Routing Protocols for Efficient Communication in Wireless Ad-hoc Networks *

Ioannis Chatzigiannakis
Computer Technology Institute (CTI) and CEID, University of Patras, 26500 Greece
ichatz@cti.gr

Panagiotis Kokkinos
Computer Technology Institute (CTI) and CEID, University of Patras, 26500 Greece
kokkinop@ceid.upatras.gr

Christos Zaroliagis
Computer Technology Institute (CTI) and CEID, University of Patras, 26500 Greece
zaro@ceid.upatras.gr

ABSTRACT

In this paper we demonstrate the significant impact of the user mobility rates on the performance on two different approaches for designing routing protocols for ad-hoc mobile networks: (a) the route creation and maintenance approach and (b) the "support" approach, that forces few hosts to move acting as "helpers" for message delivery. We study a set of representative protocols for each approach, i.e. DSR and ZRP for the first approach and RUNNERS for the second. We have implemented the three protocols and performed a large scale and detailed simulation study of their performance. Our findings are: (i) DSR achieves low message delivery rates but manages to deliver messages very fast; (ii) ZRP behaves well in networks of low mobility rate, while its performance drops for networks of highly mobile users; (iii) RUNNERS seem to tolerate well (and in fact benefit from) high mobility rates.

Based on our investigation, we design and implement two new protocols that result from the synthesis of the investigated routing approaches. We conducted an extensive, comparative simulation study of their performance. The new protocols behave well both in networks of diverse mobility motion rates, and in some cases they even outperform the original ones by achieving lower message delivery delays.

Categories and Subject Descriptors: C.2.1 [Computer-Communication Networks]: Network Architecture and Design [Wireless Communication], C.2.2 [Computer-Communication Networks]: Network Protocols [Routing protocols]

General Terms: Algorithms, Design, Experimentation, Performance

Keywords: Wireless Ad-hoc, Mobile Networks

*This work has been partially supported by the IST Programme of the European Union under contract number IST-2005-15964 (AEOLUS) and the Programme PYTHAGORAS under the European Social Fund (ESF) and Operational Program for Educational and Vocational Training II (EPEAEK II).

PE-WASUN'06, October 6, 2006, Torremolinos, Malaga, Spain.

1. INTRODUCTION

A fundamental problem in ad-hoc mobile networks is to allow communication remote hosts and is at the core of many algorithms, such as for counting the number of hosts, electing a leader, data processing etc. In ad-hoc mobile networks, a node MH_s that wants to communicate with MH_r might not be within communication range but could communicate if other hosts lying "in between" them are willing to forward packets for them. The simplest way to establish communication in such networks is to perform flooding. By flooding we mean a distributed process of broadcasting a packet to all nodes within the transmission radius of the initiating sender and so on, until all nodes in the sender's "connected component" receive the packet. It is evident that such an approach will waste a large portion of the available bandwidth and thus is inefficient.

Towards avoiding flooding several protocols have been proposed in the related literature. A possible categorization of these protocols is based on the approach taken for establishing communication. One approach tries to construct and dynamically update paths between the mobile hosts. The second approach instead avoids path creation and maintenance by rather taking advantage of the hosts movement and accidental meetings in the network area.

The Path Construction and Maintenance Approach. In an attempt to overcome the inefficiency of flooding, some algorithms have been presented that try to construct a connectivity related dynamic data structure (such as a dynamic graph) that represents the topology of the network by a series of message passes. By taking advantage of such a data structure, messages can be transmitted over a limited number of intermediate hosts that interconnect MH_s with MH_r, also known as *routing paths* or *routes*. Indeed, this approach is common in ad-hoc mobile networks that either cover a relatively small space (i.e. the temporary network has a small diameter in terms of the transmission range), or are dense (i.e. comprised of thousands of wireless hosts). In such cases, since almost all "intermediate" locations are occupied by some hosts, communication is possible and broadcasting can be efficiently accomplished [1].

In [2, 18] the authors propose the Dynamic Source Routing (DSR) protocol, which uses *on-demand route discovery.* There exist many variations of the DSR protocol that try to optimize the route discovery overhead. In [19] the authors present the AODV (Ad Hoc On Demand Distance Vector routing) protocol that also uses a demand-driven route establishment procedure. More recently TORA (Temporally-Ordered Routing Algorithm, [15, 16]) was designed to minimize reaction to topological changes by localizing routing-related messages to a small set of nodes near the change. In [17, 8] the authors attempt to combine proactive and reactive approaches in the Zone Routing Protocol (ZRP), by initiating route discovery phase on-demand, but limit the scope of the proactive procedure only to the initiator's local neighborhood. In [11] the authors propose the Location-Aided Routing (LAR) protocol that uses a global positioning system to provide location information that is used to improve the performance of the protocol by limiting the search for a new route in a smaller "request zone".

The "Support" Approach. In contrast to all such methods, in [5] a new framework is introduced for protocols designed for highly dynamic movement of the mobile users. The idea of the approach is to take advantage of the mobile hosts natural movement by exchanging information whenever mobile hosts meet incidentally. Specifically, the protocol framework proposes the idea of forcing a small subset of the deployed hosts to move as per the needs of the protocol. This set of hosts is called the "support" of the network (denoted by Σ). Assuming the availability of such hosts, the designer can use them suitably *by specifying their motion* in certain times that the algorithm dictates.

This approach may be useful in cases of rapid deployment of a number of mobile hosts, where it is possible to have a small team of fast moving and versatile vehicles (e.g. cars, jeeps, motorcycles or helicopters), to implement the support. We interestingly note that this small team of fast moving vehicles can also be a collection of independently controlled mobile modules, i.e. robots. This specific approach is inspired by [20] that studies the problem of motion co-ordination in distributed systems consisting of such robots, which can connect, disconnect and move around. [20] deals with metamorphic systems where (as is also the case in our approach) all modules are identical. Note that our approach of having the support moving in a coordinated way, *i.e. as a chain of nodes*, has some similarities to [20].

Definition 1. *The subset of the mobile hosts of an ad-hoc mobile network whose motion is determined by a network protocol* $\mathcal{P}$ *is called the* support Σ *of* $\mathcal{P}$. *The part of* $\mathcal{P}$ *which indicates the way that members of* Σ move and communicate *is called the* support management subprotocol M_Σ *of* $\mathcal{P}$.

The scheme follows the general design principle of mobile networks (with a fixed subnetwork however) called the "two tier principle" [10], stated for mobile networks with a fixed subnetwork however, which says that any protocol should try to move communication and computation to the fixed part of the network. The assumed set of hosts that are coordinated by the protocol *simulates* such a (skeleton) network; the difference is that the simulation actually constructs a coordinated *moving* set.

In a recent paper [12], Q.Li and D.Rus present a model which has some similarities to our approach. The authors give an interesting, yet different, protocol to send messages, which forces *all the mobile hosts to slightly deviate (for a short period of time) from their predefined, deterministic routes, in order to propagate the messages.* Their protocol is, thus, *compulsory* for any host and it works only for deterministic host routes. Moreover, their protocol considers the propagation of only one message (end-to-end) each time, in order to be correct. In contrast, the support scheme allows for simultaneous processing of many communication pairs. In their setting, [12] shows optimality of message transmission times.

Our Contribution. The performance and correctness of protocols of the two approaches seems to depend heavily on the mobility rate of hosts. Indeed, in certain ranges of mobility the protocols may even fail to deliver messages, thus being incorrect. In this work, we study the particular effect of this parameter on the efficiency of representative protocols following the two approaches: namely, DSR, ZRP and RUNNERS protocols. We have implemented these three protocols and carried out a large-scale simulation and extensive experimental evaluation of various important measures of their performance. We focus on the impact of mobility on the performance measures studied. The detailed investigation highlights the advantages and disadvantages of each approach and its suitability for a certain network and user profile. In particular, we demonstrate that DSR and ZRP behave very well in networks of low mobility, as shown by the high message delivery rate and the small message delivery delays they achieve in this case, while the RUNNERS performance is less dependent on the mobility rate (actually, it seems to benefit from high mobility rates). We would like to mention that our experimental outcomes validate, the conjecture stated in [5] according to which in cases of high mobility rate, any algorithm that tries to maintain a global structure with respect to the temporary network will even become erroneous if the mobility rate is faster than the rate of updates of the algorithm.

In the light of the above, we develop and evaluate two new protocols based on the *synthesis* of DSR and ZRP with RUNNERS. Our study shows that the new protocols operate well in networks of diverse mobility rates achieving high message delivery rates, while keeping the message delivery delays low. In fact, in some cases (see Sec. 5) the resulting protocol syntheses outperform the original ones by achieving lower message delivery delays. We believe that our results on synthesizing different routing protocols are a step forward in creating a unifying framework for ad-hoc mobile networks comprised of hosts with heterogeneous mobility rates.

The above findings are based on a particular (random) message generation model. The message generation pattern is in fact a fixed traffic model, arising in certain applications. In particular, the traffic model is based on the assumption that the mobile hosts do not execute a real-time application that requires continuous exchange of messages, but can tolerate delays in the order of fractions of seconds to several seconds. Our focus is on applications that are asynchronous in nature so that they can tolerate reasonable end-to-end delays. Examples of such applications include electronic mail, database synchronization between a mobile terminal and a central database, and certain types of event notification. It is interesting to also study an extended traffic model in combination with the existing mobility model.

2. THE MODEL OF MOTIONS

Throughout the paper, by *mobile hosts* (or mobile users) we shall refer to those hosts (or users) that do not belong to the support Σ. Let u be the number of mobile hosts and let the size of the support be $|\Sigma| = k$.

The model has two parts: one part models the motion space (i.e. the network area), while the other part models the motions the mobile hosts perform in the motion space.

The Motion Space. To model the motion space, the three-

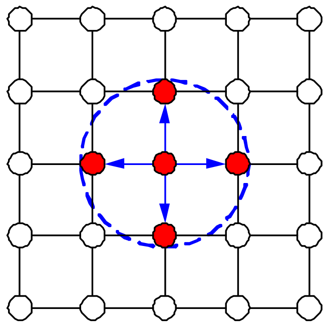

Figure 1: Transmission in 2D grid-motion graphs for $\beta = 2$

dimensional space $\mathcal{S}$ (with possible obstacles) is mapped to a *motion graph* $\mathcal{G} = (V, E)$ (which will be defined shortly). Let $|V| = n$. The area where the stations move is abstracted by a graph by neglecting the detailed geometric characteristics of the motion. It is first assumed that each mobile host has a transmission range represented by a sphere tr having the mobile host at its center. This means that any other host inside tr can receive any message broadcast by this host. Then this sphere is approximated by a cube tc with volume $\mathcal{V}(tc)$, where $\mathcal{V}(tc) \simeq \mathcal{V}(tr)$. The size of tc can be chosen in such a way that its volume $\mathcal{V}(tc)$ is the maximum integral value that preserves $\mathcal{V}(tc) < \mathcal{V}(tr)$. To capture mobility and transmission range of hosts in $\mathcal{S}$, the space $\mathcal{S}$ is quantized into cubes of volume $\mathcal{V}(tc)$.

The *motion graph* $\mathcal{G} = (V, E)$, which corresponds to the above quantization of $\mathcal{S}$, is constructed in the following way. There is a vertex $v \in V$ for each cube of volume $\mathcal{V}(tc)$. Two vertices v and w are connected by an edge $(v, w) \in E$ iff the corresponding cubes are adjacent. Let $\mathcal{V}(S)$ be the volume of space $\mathcal{S}$. Clearly, $n = \mathcal{O}(\mathcal{V}(S)/\mathcal{V}(tc))$ and consequently n is a rough approximation of the ratio $\mathcal{V}(S)/\mathcal{V}(tr)$, i.e. n measures area size in terms of transmission range.

In this work we use a variation of this basic motion graph model and instead of dividing the original motion space $\mathcal{S}$ into consecutive cubes of volume $\mathcal{V}(tc)$, we introduce a variable $\beta \geq 1$ so that $\mathcal{S}$ is divided into consecutive cubes of volume $\mathcal{V}(tc)/\beta$. When $\beta = 1$ we get the basic model, however for $\beta > 1$, if a mobile host that resides on node u broadcasts a message, this message is received by any other host i that resides on node v_i such that $d(u, v_i) < \beta$. We do this generalization to allow path construction between hosts located in adjacent cubes. For a graphical representation of a transmission in the case when $\beta = 2$ in 2D grid motion graphs, see Figs. 1.

The Motion of the Mobile Hosts. In order to model the motion of the mobile hosts we use a *random motion model* which is a natural starting point aiming at studying the average-case efficiency of protocols for basic communication. To this end, the motion of a mobile host is modelled by concurrent and independent random walks. This type of motion was used in [5, 4], assumes only local knowledge of $\mathcal{G}$ and hosts move using at most one edge in a single step. Note that movement following independent random walks approximates quite accurately dense motions of many users, since it captures the characteristics of motions. In fact, the assumption that the mobile hosts move randomly, either according to uniformly distributed changes in their directions and velocities, or according to the random waypoint mobility model by picking random destinations, has also been used in previous research (see e.g. [7, 9]).

We assume that the random motion is performed *independently* of the support motion, i.e. we exclude the case where some of the hosts are deliberately trying to avoid the members of the support. This is a pragmatic assumption usually followed by application protocols.

We define the *mobility (or motion) rate* $\mu \in [0, 1]$ of a mobile host as the probability with which the mobile host will move in a round of the protocol.

3. THE PROTOCOLS

In this section, we give a short description of DSR, ZRP, and RUNNERS protocols. Throughout the paper, by *mobile hosts* (or mobile users or mobile nodes) we shall refer to those hosts (or users or nodes) that belong to the ad-hoc network. In the case of the RUNNERS protocol only, any reference to mobile hosts (or users or nodes) *excludes* the members of the support.

3.1 The Runners Support Routing Protocol

The Runners Support Routing Protocol (RUNNERS, [5, 3]) targets (i) networks with high mobility rates and (ii) sparse networks that tend to be disconnected. In such cases, attempting to find a routing path may be very difficult (since nodes move very fast and routes become invalid) or even impossible (as no intermediate nodes exist). The protocol relies on the existence of a small group of nodes (with specialized specifications, such as fast moving and versatile vehicles) that is called the "support" Σ of the network. The protocol assumes a *setup phase* in the network, during which the support Σ of the network is formed.

The idea is to force this small team of $k = |\Sigma|$ mobile hosts to move fast enough so that they cover (in sufficiently short time) the entire network area (the move of the rest is not affected by the protocol). In particular, support members move performing concurrent and independent random walks in the network area. Each mobile host that wants to send some information to another host sends its message to any member σ of Σ whenever σ comes within its transmission range. The presence of a support member is advertised using an underlying subprotocol (see [5]) similar to that used in AODV. In addition, during this communication, it can receive any messages that have designated it as the receiver.

A member of the support needs to store all undelivered messages, and maintain a list of receipts to be given to the originating senders. When two runners meet, they exchange any information given to them by senders encountered using a *synchronization subprotocol*, a distributed algorithm that is partially based on the *two-phase commit* algorithm as presented in details in [13]. In this way, a message delivered by some runner will be removed from the memory of the rest of runners encountered, and similarly delivery receipts already given will be discarded from the memory of the rest of runners.

Hence, the support serves as a set of mobile *relay* hosts, that temporarily buffer messages (and delivery receipts) until they get delivered. The expectation is that as this small team of mobile hosts moves randomly around the area of the ad-hoc network, it will cover the entire area in sufficiently short time, thus servicing effectively all messages.

3.2 The Dynamic Source Routing Protocol

The Dynamic Source Routing (DSR, [2]) is an on-demand routing protocol that is based on the concept of source routing. Mobile nodes are required to maintain route caches that contain the source routes of which the mobile user is aware.

The protocol consists of two major phases: *route discovery* and *route maintenance*. When a mobile node has a packet to send to some destination, if the node does not have such a route, it initiates route discovery by broadcasting a *route request* packet (containing the address of the destination, the source node's address and a unique identification number). Each node receiving the packet checks whether it knows of a route to the destination. If it does not, it adds its own address to the route record of the packet and then forwards the packet along its outgoing links.

A *route reply* is generated when the route request reaches either the destination itself, or an intermediate node which contains in its route cache a route to the destination. Note that the packet reaching either the destination or such an intermediate node, contains a route record yielding the sequence of hops taken. If the node generating the route reply is the destination, then it places the route record contained in the route request into the route reply. If the responding node is an intermediate node, then it will append its cached route to the route record and then generate the route reply. Finally the node transmits a route reply packet, by using the reverse of the route record.

Route maintenance is accomplished through the use of route error packets and acknowledgments. Route error packets are generated at a node when the data link layer encounters a fatal transmission problem, while acknowledgments are used to verify the correct operation of the route links. When a route error packet is received, the hop in error is removed from the node's route cache and all routes containing the hop are truncated at that point.

3.3 The Zone Routing Protocol

The Zone Routing Protocol (ZRP, [17, 8]) is a hybrid proactive/reactive routing protocol, suitable for networks with large network spans and diverse mobility rates. Each node proactively maintains routes to destinations within a local region, referred to as the routing zone, and the topological changes are only locally propagated. A node's *routing zone* is defined as a collection of nodes whose minimum distance (in hops) from the node in question is no greater than a parameter referred to as the zone radius. When data needs to be sent to a destination-member of its routing zone, the necessary routing information is instantly available. Otherwise, a route query/reply mechanism is initiated in order to acquire a route to the destination. The knowledge of the routing zone topology improves the efficiency of the reactive route query/reply mechanism and assists in the maintenance of the discovered route after changes in network topology.

The ZRP consists of three different protocols: (i) the *Intrazone Routing Protocol* (IARP) that proactively maintains the routing zone topology of each node; (ii) the *Interzone Routing Protocol* (IERP) that re-actively acquires routes to destinations beyond the routing zone using a route query/reply mechanism and also maintains the discovered routes after changes in network topology; and (iii) the *Bordercast Resolution Protocol* (BRP) that is a service used by the IERP for the efficient distribution, in the network, of its route query messages.

A route query message is not blindly broadcasted from neighbor to neighbor, but it is directed outwards from the query source node and away from covered routing zones (after a node receives a route query its routing zone is considered as covered). IARP and IERP can be derived from existing proactive and reactive respectively protocols with only a few modifications.

The ZRP supports global route discovery based on source routing, next hop routing, or various combinations of both, providing in any case loop-free routing. Finally, the performance of the ZRP is determined by the routing zone radius. The right value for the zone radius depends on the density of the network, on the mobility of the nodes, and on the route query rate. The zone radius can be configured once for each node prior to the network deployment, or the ZRP can dynamic configure the zone radius of each node according to changes in the network behavior. For more information on how to fine-tune the zone radius see [17].

4. PROTOCOL SYNTHESES

As we will see in the Experimental Results Section, our simulation study suggests that the routing protocols that follow the path construction approach are more suitable for networks of low mobility rates since they tend to deliver the majority of the messages achieving relatively short delays. However, when the mobility rate of the hosts increases, the network experiences longer delays and lower delivery rates (i.e. messages are lost). On the other hand, the RUNNERS protocol that follows the support approach seems to be less affected by the mobility rate of the hosts, since it manages to deliver almost all the messages for all rates of motion $\mu \in [0, 1]$. Although messages reach their destination, the message delivery delays are (on the average) higher than the corresponding delays of the protocols that follow the path construction approach. However, in contrast to the path construction protocols, the RUNNERS protocol, as the mobility rate of the hosts increases, benefits from their high mobility and reduces the average message delivery delays.

In view of the above, we are interested in designing new routing protocols that achieve good results (in terms of message delivery rates and delays) in ad-hoc mobile networks consisting of mobile hosts with mixed mobility rates. Ideally, we would like to synthesize the two different approaches in a way such that the new routing protocols will benefit from the very high delivery rates of the support approach, while delivering the messages with short delays as done by the protocols that follow the path construction approach.

In the rest of this section, we present two new protocols, which we call DSR-R and ZRP-R, that result from the synthesis of DSR and ZRP with the RUNNERS protocol. The experimental study conducted indicate that the two new protocols operate well in networks of mixed mobility rates, since they manage to deliver almost all the messages with shorter delays than that of the support approach. In fact, in some cases the new protocols outperform the original ones by achieving lower message delivery delays compared to the path construction approach.

4.1 The DSR-Runners Synthesis

In this section, we present the DSR-R protocol resulting from the synthesis of the DSR and RUNNERS protocols. The general idea is that transmissions to destination nodes that are "close" to the sender are carried out by the DSR protocol,

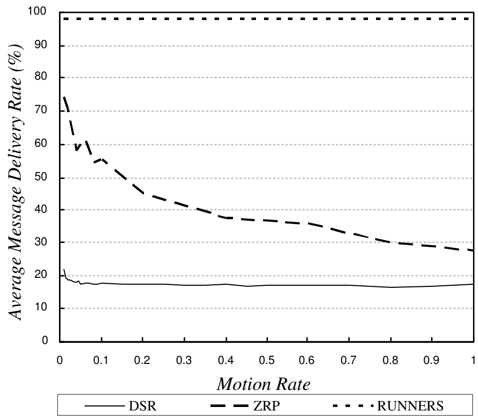

Figure 2: Average Message Delivery Rate ($\mathcal{D}$) in grid graphs ($n = 121$, $\lambda = 0.02$, $u = 50$).

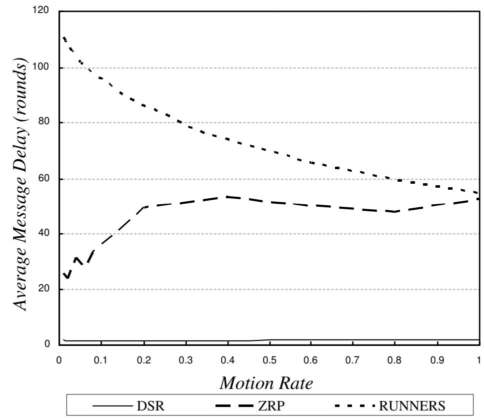

Figure 3: Average Message Delay ($\mathcal{T}$) in grid graphs ($n = 121$, $\lambda = 0.02$, $u = 50$).

while transmissions to destination nodes that are "further away" are carried out by the RUNNERS protocol. Destination nodes are characterized as being "close" if they can be reached by DSR route request packets in up to p hops, where p is a suitably chosen parameter.

When a mobile node has a message to send to some destination, it first consults its route cache to determine whether it already has a route to the destination. If it has a route to the destination, then it will use this route to send the message. If such a route does not exist, then it initiates the route discovery procedure of DSR (see Protocols Section). However, the propagation of the *route request* packet is limited by the maximum hop limit p. Based on the experiments conducted in the Experimental Section, we chose $p = 3$. In case no route is discovered connecting the sender with the receiver, no further *route request* packets will be sent (i.e. the *route request retries* parameter of DSR is set to 1), and the outgoing messages are forwarded to the RUNNERS protocol.

We here note that the synthesis proposed in this section can be appropriately modified to use other proactive protocols, i.e. other than DSR. We are currently studying such kind of variations and plan to further investigate the idea of synthesizing a proactive routing protocol with the RUNNERS protocol.

4.2 The ZRP-Runners Synthesis

The second protocol (ZRP-R) is based on the synthesis of the ZRP and RUNNERS protocols. We take advantage of the modular nature of ZRP and replace the *Interzone routing protocol* (IERP) and the *Bordercast Resolution Protocol* (BRP) by the RUNNERS protocol. Each mobile host that wants to send some information to another host first consults the routing information of its local routing zone by using the *Intrazone Routing Protocol* (IARP). Thus, when data needs to be sent to a destination-member of its routing zone, the necessary routing information is instantly available. If no such routing information is available, the outgoing messages are forwarded directly to the RUNNERS protocol.

5. SIMULATION STUDY AND RESULTS

All of our implementations follow closely the protocols described above. They have been implemented as C++ classes

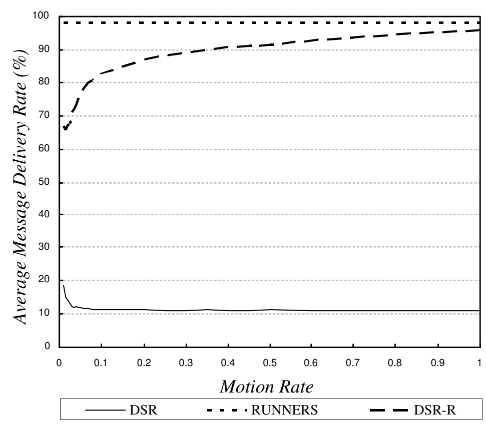

Figure 4: Average Message Delivery Rate ($\mathcal{D}$) in grid graphs ($n = 121$, $\lambda = 0.02$, $u = 50$).

using several advanced data types and structures of LEDA [14]. We used the simulation environment developed in [5, 4] (*adhocsim*) where each class is installed in an environment that allows to read graphs from files as well as to perform a network simulation for a given number of rounds, a fixed number of mobile hosts and certain communication and mobility patterns. After the execution of the simulation, the environment stores the results again on files so that the measurements can be represented in a graphical way. We chose the lightweight LEDA environment to be able to study very large instance sizes. We plan to study the behavior of the protocols using a detailed network simulator that more accurately takes into account the network parameters.

In the DSR implementation the *number of the route request retries* and the *number of the data packet send retries* between two neighboring nodes were set to 2. Also, we did not limit the propagation of the route requests in the network, by placing a hop limit.

In the ZRP implementation the *routing zone radius* was set to 3, based on the experiments presented in [17]. The number of the *route request retries* was set to 2.

A basic parameter that affects the performance of RUN-

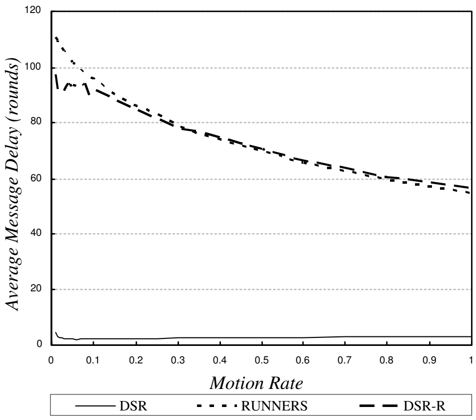

Figure 5: Average Message Delay ($\mathcal{T}$) in grid graphs ($n = 121,\ \lambda = 0.02,\ u = 50$).

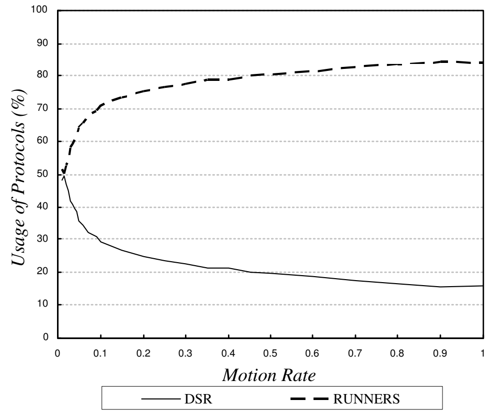

Figure 6: Average usage of each protocol in DSR-R based on delivered messages in grid graphs ($n = 121$, $\lambda = 0.02,\ u = 50$).

NERS is the size k of the support. The proper selection of a value for k is studied thoroughly in [4]. Based on that work, the values that we selected for the size of the support are $\{10, 15, 20\}$ for motion graph sizes of $\{121, 529, 1024\}$ respectively. In addition, the mobile hosts that belong to the support have motion rate $\mu = 1$, i.e. they move continuously in the area covered by the network.

In the implementation of the synthesized protocols DSR-R and ZRP-R, we simply used the above implementations of their constituent protocols.

In order to evaluate the performance of the protocols we investigated two crucial parameters:

Definition 2 (Message Delivery Rate). *Let M_{total} be the total number of messages send out and M_{succ} the number of messages that were successfully delivered to their destination. Let $\mathcal{D} = \frac{M_{succ}}{M_{total}}$ be the* delivery ratee *of a communication protocol.*

Definition 3 (Message Delay). *Let $\mathcal{T}$ be the* total time *required for a* message *to reach its final destination.*

A large number of experiments were carried out modelling the different possible situations regarding the geographic area covered by an ad-hoc mobile network and the motion of hosts within it. Each experiment proceeds in lockstep rounds called *simulation rounds* (i.e. we measure simulation time in rounds). Each mobile host h is considered identical in computing and communication capability and resides initially in a particular vertex v_h of the motion graph $\mathcal{G}$. Two mobile hosts i, j communicate by sending messages if they reside on the same vertex of the motion graph (i.e. $v_i = v_j$) or when they are within transmission range of each other (i.e. $d(v_i, v_j) \leq \beta$). We assume that in this case message exchange takes negligible time (i.e. the messages are short packets and the wireless transmission is very fast).

In a simulation round each mobile host h may move to a vertex of the motion graph $\mathcal{G}$ and performs some computation. We assume that a mobile host can receive several messages and send several messages at the same round. If $h \in \Sigma$, its move and local computation is determined by the RUNNERS protocol. If $h \notin \Sigma$, then with probability μ it moves to a vertex of the motion graph $\mathcal{G}$ (e.g. if $\mu = 0.05$ it moves roughly every 20 rounds) and with probability λ the host h generates a new message (e.g. if $\lambda = 0.01$ a new message is generated roughly every 100 rounds) by picking a random destination host. Remark that there is no limit on the total number of sender-receiver pairs.

For each motion graph $\mathcal{G}$ considered, we used various numbers of mobile hosts $u \in [10, 100]$, which were injected at random positions of $\mathcal{G}$ and generated message exchanges with various rates $\lambda \in [0.05, 1]$. The selection of this range for λ is based on the assumption that the mobile hosts execute a delay-tolerant (in the order of fractions of seconds to several seconds) application that are asynchronous in nature. Examples of such applications include electronic mail, database synchronization between a mobile terminal and a central database, and certain types of event notification (similar assumptions are made in [6]).

Finally, regarding motion graphs, we considered 2D grid graphs to model different geographic areas. This is the simplest pattern of motion one can consider (e.g. mobile hosts that move on a plane surface), and (as our previous studies showed [5, 4]) they are quite representative with respect to protocol behavior for the case of 3D grid motion graphs. We used three different $\sqrt{n} \times \sqrt{n}$ grid graphs with $n \in \{121, 529, 1024\}$. Note that for a transmission range of 250m, the first grid covers a 2750m × 2750m area, the second grid covers a 5750m × 5750m area, and the third grid covers a 8Km × 8Km area.

We carried out each experiment for a very large number of rounds (at least $10,000$) during which a very large number of messages (at least $200,000$) were delivered to the designated receivers. We here report the experimental results for 2D grid motion graphs of size $n = 121$, where the number of mobile hosts u is 50 and the message generation rate λ to 0.02 (similar results hold for the other motion graphs and network parameters considered).

We first evaluate the effect of mobility on the efficiency of the three protocols presented in Sec. 3. We measured the average message delivery rate and the average message delay for all values of the motion rate $\mu \in [0.01, 1]$ of the mobile hosts. The results are reported in Fig. 2, 3.

For the average delivery rate (see Fig. 2), we observe that

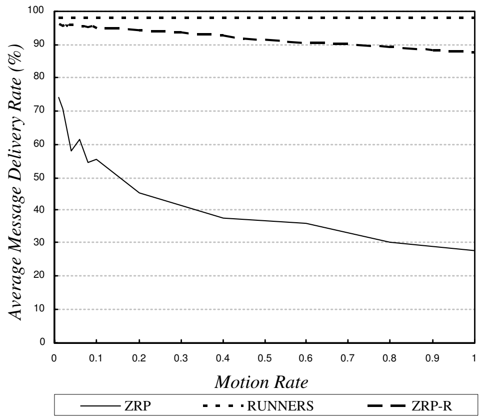

Figure 7: Average Message Delivery Rate ($\mathcal{D}$) in grid graphs ($n = 121$, $\lambda = 0.02$, $u = 50$).

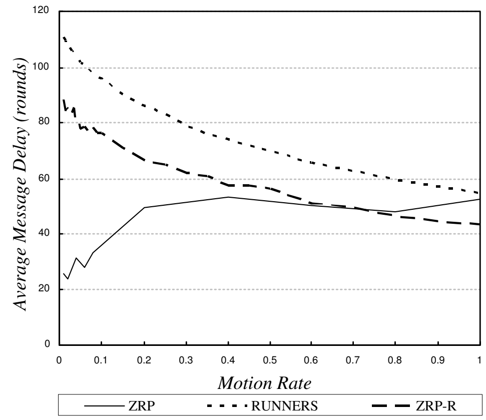

Figure 8: Average Message Delay ($\mathcal{T}$) in grid graphs ($n = 121$, $\lambda = 0.02$, $u = 50$).

the motion rate of the users has a great impact on the performance of ZRP, while it seems to have no effect on the DSR and RUNNERS protocols. In fact, DSR achieves very low delivery rates, about 20%, regardless of the motion rate of the users. On the other hand, ZRP delivers more than 50% of the messages for low rates of motion (e.g. $\mu \leq 0.2$), while for higher rates (e.g. $\mu \geq 0.6$) the delivery rate drops below 35%. The RUNNERS protocol exhibits a completely different behavior. It achieves a very high delivery rate for all "frequencies" of motion.

Fig. 3 depicts the effect of the motion rate on the average message delay. We observe that DSR is much more efficient, with respect to this metric, than the other two protocols. This, however, should be evaluated in combination with the results of Fig. 2, where DSR exhibits low delivery rates for all types of motion. We here measure delays for delivered messages only. The performance of ZRP degrades as the rate of motion increases, i.e. the average message delay increases as the users of network move more frequently. Interestingly enough, the RUNNERS protocol has again a completely different behavior; that is, the average message delay drops as the rate of motion of the hosts increases. This can be explained by the fact that as the users are moving more frequently, they will meet the nodes of the support more often. In this way, the messages will be delivered to the support and to their final destination faster. Thus, the RUNNERS protocol is particularly suitable in the case of high mobility rate, since increasing motion seems to positively affect its performance by accelerating meeting times.

The experiments point out the different behavior of the three protocols which can be summarized as follows:

(i) When the path construction mechanisms of DSR succeeds in finding a route connecting the sender-receiver pair, the messages are delivered within extremely short periods of time (see Fig. 3). However, even for almost static networks (i.e. very small values of μ), the success rate of DSR is very low (i.e. $\mathcal{D} \leq 20\%$).

(ii) ZRP establishes communication with higher success rates (i.e. $\mathcal{D} \geq 60\%$) when all mobile hosts are static or quasi-static (e.g. $\mu < 0.1$) and the messages are delivered with small delays. On the other hand, as the mobile hosts move more frequently, the protocol fails to react fast enough to the topological changes, leading to situations where messages are lost and/or delivered messages experience long delays.

(iii) RUNNERS almost always succeed in delivering the messages to their destination (i.e. $\mathcal{D} \geq 98\%$) regardless of the motion rate of the hosts. However, the messages are delivered with longer delays (on the average) than the other two protocols (see Fig. 3), while the delays are reduced as the hosts are moving more frequently (i.e. as $\mu \to 1$).

It is evident from the above experimental outcomes that the ability of the two protocols (DSR, ZRP), which follow the path construction approach, to successfully deliver messages to their final destination depends heavily on the mobility rate of the hosts. The internal mechanisms of the two protocols that maintain the constructed paths (that connect pairs of hosts) fail to react fast enough to the topological changes that result from the frequent motion of the users. Clearly, none of the two protocols considered here can guarantee delivery as the mobility rate of the users of the network gets high. In this sense, these experimental outcomes validate (for the first time) the conjecture stated in [5] according to which in cases of high mobility rate, any algorithm that tries to maintain a global structure with respect to the temporary network will even become erroneous if the mobility rate is faster than the rate of updates of the algorithm.

In the next set of experiments we investigated the performance of the DSR-R and ZRP-R protocols. Based on Fig. 4, 5, we first observe that the average message delay ($\mathcal{T}$) of the synthesized protocol is almost identical to that of RUNNERS, implying that the majority of the messages are delivered by the support Σ of the network. This is observation is also supported by Fig. 6 that depicts the usage of each of DSR and RUNNERS protocols in DSR-R based on the messages delivered. It is clear that most of the times, RUNNERS dominate DSR for the whole range of motion rates considered. In fact, as the usage of RUNNERS increases, the average message delivery rate ($\mathcal{D}$) of the synthesized protocol asymptotically reaches the corresponding curve of RUNNERS (see Fig. 4). In conclusion, the results of this set

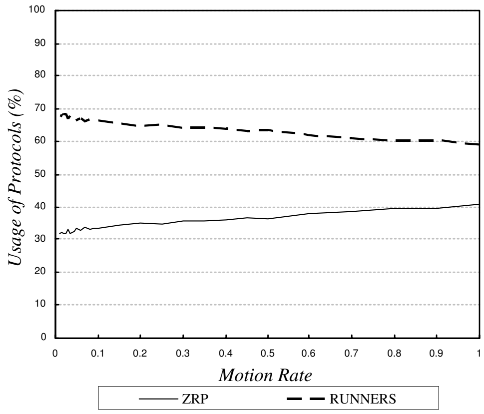

Figure 9: Average usage of each protocol in ZRP-R based on delivered messages in grid graphs ($n = 121$, $\lambda = 0.02$, $u = 50$).

of experiments indicate that the synthesis of the DSR and RUNNERS protocols does not seem promising. Even in the case of very static networks, DSR delivers a small portion of the messages which is not visible in the results.

We now turn to the performance evaluation of ZRP-R. The results of our experiments are reported in Fig. 7, 8, 9. We observe that the average message delivery rate of ZRP-R is very close to that of RUNNERS but for the average message delay, ZRP-R outperforms ZRP in cases of high mobility rates ($\mu \geq 0.65$) and it outperforms RUNNERS for all values of μ. In contrast to DSR-R, in this synthesized protocol ZRP and RUNNERS are used interchangeably for all values of the motion rates considered, as it is shown in Fig. 9. We believe that the modular nature of ZRP allows such high cohesion, resulting in this way to a protocol synthesis that, in some cases, outperforms both original ones.

6. CLOSING REMARKS

Our detailed investigation highlights the advantages and disadvantages of each approach and its suitability for a certain network and user profile. In particular, it seems that the protocols that follow the path construction approach behave well in low mobility rates, while their performance degrades in the case of highly mobile users. On the other hand, the support approach seems to behave well in the case of high mobility rates (and always succeeds in delivering messages).

Our results suggest that protocol design should carefully take into consideration such vital parameters. It seems that (at least under the model we study) none of the original three protocols is suitable for all network cases. Based on this observation, we proposed a new method for synthesizing protocols that follow different approaches in order to overcome such problems. Our method gives good results and in fact in some cases the resulting synthesized protocol outperforms the original ones by achieving lower message delivery delays while keeping high success rates.

We believe that the method of synthesizing different routing protocols is a step forward in creating a unifying framework for ad-hoc mobile networks comprised of mobile hosts with heterogeneous mobility rates.

7. REFERENCES

[1] M. Adler and C. Scheideler. Efficient communication strategies for ad-hoc wireless networks. In *10th Annual Symposium on Parallel Algorithms and Architectures*, pages 259–268. ACM, 1998.

[2] J. Broch, D. B. Johnson, and D. A. Maltz. The dynamic source routing protocol for mobile ad-hoc networks. Technical report, IETF, Internet Draft, December 1998. draft-ietf-manet-dsr-01. txt.

[3] I. Chatzigiannakis and S. Nikoletseas. Design and analysis of an efficient communication strategy for hierarchical and highly changing ad-hoc mobile networks. *Journal of Mobile Networks and Applications*, 9:319–332, 2004.

[4] I. Chatzigiannakis, S. Nikoletseas, N. Paspalis, P. Spirakis, and C. Zaroliagis. An experimental study of basic communication protocols in ad-hoc mobile networks. In *5th International Workshop on Algorithmic Engineering (WAE 2001)*, pages 159–171. Springer-Verlag, 2001. Lecture Notes in Computer Science, LNCS 2141.

[5] I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis. Distributed communication algorithms for ad hoc mobile networks. *Journal of Parallel and Distributed Computing (JPDC)*, 63(1):58–74, January 2003. Special Issue on Wireless and Mobile Ad-hoc Networking and Computing, edited by A. Boukerche.

[6] M. Grossglauser and D. Tse. Mobility increases the capacity of wireless adhoc networks. In *20th Annual IEEE International Conference on Computer Communications and Networking (INFOCOM 2001)*, pages 289–299. IEEE, 2001.

[7] Z. J. Haas and M. R. Pearlman. The performance of a new routing protocol for the reconfigurable wireless networks. In *International Conference on Communications*. IEEE, 1998.

[8] Z. J. Haas and M. R. Pearlman. The zone routing protocol (ZRP) for ad-hoc networks. Technical report, IETF, Internet Draft, June 1999. draft-zone-routing-protocol-02. txt.

[9] G. Holland and N. Vaidya. Analysis of TCP performance over mobile ad hoc networks. In *5th ACM/IEEE Annual International Conference on Mobile Computing (MOBICOM 1999)*, pages 219–230, 1999.

[10] T. Imielinski and H. F. Korth. *Mobile Computing*. Kluwer Academic Publishers, 1996.

[11] Y. Ko and N. H. Vaidya. Location-aided routing (LAR) in mobile ad-hoc networks. In *4th ACM/IEEE Annual International Conference on Mobile Computing (MOBICOM 1998)*, pages 66–75, 1998.

[12] Q. Li and D. Rus. Sending messages to mobile users in disconnected ad-hoc wireless networks. In *6th ACM/IEEE Annual International Conference on Mobile Computing (MOBICOM 2000)*, pages 44–55, 2000.

[13] N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers Inc, 1996.

[14] K. Mehlhorn and S. Näher. *LEDA: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, 1999.

[15] V. Park and M. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *16th Annual IEEE International Conference on Computer Communications and Networking (INFOCOM 1997)*, pages 1405–1413. IEEE, 1997.

[16] V. D. Park and M. S. Corson. Temporally-ordered routing algorithms (TORA) version 1 functional specification. Technical report, IETF, Internet Draft, October 1999. draft-ietf-manet-tora-spec-02. txt.

[17] M. Pearlman and Z. Haas. Determining the optimal configuration for the zone routing protocol. *IEEE Journal on Selected Areas in Communications*, 17(8):1395–1414, January 2003.

[18] C. E. Perkins. *Ad Hoc Networking*. Addison-Wesley, Boston, 2001.

[19] C. E. Perkins and E. M. Royer. Ad-hoc on demand distance vector (AODV) routing. In *2nd IEEE Annual Workshop on Mobile Computing Systems and Applications*, pages 90–100, 1999.

[20] J. E. Walter, J. L. Welch, and N. M. Amato. Distributed reconfiguration of metamorphic robot chains. In *19th ACM Annual Symposium on Principles of Distributed Computing (PODC 2000)*, pages 171–180, 2000.