TRANSFER: Transactions Routing for Ad-hoc NetworkS with eFficient EneRgy

Ahmed Helmy {helmy@usc.edu}
Department of Electrical Engineering – Systems, University of Southern California

Abstract-*In future ad hoc networks, transactions are expected to become one of the primary types of flows. Transactions require only a very small number of packets to complete. Hence, providing optimal (or shortest path) routes to such transfers consumes more energy than the actual transfer. Conventional routing protocols are thus unsuitable for transactions. In this paper, we present a novel architecture for transaction routing in large-scale ad hoc networks. In our approach we aim at reducing the total energy consumption of successful delivery as opposed to finding shortest path routes.*

Our architecture uses a hybrid approach, where each mobile node obtains information about nodes in its proximity, up to R hops away, using a proactive link state protocol. Beyond the proximity, we introduce the novel notion of contacts that act as short cuts to reduce the degrees of separation between the request source and the target. We propose an efficient, on-demand, contact selection protocol. No location information is assumed. Extensive simulations are used to evaluate the performance of our protocol in terms of energy consumption and request success rate. We compare our architecture to flooding and ZRP. Our results show substantial power savings for our contact-based technique, especially for large networks.

1. 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 constraint with communication being a major power consumer. Also mobility creates a dynamic environment in which routes become frequently invalid. Hence it becomes very inefficient to discover optimal routes for small transactions, since the cost of such discovery is quite high. Conventional routing protocols strive to provide shortest path routes. Instead of searching for optimal routes we take a novel approach to trade-off route optimality for lower energy consumption. We introduce a new routing protocol geared towards small transactions in mobile wireless ad hoc networks. We design our protocol to be scalable, power-efficient, mobility-adaptive, and self-configuring.

To our knowledge, our work is the first routing protocol for small transactions in ad hoc networks that explicitly incorporates this design trade-off. 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*. We introduce the concept of *contacts* as key for efficient transfers in our scheme. For a node, contacts are a few nodes outside of the proximity that act as *short cuts* to transform the wireless network into a small world and hence reduce the average degrees of separation between the querier and the target. When a request[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. Also, 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 and ZRP over a variety of networks. Our results show significant overall energy savings for our technique. For large networks and high request rates, TRANSFER consumes as little as 5% of flooding energy and 14% of ZRP energy.

The rest of the paper is outlined as follows. Section 2 provides overview. Section 3 presents contact-selection and search policy. Section 4 provides request processing rules. Section 5 provides evaluation results. Section 6 discusses related work and Section 7 concludes.

2. 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 re-configuration 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 1 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 small. As part of the proximity information each node keeps routes to nodes in its proximity, including borders at *R* hops.

When a querier node, *Q*, issues a transaction, 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 the proximity, as follows. First, a request is issued to *NoC* (say 3) of *Q*'s borders (*R* hops away). Each border, *B*, receiving the request would in turn select another node, *C*, at *r* hops away to which to forward the request. We call *C* a *contact* node. To increase search efficiency, *C* should have low proximity overlap with *Q*. Proper setting of the parameter *r* helps to reduce such overlap. Contact nodes act as short cuts that bridge between disjoint proximities. This helps to reduce the degrees of separation between *Q* and the target nodes.

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

Degrees of separation in this context refer to the number of intermediate nodes to get from the querier node to the target.

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

Questions regarding setting the design parameters, such as number of contacts (*NoC*), contact distance (*r*), and proximity radius (*R*), shall be investigated in the evaluation section. First, we describe our contact selection scheme.

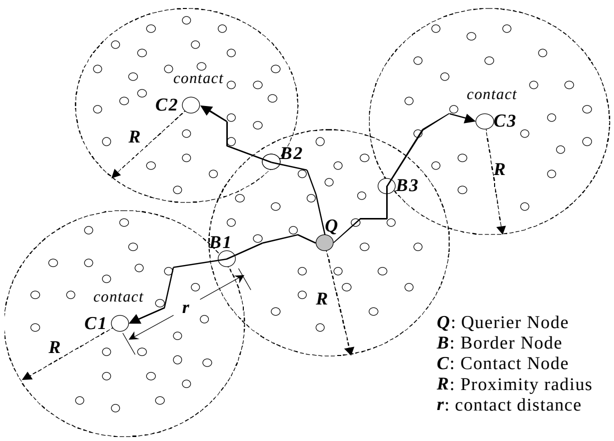

Figure 1. Each node in the network has a proximity of radius *R* hops. A querier node, *Q*, sends a request through a number of its borders equal to the number of contacts (*NoC*), in this case *NoC*=3. Each border node, *Bi*, chooses one of its borders, *Ci*, to be the direction for forwarding the request *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*.

3. Contact Selection and Search Policy

The main purpose of a contact node is to act as a short cut to increase the view of the network by searching for the target in uncovered parts of the network. 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*. The first kind of overlap occurs between the contact's proximity and the querier's proximity. To attempt to reduce this overlap we attempt to push the request as far out from the querier's proximity as possible.

A querier node, *Q*, sends a request to *NoC* of its borders. Let one of those borders be *B*. *B* has its own proximity of *R* hops. Using the proximity information (that was formed using the link state protocol), *B* constructs a topology view up to *R* hops away, and chooses a border in its proximity that has maximum distance to *Q*. 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 *direction* for choosing the contact, we call this 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 at each chosen border.

The second type of overlap occurs between proximities of contacts. To reduce this overlap, *Q* selects *NoC* borders with maximum separation. This is done using the proximity information (with no extra overhead). We call this scheme the *route overlap reduction* (*ROR*) scheme.

Levels of Contacts: The above contact selection schemes (*POR* and *ROR*) provide a mechanism 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 farther 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 shall 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 is that level-*i* contacts need to perform *POR* and *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.

Search Policy: Given a request and a number of levels, *D*, the target search process uses what we call an exponential step policy (*step*). In *step* the request 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 up to d_i³D or $d=d_{max}$, where d_{max} is the first d_i that satisfies the inequality $2d_{max}>D$ for $D>2$. (For D£2, $d_{max}=D$). For example, if $D=20$ then $d_{max}=16$.

4. Request Forwarding and Processing

The Request Message: The request message contains the target ID, which could be the node ID or the resource key (for resource discovery). For small transfers, the request may also carry the data. The destination ID in the request contains the ID of the border node (or the direction border). The request 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.

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

Search, Processing and Forwarding: A contact (or a border node) receiving the request, first searches in the proximity information. If the target is found, the request 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 request to determine which functions to perform, and whether it is a contact, two fields are included in the request 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: (1) If 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 level-count reaches '0' the contact drops the request. If level-count is not '0', the contact selects *NoC* borders (using *POR* and *ROR* as in section 3), and sends the request to those borders. (2) If the hop-count is not '0', and current node is the destination of the request message, the receiving node acts as a border node. It selects a direction border (using *POR* as in section 3), and sends the request to it. (3) Otherwise, the request is simply forwarded to the next hop to the destination.

Note that the request message is *unicast* hop by hop, it is *not* broadcast hop by hop.

5. 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][14]).

Particularly, we attempt to study the effect of changing the number of contacts (*NoC*), contact distance (*r*), *maxdepth* (*D*), and proximity radius (*R*) on protocol performance. The main performance metrics include energy consumption and request success rate. Note the trade-off 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 request failures. Any failure beyond an acceptable level will be recovered using flooding. Hence, *the scheme used in our simulations is contact-based search, if 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 request success rate.

5.1. Simulation setup

We use extensive simulations to investigate the design space 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 110m. We study a wide range of network sizes, as shown in Table 1. We also vary the area of the network to maintain connectivity. *N* nodes are randomly placed in 'ℓ m xℓ m'.

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

Table 1. Networks used in the simulation. Nodes are initially randomly distributed. Number of border and proximity nodes are given for *R*=3.

For mobility we use the random way point model, where a destination is chosen randomly and a velocity is chosen randomly from [0,*Vmax*]. Once the destination is reached, another random destination is chosen, so on. We use *Vmax*=0 to 60m/s, and various request/query rates varying from 0.01 query/km up to 1000 query/km. We developed a discrete event simulator for the protocols under study. For the purposes of our simulation we do not implement MAC layer collisions. We implement a hop-by-hop energy model taking into account transmitted and received packets.

Hop-by-Hop Communication Energy Model

The energy consumed when a request 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 *receive* state. In general, a wireless node may be in one of three power states: (*i*) transmit state, (*ii*) receive state, or (*iii*) 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 request packet as *Etx*. Similarly, *Erx* refers to the energy of request 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 the average node degree (*g*). For a unicast message, usually there is a small *handshake* phase to inform the neighbors of the impending transmission. In IEEE 802.11 (the model we adopt), CSMA/CA is used with handshake and medium reservation. The handshake involves 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 request 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

[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, other nodes within radio range backoff 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 ~10%.

accordingly. To have the results be independent of the packet size used, we measure in the energy consumed as a function of *Etx*; i.e., we have the energy measure in *Etx* units.

Each data point represents an average of 10 simulation runs with different random seeds. Low variability between runs was observed. Querier-target pairs were chosen randomly. 1000 such queries were performed in each run; i.e., a total of 10,000 queries (or requests) for each data point.

We first present the overhead per request (hereafter referred to as overhead per query), then the proximity overhead, and finally we present the overall overhead.

5.2. Overhead per Query

For this set of simulations we use the 1000 node topology in Table 1.

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. Figure 2 shows the effect of varying *r* and clearly indicates favorable settings. 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*£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.

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

Effect of Number of Contacts (NoC): Results in Figure 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 low success rate. Increasing *NoC* increases success rate until almost all requests succeed then we see an increase in overhead due to additional (unnecessary) search branches with 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 Figure 4 show that increasing *D* generally decreases the power consumption by increasing the success rate and subsequently reducing fallback to flooding. 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 re-visit prevention mechanisms this number is drastically reduced. After *D*=33 (i.e., 6 attempts) most requests (97.5% or more) become successful and energy consumption almost saturates.

5.3. Scalability Analysis of Total Overhead

In this section we evaluate the scalability characteristics of our protocol. There are two main overhead components for TRANSFER: (a) query overhead, and (b) proximity maintenance. In the previous section we have 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/sec) 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.

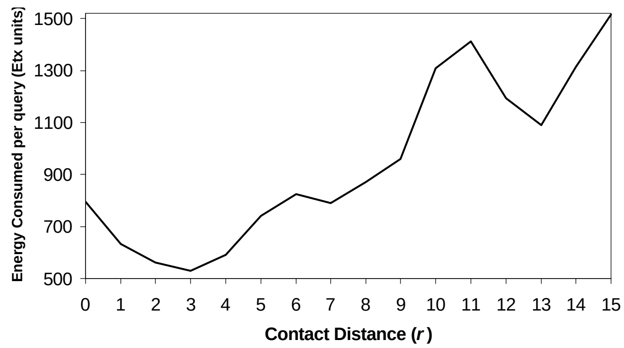

Figure 2. Effect of Contact Distance (*r*)

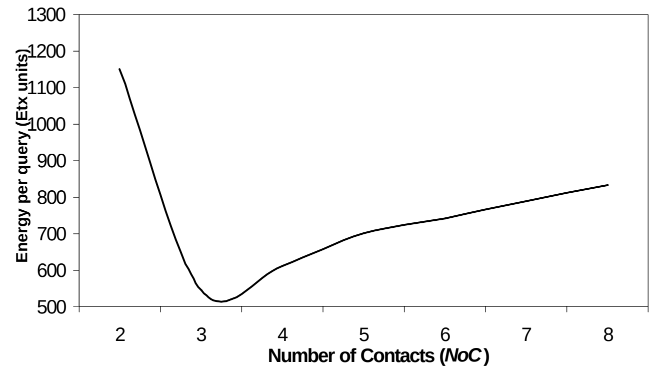

Figure 3. Effect of Number of Contacts (*NoC*)

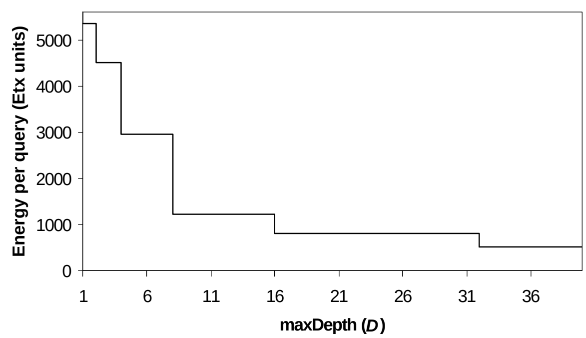

Figure 4. Effect of maximum depth (*D*)

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 present a metric called *QMR* (query-mobility-ratio, or *q*) defined per node as query/s/(m/s) or simply query/km. 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 sec per node per (m/s)'. Also, let us call the energy consumption per query 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 $E_{Qstep} = q.E_{step}$. The units of E_{Qstep} are in '*Etx* units per sec per node per m/s', compatible with *Z(R)*. The total overhead becomes $E_{Tstep}=Z(R)+E_{Qstep}$.

Our goal in this section is to obtain trends and comparisons of total overhead for 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 1).

Related schemes

We compare our protocols to flooding and ZRP. For flooding, in a network of *N* nodes, the request is transmitted by *N*-1 nodes. We get $E_{flood}=(N-1).Etx(1+f.g)»Etx(N+2Lf)$, 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 request to its zone borders, and the borders send it to their borders, so on. Request messages are broadcast (or multicast) hop by hop and nodes along the path record the request information. Requests that are sent to previously visited borders are terminated. For a zone of radius *R*, each node keeps track of nodes up to 2*R*-1 hops.

We simulate flooding and ZRP for our study. We now analyze scalability of the query overhead, then proximity overhead, followed by analysis of total overhead.

5.3.1. Comparison with Related Schemes (Query Overhead)

We compare our protocol to flooding and ZRP using the various network sizes. In Figure 5 we show the results for query overhead for *step*, *flooding* and *ZRP*. 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: (*i*) decrease in number of transmitted packets per query, and (*ii*) avoiding broadcast and using unicast for all messages. We define the query energy ratio (*QER*), as, $QER_{flood}=E_{step}/E_{floo}$,and $QER_{ZRP}=E_{step}/E_{ZRP}$. QER_{flood} ranges between 5% (for large networks) and 8.5% (for small-medium networks), while QER_{ZRP} ranges between 14% (for large networks) and 22% (for small-medium networks).

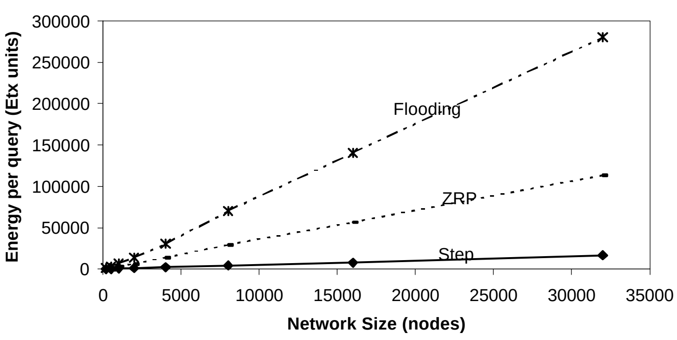

Figure 5. Ratio of *step* query overhead vs. other approaches

5.3.2. Proximity Overhead

The proximity overhead includes the energy consumed during link state message exchange. For 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 was described before, we normalize this overhead with respect to mobility using *Z(R)*. The proximity overhead is also 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). Figure 6 shows *Z(R)* for TRANSFER as *Z(*3*)* and for ZRP as *Z(5)*.

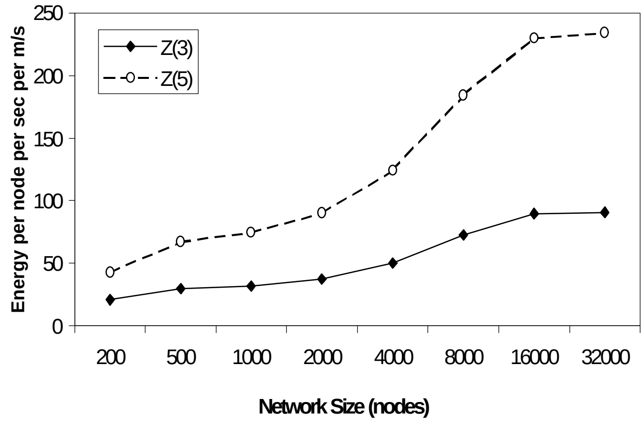

Figure 6. Normalized Intra-zone overhead for the basic proximity *R*=3, *Z(*3) and the extended zone of 2*R*-1, *Z(*5)

5.3.3. Comparisons of Total Overhead

The total energy consumed is the combined effect of proximity maintenance and query overhead. As was mentioned, metrics 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 is as follows: $E_{Tstep}=Z(R)+E_{Qstep}= Z(R)+q.E_{step}$. For flooding, the proximity overhead is not incurred, so $E_{Tflood}=E_{Qflood}= q.E_{flood}$. For ZRP the intra-zone overhead is incurred for an extended zone of 2*R*-1, hence $E_{TZRP}=Z(2R-1)+E_{QZRP}= Z(2R-1)+q.E_{ZRP}$. We evaluate the total energy-consumption ratio, *TER*, of *step* to the other protocols. We get:

$$TER_{flood} = \frac{E_{Tstep}}{E_{Tflood}} = \frac{Z(R)+q.E_{step}}{q.E_{flood}},$$

and

$$TER_{border} = \frac{E_{Tstep}}{E_{Tborder}} = \frac{Z(R)+q.E_{step}}{Z(2R-1)+q.E_{border}}.$$

Figure 7 and 8 show TER_{flood} and TER_{ZRP}, respectively, as function of the *QMR* (query-mobility ratio) *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 *TER*.

We first analyze the behavior of TER_{flood} with the change in *q*. Results are shown in Figure 7. For very low values of *q* (1-10 query/km) and small to medium network sizes (200-4000 nodes) flooding performs better. This is due to the very low number of queries triggered as compared to the intra-zone maintenance overhead[3]. Note that, zone-based protocols perform well when the intra-zone 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 medium to high rate of queries, *TRANSFER* has a significant advantage over flooding, where TER_{flood} approaches 5% for large networks.

We now turn to Figure 8 to analyze the trends in TER_{ZRP}. We notice a trend different from that for TER_{flood}, mainly because ZRP is also a zone-based approach and incurs more intra-zone overhead by using the extended zone of radius (2*R*-1). Effect of the extended zone is clearest for small *QMR* where the intra-zone 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 TER_{ZRP} ~48%, while for high *q*, TER_{ZRP} is just below 25%. For medium to large-scale networks (500-32000 nodes) and for low *q*, TER_{ZRP} ranges from 37% to 44%, and for high *q*, TER_{ZRP} ranges from 13% to 20%. Hence, the best gains for TRANSFER can be observed for higher values of *QMR*, where TER_{ZRP} approaches 14% for large networks.

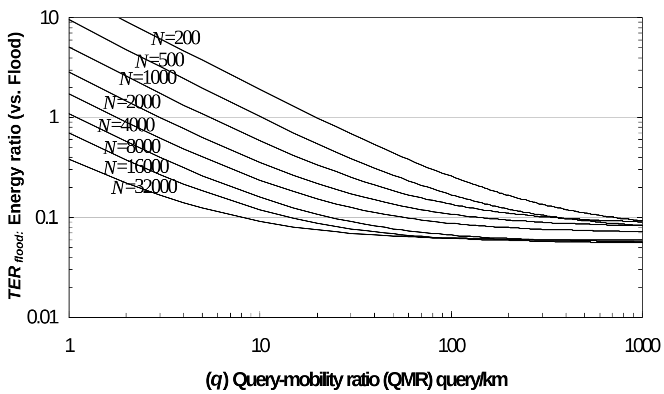

Figure 7. Total energy ratio vs. flood (TER_{flood})

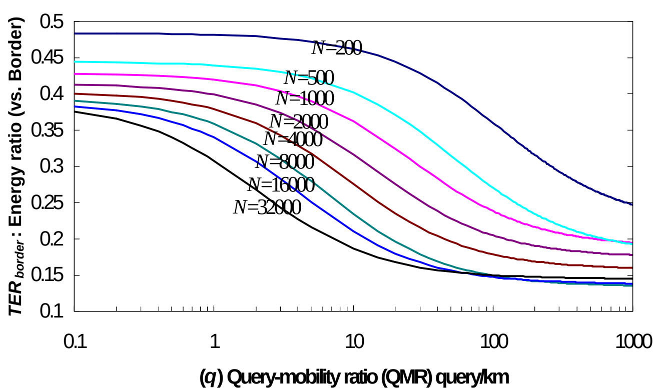

Figure 8. Total energy ratio vs. ZRP (TER_{ZRP})

[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 intra-zone exchange. Maintaining zone information without being active is not desirable.

6. Related Work

The simplest form of query/transaction is global *flooding*. This scheme does not scale well as we have shown. Expanding ring search uses repeated flooding with incremental TTL. This approach and its derivatives also do not scale well as we have shown.

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 state only for the active sources. In these schemes a search may be initiated for each new request involving flooding (or expanding ring search). These schemes may be suitable for small-medium size networks. The above protocols 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 above schemes incur excessive overhead unsuitable for our target applications.

The zone routing protocol (ZRP) [9][11][12][13][14] uses a hybrid approach, where link state is used intra-zone and on-demand routing (border-casting) is used inter-zone. A feature of ZRP is that a zone is node-specific, and no complex coordination is used. The proximity concept used in TRANSFER is similar to that used in ZRP. However, we avoid border-casting by using contacts out-of-zone. The main concepts upon which contacts were designed (short cuts to create a small world and on-the-fly contact selection) are fundamentally different than ZRP. Also, the target application is different. ZRP, as a routing protocol, attempts to find high quality routes, for prolonged transfers. In our study, we target resource discovery and small transfers, and we make a clear design trade-off between route optimality and the reduction of overall communication overhead. In section 5 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 where ZRP is used for route discovery while TRANSFER is used for resource discovery and small transfers.

Hierarchical schemes, on the other hand, e.g., CGSR [21] and [22], involve 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-based and landmark-based hierarchies rely on complex coordination and thus are susceptible to major re-configuration due to 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 re-configuration is incurred with mobility.

Related work on smart flooding is given in [17][23][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 intra-zone exchange.

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 the network map is known and geographic data is available (through GPS or other). These assumptions may not hold in our case. Also the ID of the target node must be known in GLS. By contrast, for resource discovery, the target *resource* may reside at a node with an ID *unknown* to the querier.

The algorithms proposed in [2][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.

7. Conclusion

We have presented a novel architecture for transaction 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 introduce the new concept of contacts to provide very efficient search in the ad hoc network.

This work is the first work, of which we are aware, on transaction 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
- Evaluating, in detail, the different dimensions of the design space and scalability of our protocol
- Comparing performance of our protocols against flooding and ZRP using extensive simulations over a wide array of networks and request rates

Our results show that significant savings when using our contact-based techniques. For large networks and high request rates, TRANSFER consumes as little as 5% of flooding energy and 14% of border-cast energy. The study also shows reasonable settings of parameters that work well for a wide range of network sizes (from 200-32000 nodes).

References

[1] D. Watts, S. Strogatz, “Collective dynamics of ‘small-world’ networks”, *Nature* 393, 440 (1998).

[2] J. Kleinberg, “Navigating in a small world”, *Nature*, 406, Aug. 2000.

[3] C. E. Perkins, P. Bhagwat, Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers, Comp. Commun. Rev., Oct. 1994, pp. 234-44.

[4] C. E. Perkins, E. M. Royer, Ad-hoc On-Demand Distance Vector Routing, Proc. 2nd IEEE Wksp. Mobile Comp. Sys. And Apps., Feb. 1999, pp. 90-100.

[5] D. B. Johnson, D. A. Maltz, Dynamic Source Routing in Ad-Hoc Wireless Networks, Mobile Computing, 1996, pp.153-181.

[6] P. Guangyu, M. Gerla, X. Hong, “LANMAR: landmark routing for large scale wireless ad hoc networks with group mobility”, MobiHOC ’00, p. 11-18, 2000.

[7] J. Li, J. Jannotti, D. Couto, D. Karger, R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing", ACM Mobicom 2000.

[8] L. Blazevic, S. Giordano, J.-Y. Le Boudec “Anchored Path Discovery in Terminode Routing”. *Proceedings of the Second IFIP-TC6 Networking Conference (Networking 2002), Pisa, May 2002.*

[9] M. Pearlman, Z. Haas, “Determining the optimal configuration for the zone routing protocol”, IEEE JSAC, p. 1395-1414, vol. 17, 8, Aug 1999.

[10] A. Helmy, "Architectural Framework for Large-Scale Multicast in Mobile Ad Hoc Networks", *IEEE ICC ’02.*

[11] Z. Haas, M. Pearlman, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks", IETF Internet draft for the Manet group, June '99.

[12] Z. Haas, M. Pearlman, "The Performance of Query Control Schemes for the Zone Routing Protocol", ACM SIGCOMM '98.

[13] Z. Haas, M. Pearlman, “ZRP: A Hybrid Framework for Routing in Ad Hoc Networks”, Book Chapter in Ad Hoc Networks, Editor C. Perkins, Addison Wesley, pp. 221-254, 2001.

[14] Z.J. Haas, M.R. Pearlman, "The Perforamnce of Query Control Schemes for the Zone Routing Protocol," ACM/IEEE Transactions on Networking, vol. 9, no. 4, pp. 427-438, August 2001.

[15] P. F. Tsuchiya, "The Landmark Hierarchy: A new hierarchy for routing in very large networks", CCR, Vol. 18, no. 4, pp. 35-42, Aug. 1988.

[16] S. Kumar, C. Alaettinoglu, D. Estrin, “SCOUT: Scalable object tracking through unattended techniques”, ICNP 2000.

[17] J.J. Aceves, M. Spohn, “Bandwidth-Efficient Link-State Routing in Wireless Networks”, Book Chapter in Ad Hoc Networks, Editor C. Perkins, Addison Wesley, pp. 323-350, 2001.

[18] A. Helmy, "Small Large-Scale Wireless Networks: Mobility-Assisted Resource Discovery", ACM-LANL-NCSTRL cs.NI/0207069, Technology Research News (TRN) Journal, August 2002.

[19] Tsu-Wei Chen and Mario Gerla, "Global State Routing: A New Routing Scheme for Ad-hoc Wireless Networks" Proc. IEEE ICC'98.

[20] S. Murthy and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks", ACM Mobile Networks and App. J., Special Issue on Routing in Mobile Communication Networks, Oct. 1996, pp. 183-97.

[21] C.-C. Chiang, "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel" Proc. IEEE SICON'97, Apr.1997.

[22] J. Liu, Q. Zhang, W. Zhu, J. Zhang, B. Li, “A Novel Framework for QoS-Aware Resource Discovery in Mobile Ad Hoc Networks”, *IEEE ICC ’02.*

[23] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum et L. Viennot, Optimized Link State Routing Protocol, IEEE INMIC 2001.

[24] S. Ni, Y. Tseng, Y. Chen and J. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network", ACM Mobicom, ’99.

[25] E. Cohen, S. Shenker, “Replication Strategies in Unstructured Peer-to-Peer Networks”, ACM SIGCOMM Conf., 2002.

[26] F. Bai, N. Sadagopan, A. Helmy, "IMPORTANT: A framework to systematically analyze the Impact of Mobility on Performance of RouTing protocols for Adhoc NeTworks", *IEEE INFOCOM* 2003.

[27] A. Bestavros and J. Byers and K. Harfoush, *“Inference and Labeling of Metric-Induced Network Topologies” IEEE INFOCOM '02, June 2002.*

[28] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, H. Yu, "Advances in Network Simulation", *IEEE Computer*, vol. 33, No. 5, p. 59-67, May 2000

[29] Y. Ko, N. Vaidya, “Location-aided routing (LAR) in mobile ad hoc networks”, Wireless Networks 6, 4, p. 307-321, July 2000.

[30] B. Karp, H. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", ACM MobiCom 2000.

[31] J. Navas, T. Imielinski, "GeoCast - Geographic Addressing and Routing", ACM MobiCom ‘97.