Dynamic Group Communication in Mobile Peer-to-Peer Environments

Eiko Yoneki and Jean Bacon
University of Cambridge Computer Laboratory
Cambridge CB3 0FD, United Kingdom
{eiko.yoneki, jean.bacon}@cl.cam.ac.uk

ABSTRACT

This paper presents an approach to integrate publish/subscribe semantics with on-demand multicast in wireless ad hoc networks, providing dynamic group communication with fine-grained subscriptions. A prototype application 'PGM Golf Forum' is developed, which enables a dynamic publish/subscribe system among the gallery during a golf tournament. Dynamic construction of an event dissemination structure to route events from event publishers to subscribers is important, especially to support rapidly changing multi-hop topologies over wireless links. The proposed approach aggregates content-based subscriptions in a compact data format (Bloom filters), and ODMRP (On-Demand Multicast Routing Protocol) is extended to use this context from the middleware-tier to construct an optimized dynamic dissemination mesh. It operates dynamic multicast grouping by aggregated subscriptions applying clustering techniques. Cooperation between middleware-tier and network components provides reliable fine-grained publish/subscribe by ODMRP's mesh topology.

Categories and Subject Descriptors

C.2.2 [**Computer Communication Networks**]: Network Protocols—*Routing Protocols*; C.2.4 [**Computer Communication Networks**]: Distributed Systems—*Distributed Applications*

General Terms

Algorithm, Design

Keywords

Publish/Subscribe, Mobile Ad Hoc Networks, On-Demand Multicast, Content-Based Subscription, Cross Layering

1. INTRODUCTION

There is a diversity of large scale network environments from Internet-scale peer-to-peer systems to sensor networks. Mobile ad hoc networks (MANET) lie between these extremes. MANET is a dynamic collection of nodes with rapidly changing multi-hop topologies that are composed of wireless links. The combination of mobile devices such as PDAs and ad-hoc networks is best managed by the creation of highly dynamic, self-organizing, mobile peer-to-peer systems. In such systems, mobile hosts continuously change their physical location and establish peering relationships with each other based on proximity. Asynchronous communication is essential to support anonymous coordination of communication in such ubiquitous environments. There have been efforts to create efficient multicast communication for MANET (see Section 2.2). When the group communication is fairly static and predefined channels can be deployed, multicast can be efficient. However, if the group gets larger and a multicast group must define a specific topic for each receiver, such as a consumer alert with a fine-grained subscription, it requires a more efficient communication mechanism such as the publish/subscribe paradigm.

Event-based middleware is based on publish/subscribe communication paradigms. It became popular, because asynchronous and multipoint communication is well suited for distributed computing applications. Most distributed event-based middleware contains three main elements: a producer who publishes events (messages), a consumer who subscribes his interests to the system, and an event broker with responsibility to deliver the matching events to the corresponding consumers. The first event-based middleware systems were based on the concepts of group (channel) or subject (topic) communication. These systems categorize events into predefined groups. In an attempt to overcome the limitation on subscription declarations, the content-based model [4], [7] has been introduced, which allows subscribers to use flexible querying languages to declare their interests with respect to event contents. Events are routed based on their content and consumer interests.

To achieve improved asynchronous and one-to-many communication systems in MANET environments, the semantics of publish/subscribe must be introduced. Maintaining group membership and efficient delivery of events to all members is challenging. Nodes that are increasingly mobile have a limited transmission range; a source-to-destination path could pass through several intermediate nodes, leading to a dramatic increase in complexity. Context-awareness is important for improving accuracy of data dissemination and performance in ubiquitous environments. Many mobile ad hoc routing protocols deploy contexts such as location, topology, and mobility patterns to construct optimized routing. Integration of the contexts from application through middleware to network components is becoming important.

SAC'05 March 13-17, 2005, Santa Fe, New Mexico, USA

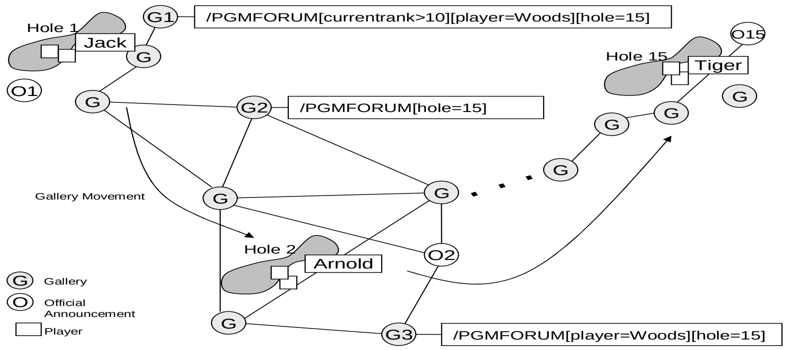

Figure 1: Prototype Application: PGM Golf Forum

We propose an adaptive message dissemination mechanism initially introduced in [21] for a content-based messaging system in MANET, which extends ODMRP (On-Demand Multicast Routing Protocol) [5] and integrates with content-based publish/subscribe. ODMRP is a multicast protocol for MANET, and builds a routing table on demand, and it has performed well with regard to throughput and control packet overhead in simulation studies [9]. ODMRP supports optimized data dissemination mechanisms with context awareness, including location, network topology, network ability (e.g., bandwidth and stability), and mobility. In this paper, we extend the definition of context further to the conditions set by the middleware. Dynamic construction of the event dissemination structure is essential to support messaging systems in ubiquitous environments. To optimize construction of an event dissemination structure, the proposed system defines an interface to apply the context from a publish/subscribe system to ODMRP. The interface is generic; to supply data to be attached to ODMRP packets and indicate how to process them. Our proposal is conceptually similar to Active Networks, which allow users to inject customized programs into the nodes of the network. Currently the context in the proposed system includes subscriptions and events themselves. Content-based subscriptions at a broker node are aggregated and summarized into a compact data format in Bloom filters [3], and the event publisher broker node operates multicast grouping by examining the propagated subscriptions. This dynamic group setting is one of the difficult challenges. Context-awareness allows both middleware and network layer components to exploit information providing an efficient and dynamic event routing mechanism for better performance.

As a prototype application, we developed 'PGM Golf Forum', which enables real time content-based publish/subscribe among the gallery during the golf tournament. The overview is depicted in Fig.1 (see more details in section 4). Initially the gallery obtains the software including players' list. Subscriptions based on the defined categories may be set anytime. Categories are players, holes, scores etc. and combinations of categories will be various. The gallery moves along the players using PDAs obtaining other players' information. Along the game, a participant may publish messages or change the subscription. Gallery G1, G2, and G3 have similar subscriptions and matching messages may be delivered in the same multicast channel depending on the topology of publishers and timing of publishing messages. The proposed approach constructs dynamic channels from subscriptions and optimizes the traffic, while maintaining changes of subscriptions and movement of devices. At the end of the game, this mechanism may be used for taxi sharing. The access point based 802.11 communication range is normally about 200m. On the other hand, ad hoc communication can extend the communication capability to cover the whole golf course. There could be another solution such as use of satellite or radio, but the proposed dynamic group communication does not require any infrastructure.

This paper continues as follows: section 2 describes the background and related work, section 3 describes the underlying messaging system, section 4 describes the prototype and experiment, and it concludes with section 5.

2. BACKGROUND AND RELATED WORK

For the past several years, many publish/subscribe systems have been reported, including Gryphon[7], SCRIBE [14], and SIENA [4]. SCRIBE is a topic-centric publish/subscribe messaging system using Distributed hash tables (DHT) over Pastry [15]. SCRIBE depends on Pastry to route messages to the destinations. Several publish/subscribe systems implement some form of content-based routing such as SIENA. SIENA is a notification service scalable to wide-area networks. Routing strategies in SIENA use two classes of algorithm: advertisement forwarding and subscription forwarding. JEDI and SIENA have an extension to support mobility by explicit movement operation to relocate clients. Nevertheless, none of these systems support extremely dynamic mobile environments such as mobile peer-to-peer networks.

Several middleware systems have been developed to support wireless ad hoc network environments such as Rebeca [6], STEAM [11] and IBM WebSphere MQ [8]. STEAM provides a proximity-based group communication. These systems construct publish/subscribe paradigm on the top of existing transport protocols. Our approach takes cross layering between middleware-tier and network components. In wireless ad hoc network environments, much research currently focuses on general datagram routing in both unicast and multicast routing. However, no definite solution to define publish/subscribe semantics using these protocols has been provided. Given the characteristics of mobile ad hoc networks, the use of distributed hash tables in a messaging system to locate objects may not work in dynamic environments. Thus, our approach is unique in taking advantage of the broadcast nature of wireless ad hoc networks and creating dynamic multicast channel by grouping subscriptions.

2.1 Subscription Models

One of the key issues in supporting messaging systems is designing subscription models. Topic-based addressing is an abstraction of numeric network addressing schemes. With the content-based subscriptions used in SIENA and Gryphon, delivery depends only on message content, extending the capability of event notification with more expressive subscription filters. The most advanced and expressive form of subscription language is content-based with pattern matching; such a language is important for event notification. Common topic-based systems arrange topics in hierarchies, but a topic cannot have several super topics. Type-based subscription provides a natural approach to this if the language offers multiple sub-typing, thus avoiding explicit message classification through topics. This works well with typed languages, but it is complex to deploy this degree of serialization of objects. Moreover, mobile applications may not have the concept of objects or typing. Thus, the combination of hierarchical topics and high speed content filtering could provide a more flexible approach for mobile applications. Our approach fits into this category. Research is also ongoing to structure complex content-based data models [12] and reflection-based filters.

2.2 Multicast Protocols in MANET

For wireless networks, the most natural form of communication is broadcasting. The publish/subscribe messaging model maps well onto a decentralized group structure in MANET. In general, there are three types of routing mechanism in MANET. Proactive (table driven) protocols use a traditional distributed shortest-path and maintain routes at all times. They impose a high overhead. Reactive (on-demand) protocols are initiated by the source. They have lower traffic compared with proactive protocols, but impose delay in route determination. Hybrid protocols are an adaptive way of combining proactive and reactive protocols. One important difference from static network environments is that routing optimality is not the highest priority because of dynamic environments. Note that the multicast service model is less powerful than that of a content-based network model, and there is currently no optimal way of using or adapting the multicast routing infrastructure to provide a content-based service. Several multicast routing protocols for MANET have been developed. DVMRP (Distance Vector Multicasting Routing Protocol) builds a source-based tree. AODV (Ad-hoc On-Demand Distance Vector Routing Protocol) builds a core-based tree. CAMP (Core Assisted Mesh Protocol) builds a mesh interconnection of hosts.

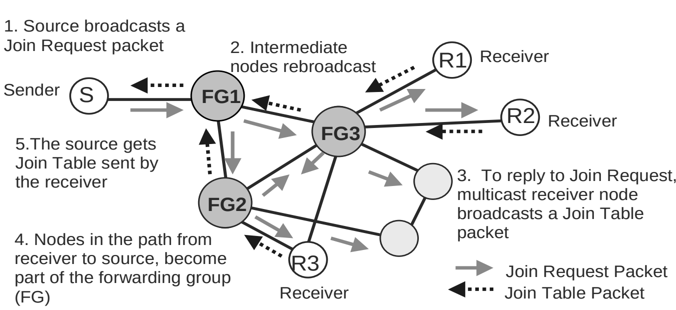

Figure 2: ODMRP Operation

ODMRP [5] applies an on-demand routing technique to avoid channel overhead and improve scalability (see Fig. 2). ODMRP is a result of an incorporation of FGMP (Forwarding Group Multicast Protocol) and an on-demand scheme. It attempts to create a group of forwarding nodes between the source and the multicast receivers. These forwarding nodes re-broadcast any packet they receive to reach all interested multicast receivers. The multicast mesh is created through a reply-response phase that is repeated at intervals to keep the routes to the multicast receivers fresh. With the concept of a forwarding group, only a subset of nodes forwards multicast packets (scoped flooding). ODMRP provides a richer connectivity among multicast members using a mesh-based approach. It supplies multiple routes for one particular destination, which helps in case of topology changes and node failure. ODMRP takes a soft-state approach to maintain multicast group members. Nodes need not to send any explicit control message to leave the group. However, some issues need to be considered, e.g. an increase of senders leads to control overhead.

3. UNDERLYING MESSAGING SYSTEM

The proposed messaging system follows the basic model of an event-based middleware. Publisher, subscriber, and broker are the elements of the system. The messaging system in MANET should be self organizing, because the topology of a mobile peer-to-peer system has to constantly adjust itself, by discovering new communication links, and also needs to be fully decentralized due to the lack of a central access point. In such environments, it is best to create a routing table on demand, according to the simulation studies [9]. ODMRP is simple and scalable, which avoids the drawbacks of multicast trees such as intermittent connectivity and frequent tree configuration. Thus, ODMRP is chosen for the base routing protocol. ODMRP improves its performance using mobility and location information. Many contexts belong to the network and are outside the scope of middleware. On the other hand, the semantic contexts from upper layers should be used for building an efficient communication by the network layer component. There is a need to exchange contexts among applications, the middleware tier, and the network layer component to build an optimized data dissemination structure. Thus, content-based subscriptions are used when ODMRP builds the routing table to disseminate the events. Content-based routing is computationally more expensive than explicit-address routing or topic-based routing. In static peer-to-peer network environments, content-based subscriptions are used to construct routing itself. However, over dynamic mobile ad hoc network environments, the multicast channel life is ephemeral and subscriptions are expected to be more specific. In such network environments, epidemic style data dissemination should work better. Thus, our approach to establish a dynamic channel would have good use for realizing content-based publish/subscribe. Note that it is necessary to develop data structures and algorithms that can introduce efficient subscription propagation and event matching. Naturally XML is a good candidate, although it lacks typing.

The proposed system currently defines events in XML format with XML schema, and XPath [2] is used as a subscription language. Processing XML in resource constrained devices could be expensive, and the subscriptions are tightly linked to the corresponding event data structure in the current system. We are investigating an integration with an ontology-based event data model [20], that will provide light weight Resource Definition Framework (RDF). Aggregated subscriptions are kept in a compact data structure within the subscriber broker. For compact encoding, Bloom filters are used. For the resource constrained devices which may not have capability to process XML, Bloom filters can be produced from the user input directly. Furthermore, the digest of publishing events and event advertisements are also transformed to the compact data structure which travel within a multicast packet header. When event digests reach the subscriber broker, a matching operation between the event digest and subscription decides either to join the particular multicast group or not. The publishing broker determines the multicast group from the propagated subscriptions. The challenge is to set the optimal subscription for the multicast group. Brokers can propagate any changes in subscriptions to ODMRP's periodic membership refreshing mechanism.

3.1 XML-based Typed Event (Message)

An event is defined with XML schema, and the root element name identifies the event type.

```
<?xml version="1.0" encoding="UTF-8"?>
<PGMFORUM id="001" timestamp="1003-02-27T12:00:00.000-00:00"
        xmlns="http://www.cl.cam.ac.uk/~ey204/lib/">
    <player>woods</player>
    <hole>15</hole>
    <holescore>4</holescore>
    <totalscore>52</totalscore>
    <currentrank>2</currentrank>
</PGMFORUM>
<PGMFORUM id="002" timestamp="1003-02-27T12:00:00.000-00:00"
        xmlns="http://www.cl.cam.ac.uk/~ey204/lib/">
    <player>palmer</player>
    <hole>2</hole>
    <holescore>4</holescore>
    <totalscore>8</totalscore>
    <currentrank>7</currentrank>
</PGMFORUM>
```

Figure 3: Messages Instance Examples

Fig.3 shows example messages. The XML schema for the event consists of a set of typed elements. Each element contains a type and a name. The element's name is a simple string, while the value can be in any range defined by the corresponding type. Events themselves can be carried in byte stream or compressed format in ODMRP packets.

3.2 Subscription Language

Most content-based publish/subscribe systems support a subscription language. The resulting filtering expression is then used by the brokers to determine whether a particular event notification is of interest to a subscriber. If the language is expressive, efficient matching of notifications will be difficult. However, if the language does not support rich constructs, its applicability is limited. A subset of XPath [2] is used as a filter-specification language. Complex XPath expressions will be transformed to simplified formats.

3.3 Compact Data Structures in Bloom Filters

Subscriptions are aggregated at the brokers and formed in compact data format. The event advertisement and notification are transformed to the compact format, which uses XPath as an intermediate expression during the transformation. Bloom filters are used for encoding. Bloom filters are compact data structures for probabilistic representation of a set in order to support membership queries. Each filter provides a constant-space approximate representation of the set. Errors between the actual set and the Bloom filter representation always take the form of false positives to inclusion tests. False positive probability decreases exponentially with linear increase in the number of hash functions and vector size. A Bloom filter is a bit vector, and items are inserted by setting the bits corresponding to a number of independent hash functions.

Given the constrained MANET environments, it is mandatory to aggregate the set of subscriptions to a compact set of content specifications. The proposed subscription summary structure in Bloom filters is based on the one described in [16], extended to take advantage of XML typed events by XML schema and the XPath subscription language (for more detail, see [21] .) Currently no hierarchical encoding is used. A subscription summary consists of five data structures described below, and a broker's subscription summary is an array of these data to keep the summarized per-broker subscription information (Fig.4 shows an example).

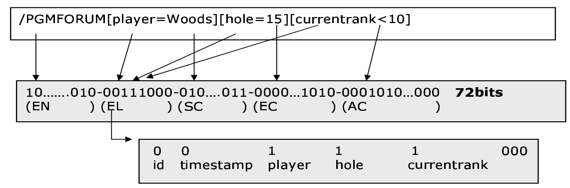

Figure 4: Encoding Subscriptions in Bloom filters

1. **Event Type Name (EN):** contains the hashed value of the root element name in the XML schema that is an identifier of the event type.
2. **Element Association List (EL):** stores information about the elements and attributes in XML schema and actual values that belong together in a subscription.
3. **Enumeration Constraint (EC):** holds the constraints of each enumerated string attribute of a subscription.
4. **Arithmetic Constraint (AC):** holds the constraints of each arithmetic attribute of a subscription.
5. **String Constraint (SC):** contains information about the constraints in the string type. For each different string type element/attribute, which appears in at least one subscription, a broker implements a SC structure using three bit vectors (SC_L, SC_R, SC_X) as Bloom filters. For containment operations, the specified string value is divided into two substrings, 'left' and 'right', defined relative to the position of the operator '*'. After the string value is divided into the two substrings, the left (right) substring is hashed and placed in the $SC_L(SC_R)$ filter for the specific string.

3.4 Global Grouping

The broker node hosting event publishing defines the multicast group from the propagated subscriptions by applying various clustering methods. Our subscription clustering has many common aspects with subscription partitioning and routing in wide area networks (see [1, 19, 18, 13]). In [18], there are two approaches to partition the overall publish/subscribe operations among multiple servers: either partition the event space or partition the set of subscription filters. In the Event Space Partitioning (ESP) approach, given the number of servers N_s, the d-dimensional space is partitioned into N_s disjoint subspaces, each assigned to a different server. A subscription is hosted by a server if its filter intersects the server's associated subspace. An event is forwarded only to the server whose subspace contains the point representing the event. The main advantage of the ESP approach is that it minimizes event traffic by forwarding each event to at most one server. The main disadvantage is that, if a subscription filter intersects multiple subspaces, the subscription need to be replicated on multiple servers, thus increasing the total number of subscriptions hosted by the network, as well as complicating the task of subscription management for subscription deletion or subscription state updates. Filter Set Partitioning (FSP) is a dual partitioning approach which always assigns each subscription to a single server. Similar subscriptions, i.e., those matched by the same events, are grouped together and assigned to the same server to allow for the construction of compact and effective summary filters. Summary filters from multiple servers may overlap and an event is forwarded to every server whose summary filter contains the event.

Our goal is similar to FSP approach. For content-based publish/subscribe, FSP approach has demonstrated good load balancing both statically and dynamically, while significantly reducing network traffic. In our system, the purpose of clustering for multicast grouping is to reduce the network load, and it is not necessary to create a hierarchical cluster even the propagated subscription can be in the hierarchical formation. Another aspect is that subscriptions may reach the publishing broker nodes in time series, and the algorithm that can accommodate better for the periodical update. Aggregation of subscriptions takes place over time and space. If the group covers a coarse-grain subscription, more noise will be delivered to the broker, while many groups will be created if it covers a fine-grain subscription. However, if a node has high mobility, setting a fine-grain subscription may prevent the overhead of group membership maintenance. The goal is to define a channel per entity, but there is no obvious one-size-fit-all solutions. It is challenging to define the balance between multicast groups and content-based subscriptions, and the optimized methods will depend on the application characteristics. Because of the dynamic MANET environment, a deterministic setting may not produce better performance than a probabilistic approach. Three different approaches are currently applied in order to obtain heuristic experimental results.

3.4.1 The least constrained subscription

This approach is set under the assumption of that subscriptions are client-specific in mobile ad hoc network environments, such as important alert messages. Thus, a subscriber who subscribes to everything, causes flooding and breaks the assumption.

3.4.2 Aggregated subscriptions in a graph

Aggregated subscriptions are described in graph representation, which keeps hierarchical patterns. When the publisher broker publishes messages, it searches any of the subscriptions in the graph and if it matches at least one of them, it publishes the message.

3.4.3 Grouping by K-means

This approach is to set the effective groups as the result of the K-means operation. K-means clustering [10] generates a specific number of disjoint, flat (non-hierarchical) clusters, and the algorithm is used for similarity-based grouping. One reason of using the K-means algorithm is to exclude subscriptions that are distant from the main group where the majority of subscribers are. The typical operation of K-means is:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

Our approach uses Bloom filters, which represent subscriptions converted from XPath for measuring the distance from the centroids. The initial K value is assigned either 2 or 3. A basic problem of K-means is that the number of the groups has to be known. Thus, some heuristic step is applied. K-means algorithms converge to a local optimum in a number of iterations, and in practice they converge quickly. Nevertheless, the processing can be stopped after any iteration, resulting in a feasible partition into K groups. This also provides an easy way to accommodate changes of grouping, simply running a number of re-balancing iterations, when new subscribers arrive or subscription changes.

3.5 Routing by Pub/Sub ODMRP

A broker can reside in an independent node or in the same node where publishers/subscribers reside. The routing between the broker and publishers/subscribers can be achieved by unicast protocols. This paper focuses on the routing between the broker publishing an event and the brokers subscribing to it. In general, there are two approaches to disseminate events to the corresponding subscribers. The first is flooding the message by broadcast, followed by filtering at the broker. The second is match-first and requires a precomputed destination list, which is broadcast to all brokers, followed by routing using the routing list. The flooding protocol is simple but may lead to network congestion. Match-first is not scalable, because the routing table must be shared by all brokers, and preprocessing may not work well in MANET environments. A MANET environment's dynamic network condition may cause frequent reconfiguration of routing tables. The third possibility is a distributed-approach by the brokers, where the brokers examine the message content and forward messages using the routing table. Our proposal is an integrated approach of the above third way with extending ODMRP (Pub/Sub ODMRP) that applies scoped flooding. The protocol operation of Pub/Sub ODMRP is described below (see Fig.5).

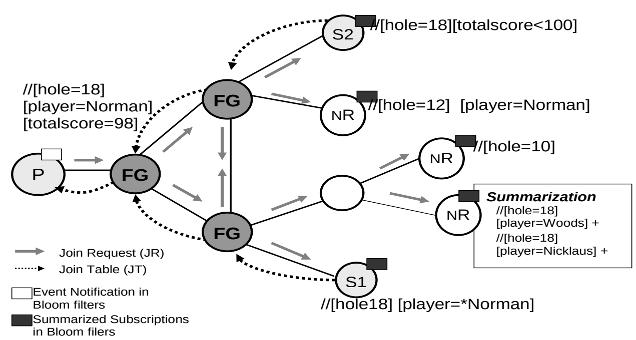

Figure 5: Pub/Sub ODMRP Operation

3.5.1 Join Request (JR) Operation

- A publisher broker node (P in Fig.5) wishing to send events periodically broadcasts a Join Request (JR) packet over the network.
- The digest of the event notification to be sent in the Bloom filter expression is attached to the JR packet.
- A node receiving a non-duplicate JR packet stores the upstream node ID and rebroadcasts the packet.

- Optionally, a JR operation provides 'advertisement' mode, which sends out a special event to establish the route before sending out the whole data. This can be deployed when the network is stable and pre-setting routing is beneficial.

3.5.2 Join Table (JT) Operation

- The subscriber broker nodes (NR as non-matching subscriber and S as matching subscriber in Fig.5) that keep the subscriptions in Bloom filters have the correct bits set for the subscription to be recognized as receiver nodes over the network.
- Once a subscriber broker node decides to join the group, it updates the publisher entry in its member table.
- Join Table (JT) packet is broadcast periodically. The subscription information is attached to the JT packet.
- An intermediate node (router node), receiving the JT packet, compares its node ID with the entries of the forwarding group table. If there is a match, it becomes a member of the forwarding group (FG in Fig.5). It sets the FG_Flag and broadcasts its JT packet. Optionally the subscription information is kept along the routing information, which is used as a filter for data forwarding.
- The JT packet is propagated by each forwarding group member until it reaches the publisher broker node via the shortest delay path. This process creates a mesh among all forwarding group members.
- The publisher broker node aggregates the subscription information for the group according to the received subscriptions in the JT packet.

3.5.3 Data Forwarding

After the group establishment and route construction process, a publisher broker can transmit 'Data packets' to subscriber brokers via selected routes and forwarding groups. Intermediate nodes relay a 'Data packet' only when it is not a duplicate, subscription information for routing destination matches and forwarding group membership has not expired. This whole operation minimizes traffic overhead and prevents sending packets through stale routes.

3.6 Subscription Maintenance

ODMRP maintains the group membership with periodic flooding, and subscriptions are also up-to-date using this mechanism. If there are any changes in subscriptions, brokers maintain the group setting. No explicit control message is required to join or leave the group. When the publisher broker leaves the group, it stops sending a JR packet, and the subscriber broker node avoids sending back a JT packet to leave the group. Forwarding group nodes demote to non-forwarding group nodes if no JR packet is received within the time out period. Periodic flooding often causes the broadcasting storm problem. Optimizing the refresh intervals (e.g. route refresh interval and forwarding group timeout interval) is critical for performance and reliability. The ODMRP scheme can use mobility prediction to adapt the route refresh interval. This scheme uses the location and mobility information provided by a system such as GPS (Global Positioning System) to determine how long routes will remain valid. The route refresh interval can then be adapted.

4. PROTOTYPE

A Java-based prototype of the proposed messaging system is developed on the top of Java ns-2 simulator, which includes Pub/Sub ODMRP as application level multicast, because a MANET routing multicast protocol is not yet integrated with the network layer in most commercially available mobile devices. The component structure is shown in Fig.6. Java Remote Method Invocation (RMI) communicates with a virtual multicast socket that can behave like a normal Java unicast/multicast socket. Thus, real world applications are able to run over the simulator. The security issues are out of scope of this paper.

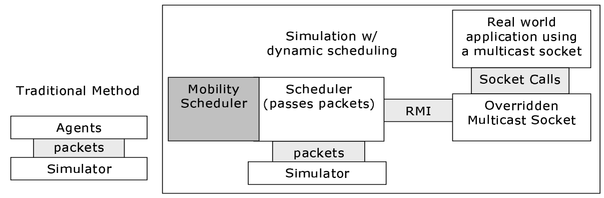

Figure 6: Components Structure of Simulator

On the top of the messaging system, we have developed the prototype application 'PGM Golf Forum'. This application provides dynamic group communication among the gallery during the golf tournament. In Fig.1, the gallery G1, G2, and G3 set subscriptions. The subscription registration user interface is shown in Fig.7. Broker participation option indicates if your device offers to be the broker node instead just to be the publisher/subscriber node. If powerful nodes become broker nodes, it helps resource constrained devices from data processing burden. The communication between brokers and clients is implemented with unicast in 'PGM Golf Forum'. Node O indicates official announcements at each hole, which regularly publishes messages. However, any gallery can publish a message. When a message is published, it establishes the routing table according to Pub/Sub ODMRP protocol. The gallery moves along the players using PDAs obtaining other players' information. Mobile peer-to-peer communication can expend the communication capability to cover an entire golf tournament area.

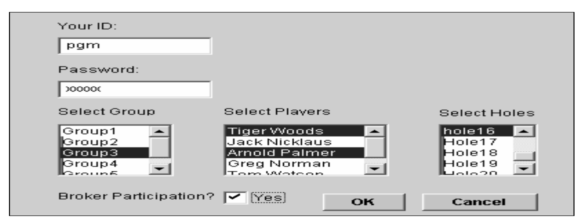

Figure 7: PGM Golf Forum: Register Subscription

Fig.8 shows the user interface when a message is published, and it is converted to one of XML messages shown in Fig.3. O15 in Fig.1 publishes a message, and it will reach all G1, G2 and G3 subscribers. Gallery nodes create forwarding mesh accordingly.

Figure 8: PGM Golf Forum: Message Publishing

When any notification arrives at the device, alerting is informed. The updated information is shown in Fig.9.

Figure 9: PGM Golf Forum: Notification

4.1 Experiments

Evaluation of the proposed system is complex, due to complexity of subscriptions and events, efficiency of global subscription aggregation, mobility patterns, network traffic load, false positive rate by Bloom filters, etc. In this paper, we show the characteristics of our approach in the aspects of clustering effects. See [22] for more generic experimental results. 100 nodes with 20 publishers and 50 subscribers are randomly placed over 1000m x 1000m area. A publisher broker sends 100 messages, which are matching 60 % of subscriptions. The route refresh interval is set to 5 seconds, and the forward group timeout interval is 25 seconds that are chosen as less influential values for the experimental environment.

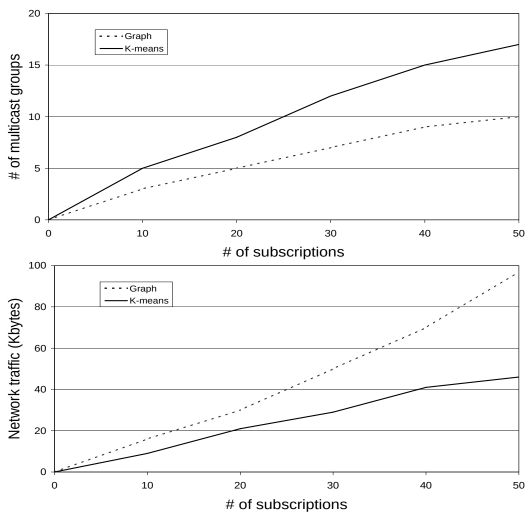

Figure 10: Cluster Algorithms Comparisons

Fig.10 shows algorithmic comparisons for global clustering in publisher brokers. K-means creates more channels than the graph form of aggregated subscriptions, but overall the traffic over the networks is reduced. This shows that K-means grouping provides efficient grouping. The simulation of node specific activity and group mobility will be required for the accurate experiments of 'PGM Golf Forum', which is out of scope of this paper.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we present an approach to integrate publish/subscribe semantics with on-demand based multicast routing in mobile ad hoc networks. A key contribution is the creation of an interface between the middleware tier and network components, for the construction of an efficient event dissemination structure. A dynamic multicast grouping is operated and maintained by aggregated subscriptions over the network. In dynamic mobile ad hoc network environments, the multicast channel life is ephemeral and subscriptions are expected to be more specific. In such network environments, our approach would have good use for efficient group communication and supporting ubiquitous computing. As an event-based middleware requires more than just a communication mechanism, further functions, most notably data aggregation and event correlation, will be the next focus in messaging over mobile ad hoc network environments.

Acknowledgment. This research is funded by EPSRC (Engineering and Physical Sciences Research Council) under grant GR/557303.

6. REFERENCES

[1] G. Banavar et al. An efficient multicast protocol for content-based publish-subscribe systems. In *Proc. of ICDCS*, 1999.

[2] Berglund, A. et al. XML Path Language (XPath) 2.0. Working Draft. In *W3C, http://www.w3c.org/TR/xpath20/*, 2001.

[3] Bloom, B. Space/time Trade-offs in Hash Coding with Allowable Errors. In *CACM, 13(7)*, 1970.

[4] Carzaniga, A. et al. Achieving scalability and expressiveness in an Internet-scale event notification service. In *Proc. 19th ACM Symp. on Principles of Distributed Computing*, 2000.

[5] Chiang, C. et al. On-Demand Multicast Routing Protocol. In *Proc. WCNC*, 1999.

[6] Fiege, L. et al. Supporting mobility in content-based publish/subscribe middleware. In *Proc. ACM/IFIP/USENIX Middleware*, 2003.

[7] IBM. Gryphon: Publish/Subscribe over Public Networks. In *http://researchweb.watson.ibm.com/gryphon/gryphon.html*.

[8] IBM. WebSphere MQ: Connecting your applications without complex programming. In *IBM WebSphere Software White Papers*, 2003.

[9] Lee, S. et al. A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols. In *Proc. INFOCOM*, 2000.

[10] MacQueen, J. Some Methods for classification and analysis of multivariate observations. In *Proc. Berkeley symposium on Mathematical Statistics and Probability*, 1967.

[11] Meier, R. et al. STEAM: Event-based middleware for wireless ad hoc networks. In *Proc. DEBS*, 2002.

[12] Muhl, G. et al. Filter Similarities in Content-Based Publish/Subscribe Systems. In *Proc. ARCS*, 2002.

[13] Riabov, A. et al. Clustering Algorithms for Content-Based Publication-Subscription Systems. In *Proc. ICDCS*, 2002.

[14] Rowstron, A. et al. SCRIBE: The design of a large-scale event notification infrastructure. In *Int'l. Workshop of NGC*, 2001.

[15] Rowstron, A. et al. Pastry: Scalable Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In *Middleware*, 2001.

[16] Triantafillou, P. et al. Subsciption Summaries for Scalability and Efficiency in Publish/Subscribe Systems. In *Proc. DEBS*, 2002.

[17] Wang, Y. et al. Subscription Partitioning and Routing in Content-based Publish/Subscribe Systems. In *Proc. 16th Symp. on Distributed Computing*, 2002.

[18] Wang, Y. et al. Subscription Partitioning and Routing in Content-based Publish/Subscribe Systems. In *Proc. 16th Symp. on Distributed Computing*, 2002.

[19] Wong, T. et al. An Evaluation of Preference Clustering in Large-Scale Multicast Applications. In *Proc. Infocom*, 2000.

[20] Yoneki, E. and Bacon, J. Towards Peer-to-Peer Event Broker Grid in Hybrid Network Environments. *Proc. OTM Workshop - GADA*, October 2004.

[21] Yoneki, E. and Bacon, J. Content-Based Routing with On-Demand Multicast. *Proc. 23rd ICDCS Workshop - WWAN*, March 2004.

[22] Yoneki, E. and Bacon, J. An Adaptive Approach to Content-Based Subscription in Mobile Ad Hoc Networks. *Proc. 2nd PerCom Workshop - MP2P*, March 2004.