An Adaptive Approach to Content-Based Subscription in Mobile Ad Hoc Networks

Eiko Yoneki and Jean Bacon
University of Cambridge Computer Laboratory
J J Thomson Avenue, Cambridge CB3 0FD, UK
Email: {eiko.yoneki, jean.bacon}@cl.cam.ac.uk

Abstract— **We describe a novel approach for content-based publish/subscribe system in mobile ad-hoc networks and show preliminary experiment results. We extended ODMRP (On-Demand Multicast Routing Protocol) using aggregated summaries of content-based subscriptions in Bloom filters expression for the dynamic construction of an event dissemination structure. ODMRP's mesh topology gives extra reliability. We describe experiment results including comparison with regular ODMRP and mobility simulation.**

I. Introduction

Mobile ad-hoc network (MANET) is a dynamic collection of nodes with rapidly changing multi-hop topologies that are composed of wireless links. The combination of mobile devices and ad-hoc networks allows 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 such MANET environments. There have been efforts to create efficient multicast communication for MANETs (see Section II). Maintaining group membership and efficient delivery of the packets to all members is challenging. To achieve improved one-to-many communication systems in MANET environments, event-based middleware must be introduced.

Event-based middleware became popular as asynchronous and multipoint communication is well suited for distributed computing applications. Most distributed event-based middleware in distributed network environments 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 pre-defined groups. In an attempt to overcome the limitation on subscription declarations, the content-based model [3] 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. A key challenge when building event-based middleware is efficient propagation of subscriptions among brokers. Brokers require this information to forward incoming events exclusively to interested consumers. Filtering out unrelated events can save significant overhead.

We propose a content-based publish/subscribe system for MANETs, which integrates an extended ODMRP (On-Demand Multicast Routing Protocol) [4] and content-based subscriptions. ODMRP builds a routing table on demand, and it has performed well with regard to throughput and control packet overhead in simulation studies [11]. ODMRP supports optimized data dissemination mechanisms with context awareness including location, network topology, network ability (e.g., bandwidth and stability), and mobility. Dynamic construction of an event dissemination structure is essential to support messaging systems in MANET 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 context is summarized subscriptions and notifications. The interface is generic to supply data to be attached to the ODMRP packet and indicate how to process them. Content-based subscriptions at a broker node are aggregated and summarized into a Bloom filter [2], and the event source broker node defines the multicast group by examining the propagated subscriptions. Context-awareness allows both middleware and network layer components to exploit information to provide an efficient and dynamic event routing mechanism for better performance.

II. Background

The messaging system in MANET should be self organized, because the topology of a mobile P2P system has to constantly adjust itself by discovering new links and also needs to be fully decentralized due to the lack of a central access point. In non-ad-hoc P2P networks, many prototype messaging systems have been reported, including Gryphon[8], JEDI [6], SCRIBE [15], and SIENA [3]. XRoute [16] takes an approach to aggregate subscription to a tree pattern. However, in ad-hoc networks, much research currently focuses on general datagram routing in both unicast and multicast routing, but no definite solution to define publish/subscribe semantics using these protocols has been provided. Given the characteristics of MANET, the use of distributed hash tables to locate objects may not work in dynamic environments.

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, to improve the performance of messaging systems, the combination of hierarchical topics and high speed content filtering could be a more flexible approach for mobile applications.

There are efforts to build content-based subscription with distributed hash tables by automatically organizing the content into several topics. Research is also ongoing to structure complex content-based data models [12] and reflection-based filters [7]. We are currently working on efficient distribution of event matching tasks over the network and brokers by establishing the concept of ”approximate matching”. The dynamic construction of event dissemination structure to route events from publishers to all interested subscribers is the most challenging task to support content-based subscription in distributed environments. For example, SIENA uses two classes of algorithm for routing strategies: advertisement and subscription forwarding.

Several Application-Level Multicast Routing Protocols (ALMRPs) have been designed. Most ALMRPs use tree routing for logarithmic scaling with respect to receiver numbers: as the node connectivity changes, the tree structure changes accordingly. Narada [13] and many subsequent designs can best be understood as two layered protocols: a protocol that maintains a mesh of hosts, and a multicast routing protocol on top of this mesh. Other examples are Bayeux/Tapestry [20] and CAN [14]. The multicast service model is less powerful than that of a content-based network, and there is currently no optimal way of using or adapting the multicast routing infrastructure to provide a content-based service. For wireless networks, the most natural communication type is broadcasting. There are several multicast routing protocols for MANET. 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. ODMRP applies an on-demand routing technique to avoid channel overhead and improve scalability (see Fig. 1). It will attempt 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 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 send any explicit control message to leave the group. It can work with any unicast protocol. For more detail on ODMRP, see [4].

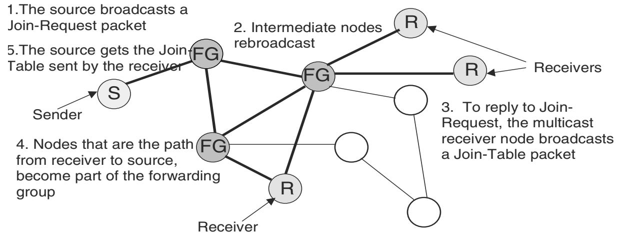
1.The source broadcasts a Join-Request packet
2. Intermediate nodes rebroadcast
5.The source gets the Join-Table sent by the receiver
FG
FG
R
R
Receivers
S
Sender
3. To reply to Join-Request, the multicast receiver node broadcasts a Join-Table packet
4. Nodes that are the path from receiver to source, become part of the forwarding group
FG
R
Receiver

Fig. 1: ODMRP Protocol

III. Publish/Subscribe System using extended ODMRP

The proposed publish/subscribe system follows the basic model of an event brokering system. In MANET environments, it is best to create a routing table on demand (see [5]). Thus, ODMRP is chosen for data dissemination mechanism. The MANET environment is dynamic, and the multi-hop routing has to consider many contexts from physical constraints, location, and mobility. Many contexts belong to the network, which 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. Here, there is a necessity 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. 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. Research is ongoing on subsumption of XML types [10]. Events are defined in XML format with XML schema, and XPath [1] is used as a subscription language. The subscriptions are tightly linked to the corresponding event data structure in this system, and summarized subscriptions are created using Bloom filters [2]. Bloom filters are compact data structures for probabilistic representation of a set in order to support membership queries. A Bloom filter is a bit vector, and items are inserted by setting the bits corresponding to a number of independent hash functions. Errors between the actual set and the Bloom filter representation always take the form of false positives to inclusion tests. Furthermore, Bloom filters are used to digest publishing events and event advertisements. An event source broker node determines the subscription belonging to the group by the propagated subscriptions. Brokers can propagate any changes in subscriptions to ODMRP's periodic membership refreshing mechanism.

A. XML-based Typed Event (Message)

The event is defined in XML format with XML schema, and the root element name identifies the event type. The XML schema for the event consists of a set of typed elements. Each element contains a type and a name. Fig. 2 shows example messages. Events themselves can be carried in byte stream or compressed format in ODMRP packets.

```
<?xml version="1.0" encoding="UTF-8"?>
<CD id="001" timestamp="1999-02-27T12:00:00.000-00:00"
        xmlns="http://www.cl.cam.ac.uk/~ey204/lib/">
    <category>jazz</category>
    <composer>Evans</composer>
    <price>18.00</price>
</CD>
----------
<CD id="002" timestamp="1003-02-27T12:00:00.000-00:00"..>
    <category>jazz</category>
    <composer>Davis</composer>
    <price>10.00</price>
</CD>
----------
<CD id="003" timestamp="1055-02-27T12:00:00.000-00:00"..>
    <category>classic</category>
    <composer>Bach</composer>
    <price>22.00</price>
</CD>
```

Fig. 2: Event Instance for Type CD

B. Subscription Language

Most event systems support a subscription language that allows a subscriber to express its information need. 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 [1] is used as a filter-specification language (see examples in Fig. 3). Some XPath expressions will be transformed to simplified and unified formats, and complex expressions are limited.

```
1. /CD[category="jazz"][composer="Evans"]
2. /CD[category="jazz"][composer="Davis"]
3. /CD[category="pop"]
4. /CD[category="classic"][composer="Bach"]
5. /CD[category="classic" and price<"20"]
```

Fig. 3: Subscription Filter for Type CD

C. Summary Structure in Bloom Filters

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 [17], extended to take advantage of XML typed events by XML schema and XPath subscription language. A subscription summary from XPath consists of four data structures described below, and a broker's subscription summary is an array of these data to keep the summarized per-broker subscription information. As an option, a broker aggregates subscription information, when it serves many subscribers or the broker acts on the second level of source node. For more detail, see [19].

- **Event Type Name (ETN):** contains the hashed value of the root element name in the XML schema that identifies the event type.
- **Element Association List (EAL):** stores information about elements and attributes in XML schema and actual values that belong together in a subscription. An EAL consists of an array of bits with a constant number of columns and a variable number of rows for subscriptions. Columns represent the ordered set of supported elements and attributes defined in XML schema, and the rows represent the unique sets of subscriptions. Redundant EALs are eliminated.
- **Arithmetic Element Constraint (AEC):** holds the constraints of each arithmetic attribute of a subscription. The first array consists of two columns and a variable number of rows. Each row represents non-overlapping sub-ranges of values specified in subscriptions for the specific attribute. The second array is used when an arithmetic constraint has an equality operator for a value that is not in the existing sub-ranges.
- **String Element Constraint (SEC):** contains information about the constraints in string type. For each different string type element/attribute, which appears in at least one subscription, a broker implements a SEC structure using three bit vectors.

D. Subscription Aggregation over Networks

The broker nodes that keep the subscriptions in Bloom filters have the correct bits set for the subscription to be recognized as receiver nodes over the network. The event source broker node defines the multicast group with the propagated subscriptions. Currently the least constrained subscription is used for multicast group setting. If the group covers a coarse-grain subscription, more noise would be delivered to the broker, while, if it covers a fine-grain subscription, many groups would be created. However, if a node has high mobility, setting a fine-grain subscription may prevent the overhead of group membership maintenance. Because of the dynamic MANET environment, a deterministic setting may not produce better performance than a probabilistic approach. It is challenging to define the balance between multicast groups and content-based subscriptions, and the current approach will be refined by evaluating the experiment results. A couple of approaches to solve this problem are a part of an ongoing project: (i) aggregate subscriptions to hierarchical patterns including the encoding mechanism; (ii) split/combine subscriptions on the group using content graphs.

ODMRP maintains the group membership with periodic flooding, and subscriptions are also up-to-date using this mechanism. If there is any change in subscriptions, brokers maintain the group setting, as well as splitting and combining the groups.

E. Routing with Extended ODMRP

A broker can reside in an independent node or in the same node where publishers/subscribers reside. The routing between the broker and the client can be achieved by ODMRP with either multicast or 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 message by broadcast, followed by filtering performed at the broker. The second is match-first that requires a precomputed destination list, which is broadcasted to all brokers, followed by routing using the routing list. 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. The third possibility is a distributed-approach by the brokers, where the brokers examine the message content and forward messages using the routing table. MANET environment's dynamic network condition may cause frequent reconfiguration of routing table. Therefore, our proposal is an integrated approach of the above third way with extending ODMRP. The protocol of extended ODMRP is described below (see Fig. 4).

Join Request (JR) Operation:

- A sender node wishing to send multicast packets periodically broadcasts a Join Request (JR) over the network.
- The summary of the event to be sent in the Bloom filter expression is attached to the JR packet.
- A node receiving a non-duplicate JR stores the upstream node ID and rebroadcasts the packet.
- Optionally, a Join Request 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.

JoinTable (JT) Operation:

- A multicast receiver, getting the JR, creates or updates the source entry in its member table. During this operation, the

Bloom filter in JR is also compared with the local summary of subscription Bloom filters.
- JoinTable (JT) are broadcasted periodically. The subscription information is attached to the JT packet.
- The source node updates subscription information for the group according to the received subscriptions in the JT packet.

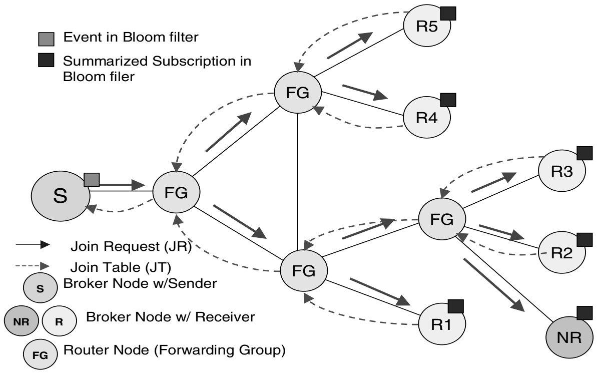

Fig. 4: Extended ODMRP Operation

Build Mesh: An intermediate node (router node), receiving the JT, compares its ID with the entries of the forwarding group table. If there is a match, it is a member of the forwarding group. It sets the FG_Flag and broadcasts its JT. This process creates a mesh between all forwarding group members.

Data Forwarding: After the group establishment and route construction process, a multicast source can transmit packets to receivers via selected routes and forwarding groups. When receiving a multicast data packet, a node forwards it ONLY IF it is not a duplicate and the setting of the FG_Flag for the multicast group has not expired. This whole operation minimizes the traffic overhead and prevents sending packets through stale routes.

Route and Subscription Maintenance: No explicit control message is required to join or leave the group. When the source node leaves the group, it stops sending a *join request* packet, and the receiver node avoids sending back a *join table* packet to leave the group. Forwarding group nodes demote to non-forwarding group nodes if no *join request* is received within the time out period. For maintaining the group ODMRP requires periodic flooding of *join request* to refresh routes and group memberships. Receivers broadcast *join table* as a reply to the *join request* or subscription change on the node. This flooding of packets often causes the broadcasting storm problem. Optimizing the refresh interval for ODMRP is critical for performance and reliability. The ODMRP scheme can utilize 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.

F. Prototype Development

A Java-based prototype development of the proposed content-based publish/subscribe system is in progress. The extended ODMRP has been implemented in Java as application level multicast based [18], because a MANET routing multicast protocol is not yet integrated to the network layer in most commercially available mobile devices. A network simulator JNS (Java Network Simulator) [9] is a Java implementation of the ns-2 simulator and extended to support the extended ODMRP, and mobility simulation is added. The current JNS can handle IP packets and if a packet destination address is within a range of multicast addresses, the packet will be copied and sent on all interfaces the node has. JNS allows dynamic scheduling of events, and the dynamic scheduler that interfaces Java Remote Method Invocation (RMI) to a virtual multicast socket that can behave like a normal Java unicast/multicast socket. The location based mobility prediction is not a part of the current ODMRP implementation. Fig. 5 shows a simulation of the extended ODMRP on JNS that is viewed in Javis (Java Visualizer).

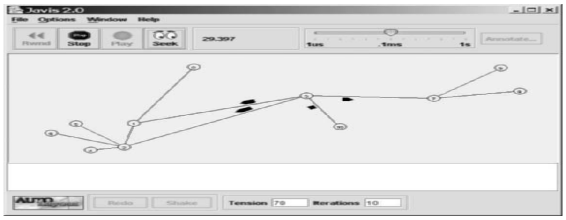

Fig. 5: Simulation of Extended ODMRP

IV. Preliminary Experiments

The experiments are proceeded using the JNS with the dynamic scheduling function. The experiment is in a preliminary stage, and a single dedicated PC is used for the experiments. A real world experiment with resource constrained mobile devices in 802.11b ad hoc network is planned. The network topology used for the experiments in section IV.A is the one shown in Fig. 4. The topology forms similar to a tree. The subscriptions on nodes R1-5 are defined with the values in Fig. 3 correspondingly. The route refresh interval is set to 5 seconds, and the forward group timeout interval is 25 seconds. These values may cause an impact to the experiment environment. Therefore at this stage of experiments, a less influential value was chosen. A source broker sends three different messages shown in Fig. 2, and repeats until the total number of messages reaches 100. Thus, there are 3 matching subscribing brokers among 5 receiver broker nodes. The messages are sent with an interval of 500 milli seconds unless stated. Because this experiment is executed with real network functions, the overhead of any substantial matter may cause an impact to the experiment results. However, the experiment result should still show the essential characteristics of the system. It would be ideal to compare with the typical messaging system such as SIENA over the same environment, but no comparable messaging system is available thus far.

A. Extended ODMRP and ODMRP

Using the original ODMRP, there are two ways to implement messaging systems:

- ODMRP (1 channel) uses a well known channel, and every subscribing broker node executes subscription matching process.
- ODMRP (N channels) predefines channels for each subscription, thus in this experiment, five channels are defined. This approach is not a content-based subscription.

The messaging system using the extended ODMRP requires the conversion process from the message to XPath expression. Ultimately both subscriptions and messages in XPath expression should be transformed to Bloom filters. We measured the conversion overhead depending on the complexity of the XML schema. This was done separated from the simulation experiments and the result indicates the conversion overhead stays as a constant value within the range of 200 nesting elements in XML schema. The comparison process between the message notification and the aggregated subscriptions is a fraction since it is comparison of simple bit sets. Fig. 6 shows the throughput of messaging systems with three different event dissemination mechanisms. The throughput is measured when all messages are delivered to the target subscribing brokers. The extended ODMRP improves performance when the message interval gets beyond the conversion overhead. ODMRP with one channel suffers with the overhead by subscription matching process.

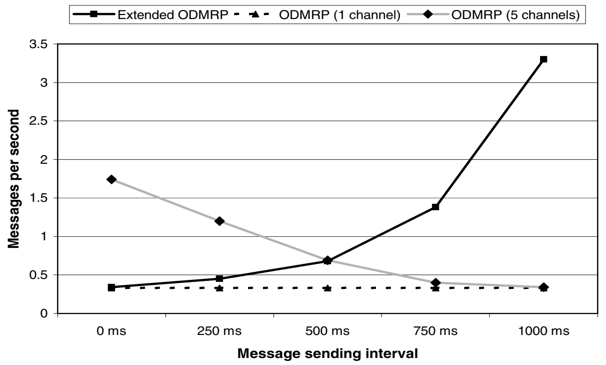

Fig. 6: Throughput Comparison

Fig. 7 shows the network traffic over three messaging systems: the number of bytes of total incoming packets. The broker 1, 2 and 4 have the matching subscriptions. Because the topology is simple, the unmatching brokers receive high volume of packets from flooding of control packets. The extended ODMRP shows comparable throughput to the predefined channel based multicast. ODMRP with one channel causes twice more network traffic than the others.

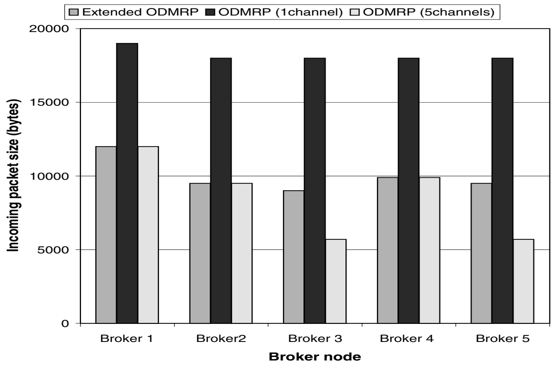

Fig. 7: Network Traffic Comparison

Fig. 8 shows the throughput over the different number of subscribers. The throughput stays the same in spite of increase in network traffic. This experiment may show different results with actual resource constrained mobile devices.

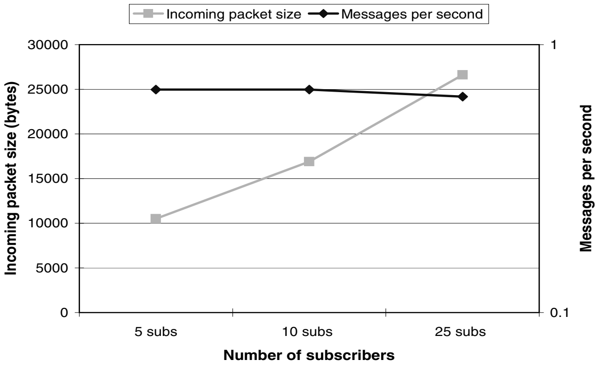

Fig. 8: Number of Subscribers over Extended ODMRP

B. Mobility and Reliability

Fig. 9A shows a topology used in the experiment of mobility impact. A receiver broker moves around, and link1 and link2 get up/down alternatively. The receiver broker is always connected to one of FG (forwarding group) nodes.

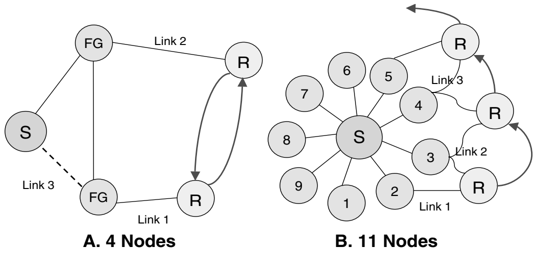

Fig. 9: Network Topology for Mobility Experiments

The message delivery ratio over changing the mobility speed is shown in Fig. 10. When the mobility speed is high, the loss of messages is unavoidable. However, if two FG nodes are apart 5 meters, moving speed with 500 milli seconds indicates that the broker node moves 9 km per hour, which is twice as fast as human walking speed.

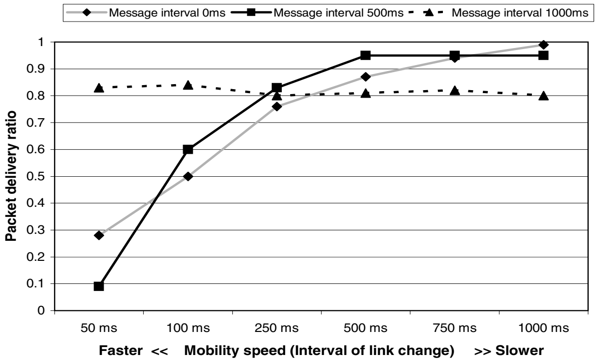

Fig. 10: Message Delivery Ratio

The second experiment focuses on reliability by the mesh topology. Fig. 9A is the topology used, and link3 between the source and the FG node is constantly broken and repaired in this experiment. The subscribing broker node always ties with FG node with link1. The shortest path link3 is supposed to be used for the packet delivery, however when link1 is

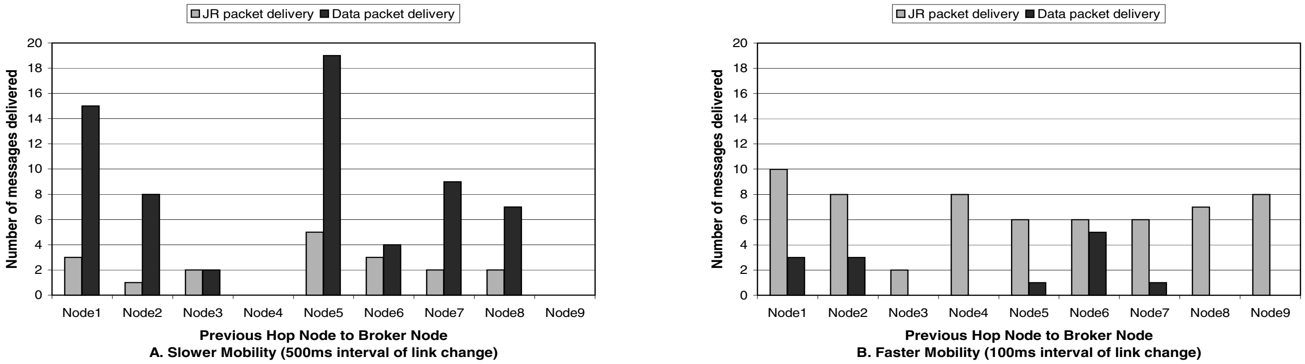

Fig. 11: Message Delivery by Join Request Packet and Data Packet

not available, alternative route is used. Fig. 12 shows that the mesh topology gives 100% message delivery rate and mesh usage. In this experiment, high mobility speed causes unavailability of link1 more often, and the alternative route is used on that occasion. Throughput, incoming packet size, and incoming packet number show constant values in this experiment.

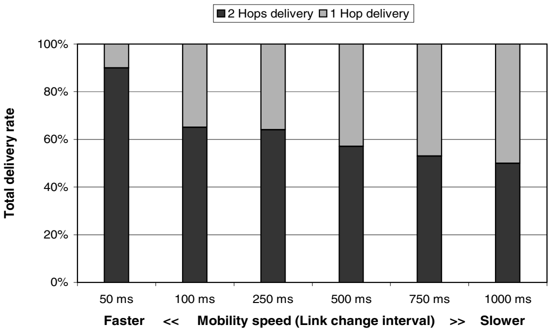

Fig. 12: Reliability by Mesh

The third experiment uses the topology described in Fig. 9B. The receiver broker node moves around from node 1 to 9 repeatedly, and the link with those nodes changes accordingly. Initially link1 and link2 are active, then down link1 and up link3 occur concurrently. The next step is down link2 and up link4, and so on. Fig. 11 shows that when the subscribing broker nodes moves faster, messages are carried by the Join Request packets rather than Data packets, indicating that the established route cannot be used because of node mobility. Packet delivery ratio in Fig. 11A is 0.82, and is 0.93 in Fig. 11B.

V. Conclusions and Ongoing Work

The proposed system optimizes the usage of network bandwidth by subscription aggregation, which reduces the size of routing tables, and it fully supports dynamic subscription registration and cancellations. The key contribution is to create an interface between middleware tier and network components to adapt different contexts for the creation of an efficient event dissemination structure. We will continue our work in the following aspects: (i) building hierarchical subscription summaries with Bloom filters including encoding of hierarchy implication; (ii) establishing an efficient event matching algorithm using approximate and fuzzy approach in order to represent content-based subscription in graphs or sets; (iii) defining subtyping mechanism of XML schema.

Evaluation requires many different aspects such as overhead of extended part of ODMRP, complexity of subscription, efficiency of global subscription aggregation, reliability of different mobility patterns, network traffic load, false positive rate by Bloom filters etc. We will study the experimental results and combine the above research work to improve the publish/subscribe system in unstably connected network environments.

References

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

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

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

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

[5] Corson, S. et al. Mobile ad hoc networking(MANET): Routing protocol performance issues and evaluation considerations. In *RFC 2501*, 1999.

[6] Cugola, G. et al. Exploiting an event-based infrastructure to develop complex distributed systems. In *Int. Conf. on Software Engineering*, 1998.

[7] Eugster, P.Th. et al. The Many Faces of Publish/Subscribe. *Technical Report TR-DSC-2001-04, EPFL*, 2001.

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

[9] Java Network Simulator (JNS). In *http://http://jns.sourceforge.net*.

[10] Kyper, G. et al. Subsumption for XML Types. In *Proceeding of ICDT*, 2001.

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

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

[13] The Narada Event Brokering System . *http://grids.ucs.narada.edu/ptliupages/projects/narada*.

[14] Ratnasamy, S. et al. Application-Level Multicast using Content-Addressable Networks. In *Proc. of NGC*, 2001.

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

[16] Snoeren, A. et al. Mesh Based Content Routing using XML. *Proc. of SOSP*, 2001.

[17] Triantafillou, Y. et al. Subscription Summaries for Scalability and Efficiency in Publish/Subscribe Systems. In *Proc. of DEBS*, 2002.

[18] Vollset, E. JMS on Mobile Ad-hoc Networks. In *PWC*, 2003.

[19] Yoneki, E. et al. Content-Based Routing with On-Demand Multicast. to appear in *24th ICDCS Workshop on WWAN*, 2004.

[20] Zhuang, S. et al. Bayeux: An architecture for scalable and fault-tolerant wide-area data dissemination. In *Proc. of NOSSDAV*, 2001.