Contact-Extended Zone-Based Transactions Routing for Energy-Constrained Wireless *Ad Hoc* Networks

Ahmed Helmy, *Member, IEEE*

***Abstract*—In future wireless *ad hoc* networks, transactions are expected to become one of the primary types of flows. Transactions require only a small number of packets to complete. Hence, providing optimal (shortest path) routes to such transactions consumes more energy than the actual data transfer. Conventional shortest path routing protocols are, thus, unsuitable for routing transactions. In this paper, we present a novel architecture, called TRANSFER, for transactions routing in large-scale wireless *ad hoc* networks. In our approach, we aim to reduce the total energy consumption by transactions as opposed to finding shortest path routes. Our architecture uses a hybrid approach, in which each mobile node obtains information about nodes in its proximity (zone), up to R hops away, using a proactive link-state protocol. Beyond the proximity, we introduce the novel notion of contacts that act as shortcuts to reduce the degrees of separation between the source of the transaction and the destination. We propose an efficient on-demand protocol for contact selection that does not assume knowledge of location information. Contacts are used during transactions and queries to discover valid routes in an energy-efficient manner. Extensive simulations are used to evaluate the performance of our protocol in terms of energy consumption and success rate. We compare our architecture to flooding, dynamic source routing (DSR), zone-routing protocol (ZRP), and two power-aware schemes. Our results show substantial power savings for our contact-based protocol, especially for large *ad hoc* networks.**

***Index Terms*—*Ad hoc* networks, contact-based architecture, energy efficiency, network simulation, routing protocol.**

I. INTRODUCTION

AD HOC wireless networks are expected to have a significant impact and have the potential for many applications. Many such applications are envisioned to include a large number of small transactions, such as text messaging, paging, resource discovery, query resolution, and sensing. In such applications, only a few packets are required to complete the transaction. Nodes in *ad hoc* networks are power constrained with communication being a major power consumer. Also, mobility creates a dynamic environment in which routes frequently become invalid. Hence, it becomes very inefficient to discover optimal routes for small transactions, since the cost of such discovery is quite high, often exceeding that of the actual data transfer. Conventional routing protocols strive to provide shortest path routes. For example, in some cases, a routing protocol may use flooding to explore all possible routes, then chooses the optimal (or shortest) route. Instead of searching for optimal routes, we take an approach to tradeoff route optimality for lower energy consumption. Our architecture searches for *a route* (even if sometimes suboptimal) by querying only a selected subset of the nodes (called *contacts*) to reduce the overall amount of energy consumed by small transactions. Based on this concept, we introduce a new routing protocol geared toward small transactions in mobile wireless *ad hoc* networks. We design our protocol to be scalable, power efficient, mobility adaptive, and self-configuring. We call our protocol "transactions routing for *ad hoc* networks with efficient energy" or *TRANSFER*.

This work presents one of the first routing protocols geared toward small transactions in *ad hoc* networks. We avoid the use of flooding or complex coordination mechanisms in our approach. In our architecture, every node independently collects information from neighboring nodes up to R hops away. This is called a node's *proximity* or *zone*. We introduce the concept of *contacts* as key for the efficient transfers in our scheme. For a node, contacts are a few nodes outside of its proximity that act as *shortcuts* to transform the wireless network into a small world and, hence, reduce the average degrees of separation between the source and destination or the querier and the target. When a query[1] is made, the contact-selection protocol is invoked. Contact selection employs a mechanism to reduce proximity overlap and to elect contacts that increase the coverage of the search. Salient features of our architecture include its ability to select useful contacts on the fly (i.e., on demand) and the ability to balance the energy-consumption among the network nodes. Also, as we will show, our protocol exhibits very good performance over a wide range of networks.

We use extensive simulations to evaluate the performance of our protocols in terms of energy consumption, success rate, and average delay (or number of attempts). We compare our protocols to flooding on-demand *ad hoc* routing with caching (DSR), power-aware broadcast (or smart flooding), minimum dominating set (cluster) schemes, and zone-routing protocol (ZRP) over a variety of networks. Our results show significant overall energy savings for our technique. For large networks and high query rates, TRANSFER consumes as little as 5% of the flooding energy and 14% of the ZRP energy consumption.

The rest of this paper is organized as follows. Section II provides an architectural overview and Section III presents the contact-selection protocol and the search policy. Section IV provides the query-processing rules. Section V provides the evaluation results and Section VI analyzes the performance of on-demand *ad hoc* routing for small transfers. Section VII discusses related work and Section VIII concludes this paper.

Manuscript received March 15, 2003; revised September 2, 2003, May 2, 2004, August 13, 2004, and September 17, 2004. This work was supported in part by the National Science Foundation CAREER Award, Intel, and the Pratt & Whitney Institute for Collaborative Engineering (PWICE).

The author is with the Electrical Engineering Department, University of Southern California, Los Angeles, CA 90089 USA (e-mail: helmy@usc.edu).

Digital Object Identifier 10.1109/TVT.2004.839671

---

[1] We use the terms *query* and *request* to refer to a small transaction.

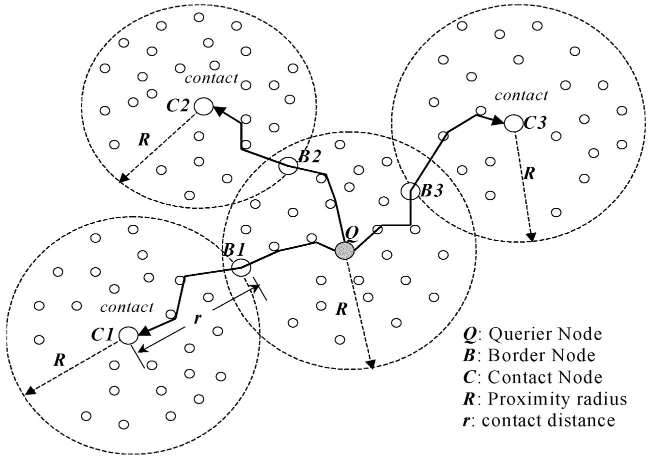

Fig. 1. Each node in the network has a proximity of radius R hops. A querier node Q sends a query through a number of its borders equal to the number of contacts (NoC); in this case, NoC $= 3$. Each border node Bi chooses one of its borders Ci to be the direction for forwarding the query r hops further until it reaches the contact. The contacts are at most $(R + r)$ hops away from Q. In this example, $r = R = 3$.

II. ARCHITECTURAL OVERVIEW

In our architecture, each node in the *ad hoc* network keeps track of a number of nodes in its vicinity within R hops away. This defines the *proximity* of a node. The proximity is maintained using a proactive localized link-state broadcast. Each node chooses its proximity independently and, hence ,no major reconfiguration is needed when a node moves or fails. There is no notion of cluster head and no elections that require consensus among nodes. We assume the existence of a neighbor discovery protocol by which each node identifies nodes one hop away (through periodic beacons). The link-state protocol provides neighbor information to other nodes in the proximity. Typically, the number of nodes in the proximity is relatively small (e.g., 1%–10% of the network nodes). As part of the proximity information, each node keeps routes to nodes in its proximity, including nodes exactly R hops away (we refer to those nodes as "border nodes" or simply "borders").

When a querier node Q issues a transaction or query, it first checks to see if the destination is in its proximity. If not, then it seeks the assistance of a number of contacts (NoC) outside its proximity, as follows. First, a query is issued to NoC (say 3) of Q's borders (R hops away). Each border B receiving the query would in turn select another node C at r hops away to which to forward the query. We call C a *contact* node for Q and r is the contact distance. To increase search efficiency, C should have low proximity overlap with Q's proximity. Proper setting of the parameter r helps to reduce such overlap. Contact nodes act as shortcuts that bridge between disjoint proximities. This helps to reduce the degrees of separation between Q and the target nodes. Degrees of separation in this context refer to the number of intermediate nodes to get from the querier node Q to the target.

The main architecture is shown in Fig. 1, where the querier node Q (potentially any node in the network) chooses three borders, $B1$, $B2$, $B3$, to which to send a query. The borders in turn choose three contacts at r hops away, $C1$, $C2$, $C3$, to which to forward the query. If $r = R$, then the contact is a border of a border of Q.

Questions regarding setting the design parameters, such as NoC, (r), and (R), will be investigated in the evaluation section. First, we describe our contact=selection scheme.

III. CONTACT SELECTION AND SEARCH POLICY

The main purpose of a contact node is to act as a shortcut to increase the view of the network by searching for the target in uncovered parts of the network. This increases the network coverage per contact query and enhances the search efficiency, effectively reducing the search overhead. Hence, it is important for a contact to have a proximity that does not overlap significantly with that of the querier node Q or the other contacts of Q. This is achieved through the contact-selection protocol, described next.

A. Reduced-Overlap Contact Selection

The first kind of overlap occurs between the contact's and querier's proximities. To attempt to reduce this overlap, we attempt to push the query as far out from the querier's proximity as possible.

A querier node Q sends a query to NoC of its borders. Let one of those borders be B, which has its own proximity of R hops. Using the proximity information (that was formed using the link state exchange), B constructs a topology view up to R hops away and chooses a border in its proximity that has maximum distance to Q. Note that B does not have geographic information about the nodes, but can calculate the relational distance in hops of all the nodes in its proximity to Q based on its proximity information. Also note that it is possible that B thinks that another node, say x, is $2R$ hops away from Q, while in fact x is less than $2R$ hops away using a route outside of B's proximity. In order to reduce this likelihood and to push the query away from Q's proximity, we use the following forwarding algorithm at the border B.

Say that node L was the node forwarding the query to node B. The distance between Q and L is, hence, $R - 1$. Since L is $R - 1$ hops from Q, then any of L's neighbors is at most R hops away from Q. Therefore, B attempts not to forward the query through one of L's neighbors. Instead, B chooses one of its borders C that is $2R$ hops away from Q (from B's perspective) such that (if possible) the route to C does not pass through L or any of its neighbors. Although this algorithm does not guarantee nonoverlap between C's and Q's proximities, its performance was found to be very good during the simulations, as we will show in Section V.

We call this scheme the *proximity-overlap reduction* (POR) scheme. In cases where r is not equal to R, POR is used to select a border for B that provides a *direction* for choosing the contact; in this case, we call B's border the *direction border*. If $r < R$, then POR is performed by B and then the contact is selected between B and its direction border. If $r > R$, then the direction border needs to perform POR again to find its own direction border and so on. POR is performed without incurring any extra communication overhead and, in general, is performed $\lceil r/R \rceil$ times overall by the chosen direction borders.

The second type of overlap occurs between proximities of contacts. To reduce this overlap, the querier node Q attempts to select borders to which it has disjoint routes. This is done using by the proximity information (with no extra overhead). If NoC borders are chosen by the end of this procedure, then Q sends the query to the chosen borders. Otherwise, borders are chosen with minimum route overlap (i.e., with different second-hop nodes, then third-hop nodes, etc.). Otherwise, new borders are chosen randomly until NoC borders are chosen. This scheme does not guarantee nonoverlap between contacts' proximities, but performs quite efficiently during query resolution, as we will show in Section V. We call this scheme the route-overlap reduction (ROR) scheme.

B. Power-Aware Contact Selection

The contact-selection criteria can (and, in fact, should) take *power* into consideration. Information about power levels in nodes and the rate of energy consumption may be piggybacked on the link-state messages exchanged within the proximity. During the contact selection mechanisms (i.e., POR and ROR), nodes with low remaining energy are given low selection priority. To achieve this, we propose a power-aware contact-selection algorithm.

The main idea behind this algorithm is to choose the most energy-capable nodes to which to forward the packets or queries, while still maintaining good search coverage by using POR and ROR. The algorithm aims to achieve power balancing and extend the network lifetime by avoiding nodes with low energy, unless necessary.

This contact-selection algorithm operates as follows. First, an attempt is made to select contacts using POR and ROR such that nodes *en route* to those contacts have power more than a power threshold $Pth1$. If there is no such available route, then the process is repeated with $Pth2 < Pth1$, and so on. To further clarify why this is done, let us define the power of node n_i to be power(n_i) and define the minPower of a route to be the $\min(\text{power}(n_i))$ for all nodes n_i on that route. Then, this algorithm effectively attempts to choose maximum minPower routes. We will show in the evaluation section (Section V) how this algorithm helps in prolonging the lifetime of the network.

C. Levels of Contacts

These contact-selection mechanisms (POR and ROR) are used to select NoC contacts that have distances up to $R+r$ hops away from Q. We call these contacts *level-1* contacts. To select the level-1 contacts Q performs ROR to reach NoC borders, then those borders (and their respective direction borders and so on $\lceil r/R \rceil$ times) perform POR to get the direction for the contacts.

To select further contacts, this process is further repeated as needed at the level-1 contacts, level-2 contacts, and so on, up to a number of levels called *maxDepth*, D. We will study the effect of D in the evaluation section. The only difference between Q selecting the level-1 contacts and level-i contacts selecting level-$i+1$ contacts (where $i > 1$) is that level-i contacts need to perform POR and ROR, while Q performs only ROR. That is, a level-i contact selects borders with disjoint routes from its set of borders that do not pass through its previous hop (L's) neighbors.

D. Search Policy

Given a query and a maximum number of levels D, the target search process uses what we call an exponential step policy (step). In step, the query is sent out in several attempts. The first attempt is performed with level depth of 1. Until and unless the target is found, each subsequent attempt i is performed with level depth $d_i = 2d_{i-1}$. Attempts continue while $d_i \leq D$.

IV. QUERY FORWARDING AND PROCESSING

A. Query Message

The query message contains the target identification (ID), which could be the node ID or the resource key (for resource discovery). For small transfers, the query may also carry the data. The destination ID in the query contains the ID of the border node (or the direction border). The query message also contains the maximum number of levels to visit D for that attempt, the querier ID Q, and a sequence number (SN). For every new attempt, a new SN is issued.

B. Loop Prevention

As the message is forwarded, each node traversed records the SN, Q, and the previous hop node P, from which the query was received. P may be used later to send a response to the querier Q through the reverse path. If a node receives a query with the same SN and Q, it drops the query. This provides for loop prevention and avoidance of revisits to the covered parts of the network. This mechanism is important to keep the overhead from exponentially growing at each level. Also, if a contact at any level exists in the same proximity as the querier, then the contact drops the query, since it must have looped.

C. Search, Processing, and Forwarding

A contact (or border node) receiving the query, first searches in the proximity information. If the target is found, the query is delivered and a response is forwarded on the reverse path, with each node forwarding the response to its recorded previous hop P. Otherwise, further processing is performed as follows. In order for a recipient of a query to determine that functions to perform, and whether it is a contact, two fields are included in the query message: a *level count* and a *hop count* for each level. Initially, the level count is set to D and the hop count set to $(R+r)$. The hop count is decremented with every hop and is checked as follows.

- If the hop count reaches 0, the receiving node acts as a contact. A contact decrements the level count and resets the hop-count field to $(R+r)$. If the level count reaches 0, the contact drops the query. If the level count is not 0, the contact selects NoC borders (using POR and ROR, as in Section III) and sends the query to those borders.
- If the hop count is not 0 and the current node is the destination of the query message, the receiving node acts as a border node. It selects a direction border (using POR, as in Section III) and sends the query to it.

TABLE I
Networks Used in the Simulation. Nodes Are Initially Randomly Distributed. Number of Border and Proximity Nodes Given for $R = 3$

Nodes	*Area (mxm)*	*Node Degree*	*Border Nodes*	*Proximity Nodes*
200	1000x1000	7.6	15.1	35
500	1400x1400	8.9	20.5	44.8
1000	2000x2000	9.1	21.7	46.8
2000	2800x2800	9.7	24.7	52.9
4000	3700x3700	11	30.3	62.2
8000	4800x4800	13	38.8	77.8
16000	6500x6500	14.3	44.6	88.2
32000	9200x9200	14.3	45	88.9

- Otherwise, the query is simply forwarded to the next hop toward the destination.

Note that the query message is *unicast* hop by hop; it is *not* broadcast hop by hop.

V. EVALUATION AND COMPARISON

In this section, we study the various dimensions of the design space for our architecture. In addition, we compare our protocols to other related approaches, including flooding and ZRP (as proposed in [13] and [14]).

Particularly, we attempt to study the effect of changing the NoC, contact distance (r), maxdepth (D), and proximity radius R on the protocol performance. The main performance metrics include energy consumption and the query-success rate. Note the tradeoff between success rate and overhead; the more the success rate, the more the overhead and *vice versa*. In order to balance these conflicting goals, we introduce a penalty for query failures. Any failure beyond an acceptable level (set to 2% in our simulations) is recovered using flooding. Hence, *the scheme used in our simulations is contact-based search; if it failed then fallback to flooding*. Since this penalty is quite expensive, it will be natural for our best performing parameters to avoid resorting to flooding by achieving a very high query-success rate.

A. Simulation Setup

We use extensive simulations to investigate the design parameters and evaluate the performance of our protocol. We put a limit of 100 nodes per proximity for our study and choose $R = 3$. Transmission radio range (tr) is taken as 110 m. We study a wide range of network sizes, as shown in Table I. We also vary the area of the network to maintain connectivity. N nodes are randomly placed in a ℓm × ℓm' square topology.

For mobility, we use the random way-point model [35], where for each node a destination is chosen randomly and a velocity is chosen randomly from $[0, V\max]$. Once the destination is reached, another random destination and velocity are chosen, and so on. For our simulations, we very $V\max$ from 0 to 60 m/s and use various request/query rates varying from 0.01 query per kilometer up to 1000 queries per kilometer. We developed a discrete event simulator for the protocols under study. For the purposes of our simulation, we do not implement muliple-access control (MAC) layer collisions. We implement a hop-by-hop energy model, taking into account the energy consumption due to transmitted and received packets, described next.

Hop-by-Hop Communication Energy Model: The energy consumed when a query is sent at each hop is due to packet transmission at the sender and packet reception at the recipient(s). Depending on the mode of the message, whether unicast, multicast, or broadcast, the number of actual recipients vary. By recipients, we do not mean only the intended recipients, but also other nodes (within the transmission range) that are in the *receive* state. In general, a wireless node may be in one of three power states: 1) transmit state; 2) receive state; or3) idle/sleep state. The power expended in each of these states may vary drastically. Also, the overall power consumed is a function of the duration of stay in any of these states (mainly a function of the packet size). We refer to the amount of energy consumed during the transmission of a query packet as Etx. Similarly, Erx refers to the energy of query reception. If a message is broadcast, it is received by all other nodes within radio range, i.e., all neighbors. The average number of neighbors per node is known as the average node degree g. For a unicast message, there usually is a small *handshake* phase to inform the neighbors of the impending transmission. In IEEE 802.11 (the model we adopt), the CSMA/CA algorithm is used with handshake and medium reservation. The handshake involves the broadcast of a small message, request-to-send (RTS), to which the intended recipient responds with a broadcast of a small clear-to-send (CTS) message. This RTS/CTS exchange causes the neighbors to transit into the idle/sleep state until the end of query transmission. We refer to the power consumption due to handshake as Eh. Based on this understanding, we use the following energy model.

- Energy consumed by a unicast message (Eu)

  $$Eu = Etx + Erx + Eh = Etx(1 + f + h)$$

  where $f = Erx/Etx$ and $h = Eh/Etx$.
- Energy consumed by a broadcast message (Eb)

  $$Eb = Etx + g.Erx = Etx(1 + f.g)$$

  where g is the average node degree.

For this study, we use $f = 0.64$ and $h = 0.1$.[2] Hence, the simulator differentiates between (hop-by-hop) unicast and broadcast messages and applies the energy model accordingly. To have the results be independent of the packet size used, we record the energy measure in Etx units. The total energy consumption in the network per query (or simply "energy per query") for our protocol is called E_{step}. This is an average measure of the sum of all the energy consumed (in both transmission and reception) from all the nodes in the network, per query.

The simulation experiments were repeated to filter out outliers. Each data point represents an average of ten simulation runs with different random seeds. Low variability between runs was observed. Querier-target pairs were chosen randomly. One thousand such queries were performed in each run; i.e., a total of 10 000 queries for each data point.

---

[2] The power-consumption numbers were based on reasonable averages of data from Lucent, Cisco, and 3Com 802.11b wireless cards. For the unicast case, a short RTS/CTS handshake reserves the channel for data transmission and other nodes within radio range back off for the duration of the transmission and go to sleep/idle mode. The handshake consumes a small fraction h of the actual transmission energy Etx. This fraction depends on the transmitted packet size. A reasonable (on the high side) estimate of h is $\sim 10\%$.

Fig. 2. Effect of contact distance r.

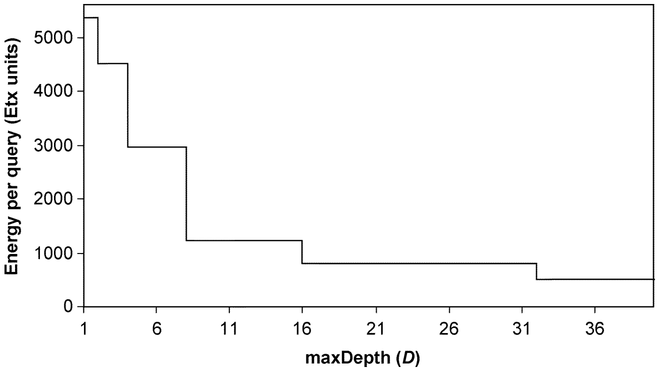

Fig. 4. Effect of maximum depth D.

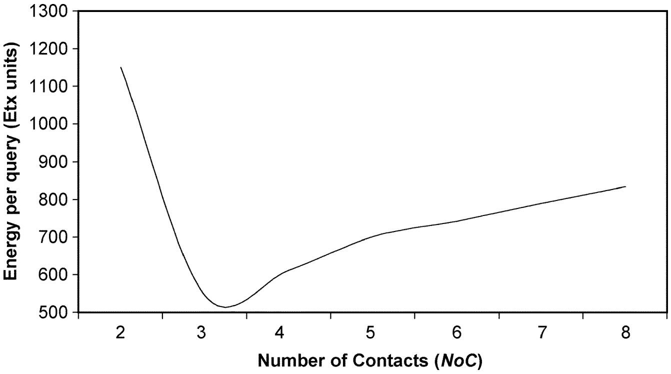

Fig. 3. Effect of NoC.

We first present the overhead per query, then the proximity overhead, and, finally, the overall overhead.

B. Energy Overhead per Query (E_{step})

The 1000-node topology in Table I was used for this set of simulations. The overall network energy consumption was measured (in Etx units) for various values of r, NoC, and D.

Effect of Contact Distance r: We have conducted several experiments with various NoC and D. We only show partial results that represent the trend, using NoC$=$ 3 and $D = 33$. Fig. 2 shows the effect of varying r and clearly indicates favorable settings to minimize the energy consumption. In general, as r grows, the contacts' location extends farther away from the querier's proximity. At very low values of r (e.g., $r \leq 2$), added overhead is incurred due to proximity overlap between Q and level-1 contacts (and, in general, between level-i contacts and their respective level-$i+1$ contacts). This overlap reduces with increase in r, with $r = 3$ being the best value.

A further increase in r generally leads to more energy consumption due to a drop in the success rate, since a failed search falls back into flooding and consumes a lot of energy.

Effect of NoC: Results in Fig. 3 are shown for $r = 3$. A very low number of contacts (NoC$<$ 3) incurs high energy consumption due to fallback to flooding because of a low success rate. Increasing NoC increases the success rate until almost all queries succeed; then we see an increase in overhead due to additional (unnecessary) search branches with further increase in NoC. The best setting is at NoC$=$ 3.

Effect of Maximum Depth D: Using favorable settings for r and NoC, we investigate the effect of increasing the maximum contact depth D. Results in Fig. 4 show that increasing D generally decreases the power consumption. This is due to the fact that increasing D allows the contact-based approach to query a larger number of contacts and, hence, gives a greater chance of success without falling back to flooding (in the case of query failures). By increasing the success rate and subsequently reducing fallback to flooding, the overall energy consumption is reduced. It is *not* the case that increasing D exponentially increases the number of contacts visited. It is true that the number of contacts that may be potentially visited is high but, due to the loop and revisit-prevention mechanisms, this number is drastically reduced. After $D = 33$ (i.e., six attempts), most queries (97.5% or more) become successful and energy consumption almost saturates.

C. Analysis of the Total Overhead and Scalability

In this section, we evaluate the scalability characteristics of our TRANSFER protocol, which consists of two main overhead components of: query overhead and proximity maintenance. In the previous section, we studied the overhead per query. The overall query overhead is a function of the overall number of queries, which in turn is a function of the query rate (query per second) per node, the number of nodes, and the simulation time. Proximity overhead, on the other hand, is a function of the degree of mobility (m/s), the number of nodes in a proximity, the number of proximities (or nodes) in the network, and the simulation duration.

In order to be able to combine these two overhead components in a meaningful way, we represent the query rate as a function of mobility. We also normalize all the measures per node per second per m/s of mobility. We use a metric called query-mobility ratio (QMR) or q, defined per node as query/s/(m/s) or simply query/km. QMR borrows from the call-mobility ratio (CMR) metric introduced by ZRP [13], [14]. Let us call the proximity overhead $Z(R)$, defined in terms of Etx energy units and is a function of the proximity radius R. $Z(R)$ has units of energy (Etx) per second per node per (m/s). Also, let us call the energy consumption per query for the TRANSFER protocol with step policy E_{step}. Similarly, for flooding and ZRP, we have E_{flood} and E_{ZRP}. The units for E_{step} are given in Etx units per query. The overall query overhead $E_{Q\text{step}} = q.E_{\text{step}}$. The units of $E_{Q\text{step}}$ are in Etx units per second per node per m/s, compatible to $Z(R)$. The total overhead becomes $E_{T\text{step}} = Z(R) + E_{Q\text{step}}$.

Our goal in this section is to obtain trends and comparisons of total overhead for the TRANSFER protocol as well as related schemes, for a wide range of query rates and over various networks (200 to 32 000 nodes; see Table I).

Comparison to Related Schemes (Query Overhead): We compare our protocol to several other related schemes, including flooding, ZRP, reduced broadcast (or smart flooding), minimum dominating sets (or clusters), and on-demand routing with caching. Some of these schemes, including minimum dominating sets and reduced broadcasting schemes, were designed to reduce the overhead of discovery and effectively extend the lifetime of the networks. In the following, we provide an overview of the related schemes used in our comparisons.

Related Schemes: For flooding, in a network of N nodes, the query is transmitted by $N-1$ nodes. We get $E_{\text{flood}} = (N-1).Etx(1+f.g) \approx Etx(N+2Lf)$ for large N, where L is the number of links in the network, g is the node degree, and $g = 2L/N$ (by definition).

In ZRP [13], [14], the querier sends the query to its zone borders and the borders send it to their borders, and so on. Query messages are broadcast (or multicast) hop by hop and nodes along the path record the query information. Queries that are sent to previously visited borders are terminated using query detection and control mechanisms ($QD-1$, $QD-2$, and early termination, as described in [13] ,and [14]). For a zone of radius R, each node keeps track of nodes up to $2R-1$ hops.

Reduced-broadcast techniques [24] use heuristics to reduce the redundancies of flooding and to conserve communication. The main idea is to exploit node density to reduce redundant transmissions, sometimes at the expense of reducing the network coverage. The heuristics are used to estimate when message rebroadcasts are likely to be effective. These schemes may be quite effective when the amount of broadcast redundancy is high due to high node density. In situations in which the wireless network is not highly dense, the effect of reduced broadcast is quite limited. Some broadcast reduction heuristics include: 1) probabilistic flooding; 2) counter-based scheme; 3) distance-based scheme; 4) location-based scheme.

In probabilistic flooding, each node rebroadcasts the message with probability p. When $p = 1$, the scheme degenerates to flooding.

The counter-based scheme takes advantage of the following observation; a node may receive the flooded message x times and the additional network coverage obtained by rebroadcasting a message decreases with increase in x. Hence, after receiving a number of messages, it may be desirable for a node to suppress its message rebroadcast if the expected added coverage is very low. Therefore, in the counter-based scheme, a node maintains a message counter x and a counter threshold C_{th}. Upon first receipt of the message, a node waits a random time during which it counts other receptions; when $x > C_{\text{th}}$, then the rebroadcast is suppressed.

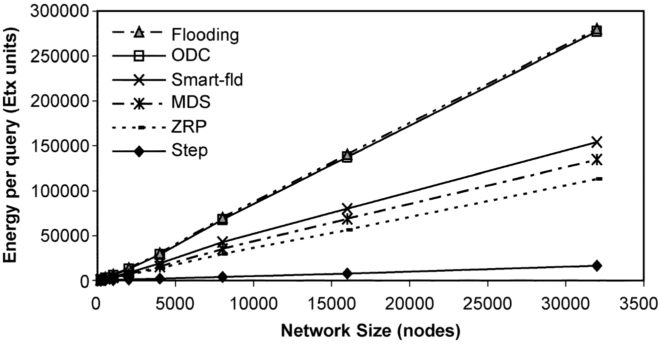

Fig. 5. Energy consumed due to query overhead using various approaches.

The distance-based scheme also attempts to suppress rebroadcasts that are expected to achieve very low coverage, but it performs its calculations based on distance. If a message is received from a nearby node, then there is a low added coverage achieved by the rebroadcast. In this scheme, a node waits a random time and maintains the least distance (d) to the nodes from which it received messages. Each node also has a distance threshold set at D_{th}. If $d < D_{\text{th}}$, then the rebroadcast is suppressed.

A related scheme is the location-based scheme. Knowing the locations of the sending nodes enables the receiving node to estimate the added area coverage a. If a is greater than a certain area threshold (A_{th}), then rebroadcast; otherwise, suppress.

The location-based scheme is the most effective among these heuristics, since many redundant rebroadcasts maybe suppressed without significantly affecting message network coverage. It also works well for various node distributions, but requires location information. The counter-based scheme is simple (it does not require location information) and achieves noticeable reduction in a number of redundant rebroadcasts if the network is dense (where the broadcast redundancy is high), but is not as effective as the location-based scheme. Later in this section, we compare our TRANSFER scheme to the location-based scheme.

Several well-known approaches for power-aware discovery are based on dominating sets (DS) [38]–[41]. A DS of nodes in a network is a subset of the nodes in the network such that each node is either in that set or is a neighbor of a node in that set. The problem of finding the minimum dominating set (MDS) has been proven to be NP-complete. Several heuristics have been proposed to approximate the optimal solution. The proposed solutions provide various tradeoffs between the establishment (and maintenance) of the dominating set (sometimes called backbone) and the cost of broadcast. A good survey on these schemes is provided in [40]. MDS establishment protocols can be classified as either: static protocols (i.e., proactive) that incur periodic overhead or dynamic protocols (i.e., reactive or on the fly). Dynamic MDS protocols in general perform better in the face of node mobility [38], [39]. We compare our protocol to a dynamic MDS protocol later in this section.

Comparing Query Overhead: We compare our TRANSFER approach using the step protocol to five other related schemes: 1) simple flooding; 2) on-demand routing with cache, i.e., DSR-like (we refer to this as ODC); 3) reduced broadcast location-based scheme, as described in [24] (we refer to this as smart flood); 4) dynamic minimum dominating set (MDS) (dominant pruning as in [38] and [39]); and (5) the zone routing protocol ZRP, as described in [13] and [14].

We compare thsee protocols using various network sizes and velocities. Fig. 5 shows the results for query overhead for $V\max = 20$ m/s (similar trends were observed for other

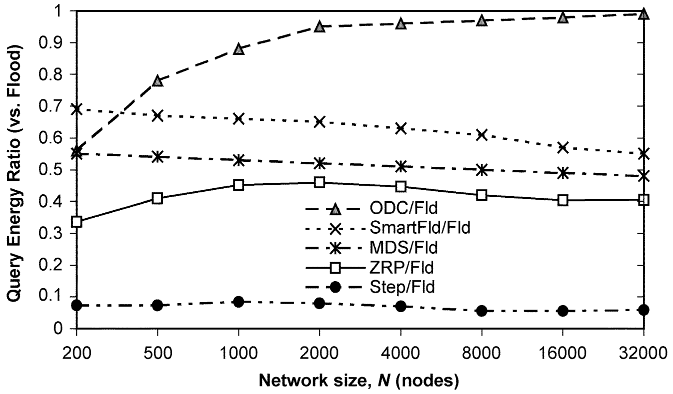

Fig. 6. Ratio of energy consumption of the various approaches normalized by flooding (Fld).

velocities). It is quite clear that there is a drastic improvement in performance using contacts, especially in large-scale networks. This is due to two main reasons: a decrease in the number of transmitted packets per query and avoiding broadcasts and using unicast for all messages in TRANSFER.

Fig. 6 shows the overhead (or energy consumption) of the various schemes relative to simple flooding. Performance of ODC degrades with an increase in network size and it approaches flooding in large-scale networks (more on this will be presented in Section VI). Reduced broadcast ranges from 55% to 70% of flooding overhead, while MDS ranges from 48% to 55% of flooding overhead. ZRP's performance varies from 33.5% of flooding in small networks to 46% at a larger scale. TRANSFER with the step policy has the best performance with 7.4% in small networks to 5.8% of flooding in large networks.

Per-Node Energy Analysis: In the analysis above, we study the total energy (per query) consumed by the whole network. Such analysis does not show the energy distribution in the network, which would be more representative of the network lifetime and potential for partitioning. In this analysis, we conduct several experiments to compare the energy distribution in the nodes for flooding, ZRP, and TRANSFER using the step policy. Furthermore, we integrate the *power-aware* contact-selection algorithm (as described in Section III) into the step scheme and call it $E-$ step. For our simulations, we take $Pth1 = 90\%$, $Pth2 = 80\%$, and so on. The 1000-node topology in Table I is used in this simulation with 1000 randomized queries. In addition, the nodes start with $E\max$ energy level, which gets reduced as the nodes transmit or receive messages according to the energy model. We conduct two sets of simulations. The first to compare the basic step protocol to flooding and ZRP, for which $E\max$ is set to 12 000 Etx units. The second set to compare step and $E-$ step, for which $E\max$ is set to 3000 Etx units. The results are shown in Figs. 7 and 8, respectively. In these figures, the horizontal axis represents the nodes as ranked (or sorted) by the remaining energy, while the vertical axis gives the actual remaining energy as percentage of $E\max$. From Fig. 7, we observe that TRANSFER protocols clearly outperform flooding and ZRP, not only in power conservation, but also in achieving a balanced power consumption between network nodes. For this set of simulations, the remaining energy for the lowest energy-ranked nodes was as follows (as a percentage of $E\max$): for flooding, 1%; for ZRP, 42%; for step, 80%; and for $E-$step, 90%. For flooding, we notice a staircase distribution for the remaining energy, which is due to the fact that all the nodes participate in the flooding for every query and nodes having higher node degree (i.e., more neighbors) consume more power.

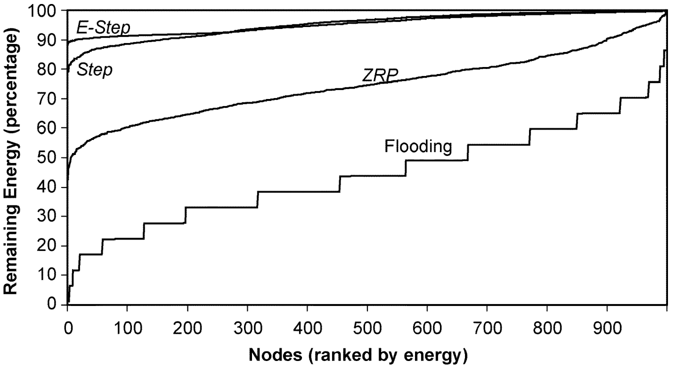

Fig. 7. Remaining energy for flooding, ZRP, step, and $E-$ step ($E\max = 12\,000$).

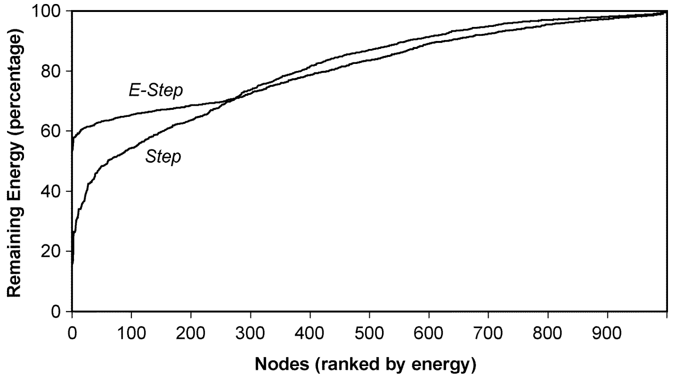

Fig. 8. Remaining energy for step and $E-$ step ($E\max = 3000$).

The second set of simulations compares step and $E-$ step. Even though the overall energy consumption was observed to be very similar, the energy-balancing capability is quite different. The least remaining energy in the nodes (as percentage of $E\max$) was 16% for step and 55% for the $E-$ step mechanism. For the lowest 280 energy-ranked nodes, $E-$ step gives more remaining energy. In sum, $E-$ step provides various improvements over the step protocol, increasing by ~40% for the lowest energy-ranked node, by more than 25% for the 20th lowest energy-ranked nodes, and by more than 15% for the 50th lowest energy-ranked nodes.

Proximity Overhead: The proximity overhead includes the energy consumed during the link-state message exchange. For the link state, the proximity exchange is in the form of broadcast messages within the proximity. This exchange increases linearly with mobility (as more link changes occur). As described before, we normalize this overhead with respect to mobility using $Z(R)$. The proximity overhead also is a function of the number of nodes in the proximity. This number is a function of R and increases with the proximity area (i.e., with R^2). Fig. 9 shows $Z(R)$ for TRANSFER as $Z(3)$ and for ZRP as $Z(5)$.

Comparisons of Total Overhead: The total energy consumed (E_T) is the combined effect of the proximity maintenance $Z(R)$ and query overhead (E_Q). As was mentioned before, metrics

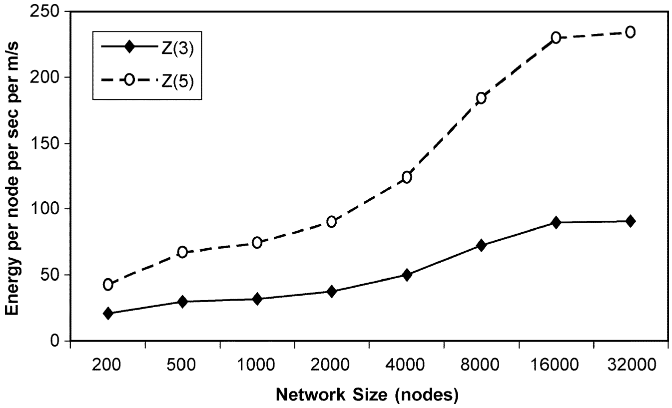

Fig. 9 Normalized intrazone overhead for the basic proximity $R = 3$, $Z(3)$, and the extended zone of $2R - 1$, $Z(5)$.

used to measure these two components need to be normalized in order to be combined in a meaningful way. This normalization is per second per node per mobility unit (m/s).

The equation for total overhead for step, $E_{T\mathrm{step}}$ is given by $E_{T\mathrm{step}} = Z(R) + E_{Q\mathrm{step}} = Z(R) + q.E_{\mathrm{step}}$. For flooding, the proximity overhead is not incurred, so $E_{T\mathrm{flood}} = E_{Q\mathrm{flood}} = q.E_{\mathrm{flood}}$. For ZRP, the intrazone overhead is incurred for an extended zone of $2R-1$; hence, $E_{T\mathrm{ZRP}} = Z(2R-1) + E_{Q\mathrm{ZRP}} = Z(2R-1) + q.E_{\mathrm{ZRP}}$. We evaluate the total energy-consumption ratio (TER) of step to the other protocols. We get

$$\mathrm{TER}_{\mathrm{flood}} = \frac{E_{T\mathrm{step}}}{E_{T\mathrm{flood}}} = \frac{Z(R) + q.E_{\mathrm{step}}}{q.E_{\mathrm{flood}}}$$

and

$$\mathrm{TER}_{\mathrm{ZRP}} = \frac{E_{T\mathrm{step}}}{E_{T\mathrm{ZRP}}} = \frac{Z(R) + q.E_{\mathrm{step}}}{Z(2R-1) + q.E_{\mathrm{ZRP}}}.$$

These formulas were validated through extensive simulations using numerous combinations of mobility degrees and query rates over all the topologies. These specific simulations were conducted with $V\max$ set to 1, 5, 20, 40, and 60 m/s. For each setting of $V\max$ the query rate q varied from 0.01 to 1000 query/km. For example, over a 1000-node topology for $V\max = 20$ m/s (i.e., $V_{\mathrm{ave}} = 10$ m/s) and 10 query/km (i.e., 0.01 query/s/(m/s) per node) the simulation assigns the rate of 0.01×10 m/s $= 0.1$ query/s for each node, i.e., the simulation triggers 100 queries/s. Simulations for the same q consistently gave very similar results for the different values of $V\max$, in line with the above formulas.

Fig. 10 and Fig. 11 show $\mathrm{TER}_{\mathrm{flood}}$ and $\mathrm{TER}_{\mathrm{ZRP}}$, respectively, as a function of the QMR q (query/km). We note that a logarithmic scale was used for q to resolve the rapid drop in the total energy-consumption ratio. Also note the difference in the y-axis scale for $\mathrm{TER}_{\mathrm{flood}}$ in Fig. 10 and $\mathrm{TER}_{\mathrm{ZRP}}$ in Fig. 11.

We first analyze the behavior of $\mathrm{TER}_{\mathrm{flood}}$ with the change in q. Results are shown in Fig. 10. For very low values of q (1–10 query/km) and small-to-medium network sizes (200–4000 nodes), flooding performs better than TRANSFER. This is due to the very low number of queries triggered as compared to the proximity maintenance overhead.[3] Note that zone-based protocols perform well when the proximity (i.e., intrazone) overhead is amortized over a reasonable number of queries in order to achieve overall gain. It is clear that for medium to large-scale networks and for a medium-to-high rate of queries, TRANSFER has a significant advantage over flooding, where $\mathrm{TER}_{\mathrm{flood}}$ approaches 5% for large networks.

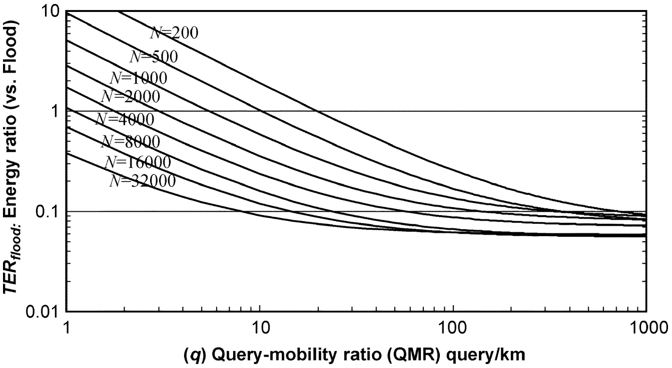

Fig. 10. Total energy ratio versus flood ($\mathrm{TER}_{\mathrm{flood}}$).

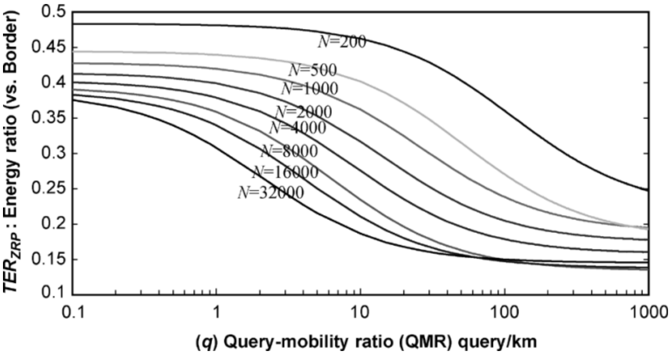

Fig. 11. Total energy ratio versus ZRP ($\mathrm{TER}_{\mathrm{ZRP}}$).

We now turn to Fig. 11 to analyze the trends in $\mathrm{TER}_{\mathrm{ZRP}}$. We notice a trend different from that for $\mathrm{TER}_{\mathrm{flood}}$, mainly because ZRP is also a zone-based approach and incurs more intrazone overhead by using the extended zone of radius $2R - 1$. Effect of the extended zone is clearest for small QMR (or q) where the proximity intrazone overhead has the dominant effect, whereas for high QMR, the effect is mainly due to the query overhead. For a small network (200 nodes) and for low q, we get $\mathrm{TER}_{\mathrm{ZRP}} \sim 48\%$, while for high q, $\mathrm{TER}_{\mathrm{ZRP}}$ is just below 25%. For medium-to-large-scale networks (500–32 000 nodes) and for low q, $\mathrm{TER}_{\mathrm{ZRP}}$ ranges from 37% to 44% and, for high q, $\mathrm{TER}_{\mathrm{ZRP}}$ ranges from 13% to 20%. Hence, the best gains for TRANSFER can be observed for higher values of QMR, where $\mathrm{TER}_{\mathrm{ZRP}}$ approaches 14% for large networks.

VI. ON-DEMAND *AD HOC* ROUTING WITH SMALL TRANSFERS

Routing protocols in *ad hoc* networks can be generally categorized as proactive (table-driven) or reactive (on-demand)

---

[3] We suspect that a scenario of very low q, indicating relatively inactive nodes, is unlikely in large-scale *ad hoc* networks. A more likely scenario is that when the nodes are inactive for extended periods of time, they may go to sleep or "OFF" mode and not participate in proximity link state exchange. Maintaining proximity information without being active is not desirable.

protocols. Previous studies [35] have shown that on-demand routing protocols with caching (e.g., DSR [5] and AODV [4]) perform much better than proactive protocols in terms of throughput and overhead, especially with node mobility. On-demand protocols operate in two main phases: route discovery (or route setup) and route repair (or maintenance). Such protocols achieve efficiency by employing efficient route-caching mechanisms to avoid unnecessary flooding of route requests and robust route-repair mechanisms to deal with frequent route breaks.

Most previous performance studies of *ad hoc* routing protocols used *long-lived* assigned connections that usually lasted throughout the simulation duration. Such simulations mainly capture the *route-repair* phase of the routing protocol, while the route-discovery/setup phase is invoked only during the initial period of the simulation. However, for *short-lived* small transfers and transactions, routing protocols may exhibit significantly different behavior, where the route-discovery/setup phase becomes the dominant factor affecting performance while route maintenance/repair is very rarely triggered.

In this section, we build a caching model for the on-demand DSR approach and evaluate the cache efficacy over large-scale wireless networks for small transfers. Note that most (if not all) previous studies on on-demand *ad hoc* routing protocols used 40–100-node networks, with long-lived connections [5], [32]–[36]. We are not aware of any previous study for cache performance in large *ad hoc* networks or for small transfers.

We develop our model to understand the behavior of on-demand routing with caching (ODC) with small transactions and to explain the that results we got for ODC in Section V.

Our caching model follows the DSR protocol design [5], [32]. A source looking for a target (or a destination) triggers a route request (RREQ) on demand. First, the source looks up its own local cache for a path to the destination. If a local cache is not found, then the source sends a query to its first-hop neighbors and they perform cache lookup. If a cached path is not found or if the found cache does not result in a positive response from the target (e.g., due to invalidity of the cache), then the source floods the route request throughout the network. The reply (from the target/destination) traverses the reverse path to the source and nodes along the path (and their neighbors) cache the path information (i.e., aggressive caching). When a cached path is used and is found to be invalid, it is attached to the following flooded route request to invalidate all copies of that path in the network.

We make minor adjustments to DSR to make it more suitable for small transfers. For example, the target responds only to the first query (as opposed to responding to several route requests as in DSR for long transfers such that the source gets multiple routes). This reduces the reply traffic overhead. Also, no intermediate caches are used. That is, when no path to the destination is found in a local or neighbor cache, then the request is flooded and is answered directly by the target. Aside from the destination itself, a node more than one hop from the source does not respond to the source (since its cache may be invalid). This reduces the number of potential floods to reach the target.

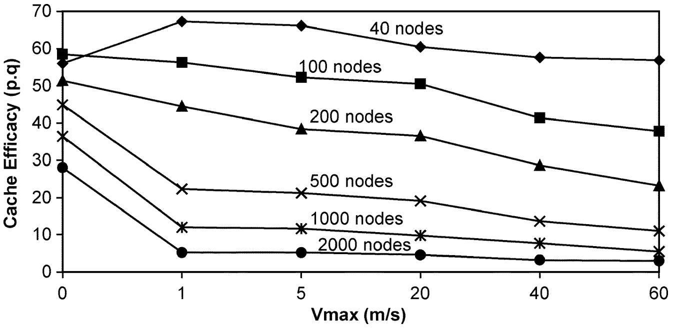

Fig. 12.Cache efficacy with various velocities and network sizes. The cache performance degrades drastically with the scale of the network and with (even very low) mobility.

The performance of DSR depends heavily on cache performance. We define the following metrics for evaluating the cache performance.

- *Cache hit ratio* (p) is the ratio of route requests that are found in the local or neighbor cache.
- *Valid cache ratio* (q) is the ratio of the cache hits that are valid (i.e., not out of- date).
- *Cache efficacy* $(p.q)$ is the ratio of route requests that are answered correctly from a valid cache.

Simply put, if we ignore the cost of local and neighbor cache lookups and the cost of replies, then the number of transmissions of DSR per query is given by

$$T_{\text{DSR}} \sim (1 - p.q).T_{\text{flood}}$$

where T_{flood} is the number of transmissions triggered by the flooding per query and p, q, as stated before.

We observe the performance of the cache for simulation settings similar to those used before. Each data point represents an average of ten simulation runs with different random seeds. Querier-target pairs were chosen randomly. one thousand such queries were performed in each run with ten queries per second; i.e., a total of 10 000 queries for each data point. Each node moves using a "random waypoint" model with no pause time. A cache warmup period was allowed before measurements were taken in each run.

The network area and radio range were set up for all topologies to have almost a constant average node degree (i.e., number of neighbors per node) equivalent to 40 nodes with a 250-m radio range in a 1000-m × 1000-m network area or 200 nodes with 110-m radio range in a 1000-m × 1000-m network area.

Results are given in Figs. 12 and 13. Fig. 12 shows the cache efficacy versus maximum velocity (Vmax) for various network sizes. For very small scale networks (40–100 nodes), the efficacy is relatively high ($\sim$50%–70%), especially for low-mobility cases. This result is consistent with previous studies on on-demand *ad hoc* routing. As the number of nodes increases, however, the cache efficacy drops dramatically, even for very low mobility (1 m/s), to $\sim$10% for 1000 nodes and to $\sim$5% for 2000 nodes.

Fig. 13 gives a closer look at the cache metrics. It is apparent that the cache performance depends on mobility, but much more so on network size. The cache hit ratio p drops from $\sim$73% (for

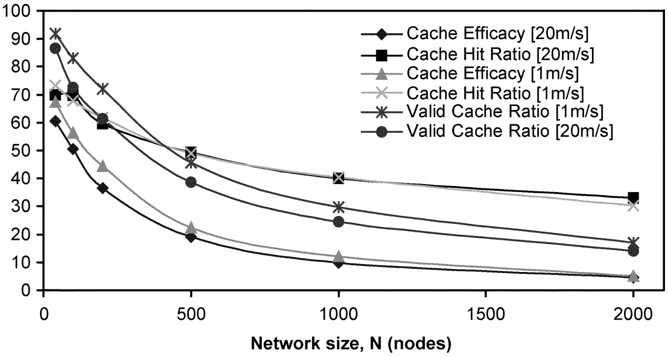

Fig. 13. Cache hit ratio (p), valid cache ratio (q), and cache efficacy $(p.q)$ with the network size for 1 and 20 m/s.

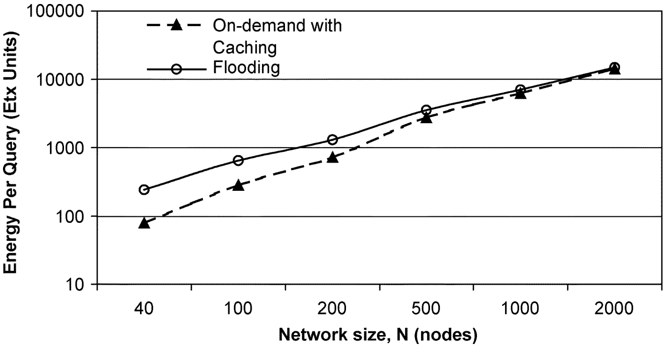

Fig. 14. Per-query overhead of on-demand routing and flooding for the same simulation setup, with 1 m/s mobility.

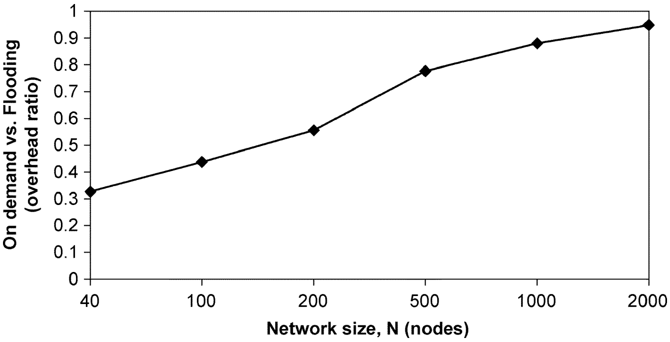

Fig. 15. Ratio of on-demand routing overhead to flooding overhead. As the network size increases, caching performance degrades and on-demand routing overhead approaches flooding overhead.

40 nodes) to ∼30% (for 2000 nodes). The more drastic drop occurs in the valid cache ratio q, from ∼92% (for 40 nodes) to ∼14% (for 2000 nodes), which brings the overall cache efficacy $p.q$ down.

For moderate-to-large-scale networks (above 1000 nodes), the performance of on-demand routing with caching approaches flooding, where the on-demand routing protocol resorts to flooding more than 90% of the time due to cache misses or invalid cache hits.

Figs. 14 and 15 show the simulation results (in terms of energy per query) for the DSR on-demand routing as compared to flooding.

Based on this analysis, we believe that on-demand routing protocols are not suitable for small transfers in large-scale wireless networks.

VII. RELATED WORK

The simplest form of query/transaction is simple *flooding*. This scheme does not scale well, as we have shown. Expanding ring search (ERS) uses repeated flooding with incremental TTL. This approach and its derivatives do not scale well also. (ERS performs worse than flooding in many cases. Our case study on ERS was not presented in this paper for clarity of presentation where flooding was considered the base line for comparisons.)

Ad hoc routing protocols, in general, may be classified as reactive, proactive, hybrid, or hierarchical. Proactive schemes such as DSDV [3], WRP [20], and GSR [19] cause updates to be periodically flooded throughout the network. These schemes may be suitable for small-scale networks, but are not suitable for large-scale networks. Reactive schemes such as AODV [4] and DSR [5] attempt to reduce the overhead due to periodic updates by maintaining the state only for the active sources. In these schemes, a search may be initiated for each new query involving flooding (or expanding ring search). These schemes may be suitable for small–medium-size networks. However, as we have shown in our study, the cache efficacy drops drastically with the scale of the network for small transfers in mobile networks. The above protocols in general attempt to find shortest path routes. In many cases, establishing such routes incurs much more overhead than is needed for small transactions. Especially for large-scale networks, the previous schemes incur excessive overhead and are unsuitable for our target applications.

The ZRP [9], [11]–[14] uses a hybrid approach, where link state is used intrazone and on-demand routing (border casting) is used interzone. A feature of ZRP is that a zone is node specific and no complex coordination is used. The concept of zone (or cluster) that is used by ZRP is similar to the concept of proximity in this paper. ZRP performs well for discovering high-quality routes, but is not well suited for small transfers and queries, as we have shown. The main concepts upon which contacts were designed in TRANSFER are fundamentally different than ZPR. Also, the target application is different. ZPR attempts to find high-quality routes for prolonged transfers. In our study, we target resource discovery and small transfers and make a clear design tradeoff between route optimality and the reduction of overall overhead and energy consumption.

There are several major differences in the design of TRANSFER that deviate significantly from ZPR's concepts of design.

1) The concept of contacts, borrowing from small worlds, is fundamentally different than bordercasting in ZPR. Bordercasting constitutes a major part of ZPR's design, along with the query termination mechanisms.
2) In TRANSFER, several novel mechanisms were designed and analyzed for: 1) choosing useful contacts on the fly; 2) specifying the proximity overlap reduction (POR) and route-overlap reduction (ROR) mechanisms; and 3) developing the power-aware contact-selection

protocol. These schemes were introduced after investigating several other schemes.

Several features of *TRANSFER* were carefully designed to enable power awareness and power balance, including: 1) the concept of selecting a small number of "contacts" (NoC), as opposed to every border (as in bordercasting) and 2) on-the-fly (on-demand) contact selection for every query attempt. Note that a query in the "step" protocol may require several attempts to be resolved. These features enable much better power balance for TRANSFER, which effectively translates to extending the network lifetime.

3) The mechanisms for the "step" protocol for repeated queries, with exponential step, are novel and were chosen after careful investigation of several alternative schemes. Details of other alternatives are omitted for readability. The repeated queries, along with on-the-fly contact selection, provide a *rotation-like* effect for the successive queries, which leads to a significant increase in network search coverage (versus single-attempt approaches, such as ZRP and flooding). Without this mechanism, the success rate of the proposed scheme may drop with the scale of the network.
4) In TRANSFER, link-state message exchange is performed for only R hops, where R is the proximity radius. However, because ZRP employs query control and termination mechanisms ($Q-1, Q-2$, and ER [13], [14]), it requires link state exchange for $2R - 1$ hops. This makes a significant difference in the zone-maintenance overhead, as described in Section V.
5) *TRANSFER* uses unicast messages, whereas ZRP uses multicast or broadcast messages that are needed in ZRP's query-termination mechanisms in which neighbors of query forwarders need to overhear and store query information. Using unicast messages translates into significant power savings.

In Section V, we presented detailed comparisons with ZRP and showed that the contact-based approach incurs significantly lower overhead for our purposes. We believe that our work may be complementary to zone routing. With simple extensions to zone routing, it is quite conceivable that the two maybe integrated in a unified architecture in which ZRP is used for route discovery while TRANSFER is used for resource discovery and small transfers. Also in Section V, we compared our scheme to DSR [5], smart flooding [24], and MDS [38], [39] and concluded that using contacts results in significant power and overhead savings over all these other protocols.

Related work on smart flooding is given in [17], [23], and [24]. The main idea is to exploit node density to reduce redundant transmissions, sometimes reducing coverage. Such work is complementary to our work. For high-density networks, this work maybe integrated with our work to provide more efficient intrazone exchange.

Hierarchical schemes, such as CGSR [21] and [22], involve the election of cluster heads. The cluster head is responsible for routing traffic in and out of the cluster. Also, a cluster head may be a single point of failure and a potential bottleneck. Other hierarchical schemes use landmarks [6], [15], [16]. Landmark routing avoids traffic concentration by using the direction of landmarks for routing. Landmarks do not necessarily forward the packets for their respective zones. Advertisement, promotion, and demotion schemes are used for node coordination to construct the hierarchy. Cluster- and landmark-based hierarchies rely on complex coordination and, thus, are susceptible to major reconfiguration with mobility, leading to serious performance degradation. We do not employ any coordinated election schemes. In our architecture, each node maintains its proximity independently, so no major reconfiguration is incurred with mobility.

In GLS [7], an architecture is presented for location discovery that is based on a *grid* map of the network. This map must be known *a priori* by all nodes. Nodes use geographic routing and the network map to recruit location servers to maintain their location. Nodes use a consistent mapping algorithm to update and search for node locations. This is a useful architecture given that the network map is known and geographic data is available (through a global positioning system or other). We do not make such assumption for our architecture.

The algorithms proposed in [2] and [8] use global information about node locations to establish short cuts or friends and use geographic routing to reach the destination. It is unclear how such architectures are feasible with mobility. Also, the destination ID (and location) must be known in advance, which may not be the case in resource discovery.

In [18], we have shown the relationship between small worlds and wireless networks. In this paper, we build upon that relationship by introducing the contacts to act as short cuts in the highly clustered multihop wireless network, proposing and evaluating—in detail—an efficient on-the-fly contact-selection mechanism. We first introduced the high level idea of using contacts in [10]. The initial work on the TRANSFER protocol was presented in [42]. This work extends the analysis of the TRANSFER protocol, introduces the power-aware contact-selection mechanism, and compares its performance to various related schemes. The MARQ architecture [43] provides a mobility-assisted contact-selection mechanism, the efficiency of which increases with mobility. In cases of static networks (e.g., sensor networks) or when mobility is low, MARQ may not be able to exploit mobility and, hence, TRANSFER may be used in conjunction with MARQ for efficient query resolution. In CARD [44], we propose a proactive contact-selection technique. Although such a technique may be able to reduce proximity overlap (due to the serial selection of contacts), it incurs overhead for contact selection and maintenance even when queries are not issued. TRANSFER provides a reactive (on-the fly) contact-selection mechanism in which contacts are selected in parallel and contact selection is done as part of the query resolution mechanism (i.e., only when queries are issued). This also allows for the power-aware contact-selection mechanism presented here.

ACQUIRE [45] is an architecture for multivariable query resolution in sensor networks, which uses the look-ahead technique to optimize overhead. The query is forwarded from one querying node to another d hops away, randomly. The work provides an analytical framework to get optimal d for given level of network and event dynamics. A variant of the TRANSFER contact selection may be used to reduce the overlap between the look-ahead zones for successive querying nodes to improve the performance of ACQUIRE.

VIII. CONCLUSION

We have presented the TRANSFER architecture for transactions routing in large-scale *ad hoc* networks. For transactions, the overhead incurred for obtaining high-quality routes is not justified as compared to the transfer of the actual data. Hence, the main design goal in such target applications is to reduce communication overhead and power consumption, rather than route optimization. We employ the concept of contacts to provide very efficient search in the *ad hoc* network.

This work is one of the first works on transactions routing in large-scale *ad hoc* networks. The main contributions of this paper include:

- introducing the TRANSFER architecture for power-efficient transaction routing in *ad hoc* networks;
- designing a simple, effective, on-the-fly contact-selection protocol for proximity-overlap reduction;
- developing a power-aware contact-selection mechanism that establishes better power balancing and prolongs the network lifetime;
- evaluating, in detail, the different dimensions of the design space and scalability of our protocol;
- comparing the performance of our protocols against five state-of-the-art routing and discovery protocols, including flooding, smart flooding, minimum dominating sets, on-demand *ad hoc* routing, and ZRP using extensive simulations over a wide array of networks and query rates.

Our results show significant savings when using our contact-based techniques. For large networks and high query rates, TRANSFER consumes as little as 5% of flooding energy and 14% of border-cast energy. This study also shows reasonable settings of parameters that work well for a wide range of network sizes (from 200–32 000 nodes) and that the performance of TRANSFER is relatively insensitive to the topology size or the degree of mobility for the scenarios in our study.

In the future, we plan to study the performance of TRANSFER under a variety of node distributions and mobility models.

REFERENCES

[1] D. Watts and S. Strogatz, "Collective dynamics of 'small-world' networks ," *J. Nature*, vol. 393, pp. 440–442, 1998.

[2] J. Kleinberg, "Navigating in a small world," *J. Nature*, vol. 406, p. 845, 2000.

[3] C. E. Perkins and P. Bhagwat, "Highly dynamic Destination-Sequenced Distance Vector routing (DSDV) for mobile computers," *Comp. Commun. Rev.*, vol. 24, no. 4, pp. 234–244, 1994.

[4] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications*, Feb. 1999, pp. 90–100.

[5] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad-hoc wireless networks," in *Mobile Computing*. New York: Kluwer, 1996, ch. 5, pp. 153–181.

[6] P. Guangyu, M. Gerla, and X. Hong, "LANMAR: landmark routing for large scale wireless ad hoc networks with group mobility," in *Proc. ACM MobiHOC*, 2000, pp. 11–18.

[7] J. Li, J. Jannotti, D. Couto, D. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," in *Proc. ACM MOBICOM'00*, 2000, pp. 120–130.

[8] L. Blazevic, S. Giordano, and J.-Y. Le Boudec, "Anchored path discovery in terminode routing," presented at the *Proc. 2nd IFIP-TC6 Networking Conf. (Networking'02)*, Pisa, Italy, May 2002.

[9] M. Pearlman and Z. Haas, "Determining the optimal configuration for the zone routing protocol," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1395–1414, Aug. 1999.

[10] A. Helmy, "Architectural framework for large-scale multicast in mobile ad hoc networks," in *IEEE Int. Conf. Communications (ICC'02)*, vol. 4, New York, Apr. 2002, pp. 2036–2042.

[11] Z. Haas and M. Pearlman, The zone routing protocol (ZRP) for ad hoc networks, in IETF Internet Draft, Manet Group, Jun. 1999.

[12] ——, "The performance of query control schemes for the zone routing protocol," in *Proc. ACM SIGCOMM '98*, 1998, pp. 167–177.

[13] ——, "ZRP: a hybrid framework for routing in ad hoc networks," in *Ad Hoc Networks*, C. Perkins, Ed. Reading, MA: Addison-Wesley, 2001, pp. 221–254.

[14] ——, "The performance of query control schemes for the zone routing protocol," *ACM/IEEE Trans. Networking*, vol. 9, no. 4, pp. 427–438, Aug. 2001.

[15] P. F. Tsuchiya, "The landmark hierarchy: a new hierarchy for routing in very large networks," *Comp. Commun. Rev.*, vol. 18, no. 4, pp. 35–42, Aug. 1988.

[16] S. Kumar, C. Alaettinoglu, and D. Estrin, "SCOUT: Scalable object tracking through unattended techniques," in *Proc. IEEE ICNP'00*, 2000, pp. 253–262.

[17] J. J. Aceves and M. Spohn, "Bandwidth-efficient link-state routing in wireless networks," in *Ad Hoc Networks*, C. Perkins, Ed. Reading, MA: Addison-Wesley, 2001, pp. 323–350.

[18] A. Helmy, "Small worlds in wireless networks," *IEEE Commun. Lett.*, vol. 7, no. 10, pp. 490–492, Oct. 2003.

[19] T.-W. Chen and M. Gerla, "Global state routing: A new routing scheme for ad-hoc wireless networks," in *Proc. IEEE ICC'98*, 1998, pp. 171–175.

[20] S. Murthy and J. J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," *ACM Mobile Networks Appl. J.*, vol. 1, no. 2, pp. 183–197, 1996.

[21] C.-C. Chiang, H. Wu, W. Liu, and M. Gerla, "Routing in clustered multihop, mobile wireless networks with fading channel," in *Proc. IEEE SICON'97*, Apr. 1997, pp. 197–211.

[22] J. Liu, Q. Zhang, W. Zhu, J. Zhang, and B. Li, "A novel framework for QoS-aware resource discovery in mobile ad hoc networks," in *IEEE ICC'02*, vol. 25, 2002, pp. 1011–1016.

[23] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, and L. Viennot, "Optimized link state routing protocol," in *IEEE INMIC'01*, 2001, pp. 62–68.

[24] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *ACM MOBICOM'99*, 1999, pp. 151–162.

[25] E. Cohen and S. Shenker, "Replication strategies in unstructured peer-to-peer networks," in *Proc. ACM SIGCOMM Conf.*, 2002, pp. 177–190.

[26] F. Bai, N. Sadagopan, and A. Helmy, "IMPORTANT: A framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks," in *IEEE INFOCOM'03*, 2003, pp. 825–835.

[27] A. Bestavros, J. Byers, and K. Harfoush, "Inference and labeling of metric-induced network topologies," in *Proc. IEEE INFOCOM'02*, Jun. 2002, pp. 628–637.

[28] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu, "Advances in network simulation," *IEEE Comp.*, vol. 33, no. 5, pp. 59–67, May 2000.

[29] Y. Ko and N. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," *Wireless Networks*, vol. 6, no. 4, pp. 307–321, 2000.

[30] B. Karp and H. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. ACM MobiCom'00*, 2000, pp. 243–254.

[31] J. Navas and T. Imielinski, "GeoCast—Geographic addressing and routing," in *Proc. ACM MobiCom'97*, 1997, pp. 66–76.

[32] D. B. Johnson, D. Maltz, and J. Broch, "DSR: The dynamic source routing protocol for multihop wireless ad hoc networks," in *Ad Hoc Networks*, C. Perkins, Ed. Reading, MA: Addison-Wesley, 2001, pp. 139–172.

[33] Y. Hu and D. Johnson, "Caching strategies in on-demand routing protocols for ad hoc wireless networks," in *Proc. ACM MobiCom'00*, 2000, pp. 231–242.

[34] Y.-C. Hu and D. B. Johnson, "Ensuring cache freshness in on-demand ad hoc network routing protocols," in *Proc. POMC Workshop Principles of Mobile Computing*, Oct. 2002, pp. 25–30.

[35] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proc. ACM MobiCom*, 1998, pp. 85–97.

[36] D. A. Maltz, "On-demand routing in multi-hop wireless ad hoc networks," Ph.D. dissertation, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, May 2001.

[37] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *ACM Mobicom'99*, 1999, pp. 151–162.

[38] H. Lim and C. Kim, "Flooding in wireless ad hoc networks," *Comp. Commun. J.*, vol. 24, no. 3–4, pp. 353–363, 2001.

[39] A. Laouiti, A. Qayyum, and L. Viennot, "Multipoint relaying: An efficient technique for flooding in mobile wireless networks," in *Proc. HICSS'02*, 2002, pp. 298–307.

[40] Y. Chen and A. Liestman, "Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks," in *Proc. ACM MobiHoc'02*, June 2002, pp. 165–172.

[41] U. Kozat and L. Tassiulas, "Service discovery in mobile ad hoc networks: An overall perspective on architectural choices and network layer support issues," *AdHoc Networks J.*, vol. 2, no. 1, pp. 23–44, 2003.

[42] A. Helmy, "TRANSFER: Transactions routing for ad-hoc networks with efficient energy," in *Proc. IEEE GLOBECOM'03*, Dec. 2003, pp. 398–404.

[43] ——, "Mobility-assisted resolution of queries in large-scale mobile sensor networks (MARQ)," *Comp. Networks J.*, vol. 43, no. 4, pp. 437–458, 2003.

[44] A. Helmy, S. Garg, P. Pamu, and N. Nahata, "Contact Based Architecture for Resource Discovery (CARD) in large scale MANets," in *Proc. IEEE/ACM Int. Parallel and Distributed Processing Symp. (IPDPS)*, Apr. 2003, pp. 219–227.

[45] N. Sadagopan, B. Krishnamachari, and A. Helmy, "The ACQUIRE mechanism for efficient querying in sensor networks," in *Proc. IEEE 1st Int. Workshop Sensor Network Protocols and Applications (SNPA), in Conjunction With IEEE ICC*, May 2003, pp. 149–155.

Ahmed Helmy (S'94–M'99) received the B.S. degree in electronics and communications engineering and the M.S. degree in engineering mathematics from Cairo University, Cairo, Egypt, in 1992 and 1994, respectively, the M.S. degree in electrical engineering, and the Ph.D. degree in computer science from the University of Southern California (USC), Los Angeles, in 1995 and 1999, respectively.

Since 1999, he has been an Assistant Professor of Electrical Engineering at USC. In 2000, he founded and is currently directing the Wireless Networking Laboratory at USC. His current research interests are in the areas of protocol design and analysis for mobile *ad hoc* and sensor networks, mobility modeling, design and testing of multicast protocols, IP micromobility, and network simulation. His personal Web site is http://ceng.usc.edu/~helmy/

Dr. Helmy is a Member IEEE Computer Society. In 2002, he received the U.S. National Science Foundation CAREER Award. In 2000, he received the USC Zumberge Research Award and, in 2002, he received the Best Paper Award from the IEEE/IFIP International Conference on Management of Multimedia Networks and Services (MMNS).