

Efficient Resource Discovery for Large Scale Ad hoc Networks using Contacts

Nitin Nahata, Priyatham Pamu, Saurabh Garg, Ahmed Helmy*

Department of Electrical Engineering, University of Southern California
 {nnahata, pamu, sgarg, helmy}@usc.edu - <http://nile.usc.edu/ee499/contacts>

The infrastructure-less nature of ad hoc networks renders resource discovery a challenging problem. Unlike wired networks, mobility induces frequent route changes. Traditional protocols proposed for resource discovery in such environments either involve global flooding or are based on complex hierarchy formation. While flooding does not scale well, hierarchy formation involves complex coordination between nodes and may suffer serious performance degradation due to mobility.

To overcome these limitations we propose a new architecture based on the concept of *small worlds*. In our architecture we adopt a hybrid approach in which a node uses (a) proactive routing (e.g., DSDV) to reach its **neighborhood** within a limited number of hops, R , (typically 3-5 hops), and (b) reactive querying beyond the neighborhood via **contacts**. *Contacts* act as *short cuts* that attempt to transform the network into a small world by reducing the degrees of separation. They help in providing a view of the network beyond the neighborhood during resource discovery. Each node maintains state for a few contacts (typically 5-15) beyond its neighborhood. Contacts are polled periodically to validate their presence and routes. For discovering resources efficiently, queries are sent to the contacts that leverage the knowledge of their neighborhood. Our approach has the following design goals: scalability to thousands of nodes, efficiency in terms of network overhead, robustness and decentralized operation. We do not assume availability of any geographic information.

Contact selection (CS) is initiated by a source node, S , by sending contact-selection queries (CSQ) through nodes at the edge of its neighborhood (edge nodes). CSQ contains source ID, hop count and list of S 's previous contacts. Our first CS protocol, the *P method* (PM), states that a node receiving the query, checks whether any of the S 's previous contacts lie inside its neighborhood. This reduces overlap between contact neighborhoods and hence improves reachability. If no contact lies in its neighborhood, the node chooses to be a contact with probability P , otherwise it forwards the query to a neighbor. $P = (H - 2R) / (r_{max} - 2R) * 100$ where H is hop count from S and r_{max} is the maximum distance (in hops) at which a contact can be selected. The above equation helps to locate contacts that lie between $2R$ and r_{max} hops from the source and reduces overlap between neighborhoods of S and the contact. If r_{max} is reached and a contact is not chosen (e.g., due to overlap or use of P), a *back-tracking* mechanism is used to select another contact. Our study shows that overhead of such

mechanism may be significant. To ameliorate such a problem we propose our second CS protocol, the edge method, EM . In EM the list of edge nodes (E -List) for S is also included in the CSQ , but P is not used. This reduces back-tracking traffic drastically. A node receiving CSQ checks if any node on the E -list lies inside its neighborhood. This ensures non-overlap with S 's neighborhood. Also, in EM , query and node IDs are kept to prevent loops. EM resulted in notable increase in reachability with drastic decrease in overhead over PM , as shown in Fig. 1.

Contact maintenance is based on S periodically polling its contacts to validate and update their routes. Broken links are recovered using neighborhood knowledge (**local recovery**). Local recovery reduces maintenance overhead and better protocol robustness.

When a source node, S , initiates a resource discovery for a target, the target is looked up in S 's neighborhood, then a query is sent to S 's contacts, which in turn look in their neighborhood. The contacts may further query their contacts, so on, up to the **query depth**. This creates S -rooted tree of contacts, which is helpful in making our approach scalable.

We carried out simulations on various networks (100-1000 nodes) under mobility, and analyzed performance of our architecture in terms of reachability and contact selection and maintenance overhead. We found that as the number of contacts increased, both reachability *and* overhead increased. This shows a clear tradeoff between reachability and overhead (shown in Fig. 1). Also, increasing R increased reachability but led to more proactive overhead inside the neighborhood. Increasing query depth improved reachability and seems scalable. We also noticed that maintenance overhead decreased over time as the nodes seemed to find more relatively stable contacts. We plan to study effects of various mobility models on maintenance overhead.

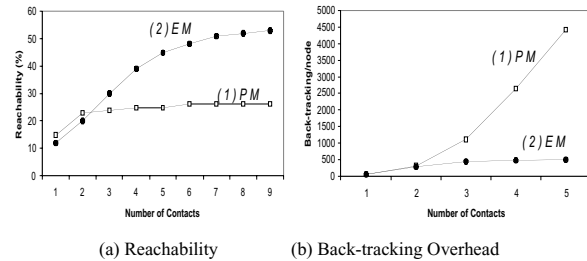


Fig. 1. Reachability vs. Overhead trade-off for (1) P method, PM , protocol, and (2) Edge method, EM , protocol. (shown for 500 nodes, 710mx710m, transmission range=50m, $R=3$, $r_{max}=20$, query depth=1).

* A. Helmy was supported by NSF CAREER Award 0134650.