HARP - Hybrid Ad Hoc Routing Protocol

Navid Nikaein, Christian Bonnet and Neda Nikaein
Institut Eurécom
2229, Route des Crêtes
B.P. 193
06904 Sophia Antipolis, France
Firstname.Name@eurecom.fr

Abstract— **This paper presents a bandwidth-efficient low-delay routing protocol for mobile ad hoc networks called HARP - hybrid ad hoc routing protocol. HARP is a hybrid scheme combining reactive and proactive approaches. The routing is performed on two levels: intra-zone and inter-zone, depending on whether the destination belongs to the same zone as the forwarding node. We propose a new architecture that separates topology creation from route determination. This architecture optimizes routing performance according to two criteria: network properties and application requirements. Topology creation generates a logical structure with respect to network properties, and the routing protocol discovers and maintains paths to satisfy application requirements.**

Keywords—**Ad hoc networks, routing, zone, stability, graph terminology.**

I. Introduction

A mobile ad hoc network *manet* is a set of wireless mobile nodes forming a dynamic autonomous network through a *fully mobile infrastructure* [1]. Nodes communicate with each other without the intervention of centralized access points or base stations. In such a network, each node acts both as a router and as a host. Due to the limited transmission range of wireless network interfaces, multiple *hops* may be needed to exchange data between nodes in the network, which is why the literature sometimes uses the term *multi-hop* network for a manet. Manet was first referred to as a *packet radio network* in the mid-1960 [2][3]. A mobile ad hoc network includes several advantages over traditional wireless networks, including: ease of deployment, speed of deployment, and decreased dependence on a fixed infrastructure. Manet is attractive because it provides an instant network formation without the presence of fixed base stations and system administrators. Many critical issues have to be addressed in manet such as unicast and multicast routing, QoS support, power control, security, etc. This paper deals with the problem of unicast routing in the mobile ad hoc network.

Designing routing protocols in a mobile ad hoc network is different from wireless networks due to its fully mobile infrastructure, which affects mobility management. In the literature related to routing protocols used in manets, there exists three main routing mechanisms: *proactive*, *reactive* and *hybrid*. In the proactive or table-driven approach, each node keeps up-to-date routes to every other node in the network in its routing tables. Routing information is periodically transmitted throughout the network in order to maintain routing table consistency. In the reactive or on-demand approach, a node initiates a route discovery procedure only when it wants to communicate with its destination. Once a route is established, it is maintained by a route maintenance process until either the destination becomes inaccessible or until the route is no longer desired. In the hybrid approach, each node maintains *only* routing information for those nodes that are within its zone and its neighboring zones. That is, it exhibits proactive behavior within a zone, and reactive behavior between zones. The size and dynamics of a zone differ from protocol to protocol. Consequently in hybrid schemes, a route to each destination within a zone is established without delay, while a route discovery and a route maintenance procedure are required for every other destination. We propose a hybrid routing protocol as its name indicates: HARP - hybrid ad hoc routing protocol.

The paper is organized as follows. Section II provides both related works and our observations about different routing mechanisms. Section III locates our contributions through an architecture that separates topology creation from route determination. Section IV reviews an algorithm that is used in HARP to build a logical zone hierarchy of nodes; and it is called DDR - distributed dynamic routing [4]. Then section V presents different phases of HARP, followed by some mathematical analysises. Finally, section VII provides concluding remarks and highlights future works.

II. Related Work and Our Observations

Several routing protocols have been proposed with the goal of achieving efficiency. Certain table-driven or proactive routing protocols are [5], [6], [7], [8], [9]. The proactive protocols decrease the delay of route determination to a destination, but they waste a significant amount of scarce wireless resources in order to maintain up-to-date routing tables. Such protocols are scalable in relation to the frequency of end-to-end connection. Although proactive protocols are not scalable in relation to the total number of nodes, they can be made scalable if a hierarchical architecture is used. Finally, proactive protocols are not scalable in relation to the frequency of topology change. On the other hand among the on-demand or reactive routing protocols, we can find [10], [11], [12], [13], [14], [15], [16], [17]. The reactive protocols decrease the communication overhead at the expense of an extra delay for route determination; and they are not optimal in terms of bandwidth utilization because of the flooding nature of the route discovery. Reactive protocols remain scalable in relation to the frequency of topology change. Such protocols are not scalable in relation to the total number of nodes, however, similarly to proactice approaches they can be made scalable if a hierarchical architecture is used. Finally, reactive protocols are not scalable in relation to the frequency of end-to-end connection. The hybrid protocols combine proactive and reactive features; and we can find zone routing protocol (ZRP) [18] and zone-based hierarchical link state (ZHLS) routing protocol [19]. The hybrid protocols can provide a better trade-off between communication overhead and delay, but this trade-off is subjected to the size of a zone and the dynamics of a zone. Furthermore, hybrid approaches provide a compromise on scalability issue in relation to the frequency of end-to-end connection, the total number of nodes, and the frequency of topology change.

We propose a zone level hierarchical routing protocol, denoted by HARP - hybrid ad hoc routing protocol. In HARP, each node maintains only routing information of those nodes that are within its zone, and its neighboring zones. The routing is performed on two levels: intra-zone and inter-zone, depending on whether the destination belongs to the same zone as the forwarding node. Intra-zone routing relies on an existing proactive mechanism, and HARP includes reactive mechanism for the inter-zone routing. Zone creation and proactive behavior in relation to network properties are provided by DDR - distributed dynamic routing [4]. On the other hand, HARP is responsible for discovering and maintaining paths to satisfy application requirements. HARP generates and selects path(s) according to the notion of *zone level stability*, which is an extension of node level stability previously used in [14]. The zone level stability is defined by the connection stability of a zone regarding its neighboring zone. HARP applies *early* route maintenance regarding the degree of zone stability. That is, HARP avoids the extra delay caused by path failure during data transmission, and refresh the path before instability period.

Similar to ZRP and ZHLS, HARP is a hybrid approach based on the notion of zone. Different from ZRP and ZHLS, HARP only concerns with finding and maintaining a path between source and destination, and leaves topology generation to DDR - distributed dynamic routing [4]. This separation simplifies the routing protocol, and makes the design modular. Different from ZHLS and ZRP, HARP limits the flooding to subset of forwarding nodes in each zone. This reduces both bandwidth utilization, and energy consumption of non-forwarding nodes. HARP applies zone level stability as a metrics of route determination which is not the case in ZRP and ZHLS. Unlike previous routing protocol, HARP applies *early* path maintenance which is more suitable for priority classes.

III. HARP and DDR

HARP finds and maintains a path in order to route user data regarding application requirements while DDR generates a logical structure with respect to network properties. The application requirements include delay, loss rate, stability, jitter, etc; and the network properties are number of nodes in the network, frequency of end-to-end connection (i.e. number of communication), and frequency of topology change. The network varies from a group of sensors to a group of cars , i.e. from no mobility at all to high mobility; and the application differs in types of the traffic, e.g. data, audio, video. HARP and DDR communicate with each other to satisfy both application requirements and network properties. Fig. 1 shows a layered view of HARP and DDR.

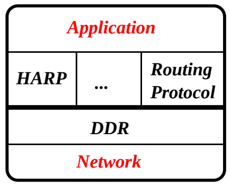


Fig. 1. Layered View of HARP and DDR

IV. Distributed Dynamic Routing Algorithm

A. Basic Idea

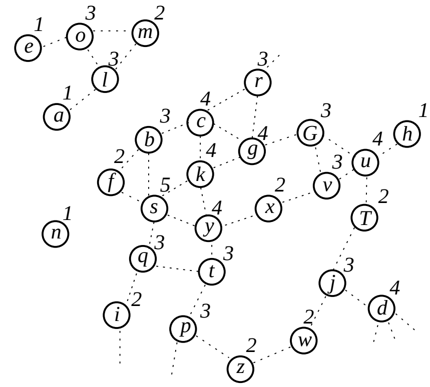
The main idea of the DDR algorithm is to construct a forest from a network topology in a distributed way by using only periodic message exchanges between nodes and their neighbors. Each tree of the constructed forest forms a zone.[1] Then, the network is partitioned into a set of non-overlapping dynamic zones. Each zone is connected via the nodes that are not in the same tree but they are in the direct transmission range of each other. So the whole network can be seen as a set of connected zones. Each node is assumed to maintain routing information *only to those nodes that are within its zone*, and information regarding *only its neighboring zones*.

B. General Description

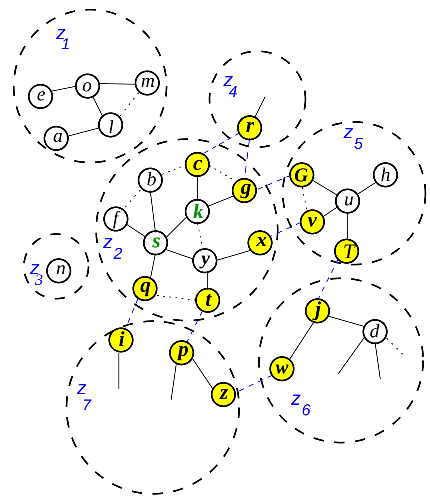
DDR combines two classical notions *forest* and *zone*. Forest is previously used in DST - distributed spanning tress for routing in mobile ad hoc networks [17]. Zone is also used in zone routing protocol (ZRP) [18], and zone-based hierarchical link state (ZHLS) routing protocol [19]. The combination of these two classical notions, zone and forest, provides us with an appropriate structure which in turn can give us a better trade-off between delay and communication overhead. Although DDR benefits from classical concepts like zone and forest, unlike previous solutions it achieves several goals at the same time. Firstly, it provides different mechanisms to reduce routing complexity and improve delay performance. Secondly, it is a fully mobile infrastructure in a strong sense: it does not even require a physical or global location information. Finally broadcast is reduced noticeably.

The DDR - algorithm consists of six cyclic time-ordered phases: *preferred neighbor election, forest construction, intra-tree clustering, inter-tree clustering, zone naming and zone partitioning*, which are executed based on the information provided by beacons. A *beacon* is a periodic message exchanged *only* between a node and its neighboring nodes. The content of a beacon is primitive at the beginning, and it will be enriched during each phase of the algorithm. At the beginning, each node in the network topology carries out the preferred neighbor election algorithm to choose a preferred neighbor. The preferred neighbor of a node is the node that owns maximum neighborhood degree among neighboring nodes. Then, a forest is constructed by connecting each node to its preferred neighbor and vice

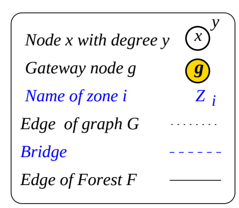
[1] We will use the terms tree and zone interchangeably.



(a) Network Topology



(b) Network Topology under DDR



(c) Legend

Fig. 2. DDR Infrastructure

versa. It has been proven that whatever is the network topology, connecting each node to its preferred neighbor always yields a forest (i.e. we have no cycle) [4]. Next, the intra-tree clustering algorithm is carried out in order to give an appropriate structure within a zone, and build the intra-zone routing table. After that, the inter-tree clustering algorithm provides a natural structure among zones which is kept in the inter-zone routing table of every gateway node. Gateway nodes are the nodes that are not in the same zone, but in the direct transmission range of each other.[2]Each tree is assigned with a name by executing the zone naming algorithm. Since the constructed forest contains a set of trees where each tree is assigned with a name, then the network is partitioned to a set of non-overlapping dynamic zones. Note that DDR only uses beacons to perform every phase of the algorithm, e.g. the forest construction, the intra-tree clustering, the inter-tree clustering, etc. Therefore, it avoids global broadcast throughout the network, thus causing a more efficient use of radio resources.

Fig. 2(a) represents an arbitrary network topology. Once DDR algorithm is executed on each mobile node, the network is partitioned into a set of non over-lapping dynamic zones, as it is illustrated in Fig. 2(b). Each node in the network maintains two tables: the intra-zone table and the inter-zone table. The intra-zone table keeps the information within a zone, and it is filled during the intra-tree clustering algorithm. It contains two fields: node ID number NID, and learned preferred neighbors LEARNED_PN. The field NID represents the ID number of a node that forms an edge of the forest with the owner of the table directly. The field LEARNED_PN represents the nodes that are reachable indirectly via their associated NID in the intra-zone table. Therefore the NID indicates the next hop for the nodes in the LEARNED_PN. Table I (a) and I (b) depict the intra-zone table of node k and s belonging to the zone z_2 regarding Fig. 2(b), and they are denoted by $Intra\_ZT_k$ and $Intra\_ZT_s$ respectively. The intra-zone table gives the current view of a node concerning its zone, and it is updated upon receiving beacons.

In contrast to the intra-zone table, the inter-zone table keeps the information concerning neighboring zones. This table represents the bridges[3], which are detected during the execution of the inter-tree clustering algorithm. Table II shows the inter-zone table of node g, which is denoted by $Inter\_ZT_g$. Each entry in $Inter\_ZT_g$ contains the ID number of a gateway node GNID, the zone ID of this gateway node, i.e. neighboring ZID NZID, and the stability of this neighboring zone regarding node x Z_STABILITY. The zone stability is defined by the connection stability of a zone regarding its neighboring zone. For each beacon received, the zone level stability of the current zone with respect to the beaconing zone is incremented if the euclidean distance of the current ZID and the old ZID becomes smaller than a critical distance; otherwise it is reset. The zone stability is directly related to the ZID which is assigned during the zone naming phase. Indeed, the ZID determination is based on randomly chosen NIDs in a zone. It therefore identifies the zone and it can simply reflect the zone stability.

TABLE I
INTRA-ZONE TABLE OF NODES k AND s REGARDING FIG. 2(B)

NID	LEARNED_PN
s	f, b, q, y, t, x
c	-
g	-

(a) Intra-zone table of node k : $Intra\_ZT_k$

NID	LEARNED_PN
y	x, t
k	c, g
b, f, q	-

(b) Intra-zone table of node s : $Intra\_ZT_s$

TABLE II
INTER-ZONE TABLE OF NODE g : $Inter\_ZT_g$

GNID	NZID	Z_STABILITY
r	z_4	++
G	z_5	++

Therefore, as it is shown in Fig. 2(b) the whole network can be seen as a set of connected zones where each node can communicate with another node in the network.

[2]There exist two kinds of gateway nodes: out-gateway and in-gateway, regarding whether a packet leaves a zone or enters to a zone.

[3]A bridge is an edge that connects two gateway nodes.

V. HYBRID AD HOC ROUTING PROTOCOL

A. Basic Idea

HARP aims at establishing the most stable path from a source to a destination in order to improve delay performance due to path failure. HARP applies the path discovery mechanism between zones that intends to limit flooding in the network, and that filters the candidate paths as soon as possible according to the stability criteria. As stability is the most desired parameter, HARP offers different mechanisms to anticipate path failure along with path maintenance procedure whose complexity is reduced by the proactive nature of the routing algorithm within a zone. These procedures reduce the delay that stems from a path failure during data transmission.

B. Routing Mechanism

The routing mechanism in HARP is performed on two levels: intra-zone and inter-zone, depending on whether the destination belongs to the same zone as the forwarding node. The intra-zone routing involves only *forwarding* because HARP applies proactive approach within a zone (inherited from DDR), which means that route generation and selection are performed during the intra-tree clustering phase of DDR. Therefore the only task of a node within a zone is to *forward* the data traffic along a pre-computed path. The inter-zone routing implies *routing* since HARP uses a reactive approach between zones to generate routes, and selects the most stable one to the destination. The inter-zone routing includes: *path discovery* and *path maintenance* phases in order to discover and maintain a path. The path discovery phase consists of two parts: path request PREP and path reply PREP. The path request propagates from zone to zone via the gateway nodes thanks to the both intra-zone table and inter-zone table, while the path reply is unicasted back from the destination to the source. Inside a zone, the path request follows the tree structure provided by DDR. As a consequence, the path request propagation is limited to a subset of forwarding nodes. After this limited flooding, several paths may candidate for a given destination. The destination chooses the most stable path and sends a path reply back to the source. Path maintenance provides different mechanisms to ensure that packets can be safely transmitted from source to destination. Each path is associated with a refresh time after which a new path discovery phase is triggered. This is done to avoid path failure as the network topology may change after a certain time. Nodes may also have unanticipated behavior that may cause path failure. In this case a reactive path recovery procedure is triggered in addition to the previous mechanism, which can be seen as a proactive path recovery.

B.1 Path Discovery Algorithm

The main objective of the path discovery algorithm is to generate and select the most stable path between source and destination. Note that the stability is a concave function, and it is defined by the connection stability of a zone regarding its neighboring zone. Assume that a node say s wants to send data to its destination d. Before sending data to the node d, node s checks if node d exists in its intra-zone table. If so, node s *forwards* the data towards node d according to its intra-zone table without any delay. We denote this case intra-zone routing. Otherwise, node d belongs to a different zone as node s, so that node s sends a path request PREQ to every other neighboring zone z via gateway nodes g. We denote this case *inter-zone routing*. The stability is unknown at the beginning, but the *out-gateway* nodes will change the stability value as they propagate PREQ message from zone to zone. Each intermediate node routes PREQ through its zone up to out-gateway nodes according to its intra-zone table. In contrast, each out-gateway node forwards PREQ from its zone to the next zone(s) according to its inter-zone table, and updates the current stability of the path. Upon receiving the PREQ by an in-gateway node g, this node verifies whether destination node d belongs to its intra-zone table or not. If so, g forwards this PREQ according to its intra-zone table; otherwise g routes this PREQ throughout its zone, and possibly to every other neighboring zones. In order to assure that the path reply PREP message traverse exactly the same path back to the source, each gateway node keeps some *routing state information* found in PREQ and PREP for each traffic. This routing state information includes: gateway ID from which a path request is received (or corresponding gateway), and path stability. The destination node d will eventually receive PREQ if it is reachable by the source. Then node d chooses the most stable path among the received path requests. If more than one path with the same stability is found, node d chooses one randomly. Afterward, node d creates a path reply PREP corresponding to the most stable path, and routes this PREP back to the gateway from which it received the PREQ. Route reply includes the discovered stability which helps the source node during the path maintenance procedure (c.f. V-B.2). Each node routes back the PREP towards the corresponding gateway node according to the either zone tables and the routing state information. Once the PREP arrives at the source node s, s creates data packets, and sends them through the discovered path. Consequently, each node forwards the data packet to the next zone via the corresponding gateway towards the destination according to the both intra-zone table and inter-zone table. Furthermore, a gateway node filters a path request if it receives a path request with a lower degree of stability than the earlier one for the same connection; and it updates the routing state information if it receives a path request with a higher degree of stability. This filtering is called *selective filtering*, and guarantees the uniqueness of the most stable pair of gateway nodes.

B.2 Path Maintenance Algorithm

The goal of the path maintenance algorithm is to improve the delay performance based on the path stability. The path maintenance algorithm includes: *path refreshment*, *path waiting time*, *path error*, and *new path discovery* phases. The *path refreshment* phase constructs a *new* path before a period of time called *path discovery update time*, and then switches the traffic to this new path at the beginning of the update time. This path discovery update time is estimated based on the discovered path stability. Although the source can keep communicating with the destination after the update time, it may confront high probability of link failures. Therefore, source node renews the path discovery phase if it approaches to the path discovery update time. If a link failure occurs in the meantime, the corresponding node holds the traffic for a duration of *waiting time* expecting to receive some routing information corresponding to its target gateway node, and at the same time it sends a *path error* back to the source. The rationale for waiting time is that HARP applies the proactive approach inside a zone, and there is a chance of receiving new routing information embedded in the periodical beacon. Furthermore, source node benefits from the discovered path stability to get the actual stability of every zone during data transmission. Note that the stability is a concave function. For this purpose, the source node puts the discovered stability in data packets so that each out-gateway node can verify whether this stability satisfies the actual stability of the next zone. If not, this out-gateway node sends back a *path error* with the actual stability to the source node so that the source can update the path discovery update time accordingly. The path error is unicasted back to the source if the return path still exists; otherwise it is broadcasted to the network with an appropriate time-to-live (TTL). Once the source receives a path error, it initiates a *new path discovery*.

B.3 Example

Fig. 3 depicts the format of both *path request* PREQ and *path reply* PREP messages; they include a source address *src@*, a destination address *dest@*, a port number *port#*, a gateway node ID *GID*, and the estimated stability *stability*. The stability is a concave function, and it is calculated by means of PREQ.*stability* = $min\{$ PREQ.*stability*, $Inter\_ZT_{out\_gateway}$.Z_STABILITY.

src@	*dest@*	*port#*	*GID*	*stability*

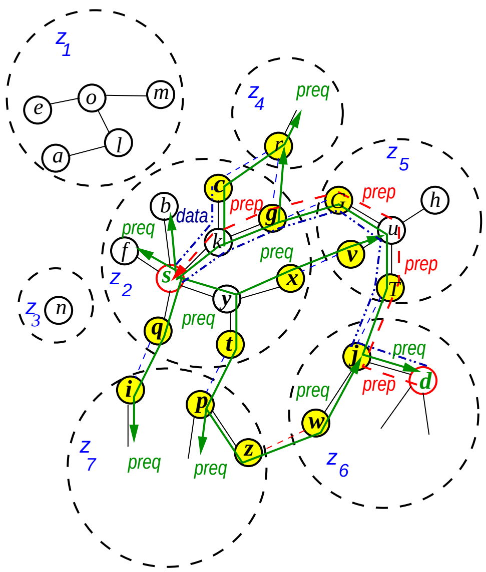
Fig. 3. Fields of a path request PREQ or a path reply PREP

Fig. 4 shows the fields of a *data packet* DPKT. The fields of a data packet includes a source address *src@*, a destination address *dest@*, a port number *port#*, a gateway node ID number *GID*, the discovered stability *stability*, and the data *data*.

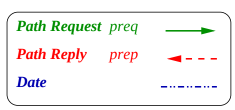
src@	*dest@*	*port#*	*GID*	*stability*	*data*

Fig. 4. Fields of a data packet DPKT

For example in Fig. 5, consider the scenario of intra-zone routing where node s wants to communicate with one of the nodes within its zone z_2, e.g. $f, b, q, y, k, c, g, x, t$. According to its intra-zone table (see table I(b)), node s can reach the nodes x, t via y, and the nodes c, g through k, while other nodes f, b, k are directly reachable. Therefore intra-zone routing table always indicates the next hop for each destination within the zone. So if node s wants to send some data to node c, it firstly forwards DPKT$(s@, c@, p\#, k@, \infty, data)$ including the date to the node k and then k passes DPKT$(s@, c@, p\#, c@, data)$ to the node c (see dash-dot-dot line between s and c in the figure).



(a) Routing under HARP



(b) Legend

Fig. 5. Routing and forwarding in HARP : path discovery, path maintenance, and data traffic delivery

Inter-zone routing occurs if the destination node belongs to a different zone as the source node. For instance in Fig. 5, source node s is a member of zone z_2 while destination node d belongs to zone z_6. Therefore node s will send a path request PREQ$(s@, d@, p\#, s@, \infty)$ through its zone. Note that at the beginning the source node sets the field of *GID* to its address $s@$, and the *stability* to ∞. This PREQ crosses from the zone z_2 to the zones z_4, z_5, z_7 via gateway nodes c, g, x, q, t respectively; as it is illustrated in the figure (follow the solid arrow lines from s to d). These gateway nodes update the field of *GID* to their ID, and store the old value of *GID* by means of corresponding_gate = s@. Also, these nodes forward the PREQ to their neighboring zones according to their interzone table: i.e. via gateways r, G, v, i, p respectively. Now the PREQ traverses the zones z_5, z_7 to the zone z_6 via z, T. Note that node T applies selective filtering to one of the path requests received from G and v. Assume that node T chooses node G as the most stable gateway node. Finally the PREQ enters the zone z_6 where node d belongs to via w, j. The nodes w, j forwards this PREQ to d according to their intrazone routing table. Upon receiving this PREQ by destination node d, d will choose the most stable one. Assume the path via d, j, T, G, g, s as the most stable path, so that node d creates the path reply corresponding to the chosen path PREP$(d@, s@, p\#, j@, stability)$; where the $j@$ indicates the address of the gateway in which the chosen path has just been received, and the $stability$ points to the whole path stability (in the figure follow the dashed arrow line from d to s). Node j updates the fields of *src@* and *GID* in the path reply to PREP$(j@, s@, p\#, T@, stability)$, and saves the old value of *src@* by means of corresponding_gate=@d. The routing state information at each gateway node points to the next gateway towards the destination at the time of data packet transmission. Node T also updates the path reply and pass the PREP to node G instead of node v since node v has been filtered out because of the stability priority. Each intermediate node routes the PREP to the next gateway found in PREP according to its intra-zone table. Once PREP arrives at the source node s, s creates data packets DPKT$(s@, d@, p\#, g@, stability, data)$; where $g@$ is the gateway node through which the path reply has been received and the *stability* indicates the discovered stability during the path discovery. Then, node s sends the DPKT to the destination for the duration of the path discovery update time (follow the dash-dot-dot line in the figure). This path is refreshed if s still wants to send packets to d.

VI. Mathematical Analysis

We compare the communication overhead for topology creation among DDR, ZHLS and ZRP. Assume that all the nodes are uniformly distributed in the network. Consider a network with N nodes and M zones. Let ρ be the routing zone radius in term of number of hops, therefore ρ^2 is the amount of intrazone control traffic produced by each node. The total amounts of communication overhead generated by ZHLS, ZRP and DDR are summarized in TABLE III.

It can be shown that the communication overhead (C.O.) of $C.O._{DDR}$ and $C.O._{ZRP}$ are close,

TABLE III
COMMUNICATION OVERHEAD COMPARISON

Protocol	Topology creation
ZHLS	$N^2/M + NM$
ZRP	$N\rho^2$
DDR	N^2/M

but it is subjected to the choice of ρ and M. ZHLS generates NM messages more than DDR for creating topology, when the number of zones are the same. The communication overhead generated to perform route discovery operation for ZHLS, ZRP, and HARP is: $C.O._{RD} = P(N + Y)$, where $P = 1 - 1/M$ is the probability that destination belongs to a different zone as the source, and Y is the total number of nodes affected by the directed path where the reply packet transits. For the same number of zones, routing overhead generated by ZHLS and HARP are close. It can be shown that HARP generates less routing overhead than ZRP, because zones (N zones) in ZRP are highly overlapped. Therefore, the overall overhead generated by HARP and DDR is less than both protocols.

VII. CONCLUSION

This paper presents a hybrid routing protocol, called HARP, which benefits from the separation of logical topology creation/maintenance tailored to the routing protocol. This separation leads to a clear and modular design of routing protocols according to specific application requirements. HARP establishes and maintains the most stable path from source to the destination in order to enhance the reliability of data transmission in mobile ad hoc networks. HARP uses a hierarchical topology provided by DDR in order to reduce the control message overhead. The mathematical analysis shows that the overall amount of overhead induced by HARP and DDR is smaller than ZRP and ZHLS. In our future work, we will evaluate the performance of HARP under various condition of traffic and mobility with the related protocols.

ACKNOWLEDGMENT

The authors wish to thank Shiyi Wu, David Turner and Sergio Loureiro for their useful discussions.

REFERENCES

[1] "Mobile ad hoc network (MANET)," Web Site at: $http://www.ietf.org/html.charters/manet-charter.html$, Work in Progress.

[2] –, "Special issue on packet radio networks," *Proceeding of IEEE*, vol. 75, no. 1, 1987.

[3] "Packet radio networks," Web Site: $http://www.tapr.org/tapr/html/pktf.html$.

[4] Na. Nikaein, H. Labiod, and C. Bonnet, "DDR-distributed dynamic routing algorithm for mobile ad hoc networks," in *MobiHOC 2000*. IEEE, Augest 2000, pp. 19–27.

[5] C. E. Perkins and P. Bhagwat, "Highly dynamic destination sequenced distance-vector routing (DSDV) for mobile computers," in *SIGCOMM'94*.

[6] S. Murthy and J.J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," *ACM Mobile Networks and Applications Journal*, 1996.

[7] T. Chen and M. Gerla, "Global state routing: A new routing scheme for ad-hoc wireless networks," in *Proc. IEEE ICC'98*.

[8] C-C. Chiang, "Routing in clustered multihop, mobile wireless networks with fading channel," in *IEEE SICON'97*, pp. 197–211.

[9] A. Iwata, C-C. Chiang, G. Peiand M. Gerla, and T.-W. Chen, "Scalable routing strategies for ad hoc wireless networks," *IEEE Journal on Selected Areas in Communications, Special Issue on Ad-Hoc Networks*, vol. 17, no. 8, pp. 1369–1379, 1999.

[10] M. Jiang, J. Li, and Y. C. Tay, "Cluster based routing protocol," IETF Draft, 1999.

[11] C. E. Perkins, E. M. Royer, and S. R. Das, "Ad hoc on-demand distance vector routing," IETF Draft, 1999.

[12] D. B. Johnson and D. A. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Network*, Kluwer Academic Publishers, 1996.

[13] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *INFOCOM'97*.

[14] C-K. Toh, "A novel distributed routing protocol to support ad-hoc mobile computing," *IEEE International Phoenix Conf. on Computers and Communications*, IPCCC'96.

[15] R. Dube, C. D. Rais, K. Wang, and S. K. Tripathi, "Signal stability based adaptive routing (SSR) for ad-hoc mobile networks," *IEEE Personal Communications*, vol. 4, no. 1, pp. 36–45, 1997.

[16] Y-B. Ko and N. H. Vaidya, "Location-aided routing in mobile ad hoc networks," *4th Annual International Conference on Mobile Computing and Networking*, MOBICOM'98.

[17] S. Radhakrishnan, N. S. V. Rao G. Racherla, C. N. Sekharan, and S. G. Batsell, "DST - a routing protocol for ad hoc networks using distributed spanning trees," *IEEE Wireless Communications and Networking Conference*, pp. 100–104, 1999.

[18] M. R. Pearlman and Z. J. Hass, "Determining the optimal configuration for zone routing protocol," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1395–1414, 1999.

[19] M. Joa-Ng and I-Tai Lu, "A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks," *IEEE on Selected Areas in Communications*, vol. 17, no. 8, pp. 1415–1425, 1999.