Architectural Framework for Large-Scale Multicast in Mobile Ad Hoc Networks

Ahmed Helmy
Department of Electrical Engineering
University of Southern California

***Abstract* - Emerging ad hoc networks are infrastructure-less networks consisting of wireless devices with various *power* constraints, capabilities and *mobility* characteristics. An essential capability in future ad hoc networks is the ability to provide scalable *multicast services*. This paper presents a *novel adaptive architecture* to support multicast services in large-scale wide-area ad hoc networks. Existing works on multicast in ad hoc networks address only small size networks. Our main design goals are *scalability, robustness* and *efficiency*. We propose a *self-configuring* hierarchy extending zone-based routing with the notion of *contacts* based on the *small world graphs* phenomenon and metrics of stability and mobility. We introduce a new *geographic-based multicast address allocation* scheme coupled with *adaptive anycast* based on *group popularity*. Our scheme promises efficient and robust operation in the common case. Also, based on the new concept of *rendezvous regions*, we provide a *bootstrap* mechanism for the multicast service; a challenge generally ignored in previous work.**

I. INTRODUCTION

Providing a scalable architecture for multicast services in large-scale networks has proven to be a challenge for years in the Internet[30][3]. The research challenges are even greater for ad hoc networks, mainly due to lack of infrastructure and the *highly dynamic* nature of wireless nodes and their unexpected *mobility*. The developed multicast service should allow group participants to join and leave at will, and should impose no restrictions on node mobility. It should also provide automatic multicast address allocation and session advertisement.

In this paper, we propose a new architecture for multicast services support in large-scale ad hoc networks. Since ad hoc networks are *infrastructure-less*, the developed protocols must be *self-configuring*. Scalability and robustness should also be addressed. Existing approaches for multicast usually employ flooding for discovery of group participants or cores and apply only to small networks. Questions are often asked about which routing mechanisms to use, pro-active or re-active routing? Hierarchical or flat? We leverage strengths of the different approaches by introducing a *hybrid* approach that *adapts* to network dynamics (e.g., *mobility* and *power*).

Our proposed architecture utilizes *highly adaptive* mechanisms. Specifically, we design adaptive zone-based hierarchy augmented by the notion of *contact* nodes to increase network coverage. For multicast service support, we provide novel schemes, based on *adaptive anycast* for resource discovery. We also introduce geographic multicast address allocation to map groups into *rendezvous regions.*

Our work targets *large-scale wide-area* ad hoc networks (with tens of thousands of nodes). Our design requirements include *scalability, service model, robustness* and *efficiency*.

A.Design Requirements

The main factors driving our design are scalability, the multicast service support, and robustness.

(a) **Scalability**: Unlike most related work that considers tens to hundreds of mobile nodes, our architecture should be able to support tens of thousands of nodes. Flat architectures are known not to scale well[7], mainly due to the far-reaching effects of network dynamics; mobility, failures and topological changes. Such effects consume network resources (bandwidth/power) and lead to recovery delays and increased route oscillations. Hierarchical architectures, on the other hand, alleviate the above problems, as they tend to localize and dampen network dynamics, and scale routing tables using aggregation. Several such architectures are based on *clustering* mechanisms, in which a single node per cluster (the master) is chosen to manage the cluster. In such architectures failure or movement of the master may have severe negative effects on the hierarchy. Furthermore, establishment and maintenance of clusters incurs a lot of overhead and complexity. We design an architecture that leverages hierarchical advantages while alleviating effects of master and hierarchy maintenance. We propose a two-level distributed hierarchical architecture. For the first level of the hierarchy we adopt a *zone-based* approach (a variant of the zone routing protocol ZRP[15]), in which each node has its own view of a zone. For the second level, we introduce a novel concept of *contacts* (based on the concept of small-world[16][34]) to enhance nodes' view and aid in route and resource discovery.

Also affecting scalability, is the choice of routing protocol. In general, ad hoc routing protocols are either pro-active or re-active. Pro-active protocols[6] exchange periodic messages to keep routes up-to-date, with low delay route discovery and significant overhead of periodic route exchange. Re-active protocols[10], by contrast, maintain routes on-demand and incur less overhead with more route discovery delay, which usually involves request broadcasts. We attempt to combine strengths of both protocols using a hybrid approach. Inside a zone pro-active routing is used to discover routes to nearby nodes, while re-active routing is used for external routes.

(b) **Multicast Service Support**: The multicast service model defines conditions for joining/leaving groups. Multicast participants should be able to join or leave groups at will. We adopt a model in which participants are not known a priori and are allowed to move freely during a multicast session.

In such a model the main problems include *rendezvous* of participants, *service bootstrap* and *multicast address allocation*. We design a novel *adaptive anycast* architecture for resource discovery to facilitate the *rendezvous* of group participants. Instead of the traditional rendezvous approaches of broadcast and prune[23][5][28] or rendezvous cores[2][4], we introduce a new multicast paradigm based on '*sender push, server cache, receiver pull'* approach, that better fits large-scale ad hoc networks. We also design mechanisms for providing participants with active session information as part of our *bootstrap* architecture, along with a new multicast address allocation scheme based on geographic address allocation. We require minimum configuration of nodes. In our scheme, nodes only need to know a *well-known* session announcement group address and an algorithmic mapping function to map groups into *rendezvous regions (RRs)*.

(c) **Robustness**: In highly dynamic ad hoc networks, robustness is of prime concern. We incorporate mobility and stability models into our hierarchy formation to achieve adaptivity. In addition, our distributed adaptive resource discovery scheme avoids single point of failure scenarios and promises continued operation and graceful recovery during network partitions. We also incorporate path redundancy mechanisms in our multicast routing protocol. Multicast *trees* do not provide sufficient robustness against mobility and failures. We use *mesh* structures for robust delivery. Unlike existing proposals for mesh construction, however, our mechanisms are designed to build meshes in anticipation of movement. This achieves better performance and provides path redundancy that may be used in case of failures. Mesh branches are activated on-demand to reduce overhead.

For energy-efficiency we limit communication by using localized mechanisms for sender advertisements, query, resource-discovery and anycast.

B. Brief Architectural Overview

We provide an architectural framework based on the following components: (i) Hierarchy formation and adaptation. (ii) Multicast service architecture consisting of: (a) the multicast model, (b) multicast routing, (c) adaptive resource discovery and (d) multicast address allocation.

II. HIERARCHY FORMATION AND ADAPTATION

We provide mechanisms for self-configuring hierarchy formation, based on zone-based routing, augmented by *contacts*. Hierarchy adaptation mechanisms is based on link availability and mobility estimation models.

A. Hierarchy Formation

As was discussed earlier, flat routing architectures do not scale well for wide-area networks, especially for highly dynamic ad hoc networks. Also, hierarchical approaches based on concept of *master*s, through which traffic from the clusters funnels, are undesirable.

One possible approach to consider is to use the master for infrequent coordination but not for forwarding packets. One such approach is the Landmark hierarchy (LMH)[19]. The traffic from/to a cluster need not go through the landmark, which adds robustness. LMH employs complex promotion, demotion, and adoption operations for hierarchy maintenance. Furthermore, effects of mobility on the hierarchy may be drastic, sometimes leading to total re-configuration of the hierarchy. In addition, sub-optimal paths may be common due to hierarchical routing.

Another approach that avoids complex coordination for architectural setup is the zone routing protocol (ZRP)[15] [33], shown in Fig. 1 (a). which defines a zone for every node as the number of nodes reachable within a radius of $\boldsymbol{R}$ hops away. Inside the zone proactive (intra-zone) routing is used, so nodes obtain routes to all nodes within their zone. To discover nodes outside of the zone, reactive (inter-zone) routing is performed by flooding through periphery or *border* nodes of each zone (known as *bordercasting*). ZRP routing overhead depends heavily on the choice of the *zone radius*. If the radius is too small the routing overhead is dominated by reactive overhead, and vice versa.

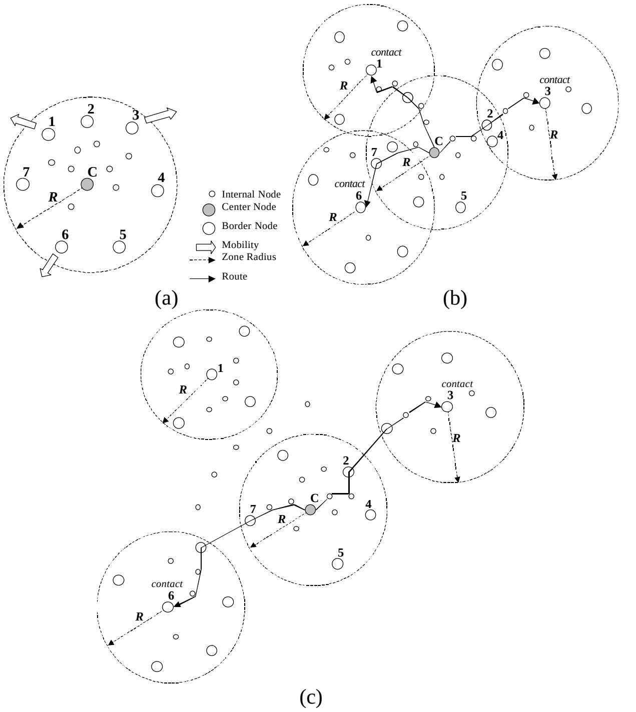
ZRP seems appealing, but experiences excessive delays and overheads in large-scale networks, where much of the traffic maybe destined out-of-zone. Therefore, we develop a novel approach that goes beyond the *zone* while maintaining similar simplicity and stability. Our approach is based on a concept we call *contacts.* Contacts, of a certain node *x*, are nodes that previously existed in *x's* zone but are drifting out-of-zone, and hence have a network view beyond that of *x* or any of its border nodes[1]. While drifting away gradually, *x* may maintain route to (some of) these drifting nodes using low overhead mechanisms (since these drifting nodes were in *x*'s zone and are close to its border nodes). Fig. 1 shows a simple illustrative example of zoning, contacts and effects of mobility.

The concept of contacts relates to the concept of *small world* graphs, where Watts[16] observes that small world graphs have low average path length with high clustering. In other words, introducing a small number of distant links as short cuts leads to significant reduction in path length. This may be achieved by picking random distant nodes. However, picking *contacts* randomly leads to unpredictable overhead for *contact* route discovery and maintenance. In addition, knowing mobility characteristics and stability of a node helps identify better (more useful) contacts. We *take advantage of node mobility* and pick contacts from those nodes drifting away from the zone. The characteristics of the resulting graphs are function of time and depend on the initial choice of contacts and the node mobility. We plan to study such characteristics in our future work.

Contacts play an important role in resource discovery in our architecture, as will be explained later. A node should choose its contacts carefully to attempt to maintain a useful contact as long as the contact route is kept. The *contact list*, maintained by a node, changes *adaptively* as the network conditions change. We further discuss how the contact list is chosen in the next section.

---

[1] It seems this is one of the few concepts that actually takes advantage of mobility. Mobility is often viewed as a disadvantage, and for good reasons, but we think it should also be utilized, when possible.



(a) (b)

(c)

Fig. 1. Example of zoning, contacts and effect of mobility: (a) Zone for node *C* is shown (with radius *R*). Border nodes are numbered (1-7). Nodes 1,3 and 6 are drifting out of zone. (b) Radii for the drifting nodes are shown. *C* stays in *contact* with the drifting nodes, which enables it to obtain better network coverage with low overhead. (c) After moving away, contact nodes drift up to a point where their zones no longer intersect with *C*'s zone. In this example, *C* maintains contact with those nodes not more than (2*R*+1) hops away, i.e. nodes 3, 6, and loses contact with node 1 as it drifts farther away.

B. Hierarchy Adaptation

The architecture presented thus far provides a framework for hierarchy formation. Such hierarchy should be highly adaptive to network dynamics and conditions (e.g., *mobility* and *energy*). To achieve this, we use mechanisms that integrate measures of stability and power. In hierarchy formation, the zone radius and contact selection may be adapted dynamically to establish certain stability measures. Each node should adapt its behavior to achieve desirable collective performance. For example, number of contacts chosen by a node should be a function of the contacts already established by other nodes in its zone. Increase in number of contacts increases overhead and is done only as necessary.

We devise a mechanism for contact selection in which a node chooses its contacts with probability *p*, where *p* is proportional to the energy estimates E_{est} of the node and the contact, their relative stability S_{est}, and the activity level of the node A_{est} measured as rate of discovery requests. Also, *p* is inversely proportional to the number of zone contacts Z_{est}. Hence, $pa\frac{E_{est}S_{est}A_{est}}{Z_{est}}$. These quantities are locally measured and may be piggybacked on intra-zone pro-active messages. In addition to energy left E_{left}, E_{est} should include drainage rate *DE*; i.e., $E_{est} = \left(\frac{E_{left}}{\Delta E}\right)_{node}\left(\frac{E_{left}}{\Delta E}\right)_{contact}$.

Stability estimate S_{est} may be derived from adaptive availability and mobility models. In[7] the *(a,t)* scheme was proposed to measure link and path *availability*, where *a* is the probability that a link will be available for time *t*. The basic idea is to build routes whose availability is probabilistically bounded. The proposed model is based on random walk mobility and determines the conditional probability that the nodes will be within range of each other at time t_0+t given that they are located within range at time t_0. Another mobility metric may be derived from signal strengths. By measuring the received signal power (*RxPwr*) between two consecutive packets received from the same transmitter we define *relative mobility* metric $rm = \frac{RxPwr_{new}}{RxPwr_{old}}$, if $rm<<1$ then nodes are moving away quickly, and vice versa.

The above adaptive mechanisms are also used in the multicast service model and resource discovery in section III. In that case, node capability (e.g., GPS) should also be considered when choosing contacts.

III. MULTICAST SERVICE ARCHITECTURE

Providing a scalable multicast service architecture is the focal point of our research. The hierarchical architecture proposed thus far provides the basis for efficient support of our multicast service model, presented in this section. As the mechanisms for our multicast model unfold, the essential role played by the adaptive hierarchical architecture will become very clear in support of multicast routing and resource discovery. Earlier work on ad hoc multicast focused on mechanisms to establish distribution trees (or meshes) in small to medium-scale networks, mainly using periodic broadcasts or relying on *cores*. In contrast, we address multicast in large-scale ad hoc networks. Our multicast service architecture consists of three main components:
(a) the multicast *rendezvous* model, (b) multicast routing, and (c) multicast service *bootstrap* using *adaptive resource discovery* and *geographic multicast address allocation*.

A. The Multicast Model

The basic premise for scalable multicast is that sources do not know who/where receivers are a priori. This model enables any node to join or leave a multicast group at any point in time. Hence, one of the main components of multicast is a mechanism for group participants to *meet* or *rendezvous*. This problem has been addressed, in wired and ad hoc networks, using *broadcast-and-prune* or *rendezvous cores*. In the former[5][28][23], a participant (sender or receiver) announces its presence by broadcasting data packets (or control messages) throughout the network. Network nodes not interested in the group send prune messages to stop the flow of packets (or simply do not respond in case of control messages broadcast). It has been shown that such model does not scale for wide-area networks[37]. The rendezvous cores approach, by contrast, uses *explicit join* mechanisms to avoid periodic broadcasts. Participants join (or send packets) towards a common core, which relays the packets from the senders to the receivers using a shared tree (or mesh)[2][13]. The major research problem associated with the core approach is the *core bootstrap* and consistency problem. How do participants know the core's address/location? Senders and receivers need to maintain a consistent view of the cores in order to *meet.* A solution was presented in[4] that uses a flooding to disseminate core-to-group mapping. This scheme does not scale well for wide-area networks and its convergence performance degrades with the size of the network. For ad hoc networks these problems are exacerbated by network dynamics and node mobility. Hence, such solution is not suitable for ad hoc networks.

Alternatively, we propose a new multicast model. We refer to our model simply as *sender push, server cache, receiver pull* model. Unlike previous works on ad hoc multicast that require periodic broadcasts throughout the entire network, our scheme incurs less overhead using only localized queries and updates. We introduce the notion of *sender discovery servers (SDS)* to aid in sender location and information dissemination. As shown in Fig. 2, a sender sends an Advertisement (*Adv*) using *localized broadcast*. *SDS*s receiving the *Adv* store this information. Receivers send join requests toward the sender based on *backward learning*; every node forwarding the *Adv* adds its address to the message to construct a path back to the source. Other, farther, receivers not receiving the *Adv* message attempt to find a nearby *SDS*, first by checking in their own zone (*SDS*s advertise in their zone), then by checking with their contact list. If *SDS* is found it is queried for group information and responds with a join reply, including approximate source location or possible routes (if available at the *SDS*). Depending on the quality of the provided routes (if any), the querying receiver(s) may opt to use these routes or use zone/contact search for other routes (in this case geographic routing[22] may be used for route discovery, as described later). If *SDS* is not found, a receiver may send a localized broadcast to discover other nearby receivers of the group. If this process fails then the receiver uses a *fallback* mechanism, described later in this section. Once the information about the group/senders is available, the route discovery/construction is initiated. An illustrative example is shown in Fig. 2.

B. Multicast Routing

Establishing multicast distribution paths for ad hoc networks has been shown to be more robust using *mesh* structures, as opposed to conventional *tree* structures[14]. For single-source groups or when sources are sparsely distributed, however, even a mesh does not provide the desired path redundancy. Local recovery mechanisms[35] may be used to alleviate such a problem.

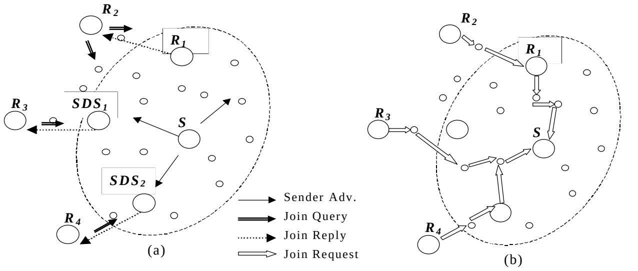


Fig. 2. Multicast service scenario. (a) Sender *S* becomes active and broadcasts an advertisement (Adv.) locally (in shaded region). Sender discovery servers SDS_1, SDS_2, and receiver R_1 receive the Adv. When new receivers join the group, they try to find: (i) a sender discovery server, (ii) nearby members of the group. Receiver R_2 finds R_1 using local broadcast, while R_3 and R_4 find SDS_1 and SDS_2. (b) Once sender information is obtained, receivers send Join Requests to build multicast distribution paths.

We propose to use mesh structures with local recovery mechanisms, similar to those used in[23][35]. For added robustness, the receiver may select multiple paths during the discovery process, during which stability metrics are included in propagated routes. Depending on the number of senders in the group, their locations and the receiver's mobility, a receiver may opt to choose several stable paths to join the group. The receiver sets an *active* flag in one join only, to activate one path at a time. Only active paths forward data packets. This reduces packet transmission overhead (a very significant factor in ad hoc networks) while maintaining robustness. If performance or stability of the active path degrades, or local recovery fails, the receiver may activate another path with high stability. Also, when the receiver moves, it may activate another path containing one of the new neighbors, thus achieving fast handoff[2]. Rules for activating/de-activating branches of the mesh should be carefully selected to avoid black holes. For example, if *any* member exists downsrtream, at least one branch must be activated. In order for a branch to be inactive, all downstream branches must be inactive.

So far, we have assumed that receivers' local search for group information is successful. In case of sparse groups, where participants are far apart, local search may not succeed and a *fallback* mechanism should be used that avoids frequent global flooding and adapts to membership dynamics. To achieve this, we introduce a novel *bootstrap* anycast scheme for multicast service, discussed next.

C. Resource Discovery and Multicast Address Allocation

A major challenge for multicast resource discovery (i.e., discovery of groups/senders) is the lack of infrastructure to hold and distribute group information. Inter-domain multicast for wired networks[30][3] utilizes the AS hierarchy of the Internet and uses BGP extensions to distribute multicast routes. AS hierarchy does not exist in ad hoc networks.

---

[2] We call this concept multicast-based mobility[1]. Our simulations show that, on average, control messages traverse 2.5 hops to reach the nearest point of the multicast tree in Internet-like topologies.

An architecture for *anycast* routing in the Internet was recently proposed in[18]. Utilizing hierarchical routing and aggregation, this work provides a scalable mechanism to discover members of anycast groups that are closer to the requester than other members. The architecture identifies one low-overhead mechanism for non-popular groups using default routes to route requests to the domain derived from the anycast address, and another mechanism for popular groups that caches routes for nearby members.

The only global infrastructure we can probably utilize in ad hoc networks is *geographic location*. Based on geographic multicast address allocation, we devise a new adaptive anycast architecture as follows. The multicast address space is broken into prefixes. Each multicast address prefix is assigned to a geographic region called the *rendezvous region (RR)*[3]. Nodes located in the *RR* have a *collective* responsibility of maintaining information about the groups belonging to the group prefix assigned to their current region. Since it is *collective* and can be done by a subset of *SDS* nodes (say 3-7 uncorrelated nodes), we use a probabilistic promotion scheme for nodes to become SDS for the group prefix. Each node decides locally whether it will become *SDS* based on its own configuration (nodes maybe configured as servers), capabilities (e.g., GPS), power and stability estimates. If so, it obtains its (approximate) geographic location[4] and determines the group prefix to which its current location maps, using algorithmic mapping in the general form of $f(x_1\text{-}x_2,\ y_1\text{-}y_2)=G_{prefix}$, or similar[5]. At that point, the node acts as a member of the anycast group of *SDS*s responsible for G_{prefix}, and advertises this information in its zone and to its contacts[6]. Other *SDS*s for the same prefix reply to update the new *SDS*. As *SDS* nodes move out of the RR for the corresponding G_{prefix}, they advertise their latest group information and *leave* message to the RR (using geocast[29], for example), which increases the probability of other nodes promoting themselves to become *SDS*s. This ensures constant replenishing of the pool of *SDS*s serving as members of the anycast group in that *RR*[7].

Session Initiation A node initiating a multicast session is expected (without necessity) to use the above algorithmic mapping to obtain a multicast address that maps into a geographical vicinity as its *RR*. In any case, group/session initiation requests/updates are sent to the *RR* to avoid collisions in multicast address allocation. When a new sender of a group belonging to G_{prefix} becomes active it performs a localized broadcast (as described earlier) and, if far from its *RR*, issues an update to *RR*. The requests and updates are sent using *lollipop-LAR* (our modified location aided routing (LAR)[22]) to improve scalability. In *lollipop-LAR*, a far away sender chooses a contact that is closer to the *RR*. If the distance between the contact and the *RR* is less than a limit *l*, the contact sends the request/update to the *RR* directly using LAR, otherwise it chooses one of its contacts closer to the *RR*, and so on. An illustration is shown in Fig. 3. (a).

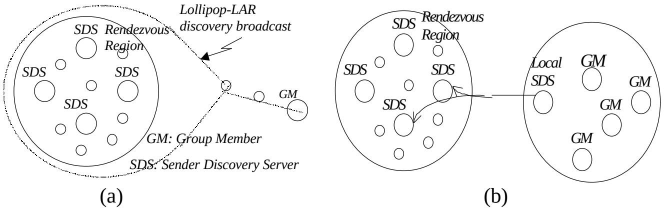


Fig. 3. (a) A group member uses algorithmic mapping to obtain the *rendezvous region (RR)* of the group, then uses *lollipop-LAR* to reach the region and contact a sender discovery server for group information. (b) If the group becomes popular in a region away from the RR a *subgroup* is formed and a local sender discovery server is chosen to contact the RR.

When a node joins the group, it first attempts localized search for *SDS*. If this fails it sends a join query to the *RR* using lollipop-LAR. A join reply is issued by a *SDS* in the *RR* and follows the backward route created by the join query (this path is only used for the reply so it need not be the shortest path). Once the receiver has group/sender information it sends a Join request as described earlier to establish multicast distribution path(s).

Popularity-based Dynamic Adaptation If used for the common case, geocasting to the *RR* may incur a lot of overhead to maintain group information. Hence, our multicast service and resource discovery paradigm should adapt to dynamics of membership, to achieve better performance for popular groups. As explained above, joining or sending to groups entail local advertisement (Adv) or query. This gives indications of the popularity of the group in the vicinity of the participants. Nodes receiving Advs and queries, those that are willing to become *SDS* for that group (based on their configuration, stability and capability), estimate the popularity of the group. The initial estimate is based on Advs/queries heard. If this initial estimate exceeds a threshold $pop_{query\text{-}th}$ a local group query is sent by the candidate-*SDS* to its own zone (using pro-active routing updates readily sent) and to its contacts. Response to this group query gives a better estimate of group popularity Grp_{est}, in addition to information about existing *SDS*s nearby SDS_{est}. Popularity estimate pop_{est} is be obtained as $pop_{est}\,\alpha\,\frac{Grp_{est}}{SDS_{est}}$. If $pop_{est} > pop_{th}$ where pop_{th} is the popularity threshold, then a node advertises itself as *SDS* for the group to its zone and contacts, and contacts the *RR SDS* for updates (using lollipop-LAR). Future nearby join queries for the group reach the local *SDS* and are answered locally, reducing overhead and delay. Furthermore, this adaptive

---

[3] This does not imply that these groups are limited to that region.

[4] Geographic location update need not be done frequently, only when a node moves noticeable distances. Afterall, this information is approximate and is used for distributed resource discovery (not for forwarding packets).

[5] Nodes express their desire to use multicast to their neighbors, zone or contacts , which will result in a reply containing the well-known algorithmic mapping function. Our scheme allows for changing this mapping to another well-known mapping (very infrequently, though).

[6] Alternatively, a node may advertise this information using localized geographically-scoped broadcast (Geocast[29]).

[7] The above scheme requires approximate knowledge of geographic location. We do not assume that all nodes are GPS capable. We do assume, however, that nodes are heterogeneous; i.e., *some* nodes are GPS capable, while others use GPS-less techniques[36] to discover their approximate relative location.

mechanism also achieves better robustness and continued operation during network partitions, when RR is unreachable. Fig. 3 (b) illustrates the popularity-based adaptation scheme.

Discussion We note that the probability of success of the localized search is affected mainly by two factors. The first is the group address, obtained during session initiation, which decides the location of the *RR*, and in turn determines the location of the *RR-SDSs*. The second factor is the nearby popularity of the group, which decides the promotion of nearby *SDSs*. Many of the offered services are expected to be *location-based* services, meaning it is targeted to a specific location. Hence, these groups will tend to be popular within certain locations more than others. In addition, initiators are expected to choose group addresses that have *RR* in the geographical vicinity. Both these factors increase the probability of success for localized search, and lead us to believe that, in the common case, our architecture is capable of high performance, with low overhead and low delays.

One essential question to ask here is 'how do participants know about newly initiated sessions and their properties?' This can be provided using the same scheme provided above, as follows. When multicast participants express their interest in multicast service, they obtain the algorithmic mapping function (described above) as well as a *well-known session advertisement* group address. Like other groups, this well-known group has its own *RR*. As groups are initiated, they are updated at the *RR-SDS* and the local *SDSs* (if any), then information about new sessions is obtained as above. This provides a *bootstrapping* mechanism essential for providing the multicast service.

IV. RELATED WORK

Related work lies in the areas of ad hoc routing, cluster formation and inter-domain multicast. Ad-hoc routing protocols may be pro-active (table-driven) or re-active (on-demand) protocols. Pro-active protocols include DSDV[6], CGSR[8], and WRP[9], and rely upon routing updates to maintain consistency of route information. Re-active protocols include AODV[10], DSR[11], TORA[12] and create routes only when required by the source node. In fisheye state routing (FSR)[38] the route update frequency to a certain destination is inversely proportional to the distance (in hops) of the destination. This reduces route overhead and reduces accuracy of routing with distance. Routing efficiency decreases and delay increases, however, with dynamics of mobility, and routing table size grows linearly with network size[25]. CEDAR[27] builds a core graph consisting of the minimum dominating set of nodes. These mechanisms were designed for small to medium size network. Works on multicast ad-hoc routing in[13][14] generally extend existing multicast routing for the Internet, such as PIM-SM[2]. Other multicast ad hoc routing protocols include tree-based and mesh-based protocols. Tree-based protocols include AMRoute[20] and AMRIS[21]. AMRoute creates a bi-directional shared core-based tree using unicast tunnels, but incurs temporary loops and chooses sub-optimal routes with mobility. AMRIS uses a shared tree, broadcasts new-session messages and uses beacons to detect disconnection and re-join to parents. However, for branch re-construction it uses expanding ring search, which does not scale well. Mesh-based protocols include ODMRP[23] and CAMP[31]. CAMP uses a shared mesh, and all nodes keep membership, routing and packet information. New members use expanding ring search to find other members. CAMP needs a special unicast protocol for its proper operation. ODMRP floods packets within mesh, but follows an on-demand policy to build and update the mesh. It uses request and reply phases, and broadcasts source announcements. The mesh is created when join requests from multiple receivers are sent to multiple-sources. Hence, for sparse groups or single-sender groups ODMRP may not be robust. A local route recovery scheme[35] may be used to address this problem. In our routing protocol, we utilize the concept of mesh construction and local recovery, but we attempt to avoid floods for resource discovery, using a contact-based query approach. We are not aware of other work on multicast for wide-area ad hoc networks or on *bootstrapping* multicast service, resource discovery or multicast address allocation in ad hoc networks. LANMAR[24][25] uses the landmark hierarchy concepts to establish hierarchy in ad hoc networks. However, landmarks are used for sets of nodes moving together as a group to reduce routing information exchange. ZHLS[32] is a GPS-based routing protocol for ad hoc networks, where a network is divided into non-overlapping zones. A node only knows node connectivity within its zone and the zone connectivity for the network. This architecture does not use cluster head to mitigate traffic concentration and avoid single point of failure. However, the zone map is defined by design for interzone routing, and hence does not adapt to network changes and dynamics. Another protocol called the zone routing protocol (ZRP)[15][33] was discussed in Section 2.1. In[17] the ZRP approach is coupled with geographic (geodesic) routing for remote routing. In[7] the link availability model is proposed (discussed in Section 2.2). The authors suggest to use it with a cluster based approach, in which a parent is selected based on the availability model to increase the lifetime of the cluster. Parent selection and cluster dynamics may complicate our architecture. Instead, we propose to incorporate the availability model with our modified ZRP approach and to use it for determining zone size, and in choosing contacts.

In the Internet, Hierarchical PIM[30] was proposed as an inter-domain architecture based on the PIM-SM protocol and suggests a hierarchy of cores. The BGMP architecture[3] was proposed for inter-domain multicast. It uses a bi-directional shared tree and the notion of root domains. The problem of multicast address allocation is coupled with BGMP for the choice of the root domain. The same study proposes the MASC scheme for multicast address allocation. Such problem is still active in research.

V. CONCLUSION AND FUTURE WORK

We have presented a new *architecture for multicast service* support in *large-scale ad hoc* networks. Our architecture is based on the zone-based routing concept, but extends it using our novel concept of *contacts* to increase zone coverage and reduce route and resource discovery overhead. Our mechanisms are *self-configuring* and *highly adaptive* to network dynamics and *mobility*, which renders our architecture more *robust*, *efficient* and *scalable*. We also provide a new architecture for *adaptive anycast* in ad hoc networks and a new scheme for geographic based multicast address allocation based on our concept of *rendezvous regions*. We hope that these mechanisms can potentially provide the *resource discovery* component for a wide-array of future applications and middle-ware. Our future work includes thorough evaluation of the architecture and fine-tuning of the mechanistic parameters presented herein.

More specifically, we plan to study the characteristics of the resulting network/small-world graphs due to the use of *contacts* and the probability p of choosing a contact. These characteristics, we expect, will be function of time and depend on the initial choice of contacts and node mobility. Furthermore, our plans include thorough analysis of the performance of the architecture as function of the popularity threshold pop_{th} and other popularity-based parameters. Our hope is that our work provides a framework for further research in the area of multicast (and other areas) in large-scale ad hoc networks.

REFERENCES

[1] A. Helmy, "A Multicast-based Protocol for IP Mobility Support", ACM SIGCOMM 2nd Int. Workshop on Networked Group Comm. Nov 2000.

[2] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, V. Jacobson, M. Handley, C. Liu, P. Sharma, "Protocol Independent Multicast – Sparse Mode (PIM-SM): Protocol Specification", *RFC 2362*, Mar '98.

[3] S. Kumar, P. Radoslavov, D. Thaler, C. Alaettinoglu, D. Estrin, M. Handley, "The MASC/BGMP Architecture for Inter-domain Multicast Routing", *Proceedings of ACM SIGCOMM*, August 1998.

[4] D. Estrin, M. Handley, A. Helmy, P. Huang, D. Thaler, "A Dynamic Bootstrap Mechanism for Rendezvous-based Multicast Routing", Proceedings of IEEE INFOCOM '99, New York, March 1999.

[5] S. Deering, D. Cheriton, "Multicast routing in data internetworks and extended lans", ACM Trans. Computer Systems, pp 85-111, May 1990.

[6] C. E. Perkins, P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance Vector Routing for Mobile Computers", CCR Oct. 1994.

[7] A. McDonald, T. Znati, "A mobility-based framework for adaptive clustering in wireless ad hoc networks", IEEE Journal on Selected Areas in Communications, V. 17 (8), pp 1466-1487, Aug. 1999.

[8] C.-C. Chiang, Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel, Proc. IEEE SICON '97, pp. 197-211.

[9] S. Murthy, J. J. Garcia-Luna-Aceves, An Efficient Routing Protocol for Wireless Networks, ACM Mobile Networks and App. J. on Routing in Mobile Communication Networks, Oct. 1996, pp. 183-197.

[10] C. E. Perkins, E. M. Royer, Ad-hoc On-Demand Distance Vector Routing, 2nd IEEE Wksp Mobile Comp Sys&Aps. Feb. 1999, p 90-100.

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

[12] V.D. Park, M.S. Corson, A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks, INFOCOM '97, Apr. 1997.

[13] J. J. Garcia-Luna-Aceves, E. L. Madruga, A Multicast Routing Protocol for Ad-Hoc Networks, Proc. IEEE INFOCOM '99, pp. 784-792, 1999.

[14] M. Gerla, C.-C. Chiang, L. Zhang, Tree Multicast Strategies in Mobile, Multihop Wireless Networks, ACM Mobile Networks&Apps. J., 1998.

[15] Z. Haas, "A new routing protocol for the reconfigurable wireless networks", *IEEE Conf. on Universal Personal Comm.*, pp. 562-6, '97.

[16] D. J. Watts. In *Small Worlds, The dynamics of networks between order and randomness*. Princeton University Press, 1999.

[17] L. Blazevic, S. Giordano, J. Boudec, "self-organizing wide-area routing", SCI/ISAS July 2000,

[18] D. Katabi, J. Wroclawski, "A framework for scalable global IP anycast (GIA), Proceedings of ACM Sigcomm 2000.

[19] 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.

[20] E. Bommaiah, M. Liu, A. McAuley, R. Talpade, "Ad-hoc Multicast Routing Protocol", *Internet-draft*, Aug. 1998.

[21] C. Wu, Y. Tay, C. Toh, "Ad hoc Multicast Routing Protocol utilizing Increasing id-numbers (AMRIS): functional specification, *I-D*, Nov 98.

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

[23] S. Lee, M. Gerla, C. Chiang, "On-demand multicast routing protocol", IEEE WCNC, p. 1298-1302, vol. 3, 1999.

[24] S. Lee, W. Su, M. Gerla, "Ad hoc wireless multicast with mobility prediction", IEEE ICCN, p. 4-9, 1999.

[25] P. Guangyu, M. Gerla, X. Hong, LANMAR: landmark routing for large scale wireless ad hoc networks with group mobility, MobiHOC '00.

[26] C. Chiang, M. Gerla, L. Zhang, "Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks", ACM/Kluwer Journal of Cluter Computing, vol. 1, no. 2, 1998.

[27] R. Sivakumar, P. Sinha, V. Bharghavan, "CEDAR: a core-extraction distributed ad hoc routing algorithm", IEEE JSAC, p 1454-65, Aug. 99.

[28] D. Estrin, D. Farinacci, A. Helmy, V. Jacobson, L. Wei, "Protocol Independent Multicast – Dense Mode (PIM-DM): Protocol Specification", Proposed RFC of the *IETF/IDMR*, September 1996.

[29] Y. Ko, N. Vaidya, "Geocasting in mobile ad hoc networks: location-based multicast algorithms", IEEE WMCSA, p. 101-110, 1999.

[30] S. Deering, W. Fenner, D. Estrin, A. Helmy, D. Farinacci, L. Wei, M. Handley, V. Jacobson, D. Thaler, "Hierarchical PIM for Inter-Domain Multicast Routing", Internet-Draft of the *IETF/IDMR*, December 1995.

[31] J. J. Aceves, E. Madruga, "The Core-Assisted Mesh Protocol", *IEEE JSAC*, vol. 17, no. 8, pp. 1380-1394, August 1999.

[32] M. Ng, I Lu, "A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks", *IEEE JSAC*, pp. 1415-1425, Aug. 1999.

[33] M. Pearlman, Z. Haas, "Determining the optimal configuration for the zone routing protocol", IEEE JSAC, p. 1395-1414, 8, Aug 1999.

[34] D. Watts, S. Strogatz, "Collective dynamics of 'small-world' networks", Nature, Vol. 393, June 4, 1998.

[35] M. Ize, Y. Kim, "PatchODMRP: an ad-hoc multicast routing protocol", ICIN, p. 537-543, 2001.

[36] S. Capkun, M. Hamdi, J. Hubaux, "GPS-free positioning in mobile ad-hoc networks", ICSC, p. 3481-3490, 2001.

[37] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, V. Jacobson, M. Handley, C. Liu, P. Sharma, L. Wei, "Protocol Independent Multicast (PIM): Motivation and Architecture", Proposed RFC, Oct 96.

[38] G. Pei, M. Gerla, T. Chen, "Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks", ICC 2000.