

EFFICIENT PUBLISH/SUBSCRIBE SYSTEM OVER

MOBILE AD-HOC NETWORK

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
Of Agriculture and Applied Science

By

Chao Liu

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

October 2011

Fargo, North Dakota

North Dakota State University
Graduate School

Title

EFFICIENT PUBLISH/SUBSCRIBE SYSTEM

OVER MOBILE AD-HOC NETWORK

By

Chao Liu

The Supervisory Committee certifies that this *disquisition* complies with North Dakota State University's regulations and meets the accepted standards for the degree of

Master of Science

SUPERVISORY COMMITTEE:

Dr. Juan Li

Chair

Dr. Jun Kong

Dr. Changhui Yan

Dr. Nan Yu

Approved by Department Chair:

03/05/2012

Date

Kenneth Magel

Signature

ABSTRACT

Information dissemination is an important issue for mobile ad-hoc communities. This issue is very challenging due to the dynamic and fragile nature of the mobile ad-hoc networks, in which participants have limited computing resources and battery, intermittent network connections, and mobile tasks. To address the aforementioned issue, this thesis proposes an efficient semantics-based publish/subscribe strategy. In our proposed publish/subscribe system, distributed mobile participants are organized into clusters based on their location proximity. A compact semantics-based indexing scheme is provided to guide information flow. Intra- and inter- cluster routings are proposed to assist efficient propagation of event notifications. A comprehensive set of simulation experiments prove the effectiveness of the proposed scheme.

Keywords: Publish/subscribe system, semantics, mobile wireless computing, routing.

ACKNOWLEDGMENTS

This thesis would have not been possible without the support of many people. I would like to express my sincere gratitude to my advisor Dr. Juan Li, who was abundantly helpful and offered invaluable assistance, support and guidance.

Deepest gratitude is also due to the members of the graduate committee: Dr. Jun Kong, Dr. Changhui Yan, and Dr. Nan Yu, this study would not have been successful without their encouragement, knowledge and assistance.

Lastly, and most importantly, I wish to thank my parents, Dazhong Liu and Fengru Cao. They bore me, raised me, supported me, taught me, and loved me. To them I dedicate this thesis.

TABLE OF CONTENTS

ABSTRACT.....	iii
ACKNOWLEDGMENTS	iv
LIST OF TABLES.....	vii
LIST OF FIGURES	viii
CHAPTER 1. INTRODUCTION.....	1
CHAPTER 2. RELATED WORK.....	5
CHAPTER 3. EVENT DELIVERY SCHEME.....	12
3.1 Clustering and Tree Construction	12
3.2 Publishing and Subscribing.....	16
3.3 Pre-match Process	19
3.4 Inter-group Communication.....	20
3.5 Dealing with Mobility.....	21
3.6 Support of Multi-Attribute Publishing/Subscribing.....	21
3.7 False Positive.....	22
CHAPTER 4. EXPERIMENT	24
4.1 Setup.....	24
4.2 Result.....	26
CHAPTER 5. CONCLUSIONS AND FUTURE WORK.....	32

REFERENCES 33

LIST OF TABLES

<u>Table</u>	<u>Page</u>
1. Process Group Joining Request Algorithm.....	15
2. Parameters Used In the Simulations.....	25

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1. Intra-group Subscribing and Publishing	18
2. Successful Event Delivery Rate vs. Nodes' Moving Speed	27
3. Message Overhead vs. Nodes' Moving Speed	28
4. The Comparison of Message Overhead.....	28
5. Average Memory Overhead per Node vs. the Network Size.....	29
6. The Average Notification Latency vs. The Network Size.....	30

CHAPTER 1. INTRODUCTION

Nowadays, personal mobile devices, such as cell phones, iPads, and laptops are equipped with wireless network interfaces, which enable them to form Mobile Ad-hoc Network (MANET). In such network, a large number of personal mobile devices are connected in a self-organized fashion. These portable and handy devices enable the immediate establishment of communication networks for spontaneous tasks such as events delivery, advertisement broadcasting, emergency/rescue operations, disaster relief, and military networking.

An important issue in MANET is to enable efficient information dissemination among the mobile ad-hoc participants. However, this problem is difficult to address due to the nature of the mobile wireless network. In such network, participants are portable devices which have limited computing power, storage and battery. In addition, the wireless transmission is prone to loss and failure. Furthermore, because the participating devices are mobile, the network topology changes over time. Although there are lots of approaches proposed to address this issue, as shown in Chapter 2 Related Work, they have some limitations and information dissemination in MANET is still an open problem.

To address the aforementioned problem, in this thesis, I propose a publish/subscribe (pub/sub) model, which enables efficient information dissemination in MANET. In the proposed pub/sub model, every participant can be a publisher, a subscriber or both at the same time for information delivering. Publisher is responsible for producing information in events form while the subscriber expresses subscriptions, which stand for their

interests in some specific events. The publisher indirectly addressed the subscriber according to the content of events. Thanks to this feature, publishers and subscribers can share information but do not need to aware each other's address. It gives us a chance to seamlessly expand and implement pub/sub to dynamic wireless network.

The pub/sub scheme provides greater scalability and supports a more dynamic network topology as required by mobile wireless communities. As an example, in a military battlefield, wireless and mobile sensors such as satellites and equipment sensors report many kinds of information. Whenever events occur, the monitoring devices as publishers can publish the events and have interested subscribers notified. In another example, vehicles on a freeway may subscribe to traffic information. Vehicles approaching subscribers may notify subscribers about their interested traffic information. The scenarios mentioned above require the deployment of a highly scalable and efficient communication infrastructure, obviously publish/subscribe is a good candidate.

The pub/sub communication paradigm has been used in distributed systems widely and presents many benefits. However, it is still a big challenge to deploy pub/sub system to mobile ad-hoc communities due to the aforementioned limitations such as the ability of computing, power, and storage of mobile devices. In addition, the complexity and diversity of resources/services in the MANET require corresponding expressiveness of the description of the resources/services. Content-based pub-sub system is proposed in this thesis since it is a more general and expressive paradigm. Subscribers are free from being limited to select pre-defined topics, while the flexibility of choosing interested content along multiple dimensions. However, it is difficult to deploy scalable

implementations of content-based pub/sub in MANET. Most existing works advocate maintaining a broadcast tree, which helps the registration of subscriptions and delivery of notification [10, 13, 20], to support content-based pub/sub in mobile wireless networks. These systems suffer from tremendous amounts of overhead to maintain the tree structure, especially when the size of the tree is large. The pub/sub system cannot be scalable in such cases.

In this thesis, I propose an efficient event delivery system based on publish/subscribe communication paradigm over mobile ad-hoc communities. The system focus on constructing a publish/subscribe scheme which organizes mobile devices into different clusters according to their geographical proximity to reduce the overhead for normally maintaining a large subscription tree in a mobile network environment. Event delivery is implemented along the Publish/subscribe tree structures, which are constructed over the clusters. New publications are sent along publish/subscribe tree structure in order to propagate them to their subscribers. A tree construction algorithm generates such a tree, described the process such as how the nodes are interconnected, and how to set up a publish/subscribe tree among random nodes network environment. A publish/subscribe tree can be seen similarly as a multicast tree in traditional multicast networks in function. Numerous multicast protocols already exists, which often try to produce an “optimal” multicast tree. These schemes seek to optimize one of a few possible metrics. For example, in order to minimize the total number of nodes participating in multicast, some protocols use Steiner trees.

The rest of this thesis is organized as follows: Chapter 2 introduces the related work. Chapter 3 presents the proposed pub/sub protocol and related issues. Chapter 4 discusses the experimental results. Conclusions are drawn in Chapter 5.

CHAPTER 2. RELATED WORK

The publish/subscribe communication paradigm has been widely used for anonymous and asynchronous group communications not only in centrally but also distributed manner. Most of the previous works which research pub/sub [2, 7, 9] focus on improving ability over usual wired networks. Centralized schemes [15, 12, 6] make the efficient optimizations in the process of matching since they have a global view of the system. In [6], they deploy an unicast message to implement a state-transfer protocol. The state-transfer protocol may result in extreme load increases on a network since events are distributed in a multicast system to static users. Multicast is the delivery of information to a group of destinations in a single transmission from the source. There are two popular network multicast approaches: shortest path multicast tree and core-based tree [25]. In the former approach, each source needs to build a tree, and the shortest path multicast tree guarantees the shortest path to each destination. However, the overhead tends to be intolerant due to the trees number usually extends extremely in the network. On the other hand, the core-based tree scheme constructs only one tree for each communication group so the number of trees and overhead are greatly reduced.

The system with central server can be a potential restriction since lacks scalability and fault tolerance. Distributed pub/sub systems [8, 18] have been introduced to address these problems. For example, P2P-based solutions [20, 3] implement pub/sub in a fully decentralized fashion. A peer-to-peer structured overlay infrastructure is a nodes set constructed self-ruling application level network. These nodes form a structured view

over a virtual key space in which each key of the virtual space is mapped to a node [26]. Efficient discovery of data items is processed in the structure, which makes the efficient unicast or multicast communication facility among the nodes real. There is a correspondence between any address and active mobile node in the system which is ensured by the structured overlay. A structured overlay does not allow to better process dynamic feature of the systems such as node joining, multicast tree maintenance, it is not suited in decentralized network environments, and not a viable solution for setting up human administration interventions in a dynamic decentralized network.

Some other P2P-based pub/sub approaches rely on a broker overlay to transmit messages. In [3], they introduce a novel architecture for implementing content-based pub/sub communications on top of structured overlay networks. This architecture is well-known because it breaks some limitations in the existing infrastructures such as lacking of self-configuration and the ability of adapt to dynamic changes. They create a trade-off between the abundant event subscription semantics of content based pub/sub systems and the standard logical addressing scheme of overlays. In [20], the authors establish a peer-to-peer network combined with a publish/subscribe system routing scheme, which contains graph topology and management. They developed their network by holding the use of fully general filters, at the same time try to keep a logarithmic bound on delivery depth with evenly distributed huddle, even in the side of dynamic participation and failures. These approaches assume universal connectivity between any two nodes in the system, and at a roughly stable cost. Although this assumption is reasonable in fixed

networks, it is not a realistic one in a mobile wireless environment because it hides the fact that an unicast is actually implemented with multi-hop broadcast.

Existing pub/sub paradigms for mobile wireless networks also can be grouped into three categories: (a) flooding, (b) gossiping, and (c) selective routing. Generally speaking, a complete dissemination of event or subscriptions to the entire system is the base of flooding algorithms. Selective algorithms focus on reducing the propagation produced in the routing structure. Event gossiping are random algorithms with no routing structure, which are suitable for highly dynamic contexts such as mobile ad-hoc network.

Flooding is the most naive way to propagate information. Because flooding own broadcast nature of wireless communication, so it is a naturally selection in a mobile wireless [27]. The flooding solution works as propagating or broadcasting each event from the publisher to all the nodes in the mobile network. Each subscription is sent to all the nodes thus each node has the complete knowledge base of the whole system, and the non-interesting events can be filtered out quickly. The drawback is obvious: the message overhead is too heavy to be scaled. It creates too much network traffic, and it cannot be scalable for a large-scale mobile wireless community [14].

A simulations study in [28] report that flooding algorithm in pub/sub system does not work well especially in a high rate moving environment because each node has to send all the changes to all other nodes. Two basic researches have been proposed to reduce the redundant information: (1) imposing a subset routing overlay structure. For instance, allow only nodes which are part of a multicast tree rebroadcast the message but

not the others [29]. (2) Dropping messages through related protocol. The thesis [30] proposed conclusions that flooding can be utilized when a large number of subscribers are interesting in most events, because it avoids the overhead produced by the storage and information updating.

In gossip-based protocols [11], each node randomly contacts one or a few nodes in each communication round and exchanges subscription information with these nodes. The protocol is simple because it does not require maintaining any data structure for the event routing during the movements of nodes. The drawback of this approach is the heavy redundancy and large message overhead [19].

Selective-based routing is also called rendezvous-based or filter-based routing. This category of pub/sub systems try to reduce the propagation overhead by constructing routing structures.

The selective event routing algorithms focus on dividing the whole network into subsets in which the nodes store each subscription. Selective routing algorithms can store network resources and reduce the overhead. In particular, [32] [33] [34] [35] discuss selective algorithms for establishing and maintaining routing trees over a mobile ad-hoc network. In [6], brokers (rendezvous) are responsible for maintaining the subscribe information from mobile clients by connecting each other. The events will be delivered to a different broker when the moving clients change it. In [31], Huang and Molina present a publish/subscribe system and modify it to be more suitable for mobile network. In the system, a multicast tree maintains all brokers. They developed the publish/subscribe

system to be operated in a mobile environment, where subscribers can be produced and send subscriptions for delivery events. After that, they did a research in [33], establish an optimal pub/sub routing tree based on the ad-hoc network. In this tree, each node's subscription is propagated in its precursor's record to avoid flood. However, the scalability is restricted especially when the number of nodes is growing. In [34], Huang and Molina proposed a high adaptive fully decentralized and modular scheme for the mobile network, which only asks for local information in order to maintain the anonymity of the distributed system. In [37], they describe an algorithm relocates the event routing tables if there is a disconnection in a generic acyclic graph topology within a mobile ad-hoc network. In [50] [51], Anceaume et al. and Cugola et al. present two routing algorithms for building and maintaining a tree structure over a mobile ad-hoc network on the top of a transport protocol. In particular, Cugola et al. describe an algorithm for relocating the event routing tables after finding a failed connection in a generic acyclic graph topology within a mobile ad-hoc network. The paper contributes on a separation of concerns between the connectivity layer and the event dispatching layer. There is an assumption in the algorithm for repairing the event routing data structure: the tree is kept connecting by some loop-free algorithm at the routing level.

Selective-based routing can be further divided into address-based routing schemes [1, 10] and content-based routing schemes [22, 13].

In the address-based routing schemes, subscriber registers its interest and its address/ID to every node in the network using a broadcast scheme. Each publisher can know which subscribers are interested in its events and send the published events to the

selected subscribers using traditional routing algorithms such as DSDV [16] or AODV [17].

In the content-based routing algorithm, subscribers, firstly, specify the notification information which they want to be delivered in order to express their interests. Subscribers are notified based on their interests rather than their address/ID. A filter in a subscription is a query formed by a set of constraints over the values of attributes of the notification composed through disjunction or conjunction operators [26]. The attribute type and the subscription language are the possible constraints. Properties of the events themselves are the key features to distinguish events from each other but not some predefined criterion. Therefore, publishers do not need to address the routing issues when notifying the subscribers. This may save a lot of network traffic. Most subscription languages contain equality and comparison operators such as regular expressions [38] [39] [40]. The complexity of matching operation is influenced by the complexity of the subscription language. It is unusual to have subscription languages allowing queries more complex than that in a conjunctive form. The examples are [41] [42]. A well-defined specification of content-based subscription models can be found in [43]. Examples of content-based systems are Gryphon [44], SIENA [45], JEDI [46], LeSubscribe [47], Ready [48], Hermes [49], Elvin [50]

The content-based pub/sub expressive power is developing, the resource consumption needed to calculate for each event is getting higher [49] [36] [51]. The content-based schemes have to build dissemination trees to deliver publications to subscribers who are interested in the publications. Flooding is avoided with the tree

structure because each node's subscription is kept in its neighboring nodes. However, the scalability is restricted especially when the tree is large.

CHAPTER 3. EVENT DELIVERY SCHEME

This exploration's purpose is to find an efficient information dissemination method for delivering event notifications to all interested parties in a large-scale mobile ad-hoc community in a timely fashion. A network-wide tree needs to be maintained in most existing content-based notification systems for mobile wireless networks. While the network grows large and becomes highly dynamic especially in a large-scale mobile ad-hoc community, the tree structure usually endures the large overhead for maintaining and updating the structure. The system separates a large wireless community into smaller groups, to avoid too much traffic cost and enhance availability by providing service locally.

The scheme organizes self-established groups into a tree structure in which each node only exchanges information with its parents. A root node is the one stands on the top of the tree in each tree group. Gateway nodes are that in the border of two or more groups, which manage communication between adjacent groups. A compact filter-based indexing scheme is deployed on the upper layer of each tree. This thesis also proposed efficient protocols to deal with content-based routing within a group (inside-group) and between different groups (outside-group).

3.1 Clustering and Tree Construction

Groups are constructed to partition the network into separate smaller manageable parties which leads to efficient communication and low processing in the network. By

applying the grouping algorithm, a special type of node named group leader is elected. It has an overview of the subscriptions within the group. Furthermore it communicates with other groups through gateway nodes which connect adjacent groups for cooperative notification.

Initially, there is not any group formed in the network. Each node is roaming in the system while trying to find a group to join in. Every node just queries its neighbor nodes in on-hop distance for possible group information. If there is not any available group to join, the querying node waits a random period and then announces itself as a group leader only if it is capable of being a group leader (enough power, bandwidth, and signal strength). The other nodes can join its group after receiving this announcement.

After receiving a joining request, the node check the querying node's Group ID for preventing loop connection. If the Group IDs are different then it allows the querying node to continue. The querying node waits for a random period, a "pre match" process will be activated if there are possibly other replies arrive at this node during this period. A "pre match" process is a work which helps a node join the groups that have higher similarity. The details about the "pre match" process will be discussed in the following section.

After joining a group, each node will be notified by its parent node with its group ID, the leader ID, and a level number which represents its distance to the root in terms of the number of hops. Obviously, the root's level number is 0. The level number is used to control the maximum level of the group tree.

Algorithm 1 illustrates how a node C processes the “group joining” message, when C receives such a message from node N. If C is already a maximum level node, it will ignore any “group joining” message. If not, C will check N’s Group ID firstly then add N to its group by setting up a parent-child link. C will notify N about the group ID, the root ID, and set N’s level as C’s level plus one. At the same time, C will send N’s level number back to the group leader by attaching a number into the subscription Bloom filter bitmap to refresh the current highest level number. After N completes the joining process, it will keep querying to join other groups. By doing this, there will be such a case exists: when C receives a “group joining” message from node N and N already belongs to a group. In this case, N may become a gateway node connecting its original group with the group C belongs to. In order to avoid creating too many small groups, after a gateway node appears, the system always tries to merge adjacent groups through gateway node firstly. The group which has a smaller level number will be merged to which has a bigger one. After a node becomes a gateway node, it will piggyback a gateway flag, its group ID and its level number with the subscription Bloom filter bitmap to notify the group leader its gateway status is activated. If the group leader finds that there are two or more gateway nodes which connect the same group, the one has a lower level number will win the gateway status. Then the group leader will send back the current highest level number to the gateway node. Here I let A denotes the gateway node’s level in the group which has a bigger level number and let b denotes the gateway node’s level in the group which has a smaller level number, and let c denotes the current highest level number in the group which has a smaller level number. If the sum of the A, B and C is less than the

maximal level limit, the two groups will be merged. N's level will be set as its "current" parent's level plus one. The level of N's "current" children will be refreshed to N's level plus one and so on.

In another case, N becomes a gateway node but the adjacent groups cannot be merged together because the sum of A, B and C defined above is bigger than the maximal level limit. So the gateway node is responsible for helping the group leader forward publications and subscriptions to adjacent groups through it.

With this group tree construction and merging algorithm, I do not only cluster nodes into groups but also construct an approximately balanced tree for each group.

TABLE. 1 Process Group Joining Request Algorithm

When node C receives a group joining request	
1:	if C is not a maximum level node
2:	&& C and N belongs different groups
3:	N joins C's group.
4:	else
5:	ignore the request
6:	
7:	pick group G which C belongs to and C.level<maxLevel
8:	
9:	if N does not belong to any group
10:	
11:	add N to group G, set parent=C, set level= C.level+1
12:	else
13:	if there exists a group Gi which is one of N's group and N.Gi.level+N.G.level+small<G, Gi >.curHL ≤MaxLevel
14:	Merge Gi with G through
15:	else
16:	N become two gateways connecting their respective groups.
17:	endif
18:	endif
19:	endif

3.2 Publishing and Subscribing

Nodes can subscribe their interests along the tree branch structure, and then publications can be efficiently propagated through the tree path to the interesting nodes which already specify their own interesting events in a type of compact filter-based indexing scheme. It is advisable to aggregate the set of subscriptions into a compact set of content specifications by given the limited nature of the mobile wireless environments. I implement the compact indexing with Bloom filters [5].

Bloom filter was first developed in 1970 by Burton H. Bloom long before Oracle existed. A Bloom filter is a data structure used to support membership queries and represents a set to support membership queries. It is a compact randomized data structure and used to test whether an element is a member of a given set or not.

A vector of m bits is the buildup of a Bloom filter. Initially all set to 0. Then k independent hash functions, h_1, h_2, \dots, h_k , each with the range $\{1, \dots, m\}$, i.e., each returning a value between 1 and m , are applied to a set of given elements (in our case, the elements are the publications and subscriptions). Every hash function from h_1 to h_k needs to be applied to each element in set. According to the return value r ($r_1, r_2, r_3, r_4 \dots$) of each hash function, the bit locates at position r_i is set to 1. A particular bit may be set to 1 multiple times. Since there are K hash functions, up to K bits in the bit vector are set to 1 for each time. To determine if an element is in the vector, we check the bits at positions $h_1(b), h_2(b), \dots, h_k(b)$. If any of them is 0, then b is certainly not in the vector. Otherwise we conjecture that b is in the set, although there is a certain probability that we are wrong

which is called a “false positive”. For an acceptable probability of false positive, the parameter k and m should be chosen carefully. The salient feature of Bloom filters is a clear tradeoff between m and the probability of a false positive. Observe shows that after inserting n keys into a table of size m , the probability, which the particular bit is still 0, is exactly $\left(1 - \frac{1}{m}\right)^{kn}$. Hence the probability of a false positive in this situation is $\left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k \approx (1 - e^{-kn/m})^k$. The right hand side is minimized for $k = \ln 2 \times m/n$, in which case it becomes $\left(\frac{1}{2}\right)^k = (0.6185)^{m/n}$

In our tree structure, every non-leaf node maintains a filter including several Bloom filter bitmaps: one bitmap comes from local subscriptions, and the rest of others come from subscriptions of its children branches. Each node sends the merged bitmap to its parent. So every internal node has a summarized view of a sub-tree rooted itself. The root has a summarized view of the entire tree. Fig. 1 illustrates the index aggregating and publication forwarding process. In this example, the Bloom bitmap size is 12 bits, and two hash functions ($H1, H2$) have been used to map a subscription. In reality, the size of the bitmap is much larger, and the number of hash functions is more. In the example, node B’s routing table includes a local bitmap, and two child- (D and E) bitmaps. A local subscription SB is mapped to two positions: 2 and 3 in the bitmap ($H1(SB)=2, H2(SB)=3$). So in B’s local bitmap, $B_2=1, B_3=1$. B merges these three bitmaps by bitwise OR, and sends the merged bitmap to its parent A. The merged bitmap represents all subscriptions from B and its descendants.

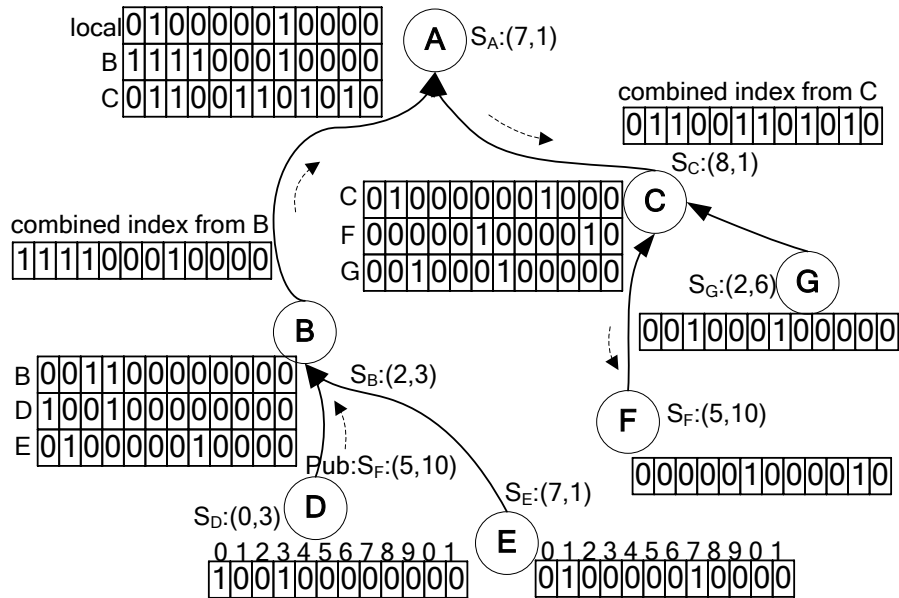


Figure 1 Intra-group Subscribing and Publishing

Every node in the network will become a publisher when it needs to publish events to the other members in its tree. When a node receives a publication notification, it checks its Bloom filter. If it finds match in its local bitmap, it notifies the local subscriber. If it finds match from a child's bitmap, it forwards the publication to that child. If neither of them matches the publication, the publication will be sent to its parent. The parent will perform the same procedure. This routing scheme forwards publication only to nodes lying on branches, which potentially can match the publications and avoids sending the publications to other nodes.

Now suppose D receives an event publication SF. It first uses the two hash functions H_1 and H_2 hashing S_F to 2 bits: 5 and 10 in the bitmap. Because D cannot find match locally, it forwards the query to its parent B. B also hashes S_F and tries to find

match in its bitmap, no matter matching or not, B will keep forwarding the publication to its parent, every node follows this rule until the publication arrives at the root. For example, B cannot find match in its routing table, it just forwards the query to its parent A. A finds match in child C's bitmap (because $C_5=C_{10}=1$), then A forwards the query to C. Similarly C finds match from child F, so the query is then forwarded to F. Finally F finds match in its local bitmap, and it will check its subscription database to further verify the subscription.

3.3 Pre-match Process

Initially, there is not any group formed in the network. Each node in the system tries to find a group to join in. Every node just queries its one-hop neighbor nodes for possible group information. After receiving a joining-reply, the node wait for a random period, if there are other replies come in during this period, the "pre-match" process will be activated. A joining-reply is a message which includes group ID, leader ID, and a level number, and the reply node's Bloom filter bitmap. The node receives several joining-replies will compare that Bloom filter bitmaps with itself and notify the nodes that have more number of the same bits as itself, which means it agrees to set up the parent-child link between them, and then it discards the rest. A "pre-match" process is a method which helps a node join the groups that have more similarity. At the same time, if there is not any bitmaps match, it means the node potentially needs to join in more groups and receive more publications to try to satisfy its subscription.

3.4 Inter-group Communication

As mentioned above, our system can effectively deliver events/services inside a tree group. Furthermore it still needs to propagate publications among groups to deliver publications community wide. Naturally a group leader, which recodes the information of gateway nodes, becomes a representative of the group since it is the root of every group tree, and it has summarized index of the entire group. Every publication will be kept forwarding to the group leader. When the publication passes the gateway node, it propagates the publication to all its parents.

It is mission-critical and time-critical for most mobile wireless communications. In this case, end-to-end QoS in latency is very important. The published events make senses only if they can be delivered quickly to the interested users. Otherwise, the event information is stale and loses value. For considering about the location and time requirement, time-critical services/events usually are delivered only in the local group or a few adjacent groups. These events will be propagated from the publishers with limited hops. By publishing and subscribing of two types of events/services: local events/services or global events/services, this system can effectively satisfy the need for mission-critical and time-critical communication. Global events/ services will be delivered to interested subscribers all over the network, while local events will be only delivered to local subscribers.

3.5 Dealing with Mobility

As we can see the pub/sub communication paradigm is constructed based on the tree group structure. However, the network topology and condition are kept changing due to the natural high mobility of a mobile wireless community. It is very necessary to maintain the tree group structure over the changing topology. In this scheme, a heartbeats message is sent between parents and children nodes in a tree periodically to identify whether they are connected to the tree. A node always tries to find another neighbor as its new parent when it is unable to connect to its parent. A node must avoid connecting to its descendents to avoid loops in the tree. This node will notify its children and let them find new parents using the same strategy if there are no such neighbors exist. When a node N finds it loses its child, N will remove the filter from its children list and notify this update to N 's upstream parents until it is forwarded to the leader. In this way, the system repairs and reconfigures the tree group structure.

3.6 Support of Multi-Attribute Publishing/Subscribing

Our proposed strategy encourages topic-based pub/sub best according to the use of hashing to map services/events to Bloom filter bit maps. By adding a disjunction of multi-attribute subscription to Bloom filters, this system can also support multi-attribute content-based pub/sub. For example, to subscribe a subscription $S=\{a \cup b \cup c\}$, a , b , and c should be added to the Bloom filters one by one. First I give these attribute an order, such as alphabetically, to add a conjunction of multi-attribute subscription to Bloom

filters, and then I add the concatenation of the ordered attributes to the Bloom filter. For example, to subscribe a subscription $S=\{b \cap a \cap c\}$, “ abc ” should be mapped to the Bloom filters.

Each of the attributes will be checked with the Bloom filter to map the disjunction of a multi-attribute publication, as long as one of them passes the filter, the publication will be forwarded to interested subscribers. We should order them and deploy all the combinations of the attributes to map the conjunction of a multi-attribute publication. A publication with n attributes may have 2^n-1 combinations. For example, a publication P with three attributes $\{a \cap b \cap c\}$ can potentially satisfy seven subscriptions: $\{a\}$, $\{b\}$, $\{c\}$, $\{ab\}$, $\{ac\}$, $\{bc\}$, $\{abc\}$. Therefore, system has to map all of these seven combinations with the Bloom filter bitmap. In this way, we can support conjunction and disjunction of multi-attribute publishing and subscribing.

3.7 False Positive

As mentioned before, the false positives may be increased by Bloom filter index and the aggregation of index. Therefore, there is a possibility that the index is only an approximation of the subscription and it may lead publications to nodes or branches which do not contain relevant subscriptions. However, the fidelity of the notification scheme will not be affected, because the node that finally receives the routed query can process further verification which checks its subscription database for insuring. Notifications will be routed along almost optimal paths and most of the nodes that finally

receive queries will, in fact, contain relevant subscriptions as long as the false positive rate is small.

CHAPTER 4. EXPERIMENT

To evaluate our pub/sub event delivery scheme, the system built up a set of simulation experiments. In this section, I present the experimental setup and then analyze the obtained results.

4.1 Setup

Many mobile wireless communications are mission-critical and time-critical. The published events are reasonable only if they can be delivered rapidly and accurately to the interested individuals. The obsolete event information is not necessary since it loses value. Our system can effectively support the need for mission-critical and time-critical communication by publishing and subscribing of local and global events/services.

We considered a local ad-hoc network environment which encloses different nodes was off an area of 2000m \times 2000m. I kept modifying the density of the nodes throughout the simulations. The mobility of the nodes was deployed similar to that of the “random waypoint” model as presented in [4]. Initially, the nodes are randomly distributed within the enclosed area in the random waypoint model, and everyone is assigned a random destination towards which moves at a predetermined speed. The node takes a natural stop for a predefined interval of time once reaches its destination, and then it repeats this movement process. I predefined 150m as the transmission range of a node. In the simulation, subscriptions and publications are represented as randomly-generated strings.

TABLE 2 Parameters Used In the Simulations

parameter	range (default)
network size	500-1000 (500)
environment area	2000m×2000m
node moving speed	0-25m/s (2m/s)
node transmission range	150 m
node pause time	0s-50s (10s)
number of attribute per subscription/publication	1-4
number of subscriptions per node	1-3
number of distinguished subscription interests	50
publication probability per node per second	5%
ratio of local and global pub/sub	1:5~5:1 (1:1)
maximum levels of each group tree	9
preferred levels of each group tree	1-9 (4)
TTL for flooding	10

For performance comparison with our tree-group pub/sub routing scheme (TG), I also simulated flooding-based pub/sub, which is commonly used as a baseline for comparison and traditional subscription tree-based (ST) pub/sub routing [22, 13]. The first one is the simplest pub/sub routing approach while the second ST approach [22, 13] constructs a subscription tree to subscribe interests and forward publications. It is one of the most popular approaches for content-based pub/sub. The same physical topology and the same set of publishers and subscribers were used for testing these three approaches.

The following performance metrics have been employed for the comparison of the algorithms: (a) the fraction of events delivered to interested subscribers, (b) the communication overhead including subscription, publication, notification, and topology maintenance overhead, (c) memory storage overhead needed for storing subscriptions, and (d) notification latency. The various simulation parameters and their default values are listed in Table I.

4.2 Result

Fig. 2 and Fig. 3 compare the event delivery rate and message overhead of the three pub/sub schemes when nodes increase their moving speed. Although flooding deliver more events notifications compared to the other two techniques is an undeniable fact, however, the message overhead of the flooding technique is much higher compared to the other two techniques. Since it is difficult to maintain a large tree connected when the network is very dynamic, the delivery rate of the subscription tree-based (ST) approach is low. In contrast, the proposed tree group (TG) technique, which incurs much fewer message overhead compared to the other two techniques, delivers a comparative number of results.

Fig. 4 shows the compare of the message overhead that includes the publication notification and maintenance overhead to process each of the routing protocols. Our proposed tree group approach has two configurations with different accepted tree level: 3 and 6 (represented by TG3 and TG6 in the figure) in this experiment. We can see that our proposed TG approach contains the smallest overhead. Both the ST and the TG

schemes contain much fewer messages overhead compared to flooding to propagate the same number of events. The figure also shows the composition of the overhead, i.e., the overhead ratio of the notification and maintenance. Notification forwarding causes all of the overheads come from the flooding. For the traditional subscription tree scheme ST, since it needs lots of overheads to maintain a large tree, the maintenance overhead accounts for a higher proportion. I reduce the levels of the subscription trees by clustering nodes into groups, thus maintaining the tree is easier and faster compared with a single subscription tree scheme.

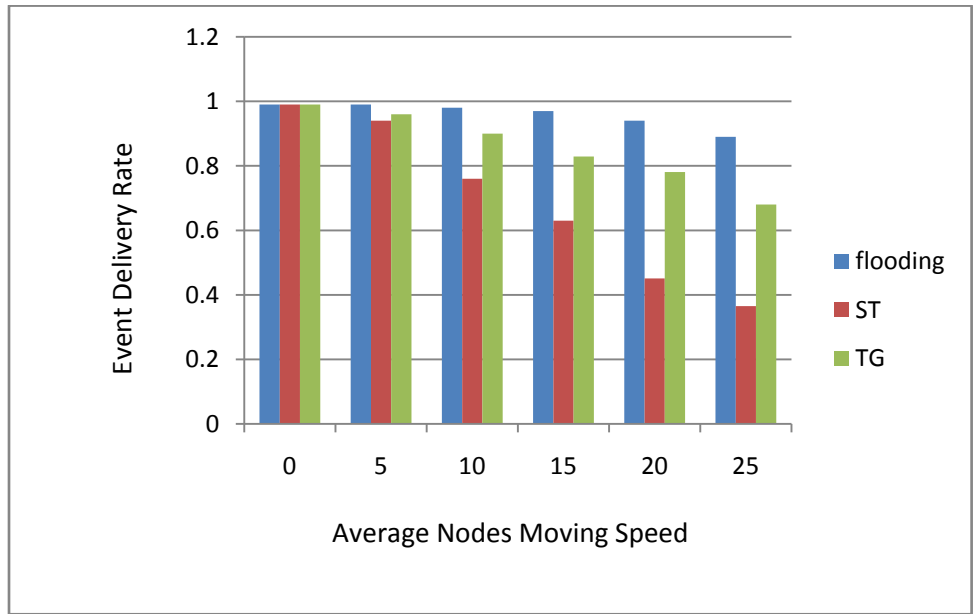


Figure 2 Successful Event Delivery Rate vs. Nodes' Moving Speed

Fig. 5 compares the memory overhead of storing subscriptions for the three pub/sub schemes. The flooding-based approach does not need to maintain nodes' subscriptions because events are flooded to every node in the network, thus the subscription overhead

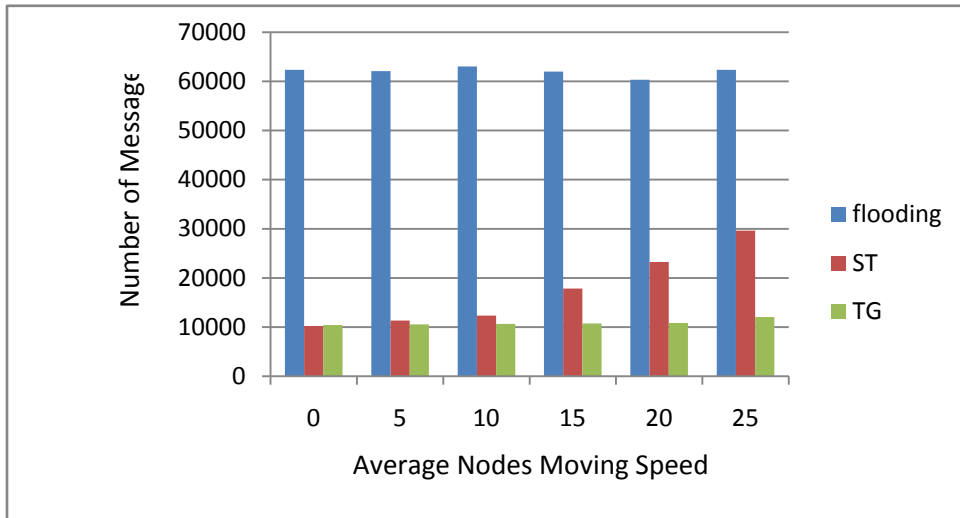


Figure 3 Message Overhead vs. Nodes' Moving Speed

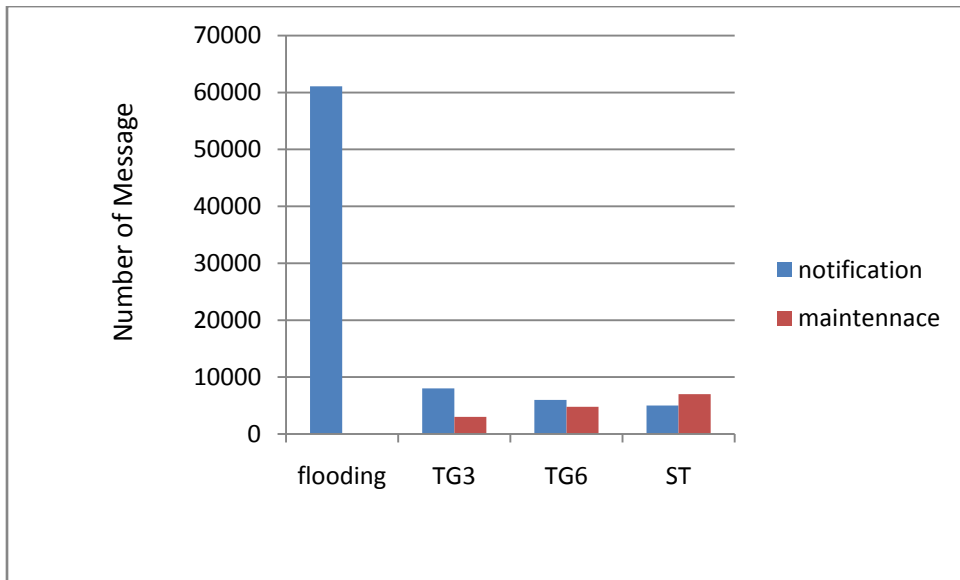


Figure 4 The Comparison of Message Overhead

is zero. For ST approach, each subscription is propagated through subscriber's neighbors to all other nodes along the subscription tree. The subscription overhead is large, although some subscriptions can be covered by others. The subscription overhead of our TG approach is light as the figure shows, because it saves memory overhead from the

following aspects: (1) Subscriptions are propagated only to parents not children within the local tree. (2) Nodes only subscribe to their local tree(s), and the size of each tree is much smaller compared to a global tree. (3) Bloom filter compresses the subscription and further reduces the subscription size.

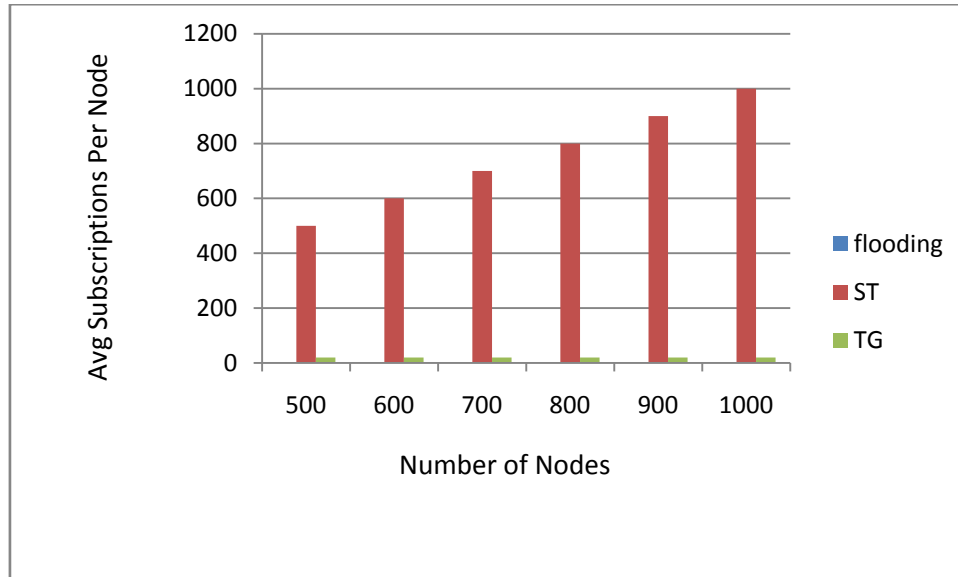
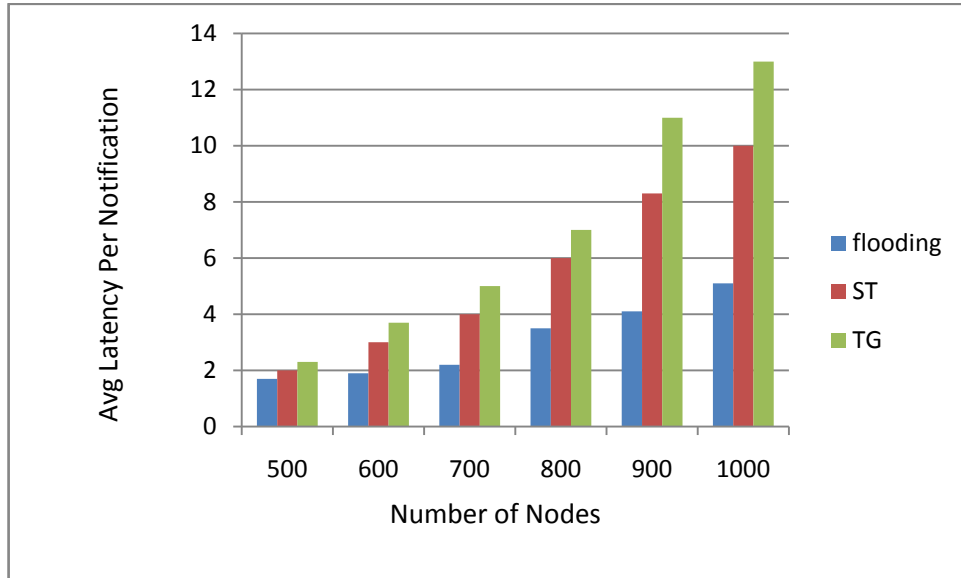


Figure 5 Average Memory Overhead per Node for Storing Subscriptions vs. the Network Size

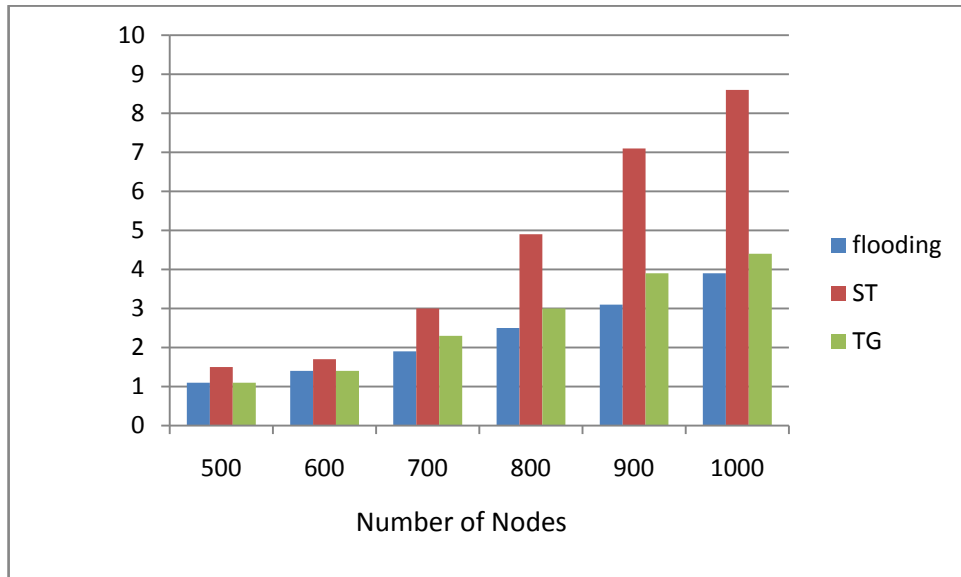
Fig. 6 shows the event notification latency for different pub/sub properties. I varied the ratio of different pub/sub events, which is 3:1 in the global events to local events in the experiment in Fig. 6 (a) and 1:3 in the experiment in Fig. 6 (b). It is obvious that flooding has the shortest latency for both cases because notifications go to all the possible ways to the subscribers, thus the shortest notification path is guaranteed.

Our approach achieves good latency performance in both cases. It achieves similar latency performance as ST in the first experiment, and it is better than ST and similar to

flooding in the second experiment. Our approach inherently has the locality preserving property because our group nodes based on their physical location.



(a) Global Notification: Local Notification=3:1



(b) Global notification: local notification=1:3

Figure 6 The Average Notification Latency vs. The Network Size

The simulation results demonstrate the advantages of our proposed pub/sub approach: it is scalable; it generates less network traffic; it consumes less memory storage; it is highly mobility-tolerant.

CHAPTER 5. CONCLUSIONS AND FUTURE WORK

Because of its decoupling and asynchrony properties, the publish/subscribe communication paradigm is inherently suited for dynamic wireless networks. In this thesis, I proposed an effective content-based publish/subscribe model to efficiently deliver events or advertisements to the interested individuals in a large-scale mobile ad-hoc community. An effective content-based information routing scheme is the key element for our scalable notification service in which nodes are dynamically clustered into limited-sized tree group. A compact subscription indexing scheme is proposed to enable efficient inside and outside tree-group event dissemination. The maintenance of the tree is simple and effective. The experimental evaluation shows that our approach is scalable and efficient.

In the future, I plan to enhance our experiments with real mobile wireless participants instead of simulation. In this way, I can evaluate the system in a more realistic environment. Our current system supports simple semantics that is between topic-based and content-based. I plan to study indexing scheme to support richer semantics.

REFERENCES

- [1] M. Avvenuti, A. Vecchio, and G. Turi. A “cross-layer approach for publish/subscribe in mobile ad-hoc networks. *Mobility Aware Technologies and Applications*”, volume 3744 of *Lecture Notes in Computer Science*, Springer, 2005, pp. 203–214.
- [2] J. Bacon, K. Moody, J. Bates, R. Hayton, C. Ma, A. McNeil, O. Seidel, and M. D. Spiteri, “Generic support for distributed applications.” *IEEE Computer*, vol. 33, no. 3, 2000, pp. 68–76.
- [3] R. Baldoni, C. Marchetti, A. Virgillito, “Content-based publish-subscribe over structured overlay networks”, in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, 2005, pp. 437-446.
- [4] C. Bettstetter and C. Wagner. “The spatial node distribution of the random waypoint mobility model”. In *Proc. WMAN*, 2002.
- [5] B. Bloom. “Space/time tradeoffs in hash coding with allowable errors.” *Communications of the ACM*, 1970.
- [6] M. Caporuscio, A. Carzaniga, and A. L. Wolf. “Design and evaluation of a support service for mobile and wireless publish/subscribe applications”. Technical Report CU-CS-944-03, Department of Computer Science and University of Colorado, Jan. 2003.
- [7] A. Carzaniga, D.S. Rosenblum, A.L. Wolf, “Design and evaluation of a wide-area event notification service.” *ACM Trans. Comput. Syst.*, vol. 19, no. 3, 2001, pp. 332–383.

- [8] A. Carzaniga, D.S. Rosenblum, A.L. “Wolf: Achieving scalability and expressiveness in an Internet-scale event notification service”. In: Proc. of ACM Symp. On Principles of Distributed Computing (PODC), Portland, OR , 2000, pp. 219-227
- [9] G. Cugola, E. D. Nitto, and A. Fuggetta, “The JEDI Event-Based Infrastructure and Its Application to the Development of the OPSS WFMS.” In IEEE Trans. Software Eng., vol. 27, no. 9, 2001, pp. 827–850.
- [10] D. Frey, G. Roman, “Context-aware publish/subscribe in mobile ad-hoc networks”. In ninth international conference on coordination models and languages, 2007, pp. 37-55.
- [11] Z. J. Haas , J. Y. Halpern , L. Li, “Gossip-based ad-hoc routing”. IEEE/ACM Transactions on Networking (TON), v.14 n.3, June 2006, p.479-491.
- [12] H. Liu, H.A. Jacobsen, “Modeling uncertainties in publish / subscribe”. In proceedings of the 20th International Conference on Data Engineering, March 30-April 02, 2004, pp.510-521.
- [13] L. Mottola, L. Cugola, G. Picco, “A self-repairing tree topology enabling content-based routing in mobile ad-hoc networks.” IEEE transaction on Mobile Computing, 2008, pp. 946-960.
- [14] G. Muhl, “Large-Scale Content-Based Publish/Subscribe Systems”. Phd thesis, Department of Computer Science, Darmstadt University of Technology (2002)

- [15] M. Petrovic, I. Burcea, H.A.Jacobsen, “S-ToPSS: Semantic Toronto publish/subscribe system.” In: Proc. of Conf. on Very Large Data Bases, (VLDB) Berlin, Germany, 2003, pp. 1101-1104.
- [16] C.E. Perkins and T.J. Watson, “Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers”, in Proc of ACM SIGCOMM, 1994. pp. 234-244.
- [17] C. Perkins, E. Belding-Royer and S. Das “Ad-hoc On-Demand Distance Vector (AODV) Routing”, RFC 3561, 2003.
- [18] P. Pietzuch and J. Bacon, “Hermes: A Distributed Event-Based Middleware Architecture.” Proceedings of the 22nd International Conference on Distributed Computing Systems, 2002, pp. 611 – 618.
- [19] C.G. Rezende, B. P. S.Rocha, A. A. F. Loureiro, "Publish/subscribe architecture for mobile ad-hoc networks", Proceedings of the 2008 ACM symposium on Applied computing, Fortaleza, Ceara, Brazil, 2008, pp. 1913-1917.
- [20] W. W. Terpstra, S. Behnel, L. Fiege, A. Zeidler, “A peer-to-peer approach to content-based publish/subscribe”, Proceedings of the 2nd international workshop on Distributed event-based systems, 2003, pp. 1-8.
- [21] P. Trianta llou, A. Economides, “Subscription summaries for scalability and efficiency in publish/subscribe”. In: Proc. of Workshop on Distributed Event-Based Systems, Vienna, Austria, 2002, pp. 619-624.

- [22] S. Yoo, JH. Son, MH. Kim, “A scalable publish/subscribe system for large mobile ad-hoc networks” *The Journal of Systems & Software*, Volume 82, Issue 7, July 2009, pp. 1152-1162.
- [23] P.T. Eugster, P. Felber, R. Guerraoui, and A.-M. Kermarrec, “The many faces of publish/subscribe,” *Tech. Rep. DSC ID:2000104*, EPFL, January, 2001.
- [24] Madhavi W. Subbarao, “Performance of Routing Protocols for Mobile Ad-Hoc Networks,” *Wireless Communication Technologies Group National Institute of Standards and Technology*.
- [25] X. Chen, J. Wu, “Multicasting techniques in mobile ad-hoc networks,” *The Handbook of Ad-hoc Wireless Networks (2003)* 25–40.
- [26] Muhl. G, “Large-Scale Content-Based Publish/Subscribe Systems.” *Phd thesis*.
- [27] W. Peng and X.-C. Lu, “On the reduction of broadcast redundancy in mobile ad-hoc networks.” In *Proc. of the 1st ACM Int. Symposium on Mobile Ad-hoc Networking & Computing (MobiHoc’00)*, pages 129–130, Piscataway, NJ, USA, 2000. ACM.
- [28] Carzaniga, A., Rutherford, M.J., Wolf, A.L, “A routing scheme for content-based networking.” *Proceedings of IEEE INFOCOM 2004*.
- [29] Muhl, G., Fiege, L., Gartner, F.C., Buchmann, A.P, “Evaluating advanced routing algorithms for content-based publish/subscribe systems”. In: *The Tenth IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2002)*. (2002) 167–176.

- [30] Y. Huang and H. Garcia-Molina. “Publish/subscribe in a mobile environment.” Proceedings of the Second ACM International Workshop on Data Engineering for Wireless and Mobile Access, 2001.
- [31] Y. Huang and H. Garcia-Molina. Publish/subscribe tree construction in wireless ad-hoc networks. In Proceedings of MDM, 2003.
- [32] Anceaume, E., Datta, A.K., Gradinariu, M., Simon, G. “Publish/subscribe scheme for mobile networks.” Proceedings of the 2002 Workshop on Principles of Mobile Computing (POMC 2002). (2002) 74–81
- [33] Picco, G.P., Cugola, G., Murphy, A.L. “Efficient content-based event dispatching in the presence of topological reconfiguration.” 23rd International Conference on Distributed Computing Systems (ICDCS 2003), 19-22 May 2003, Providence, RI, USA. (2003) 234–243
- [34] I. Burcea, H. Jacobsen, E. Lara, V. Muthusamy, and M. Petrovic. Disconnected operation in publish/subscribe middleware. In Proceedings of MDM, 2004.
- [35] Picco, G.P., Cugola, G., Murphy, A.L. “Efficient content-based event dispatching in the presence of topological reconfiguration.” 23rd International Conference on Distributed Computing Systems (ICDCS 2003), 19-22 May 2003, Providence, RI, USA. (2003) 234–243
- [36] Segall, B., Arnold, D., Boot, J., Henderson, M., Phelps, T. “Content Based Routing with Elvin4.” Proceedings of AUUG2K, Canberra, Australia. (June 2000)

- [37] Fabret, F., Jacobsen, A., Llirbat, F., Pereira, J., Ross, K., Shasha, D. “Filtering algorithms and implementation for very fast publish/subscribe.” Proceedings of the 20th Intl. Conference on Management of Data (SIGMOD 2001). (2001) 115–126
- [38] Campailla, A., Chaki, S., Clarke, E.M., Jha, S., Veith, H. “Efficient filtering in publish-subscribe systems using binary decision diagrams.” Proceedings of The International Conference on Software Engineering. (2001) 443–452
- [39] Bittner, S., Hinze, A. “On the benefits of non-canonical filtering in publish/subscribe systems.” Proceedings of the International Workshop on Distributed Event-Based Systems (ICDCS/DEBS’05). (2005)
- [40] Muhl, G. “Generic Constraints for Content-Based Publish/Subscribe.” Proceedings of the 6th International Conference on Cooperative Information Systems (CoopIS). (2001)
- [41] Gryphon Web Site: (<http://www.research.ibm.com/gryphon/>)
- [42] SIENA Web Site: (<http://www.cs.colorado.edu/users/carzanig/siena/>)
- [43] Cugola, G., Nitto, E.D., Fuggetta, A. “Exploiting an event-based infrastructure to develop complex distributed systems.” Proceedings of the 10th International Conference on Software Engineering (ICSE ’98). (April 1998)
- [44] Preotiuc-Pietro, R., Pereira, J., Llirbat, F., Fabret, F., Ross, K., Shasha, D. “Publish/subscribe on the web at extreme speed.” Proc. Of ACM SIGMOD Conf. on Management of Data, Cairo, Egypt (2000)

- [45] Gruber, R.E., Krishnamurthy, B., Panagosf, E. “The architecture of the ready event notification service.” Proceedings of The International Conference on Distributed Computing Systems, Workshop on Middleware. (Austin, Texas, 1999)
- [46] Pietzuch, P., Bacon, J. “Hermes: a distributed event-based middleware architecture.” Proceedings of the International Workshop on Distributed Event-Based Systems (DEBS’02). (2003)
- [47] Segall, B., Arnold, D. “Elvin Has Left the Building: A Publish /Subscribe Notification Service with Quenching.” Proc. of the 1997 Australian UNIX and Open Systems Users Group Conference. (1997)
- [48] Carzaniga, A., Rosenblum, D., Wolf, A.: Achieving Scalability and Expressiveness in an Internet-Scale Event Notification Service. In: Proceedings of the ACM Symposium on Principles of Distributed Computing. (2000) 219–227
- [49] Eugster, P., Felber, P., Guerraoui, R., Handurukande, S.: Event Systems: How to Have Your Cake and Eat It Too. In: Proceedings of the International Workshop on Distributed Event-Based Systems (DEBS’02). (2002)
- [50] Anceaume, E., Datta, A.K., Gradinariu, M., Simon, G. “Publish/subscribe scheme for mobile networks.” Proceedings of the 2002 Workshop on Principles of Mobile Computing (POMC 2002). (2002) 74–81
- [51] Picco, G.P., Cugola, G., Murphy, A.L. “Efficient content-based event dispatching in the presence of topological reconfiguration.” 23rd International Conference on

Distributed Computing Systems (ICDCS 2003), 19-22 May 2003, Providence, RI,
USA. (2003) 234–243