An efficient event delivery scheme in mobile ad hoc communities

Juan Li*

Department of Computer Science,
North Dakota State University,
Fargo, ND 58108, USA
E-mail: j.li@ndsu.edu
*Corresponding author

SameeUllah Khan

Department of Electrical and Computer Engineering,
North Dakota State University,
Fargo, ND 58108, USA
E-mail: samee.khan@ndsu.edu

Qingrui Li

Department of Computer Science,
North Dakota State University,
Fargo, ND 58108, USA
E-mail: Qingrui.Li@ndsu.edu

Abstract: Delivering events and advertisements to interested individuals is an important issue for mobile ad hoc communities. Providing a feasible solution to event delivery in highly dynamic and error prone mobile ad hoc environments is a challenging problem due to the inherent unreliable and unstable nature of mobile ad hoc networks. This paper proposes a novel semantics-based publish/subscribe scheme for efficient event and advertisement delivery in large-scale mobile ad hoc communities. In our proposed publish/subscribe system, mobile nodes are organised into groups and a compact semantics-based indexing is deployed in the groups. Efficient intra- and inter-group routings are proposed to facilitate efficient propagation of event notifications. The effectiveness of the system is demonstrated through a comprehensive set of simulation studies.

Keywords: publish/subscribe system; semantics; mobile wireless computing; routing.

Reference to this paper should be made as follows: Li, J., Khan, S. and Li, Q. (2013) 'An efficient event delivery scheme in mobile ad hoc communities', *Int. J. Communication Networks and Distributed Systems*, Vol. 10, No. 1, pp.25–39.

Biographical notes: Juan Li is an Assistant Professor at the Computer Science Department, North Dakota State University, Fargo, ND, USA. She received her BS degree from Beijing Jiaotong University, Beijing, China, in July 1997, and her PhD from the University of British Columbia, Vancouver, Canada, in May

2008. Her major research interest lies in distributed systems, including P2P networks, grid and cloud computing, mobile ad hoc network, social networking and semantic web technologies. She has published more than 40 papers in various areas of distributed systems. She is serving as the editorial board member of several journals such as the *International Journal of Computing & Information Technology* (IJCIT), *International Journal of Applied Research on Information Technology and Computing* (IJARITAC) and *The Journal of Convergence* (JoC). She is also serving (or has served) as technical programme committee members for many conferences.

SameeUllah Khan received his BS from GhulamIshaq Khan Institute of Engineering Sciences and Technology, Topi, Pakistan and his PhD from the University of Texas, Arlington, TX, USA in 2007. He is currently an Assistant Professor of Electrical and Computer Engineering at North Dakota State University, Fargo, ND, USA. He has extensively worked on the general topic of resource allocation in autonomous heterogeneous distributed computing systems. He is the Associate Editor of *Cluster Computing*, *International Journal of Communication Systems* and *Security and Communication Networks*.

Qingrui Li is a PhD student at the Computer Science Department, North Dakota State University, Fargo, ND, USA. He received his BS degree from Zhejiang University, Hangzhou, China, in 2002, and his MS degree from the National University of Singapore, Singapore, in 2004. His current research focuses on intelligent systems.

1 Introduction

A large percentage of personal mobile devices, such as cell phones, PDAs, and laptops are now equipped with wireless network interfaces, which paves the way to create mobile wireless communities. Mobile wireless communities are composed of autonomous wireless devices that are connected in a self-organised fashion. They can be constructed spontaneously for certain events or at locations, such as battlefield, rescue sites, conferences, expositions, and restaurants (Li and Khan, 2009). Within the communities, people can communicate and share resources/services through their mobile devices with minimum required infrastructure and without the need to have internet access. In order to locate relevant resources/services in such a community, effective communication mechanism is important. A publish/subscribe (pub/sub) model (Eugster et al., 2001) can be used to provide anonymous and asynchronous communications between community participants.

A pub/sub system connects resources/services publishers with subscribers by delivering published events from publishers to interested subscribers. Subscribers express their interests using a set of subscriptions. Publishers publish new information to the system using a set of publications. Upon receiving a publication, the system searches for matching subscriptions and notifies the interested subscribers. This communication model decouples time, space, and flow between publishers and subscribers. It can provide greater scalability and support a more dynamic network topology as required by mobile wireless communities. As an example, in a battlefield, devices or sensors are employed to get information about enemy troops. Allied troops interested in receiving such information can subscribe to their interested events. Whenever events occur, publishers, i.e., the monitoring devices 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 pub/sub paradigm is a natural fit for such scenarios.

Although the pub/sub communication paradigm exhibits many benefits and has been widely used in distributed systems, it is challenging to apply pub/sub system to mobile ad hoc communities due to the characteristic limitations of mobile devices (computing, power, and storage). Moreover, each mobile device in the community not only sends or receives but also routes data exchanged between devices. The communication between two devices normally needs multi-hop routing. Furthermore, devices are free to move within the community, which results in a continuously changing topology. These opportunistic and dynamic characteristics of the mobile ad hoc communities make communication a complex problem.

The complexity and diversity of resources/services in the community requires the corresponding expressiveness of the description of the resources/services. Content-based pub-sub is a more general and expressive paradigm, in which subscribers have the added flexibility of choosing interested content along multiple dimensions, rather than being restricted to pre-defined topics. On the other hand, scalable implementations of content-based pub/sub are difficult to realise. To support content-based pub/sub in mobile wireless networks, most existing works advocate maintaining a broadcast tree, which helps the registration of subscriptions and delivery of notification (Frey and Roman, 2007; Mottola et al., 2001; Yoo et al., 2009). However, 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 paper, we propose an efficient service/event delivery protocol that adopts pub/sub communication paradigm to mobile ad hoc communities. To reduce the overhead of maintaining a large subscription tree, the proposed pub/sub system groups mobile devices into clusters according to their geographical proximity. Subscription trees are constructed on top the clusters. Our design clearly addresses the following questions: How to divide the community network into tree clusters? How to implement event delivery based on the tree clusters? How to dynamically repair/reform the tree clusters?

The rest of this paper is organised as follows: Section 2 introduces the related work. Section 3 presents the proposed pub/sub protocol and related issues. Section 4 discusses the experimental results. Finally, conclusions are drawn in Section 5.

2 Related work

Pub/sub communication paradigm has been widely used for anonymous and asynchronous group communications in distributed systems. Most of the previous works on pub/sub (Bacon et al., 2000; Carzaniga et al., 2001; Cugola et al., 2001; Pietzuch and Bacon, 2002) focuses on increasing efficiency over traditional wired networks. These pub/sub systems can be implemented centrally or in a distributed manner. Centralised schemes (Petrovic et al., 2003; Liu and Jacobsen, 2004; Burcea et al., 2004) have a global view of the system, which enables efficient optimisations during the matching process. However, the central server can be a potential bottleneck and point of failure, which causes problems for system scalability and fault tolerance. Distributed pub/sub systems (Carzaniga and Rosenblum, 2000, 2001) have been introduced to address these problems. For example, P2P-based solutions (Terpstra et al., 2003; Baldoni et al., 2005) implement pub/sub in a fully decentralised manner. Most P2P-based pub/sub approaches construct a broker overlay to transmit messages. These approaches assume universal connectivity between any two nodes in the system, and at a roughly constant cost. Although this assumption is reasonable in fixed networks, it is not a realistic assumption in mobile wireless environment because it hides the fact that a unicast is actually implemented with multi-hop broadcast.

To implement the pub/sub communication in wireless environments, applications have to deal with many challenges, such as limited resources, mobility and intermittent connections. Existing pub/sub paradigms for mobile wireless networks can be grouped into three categories according to their routing methods:

a flooding

b gossiping

c selective routing.

Flooding is the simplest way to propagate information. Flooding is natural in mobile wireless network because of the broadcast nature of wireless communication. Obviously, flooding creates too much network traffic and it cannot be scalable for a large-scale mobile wireless community (Muhl, 2002).

In gossip-based protocols (Haas et al., 2006), 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 event routing during the movements of nodes. The drawback of this approach is the heavy redundancy and large message overhead (Rezende et al., 2008).

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. This category can be further divided into address-based routing schemes (Avvenuti et al., 2005; Frey and Roman, 2007) and content-based routing schemes (Yoo et al., 2009; Mottola et al., 2008).

In the address-based routing schemes, subscriber registers its interest and its address/ID to every node in 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 (Perkins and Watson, 1994) or AODV (Perkins et al., 2003).

In the content-based routing scheme, subscribers are notified based on their interests rather than their address/ID. Therefore, publishers do not need to address the routing issues when notifying the subscribers. This may save a lot of network traffic. The content-based schemes have to build dissemination trees to deliver publications to subscribers who are interested in the publications. With the tree structure, flooding is avoided because each node's subscription is kept in its neighbouring nodes. However, the scalability is restricted especially when the tree is large.

3 Event delivery scheme

The goal of this work is to efficiently deliver event notifications to all interested parties in a large-scale mobile ad hoc community in a timely manner. As explained earlier, most existing content-based notification systems for mobile wireless networks have to maintain a network-wide tree. The tree structure suffers from the large overhead for maintaining and updating the structure, especially when the network is large and highly dynamic, such as in a large-scale mobile ad hoc community. To avoid expensive long-range traffic and enhance availability by providing service locally, we partition a large wireless community into a number of smaller communities.

Groups are self-established and organised into a tree structure. Each node only communicates with its parent. In each tree group, there is a root node of the tree. Gateway nodes are nodes in the boundary of two groups. Gateway nodes manage communication with adjacent groups. A compact filter-based indexing scheme is deployed on top of the trees. Efficient protocols are proposed to deal with content-based routing within a group (intra-group) and between different groups (inter-group).

3.1 Location-based grouping and tree construction

Groups are formed to divide the network into manageable entities for efficient communication and low processing in the network. The grouping schemes result in a special type of node, called the group leader. It has a summarisation of the subscriptions within the group and it also communicates with other groups through gateway nodes that connect adjacent groups for cooperative notification. We adopt the heuristic weight function proposed in (Brust et al., 2007) as the criteria of selecting group leader. In particular, the weight of a device d is defined as:

$$W_d = wf_1 P_A + wf_2 s + wf_3 C_L + wf_4 \Delta dd$$

where P_A is the power-appropriateness of the device, s is the signal strength, C_L is the local clustering coefficient, Δdd is the dissemination degree, and wf_1, wf_2, wf_3, and wf_4 are weighing factors choosing according requirements.

A node joining the system will try to find a group to join. If there are no groups to join and the joining node is capable of being a group leader (based on the aforementioned weight) can it becomes a group leader and other nodes can join its group. Each node knows its group ID, the leader ID, and a level number representing its distance to the root in terms of the number of hops. Obviously, the level number of the root is 0. The level number is used to control the maximum level of the group tree.

Algorithm 1 illustrates how a node N_0 processes the ‘group joining’ message, when N_0 receives such a message from a node N_1. N_0 can add N_1 to its group by setting up a parent-child link, if the level number does not exceed the maximum limit. N_0 will notify N_1 about the group ID, the root ID, and set N_1’s level as N_0’s level plus one. Groups always try to connect to more neighbouring groups, therefore, node N_1 may already belongs to a group G_1 and this group G_1 will try to connect to N_0’s group G_0. In this case, N_1 will become a gateway node connecting its original group G_1 with the new joining group G_0.

In order to avoid creating too many small groups, periodically, we try to merge adjacent groups through boundary nodes. Assume node B is a boundary node connecting two groups G_0 (with leader L_0) and G_1 (with leader L_1). If both G_0 and G_1 are not saturated (based on the levels and number of nodes), and the merging of these two groups will not exceed the maximal limit of group level and member number, then the two groups will be merged through boundary node B. The merging decision is made by the two group leaders L_0 and L_1. The principle of the merging is to create a more balanced tree, i.e., the new leader of the merged tree should be one of the original two leaders, starting from whom, the tree is more balanced. Note: if there are multiple boundary nodes, the leader will pick one from them based on the principle to make the merged tree be more balanced. With this group tree construction and merging algorithm, we not only cluster nodes into groups but also construct an approximately balanced tree for each group.

Algorithm 1: group joining ()

*/*When node N_0 receives a group joining request from node N_1*/*

if *N_0 belongs to a group G_0 and N_0 level < maxLevel*
add N_1 to G_0, set N_1 parent = N_0, N_1 level = N_0 level + 1
else
    if *N1 belongs to a group G_1*
    add N_0 to G_1, set N_0 parent = N_1, N_0 level = N1 level + 1
    else
    N_0 and N_1 will create a group by selecting one as a group_leader
    endif
endif

3.2 Subscribing and publishing

With the tree structure constructed, nodes can subscribe their interests on the tree, and then publications can be efficiently propagated through the tree to nodes that are interested in the publications. Given the constrained nature of the mobile wireless environments, it is mandatory to aggregate the set of subscriptions into a compact set of content specifications. We implement the compact indexing with Bloom filters (Bloom, 1970).

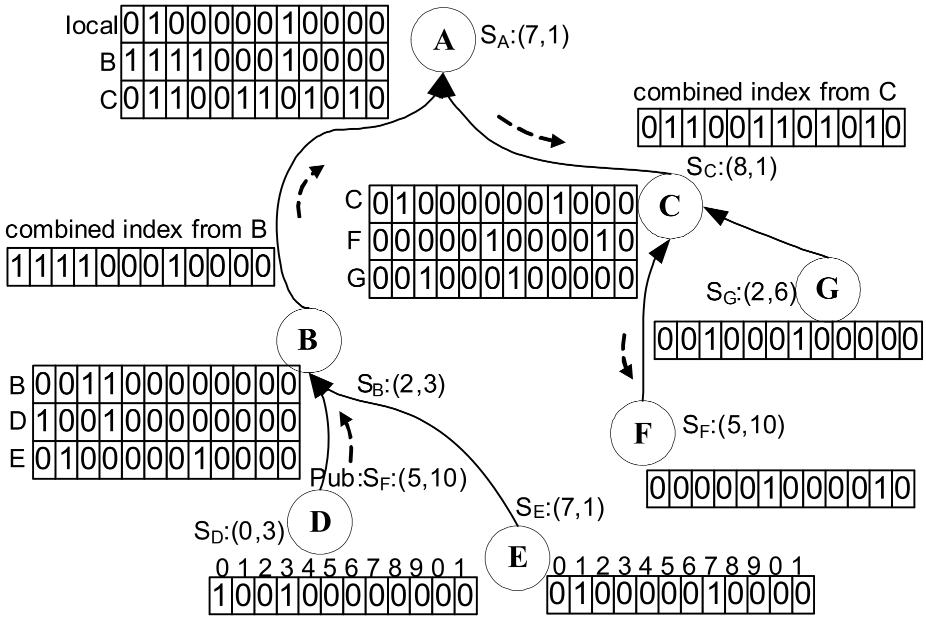
A Bloom filter is a compact randomised data structure for representing a set to support membership queries. For a set A composed of n elements: $\{a_1, a_2,\ldots,a_n\}$, a vector v of m bits, initially all set to 0, is allocated to it. Then k independent hash functions, h_1, $h_2,\ldots,h_k$, each with range $\{1,\ldots m\}$ are applied to every element of the set. For each element a in A, the bits at positions $h_1(a)$, $h_2(a),\ldots,h_k(a)$ in v are set to 1. A particular bit may be set to 1 multiple times. To determine if an element b is in the set A, we check the bits at positions $h_1(b)$, $h_2(b),\ldots, h_k(b)$. If any of them is 0, then b is certainly not in the set A. Otherwise, we conjecture that b is in the set although there is a certain probability that we are wrong. This is called a 'false positive'. The parameter k and m should be chosen such that the probability of a false positive is acceptable. There is a clear trade-off between m and the probability of a false positive rate, which can be estimated by: $(1- e^{-kn/m})^k$ (Bloom, 1970).

In our tree structure, every non-leaf node maintains a filter including several Bloom filter bitmaps: one bitmap for local subscriptions and the rest others for subscriptions from children branches. Each node sends the merged bitmap to its parent. So every internal node has a summarised view of a sub-tree rooted by itself. The root has a summarised view of the entire tree. Figure 1 illustrates the index aggregating and publication forwarding process. In this example, the Bloom bitmap size is 12 bits and two hash functions (H_1, H_2) 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 S_B is mapped to two positions: 2 and 3 in the bitmap ($H_1(S_B) = 2$, $H_2(S_B) = 3$). 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.

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, and if the publication is not received from its parent, 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 S_F. 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 cannot find match in its routing table either, then B 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.

Figure 1 Intra-group subscribing and publishing



3.3 Inter-group communication

As illustrated in the above example, our system can effectively deliver events/services inside a tree group. To deliver publications community wide, we need to route publications among groups. Since a group leader is also the root of every group tree and it has summarised index of the entire group, naturally, it becomes a representative of the group. Each group leader recodes the information of gateway nodes and then it can forward publications to adjacent groups through gateways. Periodically, gateways send updated information about adjacent group information to group leaders. To send a publication to adjacent groups, the publication is first forwarded to gateway nodes in the group and then delivered to adjacent groups through gateways. If there are several gateway nodes to the same group, the node that is closest to the group leader is chosen.

Many mobile wireless communications are mission-critical and time-critical. 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. Time-critical services/events normally are delivered only in local group or a few adjacent groups due to the location and time requirement. These events will be propagated from the publishers with limited hops. Our system can effectively support the need for mission-critical and time-critical communication by publishing and subscribing of two types of evens/services: local events/services or global events/services. Global events/services will be delivered to interested subscribers all over the network, while local events will be only delivered to local subscribers.

3.4 Dealing with mobility

The tree group structure is the routing foundation of the pub/sub communication paradigm. However, due to the high mobility of a mobile wireless community, the network topology keeps changing. How to maintain the tree group structure over the changing topology is important. In our scheme, parents and children nodes in a tree periodically send heartbeats between each other to identify if they are connected to the tree. When a node cannot connect to its parent, it tries to find another neighbour as its parent. The neighbour can be in different groups, but neighbours in the same group are preferred to become the new parent. To avoid loops in the tree, a node must avoid connecting to its descendents. If there are no such neighbours exist, this node will notify its children and let them find new parents using the same strategy. In this case, this original node can connect to its previous child/descendent and let it be its new parent. When a node N finds it loses its child, N will remove the filter from its child and notify this update to N's parent. In this way, the system repairs and reconfigures the tree group structure. At the same time, it avoids generating too many dispersed small groups.

3.5 Support of multi-attribute publishing/subscribing

Because of the use of hashing to map services/events to Bloom filter bit maps, the proposed strategy supports topic-based pub/sub best. Our system can also support multi-attribute content-based pub/sub. To add a disjunction of multi-attribute subscription to Bloom filters, we add each of attribute to the Bloom filter. 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. To add a conjunction of multi-attribute subscription to Bloom filters, we first order these attribute, for example alphabetically, and then we 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.

To map the disjunction of a multi-attribute publication, each of the attribute will be checked with the Boom filter, as long as one of them passes the filter, the publication will be forwarded to interested subscribers. To map the conjunction of a multi-attribute publication, we should order them and map all of the combinations of the attributes. 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, all of these seven combinations must be mapped with the Bloom filter bitmap. In this way, we can support conjunction and disjunction of multi-attribute publishing and subscribing.

3.6 False positive

As mentioned, the Bloom filter index may raise false positives and the aggregation of index may incur more false positives. Therefore, the index is only an approximation of the subscription. It may lead publications to nodes or branches that do not contain relevant subscriptions. Luckily, this will not affect the fidelity of the notification scheme, because node that finally receives the routed query will check its subscription database to further verify it. As long as the false positive rate is small, notifications will be routed along nearly optimal paths and most of the nodes that finally receive queries will in fact contain relevant subscriptions.

4 Experiment

We conducted a set of simulation experiments to evaluate our pub/sub event delivery scheme. In this section, we present the experimental setup and then analyse the results obtained.

4.1 Setup

An enclosed ad hoc network environment was considered. The enclosed area that contained different nodes was off an area of 2,000 m × 2,000 m. The density of the nodes was adjusted throughout the simulations. The mobility of the nodes was similar to that of the 'random waypoint' model as reported in (Bettstetter and Wagner, 2002). In the random waypoint model, initially, the nodes are randomly distributed within the enclosed area. Each node has a randomly picked destination, towards which, the node moves at a predetermined speed. Once a node reaches its destination, the node pauses for a predefined interval of time, and then it repeats this movement pattern. The transmission range of a node was predetermined to be 150 m. In the simulation, subscriptions and publications are represented as randomly-generated strings.

Table 1 Parameters used in the simulations

Parameter	*Range (default)*
Network size	500–1,000 (500)
Environment area	2,000 m × 2,000 m
Node moving speed	0–25 m/s (2 m/s)
Node transmission rage	150 m
Node pause time	0 s–50 s (10 s)
No. of attribute per subscription/publication	1–4
No. of subscriptions per node	1–3
No. 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

We also simulated two other protocols, flooding-based pub/sub and traditional subscription tree-based (ST) pub/sub routing (Yoo et al., 2009; Mottola et al., 2008) for performance comparison with our tree-group pub/sub routing scheme (TG). The first one is the simplest pub/sub routing approach which is commonly used as a baseline for comparison. The second ST approach ((Yoo et al., 2009; Mottola et al., 2008) constructs a subscription tree to subscribe interests and forward publications. It is one of the most popular approaches for content-based pub/sub. These three approaches were tested under the same physical topology and the same set of publishers and subscribers. 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

d notification latency.

The various simulation parameters and their default values are listed in Table 1.

4.2 Result

Figure 2 and Figure 3 compare the event delivery rate and message overhead of the three pub/sub schemes when nodes increase their moving speed. It is an undeniable fact that flooding can deliver more events notification compared to the other two techniques. However, the message overhead of the flooding technique is much higher compared to the other two techniques. Because it is difficult to keep 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 (TP) technique delivers comparative number of results and incurs much less message overhead compared to the other two techniques.

Figure 2 Successful event delivery rate vs. nodes' moving speed (see online version for colours)

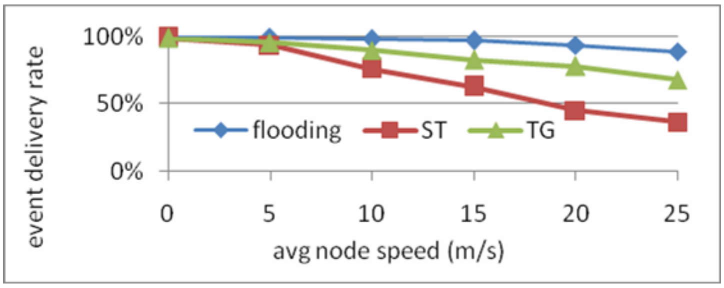


Figure 3 Message overhead vs. nodes' moving speed (see online version for colours)

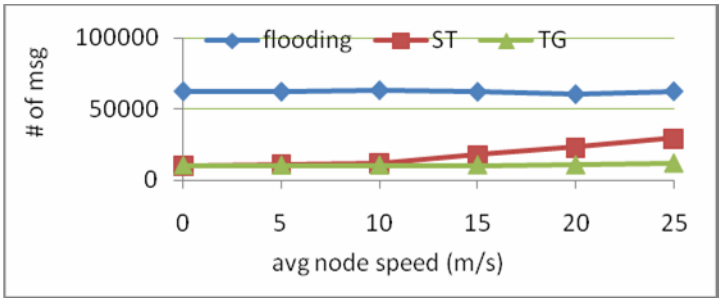


Figure 4 illustrates the message overhead that includes the publication notification and maintenance overhead to implement each of the routing protocols. In this experiment, our proposed tree group approach has two configurations with different 'preferred' tree level: 3 and 6 (TG3 and TG6 in the figure). We can see that our proposed TG approach uses the smallest overhead. Both the ST and the TG schemes use much less message overhead compared to flooding to propagate the same number of events. The figure also shows the composition of the overhead, i.e., the ratio of the notification overhead and maintenance overhead. All of the overhead of the flooding is caused by notification forwarding. For the traditional subscription tree scheme ST, the maintenance overhead accounts for a higher proportion, because it needs lots of overhead to maintain a large tree. By clustering nodes into groups we reduce the levels of the subscription trees, thus maintaining the tree is easier and costing less compared with a single subscription tree approach.

Figure 5 compares the memory overhead of storing subscriptions for the three pub/sub schemes. Because events are flooded to every node in the network, the flooding-based approach does not need to maintain nodes' subscriptions, thus the subscription overhead is zero. For ST approach, each subscription is propagated through subscriber's neighbours 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 trivial as the figure shows, because it saves memory overhead from the following aspects:

1 nodes only subscribe to their local tree(s), and the size of each tree is much smaller compared to a global tree

2 subscriptions are propagated only to parents not children within the local tree

3 Bloom filter compresses the subscription and further reduces the subscription size.

Figure 4 The comparison of message overhead (see online version for colours)

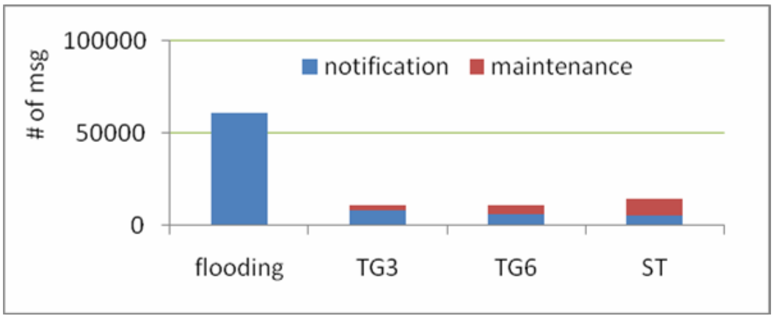


Figure 5 The average memory overhead per node for storing subscriptions vs. the network size (see online version for colours)

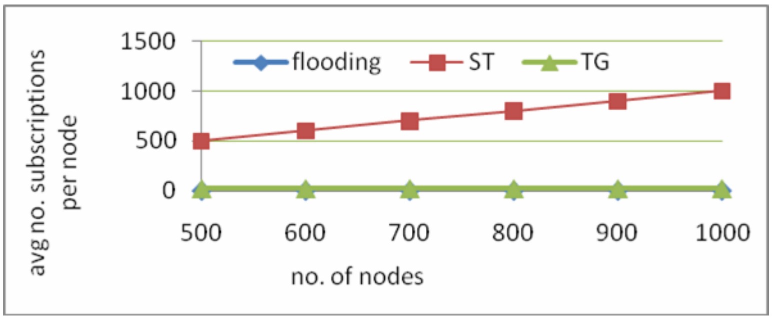
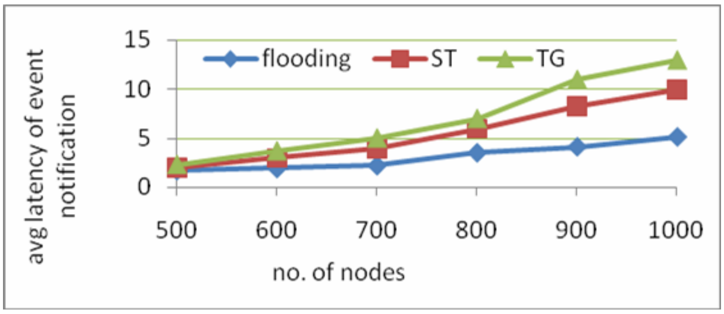


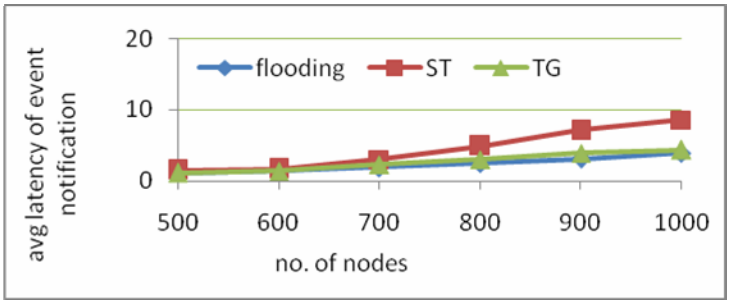
Figure 6 shows the event notification latency for different pub/sub properties. We varied the ratio of different pub/sub events. The ratio of global events to local events is 3:1 in the experiment in Figure 6(a) and 1:3 in the experiment in Figure 6(b). It is obvious that flooding has the shortest latency for both cases because notifications go 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 we group nodes based on their physical location.

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.

Figure 6 The average notification latency vs. the network size, (a) global notification: local notification = 3:1 (b) global notification: local notification = 1:3 (see online version for colours)



(a)



(b)

5 Conclusions

The publish/subscribe communication paradigm, because of its decoupling and asynchrony properties, is inherently suited for dynamic wireless networks. In this paper, we proposed an effective content-based publish/subscribe model to efficiently deliver events or advertisements in a large-scale mobile ad hoc community. The key for our scalable notification service is an effective content-based routing scheme. In this scheme, nodes dynamically form limited-sized tree structure. A compact subscription indexing scheme is proposed to enable efficient intra- and inter-tree-group event dissemination. The maintenance of the tree is simple and effective. The experimental evaluation shows that our approach is scalable and efficient.

As future work, further experiments and analysis will be performed regarding how different mobility models affect the overhead of tree construction and the behaviour of protocol responding to different patterns. Another extension will be made by studying the semantic relevance of publication and subscription to enable more intelligent semantics-based matching. We also plan to implement the protocol to test the system in more realistic environment. Then, we will implement the proposed system as a complete event-based middleware. To make it a realistic middleware, we need to work on issues such as reliability, persistence, fault-tolerance and security.

References

Avvenuti, M., Vecchio, A. and Turi, G. (2005) 'A cross-layer approach for publish/subscribe in mobile ad hoc network: mobility aware technologies and applications', *Lecture Notes in Computer Science*, Vol. 3744, pp.203–214, Springer, 2005.

Bacon, J., Moody, K., Bates, J., Hayton, R., Ma, C., McNeil, A., Seidel, O. and Spiteri, M.D. (2000) 'Generic support for distributed applications', *IEEE Computer*, Vol. 33, No. 3, pp.68–76.

Baldoni, R., Marchetti, C. and Virgillito, A. (2005) 'Content-based publish-subscribe over structured overlay networks', *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, pp.437–446.

Bettstetter, C. and Wagner, C. (2002) 'The spatial node distribution of the random waypoint mobility model', *Proc. WMAN.*

Bloom, B. (1970) 'Space/time tradeoffs in hash coding with allowable errors', *Communications of the ACM*, Vol. 13, No. 7, pp.422–426.

Brust, M.R., Andronache, A. and Rothkugel, S. (2007) 'WACA: a hierarchical weighted clustering algorithm optimized for mobile hybrid networks', *Proceedings of the Third International Conference on Wireless and Mobile Communications*, 04–09 March, p.23.

Burcea, I., Muthusamy, V., Petrovic, M., Jacobsen, H.A. and de Lara, E. (2004) 'Disconnected operations in publish/subscribe', *Proceedings of: IEEE Mobile Data Management*, pp.39–50.

Carzaniga, A., Rosenblum, D.S. and Wolf, A.L. (2000) 'Achieving scalability and expressiveness in an internet-scale event notification service', *Proc. of ACM Symp. on Principles of Distributed Computing (PODC)*, Portland, OR, pp.219–227.

Carzaniga, A., Rosenblum, D.S. and Wolf, A.L. (2001) 'Design and evaluation of a wide-area event notification service', *ACM Trans. Comput. Syst.*, Vol. 19, No. 3, pp.332–383.

Cugola, G., Nitto, E.D. and Fuggetta, A. (2001) 'The JEDI event-based infrastructure and its application to the development of the OPSS WFMS', *IEEE Trans. Software Eng.*, Vol. 27, No. 9, pp.827–850.

Eugster, P.T., Felber, P., Guerraoui, R. and Kermarrec, A-M. (2001) 'The many faces of publish/subscribe', Tech. Rep. DSC ID: 2000104, EPFL, January.

Frey, D. and Roman, G. (2007) 'Context-aware publish/subscribe in mobile ad hoc networks', *Ninth International Conference on Coordination Models and Languages*, pp.37–55.

Haas, Z.J., Halpern, J.Y. and Li, L. (2006) 'Gossip-based ad hoc routing', *IEEE/ACM Transactions on Networking (TON)*, June, Vol. 14, No. 3, pp.479–491.

Li, J. and Khan, S.U. (2009) 'MobiSN: semantics-based mobile ad hoc social network framework', *IEEE Global Communications Conference (Globecom 2009)*, Honolulu, HI, USA, December.

Liu, H. and Jacobsen, H.A. (2004) 'Modeling uncertainties in publish/subscribe', *Proceedings of the 20th International Conference on Data Engineering*, 30 March–02 April, pp.510–521.

Mottola, L., Cugola, L. and Picco, G. (2008) 'A self-repairing tree topology enabling content-based routing in mobile ad hoc networks', *IEEE Transaction on Mobile Computing*, Vol. 7, No. 8, pp.946–960.

Muhl, G. (2002) 'Large-scale content-based publish/subscribe systems', PhD thesis, Department of Computer Science, Darmstadt University of Technology.

Perkins, C., Belding-Royer, E. and Das, S. (2003) 'Ad hoc on-demand distance vector (AODV) routing', RFC 3561.

Perkins, C.E. and Watson, T.J. (1994) 'Highly dynamic destination sequenced distance vector routing (DSDV) for mobile computers', *Proc of ACM SIGCOMM*, pp.234–244.

Petrovic, M., Burcea, I. and Jacobsen, H.A. (2003) 'S-ToPSS: semantic Toronto publish/subscribe system', *Proc. of Conf. on Very Large Data Bases (VLDB)*, Berlin, Germany, pp.1101–1104.

Pietzuch, P. and Bacon, J. (2002) 'Hermes: a distributed event-based middleware architecture', *Proceedings of the 22nd International Conference on Distributed Computing Systems*, pp.611–618.

Rezende, C.G., Rocha, B.P.S. and Loureiro, A.A.F. (2008) 'Publish/subscribe architecture for mobile ad hoc networks', *Proceedings of the 2008 ACM Symposium on Applied Computing*, Fortaleza, Ceara, Brazil, pp.1913–1917.

Terpstra, W.W., Behnel, S., Fiege, L. and Zeidler, A. (2003) 'A peer-to-peer approach to content-based publish/subscribe', *Proceedings of the 2nd International Workshop on Distributed Event-based Systems*, pp.1–8.

Yoo, S., Son, J.H. and Kim, M.H. (2009) 'A scalable publish/subscribe system for large mobile ad hoc networks', *The Journal of Systems & Software*, July, Vol. 82, No. 7, pp.1152–1162.