

Efficient Resource Discovery in Wireless AdHoc Networks: Contacts Do Help

Ahmed Helmy

Department of Electrical Engineering

University of Southern California

helmy@usc.edu

Abstract- The resource discovery problem poses new challenges in infrastructure-less wireless networks. Due to the highly dynamic nature of these networks and their bandwidth and energy constraints, there is a pressing need for energy-aware communication-efficient resource discovery protocols. This chapter provides an overview of several approaches to resource discovery, discussing their suitability for classes of wireless networks. The approaches discussed in this chapter include flooding-based approaches, hierarchical cluster-based and dominating set schemes, and hybrid loose hierarchy architectures. Furthermore, the chapter provides a detailed case study on the design, evaluation and analysis of an energy-efficient resource discovery protocol based on hybrid loose hierarchy and utilizing the concept of ‘contacts’.

1. Introduction

Resource discovery is an essential element in the design of infrastructure-less networks. Many classes of multi-hop wireless networks, including mobile ad hoc networks (MANets) and sensor networks, are designed as infrastructure-less, unattended, rapidly-deployable networks. In these cases, each network node does not have prior knowledge of the resources available in the network. Hence, resource discovery is imperative to the design of ad hoc and sensor networks. In general, a resource can be any type of service or capability, such as nodes with high energy, processing power or storage, multiple interfaces, printing capability, or sensing capability. Also a resource may be shared information, as in file-sharing, publish-and-subscribe, storage-retrieval, and querying for sensed data.

Resource discovery in wireless ad hoc and sensor networks is a challenging problem mainly due to the following reasons. First, the lack of infrastructure, where there are no well-known servers in a pre-defined network structure as in DNS. Second, the highly dynamic nature of ad hoc and sensor networks that leads to frequent changes in the network topology and resource availability and locations. Network dynamics arise mainly due to the characteristics of wireless channels, in terms of fading, interference and variability

in the loss rates, in addition to possible node movement in mobile networks. Third, scarcity of energy and communication bandwidth in such networks necessitates the design of new resource discovery protocols qualitatively different from those designed for the wired Internet. In the wired Internet protocols were not designed for energy and communication constrained environments. Moreover, in many cases wireless networks may scale up to thousands of nodes rendering the discovery problem even more challenging.

Service discovery may occur between the querier and the resource node directly, according to the client-server model. Another alternative is to use the directory lookup model where the target resource node *registers* itself (or its service) in the directory. A node interested in the stored data may perform a lookup in the directory before issuing the request. In either model, an efficient distributed search protocol is necessary. The design and analysis of search protocols in wireless ad hoc networks is the topic of this chapter.

Efficiency of a resource discovery protocol may be measured using different metrics. Some of the most meaningful metrics in the context of wireless ad hoc networks include communication overhead, energy consumption, delivery success rate and response delay. In general, *bandwidth* in wireless networks is a scarce resource. In addition, due to the broadcast nature of the wireless medium, collisions, due to medium contention, may further decrease the throughput of the network. Furthermore, nodes in ad hoc networks are, in general, battery powered and hence are energy constrained. Communication is a main consumer of energy, during both transmission and reception of packets. Hence, reducing communication overhead and overall energy consumption is one of the main goals of ad hoc networking protocols.

There are several design approaches for distributed search protocols that may apply to resource discovery. In general, a resource discovery architecture in wireless ad hoc and sensor networks may be categorized as ‘location-aware’ or ‘location-free’. Location-aware architectures require that node (or resource) location information be widely available in the network. Such

architectures usually use geographic or trajectory routing to forward the updates or queries and include geographic rendezvous mechanisms (such as GLS [7], Rendezvous Regions [55] [39] [53], and GHT [33]) and trajectory advertisement schemes (such as TBF [54], and [56] [57]). These architectures may be efficient when (and if) location information is available. On the other hand, location-free architectures do not require or use location information. In this chapter we shall present and study location-free resource discovery architectures, and will not be discussing location-aware architectures.

Here we provide a simple classification of the main common location-free search techniques, and provide an overview of each of those techniques, along with a discussion on their applicability and performance in ad hoc networks. Specifically, an overview is given for the following schemes.

1. Flooding-based techniques, including (a) simple flooding (or expanding ring search), (b) scoped flooding and (c) efficient reduced broadcast,
2. Ad hoc on-demand routing (with caching),
3. Hierarchical architectures, including (a) cluster-based hierarchy, (b) landmark hierarchy (c) dominating set (or backbone) approaches, and
4. Hybrid (loose hierarchy) routing, including (a) zone routing and (b) contact-based architectures.

This is not meant to provide an exhaustive list for the large body of literature that exists in this area. Rather, it is a representative subset of the main approaches. Then we shall present an elaborate discussion of the design and evaluation of the contact-based architecture as an illustrative case study and as a promising architecture for efficient resource discovery in large-scale ad hoc networks. A comparative simulation study is given at the end of the chapter.

2. Overview of Resource Discovery Approaches

We address the problem of resource discovery in *infrastructure-less* wireless mobile ad hoc networks. Hence, architectures that require infrastructure (e.g., DNS) are not suitable for our problem. Centralized approaches are neither robust nor scalable.

Perhaps one of the simplest search schemes is flooding. Flooding each request throughout the network enables the search for the resource sought in every node in the network. Simple flooding causes every node to forward the request packet until all the nodes in the network receive it. In wireless networks, flooding leads

to what is called a ‘broadcast storm’ [23] with many packets sent in a relatively short period, potentially leading to collisions and congestion. Several variants of flooding attempt to alleviate the expensive cost of simple flood. Scoped-flooding (or expanding ring search) attempts to perform the search in expanding rings and stops when the request is satisfied. Reduced broadcast techniques use heuristics to suppress flooding redundancies.

Several on-demand ad hoc routing protocols build upon flooding to obtain up-to-date routes, but use caching to take advantage of the history of route requests to avoid unnecessary flooding. The efficacy of the cache depends on the cache spread, which in turn affects the cache hit rate, and the cache validity ratio. The cache hit and validity ratios depend on dynamics of the cached information. In case of highly dynamic scalable networks the cache efficacy may decrease, forcing more request flooding.

To provide a scalable architecture, some approaches use a hierarchical scheme based on dynamically formed ‘clusters’. The clusters are re-formed with topology dynamics due to node failure and mobility. One common method to form clusters is to elect a dominant set of nodes that reach all other nodes in the network within 1 hop. This scheme attempts to guarantee request reception at each node in the network for every request.

A hybrid approach is used in the zone routing protocols, in which a proactive (push) mechanism is used intra-zone and a reactive (pull) mechanism is used inter-zone. In this approach each node establishes its zone independently forming a loose hierarchy. Flooding requests between the zone periphery nodes (called borders) is used in the inter-zone reactive phase in a mechanism called bordercasting. Bordercasting attempts to achieve full coverage of the network by querying the border nodes. The contact-based architectures use the concept of zones, but instead of using bordercasting, only contact-nodes that lie out-of-zone are elected and queried on-the-fly. Contacts may be elected in an energy-aware fashion to achieve load balancing and to extend the network lifetime¹.

3. Resource Discovery Approaches

3.1. Flooding-based Approaches

Flooding is a simple, commonly used technique for resource discovery. In simple flooding a message is sent

¹ Several other resource discovery systems have been developed for MANets including [58] [59] [60] [61]. For brevity we do not discuss those systems.

to all nodes in the network with every discovery request. A mechanism to prevent loops is usually employed, where each node stores the request and querier IDs and forwards the query only once. In general, if there are N nodes in the network then each node gets to transmit the packet except the target node; i.e., $N-1$ transmissions. Over wireless channels when a node transmits a packet during flooding, it is broadcast to all neighbors of that node. If there are on average g neighbors per node – where g is called the average node degree – then we get $g.(N-1)$ receptions per flooding. Put differently, for every link AB (between nodes A and B) the flooded message will traverse twice; once when the node A sends to B and again when B rebroadcasts, so number of receptions is $\sim 2L$, where L is the number of links in the network. Energy is expended in both transmissions and receptions. As one might expect, it is usually not necessary for every node in the network to transmit the packet in order to achieve full network coverage (in which all nodes in the network receive the packet). Hence, simple flooding may lead to unnecessary redundant transmissions. For frequent requests flooding may incur significant communication overhead, and hence may be undesirable for wireless networks.

Expanding ring search (ERS) techniques (or scoped-flooding) are also commonly used for discovery. *ERS* uses repeated flooding with a limit on the number of flooding hops (*TTL*). If the resource is not found in a trial, then this hop limit (*TTL*) increases for subsequent trials by an *increment*. A simple version of *ERS* may use a constant increment of ‘1’. Variants of *ERS* may be derived, however, by varying the increment, either to a constant greater than 1 or to a variable such as exponential in the number of trials. *ERS* usually performs better than simple flooding when the resource is relatively close to the querier. However, *ERS* may be quite inefficient when the querier-target pairs are randomly distributed the network diameter is high, which tends to be true in many wireless networks due to clustering of nodes.

Reduced broadcast techniques [23] use heuristics to reduce the redundancies of flooding and conserve communication. The main idea is to exploit node density to reduce redundant transmissions, sometimes at the expense of reducing 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 where the wireless network is not highly dense, the effect of reduced broadcast is quite limited.

The aim of these heuristics is to deliver ‘*broadcast*’ messages that should be received by all nodes in the network; a goal sometimes different from that of resource discovery. Some broadcast reduction heuristics include (i) probabilistic forwarding, (ii) count of message receptions, (iii) node distance, or (iv) node locations.

In probabilistic flooding, each node rebroadcasts the message with probability p . When $p=1$, the scheme degenerates into 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. So, 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 $x > C_{th}$ then the rebroadcast is suppressed.

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 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 $d < D_{th}$ then the broadcast 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 achieved without affecting message reachability significantly. It also works well for various node distributions, but requires location information. The counter-based scheme is simple (does not require location information) and achieves noticeable reduction in number of redundant rebroadcasts if the network is dense (where the broadcast redundancy is high).

3.2. Ad Hoc On-demand Routing

Routing protocols in ad hoc networks can be generally classified as proactive (table-driven) or reactive (on-demand). Previous studies [43] have shown that on-demand routing protocols with caching (e.g.,

DSR [5], *AODV* [4]) achieve better throughput and overhead performance better than proactive protocols in terms of throughput and overhead especially with node mobility. On-demand protocols employ mechanisms for route caching to avoid unnecessary flooding of route requests and robust route repair mechanisms to deal with frequent route breaks.

Performance studies of ad hoc routing that usually use *long-lived* assigned connections mainly capture the *route repair* phase of the routing protocols, while the route discovery phase is invoked only during the initial setup period of the connections. However, *resource discovery* protocols may exhibit significantly different behavior, where the route discovery/setup phase (and caching) becomes the dominant factor affecting performance while route maintenance/repair is very rarely triggered.

Caching is used in on-demand (reactive) ad hoc routing [5][4] to alleviate the effects of flooding. Caching can be quite efficient, when the cache hit ratio is high and the cache is valid. The effectiveness of this approach thus depends on the cache validity, which in turn is a function of the network and information dynamics and request patterns. In cases of mobile networks and frequent un-correlated short requests, however, one might expect the cache performance to degrade as the on-demand routing degenerates with high probability into flooding.

To gain better understanding of the effect of caching on protocol overhead, we build a caching model for the on-demand *DSR* approach and evaluate it. Earlier studies on on-demand ad hoc routing protocols investigated relatively small (40-100 node) networks, with long lived connections [5] [40] [41] [42] [43] [44]. We instead study cache performance for larger scale networks and for resource discovery.

3.2.1. Caching model

Our caching model follows the dynamic source routing (*DSR*) design [5] [40]. A source looking for a target (or a destination) triggers a route request (*RREQ*) on demand. The process of route request is illustrated in Figure 1. First, the source (Q) looks up its own local

cache for a route to the target (T). If local cache is not found, then the source sends a query to its first hop neighbors and they perform cache lookup. If a cached route is not found, or if the found cache does not result in positive response from the target (e.g., due to invalidity of the cache), then the source floods the route request throughout the network. The target replies to requests from distinct neighbors to create multi-paths, and intermediate nodes with cached route to T , respond to the request. The replies (from the target and intermediate nodes) traverse the reverse pathes to the source, and nodes along the route (and their neighbors) cache the route information (i.e., aggressive caching). When a cached route is used and is found to be invalid or out-of-date, it is attached to the flooded route request to invalidate all copies of that route in the network.

Now, let us define the *cache hit ratio* as the fraction of cache lookups that are found, and the *cache validity ratio* as the fraction of cache hits that are valid. Let the local cache hit ratio be p_1 , and the local cache validity ratio be q_1 . Similarly, let neighbor cache hit ratio be p_2 , and the neighbor cache validity ratio be q_2 . Furthermore, let the validity ratio for the routes obtained and used by flooding be q_r . Note that if the flooded request reaches the target T (i.e., destination) causing a reply to be send to Q , and if that reply is used, then it should be valid. There may be other situations where nodes en-route to the destination have a cached route for T , in which case the flooded request may not reach T , and in which case there is a probability that the replies may be invalid. In these situations another request may be flooded to obtain a valid route. The state diagram for this process and the state transition probabilities are given in Figure 2.

From the previous figures we can get an estimate of the overhead (or cost) incurred. For now we define cost in terms of requests and replies transmitted and forwarded. In later analysis we shall study the cost in terms of energy consumption. The three main procedures of the route discovery mechanism are I. local cache lookup, II. neighbor cache lookup, and III. flooding the request.

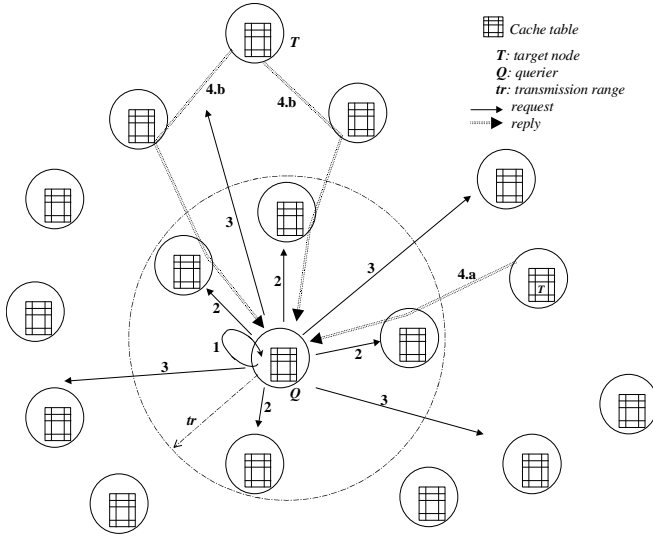


Figure 1. DSR-like routing model: a querier node Q issues a request for a target resource T . The request process progresses as follows: 1. Q performs local lookup, 2. If a cached route to T is not found then request a lookup from 1-hop neighbors (within transmission range), 3. If a cached route is not found (or is invalid) then flood a request throughout the network, 4.a. Intermediate nodes with cached route to T reply to Q , 4.b. The target T replies to requests returning multiple paths to Q .

I. Local cache lookup does not incur any communication overhead.

II. Neighbor cache lookup incurs one request transmission and replies from neighbors with a route cache. If g is the average node degree, and α is the average fraction of neighbors that respond to the request ($0 \leq \alpha \leq 1$), then number of replies is $\alpha.g$. Let the cost of neighbor cache lookup be NC , then $NC=1+\alpha.g$.

III. Request flood incurs $\sim(N-1)$ request transmissions, and replies from the target T for multiple routes equal to the number of T 's neighbors forwarding the request; i.e., $\sim g$. Each reply traverses L hops from T to Q , where L is the average path length in hops between the querier and the target. The flood will also trigger replies from $\beta.N$ intermediate nodes, each traversing $\gamma.L$ hops, where β is the average fraction of intermediate nodes responding to the request and γ is the fraction of the average path length for path lengths for the path from Q to an intermediate node. Let the cost of flooding be CF , then we get

$$CF \sim (N-1) + g.L + \beta.N.\gamma.L$$

As was mentioned previously, there may be a need for repeated floods if no valid route is obtained. Assuming, for simplicity, that the subsequent floods are independent with each having an average of qr success rate, then we get a geometric distribution and the overall (expected) cost of flooding becomes CF/qr .

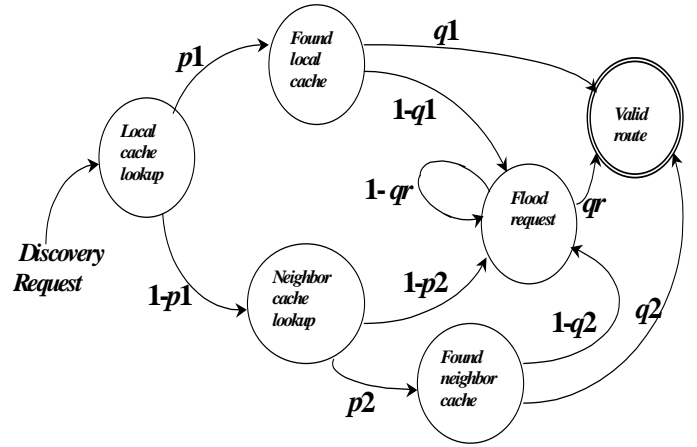


Figure 2. State diagram for the probabilistic decision chain for on-demand routing in DSR-like mechanisms.

From the state machine we get an expression for the overall overhead (Ov) per discovery for on-demand routing, as follows.

$$Ov = (1-p1).NC + (1-(p1q1+p2q2-p1p2q2)).CF/qr$$

3.2.2. Simplifications for resource discovery

Several minor modifications may apply to ad hoc on-demand routing for resource discovery. Since the resource discovery request will be sent in one-shot (i.e., in a very short transfer) there is no need to store multiple paths for future route maintenance, and only one route to the target T is needed. Hence, we can eliminate the multiple responses from the target and the responses from the intermediate nodes, thus reducing the discovery overhead. This would result in $\beta=0$, $qr=1$ and $CF=(N-1)+L$. If we consider medium to large scale networks where $N \sim 100$ to 1000 nodes or more, and consider a reasonable range for $g \sim 6$ to 12 neighbors, then we can ignore NC with a small margin of error. Also, if we combine effects of local and neighbor cache lookups simply as p for the cache hit ratio and q for the cache validity ratio, such that $p.q = p1q1+p2q2-p1p2q2$, we then get:

$$Ov \sim (1-pq).(N+L)$$

It is clear that the overhead depends heavily on the cache hit and validity ratios. Let us define the cache efficacy ($CE=pq$) as the combined effect of cache hit and validity ratios. But what are the main factors

affecting CE ? In the following subsection caching performance is evaluated and analyzed over a variety of network size and mobility degrees to attempt to extract those factors.

3.2.3. Simulation and Analysis of Caching

We observe the performance of the on-demand routing caching scheme through simulation. Networks with 40 to 2000 nodes were simulated with various mobility degrees. Each node moves using a “random waypoint” model with no pause time. Each node selects a random value between $[0, V_{max}]$ m/s and a random destination to which it moves with constant speed. This process is repeated whenever a node reaches its destination. V_{max} was varied from 0 to 60m/s. Querier-target pairs were chosen randomly and 1000 queries were performed in each run with 10 queries per sec. A cache warm-up period was allowed before measurements were taken in each run. Each data point represents an average of 10 simulation runs with

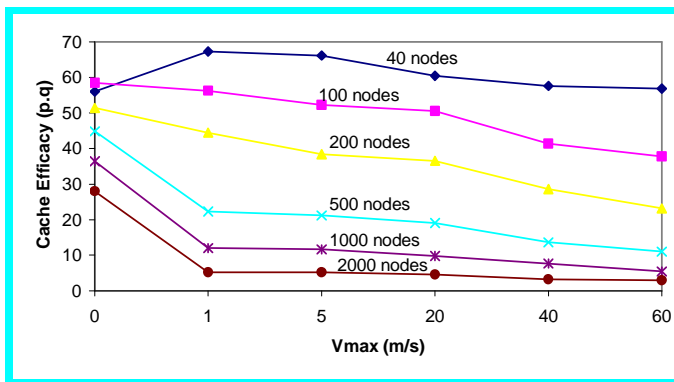


Figure 3. The cache efficacy with various velocities and various network sizes. The cache performance degrades drastically with scale of the network and with (even very low) mobility.

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.

3.3. Hierarchical Approaches

For scalability, several *hierarchical* approaches have been proposed [20][21][45]. Many hierarchical architectures are *cluster-based*, in which each cluster (or group) of neighboring nodes elects a cluster-head to relay traffic to the other clusters. One advantage of this approach is that a request may be forwarded to the cluster heads (via intermediate nodes) without flooding the whole network. A disadvantage, however, is that a

different random seeds. The average node degree was kept almost constant for all topologies at 8 neighbors per node, in 1kxm1km network.

The results are given in Figure 3 and Figure 4. Figure 3 shows the cache efficacy for various mobility degrees and network sizes. For very small scale networks (40-100 nodes) the efficacy is relatively high (~50-70%) especially for low mobility cases. As the number of nodes increases, however, the cache efficacy drops dramatically, even for very low mobility, to ~10% for 1000 nodes and to ~5% for 2000 nodes.

Figure 4 gives a closer look at the cache metrics. It is apparent that the effect of network size is more significant than the effect of mobility. The cache hit ratio (p) drops from ~73% (for 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.

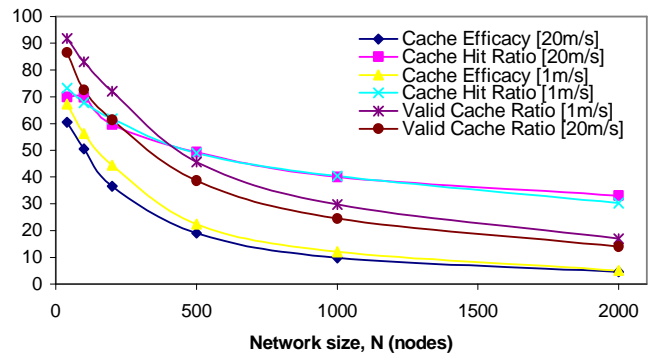


Figure 4. Cache hit ratio (p), valid cache ratio (q) and cache efficacy (p,q) with the network size for 1m/s and 20m/s.

cluster-head may become a single-point-of-failure or a point of traffic concentration and a potential bottleneck. The **landmark hierarchy** [6][14][15] uses landmarks as directions for routing, but does not use landmarks as communication relays between clusters. This alleviates traffic concentration at the landmarks and reduces the risk of single-point of failure. However, the highest level landmark needs to periodically flood its information throughout the network, which may be unsuitable for wireless networks. One major concern in these hierarchical approaches is their reliance on complex coordination mechanisms, for election, promotion and demotion. Hence, in highly dynamic environments, such as wireless ad hoc networks, complex hierarchical approaches are susceptible to major re-configuration with node failure, sleep

schedules and mobility, leading to serious performance degradation.

One common method used for clustering is based on **dominating sets** (*DS*) [24][25][46][47]. A dominating set 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 the set. The problem of finding the minimum dominating (*MDS*) set has been proven to be NP-complete. Several heuristics have been proposed to approximate the optimal solution. The proposed solutions provide various trade-offs between the establishment (and maintenance) of the

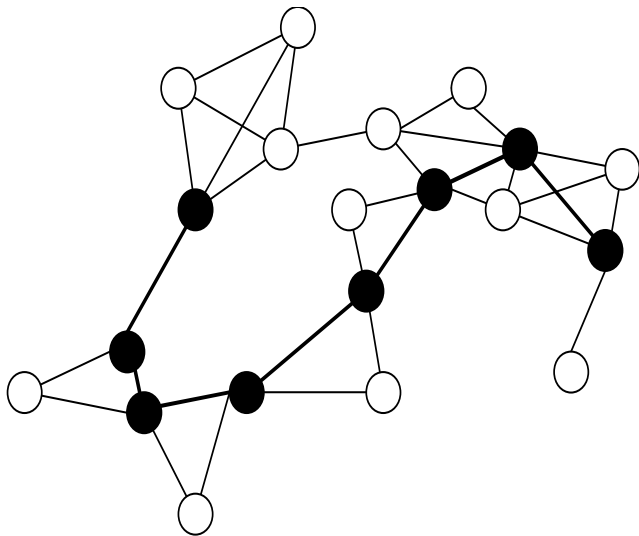


Figure 5. Example connected dominating set, black nodes belong to the backbone constituting a dominating set.

In general, *DS*-based approaches may operate in two-steps or one-step (on-the-fly). The two-step protocols (e.g., [47]) conduct the *DS* election (or backbone setup) in one phase, then use another phase to perform the discovery. Because the two-step protocols setup and maintain a backbone they are subject to re-configuration with mobility. On the other hand, the single-step protocols, such as dominant pruning as in [24][25], conduct the dominating set election while performing the discovery, and hence are more resilient to mobility effects.

In general, cluster-based approaches using dominating sets attempt to achieve complete network coverage by ensuring that every node receives every request. These approaches may be quite effective when broadcast redundancy is high (i.e., in highly-dense networks). For networks in which node distribution is not highly-dense, however, the effects of overhead

dominating set (sometimes called backbone) and the cost of broadcast. A good survey on these schemes is provided in [46]. Some protocols use a connected dominating set (*CDS*) as shown in Figure 5, while others propose a weakly-connected *DS* (*WCDS*) as shown in Figure 6. *WCDS* schemes lead to a smaller number of clusters and fewer nodes in the backbone than *CDS* schemes and thus incur less overhead to setup. Dominating set approaches, in general, guarantee (in theory) that the network will be fully covered. By contrast, the heuristics used for reduced broadcast cannot guarantee full network coverage.

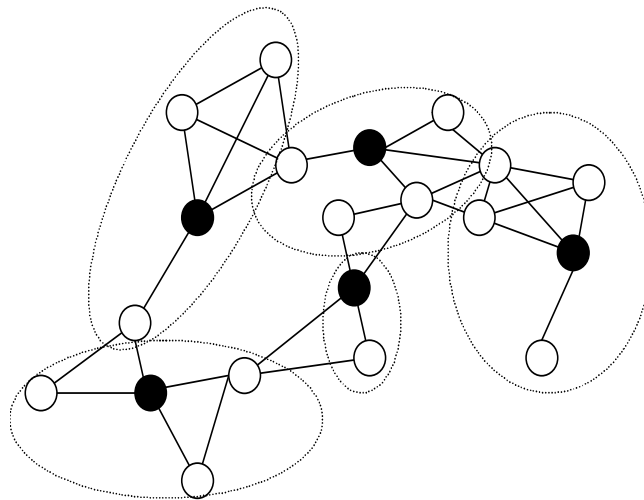


Figure 6. Example weakly connected dominating set

using these approaches is quite limited, and perhaps hybrid approaches should be investigated.

3.4. Hybrid, Loose Hierarchy

Hybrid schemes use a mix of proactive and reactive schemes in an attempt to establish a reasonable overhead balance. Nodes exchange periodic and triggered information with neighbors within a limited number of hops to form *zones*. The proactive protocol is used for intra-zone updates, while reactive, on-demand, request forwarding is used for inter-zone discovery. The zones form a *loose* hierarchy because each node maintains its zone independently, and no major re-configuration is incurred with mobility. Unlike dominating set schemes, hybrid approaches, in general, do not deliver every request to all nodes. Rather they attempt to achieve complete network information coverage by forwarding the request to nodes that (collectively) have information about network routes or

resources. In this section we discuss two hybrid approaches (a) the zone routing protocol, and (b) the contact-based architectures.

3.4.1. The Zone Routing Protocols (ZRP)

The zone routing protocol (ZRP) [9][10][11][12][13] uses a hybrid approach, where link state is used intra-zone and on-demand routing (border-casting) is used inter-zone. A feature of zone routing is that, unlike cluster-based approaches, a zone is node-specific, and

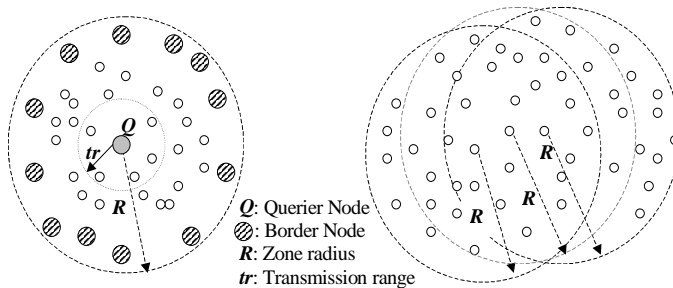


Figure 7. Concept of zone radius, R (in hops), border nodes, and per-node zone in hybrid routing with loose hierarchy

In ZRP the querier, Q , sends the request to its borders, and the borders send it to their borders, and so on. Query control mechanisms are used to reduce redundant querying. Requests are broadcast (or multicast) hop by hop and are recorded by nodes along the path. Query detection mechanisms (called QD-1 and QD-2) specify that intermediate nodes along the forwarding path (and their neighbors) record the request information. Upon receiving a request sent to a border that has been previously visited, the intermediate node terminates such request. The intermediate node has knowledge of the previously visited borders in its zone by maintaining intra-zone information of up to $2R-1$ hops (called the extended zone). Hence, the redundant request can be terminated early. This scheme is called early termination (ET).

Illustration of the basic operation of ZRP and bordercasting is given in Figure 8.

ZRP design attempts to achieve an overhead balance between intra-zone maintenance and inter-zone discovery by varying the zone radius R , which in turn varies the amount of information to store in each node. The optimal value of R depends on the network size and dynamics. More specifically, it depends on the ‘call-to-

no complex coordination is used for cluster-head or landmark election. In zone routing each node transmits its information up to R hops away, where R is called the zone radius. Based on this message transmission, each node independently collects information from its neighbors and forms its own view of the network, called a *zone*. Nodes at exactly R hops away are called *border nodes*. Figure 7 shows the concept of a zone and independent zone construction by network nodes.

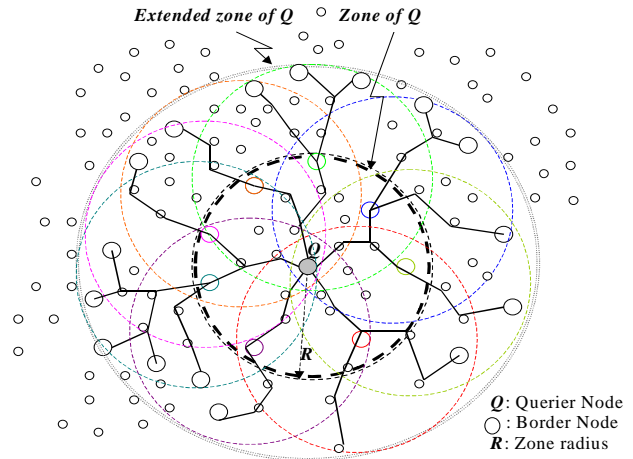


Figure 8. ZRP bordercasting with query early detection and control, various zone for the border node of Q are shown using different colors for illustration. Bold lines indicate the paths traversed by the bordercast messages

mobility ratio’ that in essence captures the amortization factor for the zone maintenance cost (due to mobility) over the number of call requests issued. It is during those bordercast requests that ZRP provides savings over flooding.

3.4.2. Contact-based Resource Discovery Architectures (CoRDs)

Contact-based resource discovery architectures (CoRDs for short) employ the hybrid approach and use loose zone-based hierarchy. However, the requests are forwarded to only a small number of selected nodes out of the zone, called *contacts*. Instead of using shortest path or optimal routes, CoRDs design goal is to conserve network bandwidth and energy, while achieving high request success ratio. CoRDs protocols avoid border-casting by using contacts out-of-zone. The concept of *contacts* is key for efficient discovery in these schemes. The idea behind the contacts borrows from *small worlds* [1][17]. Unlike relational or random graphs, wireless networks are spatial graphs (in which links are a function of distance, among other factors) that tend to be highly clustered, leading to very high degree of separation. For a node, contacts are a few

nodes outside of the zone 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 is made, a contact-selection protocol is invoked. Contact selection mechanisms aim to reduce zone-overlap and to elect contacts that increase the coverage of the search. The search proceeds according to several possible policies; single-shot, or multiple-expanding trials.

Several protocols have been designed for contact-selection, including *CARD* [48][49], *MARQ* [50], and *TRANSFER* [51]. The first two architectures, *CARD* and *MARQ*, use a pro-active approach that selects and maintains contacts. *CARD* uses zone-edge information to select useful contacts, while *MARQ* exploits mobility by choosing contacts moving away from the zone. The third architecture, *TRANSFER*, on the other hand, uses a re-active approach, by choosing contacts dynamically, on-the-fly, when the request is issued. The reactive nature of this protocol reduces the maintenance overhead and is more resilient to network dynamics.

In the rest of this chapter the contact-based architecture will be presented as an elaborate case study for the design, evaluation and analysis of an efficient resource discovery protocol for large-scale ad hoc wireless networks. The contact selection mechanism presented use the dynamic contact selection mechanism, as in *TRANSFER* [51].

4. Contact-Based Resource Discovery Architectural Overview and Design

In the contact-based 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 *zone* of a node. The zone is maintained using a proactive localized link state protocol. Each node chooses its zone 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. A neighbor discovery protocol is used by which each node identifies nodes 1 hop away (through periodic beacons). The link state protocol provides neighbor information to other nodes in the zone. Typically the number of nodes in the zone is small (less than 100 nodes). As part of the zone information each node keeps routes to nodes and pointers to resources in its zone. Nodes exactly R hops away are called *borders*.

When a querier node Q (potentially any node in the network) issues a resource discovery request, it first checks to see if the resource (or destination) is in its zone. If not, then it seeks the assistance of a number of

contacts (NoC) outside the zone, 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. C is called a *contact* node. To increase the search efficiency, C should have low zone overlap with Q . Proper setting of the parameter r helps to reduce such overlap. Contact nodes act as short cuts that bridge between disjoint zones. 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 from the querier node to the target.

The main architecture is shown in Figure 1, where the querier node Q chooses three of its borders, $B1$, $B2$, $B3$ to which to send a request message. Each of the borders in turn chooses one contact at r hops away to which to forward the request. $C1$, $C2$, and $C3$ represent the contacts. The number of borders (and subsequently contacts) chosen, NoC , and the contact distance (r hops) are design parameters. 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 zone radius (R), will be presented in the evaluation section. First, we describe the contact selection scheme.

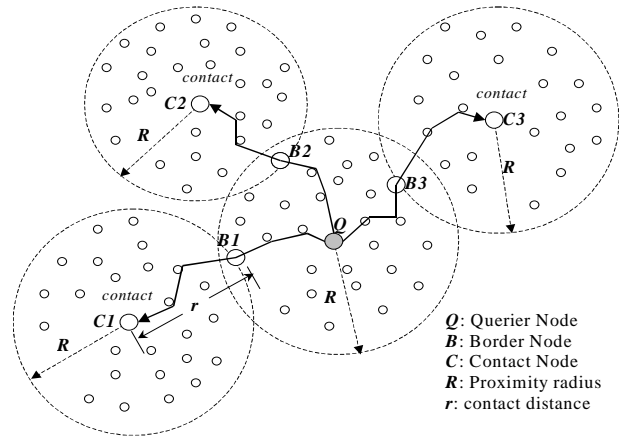


Figure 9. Each node in the network has a zone of radius R hops. A querier node, Q , sends a request through a number of its borders equal to the number of contacts (NoC), in this case $NoC=3$. Each border node, B_i , chooses one of its borders, C_i , to be the direction for forwarding the request r hops further until it reaches the contact. The contacts are at most $(R+r)$ hops away from Q . In this example $r=R=3$.

4.1. Contact Selection and Search Policies

This section describes the contact-selection protocol and the notion of levels of contacts. Then presents various search policies.

4.1.1. Contact Selection Protocol

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 zone that does not overlap significantly with that of the querier node, Q , or the other contacts of Q . However, contacts do not know about each other, and do not know their shortest distance to the querier (remember that contacts are outside of the querier's zone). To address this problem an algorithm is used to reduce zone overlaps, thus increasing coverage and reducing search overhead.

The first kind of overlap occurs between the contact's zone and the querier's zone. To reduce this overlap the request is directed out of the querier's zone. One simple approach to try to achieve this is for the border node to randomly choose one of its borders to which to forward the request. This, however, often leads to significant overlap with the querier's zone rendering the contact ineffective and the query success rate becomes low. Another simple approach is for the border node to avoid sending the request through the node from which it was received. However, wireless networks have a high clustering coefficient² [1][17]. This means that, on average, there is relatively high probability that the neighbors of a neighbor of B are also neighbors of B . Therefore, it is not sufficient to avoid only the previous hop since there may still be a good chance that the border may forward the request through nodes that belong to Q 's zone. This is illustrated in Figure 2 (a), where the border node B receives the request from node L (the previous hop), and forwards it to contact $C1$ through node x . Node x is a neighbor of node L and is within Q 's zone, and hence would lead to a contact less than $R+r$ hops away. In many cases the contact chosen this way may have a zone heavily overlapping with Q 's zone.

The problem in forwarding the request outside of Q 's zone to a useful contact is the loss of direction for the forwarded message at the border of the zone (since Q knows only about nodes R hops away). To achieve a sense of direction without location information, a mechanism is designed that uses information about the neighbors of B 's previous hop, L , as explained next.

A querier node, Q , sends a request to NoC of its borders. Consider one of those borders, B . Let node L be the last hop before B on that path. Note that B is R hops away from Q , and L is $R-1$ hops away from Q . All

L 's neighbors (including B) are 1 hop away from L , and hence are at most R hops away from Q . That is, all L 's neighbors are within Q 's zone. As was mentioned before, due to high clustering many of L 's neighbors (all of which are in Q 's zone) may also be B 's neighbors. Hence, B should attempt to avoid forwarding the request through any of L 's neighbors. As illustrated in Figure 2 (b), B avoids L 's neighbors (x,y,z) and is able to forward the request to a contact, $C2$, that has significantly less zone overlap with Q than $C1$ does. This scheme is called the *zone overlap reduction (POR)* scheme. Note that the above examples used $r=R$ for illustration. 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, this is called 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 $\lceil r/R \rceil$ times at each chosen border.

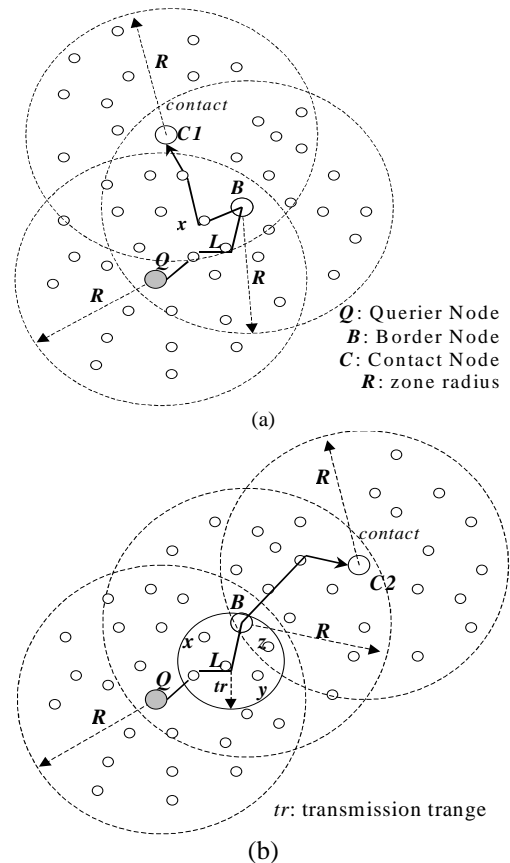


Figure 10. (a) The border node, B , forwards the request towards its border $C1$ via node x . $C1$'s zone has significant overlap with Q 's zone. By only using random forwarding or avoiding only node L (the previous hop) B can easily lose sense of direction and choose a poor contact. (b) By using neighbor information of L , B avoids forwarding the request to L or any of its neighbors (x,y,z) , all of which are in Q 's zone. Hence, B is more likely to choose a useful contact, $C2$. The overlap between $C2$'s zone and Q 's zone is a lot less than overlap between $C1$'s and Q 's zones.

²The clustering coefficient (cc) measures the probability that neighbors of a node are also neighbors of each other. In [17] it was shown that for wireless networks $cc=0.58$ (high clustering) for settings similar to our study.

The second type of overlap occurs between zones of contacts. To reduce this overlap the querier node, Q , attempts to select borders to which it has disjoint routes. This is done using the zone information (with no extra overhead). If NoC borders are chosen by the end of this procedure then Q sends the request to the chosen borders. Otherwise, borders are chosen with minimum route overlap (i.e., with different 2nd hop nodes, then 3rd hop nodes, etc.). Otherwise, new borders are chosen randomly until NoC borders are chosen. This scheme does not guarantee non-overlap between contacts' zones, but performs quite efficiently during requests, as we shall show. We call this scheme the *route overlap reduction (ROR)* scheme.

The POR and ROR mechanisms are performed as part of the query forwarding process, i.e., on-the-fly. Reduced broadcast heuristics or dominating set schemes may be used to reduce the overhead of the zone maintenance.

Power-Aware Contact Selection: The contact-selection criteria can (and in fact should) take *power* into consideration. Information about power levels in nodes and rate of energy consumption may be piggybacked upon the proximity exchange messages. During the contact selection mechanisms (i.e., POR and ROR) nodes with low remaining energy are given low selection priority. To achieve this the following power-aware algorithm is used. A node that is selecting a border or contact first applies the POR and ROR rules as described above. If the selected node(s) (or any of the nodes en route to the selected node(s)) has less than a power threshold ($Pth1$) of remaining power, then the selection process is repeated for $Pth2 < Pth1$, so on, until the selection is made. Hence, this algorithm selects the most energy-capable nodes that satisfy the non-overlap rules of POR and ROR .

4.1.2. Levels of Contacts – putting the first pieces together

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 . These contacts are called *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 contacts' directions.

To select farther contacts during the query process, level-1 contacts repeat the contact selection to choose their own contacts, which we call level-2 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 . The effect of D is studied in the evaluation section. Note that in order to select level-1 contacts, Q performs ROR , whereas to choose level- $i+1$ contacts, level- i contacts need to perform both POR and ROR . That is, a level- i contact selects borders with disjoint routes from its borders that do *not* pass through its previous hop L 's neighbors.

4.1.3. Search Policies – putting all the pieces together

Given a request and a number of levels, D , the target search process may proceed using different policies. Three different policies are presented for target search. The first is called *single-shot*, in which the querier sends out a request, in a single attempt, to traverse the contact levels in succession, up to D levels. The second policy is called *level-by-level (lbl)*, in which 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 $d_i=1+d_{i-1}$. Attempts continue up to $d_i=D$. The third policy is called *step search* (or simply *step*), and is similar to *lbl* except that increasing the depth occurs in steps instead of increments of 1. For our study we choose an exponential step increase; i.e., $d_i=2d_{i-1}$.

4.1.3.1. Single-shot Policy

In this policy the request is sent out from the querier node once, in a single attempt. The request is forwarded directly from level-1 contacts to level-2 contacts, up to level- D contacts. In a sense, this policy is analogous to flooding between contacts. To further clarify this policy we give a simple, first order, theoretical estimate of its overhead. These estimates are given only for illustration purposes. At each level- i , the theoretical number of contacts visited is $(NoC)^i$, and the theoretical number of hops traversed is $(R+r) \cdot (NoC)^i$. Hence, the number of transmissions is given by $[(R+r) \cdot \sum_{i=1}^D (NoC)^i]$. This

provides a theoretical upper bound. The search employs loop and re-visit prevention mechanisms, the effect of which are not considered in this simple theoretical analysis. After considering these mechanisms, the overhead may be reduced drastically. An example of the *single-shot* search is given in Figure 11 (a).

4.1.3.2. Level-by-level (lbl) Policy

In *lbl* the querier node, Q , may need to send the request several times, in multiple attempts, until the target is reached or D is reached. Starting with 1 level, the number of levels visited in each attempt d is

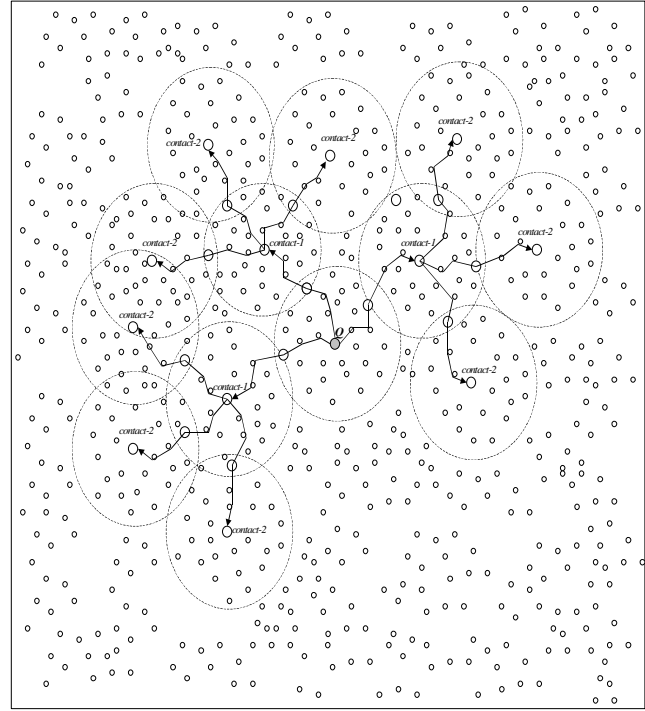
incremented by 1. If the querier does not get a positive response, it initiates another attempt³ after increasing d . Hence, the number of contacts visited in each attempt is given by $\sum_{i=1}^d (NoC)^i$, and the theoretical upper limit on

$$\text{number of transmissions is } [(R+r) \cdot \sum_{d=1}^D \sum_{i=1}^d (NoC)^i].$$

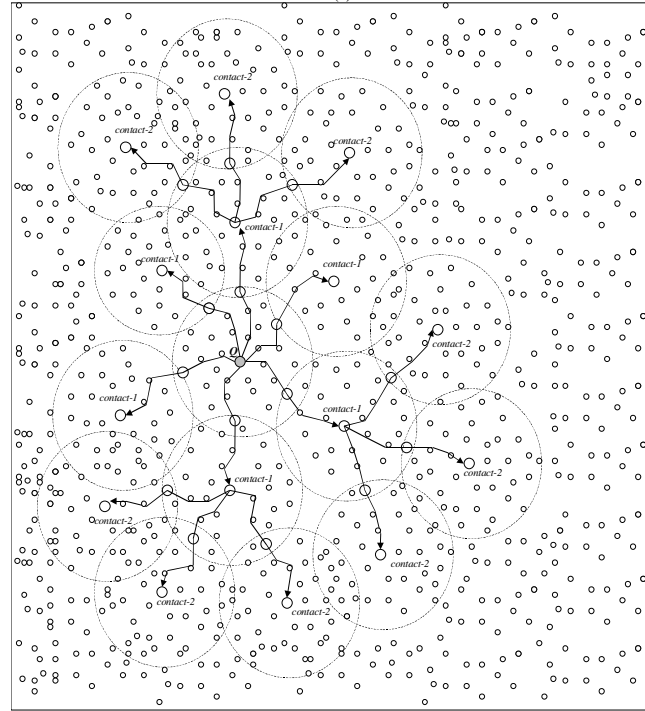
4.1.3.3. Exponential Step Search Policy

Step search is similar to *lbl*, except that the number of levels visited in attempt i , d_i , is incremented exponentially; i.e., $d_i=2d_{i-1}$ (e.g., 1,2,4,8..) until the target is found or d_{max} is reached, where d_{max} is the first d_i that satisfies the inequality $2d_{max}>D$ for $D>2$. (For $D\leq 2$, $d_{max}=D$). For example, if $D=20$ then $d_{max}=16$. For the *step* policy the upper limit on number of transmissions is given by $[(R+r) \cdot \sum_{d=1,2,4,8,\dots}^{d_{max}} \sum_{i=1}^d (NoC)^i]$.

An example of *lbl* (or *step*) with $D=2$, $R=r=3$, and $NoC=3$ is given in Figure 11 (b). Schemes *lbl* and *step* are identical for $D=2$. Note that the level-1 contacts visited on the first attempt are not necessarily similar to level-1 contacts visited on the second attempt. This is due to the randomization of the first border selection. From Figure 3 this effect is clear, and it results in different policies reaching different parts of the network. It seems, however, that *single-shot* may not reach parts of the network near the querier, but those parts are likely to be reached by *lbl* and *step* due to the randomization (rotation-like) effect, as illustrated in Figure 11 (c). Another performance implication due to the different policies is in the request latency. Intuitively, *single-shot* incurs less delivery time than the other policies because it completes its search in a single attempt. *Step* search is expected to complete its search in less number of attempts than *lbl*.



(a)



(b)

Figure 11. Examples of search policies with $D=2$, $R=r=NoC=3$: (a) The *single-shot* policy forwards the request in one attempt reaching level-1 and level-2 contacts (called *contact-1* and *contact-2*), (b) The level-by-level *lbl* policy forwards the request in multiple attempts with increasing the visited levels. In the first attempt only '3' level-1 contacts are visited. In the second attempt 3 different level-1 contacts are visited and the request is forwarded to '9' level-2 contacts. It is clear that different policies reach different parts of the network. *Single-shot* may not be able to achieve good coverage near Q with low NoC .

³ For *lbl* and *step*, the querier waits for time t between attempts; $t \propto d \cdot (R+r)$. *Single-shot* does not use t , since it uses a single-attempt per request.

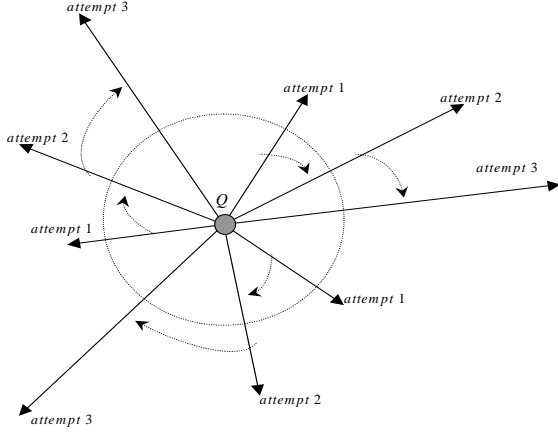


Figure 11 (c) The rotation-like effect between attempts in step and lbl increases network coverage. In lbl, attempt reaches the level contacts, so on.

4.2. Request Forwarding and Processing

The rules for processing the requests are the same for all of the above policies. This section presents the request processing, forwarding, and loop prevention.

4.2.1. The Request Message

The request message contains the target ID; the node ID or the resource key. The destination-ID in the request message 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 the querier issues new SN .

4.2.2. Loop Prevention and Re-visit Avoidance

As the message is forwarded, each node traversed records the SN , Q and P , where P is the previous hop node, 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, 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. The recorded (SN, Q, P) is kept as soft state, associated with a short timer, adding robustness against querier failure and SN wrap around. Also, if a contact reached at any level finds the querier in its own zone, indicating a loop, then the contact drops the request.

4.2.3. Search, Processing and Forwarding

A contact (or border) receiving the request, first performs a target search in its local zone information. If the target is found, the request is delivered and a response is forwarded on the reverse path (if needed),

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 message to determine which functions to perform, and whether it is a contact, two fields are included in the request message; *level-count* and the *hop-count*. Initially, the level-count is set to d and the hop-count set to $(R+r)$. Hop-count is decremented and checked at every hop:

- If hop-count reaches '0', then 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*), and sends the request to those borders.
- If the hop-count is not '0', and the current node ID is same as the destination ID of the request message, the receiving node acts as a border node. It selects a direction border (using *POR*), and sends the request towards it.
- Otherwise, the request is simply forwarded to the next hop to the destination.

The request message is unicast hop by hop, it is not broadcast hop by hop. This has an impact on the energy consumed at each hop. In broadcast, all the sender's neighbors consume energy to receive the message, whereas in unicast only the intended recipient consumes full reception energy, after a handshake for channel reservation, other neighbors may go into idle/sleep mode.

5. Evaluation and Comparison

In this section we study the various design parameters for *TRANSFER*. In addition, we compare several *resource discovery* approaches including flooding, expanding ring search⁴, ZRP, dynamic dominating set scheme and reduced broadcast.

Particularly, for *TRANSFER*, the following questions are addressed: (1) How many contacts (*NoC*) to choose? (2) What is the best contact distance (r)? (3) What should be the maximum depth (D) for the search? (4) How should we set the zone radius (R)? (5) What is the best search policy, *single-shot*, *lbl* or *step*? (6) How does replication affect the protocol performance? and (7) Is there a specific combination of settings that performs well for a wide variety of networks?

⁴ Several variants of expanding ring search - with various constant and exponential TTL increments - were studied and were found to perform worse than flooding due to the large network diameter of wireless networks. For clarity of presentation we omit results for the expanding ring search.

The main performance metrics include energy consumption due to communication overhead and the 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 a penalty is introduced for request failures. Any failure beyond an acceptable level will be recovered using flooding. Hence, the scheme used in this study is contact-based search, if failed then fallback to flooding. Since this penalty is quite expensive it will be natural for the best performing parameters to avoid resorting to flooding by achieving a very high request success rate using contacts only.

5.1. Simulation setup

Extensive network simulations are used to investigate the design space parameters and evaluate the performance of the contact-based protocols under various settings of r , NoC , and D . The overall communication overhead for the architecture is evaluated. This overhead consists of two components: (a) zone establishment and maintenance, and (b) per request (or per query) overhead. Each node keeps track of other nodes in its zone. To keep storage requirements and zone overhead at a reasonable limit, the number of nodes per zone is limited to 100 nodes. This limit is achieved for all simulated networks by setting $R=3$.

The transmission range (tr) is taken as 110m. A wide range of networks is studied, as shown in Table 1. The area of the network is varied to maintain network connectivity, and to keep the zone nodes under 100 (for zone radius of $R=3$). N nodes are randomly placed in a square of ' ℓ m x ℓ m'.

Nodes	Area (mxm)	Node Degree	Border Nodes	Zone 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 zone nodes are given for $R=3$.

For mobility the random way point model is used, where a destination is chosen randomly and a velocity is chosen randomly from $[0, V_{max}]$. Once the destination is reached, another random destination is chosen, so on. V_{max} was varied from 0 to 60 m/s⁵. In the simulations

⁵ The random way point model is commonly used in evaluating wireless ad hoc routing protocols. There are, however, recent studies [63, 64, 65, 66, 67, 68, 69, 70, 71, 72] that show the inadequacy of this mobility model to capture some important mobility characteristics. In the future we plan to evaluate our protocols using a richer set of mobility models. However, we

the energy is measured taking into account the 802.11 MAC-layer handling of both hop-by-hop transmitted and received packets.

Hop-by-Hop Communication Energy Model

The energy consumed when a request is forwarded 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. *Recipients* include the intended recipients and 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 varies 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). The amount of energy consumed during the transmission of a request packet is denoted by E_{tx} . Similarly, E_{rx} refers to the energy of request reception. If a message is broadcast, it is received by all neighbors. The average number of neighbors per node is the average node degree (g). For a unicast message, there is a small *handshake* phase to inform the neighbors of the impending transmission. In IEEE 802.11 the CSMA/CA is used with handshake and medium reservation. The handshake involves broadcast of small RTS and CTS messages causing the neighbors to *sleep* until the end of transmission. The power consumption due to handshake is denoted by E_h . Hence, the hop-by-hop energy consumption is given as follows:

- Energy consumed by a unicast message (E_u):

$$E_u = E_{tx} + E_{rx} + E_h = E_{tx}(1 + f + h),$$
 where $f = E_{rx}/E_{tx}$ and $h = E_h/E_{tx}$.
- Energy consumed by a broadcast message (E_b):

$$E_b = E_{tx} + g \cdot E_{rx} = E_{tx}(1 + f \cdot g),$$
 where g is the average node degree.

This study used $f=0.64$, and $h=0.1$ ⁶. The simulator differentiates between (hop-by-hop) unicast and broadcast messages and applies the energy model accordingly. The energy is presented in E_{tx} units.

The results of the simulations are discussed next. The first part of the results discusses the effect of r , NoC , and D on the performance of the different search

expect performance of our query protocols to be relatively robust to mobility patterns due to its reactive (on-the-fly) hierarchy establishment.

⁶ 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 (E_{tx}). This fraction depends on the transmitted packet size. A reasonable (on the high side) estimate of h is ~10%.

policies. For this set of simulations the 1000 node topology in Table 1 is used. The query overhead for *TRANSFER* is insensitive to mobility because of its *on-the-fly* contact selection and query mechanisms. The results are shown for $V_{max}=20\text{m/s}$. The second part of the results presents scalability analysis and comparisons.

Each data point represents an average of 10 simulation runs with different random seeds. 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.

First present the overhead per request (hereafter referred to as overhead per query) is presented, then the zone overhead, and finally the overall overhead.

5.2. Overhead per Query

The overhead per query is affected by the various design parameters. Here we investigate the effect of the contact distance (r), the number of contacts (NoC), the maximum depth (D), and the degree of replication. This study aims to understand the performance trends of the different *TRANSFER* policies with the various parameters.

5.2.1. Effect of contact distance (r)

Several experiments were conducted with various NoC and D . Only partial results are shown to illustrate the trend, using $NoC=3$ and $D=33$ in a 1000 node network. Figure 12 shows the effect of varying r and

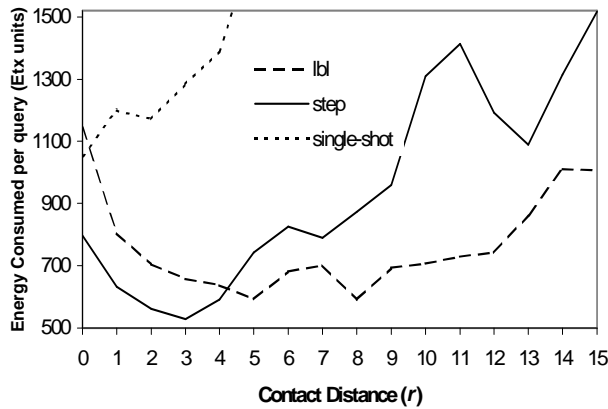


Figure 12. Effect of Contact Distance (r)

clearly indicates favorable settings for the different search policies. In general, as r grows, the contacts' location extends farther away from the querier's zone. For single-shot policy, as r increases a consistent increase in the energy consumed per query is observed. This is due to a drop in the request success rate. Remember that *drop in success rate translates into fallback to flooding*, which consistently produces more energy consumption. The drop in success rate is due to reduced coverage of areas near Q 's zone or the contacts' zones. Hence lower values of r ($0 \leq r \leq 2$) are preferred for single-shot. On the other hand, for *lbl* and *step* policies, the trend is different. Due to the contact *rotation-like* effect between attempts, *lbl* and *step* can still maintain good coverage with increasing r up to a certain distance. Hence, high request success rate is achieved with less energy consumption due to fallback to flooding. Further increase in r generally leads to more energy consumption due to drop in success rate. At very low values of r (e.g., $r \leq 2$), although *lbl* and *step* achieve high success rate, they also incur added overhead due to zone 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 the best values around 3-8 hops (3 being best for *step*, 5 and 8 being best for *lbl*).

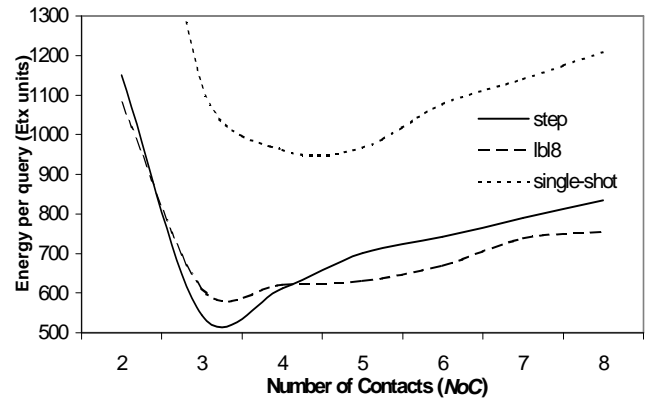


Figure 13. Effect of Number of Contacts (NoC)

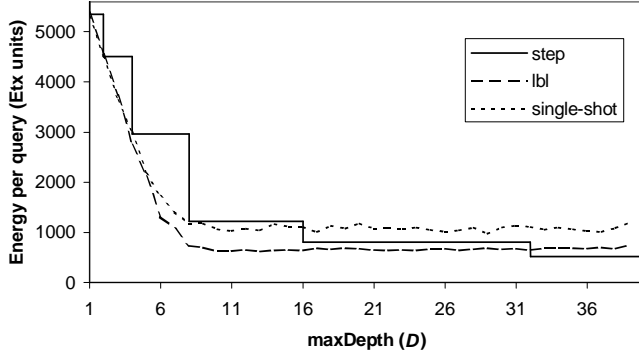


Figure 14. Effect of maximum depth (D)

5.2.2. Effect of Number of Contacts (NoC)

To understand the effects of NoC on the different policies different favorable settings of r based on our previous analysis were evaluated. Results in Figure 13 are shown for $r=2$ (for single-shot), $r=8$ (for *lbl*) and $r=3$ (for *step*). For all policies, 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 an increase in overhead is observed due to additional (unnecessary) search branches with increase in NoC . For *lbl* and *step* the best setting is at $NoC=3$, while for single-shot the best setting being $NoC=4$ mainly due to the inability of 3 contacts to establish complete coverage near Q 's zone and the contacts' zones.

5.2.3. Effect of Maximum Depth (D)

Using favorable settings for r and NoC the effect of increasing the maximum contact depth, D , is investigated. Results in Figure 14 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. Although the number of potential contacts grows, loop prevention drastically reduces the number of visited contacts. After certain values of D (10 for *lbl*, 13 for single-shot and 33 for *step*) most requests (above 97.5%) become successful and energy consumption saturates. Note that $D=33$ for *step* translates into a maximum of 6 attempts. Increase in D does not necessarily translate into increase in number of

attempts. The average number of attempts (for $D > 10$) is 3.1 attempts for *step*, 4.0 for *lbl*, and of course 1 for single-shot. For larger networks we expect this number to rise and D required for high success rates is expected rise as well.

5.3. Scalability Analysis of Total Overhead

In this section we investigate how the energy consumption grows with the increase in network size (as in Table 1). There are two main overhead components for *TRANSFER*: (a) query overhead, and (b) zone maintenance. We analyze scalability of query overhead, zone overhead, and total overhead.

5.3.1. Scalability of TRANSFER Query Overhead

Parameter setting for this experiment was based on earlier analysis. For single-shot results are presented for two settings: (a) $r=2$, $NoC=4$, (b) $r=3$, $NoC=3$. The maximum depth, D , was increased to 65 to achieve better success rate for single-shot. For *step* and *lbl* we used $D=33$ and $NoC=3$. For *step* we used $r=3$, and for *lbl* $r=8$. Results are presented in Figure 15. For all network sizes it is clear that the *step* policy achieves the best performance (with success rate of 97.5% or better without flooding for all network sizes). *lbl* incurs more overhead than *step*. *Single-shot* exhibits an interesting behavior, for $NoC=4, r=2$ it achieves between 90-96% success rate for sizes below 4000, then the success rates go above 97.5% from 4000 nodes and on, consistently rising with increase in N . For single-shot with ($NoC=3, r=3$) lower success rates (82-89%) are reached for sizes below 4000, the success rate increases to (94-97%) for 4000-8000 nodes. After 8000 nodes this setting achieves 97.5% and above success rate. It is interesting that single-shot with this (3,3) setting achieves less success rate than the previous (4,2) setting for small networks, yet there is a cross-over point at 4000 nodes after which single-shot(3,3) performs better than single-shot(4,2) and approaches performance of *step*. This is shown more clearly in Figure 16, which shows the query overhead ratio of *step* to the others. There are two reasons for this trend: a) for lower NoC single-shot incurs less overhead, b) with the increase in number of nodes there are more branches to search, giving more chance to cover, at higher contact-levels, what was not covered at lower contact-levels (near Q), thus increasing the success rate for contact-based search and decreasing fall-back to flooding.

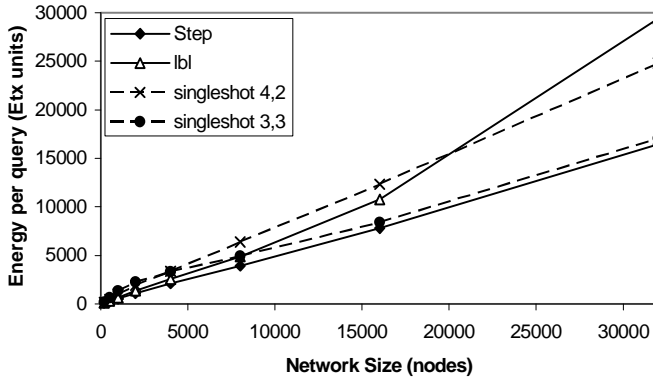


Figure 15. Scalability of query overhead for *TRANSFER* policies

Latency Analysis: Figure 17 shows the trend for average number of attempts with increase in nodes. The single-shot average is always ~ 1 , and the largest increase occurs for *lbl* (reaching 13.7 attempts for 32,000 nodes). *Step* scales reasonably well, with 5.2 average attempts for 32,000 nodes.

Based on this analysis, we observe that *lbl* provides no advantage over *single-shot* or *step*. *Step* provides the best performance in terms of energy consumption, and possesses desirable scaling characteristics in terms of overhead and latency. Single-shot exhibits the best latency among these policies and may be set to achieve good performance at higher scale. One desirable feature of *step*, however, is its persistent good performance over a wide spectrum of network sizes, with the setting ($R=3, NoC=3, r=3, D=33$ [max attempts=6]). We use these settings in the rest of the simulations.

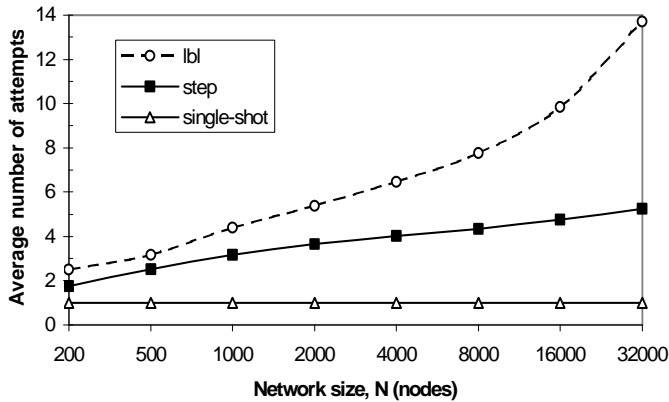


Figure 17. Average attempts per query for *TRANSFER* policies

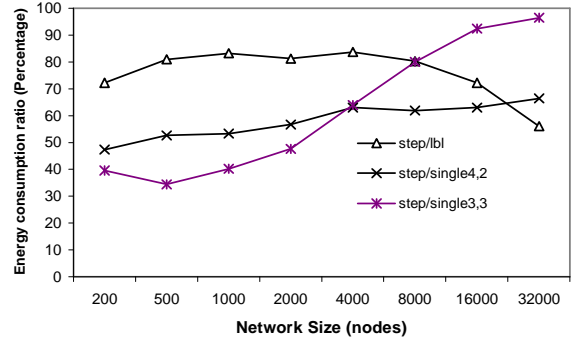


Figure 16. Query overhead of various *TRANSFER* policies relative to *step*

As mentioned before, the forwarding and processing rules are the same for all policies. Depending on the query requirement different policies may be used for different kinds of requests. This is simply achieved by setting the right parameters in the request message. For example, to implement single-shot, the querier sets the maximum level of contacts to visit (d) to the maximum depth (D) and performs a single attempt.

5.3.2. Comparison (Query Overhead)

We compare the approaches of flooding, on-demand routing with cache, i.e., DSR-like (we refer to this as *ODC*), reduced broadcast (location-based scheme as described in[23]), dynamic minimum dominating set (*MDS*) (dominant pruning as in [24][25]), *ZRP* and *TRANSFER*. For illustration we briefly explain flooding overhead.

For flooding, in a network of N nodes, the request is transmitted by $N-1$ nodes.

We get $E_{flood}=(N-1).Etx(1+f.g)\approx Etx(N+2Lf)$, for large N , where L is the number of links in the network, g is the node degree, and $g=2L/N$.

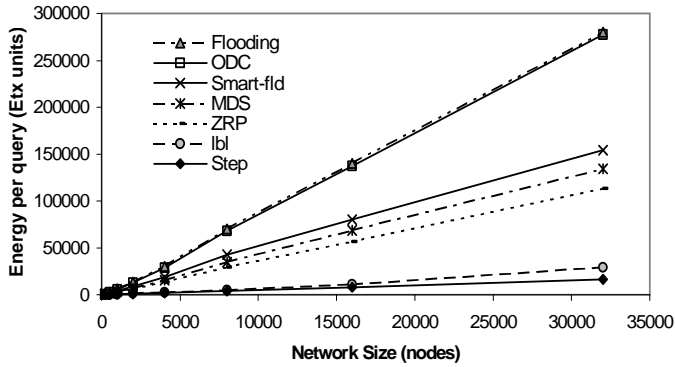


Figure 18. Comparing resource discovery schemes using energy consumption

Figure 18 shows the results for query overhead for $V_{max}=20m/s$ (similar trends were observed for other velocities). Noticeable improvement is observed in performance by using contacts, especially in large-scale networks. Figure 19 shows overhead of the various schemes relative to flooding. Performance of ODC degrades with increase in network size, and it approaches flooding in large-scale networks. Reduced broadcast ranges from 55-70% while MDS ranges from 48-55% of flooding overhead. ZRP's performance varies from 33.5% of flooding in small networks to 46% at larger scale. 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

The above analysis presented 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 several experiments are conducted to compare the energy distribution in the nodes for flooding, ZRP and Step. The *power-aware* contact selection algorithm described above is integrated into the Step scheme; the new scheme is called *E-Step*. For our simulations we take $P_{th1}=90\%$, $P_{th2}=80\%$, and so on. The previous 1000 nodes simulation setup is used for 1000 randomized small transfers. In addition, we start the nodes with E_{max} energy level that 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 contact-based approach to flooding and ZRP, for which E_{max} is set to 12,000 Etx units. And the second set to compare Step and E-Step, for which E_{max} is set to 3,000 Etx units. The results are

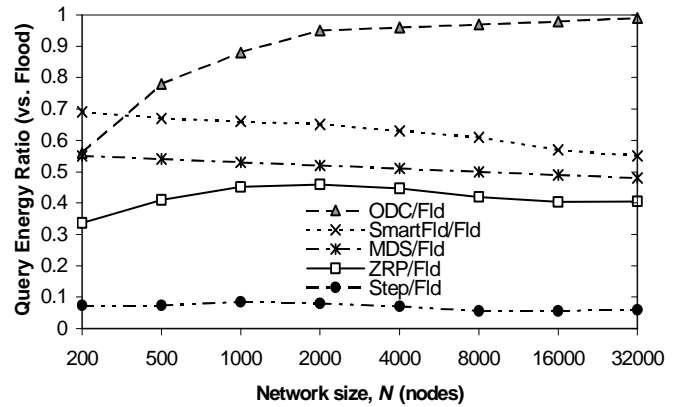


Figure 19. Query overhead of the various schemes normalized by flooding

networks. This is due to: (i) decrease in number of transmitted packets per query, and (ii) avoiding broadcast and using hop-by-hop unicast messages.

shown in Figure 20 and Figure 21, respectively. In those 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 Figure 20 we observe that the contact-based approach clearly outperforms 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, this 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.

The second set of simulations compares Step and *E-Step*. Even though the overall energy consumption was observed to be very similar but the load-balancing (or energy-balancing) ability 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 20 lowest energy-ranked nodes, and by more than 15% for the 50 lowest energy-ranked nodes.

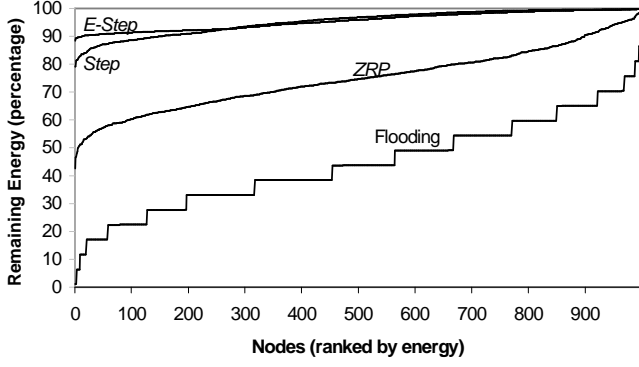


Figure 20. Remaining energy for Flooding, ZRP, Step and E-Step ($E_{max}=12,000$)

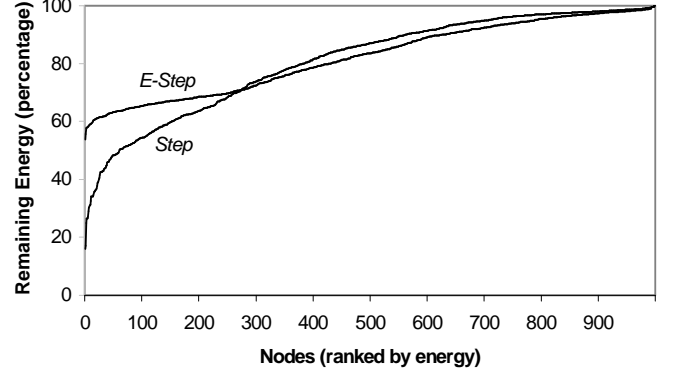


Figure 21. Remaining energy for Step and E-Step ($E_{max}=3,000$)

5.3.3. Zone Overhead

The zone overhead includes the energy consumed by the link state message exchange. For link state, the zone exchange is in the form of broadcast messages by each node, up to R hops away. This exchange increases linearly with mobility (with more link changes). So, this overhead is normalized with respect to mobility using $Z(R)$. The zone overhead is also a function of the number of nodes in the zone. Figure 22 shows $Z(R)$ for TRANSFER and for ZRP. (ZRP uses zone of $2R-1$ for early termination).

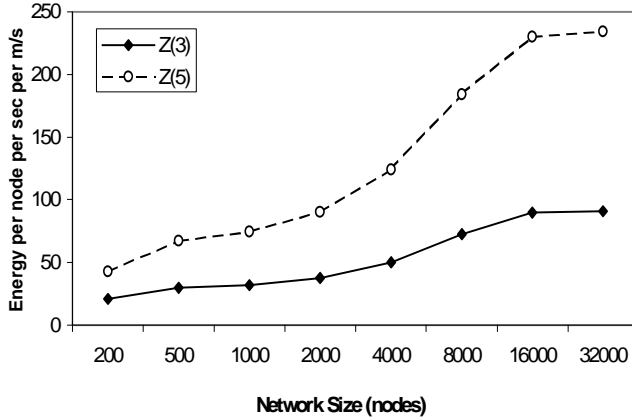


Figure 22. Normalized Intra-zone overhead for the basic proximity $R=3$, $Z(3)$ and the extended zone of $2R-1$, $Z(5)$

5.3.4. Comparisons of Total Overhead

The query overhead is a function of the query rate (query/sec), while zone overhead is a function of mobility (m/s). In order to combine these two

components we represent the query rate as a function of mobility, and normalize the energy per node per second per m/s of mobility. We use the QMR metric (query-mobility-ratio, or q) defined per node as query/s/(m/s) or simply query/km (this is similar to CMR in [12]).

$Z(R-1)$ has units of 'energy (E_{tx}) per sec per node per (m/s)'. Let the energy consumption per query for step be E_{step} in (E_{tx}) units. Similarly for flooding and ZRP, we have E_{flood} and E_{ZRP} . The overall query overhead for step, $E_{Qstep} = q \cdot E_{step}$. The units of E_{Qstep} are in ' E_{tx} units per sec per node per m/s', compatible with $Z(R)$. The total overhead for step becomes:

$$E_{Tstep} = Z(R) + E_{Qstep} = Z(R) + q \cdot E_{step}$$

For flooding, no zone overhead is incurred, so $E_{Tflood} = E_{Qflood} = q \cdot E_{flood}$. For ZRP the zone (or intra-zone) overhead is incurred for $2R-1$ hops, hence $E_{TZRP} = Z(2R-1) + E_{QZRP} = Z(2R-1) + q \cdot E_{ZRP}$. For brevity, we evaluate the total energy-consumption ratio, TER , of step to flooding and ZRP. We get:

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

$$\text{and } TER_{ZRP} = \frac{E_{Tstep}}{E_{TZRP}} = \frac{Z(R) + q \cdot E_{step}}{Z(2R-1) + q \cdot E_{ZRP}}.$$

Figure 23 shows TER_{flood} 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.

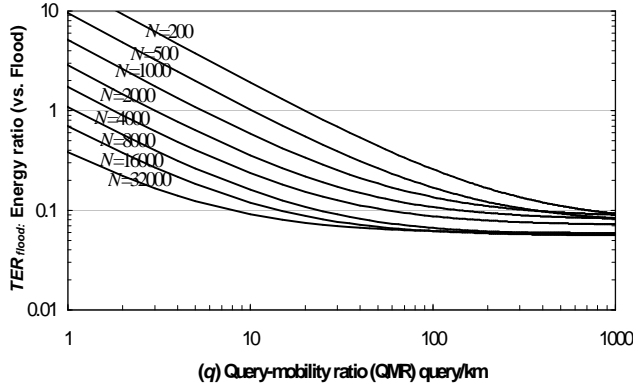


Figure 23. Total energy ratio vs. flood (TER_{flood})

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 overhead⁷. 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 24 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 $(2R-1)$. 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 $TER_{ZRP} \sim 48\%$, 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.

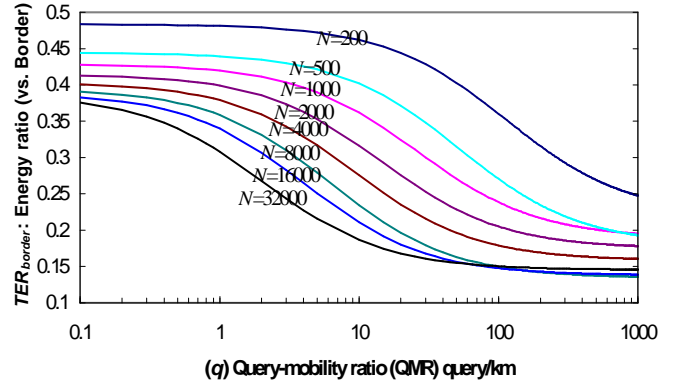


Figure 24. Total energy ratio vs. ZRP (TER_{ZRP})

6. Conclusions

In this chapter, we provided an overview of the main approaches for resource discovery in wireless networks. The problem of resource discovery is becoming more challenging and pressing as more emerging infrastructure-less networks are designed and implemented. The various approaches described in this chapter included flooding-based approaches, including expanding ring search and reduced broadcast heuristics, on-demand ad hoc routing, hierarchical cluster-based and dominating set schemes, and hybrid loose hierarchy schemes including zone routing and contact based architectures.

Reduced broadcast approaches and dominating set approaches attempt to reduce the redundancies of flooding but attempt to all discovery requests to all nodes in the network. On-demand ad hoc routing utilizes information caching to reduce effects of flooding, but is greatly affected by network dynamics that cause cache invalidation. Hybrid approaches, such as zone and contact-based routing use the proactive approach within the zones, and the reactive approach for out-of-zone discovery. Zone routing uses bordercasting, while contact-based schemes select out-of-zone contact nodes to avoid bordercasting.

After conducting an extensive comparative study between the above schemes, it was observed that hybrid approaches are useful when the cost of zone maintenance is amortized over a reasonable number of requests, otherwise flooding-based approaches work better. In general, if power control is used such that only active nodes participate in zone construction and maintenance, then hybrid approaches seem to scale best. In particular the contact based approach is promising as

⁷ 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.

it performs well for a fixed set of parameters over a relatively wide array of scenarios and network sizes.

7. Acknowledgements

The author acknowledges the useful feedback provided during discussions with the wireless networking laboratory research group at USC. Specifically the author thanks Shao-Cheng Wang for feedback on overhead analysis and cache modeling for DSR, and Karim Seada for comments on contact selection protocols. Ahmed Helmy was supported by the NSF CAREER Award 0134650, and research grants from Intel and Pratt&Whitney Institute for Collaborative Engineering (PWICE).

References

- [1] Watts,Strogatz,Collective dynamics of small-world networks,*Nature*98
- [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, CCR, Oct.94
- [4] C. E. Perkins, E. M. Royer, Ad-hoc On-Demand Distance Vector Routing, Proc. 2nd IEEE Wksp. Mobile Comp. Sys. And Apps., Feb. 99.
- [5] D. B. Johnson, D. A. Maltz, Dynamic Source Routing in Ad-Hoc Wireless Networks, *Mobile Computing*, 1996, pp.153-181.
- [6] P. Guanyu, M. Gerla, X. Hong, "LANMAR: landmark routing for large scale wireless ad hoc networks with group mobility",*Mobihoc'00*
- [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". *IFIP-TC6 Networking, Pisa, May 2002*.
- [9] M. Pearlman, Z. Haas, "Determining the optimal configuration for the zone routing protocol", IEEE JSAC, p. 1395-1414, Aug 1999.
- [10]Z. Haas, M. Pearlman, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks", IETF Internet draft for the Manet group, June '99.
- [11]Z. Haas, M. Pearlman, "The Performance of Query Control Schemes for the Zone Routing Protocol", ACM SIGCOMM '98.
- [12]Z. Haas, M. Pearlman, "ZRP: A Hybrid Framework for Routing in Ad Hoc Networks", in Ad Hoc Networks, Addison Wesley, 2001.
- [13]Z.J. Haas, M.R. Pearlman, "The Performnce of Query Control Schemes for the Zone Routing Protocol," ACM/IEEE ToN, Aug 01
- [14]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.
- [15]S. Kumar, C. Alaettinoglu, D. Estrin, "SCOUT: Scalable object tracking through unattended techniques", ICNP 2000.
- [16]J.J. Aceves, M. Spohn, "Bandwidth-Efficient Link-State Routing in Wireless Networks", in Ad Hoc Networks, Addison Wesley, 2001.
- [17]A. Helmy, "Small Worlds in Wireless Networks", IEEE Communications Letters, pp. 490-492, Vol. 7, No. 10, Oct 2003.
- [18]Tsu-Wei Chen and Mario Gerla, "Global State Routing: A New Routing Scheme for Ad-hoc Wireless Networks" Proc. IEEE ICC'98.
- [19]S. Murthy and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks", ACM Mobile Networks and App. J., Issue on Routing in Mobile Communication Networks, Oct. 1996.
- [20]C.-C. Chiang, "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel" Proc. IEEE SICON'97, Apr.1997.
- [21]J. Liu, Q. Zhang, et al, "A Novel Framework for QoS-Aware Resource Discovery in Mobile Ad Hoc Networks", *IEEE ICC '02*.
- [22]T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum et L. Viennot, Optimized Link State Routing Protocol, IEEE INMIC 2001.
- [23]S. Ni, Y. Tseng, Y. Chen and J. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network", ACM Mobicom, '99.
- [24]H. Lim, C. Kim, "Flooding in wireless ad hoc networks", *Computer Communications Journal*, 24(3-4),353-363, 2001.
- [25]A. Laouiti, A. Qayyum et L. Viennot, "Multipoint Relaying: An efficient technique for flooding in mobile wireless networks" HICSS02
- [26]L. Breslau, D. Estrin, K. Fall, et. al, "Advances in Network Simulation", *IEEE Computer*, May 2000.
- [27]H. Gupta, S. Das, Q. Gu, "Connected Sensor Cover: Self-Organization of Sensor Networks for Efficient Query Execution", *ACM,MobiHoc* 03.
- [28]C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *ACM MobiCom*, August 2000.
- [29]M. Chu, H. Haussecker, F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks." *Int'l J. High Performance Computing Applications*, 2002.
- [30]R. Govindan, J. Hellerstein, W. Hong, S. Madden, M. Franklin, S. Shenker, The Sensor Network as a Database, Technical Report 02-771, Computer Science Department, UCLA, September 2002.
- [31]P. Bonnet, J. E. Gehrke, and P. Seshadri, "Querying the Physical World," *IEEE Personal Communications*, Vol. 7, No. October 2000.

- [32] P. Bonnet, J. Gehrke, P. Seshadri, "Towards Sensor Database Systems," *Mobile Data Management*, 2001.
- [33] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, S. Shenker, "GHT – A Geo-graphic Hash-Table for Data-Centric Storage," *First ACM WSNA Workshop*, 2002.
- [34] W.R. Heinzelman, J. Kulik, and H. Balakrishnan "Adaptive protocols for information dissemination in wireless sensor networks," *ACM MobiCom*, pp. 174-185, Aug. 1999.
- [35] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan "Energy-efficient communication protocol for wireless microsensor networks," 33rd International Conference on System Sciences (*HICSS*), 2000.
- [36] David Braginsky and Deborah Estrin, "Rumor Routing Algorithm For Sensor Networks," *First WSNA Workshop*, September 2002.
- [37] N. Sadagopan, B. Krishnamachari, A. Helmy, "The *ACQUIRE* Mechanism for Efficient Querying in Sensor Networks", *First IEEE ICC SNPA Workshop* May 2003.
- [38] B. Karp, H. Tung, "Greedy Perimeter Stateless Routing for Wireless Networks", *ACM MobiCom* 2000.
- [39] K. Seada, A. Helmy, "Rendezvous Regions: A Scalable Architecture for Service Provisioning in Large-Scale Mobile Ad Hoc Networks", *ACM SIGCOMM*, Refereed poster, 2003.
- [40] D. B. Johnson, D. Maltz, J. Broch, "DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks", Book Chapter in *Ad Hoc Networks*, Editor C. Perkins, Addison Wesley, pp. 139-172, 2001.
- [41] Y. Hu, D. Johnson, "Caching Strategies in On-Demand routing protocols for Ad Hoc wireless networks", *ACM MobiCom*, 2000.
- [42] Yih-Chun Hu and David B. Johnson. Ensuring Cache Freshness in On-Demand Ad Hoc Network Routing Protocols. *Proc. POMC Workshop on Principles of Mobile Computing*, pp. 25-30, October 2002.
- [43] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols, *ACM MobiCom*, 1998.
- [44] David A. Maltz. On-Demand Routing in Multi-hop Wireless Ad Hoc Networks. Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, May 2001.
- [45] M. Steenstrup, "Cluster-Based Networks", Book Chapter in *Ad Hoc Networks*, Editor C. Perkins, Addison Wesley, pp. 75-138, 2001.
- [46] Y. Chen, A. Liestman, "Approximating Minimum Size Weakly-Connected Dominating Sets for Clustering Mobile Ad Hoc Networks", *ACM MobiHoc* June 2002.
- [47] U. Kozat, L. Tassiulas, "Service discovery in mobile ad hoc networks: an overall perspective on architectural choices and network layer support issues", *AdHoc Networks Journal – Elsevier*, In press, 2003.
- [48] A. Helmy, S. Garg, P. Pamu, N. Nahata, "Contact Based Architecture for Resource Discovery (*CARD*) in Large Scale MANets", *IEEE/ACM IPDPS Int'l Workshop on Wireless, Mobile and Ad Hoc Networks (WMAN)*, pp. 219-227, Apr 2003.
- [49] A. Helmy, S. Garg, P. Pamu, N. Nahata, "CARD: A Contact-based Architecture for Resource Discovery in Ad Hoc Networks", *ACM Baltzer Mobile Networks and Applications (MONET) Journal, Kluwer publications, Special issue on Algorithmic Solutions for Wireless, Mobile, Ad Hoc and Sensor Networks*. To appear (1st Quarter 2004).
- [50] A. Helmy, "Mobility-Assisted Resolution of Queries in Large-Scale Mobile Sensor Networks (*MARQ*)", *Computer Networks Journal - Elsevier Science (Special Issue on Wireless Sensor Networks)*, Vol. 43, Issue 4, pp. 437-458, November 2003.
- [51] A. Helmy, "TRANSFER: Transactions Routing for Ad-hoc Networks with eEfficient Energy", *IEEE GLOBECOM*, December 2003.
- [52] J. Liu and Q. Zhang and J. Zhang and W. Zhu and B. Li, "A Novel Framework for QoS-Aware Resource Discovery in Mobile Ad Hoc Networks", *IEEE ICC*, 2002.
- [53] K. Seada and A. Helmy, "Rendezvous Regions: A Scalable Architecture for Service Location and Data-Centric Storage in Large-Scale Wireless Networks", *IEEE/ACM IPDPS 4th Int'l Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN)*, Santa Fe, April 2004.
- [54] D. Niculescu and B. Nath, "Trajectory-based Forwarding and its Applications", *ACM MobiCom*, September 2003.
- [55] A. Helmy, "Architectural Framework for Large-Scale Multicast in Mobile Ad Hoc Networks", *IEEE International Conference on Communications (ICC)*, Vol. 4, pp. 2036-2042, April 2002.
- [56] J. Tchakarov and N. Vaidya, "Efficient Content Location in Wireless Ad Hoc Networks", *IEEE International Conference on Mobile Data Management (MDM)*, January 2004.
- [57] I. Aydin and C.-C. Shen, "Facilitating match-making service in ad hoc and sensor networks using pseudo quorum", *IEEE ICCCN* 2002.
- [58] D. Doval and D. O'Mahony, "Nom: Resource location and discovery for ad hoc networks", *Med-hoc-Net* 2002.
- [59] V. Verma and S. Helal and N. Desai and C. Lee, "Knoark: A service discovery and delivery protocol for ad hoc networks", *IEEE WCNC*, 2003.
- [60] O. Ratsimor and D. Chakraborty, "Allia: Alliance-based service discovery protocol for MANets", *ACM Mobile Commerce Workshop*, 2002.
- [61] D. Chakraborty and A. Joshi and T. Finin and Y. Yesha, "GSD: a novel group-based service discovery protocol for MANets", *MWCN* 2002.
- [62] B. Williams and T. Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks", *ACM MobiHoc* 2002.
- [63] F. Bai and N. Sadagopan and A. Helmy, "The *IMPORTANT* Framework for Analyzing the Impact of

Mobility on Performance of Routing for Ad Hoc Networks”, AdHoc Networks Journal - Elsevier Science, Vol. 1, Issue 4, pp. 383 - 403, November 2003.

- [64] N. Sadagopan and F. Bai and B. Krishnamachari and A. Helmy, “PATHS: analysis of PATH duration Statistics and their impact on re-active MANET routing protocols”, *ACM MobiHoc (The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing)*, pp. 245-256, Annapolis, MD, June 2003.
- [65] F. Bai and A. Helmy, “A Survey of Mobility Modeling and Analysis in Wireless Adhoc Networks”, Book Chapter in the upcoming book on "Wireless Ad Hoc and Sensor Networks" to be published by Kluwer Academic Publishers. To appear June 2004.
- [66] F. Bai and A. Helmy, The IMPORTANT Framework for Analyzing and Modeling the Impact of Mobility in Wireless Adhoc Networks, Book Chapter in the upcoming book on "Wireless Ad Hoc and Sensor Networks" to be published by Kluwer Academic Publishers. To appear June 2004.
- [67] D. Son and A. Helmy and B. Krishnamachari, “The Effect of Mobility-induced Location Errors on Geographic Routing in Ad Hoc Networks: Analysis and Improvement using Mobility Prediction”, *IEEE Wireless Communications and Networking Conference (WCNC)*, March 2004.
- [68] Y. Lu and H. Lin and Y. Gu and A. Helmy, “Towards Mobility-Rich Performance Analysis of Routing Protocols in Ad Hoc Networks: Using Contraction, Expansion and Hybrid Models”, *IEEE International Conference on Communications (ICC)*, June 2004.
- [69] C. Shete and S. Sawhney and S. Herwadka and V. Mehandru and A. Helmy, “Analysis of the Effects of Mobility and Node Density on the Grid Location Service in Ad Hoc Networks”, *IEEE International Conference on Communications (ICC)*, June 2004.
- [70] J. Yoon and M. Liu and B. Noble, “Sound Mobility Models”, *ACM MobiCom*, September 2003.
- [71] J. Yoon and M. Liu and B. Noble, “Random Waypoint Considered Harmful”, *IEEE INFOCOM*, vol 2, pp 1312-1321, April 2003.
- [72] A. Jardosh and E. Belding-Royer and K. Almeroth and S. Suri, “Towards Realistic Mobility Models For Mobile Ad hoc Networks”, *ACM Mobicom*, San Diego, California, September 2003.