node (i.e. the pivoting node) removes its outgoing node entry and its immediate downstream neighbor propagates an RN[1] packet towards the DEST, thereby deleting all the subsequent downstream nodes' invalid routing entries. A new partial route to the DEST needs to be found.

A LQ[H] process is then invoked by the pivoting node to locate alternate partial routes. The DEST may receive multiple LQs, hence it needs to select the best partial route and return a REPLY to the pivoting node. This causes all INs between DEST and the pivoting node to update their RTs. On receiving the REPLY, the pivoting node updates its RT entries and appends the next hop (outgoing) node ID into the data packet. This ensures that only one partial route is selected.

If no partial route exists, the LQ-TIMEOUT will expire and an RN [0] packet will be sent by the pivoting node to the next upstream node and the cycle repeats until the next pivoting node has a hop count greater than half hop or when a new partial route to the DEST is found, as shown in Fig. 11.

Concurrent nodes movements: Competition situations exist due to multiple invocations of RRC processes as a result of concurrent movements by SRC, DEST and INs. The following put in plain words why the EABR protocol is challenging with 'multiple-RRCs' conflicts and how one RRC is valid eventually.

Destination moves RRC interrupted by upstream IN's moves: When the DEST moves and while the RRC is in progress, any upstream INs moves will cause their respective downstream neighbors' route to be deleted. The new pivoting node nearest to the SRC will perform the RRC and all other RRCs will be passive when they hear the newer LQ broadcast for the same route. Hence, only one RRC is valid.

Upper-arm IN RRC interrupted by lower arm IN's moves: This is the same as the above-mentioned case. Note that the same argument can be applied to the case when an LQ process has to be aborted and an RN[1] packet has to be sent to the SRC to invoke a BQ, but is hindered due to some upstream INs' movements. Then a new pivoting node nearest to the SRC will destroy the earlier RRC processes by invoking a new LQ.

Lower-arm IN RRC interrupted by upper arm INs' moves: While a lower arm IN RRC is taking place, any movements by any upper arm INs will not result in an LQ[H] or RN[1] process being initiated since the lower arm IN has earlier sent RN[1] packet downstream to erase invalid routes. If the RN [1] packet does not succeed in propagation towards the DEST, the LQ[H]

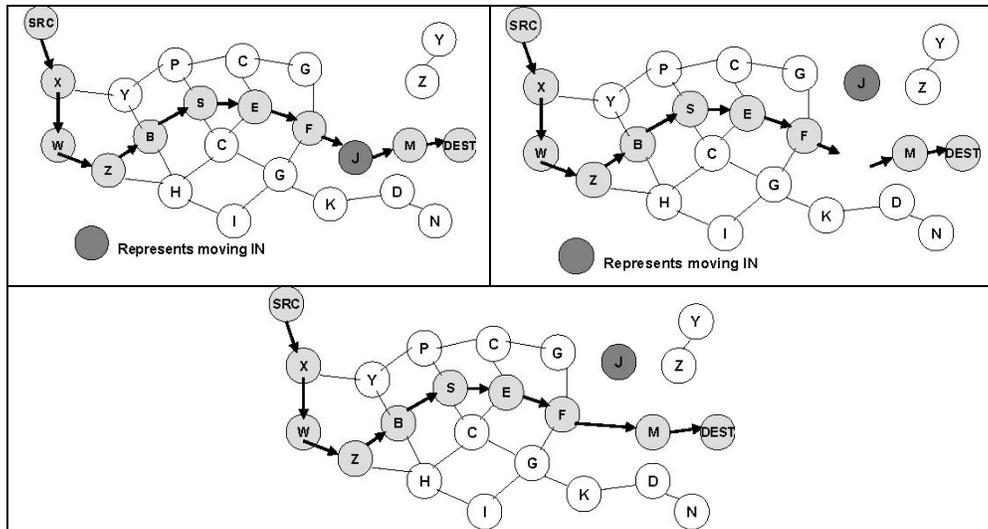


Fig. 11: Route maintenance when IN moves in upper arm the EABR

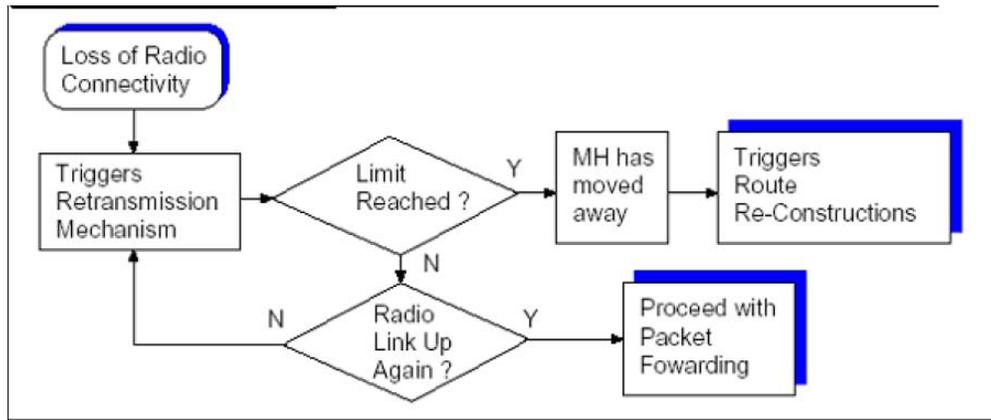


Fig. 12: Handling packet retransmission in EABR

Table 1: The EABR routing table example

Destination	Source	Incoming IN	Outgoing IN	Serial No	Distance
Na	Nx	Nz	Nj	5	4
Nk	Ny	Ni	No	4	3

Total No. of Active Routes Supported (Relay Load): 2

process initiated by the lower arm IN will also serve to delete these invalid routes.

Lower/Upper-arm IN RRC interrupted by destination's moves: This will have no effect on the RRC, as the LQ[H] process uses a localized query approach by which the DEST will locate one IN of the route. Once the DEST is in its stability state and is reachable from its new/old-pivoting node, the RRC process will be successful.

Lower/upper-arm IN RRC interrupted by source's moves: While lower or upper arm IN RRC is in progress, any moves by SRC will result in an LQ, which uses a localized query approach by which SRC will locate one IN of the route. If LQ fails to reconstruct

the route, then the SRC will broadcast a BQ, which will then destroy all on-going LQ-REPLY-RN processes related tie that route. Consequently, unsuccessful and out of date RRCs will not continue and a new route has to be discovered via the BQ process.

Source and destination nodes moving away from Ins: In this case, RRCs, as a result of DEST and SRC moves, will be initiated. However, the BQ process initiated by the SRC or DEST will again destroy all currently running RRCs.

LQ-REPLY cycle interruption: During an LQ-propagation and REPLY-await process, if any of the upstream nodes (i.e. lower arm INs) break up, an RN [1] packet will be propagated downstream, erasing all

the downstream IN's routes entries. The existing pivoting node will ignore any subsequent REPLY to its LQ. The new pivoting node will resume with a new LQ-REPLY process.

The EABR route erasure and updates: In the EABR, no attempt is made to keep alternate routes, as maintaining those causes overhead. Only one route will be selected and only one route is valid for a particular route request. The avoidance of using alternate routing information means that problems associated with looping due to INs having out of date routes are absent and there is no need for periodic network-wide broadcast and route updates.

Any alternate route will have to be discovered via an LQ or BQ process, which may give rise to better (shorter hop and long-lived) routes. The DEST, on receiving multiple LQs, will select the best route any reply to the SRC. During the LQ_REPLY_RN cycle, invalid INs routes are erased by RN[1] packets and INs forming the new partial route will have their route entries updated when they relay the REPLY packet from the DEST to SRC. If all the possible LQ_REPLY_RN cycles fail, all the upstream/downstream nodes will have their route entries erased via RN [0] and RN[1] packet and the SRC / DEST will then revert back to the BQ_REPLY cycle.

Lastly, for the case of BQ query, any INs receiving a BQ and having invalid routes will result in these being erased, therefore ensuring that no invalid routes exist in the INs.

The EABR packet header: Since a long packet header results in low channel utilization efficiency, in the EABR protocol, each data packet header will only contain the neighboring node routing information, not all the nodes in the route. Each IN will renew the next-hop information contained in the header before propagating the packet upstream or downstream. Thus a hybrid routing scheme, which is a combination of broadcast and point-to-point routing, is used. The structure and field descriptions of the packet headers are shown in Appendix A.

Data flow acknowledgement: When a node receives a packet and performs relaying via a radio transmission to its neighbors, its previous neighbor that has sent it the packet will have heard the transmission and hence this is indirectly used as an acknowledgement to the packet sent. On the other hand, *active acknowledgements* will only be sent by the DEST, as it no longer has a neighbor to relay the packet to. Hence, this provides a data flow acknowledgment mechanism for packet forwarding in an ad-hoc mobile network, which is only presented in the ABR.

Packet retransmission: While the data flow acknowledgement scheme allows forwarded packets to

Table 2: The EABR neighboring table

Neighbors	Associativity Ticks (units)	Link Delays (msecs)
Na	5	100
Nb	15	50

be acknowledged, there are situations where the acknowledgements never reach the intended receiver. This can be a result of radio interference, which causes a sudden loss of radio connectivity. Hence, if a MH has forwarded a packet and does not receive an acknowledgement within a certain time interval, it will retransmit the packet for X times, after which the neighboring MH will be considered as out-of-reach and RRC procedures will be invoked. If however, the radio link returns before the retransmission counter expires, the packet forwarding process continues, as illustrated in Fig. 12.

The EABR routing table (RT): The RT of a node supporting existing routes is shown in Appendix A. Table 1 shows a snapshot from routing table. The table shows that every node supporting on-going routes will map incoming packets from a particular upstream node to the corresponding downstream node. Every node will also keep track of its distance (hop count) to the DEST and a record of the total routes that it is currently supporting, in addition to the SN field, which will store the serial number of the Intermediate Node in the whole path.

The EABR neighboring table (NT): The Neighboring Table (NT) is usually updated by the data-link layer protocol, which will generate, receive and understand beacons from the neighboring MHs and pass this information up to the higher protocol layers. The structure of an NT is shown in Table 2.

The EABR control packets 'seen' tables: While the BQ query process is activated via a radio broadcast, the LQ query process is invoked via a localized broadcast. To avoid MHs from processing and relaying the same BQ or LQ packet twice, BQ and LQ 'seen' tables are needed. If the received control packet type, route identifier and sequence number match an entry in the 'seen' table list, then the packet is discarded. On the other hand, because the REPLY and RN control packets utilize 'directed' broadcast (since intended recipients' addresses are contained in the control packet), 'seen' tables for these packets are not necessary.

The EABR protocol summary: Table 3 details the procedures of the EABR protocol under different MHs' associativity states. The outstanding feature is that no RRCs is needed so long as the property of associativity interlock remains valid. When this property is violated, the protocol will invoke an LQ or BQ process to quickly locate alternate routes.

Table 3: Summary of route re constructions under varying MH's migrations in the EABR

Associativity valid	Associativity violated		Subnet Bridging MH Moves				Concurrent Moves		
No Route RE-Constructions Are Needed	SRC, DEST & INs Moves		Route subnet	Within	Route Across Subnets	Spans	Ultimately Route Cycle Is Valid	Only Re-	One Construction
	Normal case	Worst Case	No Route Constructions Needed	RE- Are	Network Partitioned REPLY will aborting	is BQ Cycle retry before			
	Moving node will send REPLY Success	BQ, REPLY Cycle either SRC or DEST							

CONCLUSION

In this study, a new method for route reconstruction for the ABR has been proposed. The routing table was amended with a Serial Number field to enhance the optimization while reconstruction of the route. Three MH movements were covered, namely SRC, IN and DEST.

The route re-construction process makes use of the advantage of locality of neighboring MHs to quickly construct alternate and even shorter routes, i.e. route optimization through using the "Here I am" packets

In the original ABR the DEST role in route reconstruction is passive in the case of DEST movement, while in EABR the DEST has an active role in route reconstruction. Beside that, ABR route invalidation is always performed toward the DEST in the case of IN movement, while an optimization is achieved in EABR to invalidate the shortest partial route from the IN toward either the DEST or the SRC.

Future work includes the communication and operation complexity of the EABR compared with the original ABR.

Appendix A: Field description of EABR Packets

Table A.1: The EABR packet header

Routing Header Field	Function
SRC ID	Packet Forwarding
DEST ID	Route Identification
Sequence No.	Duplicates Prevention, Uniqueness
Service Type	Packet Priority
Last IN	Passive Acknowledgement
Next IN	Duplicates Prevention, Routing
Current IN	Acknowledgement, Routing
Serial No	Serial # of IN participating in the constructed route

Table A.2: EABR base header format

Field	Bytes
Type	1/2
Version	1/2
Padding	1
Length	2
Source Address	4
Destination Address	4
CRC	4
Subtotal	16

Table A.3: Formats of the EABR packet headers

Type	Field	Bytes
BQ	Base Header	16
	Sequence No.	4
	No. of Hosts in Route	2
	IN Address	*4
	Associativity Ticks	*2
	No. of Neighboring Nodes	2
	Neighboring Host Address	*4
	Associativity Ticks with Neighbors	*2
	Total	Variable
LQ	Base Header	16
	Sequence No.	4
	Live	2
	Number of Hosts in Route	2
	IN Addresses	*4
	Associativity Ticks	*2
	No. of Neighboring Nodes	2
	Neighboring Host Address	*4
	Associativity Ticks with Neighbors	*2
Total	Variable	
RN	Base Header	16
	Originator ID	4
	Dir	1/2
	Step	1/2
	Total	21
RD	Base Header	16
	Previous Node ID	4
	Sequence No	4
	Total	24
Reply	Base Header	16
	No. of Hosts in Route	2
	Sequence No.	4
Total	Variable	
Data	Base Header	16
	Previous Node Address	4
	Next Node Address	4
	Total	24

Table A.4: The structure of EABR routing table

Field	Byte
Destination ID	4
Source ID	4
Incoming IN ID	4
Outgoing IN	4
Serial No	2
Distance	2
Total No Of Active Routes	2

REFERENCES

1. Royer, E. and C.K. Toh, 1999. A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Commun. Mag.*, First Quarter.
2. Toh, C.K., 2002. *Ad Hoc Mobile Wireless Networks: Protocols and Systems*. Prentice Hall PTR, Prentice Hall, Inc.
3. Perkins, C.E. and P. Bhagwat, 1994. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *Computer Commun. Rev.*, pp: 234-244.
4. Chiang, C.C., H.K. Wu, W. Liu and M. Gerla, 1997. Routing in clustered multihop, mobile wireless networks with fading channel. *Proc. IEEE SICON'97*, pp: 197-211.
5. Murthy, S. and J.J. Garcia-Luna-Aceves, 1996. An efficient routing protocol for wireless networks. *ACM Mobile Networks and Applications J.*, Special Issue on Routing in Mobile Communication Networks, pp: 183-197.
6. Perkins, C.E. and E.M. Royer, 1999. Ad-hoc on-demand distance vector routing. *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications*.
7. Johnson, D.B. and D.A. Maltz, 1996. Dynamic source routing in ad-hoc wireless networks. *Mobile Computing*, Eds. T. Imielinski and H. Korth, Kluwer Academic Publishers, pp: 153-181.
8. Park, V.D. and M.S. Corson, 1997. A highly adaptive distributed routing algorithm for mobile wireless networks. *Proc. INFOCOM '97*.
9. Toh, C.-K., 1996. A novel distributed routing protocol to support ad-hoc mobile computing. *Proc. IEEE Fifteenth Annual International Phoenix Conf. Computers and Communication*, pp: 480-486.
10. Toh, C.K., 1997. Associativity-Based Routing for Ad-Hoc Mobile Networks. *Wireless Personal Commun.*, 4: 1-36.
11. Toh, C.K. and G. Lin, 1998. Implementing associativity-based routing for ad hoc mobile wireless networks. Unpublished Article, March 1998.